

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การพัฒนาโปรแกรมประยุกต์ด้านมัลติมีเดียด้วยภาษาจาวา
MULTIMEDIA APPLICATION DEVELOPMENT USING JAVA



โดย

นายวิรัช วสันปิยมงคล

นายสิวณัฐ มาศสุรางค์

อาจารย์ที่ปรึกษา

ศส.บรรจง ปิยะธำรง

ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2542

เลขหน้.....
เลขทะเบียน 37048
วัน, เดือน, ปี 30 ธ.ค. 2543

รับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
หากมีการนำหนังสือนี้ไปใช้โดยไม่ได้รับอนุญาตให้ติดต่อแจ้งถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2542

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาโปรแกรมประยุกต์ด้านมัลติมีเดียด้วยภาษาจาวา

MULTIMEDIA APPLICATION DEVELOPMENT USING JAVA

ผู้จัดทำ

1. นาย วิรัช วศินปิยมงคล รหัสประจำตัว 40013269
2. นาย ศิวณัฐ ฆาตสุรางค์ รหัสประจำตัว 40013274



อาจารย์ที่ปรึกษา

(ศศ.บรรจง ปิยะธำรง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาแอปพลิเคชันด้านมัลติมีเดียด้วยภาษาจาวา

นายวิรัช วศินปิยมงคล 40013269

นายศิวณัฐ มาศสุรางค์ 40013274

ยศ บรรจง ปิยะธำรง

ปีการศึกษา 2542

บทคัดย่อ

ระบบ Oracle Video Server เป็นระบบที่มีความสามารถในการให้บริการเกี่ยวกับข้อมูลมัลติมีเดียเช่น Video-on-Demand มีองค์ประกอบหลักคือ วิดีโอเซิร์ฟเวอร์, คาตาเบสเซิร์ฟเวอร์ และไคลเอนต์ ทางฝั่งวิดีโอเซิร์ฟเวอร์ทำการติดตั้ง Oracle Video Server สำหรับจัดการข้อมูลมัลติมีเดีย และ Video Server Manager เพื่อเพิ่มประสิทธิภาพของระบบในการสร้าง Logical File ทางฝั่งคาตาเบสเซิร์ฟเวอร์ทำการเก็บข้อมูลที่จำเป็นสำหรับวิดีโอเซิร์ฟเวอร์และแอปพลิเคชันที่พัฒนาขึ้น และฝั่งไคลเอนต์จะทำการติดตั้ง Oracle Video Client เพื่อทำหน้าที่จัดการข้อมูลมัลติมีเดียที่มาจากวิดีโอเซิร์ฟเวอร์ให้กับแอปพลิเคชันที่พัฒนาขึ้นมา แอปพลิเคชันที่พัฒนาขึ้นมาก็จะติดตั้งอยู่บนส่วนนี้ด้วย

การพัฒนาแอปพลิเคชันด้านมัลติมีเดียด้วยภาษาจาวานั้น เพื่อให้แอปพลิเคชันสามารถทำงานร่วมกับระบบ Oracle Video Server ได้จึงจำเป็นที่จะต้องนำ Oracle Video JAVA Library มาใช้ ซึ่งไลบรารีนี้เป็นกลุ่มคลาสของจาวา กลุ่มคลาสเหล่านี้จะให้ความสามารถเช่น การควบคุมการเล่น, การโหลดไฟล์มัลติมีเดีย, การสร้างส่วนเชื่อมโยงกับผู้ใช้, การตรวจสอบสถิติของการส่งข้อมูลระหว่างเซิร์ฟเวอร์และไคลเอนต์ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Multimedia Application Development Using JAVA

Wirat Wasinpiyamongkol

Sivanath Massurang

Asst.Prof Banjong Piyatamrong Advisor

ABSTRACT

Oracle Video Server system has ability to manage multimedia information such as Video-on-Demand. It divide into 3 parts, which are Video Server, Database Server, and Client. Video Server will set up the Oracle Video Server for the managing of multimedia information and Video Server Manager, to increase the efficiency of Logical File Creation. Database Server will store the important information for Video Server and Developed Application. And finally Client will set up the Oracle Video Client, which manage the multimedia information from Video Server for developed application. The developed application will install in this last component, too.

We develop the multimedia application by JAVA, for the purpose that the developed application can work with the Oracle Video Server System. In this work we must use Oracle Video JAVA Library, a group class of JAVA, and this group class has many ability such as Control Playback, Load Multimedia File , User Interface, and Information Static Checking between Server and Client.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้คงจะไม่เสร็จสมบูรณ์ถ้าไม่ได้ อาจารย์ บรรจง ปิยะธำรง ซึ่งเป็นทั้งผู้ให้โอกาสและคำแนะนำต่างๆ รวมไปถึงอิสระในการทำงานแก่ผู้จัดทำ ผู้จัดทำขอขอบพระคุณอย่างยิ่งสำหรับทุกสิ่งที่ได้จากท่านผู้นี้

ขอบคุณหนังสือ Core Java ที่ทำให้ข้าพเจ้าได้เดินก้าวแรกในโลกของโปรแกรมเมอร์ และทราบถึงความมหัศจรรย์ของภาษาจาวา

ขอบคุณบริษัทออราเคิล (ประเทศไทย) ที่เอื้อเพื่อผลิตภัณฑ์ที่ใช้ในโครงการงาน

ขอบคุณคุณวัฒนพงษ์ วงศ์ตระกูล ที่ให้ความช่วยเหลือและคำปรึกษาในโครงการงาน

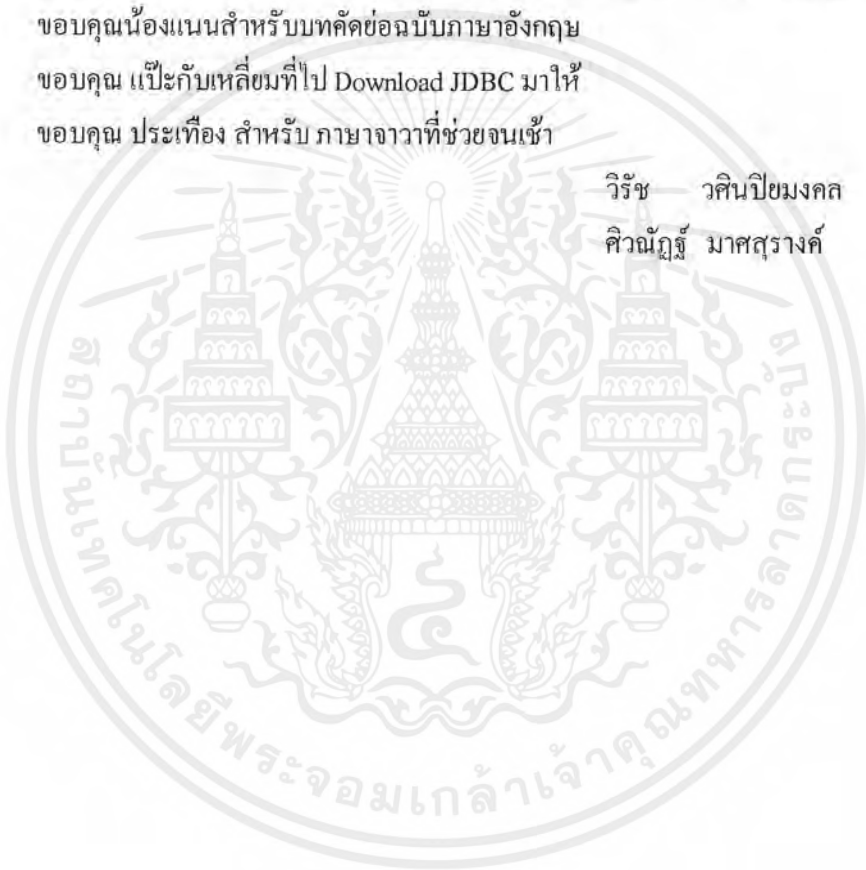
ขอบคุณน้องแนนสำหรับบทความฉบับภาษาอังกฤษ

ขอบคุณ เป๊าะกับเหลื้มที่ไป Download JDBC มาให้

ขอบคุณ ประเทือง สำหรับ ภาษาจาวาที่ช่วยจนเข้า

วิรัช วศินปิยมงคล

ศิวณัฐ มาศสุรางค์



สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VII
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตของโครงการ	1
1.4 วิธีการดำเนินงาน	2
บทที่ 2 ทฤษฎีและความรู้พื้นฐานเกี่ยวกับออบเจกต์	3
2.1 หลักการของออบเจกต์	3
2.1.1 ออบเจกต์	3
2.1.2 การนามธรรม	4
2.1.2.1 การนามธรรมสถานะ	4
2.1.2.2 การนามธรรมพฤติกรรม	4
2.1.3 โพลิมอร์ฟิซึม	5
2.1.4 คลาส	5
2.1.5 การซ่อนเร้นข้อมูล	5
2.1.6 ความสัมพันธ์ระหว่างคลาส	6
2.1.6.1 คลาสที่มีคลาสอื่นอยู่ภายใน	6
2.1.6.2 คลาสหนึ่งที่ใช้งานคลาสอีกคลาสหนึ่ง	6
2.1.6.3 คลาสหนึ่งสืบทอดคุณสมบัติจากคลาสอีกคลาสหนึ่ง	6
บทที่ 3 ทฤษฎีและความรู้พื้นฐานเกี่ยวกับ Oracle Video Server	7
3.1 สถาปัตยกรรมของ Oracle Video Server	7
3.1.1 ส่วนของไคลเอนต์	7
3.1.2 ส่วนแอปพลิเคชันเซิร์ฟเวอร์	8
3.1.3 ส่วนดาตาเบสเซิร์ฟเวอร์	8
3.2 ส่วนประกอบของระบบ Oracle Video Server	8
3.2.1 Oracle Video Server	8
3.2.1.1 คอนเทนส์	8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้าที่
3.2.1.2 Oracle Media Data Store	10
3.2.2 Oracle Video Server Manager	12
3.2.3 Oracle Media Net	13
3.2.4 Oracle Video Client	13
3.2.4.1 Oracle Video Client	14
3.2.4.2 Client Interface	14
3.2.4.3 Client Software Components	14
3.2.4.4 Oracle Video JAVA Library	15
3.2.5 Oracle Database Server	15
3.3 ฮาร์ดแวร์ของระบบ Oracle Video Server	16
3.4 ลักษณะของการติดต่อระหว่างส่วนประกอบภายในระบบ OVS	16
3.5 ระบบเครือข่ายในระบบ Oracle Video Server	18
3.6 กระบวนการทำงานของไคลเอนต์ต่อเซิร์ฟเวอร์	18
3.7 รูปแบบของ OGF	19
3.8 การตรวจสอบการเกิด Discontinuity ในข้อมูล	20
บทที่ 4 ทฤษฎี JDBC และ JDBC Drivers ของ Oracle	23
4.1 ความหมายของ JDBC	23
4.2 โครงสร้างของ JDBC	23
4.3 รูปแบบของ JDBC ไคล์ฟเวอร์	23
4.4 JDBC Drivers ของ Oracle	24
บทที่ 5 การปฏิบัติและการออกแบบแอปพลิเคชัน	25
5.1 การพัฒนาไคลเอนต์เซิร์ฟเวอร์	25
5.1.1 ติดตั้งลอจิกคอนเทนเนอร์	25
5.1.2 การเพิ่มวอลุ่ม	26
5.1.3 การลงทะเบียนคอนเทนเนอร์	26
5.1.4 การสร้างตัวอย่างคอนเทนเนอร์	27
5.1.5 กลไกการทำงานของระบบโดยรวม	27
5.2 การพัฒนาไคลเอนต์	28
5.2.1 Use Case Diagram	29
5.2.2 Class Diagram	29
5.2.3 Collaboration Diagram	31
5.2.4 อธิบายการทำงานและอิมพลีเมนต์	32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้าที่
5.2.4.1 คลาส OVCPlayer	32
5.2.4.2 คลาส AboutDialog	35
5.2.4.3 คลาส LoadDialog	35
5.2.4.4 คลาส LoadFile	36
5.2.4.5 คลาส ConnectDB	36
5.2.4.6 คลาส NetworkSet	37
5.2.5 คาตาเบส	37
5.2.6 การทดลองและผลการทดลอง	38
บทที่ 6 บทสรุปและวิจารณ์	44
ภาคผนวก ก.	45
ภาคผนวก ข.	58
ภาคผนวก ค.	62
ภาคผนวก ง.	68
ภาคผนวก จ.	82
ภาคผนวก ฉ.	84
บรรณานุกรม	103



สารบัญรูป

	หน้าที่
รูปที่ 3-1 แสดงสถาปัตยกรรมของ Oracle Video Sever	7
รูปที่ 3-2 แสดงความสัมพันธ์ของข้อมูลภายในระบบ Oracle Video Server	9
รูปที่ 3-3 แสดงสติปหนึ่งสติปบนดิสก์	10
รูปที่ 3-4 แสดงวอลลุ่มของ MDS ที่มีขนาด RAID 4	11
รูปที่ 3-5 Software Component ของ Oracle Video Client	15
รูปที่ 3-6 แสดงฮาร์ดแวร์ของระบบ Oracle Video Server	16
รูปที่ 3-7 แสดง Session ในระบบเครือข่ายแบบ Symmetric	17
รูปที่ 3-8 แสดง Session ในระบบเครือข่ายแบบ Asymmetric	18
รูปที่ 3-9 แสดงการส่งข้อมูลของเซิร์ฟเวอร์ไปยังไคลเอนต์	19
รูปที่ 3-10 แสดงรายละเอียดของ Oracle Generic Framing header	20
รูปที่ 4-1 แสดงระดับการเชื่อมต่อของ JDBC API	24
รูปที่ 5-1 แสดง Use Case Diagram ของระบบ	29
รูปที่ 5-2 แสดงคลาสไดอะแกรม	30
รูปที่ 5-3 แสดงลำดับของการไหลคไฟล์มัลติมีเดียจากเซิร์ฟเวอร์	31
รูปที่ 5-4 แสดงลำดับของการสร้างออบเจกต์ Player	31
รูปที่ 5-5 แสดงลำดับของการควบคุมการเล่น	32
รูปที่ 5-6 แสดงหน้าจอหลักของแอปพลิเคชัน	38
รูปที่ 5-7 แสดงเมนู File	39
รูปที่ 5-8 แสดงหน้าจอที่เกิดจากการเลือกเมนู Load	40
รูปที่ 5-9 แสดงไฟล์ไดอะล็อกของระบบวินโดว	40
รูปที่ 5-10 แสดงรายการของไฟล์มัลติมีเดียที่เลือกจากไฟล์ไดอะล็อก	41
รูปที่ 5-11 แสดงการ Play	41
รูปที่ 5-12 แสดงเมนู Playback	42
รูปที่ 5-13 แสดงเมนู Preference	43
รูปที่ 5-14 แสดงเมนู Network Setting	43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มา

ด้วยความก้าวหน้าของเทคโนโลยีผสมผสานกับการประยุกต์ใช้งานอย่างเหมาะสม ทำให้เทคโนโลยีเข้ามามีส่วนร่วมในชีวิตประจำวันมากขึ้น การบริการข้อมูลมัลติมีเดียเป็นบริการหนึ่งในด้านบันเทิงภายในบ้าน ที่ได้นำเอาเทคโนโลยีอินเทอร์เน็ต ซึ่งเป็นระบบเครือข่ายคอมพิวเตอร์ที่มีขนาดใหญ่มาใช้เป็นสื่อกลางในการบริการข้อมูลมัลติมีเดียสู่ลูกค้า เนื่องจากระบบเครือข่ายอินเทอร์เน็ตมีพื้นที่การให้บริการครอบคลุมทั่วทุกมุมโลก อีกทั้งเป็นสื่อกลางที่ลดข้อจำกัดทางค่านเวลา และสามารถตอบสนองความต้องการของลูกค้าได้อย่างเหมาะสมกับค่าใช้จ่ายที่เสียไป

ระบบเครือข่ายอินเทอร์เน็ตเป็นเครือข่ายที่ประกอบขึ้นจากเครือข่ายเล็กๆทั่วโลกที่ต่อถึงกัน เพื่อจุดประสงค์หลักคือ การแบ่งปันข้อมูลซึ่งกันและกัน เพราะฉะนั้นสิ่งหนึ่งที่เป็นปัญหาที่ไม่สามารถหลีกเลี่ยงได้คือ การที่เครือข่ายเล็กๆเหล่านั้นจะมีสภาพแวดล้อมที่แตกต่างกัน ทั้งในด้านฟิสิกส์คอลและลจิกคอลล ทั้งนี้ทั้งนั้นขึ้นอยู่กับข้อกำหนดและมาตรฐานที่เครือข่ายเหล่านั้นใช้ ดังนั้นสิ่งที่ต้องคำนึงถึงเมื่อจะให้บริการผ่านทางอินเทอร์เน็ตคือ จะทำอย่างไรจึงจะสามารถเข้าถึงลูกค้าได้ในสภาพแวดล้อมที่แตกต่างกัน โดยเสียค่าใช้จ่ายต่ำสุด เทคโนโลยีหนึ่งที่เกิดขึ้นมาเพื่อขจัดปัญหาเหล่านี้คือ “เทคโนโลยีภาษาจาวา”

ภาษาจาวาเป็นภาษาที่ถูกเลือกใช้เพื่อพัฒนาแอปพลิเคชันในโครงการนี้ เพราะภาษาจาวามีคุณสมบัติการโปรแกรมเชิงวัตถุและจะไม่ยึดติดกับรูปแบบของเครื่องที่ใช้รันโปรแกรม ไม่ว่าจะเป็นทางฮาร์ดแวร์หรือซอฟต์แวร์

1.2 วัตถุประสงค์ของโครงการ

1. ศึกษาภาษาจาวาในการพัฒนางานด้านมัลติมีเดีย
2. ศึกษาหลักการและการทำงานของ Oracle Video Server
3. ศึกษาหลักการและการทำงานของ Oracle Database
4. พัฒนาแอปพลิเคชันที่ใช้งานข้อมูลมัลติมีเดีย

1.3 ขอบเขตของโครงการ

1. ให้ได้มาซึ่งแอปพลิเคชันสาริตของ Video-on-Demand
2. ให้ได้มาซึ่งคู่มือที่อธิบายการติดตั้ง หลักการและการใช้งานแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 วิธีการดำเนินงาน

เบื้องต้นจะต้องทำการศึกษาค้นคว้าชนิดข้อมูลมัลติมีเดียแบบต่างๆ เพื่อนำไปพิจารณาร่วมกับสภาพแวดล้อมที่ใช้ในการพัฒนาแอปพลิเคชัน ไม่ว่าจะเป็นด้านภาษาโปรแกรมมิ่ง ด้านระบบปฏิบัติการ หรือด้านของระบบเครือข่าย โครงการนี้เลือกใช้ Oracle Video Server เป็นเครื่องมือในการพัฒนาแอปพลิเคชัน เนื่องมาจากมีการสนับสนุนการพัฒนาแอปพลิเคชันด้วยภาษาจาวา และยังรองรับระบบเครือข่ายแบบต่างๆ ได้พอสมควร รวมถึงสามารถใช้งานร่วมกับฐานข้อมูลได้อีกด้วย หลังจากทำการกำหนดเครื่องมือที่จะใช้ช่วยในการพัฒนาแอปพลิเคชันแล้ว ขั้นตอนต่อไปคือการศึกษาและใช้งานเครื่องมือ ทั้งในด้านการเรียกใช้งานไลบรารีภาษาจาวา ด้านการติดตั้งและการควบคุมในทางเซิร์ฟเวอร์ ทั้งหมดนี้จะทำไปพร้อมๆ กับการกำหนดขอบเขต และออกแบบแอปพลิเคชันในด้านไคลเอนต์ กำหนดความสามารถของแอปพลิเคชัน หลังจากทำการออกแบบและกำหนดความสามารถในการทำงานของแอปพลิเคชันแล้ว ขั้นตอนต่อไปคือการเขียนโปรแกรม และทดสอบโปรแกรมว่ามีความสามารถตรงกับที่ตั้งขอบเขตไว้หรือไม่



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและความรู้พื้นฐานเกี่ยวกับออบเจกต์

โดยในบทนี้จะกล่าวถึงในเรื่องของทฤษฎีและความรู้พื้นฐานของออบเจกต์ที่จำเป็น เพื่อที่จะนำไปใช้ในการศึกษาและทำความเข้าใจกับแอปพลิเคชันที่พัฒนาขึ้นมา

2.1 หลักการของออบเจกต์ (Object-Orient Concept)

หลักการของออบเจกต์เป็นแนวความคิดหนึ่งในการวิเคราะห์ ออกแบบและเขียนโปรแกรมของระบบที่มีขนาดใหญ่และซับซ้อน เพื่อให้สามารถพัฒนาและแก้ไขโปรแกรมได้ง่ายและมีความเป็นอิสระในการปรับปรุงแก้ไข ซึ่งมีหลักการดังนี้

2.1.1 ออบเจกต์ (Object)

ออบเจกต์ หรือ วัตถุ คือสิ่งที่เรากำหนดจากนามธรรมขึ้นมาเป็นรูปธรรมที่มีอยู่จริง โดยสิ่งที่มีอยู่จริงไม่จำเป็นต้องจับต้องได้เท่านั้นแต่ยังรวมไปถึงความสามารถในการแสดงลักษณะใดๆก็ได้ซึ่งคุณสมบัติเหล่านี้เป็นคุณสมบัติเฉพาะของแต่ละวัตถุซึ่งสามารถเปลี่ยนแปลงไปมาได้ เช่น โด๊ยะจะต้องมี 4 ขา นี่คือการนิยามเป็นนามธรรม(Abstraction)ของโด๊ยะที่เรากำหนดขึ้น ดังนั้น โด๊ยะญี่ปุ่นหรืออื่นๆคือสิ่งที่เราเรียกว่าโด๊ยะ เพราะสิ่งที่เรากล่าวมานั้นมีลักษณะเหมือนกันคือมี 4 ขา แต่หากเป็นโด๊ยะที่เรากำลังใช้เขียนหนังสือหรือที่เราจริงๆจะไม่ใช้นามธรรมแล้ว แต่เป็นโด๊ยะที่มีอยู่จริง ดังนั้นโด๊ยะเหล่านี้คือ ออบเจกต์ นั่นเองโดยส่วนประกอบภายในของออบเจกต์นั้นจะมีการกำหนดสถานะ (State) และเมธอด (Method)เอาไว้ด้วย

สถานะของออบเจกต์คือสภาวะการทำงานที่ภายในออบเจกต์นั้นมีอยู่ซึ่งออบเจกต์อื่นจะไม่สามารถมองเห็นได้ และเมธอดคือการทำงานของออบเจกต์นั้นว่ามีการทำงานอะไรบ้างและเป็นส่วนที่ออบเจกต์อื่นสามารถเห็นได้

ลักษณะของออบเจกต์อีกอย่างที่เราให้ความสำคัญเป็นอย่างมากคือการสร้างส่วนที่ใช้ติดต่อหรือเชื่อมต่อ (Interface) กันระหว่างออบเจกต์โดยที่ออบเจกต์แต่ละตัวจะไม่สามารถเข้าไปเกี่ยวข้องกับการทำงานของออบเจกต์อื่นได้ และไม่จำเป็นต้องรู้ว่าภายในออบเจกต์นั้นทำงานอย่างไรแต่ให้ความสนใจในการเชื่อมต่อหรือติดต่อกับออบเจกต์นั้นเท่านั้น ซึ่งสิ่งที่ต้องรู้คืออินพุต (Input) เพื่อให้ได้เอาท์พุต (Output) ตามที่ต้องการเท่านั้นซึ่งเราจะเรียกข้อมูลที่ส่งใช้ในการเชื่อมต่อว่าเมสเสจ (Message) การเชื่อมต่อนี้จะทำให้เราสามารถซ่อนรายละเอียดต่างๆของออบเจกต์ได้ยกตัวอย่างเช่น เราสามารถดูทีวี เปลี่ยนช่องทีวีได้โดยที่เราไม่ต้องรู้ว่าทีวีมีทำงานอย่างไรเลย เราสนใจเพียงให้ทีวีทำงานตามที่เรต้องการเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 การนามธรรม (Abstraction)

การนามธรรมคือการกำหนดขอบเขตหรือนิยามขึ้นมาให้เด่นชัดว่าสิ่งที่เรากำหนดนั้นคืออะไร แตกต่างกับสิ่งอื่นอย่างไรบ้าง โดยข้อกำหนดต่างๆที่กำหนดขึ้นจะพิจารณาจากความเป็นจริงของสิ่งหลายๆสิ่งที่จัดอยู่ในประเภทเดียวกันว่ามีความเหมือนกันที่ใด เช่น การทำงาน, การใช้งาน และอาจมองข้ามสิ่งแตกต่างเล็กน้อย ซึ่งการกำหนดขอบเขตดังกล่าวจะมีเกี่ยวข้องกับมุมมองของผู้พิจารณาด้วยว่าต้องการแสดงให้เห็นถึงสิ่งนั้นในด้านใดเช่น เรากำหนดว่าโต๊ะต่างๆที่เราใช้อยู่มีลักษณะเหมือนกันที่ใดบ้าง เช่น พื้นโต๊ะเป็นสี่เหลี่ยมหรือวงกลม, ชนิดของไม้ที่ใช้ทำเป็นต้น

เราสามารถกำหนดรายละเอียดเจาะจงลงไปได้ว่าออบเจกต์แต่ละตัวมีจุดเด่นอย่างไร จะเห็นได้ว่าการทำนามธรรมสิ่งต่างๆที่ดีได้นั้นจะช่วยให้เราไม่ต้องมีการทำงานที่ซับซ้อน ไม่ต้องมีการสร้างใหม่หลายครั้ง และขุดเอาหลายๆสิ่งมาเป็นสิ่งเดียวกัน แต่จะยากในเรื่องการมองสิ่งต่างๆมากำหนดเป็นนามธรรมการทำนามธรรมจะเป็นส่วนที่สำคัญมากในเรื่องของการออกแบบโปรแกรมแบบออบเจกต์ แบ่งออกเป็น 2 ชนิดคือ

2.1.2.1 การนามธรรมสถานะ(State)

เป็นการนามธรรมสิ่งที่เกี่ยวข้องกับการทำงานของออบเจกต์ว่าแต่ละ ออบเจกต์มีการทำงานอะไรบ้าง ใช้ข้อมูลอะไรบ้างในการทำงาน(แต่ไม่กล่าวถึงว่ามีการทำงานอย่างไร) เมื่อออบเจกต์ได้รับเมสเสจเข้ามาทางส่วนของการเชื่อมโยงแล้ว ออบเจกต์นั้นจะเอาข้อมูลที่ต้องการมาไว้ภายในออบเจกต์เพื่อให้ออบเจกต์อื่นไม่สามารถทราบได้ว่าออบเจกต์นั้นเอาข้อมูลไปทำอะไร โดยเราจะเรียกการซ่อนรายละเอียดของการทำงานกับข้อมูลเช่นนี้ว่า Encapsulation ซึ่งในส่วนนั้นเหมือนกับการเตรียมข้อมูลภายในการทำงานของออบเจกต์

การกำหนดการเข้าถึงข้อมูลภายในของออบเจกต์จะทำ รแยกส่วนของ การติดต่อกับข้อมูลภายนอกและวิธีการทำงานออกจากกัน พร้อมทั้งทำการซ่อนการทำงานและรายละเอียดของข้อมูลไว้ภายใน การ Encapsulation ทำอย่างนี้เพื่อไม่ให้มีการเข้าถึงข้อมูลได้โดยตรงอีก นั่นคือการซ่อนข้อมูล (Data Hiding) นั่นเอง ซึ่งการเข้าถึงข้อมูลนั้นต้องเรียกผ่านเมธอดเท่านั้น โดยการซ่อนข้อมูลนี้จะมองว่าออบเจกต์แต่ละตัวได้นำเอาข้อมูลและฟังก์ชันที่เกี่ยวข้องมารวมกันโดยมีข้อมูลที่ถูกซ่อนอยู่ได้ดังนั้นจะ เห็นว่าเราจะ เปลี่ยนแปลงส่วนของโปรแกรมได้โดยที่ไม่ต้องเปลี่ยนส่วนเชื่อมต่อทำให้โปรแกรมยังคงทำงานได้เหมือนเดิม แม้รายละเอียดภายในได้เปลี่ยนไปแล้ว

ข้อดีอีกประการหนึ่งของการซ่อนเร้นข้อมูลคือ แบ่งระดับการเข้าถึงข้อมูลได้ด้วยตนเอง โดยจะใช้เมธอดของออบเจกต์นั้นเป็นตัวแบ่งระดับการเข้าถึงข้อมูล

2.1.2.2 การนามธรรมพฤติกรรม(Abstraction of behavior)

เป็นการนามธรรมส่วนที่จะใช้ในการเชื่อมโยงกับออบเจกต์อื่น เพื่อให้ออบเจกต์อื่นสามารถทราบว่าหากต้องการให้ออบเจกต์นี้ทำงานหรือติดต่อกับออบเจกต์นี้จะต้องติดต่อกับส่วนการเชื่อมโยง (Interface) ที่เรากำหนดไว้เท่านั้นจึงจะสามารถทำงานตามหน้าที่ของออบเจกต์นี้ได้

การทำงานของพฤติกรรม(Behavior) ของแต่ละออบเจกต์ที่สามารถเรียกใช้งานออบเจกต์อื่นเพื่อเอาผลลัพธ์ที่ได้จากออบเจกต์ตัวนั้นมาใช้งานต่อไป หรือยอมให้มีการใช้งานร่วมระหว่างออบเจกต์(Collaboration Among Object) ใต้นั้นเอง

2.1.3 โพลิมอร์ฟิซึม (Polymorphism)

คำว่าโพลิมอร์ฟิซึมแปลว่าหลายรูปแบบ ในทางการออกแบบโปรแกรมแบบออบเจกต์จะให้ความหมายของคำนี้คือแต่ละออบเจกต์อาจที่จะใช้ชื่อในการเชื่อมต่อของแต่ละออบเจกต์นั้นเหมือนกันก็ได้ แต่การทำงานภายใน (Behavior) ของชื่อที่ใช้ในการเชื่อมต่อของแต่ละออบเจกต์นั้นไม่เหมือนกันก็ได้ซึ่งในการทำงานของโปรแกรมจะเป็นตัวเลือกหรือจัดการเองว่าจะเอา ออบเจกต์ใดที่ชื่อเหมือนกันในขณะนั้นมาทำงาน ยกตัวอย่างของโพลิมอร์ฟิซึมเช่นการเปลี่ยนเกียร์ของรถยนต์ คือชื่อการทำงานที่เรากำหนดไว้ในส่วนของการทำการเชื่อมโยงหรือติดต่อกับออบเจกต์ 2 ตัวที่ใช้ชื่อนี้ในการติดต่อก็คือ อัตโนมัต(Auto) กับ แบบมือ (Manual) ซึ่งการเปลี่ยนเกียร์ของเกียร์แบบอัตโนมัติจะใช้เครื่องยนต์ควบคุมการทำงาน แต่การเปลี่ยนเกียร์แบบมือจะใช้คนควบคุมเป็นต้น

2.1.4 คลาส

คลาสคือสิ่งที่เราจะต้องกำหนดขึ้นตามที่ได้นามธรรมที่เราวางไว้เพื่อให้แสดงถึงออบเจกต์ต่างๆต่อไป ภายในคลาสจะต้องมีการกำหนดการติดต่อ หรือการเชื่อมโยงมีการกำหนดโครงสร้างของข้อมูลที่ใช้ในออบเจกต์และรายละเอียดที่เกี่ยวกับการทำงานของเมธอดโดยเราสามารถที่จะสร้างออบเจกต์หลายๆออบเจกต์จากคลาสเพียงคลาสเดียวได้ โดยแต่ละออบเจกต์ที่เกิดจากคลาสนี้จะเรียกว่าอินสแตนซ์ (Instance) และในส่วนการทำงานภายในของออบเจกต์ที่เกิดจากคลาสนี้จะเรียกว่า Instantiation

2.1.5 การซ่อนเร้นข้อมูล (Encapsulation)

จากที่ได้อธิบายมาข้างแล้วเกี่ยวกับออบเจกต์ที่มีความสามารถในการซ่อน เร้นข้อมูลต่างๆภายในตัวมันเองจากออบเจกต์อื่นๆได้ซึ่งคุณสมบัตินี้เองมีประโยชน์เมื่อเราทำการเปลี่ยนแปลงสิ่งต่างๆภายในออบเจกต์แล้วจะไม่มีผลกระทบต่อส่วนอื่นที่เกี่ยวข้อง เนื่องมาจากออบเจกต์อื่นไม่สามารถเข้าถึงข้อมูลนั้นๆได้โดยตรงนั่นเอง สำหรับการกำหนดคุณสมบัติการเข้าถึงนั้นมี 3 แบบคือ Public , Private และ Protect โดยแต่ละแบบมีคุณสมบัติดังนี้

1. Public เป็นการกำหนดให้ออบเจกต์อื่นเข้าถึงข้อมูลที่มีการประกาศแบบนี้
2. Private เป็นการกำหนดให้ออบเจกต์อื่นไม่สามารถเข้าถึงข้อมูลเหล่านี้ได้แต่จะต้องเข้าถึงโดยอาศัยการผ่านฟังก์ชัน
3. Protected เป็นแบบที่ออบเจกต์ไม่สามารถเข้าถึงข้อมูลเหล่านี้ได้ แต่จะมีเพียงออบเจกต์ที่ได้รับการสืบทอดมาหรือเป็นออบเจกต์ลูกจึงจะอ้างข้อมูลเหล่านี้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.6 ความสัมพันธ์ระหว่างคลาส (Class Relationships)

ความสัมพันธ์ระหว่างคลาสแต่ละคลาสสามารถแยกตามลักษณะความสัมพันธ์ระหว่างคลาสว่ามีการทำงานร่วมกันอย่างไร ซึ่งเราจะนำแต่ละคลาสมาทำงานร่วมกันตามความสัมพันธ์ที่เหมาะสมที่สุด

2.1.6.1 คลาสที่มีคลาสอื่นอยู่ภายใน (The “Contain” Relationship)

เป็นลักษณะความสัมพันธ์ที่คลาสหนึ่งจะบรรจุอีกคลาสหนึ่งเอาไว้ภายใน ซึ่งคลาสที่โดนบรรจุลงไปจะถูกซ่อนเอาไว้ไม่ให้ขอบเจ็ทต์หรือคลาสอื่นมองเห็นว่าขอบเจ็ทต์นี้มีคลาสอื่นอยู่ด้วย ดังนั้นมันจะไม่อยู่ในส่วนของการเชื่อมต่อ (Interface) ซึ่งการเรียกใช้งานคลาสที่บรรจุลงไปภายในนั้นจะถูกเรียกใช้ หรือ interface ได้เฉพาะการทำงานภายในขอบเจ็ทต์หรือคลาสเท่านั้น

2.1.6.2 คลาสหนึ่งใช้งานอีกคลาสหนึ่ง (The “Uses” Relationship)

เป็นลักษณะความสัมพันธ์ที่คลาสหนึ่งต้องการทำงานร่วมกับคลาสอื่น โดยการส่งเมสเสจไปให้คลาสที่ต้องการเพื่อให้ได้ผลลัพธ์ที่จะนำมาใช้งาน ความสัมพันธ์แบบนี้จะเป็นการส่งข้อมูลกันระหว่างคลาสซึ่งส่วนนี้เราจะกำหนดให้อยู่ในส่วนของการติดต่อแล้วขอบเจ็ทต์อื่นสามารถมองเห็นได้ว่าคลาสนั้นมีการส่งผ่านข้อมูลไปยังคลาสนั้น หากต้องการติดต่อหรือส่งเมสเสจไปให้คลาสที่มีการส่งผ่านไปให้คลาสนั้นก็จะมีการทำงานเหมือนการส่งเมสเสจไปยังคลาสที่ถูกเรียกใช้พร้อมๆกัน

ความสัมพันธ์แบบบรรจุหรือมีขอบเจ็ทต์อื่นอยู่ภายใน กับความสัมพันธ์แบบเรียกใช้แตกต่างกันที่แบบบรรจุถูกบรรจุไว้ภายในขอบเจ็ทต์ เพราะเราทราบแน่นอนว่าขอบเจ็ทต์นั้นจะต้องใช้คลาสที่ประกาศไว้ทุกครั้ง แต่ในแบบเรียกใช้เราจะไม่ทราบเวลาแน่นอน

2.1.6.3 คลาสหนึ่งสืบทอดคุณสมบัติจากอีกคลาสหนึ่ง (The “Inheritance” Relationship)

เป็นลักษณะความสัมพันธ์ที่คลาสสามารถสืบทอด ข้อมูล ส่วนการเชื่อมโยง และ เมธอดต่างๆ จากคลาสนั้นๆ ได้ ทำให้คลาสที่ได้รับการสืบทอดจากคลาสนั้นๆ ไม่ต้องมีการกำหนด ข้อมูล ส่วนการเชื่อมโยง และเมธอด อีกโดยสามารถที่จะเรียกใช้งานตัวแปรและการทำงานเหล่านั้นได้ทันที คลาสที่สืบทอดมาจะเรียกว่า Subclass ส่วนคลาสที่ให้การสืบทอดเรียกว่า Superclass

คลาสที่ให้คลาสอื่นสืบทอดนั้นไม่จำเป็นต้องมีคุณสมบัติเหมือนกับคลาสดู แต่คลาสดูจะต้องมีคุณสมบัติเหมือนกับคลาสมแม่ทั้งหมด และคลาสดูยังสามารถเปลี่ยนแปลงและเพิ่มเติมการทำงานของเมธอดจากการทำงานเดิมของคลาสมแม่ให้เหมาะสมกับการทำงานของคลาสดูได้ โดยลักษณะการถ่ายทอดจะมี 2 แบบ คือ Single Inheritance คือ การสืบทอดจาก Superclass เพียงคลาสดูเดียว กับ Multiple Inheritance คือการสืบทอดมากกว่าหนึ่งคลาสนั้นเอง

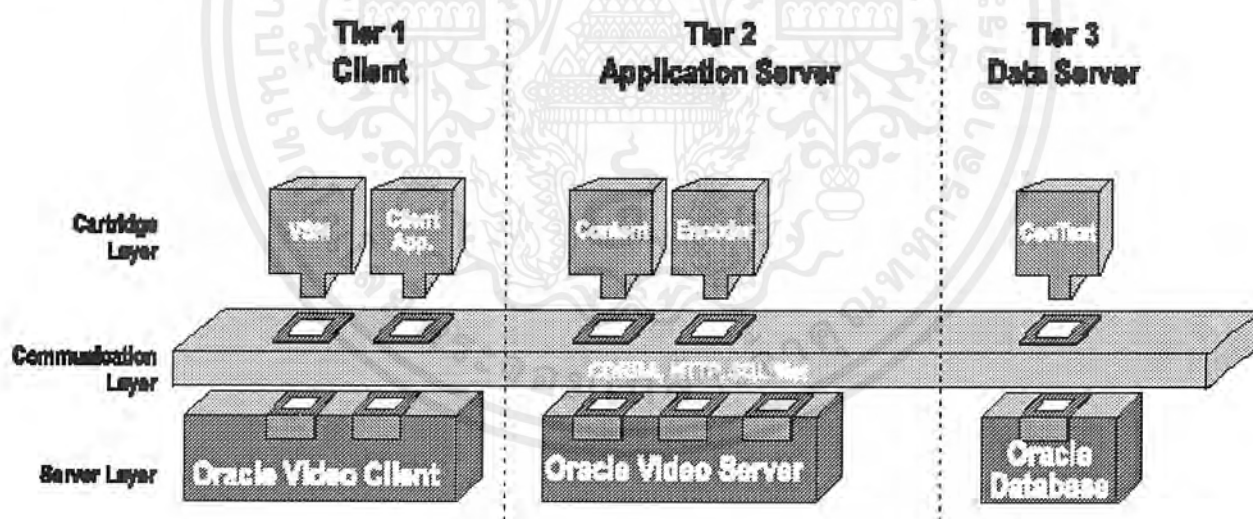
บทที่ 3

ทฤษฎีและความรู้พื้นฐานเกี่ยวกับ Oracle Video Server

Oracle Video Server ได้จัดเตรียมเครื่องมือและโปรแกรมต่างๆสำหรับการพัฒนาแอปพลิเคชันด้านมัลติมีเดียไว้อย่างสมบูรณ์ ไม่ว่าจะเป็นทางด้านการจัดการกับข้อมูลมัลติมีเดีย การบริหารงานข้อมูลมัลติมีเดีย การจัดส่งข้อมูลมัลติมีเดียจากด้านเซิร์ฟเวอร์ไปยังด้านไคลเอนต์ รวมไปถึงการเตรียมไลบรารีต่างๆสำหรับให้ด้านไคลเอนต์ใช้งานเพื่อดึงข้อมูลมัลติมีเดียจากด้านเซิร์ฟเวอร์ได้ ซึ่งในโครงการนี้ได้เลือกที่จะใช้ Oracle Video Server เป็นเครื่องมือช่วยในการพัฒนาแอปพลิเคชันที่ใช้สำหรับชมข้อมูลมัลติมีเดียจากวีดีโอเซิร์ฟเวอร์

3.1 สถาปัตยกรรมของ Oracle Video Server

สถาปัตยกรรมของ Oracle Video Server เป็นไปตามรูปแบบของ Oracle's Network Computing Architecture (NCA) ซึ่งประกอบไปด้วยส่วนต่างๆ ดังนี้



รูปที่ 3-1 แสดงสถาปัตยกรรมของ Oracle Video Server

3.1.1 ส่วนของไคลเอนต์ (Tier1: Client)

- Oracle Video Client (OVC) ใช้เป็นเครื่องมือในการพัฒนาแอปพลิเคชัน เช่น ไลบรารีของภาษาจาวา, Active-X Control และเว็บปลั๊กอิน เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Oracle Video Server Manager (VSM) เป็นโปรแกรมที่ถูกพัฒนาขึ้นเพื่อใช้สำหรับการบริหารระบบ OVS จากด้านไคลเอนต์

3.1.2 ส่วนแอปพลิเคชันเซิร์ฟเวอร์ (Tier2: Application Server)

ทำหน้าที่รับและส่งข้อมูลหรือคอยรับบริการทขอข้อมูลมัลติมีเดียที่จะเกิดจากด้านไคลเอนต์ เช่น โปรแกรมOVS เป็นต้น

3.1.3 ส่วนดาตาเบสเซิร์ฟเวอร์ (Tier3: Database Server)

ทำหน้าที่ให้บริการข้อมูลแก่แอปพลิเคชันเซิร์ฟเวอร์ ข้อมูลที่เก็บจะเป็นข้อมูลแบบ Meta-Data

3.2 ส่วนประกอบของระบบ Oracle Video Server

ระบบ OVS จะมีการทำงานที่ซับซ้อน และมีส่วนการทำงานที่ต่าง ๆ กัน เช่น ส่วนที่ทำหน้าที่ในการเก็บข้อมูลมัลติมีเดียหรือต่อไปจะเรียกว่าคอนเทนต์ (Content) ส่วนที่ทำหน้าที่เชื่อมการทำงานระหว่างเครื่องในระบบ(Oracle Media Net) หรือส่วนที่ทำหน้าที่ในการประสานงานระหว่างการทำงานต่างๆที่เกิดขึ้นในระบบเอง ซึ่งส่วนประกอบหลักๆของระบบ OVS มีดังนี้

3.2.1 Oracle Video Server (OVS)

OVS จะทำหน้าที่การจัดเก็บและจัดการคิงคอนเทนต์ต่างๆในวิดีโอเซิร์ฟเวอร์(Video Server) เพื่อนำมาบริการให้กับไคลเอนต์ ซึ่งองค์ประกอบภายใน OVS มีดังนี้

3.2.1.1 คอนเทนต์ (Content)

คอนเทนต์ คือ ข้อมูลมัลติมีเดียที่ใช้เก็บและใช้ส่งไปยังไคลเอนต์ และรูปแบบของคอนเทนต์ในระบบ OVS จะมีอยู่ 2 ลักษณะ คือ

- ไฟล์ฟิสิกส์คอลคอนเทนต์ (Physical Content)
- ลอจิกคอลคอนเทนต์ (Logical Content)

ไฟล์ฟิสิกส์คอลคอนเทนต์ หมายถึง ไฟล์มัลติมีเดียที่ถูกสร้างขึ้นจากการเข้ารหัสในแบบต่างๆ เช่น ไฟล์ตระกูล MPEG และ ไฟล์นามสกุลAVI เป็นต้น และไฟล์ฟิสิกส์คอลคอนเทนต์อีกลักษณะหนึ่งคือ ไฟล์แท็ก (Tag file) ไฟล์แท็กจะเป็นไฟล์ที่เก็บรายละเอียดต่างๆของคอนเทนต์นั้นๆ (Meta Data) เช่น ชื่อคอนเทนต์ ความยาวของคอนเทนต์ เป็นต้น ทุกๆครั้งที่มีการเพิ่มไฟล์มัลติมีเดียให้กับระบบ จะต้องทำการลงทะเบียน (Register) ไฟล์มัลติมีเดียเหล่านั้นๆให้กับ OVS รู้ และในขั้นตอนการลงทะเบียนจะต้องมีการสร้างไฟล์แท็กขึ้นทุกครั้ง เพื่อOVSจะได้นำมาใช้เป็นข้อมูลร่วมกับของคอนเทนต์นั้นๆ การลงทะเบียนจะกระทำกับ Oracle Media Data Store (MDS) ซึ่งจะกล่าวถึงรายละเอียดในส่วนถัดไป

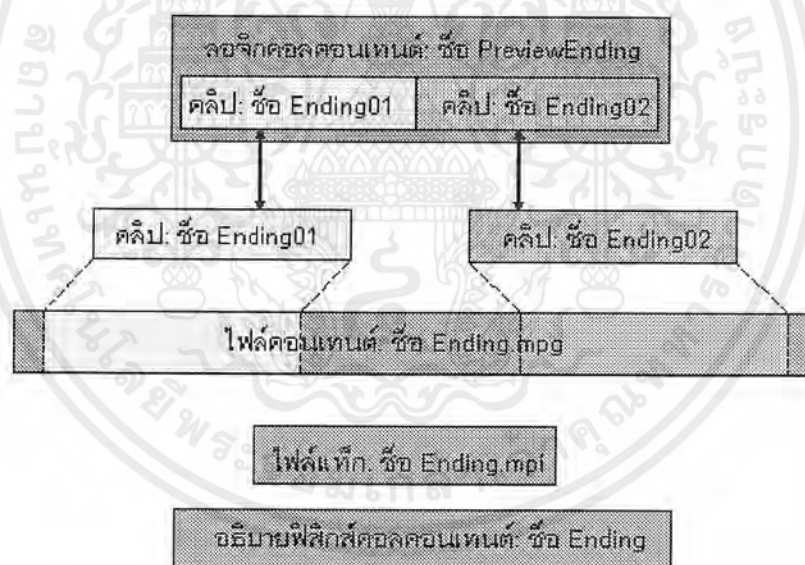
หมายเหตุ ในระบบ Oracle Video Server จะรู้จักไฟล์มัลติมีเดียตระกูล MPEG และ OSF เท่านั้น โดย OSF จะเป็นไฟล์มัลติมีเดียในรูปแบบของทาง Oracle เอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟล์ลอจิกคอลคอนเทนต์ หมายถึง ไฟล์คอนเทนต์ที่เรียกใช้งานไฟล์ฟิสิกส์คอลคอนเทนต์อีกทีหนึ่ง ไฟล์ลอจิกคอลคอนเทนต์จะเสมือนไม่มีตัวตน กล่าวคือไฟล์ฟิสิกส์คอลคอนเทนต์จะมีข้อมูลที่ประกอบไปด้วยภาพและเสียงจริงๆ ต้องมีการเก็บข้อมูลดังกล่าวไว้เพื่อเรียกใช้งานจริงๆ แต่ไฟล์ลอจิกคอลคอนเท็นต์จะมีเพียงแต่ข้อมูลที่คอยกำกับว่าจะต้องไปเรียกไฟล์ฟิสิกส์คอลคอนเทนต์ในส่วนไหนมาแสดง และจะหยุดเมื่อไร ดังนั้นไฟล์ลอจิกคอลคอนเทนต์จึงเก็บเฉพาะรายละเอียดที่พูดถึงข้อมูลมัลติมีเดียอีกที (Meta Data)

ความสัมพันธ์ของไฟล์ต่างๆ ใน OVS มีดังต่อไปนี้

- ไฟล์คอนเทนต์ เป็นไฟล์ที่ OVS เก็บและใช้สำหรับส่งไปยังไคลเอนต์ โดยข้อมูลภายในจะเป็นข้อมูลมัลติมีเดียที่ประกอบไปด้วยภาพและเสียงที่ถูกเข้ารหัสเอาไว้
- ไฟล์แท็กเป็นไฟล์ที่เก็บข้อมูลเกี่ยวกับคอนเทนต์นั้นๆ ซึ่งจะประกอบไปด้วย ชื่อคอนเทนต์ ฟอรัมเมตที่ใช้เข้ารหัส อัตราบิตที่ใช้ในการเข้ารหัส และความยาวของคอนเทนต์ เป็นต้น ไฟล์แท็กจะมีนามสกุลเป็น .mpi



รูปที่ 3-2 แสดงความสัมพันธ์ของข้อมูลภายในระบบ Oracle Video Server

- ส่วนอธิบายฟิสิกส์คอลคอนเทนต์ เป็นรายละเอียดข้อมูลที่เหมือนในไฟล์แท็ก
- คลิป เป็นข้อมูลที่ตัดเอาเพียงบางส่วนของไฟล์คอนเทนต์มา ซึ่งแต่ละคลิปจะประกอบไปด้วยไฟล์คอนเทนต์ของตัวเอง
- ลอจิกคอลคอนเทนต์เป็นการนำเอาคลิปหลายๆชิ้นมารวมกัน

จากรูปที่ 3-2 จะแสดงให้เห็นถึงความสัมพันธ์ของข้อมูลต่างๆมีต่อกัน โดยข้อมูลต่างๆเหล่านี้จะถูกแยกเก็บไว้ในวอลลุ่มในMDS และในฐานข้อมูล(Oracle Database) โดยในวอลลุ่ม MDS นั้นจะเก็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟล์แท้กและไฟล์คอนเทนตต์ที่เป็นเนื้อข้อมูลมัลติมีเดียภาพและเสียงจริงๆ ส่วนในฐานข้อมูลนั้นจะเก็บข้อมูลของลอจิกคอลคอนเทนตต์ คลิป และส่วนที่อธิบายถึงฟิสิกส์คอลคอนเทนตต์

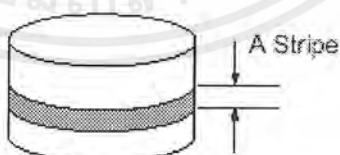
ในโครงการนี้ได้นำเอาลอจิกคอลคอนเทนตต์มาทำเป็นตัวอย่างของภาพยนต์ โดยการทำคลิปหลายคลิป แต่ละคลิปจะดึงเอาบางส่วนของภาพยนต์เต็มออกมา จากนั้นเอาคลิปเหล่านั้นมาต่อๆกันเป็นลอจิกคอลคอนเทนตต์

3.2.1.2 Oracle Media Data Store (MDS)

MDS เป็นแหล่งเก็บข้อมูลที่สำคัญมากสำหรับ OVS เพราะไฟล์ข้อมูลต่างๆจะถูกเก็บไว้ในวอลลุ่มดิสก์ของ MDS หากไม่มีวอลลุ่มดิสก์ระบบจะไม่สามารถบริการวิดีโอได้เลย ซึ่งต่างจากฐานข้อมูล หากระบบใดขาดฐานข้อมูลก็ยังสามารถให้บริการวิดีโอได้ แต่จะไม่สามารถทำในส่วนของลอจิกคอลคอนเทนตต์ได้เท่านั้น

MDS คือ ส่วนที่ OVS จะใช้เก็บไฟล์คอนเทนตต์และไฟล์แท้ก ภายใน MDS เองจะมีหน้าที่การทำงานหลายอย่าง อย่างเช่น การจัดการเกี่ยวกับการเก็บข้อมูลในตัวมันเอง การจัดการคิงข้อมูลมัลติมีเดียส่งให้แก่ OVS เป็นต้น โดย MDS จะมีส่วนที่เรียกว่า “วอลลุ่ม” (Volume) ซึ่งเป็นพื้นที่ที่ใช้เก็บข้อมูลต่างๆ และแต่ละวอลลุ่มจะมีตารางที่ใช้เก็บชื่อคอนเทนตต์ภายใน เพื่อใช้สำหรับการเข้าถึงข้อมูลคอนเทนตต์เหล่านั้น วอลลุ่มหนึ่งของ MDS อาจจะประกอบขึ้นจากดิสก์หนึ่งดิสก์ หรือหลายดิสก์ก็ได้ ขึ้นอยู่กับการออกแบบของระบบว่าดิสก์ควรจะมีจำนวนเท่าไรต่อหนึ่งวอลลุ่ม (จำนวนของดิสก์ต่อวอลลุ่มจะเรียกว่า RAID size) และภายใน OVS สามารถที่จะประกอบไปด้วยหลายๆวอลลุ่มได้

เนื่องจากการที่วอลลุ่มจะเป็นส่วนที่เก็บเนื้อข้อมูลทั้งหมด ดังนั้นหากเกิดการเสียหายขึ้นกับวอลลุ่มจะส่งผลกระทบต่อให้ระบบเสียหายได้ ดังนั้น MDS จึงได้เตรียมความพร้อมในการรองรับกับเหตุการณ์ที่ไม่อาจคาดคิดที่จะก่อให้เกิดความเสียหายกับวอลลุ่ม คือการเตรียมกลไกกู้ข้อมูลจากวอลลุ่มที่เสียหายไปแล้วขึ้นมาใหม่ จะเรียกวิธีนี้ว่า “การทำกรป้องกันแบบ RAID” (Redundant Array of Inexpensive Disk)

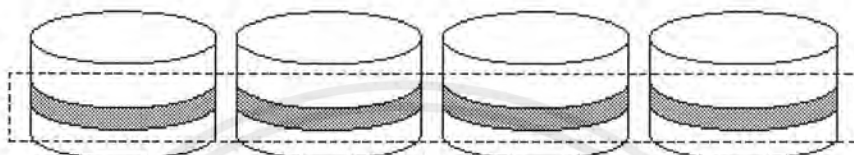


รูปที่ 3-3 แสดงสติปหนึ่งสติปบนดิสก์

การทำกรป้องกันแบบ RAID คือการที่เรานำดิสก์หลายๆดิสก์มาทำเป็นหนึ่งวอลลุ่ม และการทำ RAID นั้น จะต้องแบ่งพื้นที่แต่ละดิสก์ออกเป็นส่วนเล็กๆ ที่เรียกว่า “สติป” (Stripe) โดยสติปจะเป็นพื้นที่ที่ใช้เก็บข้อมูล ซึ่งจะมีขนาดความจุ 32Kbyte หรือ 64Kbyte เท่านั้น ซึ่งกำหนดได้ในขั้นตอนการสร้างดิสก์วอลลุ่ม กลไกการกู้ข้อมูลคือการที่ MDS จะทำการเก็บข้อมูลแบบกระจายทั้งวอลลุ่ม หมายถึงหากมีการ

เขียนข้อมูลลงดิสก์วอลลุ่ม MDS จะทำการเขียนข้อมูล 32KBแรกที่สลิปแรกของดิสก์ที่ 1 ในวอลลุ่ม ข้อมูลที่ถูกเขียนลงนั้นจะมีทั้งข้อมูลจริงและข้อมูลที่ใช้สำหรับกรณีกู้คืนเสียหาย (Parity) ด้วย เสร็จแล้ว MDS จะทำการเขียนข้อมูลชุดต่อไปที่สลิปแรกในดิสก์ที่ 2 ในวอลลุ่ม และจะเขียนข้อมูลเช่นนี้ไปเรื่อยๆจนเสร็จจากการกระทำดังกล่าวของ MDS จะทำให้ข้อมูลกระจายอยู่บนดิสก์ทุกตัวในวอลลุ่มนั้นๆ หากเกิดดิสก์ตัวใดตัวหนึ่งเสียหายขึ้นมาในวอลลุ่ม MDSจะสร้างข้อมูลที่หายไปโดยการเอาข้อมูลจากดิสก์อื่นๆ(Parity)ในวอลลุ่มมาใช้พิจารณาในการสร้างข้อมูลชุดเดิมขึ้นมาใหม่อีกที

หมายเหตุ ขนาด RAID 4 หมายถึงในวอลลุ่มนั้นมีดิสก์อยู่ 4 ตัว



รูปที่ 3-4 แสดงวอลลุ่มของ MDS ที่มีขนาด RAID 4

ความสามารถหนึ่งของ OVS คือ OVS สามารถทำงานร่วมกับไคลเอนต์ได้หลายๆไคลเอนต์ในเวลาเดียวกัน ดังนั้นสิ่งสำคัญอย่างหนึ่งที่ MDS พึ่งต้องคำนึงถึงคือการที่เกิดการเขียนข้อมูลพร้อมกันในเวลาเดียวกันจากไคลเอนต์ที่ต่างกัน

MDS จะมีข้อกำหนดต่อการเขียนข้อมูลพร้อมๆกันดังนี้

- ไฟล์หนึ่งไฟล์จะสามารถถูกเขียนได้เพียงไคลเอนต์เดียวในเวลาใดเวลาหนึ่ง
- ขณะที่ไฟล์กำลังถูกเขียนลง MDS โดยไคลเอนต์หนึ่ง ไคลเอนต์อื่นๆจะไม่สามารถทำการแก้ไขรายละเอียดใดๆได้ เช่น การเปลี่ยนชื่อไฟล์, การลบไฟล์, การตัดต่อบางส่วนของไฟล์ เป็นต้น
- หากกระบวนการเขียนไฟล์เกิดหยุดชะงักจากสาเหตุบางประการ โดยทำให้การเขียนไฟล์ในครั้งนั้นไม่สามารถทำงานต่อไปได้ MDS จะทำเก็บระบบขึ้นมาใหม่โดยอาศัยกระบวนการการแก้ไขระบบเสียหาย

เนื่องจาก OVS เป็นระบบที่ไม่ขึ้นกับรูปแบบข้อจำกัดต่างๆของเครือข่าย ทั้งด้านเซิร์ฟเวอร์และด้านไคลเอนต์ ดังนั้น OVS จะสามารถปรับอัตราการส่งข้อมูลให้เหมาะสมกับระบบเครือข่ายของด้านไคลเอนต์ ซึ่งระบบเครือข่ายที่ OVS รองรับมีดังนี้

- UDP (User Datagram Protocol)
- TCP (Transmission Control Protocol)
- ATM (Asynchronous Transfer Mode)
- AAL5 (ATM Adaptation Layer5)
- DVB (Digital Video Broadcast)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 Oracle Video Server Manager (VSM)

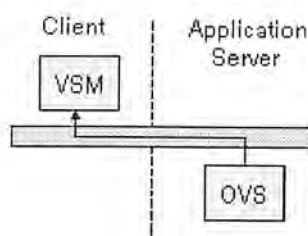
VSM คือ โปรแกรมที่ถูกออกแบบมาเพื่อใช้งานโดยผู้ดูแลระบบวิดีโอเซิร์ฟเวอร์ โดย VSM จะถูกใช้งานในเครื่องไคลเอนต์เพื่อทำการริโมตมาควบคุมเซิร์ฟเวอร์อีกที VSM ยังถูกออกแบบมาเพื่อใช้งานด้านลอจิกคอลคอนเทนต์อีกด้วย คือการทำคลิป การตัดต่อข้อมูลมัลติมีเดีย หากสกรูปรงานของ VSM จะมีดังนี้

- เปิดและปิดวิดีโอเซิร์ฟเวอร์
- ตรวจสอบ และติดตามสถานะการทำงานของบริการต่างๆว่ามีปัญหาเกิดขึ้นหรือไม่
- ทำงานในส่วนของลอจิกคอลคอนเทนต์, คลิป และ Media Data Store เช่น วอลลุ่ม ดิสก์ และไฟล์ เป็นต้น
- โหลดและวีจิสเตอร์ข้อมูลมัลติมีเดีย

VSM จะเป็นโปรแกรมที่ต้องติดต่อกับ OVS ดังนั้น VSM จะต้องมีลำดับขั้นตอนในการติดต่อสื่อสารกับ OVS ดังต่อไปนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ขั้นที่ 4 เมื่อ OVS ได้ข้อมูลทั้งจาก MDS และตัวฐานข้อมูลแล้ว OVS จะทำการส่งข้อมูลเหล่านั้นกลับไปให้แก่ VSM เป็นอันสิ้นสุดการติดต่อระหว่าง VSM และ OVS

3.2.3 Oracle Media Net (OMN)

OMN ทำหน้าที่ในการเชื่อมการทำงานระหว่างเซิร์ฟเวอร์และไคลเอนต์ให้สามารถทำงานร่วมกัน ถึงแม้ว่าสภาพแวดล้อมของทั้งด้านเซิร์ฟเวอร์และไคลเอนต์จะแตกต่างกัน ตัว OMN เองนั้นถูกพัฒนาขึ้นจากเทคโนโลยี CORBA (Common Object Request Broker Architecture) ซึ่งเป็นเทคโนโลยีที่สามารถทำงานได้อย่างเป็นอิสระต่อข้อจำกัดต่างๆ ไม่ว่าจะเป็นภาษาโปรแกรมมิ่งที่ต่างกัน ฮาร์ดแวร์หรือระบบปฏิบัติการที่ต่างกัน และยังมีการกระจายงานให้กับเครื่องภายในระบบเครือข่ายเพื่อให้ทำงานร่วมกัน และสามารถใช้อุปกรณ์ร่วมกันได้โดยไม่ต้องคำนึงถึงส่วนที่ทำหน้าที่รับข้อมูลให้กับสภาพแวดล้อมที่ต่างกัน

CORBA นั้นจะมีมาตรฐานรองรับระบบเปิดสำหรับระบบเครือข่ายแบบกระจายเชิงวัตถุที่ถูกกำหนดโดยกลุ่ม OMG (Object Management Group) ซึ่งมาตรฐานดังกล่าวได้บอกถึงกลไกที่ใช้ในการร้องขอและการรับข้อมูลระหว่างกันของโปรแกรมในสภาพแวดล้อมของ CORBA หน้าที่อีกอย่างของ OMN คือ การทำโหลดสมดุล กล่าวคือ หากเกิดกรณีที่เซิร์ฟเวอร์ใดเซิร์ฟเวอร์หนึ่งรับภาระจากปริมาณไคลเอนต์มากเกินไป จนถึงขั้นเกิดการเสียหายของเซิร์ฟเวอร์ OMN จะแบ่งภาระจากเซิร์ฟเวอร์หนึ่งไปยังอีกเซิร์ฟเวอร์หนึ่งโดยอัตโนมัติ เพื่อทำให้เซิร์ฟเวอร์ต่างๆ รับภาระในปริมาณที่เหมาะสม ส่งผลให้เสถียรภาพของระบบดีขึ้น

3.2.4 Oracle Video Client (OVC)

Oracle Video Client เป็นซอฟต์แวร์ที่ช่วยในการพัฒนาระบบงานมัลติมีเดีย เช่น video-on-demand โดยแอปพลิเคชันของ Oracle Video Client ที่พัฒนาขึ้นมาสามารถดำเนินการกับไฟล์มัลติมีเดียที่มาจาก Oracle Video Server เช่น MPEG (Motion Picture Experts Group) video, MPEG Audio และ OSF (Oracle Streaming Format ซึ่งเป็นรูปแบบการจัดเก็บของ Oracle เอง) ไฟล์ที่สามารถนำมาแปลงเป็นรูปแบบ OSF ได้คือ WAV และ AVI

จากความสามารถของ Oracle Video Server ซึ่งยอมให้เครื่องไคลเอนต์สามารถเข้าถึงไฟล์มัลติมีเดียไฟล์เดียวกันได้มากกว่าหนึ่งเครื่องในเวลาเดียวกัน ทำให้ผู้ใช้รับสตรีมของวิดีโอ ในตำแหน่งที่ต่างกันในเวลาเดียวกันได้ สำหรับ Oracle Video Client นั้นมีเครื่องมือที่ช่วยในการพัฒนาแอปพลิเคชันเหล่านี้ด้วยเช่น Oracle Video ActiveX Control, Oracle Video Web Plug-in และ Oracle Video Java Library

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4.1 Oracle Video Client (OVC)

Oracle Video Client (OVC) เป็นซอฟต์แวร์ที่ทำการจัดการกับสตรีมภาพและเสียงที่ได้รับมาจาก Oracle Video Server (OVS) และส่งสตรีมเหล่านั้นมายังแอปพลิเคชัน และยังมีการฝังอินเทอร์เฟซลงไป ในแอปพลิเคชันที่พัฒนาขึ้นมาได้อีกด้วย โดยส่วนของการอินเทอร์เฟซจะใช้งานผ่าน Oracle Video Java Library ซึ่งเป็นกลุ่มคลาสของจาวา

ส่วนอินเทอร์เฟซนี้ทำให้แอปพลิเคชันที่พัฒนาขึ้นมาสามารถที่จะโหลด, แสดงผลไฟล์มัลติมีเดียที่โหลด และผู้ใช้สามารถควบคุมการแสดงผลเหล่านั้นได้ โดยผ่านการร้องขอไปยังวีดีโอเซิร์ฟเวอร์ ผ่านทางซอฟต์แวร์เบราว์เซอร์ ซึ่งซอฟต์แวร์เบราว์เซอร์นี้จะทำการจัดการในเรื่องของการติดต่อกับเซิร์ฟเวอร์, ควบคุมสตรีมให้เป็นแบบ real-time รวมไปถึงการจัดการกับภาพและเสียงอีกด้วย

3.2.4.2 Client Interface

การใช้ไคลเอนต์ในรูปแบบของระบบ OVC/OVS นั้นสามารถสรุปความหมายของคำว่าไคลเอนต์ ได้ดังนี้

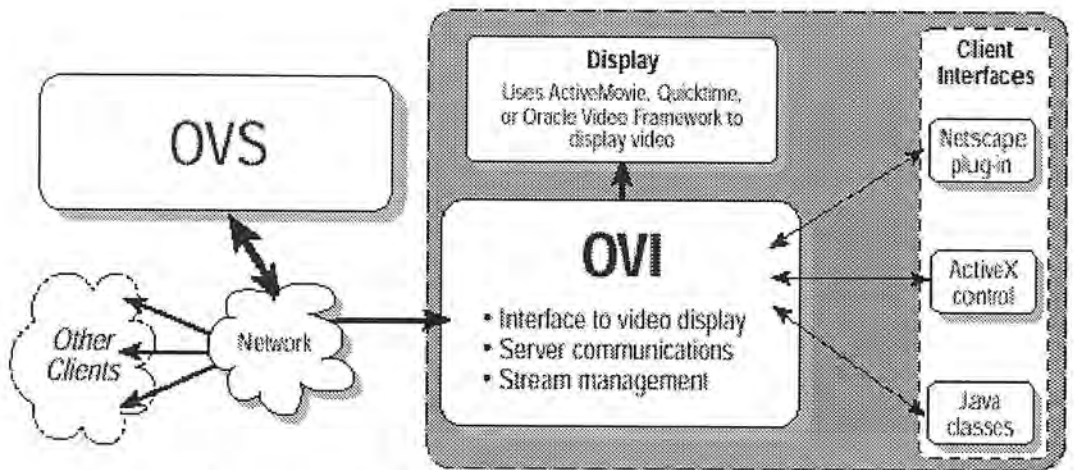
- ระดับแรกของไคลเอนต์คือ enterprise client : ไคลเอนต์ที่เข้าถึงวีดีโอเซิร์ฟเวอร์ จากที่อื่นๆได้เรียกว่าเครื่องไคลเอนต์
- เครื่องไคลเอนต์จะประกอบด้วย Oracle Video Client(OVC) Software โดย จะแบ่งออกเป็น 2 ส่วนคือ OVI Abstraction Layer และ Client Interface (ในที่นี้หมายถึง Oracle Video Java Library)
- OVI คือไคลเอนต์ของระบบ OVS และ Client Interface ก็ไคลเอนต์ของ OVI อีกทีก็ว่าได้ การพัฒนาแอปพลิเคชันนั้นจะทำโดยผ่านทาง Client Interfaces Component

หลักการของ OVC ก็คือ Client Interfaces ซึ่งการอินเทอร์เฟซนี้จะทำผ่านทาง OVI ซึ่งจะเข้าถึงสตรีมของภาพจาก OVS การแยกส่วนของไคลเอนต์ออกไปทำให้เราสามารถควบคุมการทำงานต่างๆได้หลากหลายแพลตฟอร์มโดยไม่ต้องคำนึงถึงกลไกของเครื่องเลย

3.2.4.3 Client Software Components

ในระบบ Oracle Video Server นั้นจะต้องประกอบด้วยองค์ประกอบดังต่อไปนี้

- Oracle Video Server เป็นตัวจัดการและส่งลำดับของภาพเสียงแก่เครื่องไคลเอนต์
- Oracle Video Interfaces ทำการประมวลผลบนไคลเอนต์จะจัดการกับลำดับของข้อมูลที่มาจาก OVS และส่งข้อมูลไปยัง OVS ด้วย
- OVC client interfaces ให้ผู้ใช้สามารถควบคุมลำดับของภาพและเสียง กำหนดบริเวณที่จะแสดงผล เป็นต้น



รูปที่ 3-5 Software Component ของ Oracle Video Client

จากรูปที่ 3-5 ส่วนประกอบต่างๆจะอธิบายได้ดังนี้

- OVS เป็น backbone ของระบบ ซึ่งมีหน้าที่คือ จัดการเกี่ยวกับการเก็บและให้การดึงภาพและเสียง
- OVI เป็น gateway ของ OVS คอยให้บริการดังนี้
 1. ติดต่อกับ Client interface,ส่งการร้องขอจากส่วนอินเตอร์เฟซ ไปยังเซิร์ฟเวอร์
 2. จัดการติดต่อกับเน็ตเวิร์คระหว่างเครื่องไคลเอนต์และวีดีโอเซิร์ฟเวอร์ รวมไปถึงการจัดการภาพ,เสียงที่มาจากวีดีโอเซิร์ฟเวอร์ด้วย
 3. ติดต่อกับส่วนของการควบคุมการแสดงผลเพื่อควบคุมลำดับของข้อมูลที่เข้ามา

3.2.4.4. Oracle Video Java Library

ในการที่จะแสดงผลหรือควบคุมข้อมูลมัลติมีเดียที่มาจาก Oracle Video Server นั้น จำเป็นต้องสร้างส่วน Client Interface ขึ้นมา Oracle Video Java Library นี้จะเป็นส่วนของ Client Interface ที่ช่วยให้สามารถพัฒนาแอปพลิเคชันที่สามารถจัดการกับข้อมูลมัลติมีเดียที่มาจาก Oracle Video Server ได้โดย Oracle Video Java Library จะประกอบไปด้วยคลาสต่างๆดังนี้

1. กลุ่มคลาสที่เกี่ยวข้องกับตัวควบคุมภาพและเสียง(Player Classes)
2. กลุ่มคลาสที่เกี่ยวข้องกับข้อมูลของภาพและเสียง(Stream Information Classes)
3. กลุ่มคลาสที่เกี่ยวข้องกับการเข้าถึงข้อมูล(Content Query Classes)

ซึ่งรายละเอียดและการใช้งานสามารถดูได้จากภาคผนวกเรื่อง Oracle Video JAVA Library

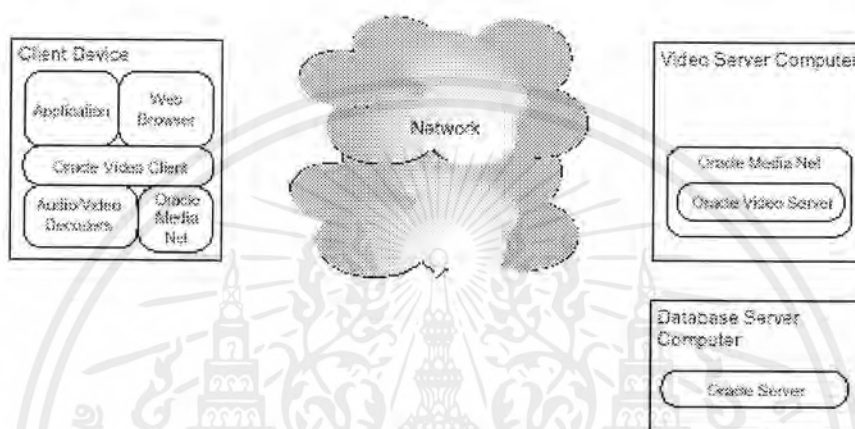
3.2.5 ออราเคิลดาตาเบสเซิร์ฟเวอร์ (Oracle DataBase Server)

ดาตาเบสเซิร์ฟเวอร์จะทำหน้าที่ในการเก็บข้อมูลที่เป็นข้อมูลของคอนเทนต์อ็อกที และยังใช้ในเก็บข้อมูลในส่วนที่ทำลอคจิกคอลคอนเทนต์ และคลิปกอีกด้วย ในระบบOVS ดาตาเบสเซิร์ฟเวอร์จะไม่ได้เก็บ

ข้อมูลมีคมีเดียจริงๆ แต่จะเก็บเพียงข้อมูลที่ใช้ควบคู่กับข้อมูลมีคมีเดียเหล่านั้น เช่น ชื่อของไฟล์ข้อมูล ขนาดความยาวของข้อมูล อัตราการเข้ารหัสของข้อมูล และอื่นๆ

สำหรับดาตาเบสเซอร์ฟเวอร์ในระบบ OVS จะมีหรือไม่มีก็ได้ เนื่องจากในส่วนนี้ถือว่าเป็นการเพิ่มประสิทธิภาพในเรื่องของการทำลอคจิคคอลคอนเทนต์ ซึ่งในโครงการนี้ได้มีการทำลอคจิคคอลคอนเทนต์ไว้ในส่วนที่ใช้เรียกดูตัวอย่างของคอนเทนต์

3.3 ฮาร์ดแวร์ของระบบ Oracle Video Server



รูปที่ 3-6 แสดงฮาร์ดแวร์ของระบบ Oracle Video Server

จากรูปจะแสดงรายละเอียดของซอฟต์แวร์ที่ต้องติดตั้งบนฮาร์ดแวร์ในระบบ OVS โดยฝั่งไคลเอนต์จะประกอบไปด้วย โปรแกรมหรือแอปพลิเคชันต่างๆ ที่ถูกสร้างขึ้นจากเครื่องมือที่ OVC เตรียมไว้ให้ โดย OVC ที่ถูกติดตั้งทางไคลเอนต์จะทำหน้าที่รับ-ส่งข้อมูลระหว่างไคลเอนต์และเซิร์ฟเวอร์ โดยจะติดต่อผ่าน OMN และข้อมูลวิดีโอที่ถูกเข้ารหัสมาจะถอดรหัสโดย Audio/Video Decoder ฝั่งแอปพลิเคชันเซิร์ฟเวอร์จะประกอบไปด้วย OMN และ OVS โดย OVS จะรับการร้องขอของ OVC ผ่านทาง OMN ส่วนดาตาเบสเซอร์ฟเวอร์จะประกอบด้วย ฐานข้อมูลของออรากิลเอง เช่น Oracle 8 เป็นต้น

3.4 ลักษณะการติดต่อระหว่างส่วนประกอบภายในระบบ Oracle Video Server

Circuit คือ ส่วนของเส้นทางการติดต่อระหว่างไคลเอนต์และเซิร์ฟเวอร์

Circuit แบ่งได้ 3 ชนิด ดังต่อไปนี้

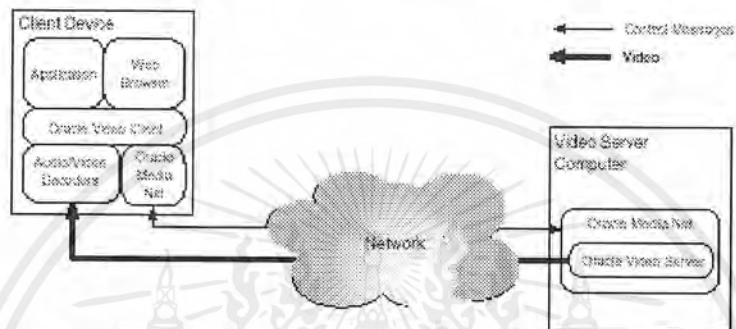
Upstream : เป็นการส่งสัญญาณควบคุมจากไคลเอนต์ไปยังเซิร์ฟเวอร์ อย่างเช่น สัญญาณการร้องขอข้อมูลวิดีโอ เป็นต้น

Downstream : เป็นการส่งสัญญาณควบคุมหรือข้อมูลวิดีโอจากเซิร์ฟเวอร์ไปยังไคลเอนต์ อย่างเช่น สัญญาณตอบรับต่อการร้องขอวิดีโอของไคลเอนต์ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

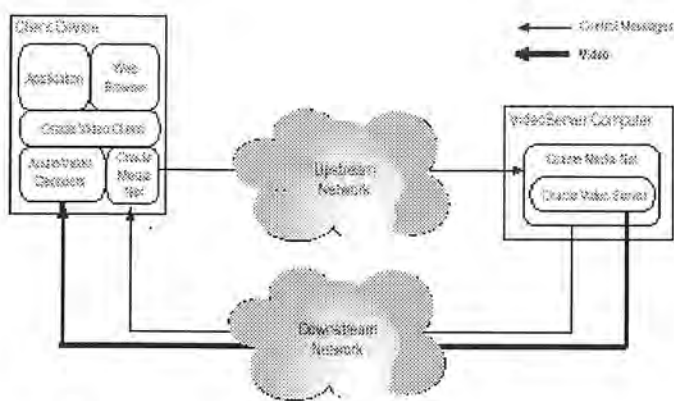
Bi-Direction : ทั้งสองฝ่าย(เซิร์ฟเวอร์และไคลเอนต์)ส่งสัญญาณควบคุมถึงกันและกัน
ลักษณะ Session ในระบบ Oracle Video Server สามารถแบ่งได้ 2 ชนิดคือ

1. Session ในระบบเครือข่าย Symmetric คือ การติดต่อระหว่างไคลเอนต์ และเซิร์ฟเวอร์บนบนเส้นทางเดียวเท่านั้น ซึ่ง Circuit ภายในจะประกอบไปด้วย ชนิด Bi-Direction และชนิด Downstream โดยชนิด Bi-Direction นั้นจะใช้สำหรับการส่งสัญญาณควบคุมระหว่างกัน แต่ชนิด Downstream นั้นจะใช้สำหรับไคลเอนต์รับข้อมูลวิดีโอจากเซิร์ฟเวอร์ สามารถแสดงได้ดังรูปที่ 3-6



รูปที่ 3-7 แสดง Session ในระบบเครือข่ายแบบ Symmetric

2. Session ในระบบเครือข่าย Asymmetric คือ การติดต่อระหว่างไคลเอนต์และเซิร์ฟเวอร์นั้นจะมีอย่างน้อย 2 เส้นทาง ซึ่ง Circuit ที่ใช้ภายใน Session ในระบบเครือข่าย Asymmetric จะมี 2 ชนิดคือ ชนิด Upstream ที่ใช้สำหรับให้ไคลเอนต์ส่งสัญญาณการร้องขอข้อมูลไปยังเซิร์ฟเวอร์ และ Circuit อีกชนิดคือ ชนิด Downstream ซึ่งใช้สำหรับเซิร์ฟเวอร์ส่งสัญญาณการควบคุมและข้อมูลวิดีโอไปยังไคลเอนต์ สามารถแสดงได้ดังนี้



รูปที่ 3-8 แสดง Session ในระบบเครือข่ายแบบ Asymmetric

3.5 ระบบเครือข่ายในระบบ Oracle Video Server

ระบบ Oracle Video Server ทำงานได้ในระบบเครือข่ายดังต่อไปนี้

- PSTN (Public Switched Telephone Network) ใช้ในสภาพแวดล้อมของ Internet กล่าวคือ สภาพแวดล้อมของ Internet นั้น โคลเอนต์และ OVS จะทำการติดต่อกันได้ต้องผ่านทางผู้ให้บริการการเชื่อมต่อ (ISP: Internet Service Provider) โดยโคลเอนต์ติดต่อกับ ISP ด้วย PSTN และใช้ IP ในการติดต่อ และ โคลเอนต์ติดต่อกับ OVS ด้วย TCP หรือ UDP

หมายเหตุ PSTN จะประกอบด้วย POTS, ISDN, ADSL

- LAN (Local-Area Network) ใช้ในสภาพแวดล้อมของ Intranet ระบบเครือข่ายที่ใช้ก็อย่างเช่น Ethernet, FDDI, CDDI เป็นต้น
- Broadband Network ระบบเครือข่ายที่ใช้ก็อย่างเช่น ATM หรือ X.25 (สำหรับการส่งสัญญาณควบคุมเท่านั้น)

3.6 กระบวนการการทำงานของโคลเอนต์ต่อเซิร์ฟเวอร์ มีดังนี้

ขั้นที่ 1 สร้าง Session และวงจรควบคุมขึ้น (Control Circuit) โดยหนึ่งโคลเอนต์จะมีเพียงหนึ่ง Session ในเวลาใดเวลาหนึ่ง โคลเอนต์จะมี ID ที่บ่งบอกถึงตัวมันเอง โดย ID ดังกล่าวจะถูกสร้างให้โดยเซิร์ฟเวอร์เอง ถ้าโคลเอนต์ได้มีการเชื่อมต่อกับเซิร์ฟเวอร์มากกว่าหนึ่งเส้นทาง โคลเอนต์นั้นสามารถที่จะเพิ่ม Circuit ใหม่ได้ในแต่ละเส้นทางที่เชื่อมต่อ และในหนึ่ง Session นั้น โคลเอนต์สามารถใช้ในการจัดการกับ Circuit ใดๆ Circuit ได้ รวมไปถึงการจัดการกับทรัพยากรของเซิร์ฟเวอร์ด้วย

Session หนึ่งจะต้องประกอบด้วย

- ID ที่บ่งบอกถึงโคลเอนต์
- วงจรควบคุมอย่างน้อยหนึ่งวงจร โดยวงจรควบคุมนี้จะถูกสร้างขึ้นเมื่อ Session ถูกกำหนดขึ้น

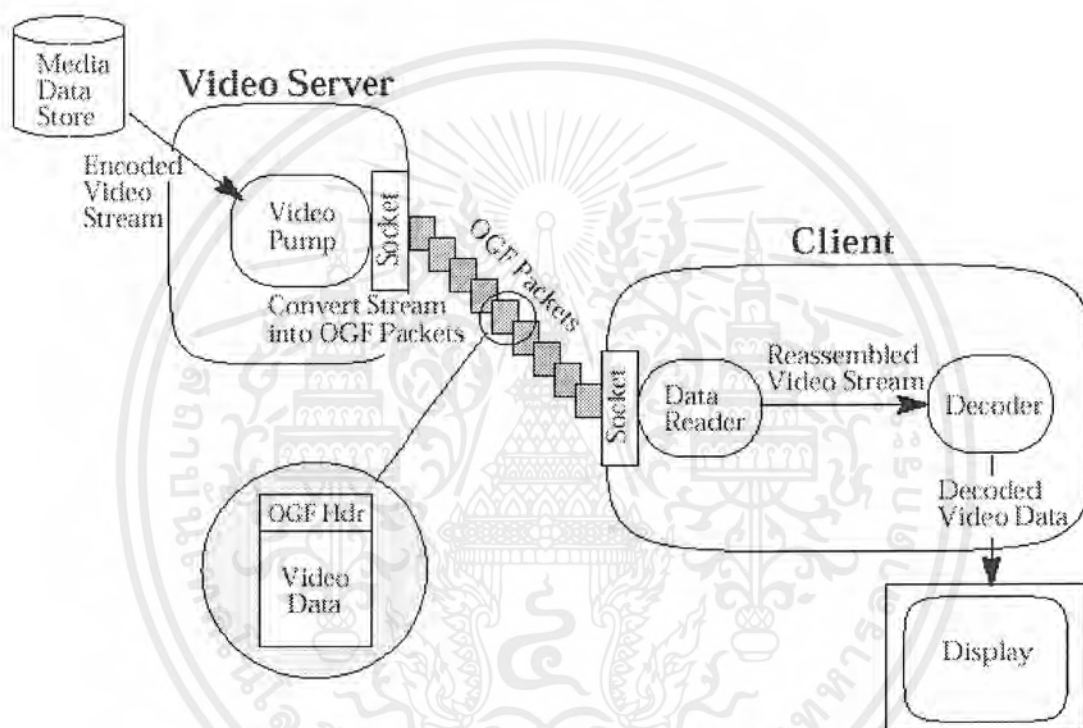
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการสร้าง Session สามารถทำได้ดังนี้

- กำหนด ID ให้แก่ไคลเอนต์
- สร้าง Network Socket สำหรับใช้รับข้อมูล
- กำหนด Session ที่ใช้ และสร้างวงจรควบคุม

ขั้นที่ 2 ไคลเอนต์ขอคิวรีข้อมูลจากเซิร์ฟเวอร์ เพื่อทำการเลือกคอนเทนต์ที่จะเพลย์

ขั้นที่ 3. ไคลเอนต์จะถูกกำหนด Session จากเซิร์ฟเวอร์ ซึ่งด้านเซิร์ฟเวอร์จะใช้ Stream Service ในการเซดอัพและควบคุมสายของข้อมูลจากเซิร์ฟเวอร์สู่ไคลเอนต์



รูปที่ 3-9 แสดงการส่งข้อมูลของเซิร์ฟเวอร์ไปยังไคลเอนต์

ขั้นที่ 4 ไคลเอนต์ทำการเพลย์คอนเทนต์

ขั้นที่ 5 ทำลาย Session ที่ได้สร้างขึ้นในตอนแรกที่มีการติดต่อ

3.7 รูปแบบของ OGF (Oracle Generic Framing Header)

OGF หรือ Oracle Generic Framing header คือ ข้อมูลที่ถูกสร้างขึ้นโดยวิดีโอปั๊ม (Video Pump) ที่จะใช้สำหรับการนำพาข้อมูลที่ต้องการส่งไปยังไคลเอนต์ เพื่อให้ไคลเอนต์ใช้สำหรับการถอดข้อมูลจริงออกจาก OGF แพ็กเกจอีกทีหนึ่ง ข้อมูลของ OGF มีขนาด 20 byte โดยสามารถแบ่งได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- FOF0F003 : เป็นชุดข้อมูลที่บ่งบอกถึงจุดเริ่มต้นของ OFG แพ็กเกจ
- Size : เป็นขนาดที่แท้จริงของข้อมูลวิดีโอ ที่ไม่ได้รวมส่วนหัวด้วย
- Fseq : เป็นจำนวนลำดับของแพ็กเกจที่วิดีโอปั๊ม (Video Pump) ส่งออกไป โดยวิดีโอปั๊มจะบวกค่า Fseq ด้วย 1 ทุกๆแพ็กเกจ และค่า Fseq จะถูกรีเซตเมื่อมีโคลเอนต์ใหม่มาเชื่อมต่อเข้าเซิร์ฟเวอร์ในระบบ ค่า Fseq มักจะถูกโคลเอนต์นำไปตรวจสอบในเรื่องของการครอบของแพ็กเกจ

FOF0F003	Size	FSeq	RV	DP	Type	CSeq	CNum	
0	4	8	10	11	12	16	18	20

รูปที่ 3-10 แสดงรายละเอียดของ Oracle Generic Framing header

- RV : ระดับของการเปลี่ยนมุมมอง ค่าดังกล่าวจะมีการเปลี่ยนแปลงในกรณีเกิดการเปลี่ยนขนาดของการแสดงผลข้อมูลวิดีโอ
- DP : คือจุดเริ่มต้นของข้อมูล หรือเป็นตัวชี้ข้อมูล ซึ่งจะเก็บค่าของออฟเซตของไบต์แรกของข้อมูลภายในแพ็กเกจ
- Type : ชนิดของข้อมูลในแพ็กเกจ เช่น วิดีโอ, สัญญาณควบคุม, ข้อมูล หรืออื่นๆ
- Cseq : ลำดับของสายในปัจจุบัน (Current Sequence in the stream) การส่งผ่านจากลำดับของแพ็กเกจหนึ่งไปยังอีกแพ็กเกจหนึ่ง สามารถกำหนดได้โดยเซิร์ฟเวอร์เท่านั้น (กรณีของไฟล์สื่อจิกคอลลคอนเทนต์ที่ประกอบขึ้นจากคลิป ซึ่งมาจากฟิสิกส์คอลลคอนเทนต์ที่ต่างกัน) กล่าวคือการเกิดการ “discontinuity” ทุกครั้ง ค่าของ Cseq จะเซตค่าให้เท่ากับค่าของ Fseq ในแพ็กเกจแรกเสมอ และจะคงค่าดังกล่าวไปจนกระทั่งเกิดการ discontinuity ครั้งใหม่ ค่า Cseq จะถูกนำไปใช้ในการตรวจสอบการเกิด discontinuity และการครอบของแพ็กเกจ
- Cnum : เป็นจำนวนคำสั่งการเกิด discontinuity ทุกๆครั้งที่เกิดการ discontinuity ค่าใน Cnum จะถูกบวกด้วย 1 เสมอ และจะคงค่าไว้จนกระทั่งเกิดการ discontinuity ในครั้งถัดไป ค่า Cnum จะถูกนำไปใช้ในการตรวจสอบการครอบของแพ็กเกจทั้งหมดใน discontinuity

3.8 การตรวจสอบการเกิด Discontinuity ในข้อมูล

Discontinuity เป็นความไม่ต่อเนื่องของข้อมูลที่เกิดขึ้น กล่าวคือ การที่สื่อจิกคอลลคอนเทนต์หนึ่งประกอบขึ้นจากคลิปหลายๆคลิป ซึ่งคลิปเหล่านี้อาจจะมาจากฟิสิกส์คอลลคอนเทนต์เดียวกัน หรือไม่ก็ได้ เมื่อนำมาเพลย์ต่อกัน การที่เนื้อข้อมูลไม่ได้เป็นเนื้อข้อมูลที่ติดกันจริงๆในทางฟิสิกส์คอลล จึงเกิดเรียกว่า ความไม่ต่อเนื่องของข้อมูล การอ่านข้อมูลฟิสิกส์คอลลของเซิร์ฟเวอร์นั้น เซิร์ฟเวอร์จะรู้ว่า ณ. ตอนนี้อยู่ที่เวลาใดจะต้องไปอ่านข้อมูลจากจุดใดในข้อมูลฟิสิกส์คอลลคอนเทนต์ เซิร์ฟเวอร์เองจะมีการใส่ข้อมูลเกี่ยวกับ Discontinuity ลงใน OGF เฮดเดอร์ เพื่อให้โคลเอนต์รู้ว่าเกิด discontinuity ขึ้นที่แพ็กเกจใดบ้าง และโคลเอนต์ยังใช้ข้อมูลนี้เป็นตัวตรวจสอบว่าข้อมูลเกิดการหล่นหายไปหรือเปล่าด้วย

พิจารณาดังตัวอย่างข้อมูลต่อไปนี้

	Fseq	Cseq	Cnum
	1	1	1
	2	1	1
	3	1	1
Discontinuity ->>	4	4	2
	5	4	2
Discontinuity ->>	6	6	3
	7	6	3
	8	6	3

จากข้อมูลข้างต้น จะประกอบไปด้วย 3 เซ็กเมนต์ คือ เซ็กเมนต์แรกจะประกอบด้วย 3 แพ็กเกจ เซ็กเมนต์ที่สองจะมีแพ็กเกจ 2 แพ็กเกจ และเซ็กเมนต์สุดท้ายมี 3 แพ็กเกจ โดยข้อมูลดังกล่าวเป็นข้อมูลของ OGF เสดเดอร์ที่ถูกสร้างขึ้นโดยวีดีโอบีเอ็ม ไคลเอนต์จะนำข้อมูลในส่วนของ Fseq, Cseq และ Cnum มาทำการวิเคราะห์การเกิด Discontinuity ของข้อมูลในแต่ละกรณีดังนี้

	Fseq	Cseq	Cnum
	1	1	1
	2	1	1
	3	1	1
Discontinuity ->>	5	4	2
Discontinuity ->>	6	6	3
	7	6	3
	8	6	3

จากข้อมูลด้านบนเป็นการสมมติให้เกิดการครอบของแพ็กเกจที่ 4 ในสายสัญญาณ เมื่อไคลเอนต์ตรวจสอบพบที่มีการครอบเกิดขึ้น ไคลเอนต์สามารถที่จะทำการตรวจสอบการเกิด Discontinuity ได้โดยไคลเอนต์จะทำการดูค่าใน Cseq ใน Fseq ที่ 5 ก็จะได้รู้ได้ทันทีว่าเกิด Discontinuity ขึ้นใน Fseq 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	Fseq	Cseq	Cnum
	1	1	1
	2	1	1
	3	1	1
Discontinuity ->>	6	6	3
	7	6	3
	8	6	3

อีกตัวอย่างหนึ่ง หากเกิดการครอปของแพ็คเกจที่ 4 และ 5 ในสายสัญญาณ จากตัวอย่างนี้หากพิจารณาในส่วนขอ Cseq 6 จะไม่มีทางทราบเลยว่าเกิดการ Discontinuity ที่ไหน เพราะการครอปเกิดครอปทั้งเซ็กเมนต์ แต่โคลเอนต์สามารถตรวจสอบการเกิด Discontinuity ได้ โดยการดูค่าของ Cnum ที่มีการเพิ่มขึ้นถึง 2



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ทฤษฎี JDBC และ JDBC Drivers ของ Oracle

4.1 ความหมายของ JDBC

JDBC ก็คือฟังก์ชันมาตรฐานสำหรับการเชื่อมต่อกับฐานข้อมูล มีการสร้างระดับการเชื่อมต่อในการสื่อสารกับฐานข้อมูล หลักการทำงานของ JDBC ตั้งอยู่บนมาตรฐาน X/Open SQL Call-Level Interface ของระบบ X-Windows และ JDBC Driver และต้องเข้ากันได้กับมาตรฐานในการเข้าถึง SQL (ANSI SQL Entry Level Standard) และต้องผ่าน Conformance test ซึ่ง JavaSoft เป็นผู้กำหนดขึ้นและ JDBC (Java Database Connectivity) ก็ถูกพัฒนาโดย JavaSoft Department ของบริษัท Sun Microsystems ด้วยเช่นกัน

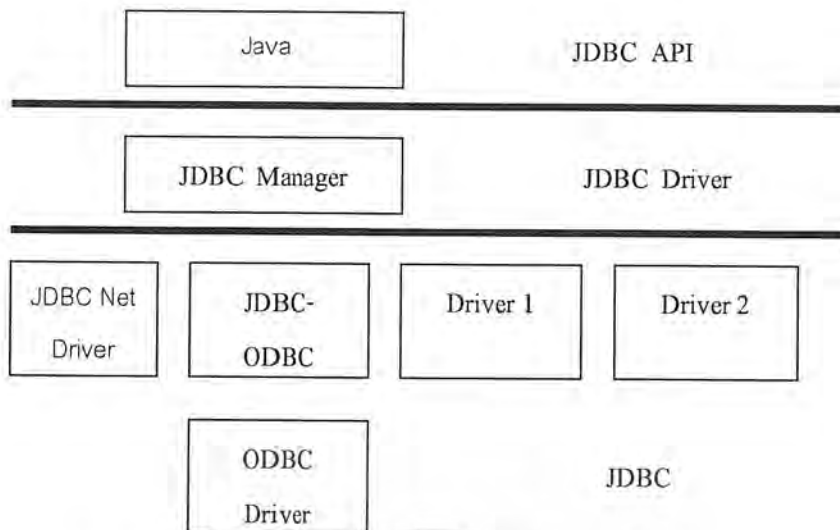
4.2 โครงสร้างของ JDBC

โครงสร้างการเชื่อมต่อภายใน JDBC ประกอบด้วย 3 ระดับหลักคือ JDBC API, JDBC Driver API และ JDBC Driver ดังรูปที่ 4-1 ระดับบนสุด JDBC API เป็นระดับของฟังก์ชัน API ที่อำนวยความสะดวกให้แก่โปรแกรมประยุกต์ ระดับกลาง JDBC Driver (มีไดรฟ์เวอร์ที่ต่างกันอยู่ 4 ชนิด) ที่เหมาะสม

4.3 รูปแบบของ JDBC ไดรฟ์เวอร์

JDBC ไดรฟ์เวอร์(Drivers) หรือส่วนที่อยู่เบื้องหลังการทำงานของฟังก์ชัน API ต่าง ๆ ของ JDBC ถูกจัดแบ่งออกเป็นทั้งหมด 4 ชนิด คือ

1. JDBC – OCBC Bridge
2. Native – API (Partly – Java) Drivers
3. Net – Protocol (All – Java) Drivers
4. Native – Protocol (All – Java) Drivers



รูปที่ 4-1 แสดงระดับการเชื่อมต่อของ JDBC API

4.4 JDBC Drivers ของ Oracle

บริษัท Oracle ได้ทำการพัฒนาตัว JDBC Drivers สำหรับใช้ในการติดต่อระหว่างแอปพลิเคชันและตัวฐานข้อมูลที่พัฒนาขึ้นมาจากภาษาจาวา โดยสามารถที่จะแยกออกเป็นไดร์ฟเวอร์สำหรับไคลเอ็นท์และไดร์ฟเวอร์สำหรับเซิร์ฟเวอร์ ซึ่งไดร์ฟเวอร์สำหรับไคลเอ็นท์จะมีด้วยกัน 2 ชนิด คือ JDBC Thin Driver และ JDBC OCI Driver ส่วนไดร์ฟเวอร์สำหรับเซิร์ฟเวอร์นั้นจะมี Server side JDBC เป็นตัวจัดการให้จาวาเวอร์ชวลแมชชีน (JVM) ติดต่อกับเอนจิน (SQL engine) ซึ่งสามารถแบ่งตัวไดร์ฟเวอร์ออกได้เป็น 3 แบบคือ

1. JDBC Thin Driver จะใช้ Java Sockets ในการติดต่อกับดาตาเบสเซิร์ฟเวอร์ (Database Server) ซึ่งเหมาะสำหรับ Java Applet โดยเมื่อเราทำการเลือก URL จาก HTML page ที่มี Java Applet tag อยู่เว็บเซิร์ฟเวอร์ (Web Server) จะทำการโหลด Java Applet และ JDBC Thin Driver ไปยังไคลเอ็นต์

2. JDBC OCI Driver จะทำการเชื่อมต่อ JDBC กับดาตาเบสเซิร์ฟเวอร์โดยใช้ Oracle Call Interface (OCI) เรียกว่า Call native methods ซึ่งถูกเขียนขึ้นมาด้วยภาษาซี ทำให้ไม่เหมาะที่จะใช้ใน Java Applet แต่จะเหมาะกับ Java Application หรือ Java Middle tiers

3. JDBC Server Driver จะใช้สำหรับ เซิร์ฟเวอร์เท่านั้นซึ่ง Server Driver นี้จะมี JDBC สำหรับโปรแกรมภาษา Java ที่ใช้ในดาตาเบสเซิร์ฟเวอร์ เช่น Java Stored Procedure, Enterprise Java Beans (EJB) , SQL และ PL/SQL

ลำดับในการติดต่อ Database Server ในการ Query ข้อมูลจากตัว Client

- อิมพอร์ตแพ็คเกจ
- ทำการรีจิสเตอร์ JDBC ไดร์ฟเวอร์ Opening a Connection to a Database
- สร้าง สเตทเมนต์ ออบเจกต์ ทำการคิวรีและรับ Result Set Object
- นำ Row ที่อยู่ใน Result Set มาใช้
- ปิด Result Set และ Statement Object จบการเชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5.

การปฏิบัติและการออกแบบแอปพลิเคชัน

การพัฒนาแอปพลิเคชันในโครงการนี้ จะเป็นแอปพลิเคชันสาริตการทำ Video-on-Demand โดยแอปพลิเคชันจะแบ่งออกเป็น 2 ส่วน คือ ส่วนเซิร์ฟเวอร์ และส่วนไคลเอนต์ โดยในโครงการนี้เซิร์ฟเวอร์จะให้บริการวิดีโอในรูปแบบไฟล์ MPEG แก่ไคลเอนต์ ซึ่งไคลเอนต์จะเป็นแอปพลิเคชันที่ถูกพัฒนาขึ้นมาจากภาษาจาวา ที่ใช้สำหรับแสดงข้อมูลวิดีโอที่มาจากเซิร์ฟเวอร์ โดยการพัฒนาแอปพลิเคชันทั้งหมดนี้จะใช้ Oracle Video Server เป็นเครื่องมือช่วยในการพัฒนา

5.1 การพัฒนาด้านเซิร์ฟเวอร์

จากสถาปัตยกรรมของ OVS ซึ่งเป็นแบบ 3 Tier เมื่อนำมาใช้พัฒนาแอปพลิเคชันในโครงการนี้ ทำให้งานในส่วนของเซิร์ฟเวอร์จะครอบคลุมทั้งแอปพลิเคชันเซิร์ฟเวอร์ และ ดาตาเบสเซิร์ฟเวอร์ ในการศึกษาการพัฒนาเซิร์ฟเวอร์นั้น ส่วนใหญ่จะเป็นการศึกษางานของระบบ Oracle Video Server, การบริหารบริการต่างๆของเซิร์ฟเวอร์ รวมไปถึงขั้นตอนการติดตั้ง เพื่อให้ได้มาซึ่งวิดีโอเซิร์ฟเวอร์ที่สามารถรองรับการให้บริการวิดีโอแก่ไคลเอนต์หลายๆไคลเอนต์ได้พร้อมๆกัน โดยสามารถสรุปการทำงาน ได้ดังนี้

5.1.1 ติดตั้งลोजิกคอลคอนเทนต์

หลังจากทำการติดตั้งโปรแกรม OVS แบบสาริตแล้วต้องทำการติดตั้งส่วนของลोजิกคอลคอนเทนต์ เพื่อทำส่วนตัวอย่างของคอนเทนต์ ซึ่งมีขั้นตอนดังนี้

ขั้นที่ 1. สร้าง Service ชื่อ oracle_ovs ในเครื่องแอปพลิเคชันเซิร์ฟเวอร์

ขั้นที่ 2. สร้าง Database_user สำหรับใช้งานโดย OVS

กำหนดให้ User_Name : ovs

Password : oraclevideoserver

Tablespaces\Default : USER_DATA

Tablespaces\Temporary : TEMPORARY_DATA

ขั้นที่ 3. เรียกใช้คำสั่ง vdbuild เพื่อให้ OVS ทำการสร้างตารางในดาตาเบสเซิร์ฟเวอร์โดยมีรายละเอียดดังนี้

```
> vdbuild -T ovs/oraclevideoserver@oracle_ovs
```

ขั้นที่ 4. ทำการแก้ไขไฟล์ D:\ORANT\VS30\ADMIN\mmrc โดยเพิ่มคำสั่งต่อไปนี้

```
vscontsrv.connect=ovs/oraclevideoserver@oracle_ovs
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
vsbcastsrv.connect=ovs/oraclevideosever@oracle_ovs
```

ขั้นที่ 5. ทำการ Restart เซิร์ฟเวอร์

ขั้นที่ 6. ใช้คำสั่ง vscontreg เพื่อสร้างข้อมูลใหม่ในฐานข้อมูล โดยใช้คำสั่งดังต่อไปนี้

```
> vscontreg /mds/video/*.mpi
```

หากทำครบทั้ง 6 ขั้นตอนข้างต้น ส่วนลوجคอลลคอนเทนตจะถูกรวบรวมเสร็จเรียบร้อยแล้ว

5.1.2 การเพิ่มวอลลุ่ม มีขั้นตอนดังนี้

ขั้นที่ 1. ทำการปิดเซิร์ฟเวอร์ทั้งหมด

ขั้นที่ 2. แก้ไขไฟล์ D:\ORANT\VS30\ADMIN\voltab โดยสมมติให้ต้องการเพิ่มวอลลุ่มชื่อ video1 กำหนดให้สตริปมีค่าเท่ากับ 32KB โดยใช้ไฟล์ dsk01.dat เป็นไฟล์วอลลุ่ม สามารถเขียนรายละเอียดได้ดังนี้

```
video1 striped,width=32k,raidsize=1,noparity D:\ORANT\VS30\DATA\dsk01.dat
```

หมายเหตุ ในโครงการนี้จะมีวอลลุ่มทั้งหมด 2 วอลลุ่มคือ video และ video1 ซึ่งวอลลุ่ม video นั้น ถูกสร้างขึ้นในขั้นตอนการติดตั้งโปรแกรม OVS ในแบบสาธิต (Demonstration Version) และวอลลุ่ม video1 ถูกสร้างขึ้นภายหลัง เนื่องจากความจุของวอลลุ่ม video ไม่เพียงพอต่อความต้องการของโครงการ โดยทั่วไปแล้วระบบ Oracle Video Server ที่ใช้ด้านพาณิชย์ต้องการฮาร์ดดิสก์ความเร็วสูง (SCSI) ที่มีความจุราวๆ 2 GB ขึ้นไป มาทำเป็นวอลลุ่มสำหรับเก็บคอนเทนต์ต่างๆ โดยก่อนใช้งานจะต้องนำฮาร์ดดิสก์เหล่านี้ไปผ่านกระบวนการเซตอัปดิสก์ เพื่อให้ได้ดิสก์ที่มีรูปแบบตามที่ทาง Oracle กำหนด แต่ในโครงการนี้จะใช้ดิสก์ที่เป็นลักษณะไฟล์พิเศษที่มีขนาดคงที่ คือ 50 MB ที่ถูกสร้างโดย OVS เองมาทำเป็นวอลลุ่ม โดยไฟล์ D:\ORANT\VS30\DATA\dsk01.dat จะแทนด้วยวอลลุ่ม video ส่วนไฟล์ dsk02.dat ใช้โดยวอลลุ่ม video1

ขั้นที่ 3. เปิดการทำงานเฉพาะ OMN

ขั้นที่ 4. เรียกใช้คำสั่ง mdsvolinit โดยมีรูปแบบการเรียกดังนี้

```
> mdsvolinit -s -t -f D:\ORANT\VS30\ADMIN\voltab video1
```

ขั้นที่ 5. ปิดการทำงานของ OMN แล้วทำการเปิดวีดีโอเซิร์ฟเวอร์อีกครั้ง จะถือว่าเสร็จขั้นตอนการเพิ่มวอลลุ่ม

หมายเหตุ รายละเอียดของข้อมูลภายในไฟล์ voltab สามารถดูได้จากภาคผนวก

5.1.3 การลงทะเบียนคอนเทนต์

ใช้คำสั่ง mdscopy เพื่อทำการคัดลอกคอนเทนต์ลงวอลลุ่มก่อน แล้วใช้คำสั่ง vstag เพื่อทำการสร้างไฟล์เท็กของคอนเทนต์ ดังตัวอย่างการทำการลงทะเบียนให้กับคอนเทนต์ที่ชื่อ ending.mpg สามารถทำได้ดังนี้

```
> mdscopy .\ending.mpg /mds/video
```

```
> vstag /mds/video/ending.mpg /mds/video/ending.mpi
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.4 การสร้างตัวอย่างคอนเทนต์ มีขั้นตอนดังนี้

ขั้นที่ 1. เลือกฟิสิกส์คอลคอนเทนต์ที่ต้องการทำตัวอย่างคอนเทนต์

ขั้นที่ 2. สร้างคลิป โดยการเลือกเอาเพียงบางช่วงของฟิสิกส์คอลคอนเทนต์นั้นๆออกมา อาจจะทำคลิปไปมากกว่าหนึ่งคลิปต่อหนึ่งฟิสิกส์คอลคอนเทนต์ก็ได้

ขั้นที่ 3. สร้างลอจิกคอลคอนเทนต์ซึ่งประกอบจากหนึ่งคลิป หรือหลายๆคลิปมาต่อกัน ซึ่งลอจิกคอลคอนเทนต์นี้จะถูกนำมาใช้เป็นตัวอย่างเป็นตัวอย่างของคอนเทนต์

ขั้นที่ 4. ทำการเพิ่มข้อมูลในฐานข้อมูลที่เราวางตัวอย่างคอนเทนต์ โดยกำหนดชื่อฟิสิกส์คอลคอนเทนต์ ร่วมกับชื่อของลอจิกคอลคอนเทนต์ที่สร้างขึ้นเพื่อเป็นตัวอย่างคอนเทนต์

5.1.5 กลไกการทำงานของระบบโดยรวม

ในส่วนนี้จะอธิบายถึงกลไกการทำงานของระบบว่าจะมีการรับ-ส่งข้อมูลกันอย่างไร โดยจะเริ่มที่ไคลเอนต์ก่อน คือ ในโครงงานนี้มีแอปพลิเคชันที่ถูกพัฒนามาจากไลบรารีภาษาจาวาที่จัดเตรียมไว้โดยระบบ OVS ซึ่งแอปพลิเคชันนี้ถือว่าเป็นส่วนหนึ่งในระบบที่มีกลไกการทำงานร่วมกับส่วนอื่นๆ ระบบ OVS ใช้ CORBA เป็นตัวเชื่อมการทำงานของส่วนต่างๆ ไม่ว่าจะเป็นระหว่างไคลเอนต์และเซิร์ฟเวอร์ หรือเซิร์ฟเวอร์ด้วยกันเอง การที่ OVS ใช้ CORBA เป็นตัวเชื่อมการทำงานก็ส่งผลหลากหลายอย่าง เช่น การกระจายงาน โดยอาศัยโหนดบาลานซ์ หรือการที่เซิร์ฟเวอร์หนึ่งสามารถบริการให้กับไคลเอนต์ต่างๆ ได้พร้อมๆกันหลายๆเครื่อง โดยที่สภาพแวดล้อมของไคลเอนต์เหล่านั้นอาจจะแตกต่างกันในด้านต่างๆ เช่น ระบบปฏิบัติการที่ต่างกัน ฮาร์ดแวร์ที่ต่างกัน เป็นต้น ระบบ OVS จะมีแหล่งเก็บข้อมูลอยู่ 2 ที่ คือที่ดาตาเบสเซิร์ฟเวอร์ และที่แอปพลิเคชันเซิร์ฟเวอร์ แหล่งเก็บข้อมูลทั้งสองแห่งจะเก็บข้อมูลที่ไม่เหมือนกัน กล่าวคือ ในแอปพลิเคชันเซิร์ฟเวอร์จะเก็บข้อมูลที่เป็นไฟล์ MPEG และแท็กไฟล์ ซึ่งในส่วนนี้เป็นข้อมูลหลักของระบบเลขที่เดียว หากเกิดการเสียหายระบบจะไม่สามารถทำงานต่อได้เลย ที่เก็บข้อมูลในแอปพลิเคชันเซิร์ฟเวอร์เราเรียกว่า “วอลลุ่ม” หรือ “ดิสก์ MDS”

ข้อมูลในดาตาเบสเซิร์ฟเวอร์จะเป็นข้อมูลที่เรียกว่า Meta-Data หรืออาจจะอธิบายได้ว่าเป็นข้อมูลที่อธิบายเนื้อหาของข้อมูลหลักอีกที ซึ่งเนื้อหาของข้อมูลหลักจะหมายถึงข้อมูลวิดีโอ MPEG ส่วนข้อมูลที่อธิบายข้อมูลหลักก็อย่างเช่น ข้อมูลที่บอกถึงว่าอัตราการเข้ารหัสของไฟล์ MPEG มีค่าเท่าไร หรือข้อมูลที่บอกถึงขนาดความยาวของไฟล์ MPEG หรืออาจจะเป็นข้อมูลที่บอกถึงว่าหากต้องการเล่นไฟล์ MPEG ดังกล่าว จะมีวิธีการอย่างไรเรียกได้ทีไหน เป็นต้น ข้อมูลภายในดาตาเบสเซิร์ฟเวอร์จะเป็นข้อมูลชนิดพื้นฐาน ซึ่งต่างจากข้อมูลในแอปพลิเคชันเซิร์ฟเวอร์ที่มีลักษณะที่ซับซ้อนกว่า เพราะประกอบไปด้วยภาพและเสียง ดังนั้นกระบวนการที่จะส่งข้อมูลภาพและเสียงไปยังด้านไคลเอนต์จะต้องมีกลไกที่เหมาะสมและมีประสิทธิภาพอย่างดี

กระบวนการติดต่อสื่อสารระหว่างไคลเอนต์และเซิร์ฟเวอร์นั้นสามารถอธิบายได้ดังนี้ หากไคลเอนต์ต้องการเพลย์ไฟล์ MPEG ที่เครื่องเซิร์ฟเวอร์ ไคลเอนต์จะต้องทำการสร้างเส้นทางการติดต่อ หรือ Circuit ขึ้นมาก่อน จากนั้นจึงทำการส่งสัญญาณการร้องขอข้อมูลที่เซิร์ฟเวอร์ การส่งสัญญาณดังกล่าวจะเป็นการส่งผ่านทาง OMN ซึ่งเป็นเทคโนโลยี CORBA ซึ่ง OMN ทางเซิร์ฟเวอร์เมื่อได้รับสัญญาณการร้องขอจากไคลเอนต์ OMNก็จะนำสัญญาณการร้องขอดังกล่าวส่งต่อไปให้กับ OVS เมื่อ OVS ทราบถึงความต้องการของไคลเอนต์แล้ว OVS จะไปดึงข้อมูลจากดาตาเบสเซิร์ฟเวอร์ (กรณีการขอข้อมูลที่เกี่ยวข้องจะดึงจากดาตาเบส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซิร์ฟเวอร์) เมื่อ OVS ได้ข้อมูลตามที่ไคลเอนต์ต้องการแล้ว OVS จะทำการส่งข้อมูลดังกล่าวกลับไปยังไคลเอนต์ ซึ่งกระบวนการรับ-ส่งข้อมูลระหว่างเซิร์ฟเวอร์กับไคลเอนต์นั้น จะเป็นหน้าที่ของ Video Pump

Video Pump เมื่อได้รับหน้าที่ในการส่งข้อมูล Video Pump จะทำการแบ่งข้อมูลออกเป็นขนาดเล็กๆ แล้วทำการเพิ่มข้อมูลในส่วนหัว (OGF header) ให้กับข้อมูลเล็กๆเหล่านั้น เราจะเรียกชิ้นของข้อมูลนี้ว่า “OGF แพ็กเกจ” ข้อมูลภายใน OGF header จะถูกจัดเตรียมไว้สำหรับให้ไคลเอนต์ใช้ในกระบวนการตรวจสอบความผิดพลาดของข้อมูล และใช้เพื่อดึงข้อมูลภาพและเสียงออกจาก OGF แพ็กเกจ

วิธีการรับข้อมูลของไคลเอนต์ก็สามารถทำได้โดยการค้นหาต้นแพ็กเกจ โดยการมองหากลุ่มของข้อมูล “FOF0F003” ซึ่งเป็นตัวบ่งชี้ว่าเป็นต้นแพ็กเกจ OGF จากนั้น โปรแกรมทางด้านไคลเอนต์จะต้องพยายามทำการดึงเอาข้อมูลภายในที่เป็นข้อมูลภาพและเสียงออกมาจาก OGF แพ็กเกจ โดยจะต้องมีการตรวจสอบด้วยว่าข้อมูลที่มาถึงนั้นมีความถูกต้องและครบถ้วนไหม มีแพ็กเกจที่หล่นหายไหม หากมีการหล่นหายเกิดขึ้น จะต้องมีการขอข้อมูลจากเซิร์ฟเวอร์ใหม่อีกครั้งหรือไม่ ทั้งนี้ทั้งนั้นระหว่างการพิจารณาจะต้องมีการนำข้อมูลภาพและเสียงที่เคยถูกแยกเป็นชิ้นเล็กๆ กลับมาประกอบกลับใหม่ให้ได้เหมือนเดิมด้วย เพื่อจะนำข้อมูลเหล่านี้ไปเข้าหน่วยถอดรหัส MPEG อีกต่อหนึ่ง แล้วจึงนำข้อมูลที่ถูกรหัสเรียบร้อยแล้วมาแสดงผลต่อไป ทั้งหมดนี้เป็นกระบวนการทำงานโดยรวมของระบบ OVS ในโครงการนี้

5.2 การพัฒนาในด้านไคลเอนต์

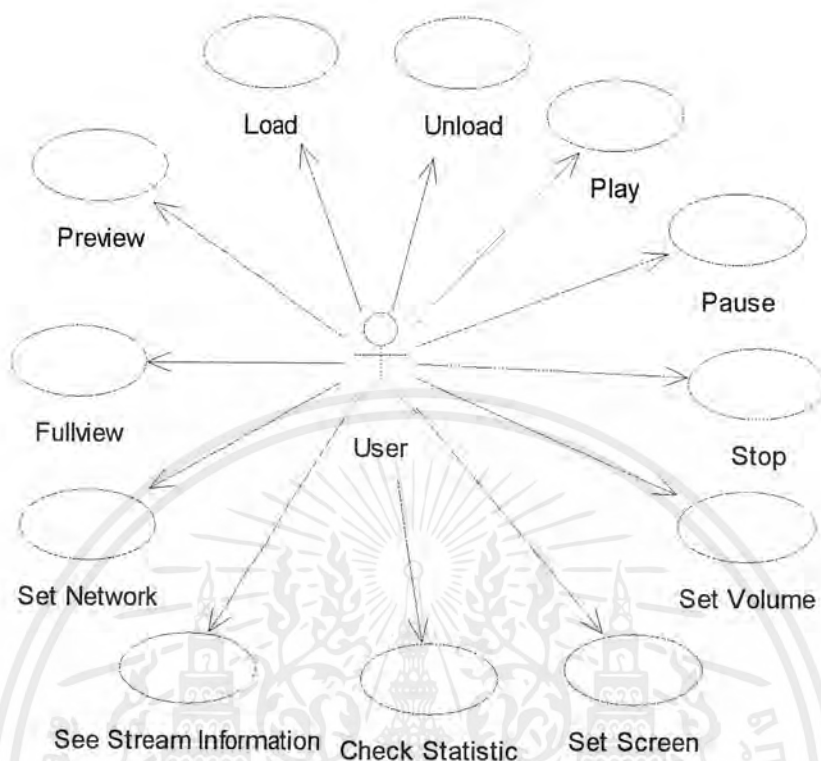
ในส่วนของการพัฒนาแอปพลิเคชันจะทำการพัฒนาให้แอปพลิเคชันสามารถใช้งานร่วมกับระบบ Oracle Video Server ได้ โดยมีเป้าหมายอยู่ที่ให้แอปพลิเคชันที่พัฒนาเสร็จแล้วสามารถทำงานได้ดังนี้

1. ทำการควบคุมการเล่น ไฟล์มัลติมีเดีย (Control Playback) ได้ เช่น การ Play, Pause, Stop เป็นต้น
2. ทำการเล่น ไฟล์มัลติมีเดียทั้งแบบเต็ม(Fullview)และ ไฟล์คลิป(Preview)ได้
3. ตรวจสอบสถิติของสตรีมของการรับส่ง ไฟล์มัลติมีเดียของ ไฟล์มัลติมีเดียที่มาจากฝั่งเซิร์ฟเวอร์ได้
4. ตรวจสอบข้อมูลของไฟล์มัลติมีเดียที่จะนำมาใช้งานได้

โดยภาษาที่ใช้ออกแบบจะใช้ภาษาจาวาและเครื่องมือที่ใช้ในการเขียนภาษาจาวาจะใช้ JDK 1.1.8 ของบริษัท Sun Microsystems และใช้สัญลักษณ์เพื่อประกอบการออกแบบตามทฤษฎี UML ดังนี้

5.2.1 ยูสเคสไดอะแกรม (Use Case Diagram)

แสดงได้ดังรูปที่ 5-1



รูปที่ 5-1 แสดง Use Case Diagram ของระบบ

จาก Uses Case Diagram ที่ได้ออกแบบมานั้นจะเห็นว่าผู้ใช้ซึ่งใช้งานแอปพลิเคชันบนฝั่งไคลเอนต์จะสามารถใช้งานแอปพลิเคชันตามที่แสดงใน Use Case Diagram โดยผู้ใช้ไม่ต้องยุ่งกับการเชื่อมต่อระหว่างเซิร์ฟเวอร์และไคลเอนต์เลย ผู้ใช้รู้เพียงว่าทำอะไรให้สามารถควบคุมไฟล์มัลติมีเดียได้ก็พอ ซึ่งตรงกับหลักการของการ Encapsulation ที่กล่าวในบทที่ 2

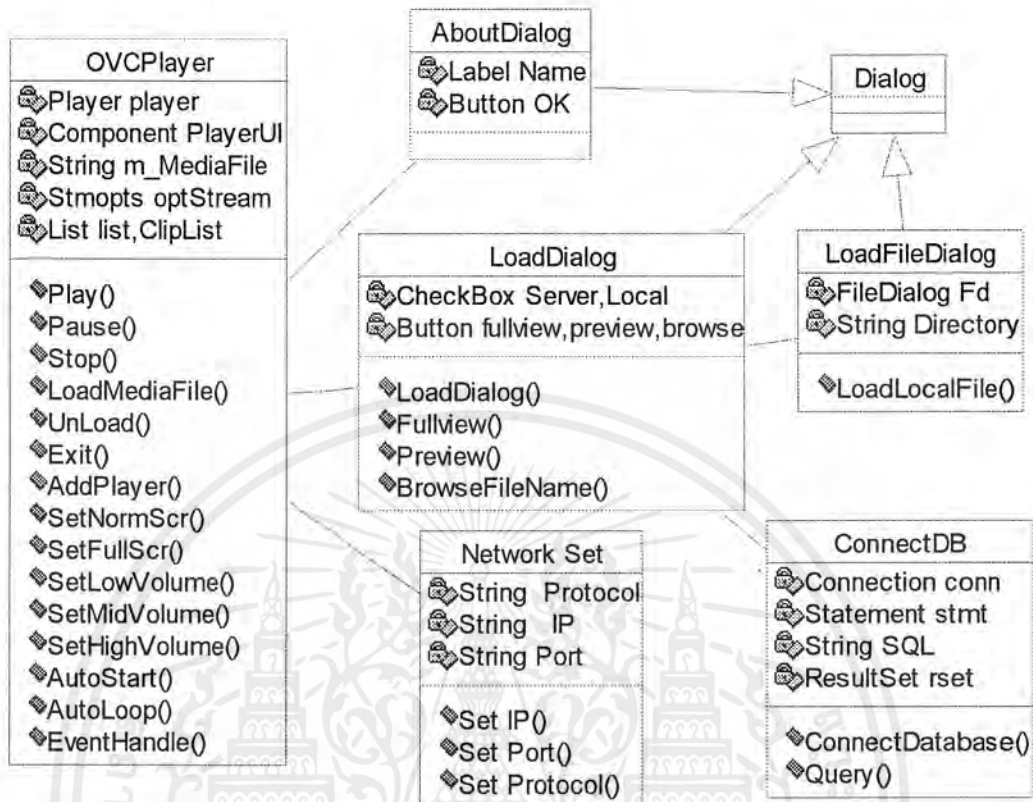
5.2.2 คลาสไดอะแกรม (Class Diagram)

จาก Use Case Diagram การใช้งานของผู้ใช้สามารถกำหนดขึ้นมาเป็นเมธอดและจะรวมไว้ในคลาสเพียงคลาสเดียวซึ่งเป็นคลาสของระบบและจะประกอบด้วยคลาสอื่นๆซึ่งเกี่ยวข้องและทำงานร่วมกับคลาสหลักโดยคลาสเหล่านี้ต้องหน้าที่ดังนี้

- การแสดงผลไฟล์มัลติมีเดียรวมไปถึงการควบคุมการเล่น (Control Playback)
- การโหลดไฟล์เพื่อการเล่น
- การเชื่อมต่อกับดาตาเบสเซิร์ฟเวอร์
- การกำหนดค่าต่างๆที่เกี่ยวกับการเชื่อมต่อเน็ตเวิร์ค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถออกแบบได้ดังนี้



รูปที่ 5-2 แสดงคลาสไดอะแกรม

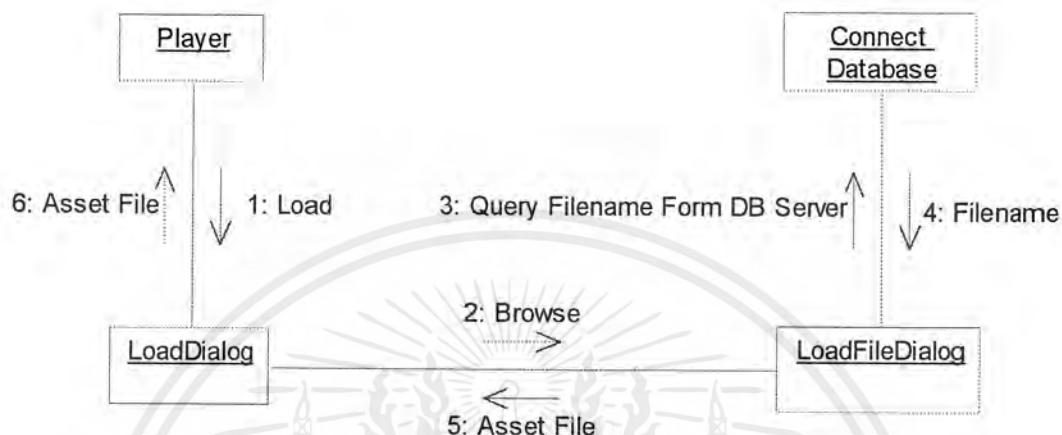
จากรูปที่ 5-2 คลาสไดอะแกรมที่แสดงจะประกอบด้วยคลาสทั้งหมด 6 คลาส คือ คลาส OVCPlayer ,LoadDialog , LoadFileDialog , AboutDialog ,ConnectDB และ NetworkSet ซึ่งสามารถอธิบายความสัมพันธ์ของแต่ละคลาสได้ดังนี้ คลาสหลักของระบบคือคลาส OVCPlayer ซึ่งคลาสนี้จะมีเมธอดที่เกี่ยวข้องกับการควบคุมการเล่นทั้งหมดอยู่และจะเกี่ยวข้องกับคลาส LoadDialog , LoadFileDialog , AboutDialog เพราะเมื่อทำการเรียกเมธอด LoadMediaFile() เมธอดนี้ก็จะทำการหาชื่อไฟล์มีัลติมีเดียที่จะมาแสดง โดยการเรียกคลาส LoadDialog , LoadFileDialog เพื่อทำหน้าที่ หากในกรณีที่การโหลดเป็นการโหลดไฟล์มีัลติมีเดียที่เก็บบนดิสก์เครื่องเดียวกับโคลเอนต์ก็ไม่จำเป็นต้องทำการติดต่อกับดาตาเบส แต่ถ้าทำการโหลดไฟล์มีัลติมีเดียบนเซิร์ฟเวอร์จะต้องมีการเรียกคลาส ConnectDB เพื่อจัดการเกี่ยวกับดาตาเบส และคลาส LoadDialog , LoadFileDialog , AboutDialog จะเกี่ยวกับคลาส Dialog ในแบบสืบทอด (Inheritance) ส่วนการกำหนดค่าที่เกี่ยวข้องกับเน็ตเวิร์คคือ ค่าเลขที่อยู่ของเซิร์ฟเวอร์ ค่าพอร์ต และ ทรานสปอร์ตโพรโตคอล จะจัดการโดยคลาส NetworkSet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.3 คอลลาบอเรชันไดอะแกรม (Collaboration Diagram)

การ call กันที่สำคัญระหว่างคลาสสามารถออกแบบได้ดังนี้

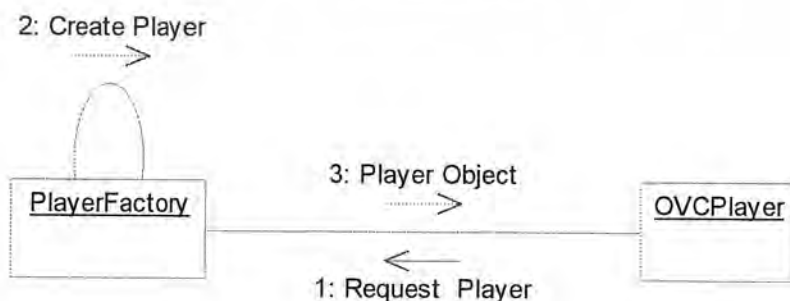
- การโหลดไฟล์จากเซิร์ฟเวอร์



รูปที่ 5-3 แสดงลำดับของการโหลดไฟล์มัลติมีเดียจากเซิร์ฟเวอร์

จากรูป 5-3 เมื่อออกแบบเจ็ท Player ซึ่งเป็นออกแบบเจ็ทที่ทำหน้าที่ควบคุมการเล่นทั้งหมดจะทำการส่งเมสเสจ Load ไปยังออกแบบเจ็ท LoadDialog ออกแบบเจ็ท LoadDialog จะทำการจัดการเกี่ยวกับหาชื่อไฟล์มัลติมีเดียที่เก็บบนเซิร์ฟเวอร์และบนคิสก์บนไคลเอนต์ โดยทำการส่งเมสเสจ Browse ไปยังออกแบบเจ็ท LoadFileDialog ซึ่งการหาชื่อไฟล์จะโดนทำที่ออกแบบเจ็ทนี้รวมไปถึงการเชื่อมต่อกับดาตาเบสด้วย และเมื่อได้ชื่อไฟล์มัลติมีเดียมาก็จะทำการส่งกลับไปยังออกแบบเจ็ท LoadDialog และจะส่งต่อ ไปยังออกแบบเจ็ท Player อีกที

- การสร้างออกแบบเจ็ท Player

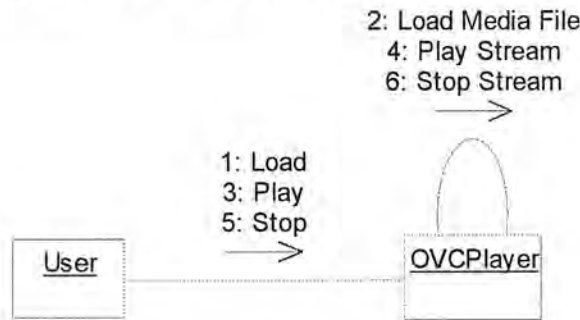


รูปที่ 5-4 แสดงลำดับของการสร้างออกแบบเจ็ท Player

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป 5-4 ลำดับการสร้างออบเจกต์ Player นั้นสามารถทำได้โดยการส่งเมสเซจไปยังคลาส PlayerFactory เพื่อให้คลาสนี้ทำการสร้างออบเจกต์ Player

- การควบคุมการเล่น (Control Playback)



รูปที่ 5-5 แสดงลำดับของการควบคุมการเล่น

5.2.4 อธิบายการทำงานและการอิมพลิเมนต์

คลาสต่างๆมีการทำงานและสามารถอิมพลิเมนต์ดังนี้

5.2.4.1 คลาส OVCPlayer

เป็นพับลิคคลาสและเป็นคลาสหลักของ โปรแกรมซึ่งมีหน้าที่ดังต่อไปนี้

- ทำการสร้าง User Interface คือ Video Screen , Status Bar, Visual Component

```

try {
    playerUI = player.getPlayerUI ();
}
catch ( PlayerException e ) {
    showStatus.setText("exception in getPlayerUI: " + e);
}
  
```

โดยการดึง User Interface นั้นจะรวม User Interface ทั้งสามตัวเป็นออบเจกต์เดียวแล้วนำมาเก็บไว้ในตัวแปร playerUI ซึ่งมีชนิดเป็น Component

- ทำการสร้างเมนูซึ่งประกอบด้วยเมนูดังต่อไปนี้

1. File ซึ่งประกอบด้วยเมนูย่อยคือ Load , Unload , Quit
2. Playback ซึ่งประกอบด้วยเมนูย่อย Play , Pause , Stop , Video Size , Volume
3. Preference ซึ่งประกอบด้วยเมนูย่อย Auto Start , Auto Loop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MenuBar menubar = new MenuBar();
Menu menu = new Menu("File");
menu.add(new MenuItem("Load Content"));
menu.add(new MenuItem("Unload"));
menu.addSeparator();
menu.add(new MenuItem("Quit"));
menubar.add(menu);
menu = new Menu("Playback");
menu.add(new MenuItem("Play"));
menu.add(new MenuItem("Pause"));
menu.add(new MenuItem("Stop"));
menu.addSeparator();
Menu videoSize = new Menu("Video Size");
normScr = new CheckboxMenuItem("Normal");
normScr.setState(true);
videoSize.add(normScr);
fullScr = new CheckboxMenuItem("Full Screen");
videoSize.add(fullScr);
menu.add(videoSize);
Menu volume = new Menu("Volume");
lowVol = new CheckboxMenuItem("Low");
lowVol.setState(true);
volume.add(lowVol);
midVol = new CheckboxMenuItem("Medium");
volume.add(midVol);
hiVol = new CheckboxMenuItem("High");
volume.add(hiVol);
menu.add(volume);
menubar.add(menu);
menu = new Menu("Preferences");
menu.add(new CheckboxMenuItem("Auto Start"));
menu.add(new CheckboxMenuItem("Auto Loop"));
menubar.add(menu);
menu = new Menu("Help");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

menu.add(new MenuItem("About"));
menubar.add(menu);
setMenuBar(menubar);

```

- ทำการสร้างออบเจกต์ Player

```

try {
    player = PlayerFactory.getPlayer ();
}
catch ( UnsatisfiedLinkError e ) {
    showStatus.setText(e.getMessage());
}
catch ( PlayerException e ) {
    showStatus.setText("Unable to create a Player object: " + e);
}

```

- จัดการกับเหตุการณ์ที่เกิดขึ้นกับออบเจกต์ Player ทั้งหมด

```

public boolean handleEvent(Event evt)
{
}
public boolean action(Event evt, Object arg)
{
}

```

เหตุการณ์ที่จะเกิดขึ้นกับออบเจกต์ Player นั้นแบ่งออกเป็นได้ดังนี้

1. เหตุการณ์ที่เกิดกับเมนูซึ่งเหตุการณ์เหล่านี้จะโดนจัดการในเมธอด action
2. เหตุการณ์ที่เกิดกับ Visual Component เหตุการณ์เหล่านี้จะโดนจัดการโดยออบเจกต์ player โดยอัตโนมัติ
3. เหตุการณ์การปิดโปรแกรมจะโดนจัดการโดยเมธอด handleEvent

- Load ไฟล์มีลติมีเดียเพื่อทำการแสดงผล

```

try {
    player.load(m_MediaFile,optStream);
}
catch ( PlayerException e ) {
    showStatus.setText("Could not load content: " + e);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับพารามิเตอร์ `m_MediaFile` นั้นจะระบุตำแหน่งของไฟล์ที่จะทำการโหลดและพารามิเตอร์ `optStream` จะกำหนดลักษณะของการเล่นสตรีมเช่น การวนลูปเมื่อจบสตรีม

5.2.4.2 คลาส About Dialog

เป็นคลาสที่ Inheritance มาจากคลาส Dialog ทำการแสดงผลข้อความของชื่อสมาชิกในกลุ่มที่ทำการพัฒนาโปรเจกต์นี้ขึ้นมา

5.2.4.3 คลาส LoadDialog

ทำหน้าที่ดังต่อไปนี้

- ทำการหาตำแหน่งที่อยู่ของไฟล์มัลติมีเดียบนฝั่งเซิร์ฟเวอร์โดยการ คิวรี่ชื่อไฟล์และชื่อไฟล์คลิปจากดาตาเบสเซิร์ฟเวอร์จากรายชื่อ Preview มาเก็บไว้ในคอมโพเนนท์ List
 - ทำการหาตำแหน่งที่อยู่ของไฟล์มัลติมีเดียบนดิสก์ที่อยู่บนเครื่องไคลเอ็นท์โดยทำการเรียกคลาส LoadFile อีกที
 - กำหนดให้ทำการเล่นไฟล์มัลติมีเดียแบบเต็มคือเล่นตลอดทั้งความยาวไฟล์
- เมื่อมีเหตุการณ์เกิดกับปุ่ม FullView จะเป็นการกำหนดคิวรี่ Simple.m_MediaFile เป็น 2 ลักษณะคือ หากเป็นการดึงไฟล์มัลติมีเดียจากฝั่งเซิร์ฟเวอร์คำสั่งจะเป็นดังนี้

```
Simple.m_MediaFile="vsudp://161.246.6.118:5000/vscontsrv:"
+Simple.list.getSelectedItem();
```

โดยจะต้องเพิ่มการระบุตำแหน่งไฟล์เข้าไปดังนี้

1.1 vsudp : เป็นโปรโตคอลในการติดต่อสื่อสาร

1.2 161.246.6.118 : IP Address ของเซิร์ฟเวอร์

5000 : พอร์ตในการติดต่อ

รูปแบบของการระบุตำแหน่งไฟล์มัลติมีเดียสามารถดูได้จากภาพผนวก

หากเป็นการดึงจากดิสก์บนเครื่องเดียวกับไคลเอ็นท์คำสั่งจะเป็นดังนี้

```
Simple.m_MediaFile=Simple.list.getSelectedItem();
```

- กำหนดให้ทำการเล่นคลิปไฟล์ซึ่งเป็น Logical Content ที่เก็บอยู่บนเซิร์ฟเวอร์ จะมีหลักการเหมือนกับการเล่นไฟล์มัลติมีเดียแบบเต็มต่างกันแค่เอาชื่อไฟล์จากคอมโพเนนท์ ClipList แทนที่จะเอามาจากคอมโพเนนท์ List เพราะคอมโพเนนท์ ClipList จะเก็บชื่อไฟล์คลิปไว้

5.2.4.4 คลาส LoadFile

ทำการหาชื่อไฟล์ที่มีคติมิดีเดียวที่เก็บบนดิสก์เกี่ยวกับโคลเอนต์โดยใช้คลาส FileDialog เพื่อระบุตำแหน่งของไฟล์ที่ต้องการจะเล่น

```
FileDialog fd = new FileDialog(new Frame(), null, FileDialog.LOAD);
fd.setDirectory("D:\\MyDocument\\Project");
fd.setFile("*.mpg");
fd.setVisible(true);
Simple.m_MediaFile=fd.getDirectory()+fd.getFile();
Simple.m_Name=fd.getFile();
```

คลาสนี้จะพิลเตอร์ชื่อไฟล์ที่เก็บใน D:/MyDocument/Project และเป็นสกุล mpg เท่านั้น

5.2.4.5 คลาส ConnectDB

ทำการคิวรี่ชื่อไฟล์ที่มีคติมิดีเดียวและไฟล์คลิปที่เก็บบนดาตาเบสเซิร์ฟเวอร์

```
DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());
Connection conn = DriverManager.getConnection
("jdbc:oracle:thin:@dbserver:1521:orcl", "ovs", "oraclevideoserver");
Statement stmt = conn.createStatement ();
String sql = "select content, clips from preview";
ResultSet rset = stmt.executeQuery (sql);
while (rset.next ())
{
    Simple.list.addItem(rset.getString(1));
    Simple.clipList.addItem(rset.getString(2));
}
conn.close();
```

ขั้นแรกต้องทำการ Register Driver Manager ก่อนที่จะใช้งานจากนั้นก็สร้างการเชื่อมต่อขึ้นมาโดยจะต้องกำหนดชื่อผู้ใช้และรหัสผ่าน ซึ่งหากทั้งชื่อผู้ใช้และรหัสถูกต้องก็ถือว่าการเชื่อมต่อไปยังดาตาเบสเซิร์ฟเวอร์สมบูรณ์ ขั้นตอนต่อไปคือการคิวรี่คอตัมที่ที่ต้องการและผลลัพธ์ของการคิวรี่จะเก็บอยู่ในตัวแปรที่มีชนิดเป็นคลาส ResultSet ซึ่งอินเด็กซ์ของจะขึ้นอยู่กับแถวแรกของผลลัพธ์ การนำผลลัพธ์ไปใช้งานสามารถเรียกคำสั่ง getString (number) โดย number จะเป็นคอตัมที่ได้จากการคิวรี่ ส่วนการเลื่อนอินเด็กซ์ที่ผลลัพธ์ให้ใช้คำสั่ง next() และจบการเชื่อมต่อด้วยคำสั่ง Close

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.4.6 คลาส NetworkSet

ทำการกำหนดค่าในเชื่อมต่อกับเน็ตเวิร์คของระบบ Oracle Video Server ซึ่งรูปแบบของการกำหนดชื่อไฟล์ที่เก็บบนเซิร์ฟเวอร์นั้นสามารถดูได้จากภาคผนวก โดยค่าต่างที่คลาสนี้กำหนดคือ

- ค่าเลขที่อยู่ของเซิร์ฟเวอร์ (IP) เช่น 161.246.6.118
- ค่าพอร์ตที่ใช้ในการติดต่อสื่อสาร เช่น 5000
- ค่าทรานสปอร์ตโพรโทคอล เช่น TCP หรือ UDP

5.2.5 ดาตาเบส

ตารางที่จำเป็นสำหรับการคิวรี่ของออบเจกต์ Player มีหนึ่งตารางคือตารางชื่อ Preview ซึ่งมีแอตทริบิวต์ดังต่อไปนี้

Content	Clips
---------	-------

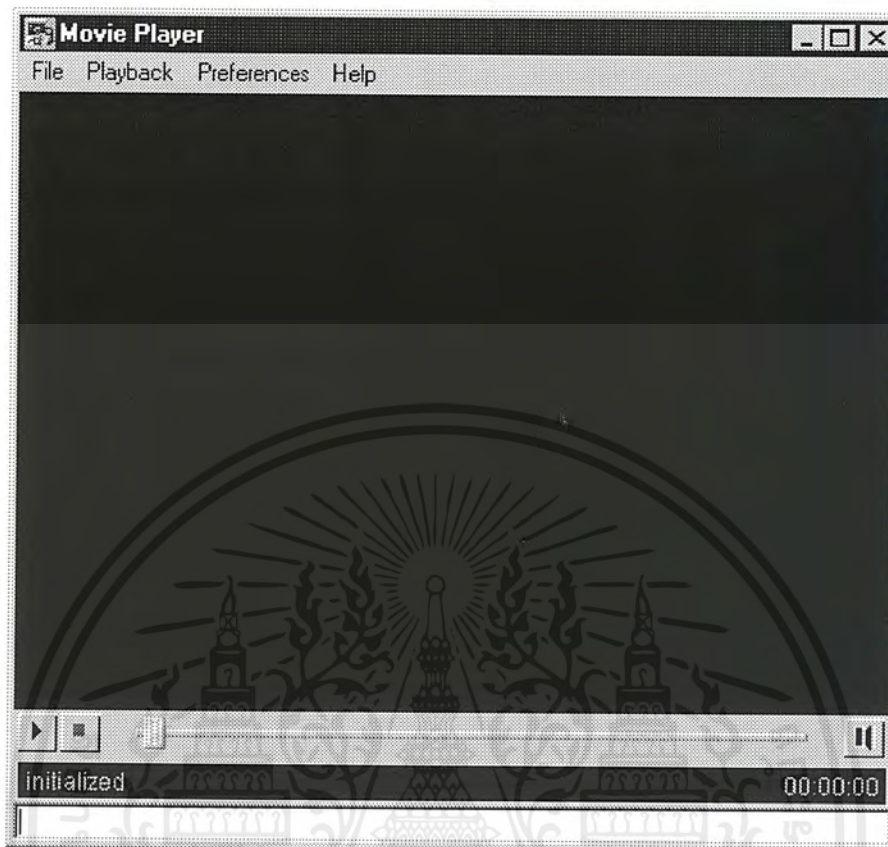
คอลัมน์แรกจะเก็บชื่อไฟล์มัลติมีเดียและคอลัมน์หลังเก็บชื่อไฟล์คลิป โดยทั้งสองคอลัมน์จะมีชนิดเป็น VarChar

สำหรับเหตุผลที่ต้องทำการติดต่อกับดาตาเบสเพื่อดึงชื่อไฟล์มัลติมีเดียทั้งที่มีคลาส Content ซึ่งสนับสนุนการคิวรี่ชื่อไฟล์มัลติมีเดียบนเซิร์ฟเวอร์อยู่แล้ว ก็เพราะในการทำไฟล์คลิปเพื่อทำการดูเพียงตัวอย่างของไฟล์มัลติมีเดียบางส่วนนั้นจะมีไฟล์ต่างๆที่ประกอบการสร้างเกิดขึ้นมาหลายไฟล์ซึ่งเมื่อทำการคิวรี่แล้วจะได้ผลลัพธ์จากการคิวรี่ ซึ่งเป็นชื่อไฟล์พวกนี้ติดมาด้วยจึงต้องใช้การแก้ปัญหาเพื่อให้ผู้ใช้มองเห็นเพียงไฟล์มัลติมีเดียที่ใช้เล่นจริงเท่านั้น จึงจำเป็นต้องสร้างตารางนี้ขึ้นมา

ในส่วนของการแก้ไขตาราง จะทำการแก้ไขคือเมื่อไฟล์มัลติมีเดียที่เก็บอยู่บนเซิร์ฟเวอร์มีการเปลี่ยนแปลงเช่นการนำไฟล์มัลติมีเดียมาใส่เพิ่มในเซิร์ฟเวอร์ผู้ดูแลระบบต้องทำการแก้ไขตารางนี้ด้วยเพื่อให้ข้อมูลสัมพันธ์กันและถูกต้อง

5.2.5 การทดลองและผลการทดลอง

- เมื่อทำการรันแอปพลิเคชันที่พัฒนาขึ้นจะปรากฏหน้าจอดังนี้



รูปที่ 5-6 แสดงหน้าจอหลักของแอปพลิเคชัน

หน้าจอหลักจะประกอบด้วย 5 ส่วนคือ

1. เมนู
2. ส่วนแสดงผล
3. แผงควบคุมการเล่น ประกอบด้วยปุ่ม Play/Pause , Stop บาร์แสดงตำแหน่งของสตรีม และปุ่มปรับระดับความดัง
4. แผงแสดงสถานะของออบเจกต์ Player และแสดงตำแหน่งของสตรีมในรูปของเวลา

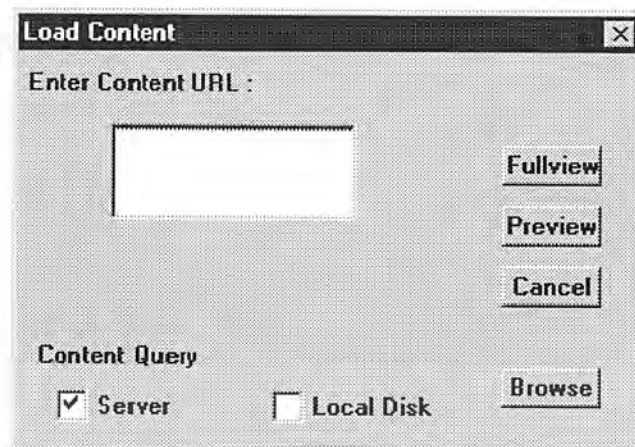
5. ที่แสดงผลข้อความต่างๆเช่น ข้อผิดพลาดที่เกิดขึ้น เป็นต้น
- เมื่อทำการเลือกที่เมนู File จะปรากฏหน้าจอดังนี้



รูปที่ 5-7 แสดงเมนู File

เมื่อเข้ามาดูเมนู File จะมีเมนูย่อยอีก 3 ตัวคือ Load, Unload และ Quit ซึ่งเมื่อทำการเลือกที่เมนูย่อย Quit จะเป็นการออกจากโปรแกรมแต่ถ้าหากเลือกที่เมนู Unload จะทำให้ไฟล์มีลคมีเดียที่โหลดไว้หายไปและสถานะของออบเจกต์ Player จะเปลี่ยนเป็น initialized เหมือนดังรูป

- ทำการเลือกที่เมนู Load จะปรากฏหน้าจอดังนี้



รูปที่ 5-8 แสดงหน้าจอที่เกิดจากการเลือกเมนู Load

ผู้ใช้ต้องทำการหา URL ซึ่งสามารถหาได้จากเซิร์ฟเวอร์และดิสก์บนไคลเอนต์เองด้วยการ Browse

- ทำการ Browse ไฟล์มีลิตมีเดีย จากดิสก์บนไคลเอนต์

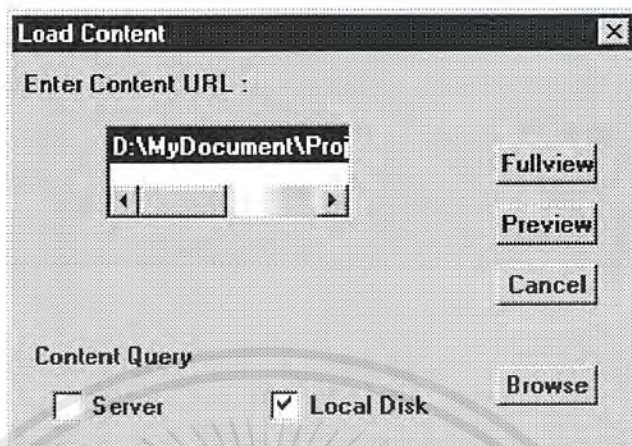


รูปที่ 5-9 แสดงไฟล์ไออะลือกของระบบวินโดวส์

จะปรากฏไฟล์ไออะลือกของระบบวินโดวส์ ซึ่งไฟล์ไออะลือกที่ปรากฏนี้จะแตกต่างกันตามแพลตฟอร์มที่แอปพลิเคชันได้ทำการรันอยู่แอปพลิเคชันจะทำการฟิลเตอร์ไฟล์สกุล MPG โดยอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทำการเลือกไฟล์มัลติมีเดียที่ชื่อ eyes_on_me.mpg แล้วไฟล์ไดอะล็อกจะปิดไปและปรากฏหน้าจอ Load อีกครั้งดังรูป



รูปที่ 5-10 แสดงรายการของไฟล์มัลติมีเดียที่เลือกจากไฟล์ไดอะล็อก

- ทำการเลือก Fullview หรือ Preview เพื่อทำการเล่นไฟล์มัลติมีเดียที่เลือกแล้วหน้าจอไหลจะหายไปและกลับสู่หน้าจอหลักเพื่อแสดงผลดังรูป

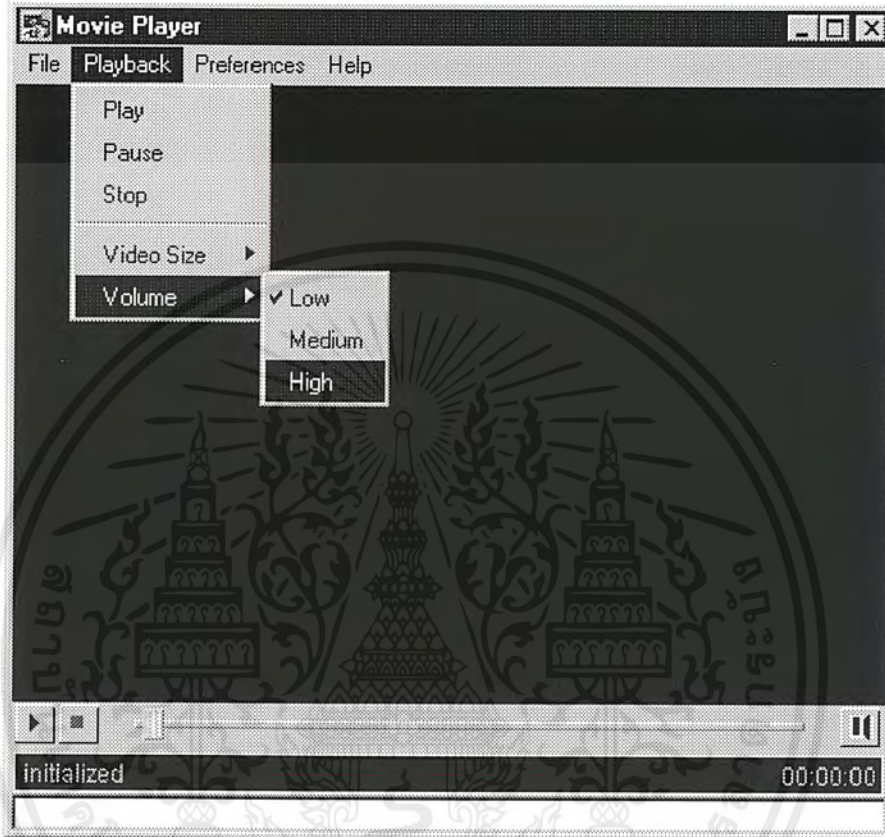


รูปที่ 5-11 แสดงการ Play

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะสังเกตเห็นสถานะของออบเจกต์ Player เปลี่ยนเป็น Playing และบาร์แสดงตำแหน่งสตรีมจะเลื่อนไปเรื่อย ตามเวลาที่เพิ่ม

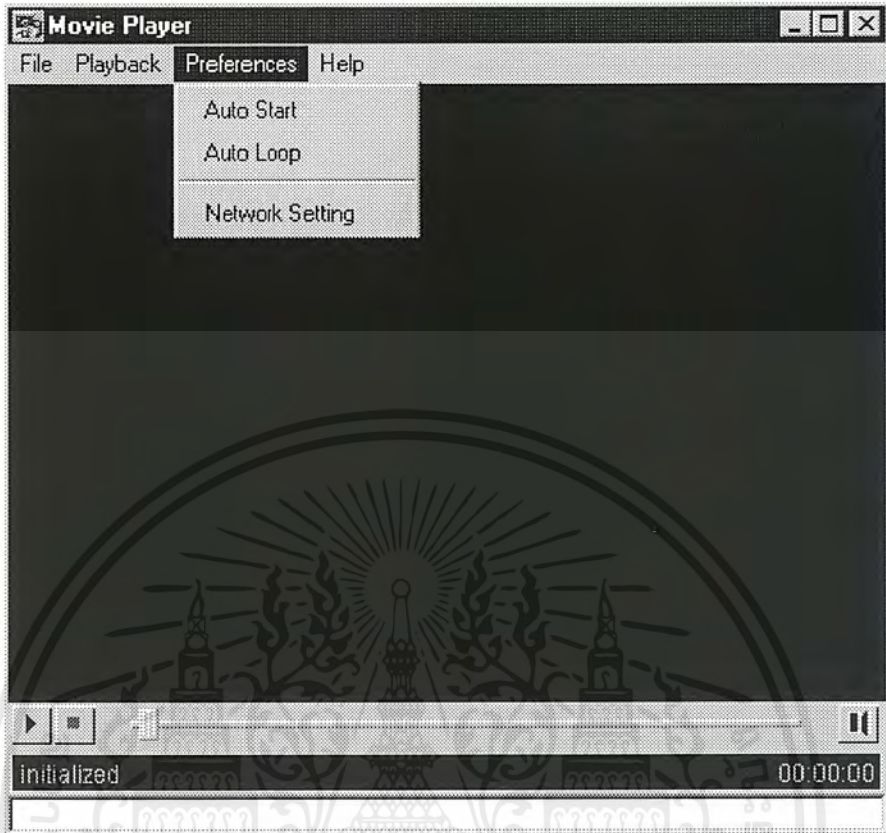
- ทำการเลือกเมนู Playback จะปรากฏหน้าจอ ดังรูป



รูปที่ 5-12 แสดงเมนู Playback

ในเมนูนี้จะประกอบไปด้วยเมนูย่อย Play , Pause , Stop และเมนูในการกำหนดระดับเสียง 3 ระดับคือ ระดับต่ำ , กลาง ,สูง กับขนาดจอภาพ 2 ขนาดคือเต็มจอกับปกติ

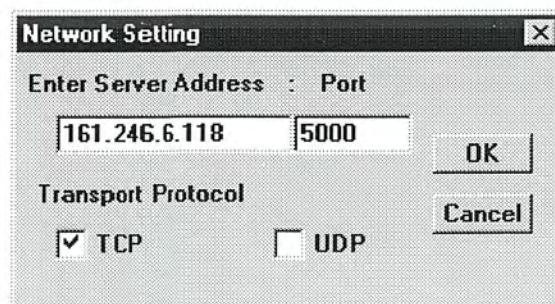
- ทำการเลือกเมนู Preference จะปรากฏหน้าจอ ดังรูป



รูปที่ 5-13 แสดงเมนู Preference

เป็นการกำหนดลักษณะการเล่นของไฟล์มัลติมีเดียคือ Auto Start จะทำการเริ่มเล่นไฟล์มัลติมีเดียทันทีหลังจากที่ Load เสร็จ กับ Auto Loop จะทำการเริ่มเล่นไฟล์ และเมนูย่อย Network Setting เป็นการกำหนดค่าต่างๆที่เกี่ยวข้องกับการเชื่อมต่อกับเน็ตเวิร์ค

- ทำการเลือกเมนู Network Setting จะปรากฏหน้าจอ ดังรูป



รูปที่ 5-14 แสดงเมนู Network Setting

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

สรุปและวิจารณ์การทำงาน

6.1 บทสรุปและปัญหาในการทำงาน

ระบบที่พัฒนาขึ้นมา สามารถทำงาน ได้ตรงกับความต้องการของผู้จัดทำ คือ สามารถจัดการและแสดงผลไฟล์มัลติมีเดียในลักษณะการติดต่อแบบไคลเอนต์เซิร์ฟเวอร์ได้ อีกทั้งความเร็วในการแสดงผลก็มากพอที่ไม่ทำให้เกิดการหยุดของภาพ ความคมชัดของภาพและเสียงก็อยู่ในระดับที่น่าพอใจ

ปัญหาที่เกิดขึ้น ในการทำงานคือ

1. เนื่องจากระบบ Oracle Video Server เป็นระบบที่ยังไม่แพร่หลายมากนัก จึงทำให้ข้อมูล เอกสารที่เกี่ยวข้องยังมีปริมาณน้อย
2. ปัญหาในการพัฒนาแอปพลิเคชันคือ เมื่อได้ทำการพัฒนาแอปพลิเคชัน ไปได้ระยะเวลาหนึ่ง เครื่องที่ใช้เขียน โปรแกรมจะเกิดปัญหาคือไม่สามารถรัน โปรแกรมที่พัฒนาขึ้นมา ได้ ซึ่งตรงนี้ผู้พัฒนายังไม่ทราบสาเหตุของปัญหา
3. ไม่สามารถทำการทดสอบประสิทธิภาพของระบบได้เช่น การตรวจสอบความเร็วของระบบ โดยทำการดึงไฟล์มัลติมีเดียจากเซิร์ฟเวอร์ด้วยไคลเอนต์จำนวนมาก เนื่องจากไม่มีเครื่องคอมพิวเตอร์ในการติดตั้งแอปพลิเคชันมากพอ

6.2 แนวทางการพัฒนาต่อไป

แอปพลิเคชันที่พัฒนาเสร็จแล้วนี้ยังใช้ความสามารถของระบบ Oracle Video Server ไม่เต็มที่หากทำการศึกษา Oracle Video JAVA Library ให้ดีจะพบว่าระบบยังมีลูกเล่นอื่นๆอีก และแอปพลิเคชันที่พัฒนาเสร็จแล้วนี้พัฒนาบนแอปพลิเคชันของจาวา ซึ่งระบบ Oracle Video Server ยังมีการพัฒนาในหลายรูปแบบเช่น Applet, Active X เป็นต้น

ภาคผนวก ก

การติดตั้งและใช้งาน Oracle Video Server

1. วิธีการติดตั้ง Oracle Video Server

ก่อนการติดตั้งจะต้องตรวจสอบก่อนว่าเครื่องที่จะทำการติดตั้ง OVS นั้นมีทรัพยากรตรงกับความต้องการ หรือมีปริมาณเพียงพอหรือไม่ต่อระบบเสียก่อน โดยพิจารณา ดังนี้

1. CPU จะต้องเป็น Intel-base Pentium 166 MHz หรือสูงกว่า
2. หน่วยความจำหลัก (RAM) จะต้องไม่ต่ำกว่า 48 MB หากต้องการจะให้ทำงานได้อย่างมีประสิทธิภาพต้องใช้หน่วยความจำ 128 MB
3. พื้นที่ของฮาร์ดดิสก์ที่จะติดตั้งเฉพาะ OVS และ VSM จะต้องมีเนื้อที่ว่างอย่างน้อย 150 MB หากต้องการคอนเทนส์ตัวอย่างด้วยจะต้องใช้พื้นที่ว่างอย่างน้อย 345 MB
4. ฮาร์ดดิสก์สำหรับเก็บข้อมูลมีคิมเดียว ควรจะมีความจุ 2GB ขึ้นไป จะให้ดีกว่าเป็นแบบ SCSI หากติดตั้ง OVS ในเวอร์ชันสาริต ไม่จำเป็นต้องมีในส่วนนี้ก็ได้ แต่ MDS จะมีข้อจำกัดที่ความจุ 50MB โดย OVS จะทำการสร้างไฟล์ดิสก์ MDS ไว้ที่ราก D:\Orant\Vs30\Data\dsk00.dat
5. เครื่องจะต้องมีระบบเครือข่ายที่สนับสนุน TCP/IP หรือ ระบบเครือข่ายแบบ ATM
6. ไดรฟ์ CD-ROM เพื่อใช้สำหรับการติดตั้ง OVS
7. มีระบบปฏิบัติการของเครื่องเป็น Windows NT 4.0 พร้อมติดตั้ง Service Pack 3 สำหรับ OVS ที่ติดตั้งบน Windows NT
8. มีฐานข้อมูล Oracle RDBMS Server v7.3.3.1 หรือสูงกว่า และมีการติดตั้ง Oracle SQL*Net Server v2.3.3.0 หรือสูงกว่า และติดตั้ง Required Support Files v7.3.3.1

1.1 ขั้นตอนการติดตั้ง Oracle Video Server

ขั้นที่ 1. ทำการใส่แผ่นซีดีรอม Oracle Video Server ในไดรฟ์ (สมมติไดรฟ์ซีดีรอมเป็น ไดรฟ์ E:)

หมายเหตุ ถ้าไดรฟ์ซีดีรอมอยู่อีกเครื่องหนึ่งในระบบเครือข่ายจะต้องทำการแมบ ไดรฟ์เสียก่อน

ขั้นที่ 2. เลือก Run... จาก เมนู Start ของ Windows NT

ขั้นที่ 3. ทำการเรียกโปรแกรม E:\WIN32\INSTALL\SETUP.EXE หลังจากนั้น โปรแกรมติดตั้งจะให้เลือกภาษาที่จะใช้ในผลิตภัณฑ์ (ถ้าไม่เลือกโปรแกรมจะกำหนดให้เป็นภาษาอังกฤษ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 4. โปรแกรมติดตั้งจะให้กำหนดรากลไคระทอรีที่จะใช้ในการติดตั้งผลิตภัณฑ์ของออรากิล ซึ่งในกรณีนี้หมายถึงรากลไคระทอรีที่จะติดตั้งโปรแกรม Oracle Video Server รากลไคระทอรีดังกล่าว เรียกว่า “ORACLE_HOME” สมมติให้เป็นรากลไคระทอรี D:\ORANT จากนั้นให้ทำการกดปุ่ม OK

หลังจากการกำหนดรากลไคระทอรี ORACLE_HOME แล้ว โปรแกรมจะแสดงตัวเลือกของประเภทของการติดตั้ง โดยจะมี 2 ประเภทคือ ประเภทสาริตและประเภทเลือกผลิตภัณฑ์ ซึ่งความแตกต่างของทั้งสองประเภทนี้ คือประเภทสาริตจะเป็นประเภทที่ใช้ทดลองก่อนการติดตั้งแบบประเภทเลือกผลิตภัณฑ์ ซึ่งหากเครื่องใดไม่เคยติดตั้ง OVS เลย จะต้องทำการติดตั้งประเภทสาริตก่อน ประเภทสาริตจะไม่สามารถทำลอจิกคอลคอนเทนต์และคลิปได้ อีกทั้งขนาดความจุของดิสก์ MDS จะถูกกำหนดที่ 50 MB เท่านั้น แต่หลังจากติดตั้งประเภทสาริตไปแล้ว สามารถที่จะทำการการติดตั้งส่วนของการทำลอจิกคอลคอนเทนต์ และสามารถกำหนดขนาดของดิสก์ MDS ใหม่ได้ในภายหลัง

ขั้นที่ 5. กำหนดให้เลือกประเภทสาริต (Install Demonstration Version) หลังจากเลือกเสร็จ โปรแกรมติดตั้งจะทำการติดตั้งโปรแกรมต่างๆที่จำเป็นให้แบบอัตโนมัติ แต่หากเครื่องใดที่เคยติดตั้ง Oracle Video Server มาแล้ว ในขั้นตอนนี้สามารถเลือกการติดตั้งประเภทเลือกผลิตภัณฑ์ได้ โยสิ่งที่จะต้องรู้ในการติดตั้งแบบเลือกผลิตภัณฑ์ จะมีดังนี้

- หมายเลข IP แอดเดรสของเซิร์ฟเวอร์ จะต้องกำหนดเมื่อมีการติดตั้ง Oracle Media Net ตัวอย่างของ IP แอดเดรส เช่น 161.246.6.118 หรือ venus08
- หมายเลขพอร์ตของเซิร์ฟเวอร์UDP จะมีค่าได้ตั้งแต่ค่า 1025 ถึง 65535 หากไม่กำหนดค่าพอร์ต โปรแกรมจะให้ค่าพอร์ตที่ 5000
- ชื่อของวอลุ่มMDS โดยชื่อที่ตั้งจะมีจำนวนตัวอักษรตั้งแต่ 1 ถึง 32ตัว หากไม่กำหนด โปรแกรมจะให้ชื่อวอลุ่มเป็น video
- จำนวนของดิสก์แบบ SCSI ที่ใช้ ค่าเริ่มต้นที่ 1 (จะมีหรือไม่มีก็ได้)
- จำนวนของดิสก์ที่จะใช้เป็นดิสก์ MDSซึ่งจะเก็บคอนเทนต์ หากไม่กำหนดค่าจะเท่ากับ 1
- ขนาดของสตริป มีค่าได้2ค่าคือ 32KB และ 64KB หากไม่มีการกำหนด โปรแกรมจะให้ขนาดสตริปเท่ากับ 32KB
- การกำหนดการทำในส่วนของParity (RAID)ว่าจะให้มีในระบบใหม่ เป็นการป้องกันการสูญหายของข้อมูลเมื่อดิสก์ MDS เกิดปัญหาไม่สามารถทำงานต่อไปได้
- ขนาดของ RAID ที่จะใช้ในดิสก์MDS
- จำนวนของดิสก์สำรองในระบบที่จะใช้เก็บข้อมูลชุดสำรอง

หลังจากการติดตั้งโปรแกรม OVS เสร็จแล้ว เราจะมาดูวิธีการเปิดเซิร์ฟเวอร์ให้บริการ และปิดบริการของเซิร์ฟเวอร์ ดังนี้

2. การใช้งาน Oracle Video Server

เนื่องด้วยโปรแกรม Oracle Video Server มีการทำงานหลากหลายหน้าที่ และมีการใช้งานในหลายๆด้าน และในโครงการนี้ได้นำเอาการทำงานเพียงบางส่วนของ Oracle Video Server มาใช้งาน ดังนั้นในภาคผนวกนี้จะยกเอาเฉพาะฟังก์ชันการทำงานที่จำเป็นต่อการศึกษาโครงการนี้มาเท่านั้น ในบางส่วนจะไม่ขอพูดถึง หากผู้ที่ต้องการศึกษาในฟังก์ชันที่ไม่ได้กล่าวถึง สามารถค้นคว้าได้จากคู่มือการดูแลและบริหารที่จะมีมาพร้อมกับโปรแกรม Oracle Video Server

2.1 การเปิดเซิร์ฟเวอร์ OVS

การเปิดเซิร์ฟเวอร์จะสามารถเปิดได้ 2ทาง คือ เรียกจากเมนู Start ของ Windows NT และเรียกทางคอมมานไลน์ การเรียกทางเมนู Start สามารถแสดงได้ดังนี้



การเปิดเซิร์ฟเวอร์ทางคอมมานไลน์สามารถทำได้ ดังนี้

```
D:\ORANT\VS30\ADMIN\OVSSTART
```

(กำหนดให้ ORACLE_HOME = D:\ORANT)

การเปิดเซิร์ฟเวอร์ขึ้นต้นนั้นจะเป็นการเรียกการทำงานเฉพาะในส่วนของ OVS และ OMN เท่านั้น ส่วนของการบริหารระบบผ่านทางโปรแกรม VSM จากเครื่องไคลเอนต์นั้นยังไม่สามารถใช้งานได้ หากต้องการใช้โปรแกรม VSM จะต้องทำเปิดเซิร์ฟเวอร์ที่ Shortcut "OVS and VSM Startup"

2.2 การปิดเซิร์ฟเวอร์

การปิดเซิร์ฟเวอร์ทางเมนูสามารถเรียกได้ดังแสดงในรูปข้างต้น หากต้องการปิดทางคอมมานไลน์สามารถทำได้โดยใช้คำสั่ง ดังต่อไปนี้



```
D:\ORANT\VS30\ADMIN\OVSTOP
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 การเปิดและปิด OMN

การเปิดและปิด OMN จะใช้ในกรณีที่เรารู้จำเป็นต้องทำการ Restart ระบบใหม่เฉพาะส่วน เช่น กรณีการเพิ่มวอลลุ่มในระบบ จะต้องทำการปิดเซิร์ฟเวอร์ทั้งหมดก่อน แล้วทำการเปิด OMN เพื่อเชื่อมการทำงานภายในเฉพาะส่วน เป็นต้น

การเปิด OMN ทางคอมมานด์ไลน์สามารถทำได้โดยใช้คำสั่ง ดังนี้

```
D:\ORANT\MEDIANET\ADMIN\MNSTART
```

การปิด OMN ใช้คำสั่งต่อไปนี้

```
D:\ORANT\MEDIANET\ADMIN\MNSTOP
```

2.4 คำสั่ง mdmdir

คำสั่ง mdmdir ใช้สำหรับดูรายละเอียดเกี่ยวกับวอลลุ่มของ MDS และไฟล์ที่อยู่ภายในวอลลุ่มนั้นๆ ด้วย

รูปแบบ `mdmdir [-a] [-l] [volume-name]`

เมื่อ `-a` : จะแสดงรายชื่อของไฟล์ทั้งหมดในวอลลุ่ม

`-l` : จะแสดงรายชื่อไฟล์ทั้งหมด รวมทั้งรายละเอียดต่างๆ อย่างเช่น ขนาด วันที่ และ

สถานะ เป็นต้น

`volume-name` : เป็นชื่อของวอลลุ่มที่ต้องการดูรายละเอียด

ตัวอย่างการใช้งาน

ตัวอย่างการใช้ `-a` เพื่อไฟล์ในวอลลุ่ม video1

```
> mdmdir -a /mds/video1
```

```
Volume /mds/video1 (rw): 2 matches
```

```
ending.mpg ending.mpi
```

ตัวอย่างการใช้ `-l` เพื่อไฟล์ในวอลลุ่ม video1

```
> mdmdir -l /mds/video1
```

```
Volume /mds/video1 (rw): 2 matches
```

```
48m Feb 20 01:30:33 rw ending.mpg
```

```
30k Feb 20 01:33:02 rw ending.mpi
```

2.5 คำสั่ง mdscopy

คำสั่ง mdscopy ใช้สำหรับการคัดลอก(Copy)ไฟล์ จากที่หนึ่งไปยังอีกที่หนึ่ง ไฟล์อาจจะอยู่ในวอลลุ่มหรือจะเป็นไฟล์ที่อยู่นอกวอลลุ่มก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบ `mdscopy filename1 filename2`

เมื่อ filename1 : ชื่อไฟล์ที่จะคัดลอก โดยมีรูปแบบดังนี้

`/mds/ชื่อวอลลุ่ม/ชื่อไฟล์`

filename2 : ชื่อไฟล์ปลายทาง

หมายเหตุ ชื่อไฟล์ที่จะใช้เป็นพารามิเตอร์สามารถที่จะใช้ "*" (wildcard) แทนชื่อไฟล์ต่างๆได้

ตัวอย่างการใช้งาน

ตัวอย่างการคัดลอกไฟล์คอนเทนต์ ending.mpg จากวอลลุ่ม video ไปยังวอลลุ่ม video1

```
> mdscopy /mds/video/ending.mpg /mds/video1/ending.mpg
```

ตัวอย่างการคัดลอกไฟล์ภายในวอลลุ่ม video ทั้งหมด ไปยังวอลลุ่ม video1

```
> mdscopy /mds/video/* /mds/video1
```

2.6 คำสั่ง mdsdelete

คำสั่ง mdsdelete ใช้สำหรับการลบไฟล์ข้อมูลภายในวอลลุ่ม

รูปแบบ `mdsdelete file-name`

เมื่อ file-name : ชื่อไฟล์ที่ต้องการลบ

ตัวอย่างการใช้งาน

ตัวอย่างการลบไฟล์ ending.mpg ออกจากวอลลุ่ม video

```
> mdsdelete /mds/video/ending.mpg
```

ตัวอย่างการลบไฟล์ทั้งหมดที่มีนามสกุล .mpg ออกจากวอลลุ่ม video

```
> mdsdelete /mds/video/*.mpg
```

2.7 คำสั่ง vstag

คำสั่ง vstag ใช้สำหรับการสร้างไฟล์แท็กให้แก่ไฟล์ MPEG หรือ OSF

รูปแบบ `vstag input-filename output-filename`

เมื่อ input-filename : ชื่อไฟล์สกุล mpg ซึ่งเป็นไฟล์ที่จะนำมาสร้างไฟล์แท็ก

output-filename : ชื่อไฟล์แท็กที่จะถูกสร้างขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการใช้งาน

ตัวอย่างการสร้างไฟล์แท็กของคอนเทนต์ ending ภายในวอลครุม video

```
> vstag /mds/video/ending.mpg /mds/video/ending.mpi
```

2.8 คำสั่ง vsmkosf

คำสั่ง vsmkosf ใช้สำหรับกรณีต้องการแปลงไฟล์ AVI ไปเป็นไฟล์ OSF โดยคำสั่ง vsmosf จะทำการสร้างไฟล์แท็กและจะทำการคัดลอกเก็บไว้ในวอลครุมให้อัตโนมัติ

รูปแบบ `vsmkosf input-filename output-filename`

เมื่อ input-filename : ชื่อไฟล์ AVI ที่จะแปลงเป็นไฟล์ OSF

output-filename : ชื่อไฟล์ OSF ที่จะสร้างขึ้น

ตัวอย่างการใช้งาน

ตัวอย่างการแปลงไฟล์ sample01.avi ไปเป็นไฟล์ OSF

```
> vsmkosf .\sample01.avi /mds/video
```

หมายเหตุ การทำตามคำสั่งข้างต้นจะมีการสร้างไฟล์แท็ก และทำการคัดลอกลงวอลครุม video ให้อัตโนมัติ

2.9 คำสั่ง vsdbbuild

คำสั่ง vsdbbuild ใช้สำหรับการสร้างตาราง(Schema)ของ OVS บนฐานข้อมูล (Oracle Database) เพื่อใช้ในกรณีการทำลอคคอสคอนเทนต์

รูปแบบ `vsdbbuild [-hT] connect_string`

เมื่อ -h : แสดงวิธีการใช้งานของคำสั่ง

-T : แสดงผลของคำสั่ง SQL *Plus ที่ทำงานตลอดระยะเวลาการทำงานของคำสั่ง

connect_string : ใช้สำหรับบ่งบอกถึงตำแหน่งของฐานข้อมูล และระบุผู้ใช้ที่จะเข้าไปใช้ฐานข้อมูลด้วย โดย connect_string จะมีรูปแบบ `username/password@alias`

เมื่อ username : เป็นรหัสของผู้ใช้ที่ใช้ในการเข้าใช้ระบบ

password : เป็นรหัสผ่านที่ใช้แสดงตัวผู้ใช้

alias : เป็น TNS alias ที่ใช้เจาะจงถึงฐานข้อมูล โดยจะมีรายละเอียดอยู่ในไฟล์

`tnsname.ora`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการใช้งาน

ตัวอย่างการสร้างตารางให้กับยูสเซอร์ชื่อ ovs ในเซิร์ฟเวอร์ oracle_ovs

```
> vsdbuild -T ovs/oraclevideoserver@oracle_ovs
```

2.10 คำสั่ง vscontreg

คำสั่ง vscontreg ใช้เมื่อมีการเพิ่มความสามารถในการทำลจิกคอลคอนเทนต์ โดยคำสั่ง vscontreg จะทำการสร้างข้อมูลต่างๆในฐานข้อมูล เมื่อมีการใช้งานลจิกคอลคอนเทนต์ครั้งแรก

รูปแบบ `vscontreg [-h] tagfile`

เมื่อ `-h` : แสดงวิธีการใช้งานคำสั่ง
`tagfile` : ชื่อแท็กไฟล์ที่จะทำการสร้าง

ตัวอย่างการใช้งาน

ตัวอย่างการสร้างแท็กไฟล์ทั้งหมดในวอลลุ่มใหม่ หลังจากการติดตั้งลจิกคอลคอนเทนต์เป็นครั้งแรก

```
> vscontreg /mds/video/*.mpi
```

2.11 คำสั่ง mdsvolinit

คำสั่ง mdsvolinit ใช้ในกรณีการเซตอัพวอลลุ่ม MDS ซึ่งจะมีรายละเอียดอยู่ในไฟล์ voltab หรืออาจใช้ในการทำการลบข้อมูลในตารางชื่อคอนเทนต์ภายในวอลลุ่มก็ได้

รูปแบบ `mdsvolinit [-s] [-t] [-h] -f voltab volume...`

เมื่อ `-s` : จะใช้กำหนดรายละเอียดของวอลลุ่มในกรณีการสร้างวอลลุ่มใหม่
`-t` : ทำการเซตอัพตารางชื่อคอนเทนต์
`-h` : แสดงรายละเอียดของคำสั่ง
`-f` : ใช้กำหนดไฟล์ voltab ที่จะใช้งาน
`voltab` : ตำแหน่งของไฟล์ voltab ที่ใช้งาน เช่น D:\ORANT\VS30\ADMIN\VOLTAB
`volume` : ชื่อของวอลลุ่มที่จะกระทำ

ตัวอย่างการใช้งาน

ตัวอย่างการเซตอัพวอลลุ่ม video ซึ่งเป็นวอลลุ่มใหม่ ที่จะทำการล้างข้อมูลทั้งหมดออกจากวอลลุ่ม และรายละเอียดของการเซตอัพ ในไฟล์ voltab ที่รากไดเรกทอรี D:\ORANT\VS30\ADMIN

```
> mdsvolinit -s -t -f D:\ORANT\VS30\ADMIN\voltab video
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.12 คำสั่ง vscontsrv.connect

คำสั่ง vscontsrv.connect จะเป็นคำสั่งที่ใช้สำหรับการเชื่อมต่อกับฐานข้อมูล เพื่อให้ไคลเอนต์สามารถทำการคิวรีไฟล์ของวอลคุ่มในฐานข้อมูลได้

รูปแบบ vsbcastsrv.connect = *database_user/password@alias*

เมื่อ database_user : ชื่อของผู้ใช้งานในระบบฐานข้อมูล

password : รหัสผ่านผู้ใช้งาน

alias : ชื่อบ่งบอกเซิร์ฟเวอร์ฐานข้อมูล

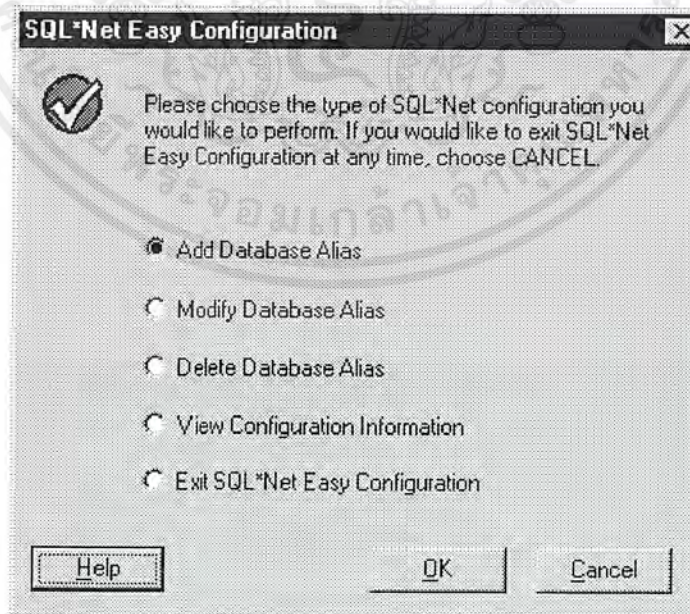
ตัวอย่างการใช้งาน

```
> vscontsrv.connect=ovs/oraclevideosever@oracle_ovs
```

จากตัวอย่างแสดง ผู้ใช้ชื่อ “ovs” โดยมีรหัสผ่านเป็น “oraclevideosever” และเซิร์ฟเวอร์ฐานข้อมูล คือ “oracle_ovs”

3. การใช้งานโปรแกรม SQL *Net Easy Configuration

โปรแกรม SQL *Net Easy Configuration เป็นโปรแกรมช่วยในการติดตั้งการเชื่อมต่อกับเซิร์ฟเวอร์ฐานข้อมูลให้ง่ายขึ้น โดยการทำให้ Database Alias ซึ่งโปรแกรมจะมีฟังก์ชันการทำงานดังนี้ คือ



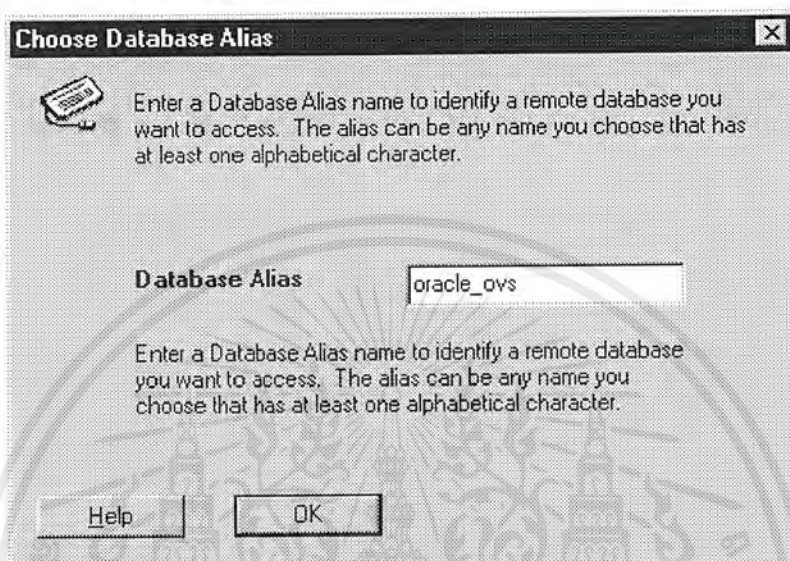
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 สร้าง Database Alias

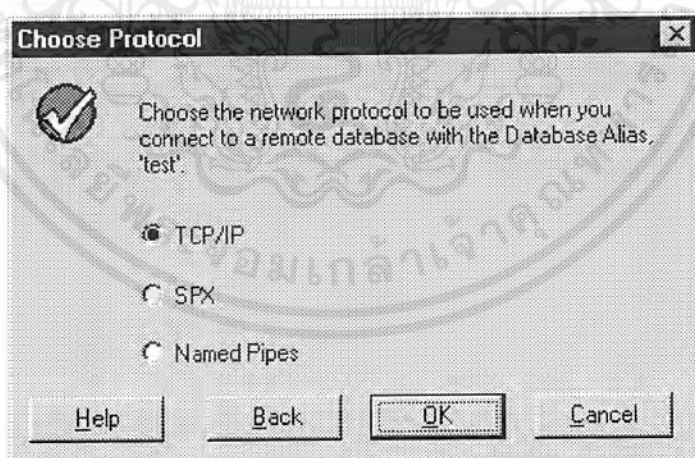
การสร้าง Database Alias มีขั้นตอนดังนี้

ขั้นที่ 1. เลือก “Add Database Alias”

ขั้นที่ 2. ตั้งชื่อ Database Alias ตัวอย่างเช่น “oracle_ovs” เป็นต้น แล้วคลิก OK

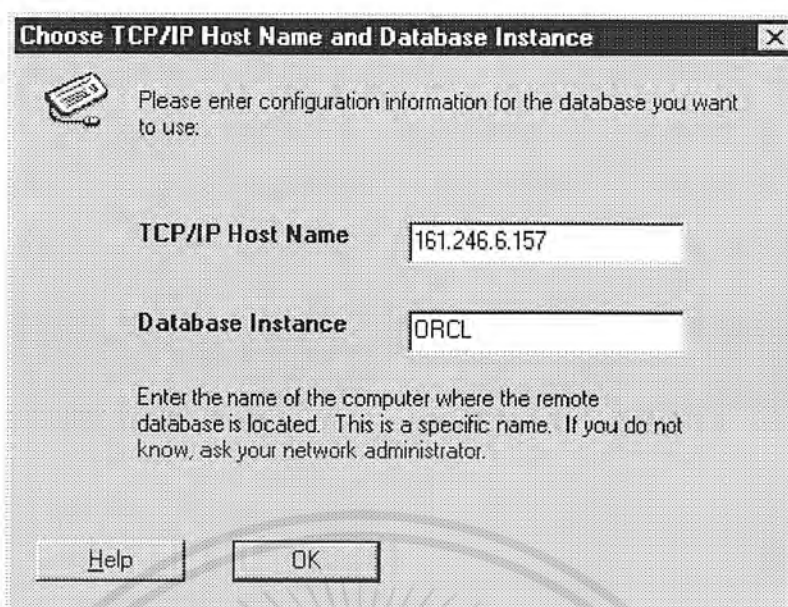


ขั้นที่ 3. ทำการเลือกโพรโตคอลที่ใช้สำหรับการติดต่อกับเซิร์ฟเวอร์ฐานข้อมูล แล้วคลิก OK



ขั้นที่ 4. กำหนดตำแหน่ง IP แอดเดรสของเซิร์ฟเวอร์ฐานข้อมูลที่ใช้งาน เช่น 161.246.6.157 เป็นต้น คลิก OK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

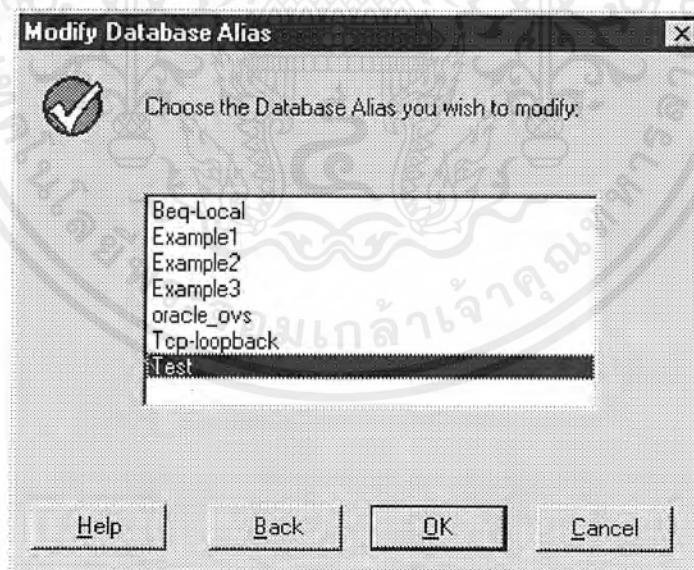


ขั้นที่ 5. คลิก “Yes” เพื่อทำการสร้าง Database Alias

3.2 แก้ไขข้อมูล Database Alias

การแก้ไขรายละเอียดของ Database Alias มีขั้นตอนดังนี้

ขั้นที่ 1. เลือก Database Alias ที่ต้องการแก้ไข แล้วคลิก OK



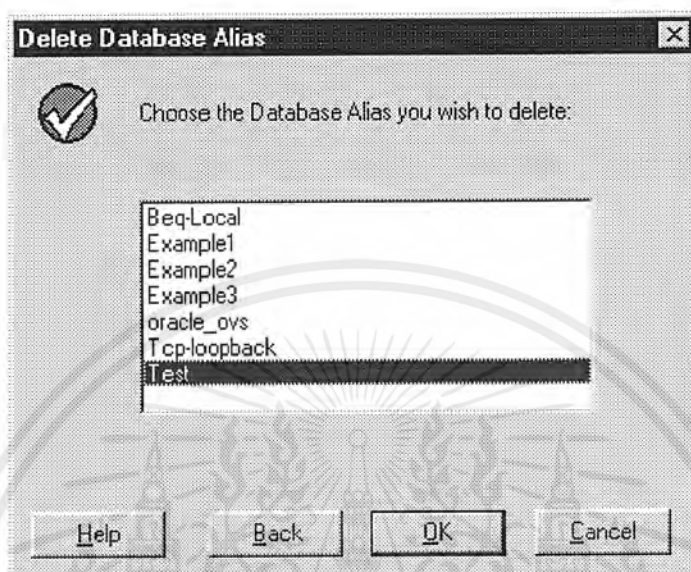
ขั้นที่ 2. จากนั้นขั้นตอนจะเหมือนกับขั้นที่ 3-5 ของการเพิ่ม Database Alias

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ลบ Database Alias

การลบ Database Alias มีขั้นตอนดังนี้

ขั้นที่ 1. เลือก Database Alias ที่จะลบออกจากระบบ



ขั้นที่ 2. คลิก "Yes" เพื่อทำการลบ

4. ไฟล์ configuration ของระบบ Oracle Video Server

4.1 ไฟล์ voltab

ไฟล์ voltab จะเป็นไฟล์ที่เก็บรายละเอียดต่างๆของวอลลุ่มทั้งหมดภายในระบบ

รูปแบบ *volume striped, [width=integer.] [noparity.] [raidsize=integer.] disks ...*

เมื่อ volume : ชื่อวอลลุ่ม

width : ขนาดความกว้างของสติปที่ใช้จะมีขนาดได้เท่ากับ 32KB และ 64KB เท่านั้น

noparity : กำหนดให้ไม่มีการทำข้อมูล parity

raidsize : จำนวนของดิสก์ใน RAID

disks... : กลุ่มของดิสก์ที่ใช้ในวอลลุ่ม

ตัวอย่างข้อมูลในไฟล์ voltab

video striped,width=32k,raidsize=1,noparity D:\ORANT\vs30\data\dsk00.dat

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตัวอย่างการใช้งานสามารถแยกรายละเอียดได้ตามนี้ คือ วอลลุ่มชื่อ = "video" โดยมีขนาดของสติปเท่ากับ 32KB มีติสท์อยู่ในวอลลุ่มเพียงติสท์เดียว คือ D:\ORANTVS30\DATA\DSK00.DAT และระบบดังกล่าวไม่มีการกู้ข้อมูลเมื่อมีความเสียหายกับวอลลุ่ม เพราะ "noparity"

4.2 ไฟล์ mnrc

ไฟล์ mnrc จะกำหนดรายละเอียด ข้อมูลต่างๆที่ใช้สำหรับการเชื่อมต่อเข้ากับเซิร์ฟเวอร์ฐานข้อมูล ข้อมูลภายใน mnrc ในโครงการนี้มีดังนี้

ตัวอย่างข้อมูลในไฟล์mnrc

```
ys.log.msg-path=D:\ORANT\MEDIANET\MESG
ys.log.msg-path=D:\ORANT\VS30\MESG
vscontsrv.connect=ovs/oraclevideosever@oracle_ovs
vsbcastsrv.connect=ovs/oraclevideosever@oracle_ovs
```

4.3 ไฟล์ Tnsnames.ora

ไฟล์ Tnsnames.ora จะเป็นไฟล์ที่ถูกสร้างโดยโปรแกรม SQL*NetEasy Configuration ที่ใช้งานของ SQL*Net โดยภายในจะแสดงรายละเอียดต่างๆของเซิร์ฟเวอร์ฐานข้อมูลที่ใช้ในระบบ

ตัวอย่างข้อมูลในไฟล์ Tnsnames.ora

```
oracle_ovs.world =
(DESCRIPTION =
(ADDRESS_LIST =
(ADDRESS =
(COMMUNITY = tcp.world)
(PROTOCOL = TCP)
(Host = 161.246.6.157)
(Port = 1521)
)
)
(ADDRESS =
(COMMUNITY = tcp.world)
(PROTOCOL = TCP)
(Host = 161.246.6.157)
(Port = 1526)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
)
)
(CONNECT_DATA = (SID = ORCL)
)
```

จากตัวอย่างจะมี connect_string ชื่อ "Oracle_ovs" โดยมีรูปแบบการเชื่อมต่อแบบ TCP โดยเครื่องเซิร์ฟเวอร์ฐานข้อมูลมีตำแหน่ง IP ที่ 161.246.6.157 เป็นต้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

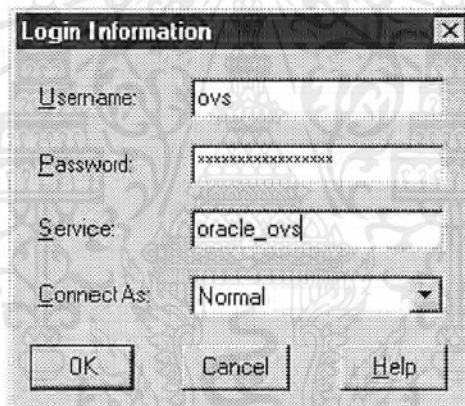
การใช้งาน Oracle Database

1. วิธีการสร้าง Account

การสร้าง Database_user สามารถทำได้โดยการใช้โปรแกรม “Security Manager” ซึ่งเป็นโปรแกรมหนึ่งในกลุ่ม “Oracle Enterprise Manager” โดยมีขั้นตอนดังนี้

ขั้นที่ 1. เรียกโปรแกรม Security Manager จากเมนู Start ของ Windows NT ในเครื่องดาตาเบส เซิร์ฟเวอร์

ขั้นที่ 2. ใส่ User_Name, รหัสผ่าน และ Connect_String โดย User_Name ที่ใช้จะต้องมีสิทธิ์ในการสร้าง Account ใหม่ได้ คลิก OK



ขั้นที่ 3. ทำการคลิกขวาที่ Connect_String\Users จะปรากฏเมนูย่อย ทำการเลือกที่ Create...



ขั้นที่ 4. กำหนดรายละเอียดของ Account ที่ต้องการ แล้วคลิก Create

กำหนดค่าให้ User_Name : ovs
 Password : oraclevideosever
 Tablespaces\Default : USER_DATA
 Tablespaces\Temporary : TEMPORARY_DATA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 5. ออกจากโปรแกรม Security Manager

2. ชุดคำสั่ง SQL

2.1 คำสั่ง CREATE

คำสั่ง CREATE ใช้สำหรับการสร้างตารางในฐานข้อมูล

รูปแบบ

```
CREATE TABLE table_name
(column_name data_type [NOT NULL], -
[column_name data_type [NOT NULL]]...)
```

เมื่อ table_name : ชื่อของตารางที่จะสร้าง
column_name : ชื่อของคอลัมน์ที่จะสร้าง
data_type : ชนิดของข้อมูลในคอลัมน์

ตัวอย่างการใช้งาน

```
CREATE TABLE preview (content VARCHAR, clips VARCHAR);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างชนิดของข้อมูล เช่น INTEGER เป็นชนิดข้อมูลตัวเลขจำนวนเต็มที่มีค่าตั้งแต่ -2,147,483,647 ถึง +2,147,483,647 และชนิด VARCHAR เป็นสตริงที่มีความยาว ตั้งแต่ 1 ถึง 254 เป็นต้น

2.2 คำสั่ง DROP

คำสั่ง DROP ใช้สำหรับการทำลายตารางที่เคยสร้างไว้

รูปแบบ

```
DROP TABLE table_name;
```

ตัวอย่างการใช้งาน

```
DROP TABLE preview;
```

2.3 คำสั่ง INSERT

คำสั่ง INSERT ใช้สำหรับการเพิ่ม ROW ข้อมูลให้กับตารางในฐานข้อมูล

รูปแบบ

```
INSERT INTO table_name [(column_name)]
VALUES (data_item);
```

เมื่อ data_item : ข้อมูลที่ต้องการเพิ่มเข้าในตาราง

ตัวอย่างการใช้งาน

```
INSERT INTO preview VALUES('ending', 'simple0');
```

2.4 คำสั่ง DELETE

คำสั่ง DELETE ใช้สำหรับลบ ROW ข้อมูลในตารางฐานข้อมูล

รูปแบบ

```
DELETE FROM table_name –
[WHERE search_condition];
```

ตัวอย่างการใช้งาน

```
DELETE FROM preview –
WHERE content = 'ending';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 คำสั่ง SELECT

คำสั่ง SELECT ใช้สำหรับการคิวรีฐานข้อมูลเพื่อนำข้อมูลต่างๆออกมาใช้งาน

รูปแบบ

```
SELECT column_name, or *
FROM table_name
[WHERE search_condition]
[GROUP BY column_list]
[HAVING group_condition]
[ORDER BY column_list]
```

ตัวอย่างการใช้งาน

```
SELECT * FROM preview;
```

หรือ

```
SELECT * FROM preview WHERE content = 'ending';
```

ภาคผนวก ก

คู่มือการติดตั้งและใช้งาน Oracle Video Server Manager

1. การติดตั้งโปรแกรม VSM มีขั้นตอนดังนี้

ก่อนการติดตั้งจะต้องพิจารณาทรัพยากรของเครื่องที่จะติดตั้งว่าตรงตามความต้องการ และมีปริมาณเพียงพอหรือไม่? โดยสิ่งที่จะต้องพิจารณา มีดังต่อไปนี้

- การ์ด VGA ควรมีความละเอียดอย่างน้อย 800x600จุด
- หน่วยความจำหลักมีขนาด 32MB
- เครื่องมีระบบปฏิบัติการเป็น Windows 95 พร้อม Service Pack 1 หรือ Windows NT พร้อม Service Pack 3
- ต้องติดตั้งโปรแกรม Java Runtime Environment (JRE) 1.1.5 ไว้แล้ว
- ต้องติดตั้ง Oracle Video Client ด้วย หากต้องการแสดงข้อมูลบนเครื่องที่จะติดตั้ง VSM
- ไดรฟ์ซีดีรอม
- มีพื้นที่ว่างในฮาร์ดดิสก์ที่จะติดตั้ง ตามนี้
 - Oracle Video Server มีขนาด 2.3 MB
 - คู่มืออิเล็กทรอนิกส์ของ Oracle Video Server มีขนาด 4MB
 - Oracle Video Client 3.0.3 มีขนาด 3.5 MB
 - Adobe Acrobat Reader 3.0 ขนาด 4 MB

1.1 วิธีการติดตั้งโปรแกรม VSM มีขั้นตอนดังนี้

ขั้นที่ 1. ทำการปิดโปรแกรมอื่นๆให้หมด เพื่อจะทำการติดตั้ง VSM

ขั้นที่ 2. ใส่แผ่นซีดีรอมของ VSM ในไดรฟ์

ขั้นที่ 3. เรียก Run... จากเมนูStart ของWindows NT

ขั้นที่ 4. ทำการเรียกโปรแกรมเซตอัพ โดยพิมพ์คำว่า "E:\SETUP.EXE" ซึ่งกำหนดให้ไดรฟ์ซีดีรอมอยู่ที่ไดรฟ์ E:

ขั้นที่ 5. Install Shield จะปรากฏขึ้นมาและจะแสดงข้อความ ,คลิก Next

ขั้นที่ 6. โปรแกรมติดตั้งจะทำการแสดงรายละเอียดของลิขสิทธิ์ อ่านจบแล้วให้คลิกNext

ขั้นที่ 7. ใส่รากไดเรกทอรีที่จะทำการติดตั้งโปรแกรม VSM โดยออรากเซลจะเรียกรากไดเรกทอรีนี้ว่า "ORACLE_HOME" เช่น D:\ORANT

ขั้นที่ 8. หากเครื่องยังไม่ได้ติดตั้ง Oracle Video Client 3.0.3, Adobe Acrobat Reader 3.0 และ JRE 1.1.5 หรือไม่ได้ติดตั้งตัวใดตัวหนึ่ง โปรแกรมติดตั้งจะเตือนว่าโปรแกรม VSMจำเป็นต้องเรียกใช้โปรแกรมเหล่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 9. ใส่ชื่อของเซิร์ฟเวอร์และชื่อโดเมนของOVS

ตัวอย่างเช่น

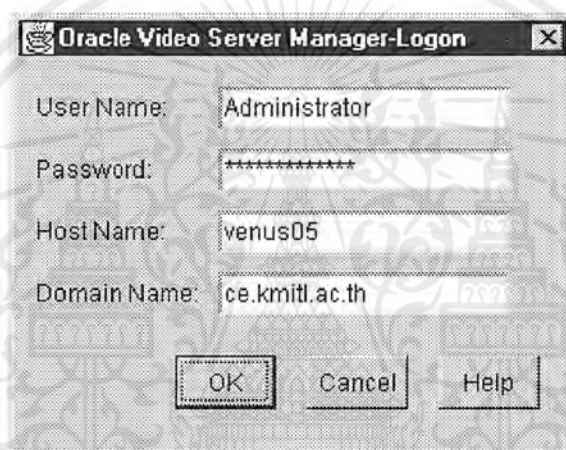
ชื่อเซิร์ฟเวอร์ : venus08

ชื่อโดเมน : ce.kmitl.ac.th

ขั้นที่ 10. เสร็จขั้นตอนการติดตั้ง

หมายเหตุ VSM จะทำการติดตั้งไฟล์ที่ชื่อว่า vsm.jar ซึ่งเวลาใช้งานจะต้องใช้ร่วมกับ JRE ในโครงการนี้จะใช้โปรแกรม VSM ในการทำตัวอย่างคอนเทนต์ (Preview)

2. การใช้งานโปรแกรม Oracle Video Server Manager



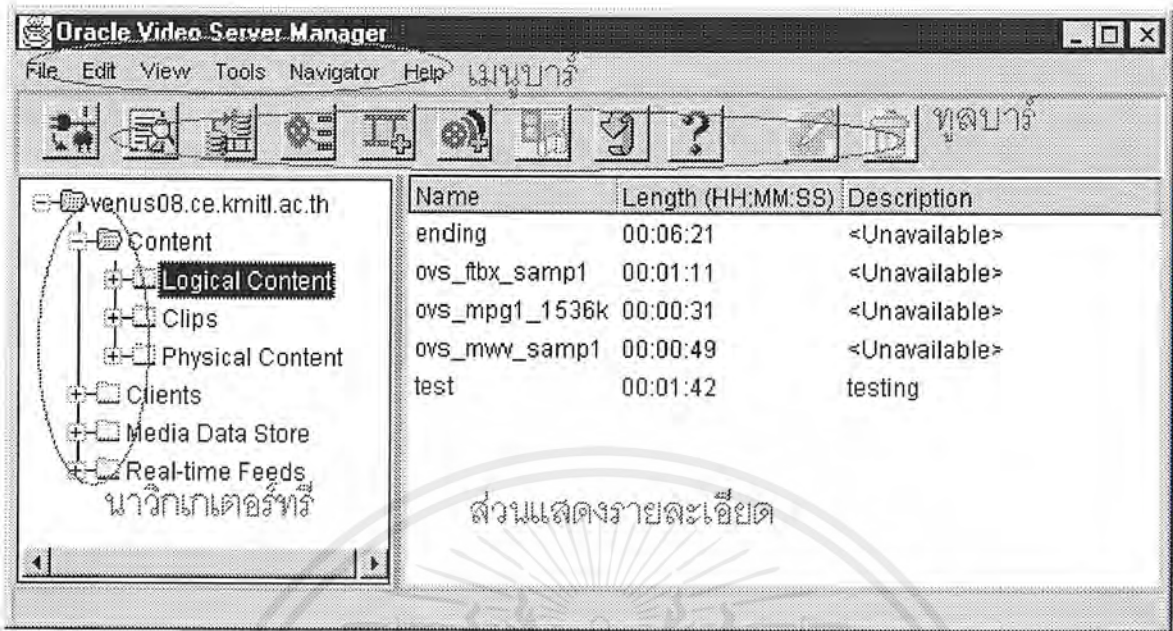
สิ่งแรกที่ต้องทำก่อนเข้าไปใช้งาน VSM คือการล็อกออน(Logon) การล็อกออนของ VSM นั้นจะอ้างอิงกับฐานข้อมูลรายชื่อผู้ใช้เครื่องของเครื่องวิดีโอเซิร์ฟเวอร์นั้นๆ เท่านั้น สิ่งหนึ่งที่ต้องคำนึงถึงในการล็อกออนคือโปรแกรม VSM เป็นโปรแกรมที่เคสเซนซิทีฟ(CASE-Sensitive)

โปรแกรม VSM นั้นใช้แนวทางกราฟฟิคอินเตอร์เฟส เพื่อให้ง่ายแก่การใช้งาน โดยหน้าต่างของโปรแกรม VSM นั้นจะประกอบไปด้วยหลายๆ ส่วน โดยเราจะแบ่งส่วนต่างๆ เหล่านี้ ได้ดังนี้ คือ

- **เมนูบาร์ (Menu bar)** จะประกอบไปด้วยคำสั่งต่างๆ ซึ่งจะกล่าวในส่วนถัดไป
- **ทูลบาร์ (Tool bar)** จะประกอบไปด้วยคำสั่งที่มักจะถูกใช้งานบ่อยๆ อยู่ในรูปของไอคอน
- **นาวิกเตอร์ทรี (Navigator tree)** จะแสดงให้เห็นถึงส่วนต่างๆ ของเซิร์ฟเวอร์ที่ใช้งานอยู่ โดยได้แบ่งส่วนต่างๆ ดังนี้
 1. คอนเทนต์
 2. ไคลเอนต์
 3. ดิสก์ MDS
 4. ส่วนที่ทำเรียลไทม์ฟีด(Real-time Feed)

หมายเหตุ ส่วนของเรียลไทม์ฟีดไม่ได้ถูกใช้ในโครงการนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



- ส่วนที่แสดงรายละเอียดปลีกย่อย (Detail pane)
- ส่วนที่แสดงสถานะการทำงาน (Status bar)

3. สรุปคำสั่งการทำงานของ VSM มีดังต่อไปนี้

3.1 กลุ่มคำสั่งทั่วไป มีดังต่อไปนี้

- คำสั่งกำหนดเซิร์ฟเวอร์ที่จะกระทำงาน (Switch Video Server)



อธิบาย: เป็นคำสั่งที่ใช้สำหรับการกำหนดชื่อเซิร์ฟเวอร์ที่ต้องการกระทำ กล่าวคือเป็นการเปลี่ยนการทำงานจากเซิร์ฟเวอร์หนึ่งไปยังอีกเซิร์ฟเวอร์หนึ่ง

- คำสั่งดูสถานะของงานที่ทำ (Job Status)



อธิบาย: ใช้สำหรับดูงานที่สั่งทำว่าเสร็จหรือยัง

- คำสั่งเล่นวิดีโอ (Play)



อธิบาย ใช้สำหรับการเล่นวิดีโอ ในกรณีการทำงานของลอคัลคอนเทนนต์

- คำสั่งรีเฟรชหน้าจอแสดงผล (Refresh display)



อธิบาย ใช้สำหรับการแสดงข้อมูลล่าสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 กลุ่มคำสั่งของการบริหารระบบ OVS

□ คำสั่งโหลดคอนเทนต์



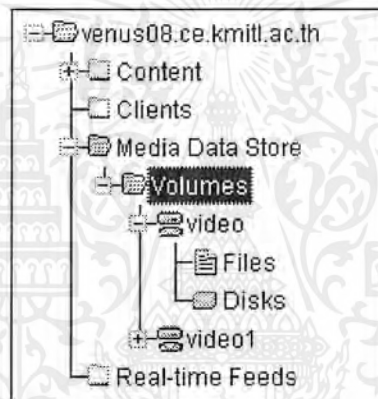
อธิบาย ใช้สำหรับการโหลดไฟล์มัลติมีเดียไปเก็บไว้ในดิสก์ MDS โดยโปรแกรม VSM จะทำการลงทะเบียนไฟล์มัลติมีเดียให้โดยอัตโนมัติ

□ คำสั่งแสดงรายละเอียดของวอลลุ่ม

อธิบาย ใช้แสดงรายละเอียดของวอลลุ่มว่าในเซิร์ฟเวอร์ประกอบไปด้วยวอลลุ่มอะไรบ้าง มีขนาดความจุเท่าไร และใช้ไปแล้วเท่าไร และยังสามารถดูคอนเทนต์ภายในดิสก์ MDS ต่างๆได้ด้วยว่าประกอบไปด้วยคอนเทนต์อะไรบ้าง เป็นต้น

ตำแหน่งเรียกใช้งาน เลือกที่ \Media Data Store\Volumes ในนาวิกเตอร์ทรี

อ้างอิง คำสั่ง mdmdir



□ คำสั่งลงทะเบียนคอนเทนต์กับMDS



อธิบาย เป็นคำสั่งที่ใช้ในการลงทะเบียนไฟล์ข้อมูลมัลติมีเดียให้กับMDS เพื่อMDSจะทำการสร้างไฟล์แท็กของคอนเทนต์นั้นๆขึ้น (ไฟล์ .mpi) ซึ่งทุกๆคอนเทนต์จะต้องทำการลงทะเบียนก่อนการใช้งานเสมอ

3.3 กลุ่มคำสั่งการทำลอจิกคอลลคอนเทนต์

□ คำสั่งการสร้างไฟล์คลิป



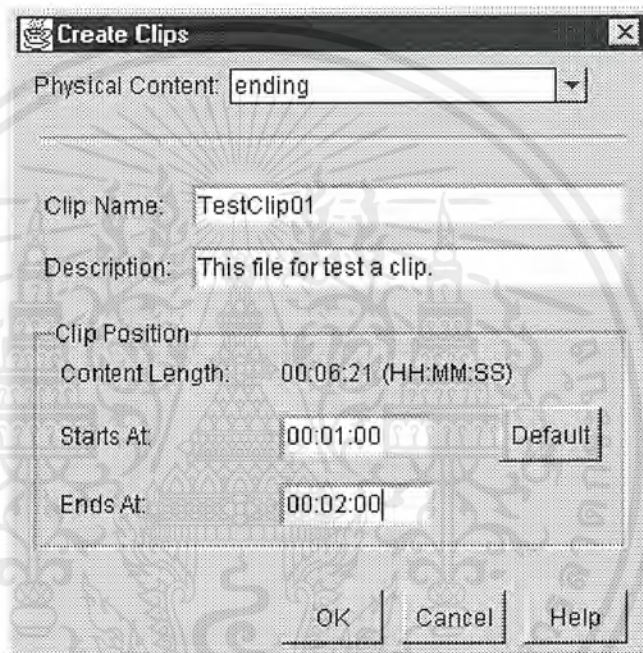
คำอธิบาย ใช้สำหรับการสร้างไฟล์คลิป โดยรายละเอียดในการสร้างไฟล์คลิปจะมีดังต่อไปนี้

- Physical Content: สำหรับกำหนดชื่อไฟล์ฟิสิกส์คอนเทนต์ที่จะนำมาทำคลิป โดยคลิปหนึ่งคลิปจะได้มาจากฟิสิกส์คอลลคอนเทนต์เดียวเท่านั้น
- Clip Name: สำหรับกำหนดชื่อของไฟล์คลิปที่จะสร้าง
- Description: สำหรับเพิ่มเติมคำอธิบายเพิ่มเติมเกี่ยวกับไฟล์คลิปนั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Clip Position: จะประกอบไปด้วย 2 ค่าคือ ตำแหน่งเริ่มต้นและตำแหน่งสุดท้ายของคลิป ในไฟล์ฟิสิกส์คอลคอนเทนต์ โดยค่าดังกล่าวจะต้องกำหนดเป็นหน่วยของวินาที

ตัวอย่างการสร้างไฟล์คลิป กำหนดให้มีฟิสิกส์คอลคอนเทนต์ชื่อ Ending โดยมีความยาวในการเล่นเท่ากับ 00:06:21 (ชั่วโมง:นาที:วินาที) เราจะทำการสร้างคลิปจากไฟล์Ending โดยจะเอาเพียงส่วนกลางของไฟล์คือ เริ่มที่เวลา 00:01:00 จนถึงเวลา 00:02:00 โดยจะตั้งชื่อไฟล์คลิปว่า TestClip01



□ คำสั่งการสร้างไฟล์ลอจิกคอลคอนเทนต์

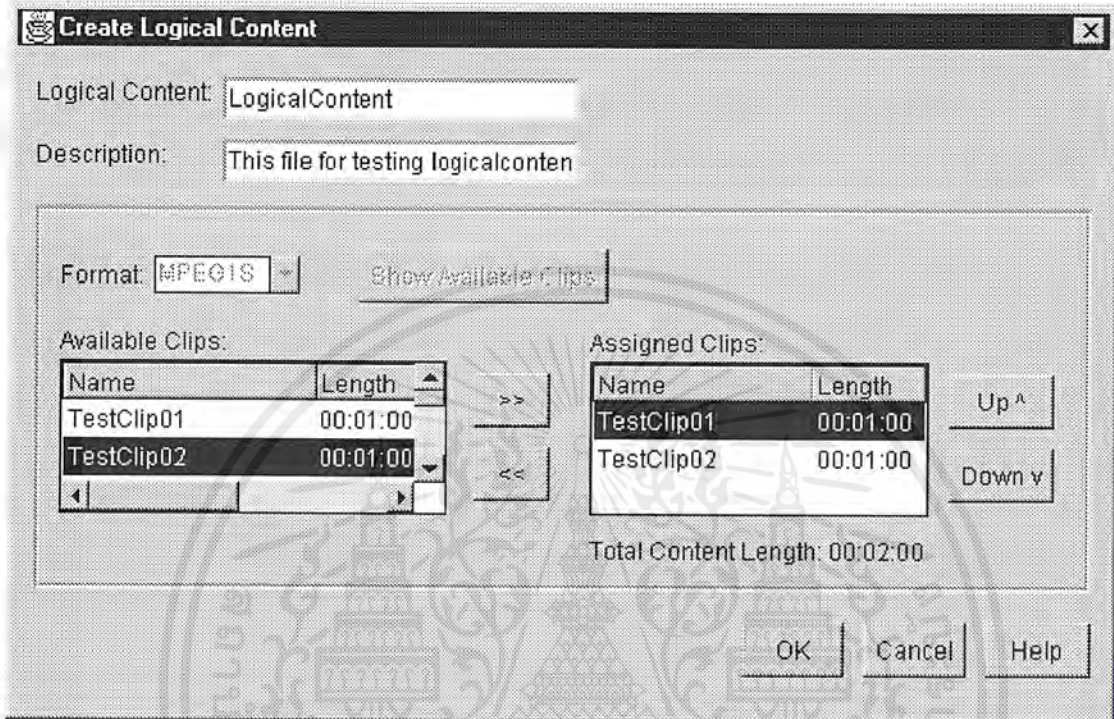


คำอธิบาย ใช้สำหรับการสร้างไฟล์ลอจิกคอลคอนเทนต์ โดยจะมีรายละเอียดดังต่อไปนี้

- Logical Content: สำหรับกำหนดชื่อของไฟล์ลอจิกคอลคอนเทนต์ที่จะสร้าง
- Description: สำหรับคำอธิบายเพิ่มเติมเกี่ยวกับไฟล์ลอจิกคอลคอนเทนต์ที่จะสร้าง
- Show Available Clips: เป็นปุ่มที่ใช้สำหรับแสดงไฟล์คลิปต่างๆที่จะสามารถนำมาใช้ในไฟล์ลอจิกคอลคอนเทนต์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการสร้างไฟล์ลอจิกคอลคอนเทนต์ กำหนดให้มีไฟล์คลิปชื่อ TestClip01 และ TestClip02 อยู่ และเราจะทำการไฟล์ลอจิกคอลคอนเทนต์ซึ่งประกอบด้วยไฟล์คลิปลงกล่าวอยู่ โดยให้ชื่อไฟล์ว่า LogicalContent ซึ่งจะมีความยาวเวลาเล่นเท่ากับ 2 นาที



□ คำสั่งการแก้ไขและเปลี่ยนชื่อ



คำอธิบาย ในการจะทำการแก้ไขรายละเอียดต่างๆของไฟล์คลิปและไฟล์ลอจิกคอลคอนเทนต์นั้น ขั้นตอนแรกจะต้องทำการคลิกเลือกที่ไฟล์ดังกล่าวเสียก่อน แล้วจึงทำการคลิกที่ไอคอนมอดิไฟ (Modify) เมื่อคำสั่งมอดิไฟทำงาน โปรแกรมVSM จะแสดงรายละเอียดต่างๆที่อยู่ภายในไฟล์คลิปและไฟล์ลอจิกคอลคอนเทนต์

□ คำสั่งลบคลิปหรือลอจิกคอลคอนเทนต์



คำอธิบาย การลบไฟล์คลิปหรือไฟล์ลอจิกคอลคอนเทนต์ก็จะคล้ายกับการทำการแก้ไข คือ จะต้องทำการเลือกไฟล์ที่ต้องการจะลบก่อน แล้วจึงทำการเลือกไอคอนลบ (Delete) เมื่อสั่งลบ โปรแกรมจะถามซ้ำอีกครั้งว่าต้องการลบไฟล์ดังกล่าวแน่ไหม ถ้าตอบใช่VSMจะทำการลบข้อมูลมัลติมีเดียออกจากคิสก์MDS

อ้างอิง คำสั่ง mdsdelete

หมายเหตุ ในภาคผนวกนี้จะขอกล่าวถึงเฉพาะคำสั่งที่ใช้ทำในโครงการรวมถึงคำสั่งบางคำสั่งที่คาดว่าจะได้ใช้งานเท่านั้น ส่วนคำสั่งส่วนที่ไม่ได้ใช้งานในโครงการนี้จะไม่ขอกล่าวถึง เพื่อมุ่งให้ผู้อ่านทำความเข้าใจเฉพาะส่วนที่จำเป็นต่อการศึกษาคำสั่งนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง

Oracle Video JAVA Library

ในการพัฒนาแอปพลิเคชันบนเครื่องมือในการพัฒนาแล้ว ช่วยอำนวยความสะดวกในเรื่องต่างๆ เป็นสิ่งจำเป็น เพื่อให้การพัฒนาสามารถทำได้ง่ายและรวดเร็ว การพัฒนาแอปพลิเคชันของ Oracle Video Client นั้น จะมีไลบรารีที่ช่วยในการพัฒนาคือ Oracle Video JAVA Library การติดตั้งและรูปแบบการเรียกใช้คลาสต่างๆ มีดังนี้

1. การติดตั้ง Oracle Video JAVA Library สำหรับ JDK

ไลบรารีนี้จะได้จากการติดตั้ง Oracle Video Server เรียบร้อยแล้ว และจะอยู่ในไฟล์สกุล JAR ชื่อ ovc (ovc.jar) โดยจะเก็บอยู่ในไดเรกทอรี `C:\orant\jbin` การติดตั้งจะต้องนำไฟล์นี้ไปเก็บไว้ในไดเรกทอรีของโปรแกรม JDK ที่เราจะใช้พัฒนาแอปพลิเคชัน โดยมีข้อกำหนดว่าเวอร์ชันของ JDK ที่จะใช้ต้องไม่ต่ำกว่า 1.1.X และนำไปใส่ในไดเรกทอรีดังต่อไปนี้ `c:\jdk1.1.X\lib\ovc.jar` จากนั้นก็ไปกำหนดค่าในไฟล์ Autoexec.bat เพื่อให้ JDK รู้จักและมองเห็นไฟล์ ovc.jar ดังนี้ `set classpath=c:\jdk1.1.X\lib\ovc.jar`;

2 กลุ่มคลาสของ Oracle Video JAVA Library

กลุ่มคลาสของ Oracle Video JAVA Library นั้นจะแบ่งออกได้เป็น 3 พวกดังนี้

2.1 กลุ่มคลาสที่เกี่ยวข้องกับตัวควบคุมภาพและเสียง (Player Classes)

จะประกอบด้วยคลาสต่างๆดังนี้

- คลาส Player จะมีฟังก์ชันที่ควบคุมการโหลดมีเดียไฟล์และการควบคุมการแสดงผลไฟล์เหล่านั้น
- คลาส PlayerFactory จะทำการสร้างออบเจกต์ Player ขึ้นมา
- คลาส PlayerListener ทำให้แอปพลิเคชันสามารถรับรู้ถึงเหตุการณ์ที่เกิดขึ้นกับออบเจกต์ Player ได้
- คลาส PlayerException คอยตรวจจับ Exception ที่เกิดขึ้นกับออบเจกต์ Player

2.1.1 คลาส Player

คลาส Player จะไม่มี Public Constructor การสร้างออบเจกต์ Player จะต้องทำการเรียกเมธอด `PlayerFactory.getPlayer()`

ค่าคงที่

ค่าคงที่ของคลาสนี้ทั้งหมดจะเป็นค่าสถานะของออบเจกต์ Player ซึ่งเป็นจำนวนอินทิเจอร์ มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Player.ST_EOS บอกถึงสถานะที่สตรีมถึงจุดจบแล้ว
- Player.ST_PAUSED บอกสถานะของออบเจกต์ Player ว่ากำลังหยุดชั่วคราว
- Player.ST_REALIZED บอกสถานะของออบเจกต์ Player ว่าทำการโหลดสตรีมเสร็จแล้ว
- Player.ST_ERROR บอกถึงสถานะผิดพลาดที่ไม่สามารถหาสาเหตุได้
- Player.ST_PLAYING บอกสถานะของออบเจกต์ Player ว่ากำลังทำการเล่นสตรีมอยู่
- Player.ST_UNINIT บอกสถานะของออบเจกต์ Player ว่าไม่ได้กำหนดค่าเริ่มต้น
- Player.ST_INIT บอกถึงสถานะของออบเจกต์ Player ว่าได้สร้างและกำหนดค่าเริ่มต้นแล้ว

เมธอด

แบ่งเมธอดออกได้เป็น 3 กลุ่มดังนี้

1. เมธอดที่ควบคุมการแสดงผลของ User Interface Component จะแบ่งออกเป็น 3 ส่วน คือ video screen, playback control และ status bar โดยสามารถรวมทั้งสาม User Interface Component นี้เป็นออบเจกต์เดียวหรือจะแยกแสดงผลตัวใดก็ได้มีดังนี้

- `getStatusComp()` ทำการดึงคอมโพเนนต์ที่แสดงสถานะเพื่อใส่ไปใน Player interface โดยคอมโพเนนต์ที่แสดงสถานะนี้จะแสดง ตำแหน่งสตรีมทั้งในแบบเฟรมและวินาที,สถานะของออบเจกต์ Player
Syntax: `Component getStatusComp()`
- `getVisualComp()` ทำการดึงจอภาพที่ใช้แสดงผลสตรีมมาใส่ในออบเจกต์ Player
Syntax: `Component getVisualComp()`
- `getControlComp()` ทำการดึงเอาคอมโพเนนต์มาตรฐานต่างๆ เช่น ปุ่ม play/pause, stop, scroll bar แสดงเวลา
Syntax: `Component getControlComp()`
- `getPlayerUI()` ดึง video screen (visual component), controller panel และ status line panel รวมกันเป็นหนึ่งออบเจกต์มาใส่ในส่วน Player Interface
Syntax: `Component getPlayerUI()`
Syntax: `Component getPlayerUI(boolean Video, boolean Controls, boolean Status)`
Syntax: `Component getPlayerUI(boolean Video, boolean Controls, boolean Status, int width, int height)`

เห็นได้ว่าเมธอดนี้มีคอนสตรัคเตอร์ 3 ตัว โดยตัวที่สองและสามนั้นหากเรากำหนดให้พารามิเตอร์ตัวไหนเป็น false ก็จะไม่มีการดึงเอาคอมโพเนนต์นั้นมาโดยพารามิเตอร์แต่ละตัวจะมีความหมายดังนี้

- Video เป็นชนิด Boolean กำหนดการดึง Video Screen
- Controls เป็นชนิด Boolean กำหนดการดึง Control Panel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Status เป็นชนิด Boolean กำหนดว่าต้องการจะดึง Status Bar
 - Width กำหนดความกว้างของจอแสดงผล
 - Height กำหนดความสูงของจอแสดงผล
- สามารถทำการอิมพลีเมนต์ที่ได้ดังนี้

```
public class Spud extends Frame {
    public static void main(String argv[]) {
        Component myUI = null;
        Spud spud = new Spud();
        try {
            Player m_player = PlayerFactory.getPlayer();
            myUI = m_player.getPlayerUI();
        }
        catch(PlayerException e) {
            System.out.println("Exception raised: " + e.toString());
        }
        spud.add(myUI);
        spud.show();
    }
}
```

2. เมธอดที่ทำการควบคุมภาพและเสียงเช่น Play, Pause, Start, Stop และ Resume รวมไปถึงการกำหนดตำแหน่งของมีเดีย มีดังนี้

- setFullScreen() เปลี่ยนเป็นโหมดแสดงผลแบบเต็มจอ

Syntax: void setFullScreen(boolean mode)

พารามิเตอร์ mode มีชนิดเป็นบูลีน โดยหากกำหนดเป็น true จะกำหนดโหมดแสดงผลแบบเต็มจอ หากเป็น False จะกำหนดให้เป็นโหมดแสดงผลแบบปกติคือเท่ากับขนาดที่กำหนดให้ตัวแปร Width, Height ตอนที่ทำการสร้างออบเจกต์ Player

- setPos() เปลี่ยนตำแหน่งของสตรีมไปยังจุดที่ออบเจกต์ StmPos กำหนด

Syntax: void setPos(StmPos pos)

พารามิเตอร์ pos กำหนดตำแหน่งของสตรีมที่ต้องการจะเล่น โดยกำหนดในออบเจกต์

Stmpos

- getVol() หาค่าความดัง

Syntax: void getVol(int vol)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พารามิเตอร์ `vol` มีชนิดเป็นจำนวนอินทิเจอร์โดยเก็บค่าความดัง กำหนดค่าความดังตั้งแต่ 0-100

- `setVol()` กำหนดค่าความดัง

Syntax: void setVol(int vol)

พารามิเตอร์ `vol` มีชนิดเป็นจำนวนอินทิเจอร์โดยกำหนดค่าความดังตั้งแต่ 0-100

- `load()` ทำการ โหลดไฟล์มัลติมีเดียจากเซิร์ฟเวอร์หรือจากดิสก์บนเครื่องไคลเอ็นท์

Syntax: void load(String mediafile, StmOpts opts)

พารามิเตอร์ `mediafile` มีชนิดเป็นสตริงเก็บที่อยู่ของไฟล์มัลติมีเดีย และ `opts` มีชนิดเป็นคลาส `StmOpts` กำหนดลักษณะของสตรีมที่จะเล่นเช่น `Auto Start` , `Auto Loop`

สามารถแสดงทำการอิมพลิเมนต์ที่ได้ดังนี้

```
public class Spud extends Frame {
    public static void main(String argv[]) {
        Spud app = new Spud("Load a file");
        Player player = null;
        try {
            player = PlayerFactory.getPlayer();
            app.add(player.getPlayerUI());
            app.show();
            player.load("/mds/video/intro.mpi", null);
        }
        catch (PlayerException e) {
            System.out.println("Exception: " + e.m_msg);
        }
    }
    public Spud(String title) {
        super(title);
    }
}
```

- `getPos()` หาค่าตำแหน่งปัจจุบันของสตรีม

Syntax: StmPos getPos(int fmt)

พารามิเตอร์ `fmt` มีชนิดเป็นจำนวนอินทิเจอร์โดยตรงกับค่าคงที่ของออบเจกต์ `StmPos` คือ `StmPos.POSFMT_TIME` และ `StmPos.POSFMT_FRAMES`

- `pause()` หยุดการเล่นสตรีม ณ ตำแหน่งปัจจุบัน
Syntax: void pause()
 - `stop()` หยุดการเล่นสตรีมและกำหนดตำแหน่งของสตรีม ไปยังจุดเริ่มต้น
Syntax: void stop()
 - `getSelRange()` หาค่าการเล่นที่จุดเริ่มต้นและจุดจบของออบเจกต์ `Player`
Syntax: void getSelRange(StmPos from, StmPos to)
พารามิเตอร์ `from` จะเก็บค่าของจุดเริ่มต้นที่ได้ และ `to` จะเก็บค่าของจุดสิ้นสุดที่ได้
 - `play()` ทำการเล่นสตรีม
Syntax: void play()
Syntax: void play(StmPos from, StmPos to)
พารามิเตอร์ `from` มีชนิดเป็นคลาส `StmPos` กำหนดตำแหน่งเริ่มต้นที่จะทำการเล่นสตรีม `to` มีชนิดเป็นคลาส `StmPos` กำหนดตำแหน่งสุดท้ายที่จะทำการเล่นสตรีม
 - `resume()` ทำการเล่นสตรีมจากสถานะ `Player.ST_PAUSED`
Syntax: void resume()
 - `unload()` ปลดทรัพยากรทั้งหมดที่ออบเจกต์ `Player` ได้ทำการ โหลดไว้
Syntax: void unload()
3. เมธอดที่ให้ฟังก์ชัน เกี่ยวกับออบเจกต์ `Player` เช่น ทำลายออบเจกต์ `Player`, ตรวจสอบสถานะของออบเจกต์ และการลงทะเบียน `Listener` ซึ่งจะทำให้สามารถจัดการกับเหตุการณ์ที่เกิดขึ้นกับออบเจกต์ `Player` ได้ มีดังนี้
- `stateToString()` เปลี่ยนค่าสถานะของออบเจกต์ `Player` เป็นคำอธิบาย
Syntax: String stateToString(int state)
พารามิเตอร์ `state` มีชนิดเป็นจำนวนอินทิจอร์ ซึ่งตรงกับค่าคงที่ของสถานะออบเจกต์ `Player`
 - `addListener()` ทำการริจิสเตอร์ `PlayerListener` ออบเจกต์ซึ่งเป็นออบเจกต์ที่คอยตรวจจับเหตุการณ์ที่เกิดขึ้นกับออบเจกต์ `Player`
Syntax: void addListener(PlayerListener listener)
 - `term()` เป็นการ จบการทำงานของ Oracle Video Client
Syntax: int term()
 - `getState()` หาค่าสถานะปัจจุบันของออบเจกต์ `Player` โดยจะผ่านค่าคงที่สถานะของออบเจกต์ `Player` กลับมา
Syntax: int getState()
 - `getStats()` หาค่าสถิติต่างๆที่เกี่ยวข้องกับการเล่น โดยจะต้องมีตัวแปรชนิด `StmStats` เพื่อรับค่าที่ผ่านกลับมา
Syntax: StmStats getStats()

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- `getInfo()` หาค่าข้อมูลเกี่ยวกับไฟล์มัลติมีเดียที่กำลังทำการเล่นอยู่ โดยจะต้องมีตัวแปรชนิด `StmInfo` เพื่อรับค่าที่ผ่านกลับมา
Syntax: StmInfo getInfo()

2.1.2 คลาส `PlayerFactory`

เป็นคลาสที่ทำการสร้างออบเจกต์ `Player`

เมธอด

- `createPlayer()` สร้างออบเจกต์ `Player`
Syntax: Player createPlayer()
- `getPlayer()` ให้คืนออบเจกต์ `Player` ที่สร้างขึ้นมาใช้งาน
Syntax: static synchronized Player getPlayer()

การสร้างออบเจกต์ `Player` สามารถทำซึ่งเป็นสแตติกเมธอดโดยไม่ต้องสร้าง Instance ของคลาสนี้เลยคือได้อีกวิธีโดยการเรียกเมธอด `PlayerFactory.getPlayer()` ซึ่งจะส่ง ออบเจกต์ `Player` กลับมาเช่นเดียวกัน

ขั้นตอนการสร้างออบเจกต์ `Player` มีดังนี้

```
static Player player = null;
player = PlayerFactory.getPlayer();
```

2.1.3 `layerListener`

จะจัดการกับเหตุการณ์ที่เกิดขึ้นกับออบเจกต์ `Player` การที่จัดการกับเหตุการณ์เหล่านี้ต้องทำการสร้างคลาสที่อิมพลีเมนต์ `PlayerListener` ขึ้นมา

ขั้นตอนการอิมพลีเมนต์ `PlayerListener` มีดังนี้

```
public class myApp implements PlayerListener
{
    //class myApp จะ implement มาจาก PlayerListener
    public void stateChange(int newState) {
        System.out.println("newState: " + newState);
    }
    public void error(int code, String msg){
        System.out.println("OVC- " + code + ": " + msg);
    }
    public void endOfStream() {
        System.out.println("end of stream reached");
    }
}
```

การลงทะเบียนที่อิมพลีเมนต์ `Listener` ทำได้โดยการเรียกเมธอด `Player.addListener()` และทำการผ่านค่าของออบเจกต์ที่อิมพลีเมนต์ `PlayerListener` เข้าไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4 คลาส PlayerException

จะทำการ throw exception จากการเรียกเมธอดของออบเจกต์ Player เงื่อนไขที่ทำให้เกิด Exception มีดังนี้

- การร้องขอต่างของ Player ไม่ถูกต้อง
- ออบเจกต์ Player อยู่ในสถานะที่ไม่ถูกต้องขณะเกิดเหตุการณ์อย่างใดอย่างหนึ่ง
- การผ่านค่าพารามิเตอร์ที่ผิด
- การเกิดข้อผิดพลาดภายในระบบ Oracle Video Client เอง
- ข้อผิดพลาดอื่นๆที่ยังไม่มีการจับ Exception

ค่าคงที่

ค่าคงที่ทั้งหมดจะมีชนิดเป็นจำนวนอินทิเจอร์ ดังนี้

- PlayerException.EX_BADPARAM บอกว่าค่า Parameter ที่ใช้เรียกเมธอดของออบเจกต์ Player นั้นผิด
- PlayerException.EX_BADSTATE บอกว่าเมธอดที่เรียกนั้น เรียกในขณะที่สถานะของออบเจกต์ Player ไม่ถูกต้อง
- PlayerException.EX_ERROR บอกข้อผิดพลาดอื่นๆที่นอกเหนือจากค่าคงที่เหล่านี้กำหนด
- PlayerException.EX_INTERNAL เป็นข้อผิดพลาดของ Oracle Video Client เอง
- PlayerException.EX_NOTIMPL บอกว่าเมธอดที่ทำการเรียกไม่มีในเวอร์ชันปัจจุบันของ Oracle Video JAVA Library

ตัวแปรสมาชิก

- m_type มีชนิดเป็นจำนวนอินทิเจอร์ บอกถึงข้อผิดพลาดที่ Thrown จากการเรียกเมธอดต่างๆในออบเจกต์ Player ซึ่งก็คือค่า Constants m กล่าวมาแล้วนั่นเอง
- m_code มีชนิดเป็นจำนวนอินทิเจอร์ รหัสข้อผิดพลาดของที่ออราเคิลกำหนดขึ้นมา
- m_msg มีชนิดเป็นจำนวนอินทิเจอร์ รหัสของสตริงที่เป็นข้อมูลของข้อผิดพลาดที่เกิดขึ้น

เมธอด

- toString()

Syntax: String toString()

ทำการหาข้อมูลในรูปสตริงที่อยู่ในตัวแปรสมาชิกต่างๆ

2.2 กลุ่มคลาสที่เกี่ยวข้องกับข้อมูลของภาพและเสียง (Stream Information Classes)

ข้อมูลเกี่ยวกับสตรีมจะอยู่ในคลาสดังนี้

2.2.1 คลาส StmPos

เป็นคลาสที่มีข้อมูลตำแหน่งของสตรีมในหลายรูปแบบ ซึ่งทำการกำหนดตำแหน่งได้ โดยเรียกเมธอด `Player.setPos()`

ค่าคงที่

ค่าคงที่ทุกตัวของคลาสนี้ทุกตัวจะมีชนิดเป็นจำนวนอินทิเจอร์มีดังนี้

- `StmPos.POSFMT_BEGINNING` บอกว่าตำแหน่งของสตรีมที่ต้องการเล่น จะอยู่ที่จุดเริ่มต้น
- `StmPos.POSFMT_CURRENT` บอกว่าตำแหน่งของสตรีมที่ต้องการเล่น จะอยู่ที่ตำแหน่งปัจจุบัน
- `StmPos.POSFMT_DEFAULT` บอกว่าตำแหน่งของสตรีมที่ต้องการเล่น จะอยู่ที่ตำแหน่งมาตรฐานคือที่จุดเริ่มต้น
- `StmPos.POSFMT_END` บอกว่าตำแหน่งของสตรีมที่ต้องการเล่น จะอยู่ที่จุดสิ้นสุด
- `StmPos.POSFMT_FRAMES` บอกว่าตำแหน่งของสตรีมที่ต้องการเล่น จะคิดเป็นจำนวนเฟรมจากจุดเริ่มต้น
- `StmPos.POSFMT_TIME` บอกว่าตำแหน่งของสตรีมที่ต้องการเล่น จะคิดเป็นหน่วยมิลลิวินาทีจากจุดเริ่มต้น

ตัวแปรสมาชิก

- `m_fmt` มีชนิดเป็นจำนวนอินทิเจอร์ กำหนดรูปแบบของสตรีมที่จะเล่นว่าเป็นเวลาหรือเฟรม
- `m_val` มีชนิดเป็นจำนวนคอง เป็นการกำหนดเวลาเริ่มต้นของสตรีมที่จะเล่น

เมธอด

- `StmPos()` เมธอดนี้เป็นคอนสตรัคเตอร์ของคลาสมี 2 รูปแบบ

Syntax: StmPos(int fmt)

Syntax: StmPos(int fmt, long val)

โดยสองคอนสตรัคเตอร์นี้จะต่างกัน คือ คอนสตรัคเตอร์ตัวแรกมีเพียงพารามิเตอร์เดียวคือ `m_fmt` ซึ่งบอกเพียงชนิดของสตรีมที่จะเล่นว่าจะกำหนดเป็นเฟรมหรือเป็นเวลา คอนสตรัคเตอร์ตัวที่สองจะมีการกำหนดตำแหน่งของสตรีมที่จะทำการเล่นด้วย

- `toString()`

Syntax: String toString()

ทำการหาข้อมูลในรูปสตริงที่อยู่ในตัวแปรสมาชิกต่างๆ

2.2.2 คลาส StmInfo

เป็นคลาสที่แสดงข้อมูลของไฟล์มัลติมีเดียที่ทำการเล่นอยู่ เช่น ชื่อ, transport protocol, Frame Rate
ค่าคงที่

ค่าคงที่ทุกตัวของคลาสนี้ทุกตัวจะมีชนิดเป็นจำนวนอินทีเจอร์มีดังนี้

- StmInfo.CSTAT_DISK บอกว่าไฟล์มัลติมีเดียที่ทำการเล่นอยู่นั้นดึงมาจากดิสก์บน Oracle Video Server
- StmInfo.CSTAT_FEED บอกว่าไฟล์มัลติมีเดียที่ทำการเล่นอยู่มีการเข้ารหัสไว้ 1 ครั้ง
- StmInfo.CSTAT_LOCALFILE บอกว่าไฟล์มัลติมีเดียที่ทำการเล่นอยู่ดึงมาจากดิสก์บนเครื่องเดียวกัน
- StmInfo.CSTAT_TAPE บอกว่าไฟล์มัลติมีเดียที่ทำการเล่นอยู่ดึงมาจาก Hierarchical Storage Manager บน Oracle Video Server
- StmInfo.CSTAT_TERMINATED บอกว่าไฟล์มัลติมีเดียที่ทำการเล่นอยู่ได้ถึงจุดสุดท้ายแล้ว
- StmInfo.CSTAT_UNKNOWN บอกว่าไฟล์มัลติมีเดียที่ทำการเล่นอยู่มีสถานะนอกเหนือจากที่กล่าวมา

ตัวแปรสมาชิก

- m_asset มีชนิดเป็นสตริงเป็นชื่อของ Logical Content asset Cookies
- m_bitrate มีชนิดเป็นจำนวนอินทีเจอร์ อัตราการส่งข้อมูลเป็นบิต ซึ่งประกอบด้วย ภาพ เสียงและ โอเวอร์เฮดอื่นๆ
- m_bytes มีชนิดเป็นจำนวนลอง ขนาดของไฟล์มัลติมีเดียในหน่วยที่เป็นไบต์
- m_contStat มีชนิดเป็นจำนวนอินทีเจอร์ เป็นจำนวนอินทีเจอร์ที่ตรงกับค่าคงที่ที่กล่าวมาแล้วซึ่งสามารถแปลงค่าเหล่านี้เป็นคำอธิบายได้โดยเรียกเมธอด conStatToString()
- m_desc มีชนิดเป็นสตริง คำอธิบายของสตรีม
- m_fps มีชนิดเป็นจำนวนอินทีเจอร์ อัตราการส่งข้อมูลเป็นเฟรม * 1000 ต่อวินาที
- m_name มีชนิดเป็นสตริง ชื่อของสตรีม
- m_proto มีชนิดเป็นสตริง Transport Protocol ของสตรีม
- m_url มีชนิดเป็นสตริง เป็นที่อยู่ของไฟล์มัลติมีเดียเพื่อใช้ในการ โหลด
- m_msecs มีชนิดเป็นจำนวนลอง ขนาดของไฟล์มัลติมีเดียในรูปของเวลามีหน่วยเป็นมิลลิวินาที
- เมธอด
- toString()

Syntax: String toString()

ทำการหาข้อมูลในรูปสตริงที่อยู่ในตัวแปรสมาชิกต่างๆ

2.2.3 คลาส StmStats

เป็นคลาสที่ให้ข้อมูลเกี่ยวกับสถิติของสตรีมที่กำลังทำการเล่นอยู่ในขณะนั้นซึ่งสถิติเหล่านี้จะผ่านกลับมาในรูปแบบของออบเจกต์เมื่อเรียกเมธอด `Player.getStats()`

ค่าคงที่

ค่าคงที่เหล่านี้จะใช้แทนสถิติการใช้นิตเวิร์ค ซึ่งจะไม่ใช่สถานะที่ได้จากการเรียกเมธอด `Player.getStats()` มีดังนี้

- `StmStats.STM_CONTROL` บอกว่าสตรีมอยู่ระหว่างการทำให้ Network Transaction
- `StmStats.STM_ENDED` บอกว่าสตรีมได้ถึงจุดสุดท้ายแล้ว
- `StmStats.STM_IDLE` บอกว่าสตรีมอยู่ในสถานะ Idle
- `StmStats.STM_PAUSED` บอกว่าสตรีมหยุดชั่วขณะ
- `StmStats.STM_PLAYING` บอกว่าสตรีมกำลังทำการเล่นอยู่
- `StmStats.STM_STALLED` บอกว่าสตรีมอยู่ในสถานะ Stall

ตัวแปรสมาชิก

- `m_bps` มีชนิดเป็นจำนวนอินทิจอร์ บอกอัตราเฉลี่ยของการส่งเป็น บิตต่อวินาที
- `m_cnsState` มีชนิดเป็นจำนวนอินทิจอร์
- `m_curFrame` มีชนิดเป็นจำนวนลอง บอกเฟรมปัจจุบันที่ปรากฏบนหน้าจอแสดงผล ซึ่งหากเริ่มต้นจำนวนเฟรมจะเป็น 0 และจะเพิ่มไปเรื่อยๆ เท่ากับจำนวนเฟรมที่ทำการแสดง
- `m_curTime` มีชนิดเป็นจำนวนลอง บอกตำแหน่งของสตรีมปัจจุบันในรูปแบบเวลาคือ มิลลิเซ็คันด์
- `m_drops` มีชนิดเป็นจำนวนอินทิจอร์ บอกจำนวนแพ็กเก็ตที่หายไปตั้งแต่เริ่มทำการเล่น
- สตรีม
- `m_fBytes` มีชนิดเป็นจำนวนอินทิจอร์ บอกจำนวน ไบต์ที่เหลือใน cache
- `m_fps` มีชนิดเป็นจำนวนอินทิจอร์ บอกอัตราเฉลี่ยของการส่งเป็น เฟรมต่อวินาที
- `m_maxTime` มีชนิดเป็นจำนวนลอง บอกเวลาในการหาสตรีมครั้งล่าสุดในหน่วยเวลา คือ มิลลิเซ็คันด์
- `m_minTime` มีชนิดเป็นจำนวนลอง บอกเวลาในการหาสตรีมครั้งแรกในหน่วยเวลา คือ มิลลิเซ็คันด์
- `m_pkts` มีชนิดเป็นจำนวนอินทิจอร์ บอกจำนวนแพ็กเก็ตที่ได้รับตั้งแต่การเล่นสตรีม
- `m_rBytes` `m_prdState` มีชนิดเป็นจำนวนอินทิจอร์ บอกจำนวนของ ไบต์ของ cache
- บนฝั่ง ไคลเอนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมธอด

- `stateToString()`
Syntax: static String stateToString(int state)
 ทำการเปลี่ยนสถานะของสตรีมเป็นคำอธิบายในรูปสตริง
- `toString()`
Syntax: String toString()
 ทำการหาข้อมูลในรูปสตริงที่อยู่ในตัวแปรสมาชิกต่างๆ

2.2.4 คลาส `StmOpts`

เป็นคลาสที่ประกอบด้วยตัวแปรต่างๆที่ใช้กำหนดลักษณะของการควบคุมสตรีมที่กำลังเล่นโดยทำการออบเจกต์ของคลาสนี้ไปตอนทำการเรียก `Player.load()`

ค่าคงที่

มีชนิดเป็นจำนวนอินทิเจอร์ดังนี้

- `StmOpts.DEFAULT_VOL` เป็นระดับเสียงของออบเจกต์ `Player` ซึ่งเป็นค่าปกติที่กำหนดมาแล้ว เมื่อกำหนดค่าในตัวแปร `StmOpts.m_volume` ขณะทำการเรียก `Player.Load()` จะทำให้ไม่สามารถเปลี่ยนแปลงค่าความดังได้อีก

ตัวแปรสมาชิก

- `m_autoStart` มีชนิดเป็นบูลีน เป็นกำหนดให้ทำการเล่นหลังจากโหลดแบบอัตโนมัติ หากเป็น `true` ออบเจกต์ `Player` จะทำการเรียกเมธอด `play()` โดยอัตโนมัติ แต่ถ้าเป็น `false` ผู้ใช้ต้องทำ `play` เอง
- `m_img` มีชนิดเป็นสตริง กำหนดชื่อไฟล์ที่เป็นรูปที่จะแสดงขณะจอแสดงผลไม่ได้ทำการแสดงสตรีมใดๆ
- `m_leftClick` มีชนิดเป็นบูลีน กำหนดให้มีการกดปุ่มเมาส์ซ้ายบนจอแสดงผลเพื่อทำการควบคุมการเล่นคือการ `play` และ `pause`
- `m_loop` มีชนิดเป็นบูลีน กำหนดให้การเล่นเป็นแบบวนเมื่อถึงจุดสิ้นสุดของสตรีม และกลับไปเริ่มเล่นที่จุดเริ่มต้นของสตรีมใหม่
- `m_playFrom` มีชนิดเป็น `StmPos` กำหนดตำแหน่งสตรีมเริ่มต้นของการเล่น
- `m_playTo` มีชนิดเป็น `StmPos` กำหนดตำแหน่งสตรีมสุดท้ายของการเล่น
- `m_popup` มีชนิดเป็นบูลีน กำหนดให้เมนูแสดงหากมีการกดเมาส์ปุ่มขวาบริเวณจอแสดงผล โดยเมนูที่แสดงจะมี คำสั่งการทำงานพื้นฐานเช่น `Play/Pause`, `Stop`, `rewind to beginning` เป็นต้น
- `m_volume` มีชนิดเป็นจำนวนอินทิเจอร์ เป็นการกำหนดค่าระดับเสียงของออบเจกต์ `Player` ตั้งแต่ 1- 100

เมธอด

- `StmOpts()` เป็นคอนสตรัคเตอร์ของคลาสนี้ เพื่อการกำหนดค่าในตัวแปรสมาชิกให้เป็นค่ามาตรฐาน

Syntax: StmOpts()

2.3. กลุ่มคลาสที่เกี่ยวข้องกับการเข้าถึงข้อมูล (Content Query Classes)

เป็นกลุ่มของคลาสที่ให้การควิรีไฟล์ในวิดีโอเซิร์ฟเวอร์ประกอบด้วยคลาสคือ

1. คลาส `Content` คลาสที่ทำการควิรีครั้งเดียว
2. คลาส `ContentIter` คลาสที่ทำการควิรี ซ้ำหลายครั้ง
3. คลาส `ContentException` เป็นคลาสที่ตรวจสอบ `Exception` ขณะทำการควิรีข้อมูล

2.3.1 คลาส Content

คลาสนี้จะให้ความสามารถในการควิรีรายชื่อไฟล์มัลติมีเดียที่เก็บในไดเรกทอรี `MDS (Media Data Store)` บนเซิร์ฟเวอร์โดยคลาสนี้จะไม่มีค่าคงที่และตัวแปรสมาชิก มีเพียงเมธอดเพียงเมธอดเดียว

เมธอด

- `query()` เป็นสแตติกเมธอดไม่ต้องทำการสร้างอินสแตนซ์

syntax `static StmInfo[] query(String srvAddr, String spec, ContentIter iter)`

พารามิเตอร์ `srvAddr` จะเป็นการกำหนดเลขที่อยู่ของเซิร์ฟเวอร์, `spec` เป็นการกำหนดไฟล์ที่ต้องการสามารถกำหนดเป็น wildcard ได้ และพารามิเตอร์ตัวสุดท้ายคือออบเจกต์ `ContentIter` โดยเมธอดนี้จะผ่านค่าอาร์เรย์ของ `StmInfo` ออบเจกต์กลับมา ขั้นตอนการควิรีสามารถอิมพลิเมนต์ได้ดังนี้

```
int i;
ContentIter iter = ContentIter(0, 10);
StmInfo[] contentList;
try {
while (iter.m_pos!=-1) {
contentList = Content.query(null, "*", iter);
for(i=0; i<iter.m_num;i++) {
System.out.println("Name: " + contentList[i].m_name);
System.out.println("Description: " + contentList[i].m_desc);
}
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

catch(ContentException e) {
    System.out.println("ContentException raised: " + e.toString());
}

```

2.3.2 คลาส ContentIter

ทำงานร่วมกับคลาส Content ในการดึงรายชื่อของไฟล์มัลติมีเดียจากเซิร์ฟเวอร์

ตัวแปรสมาชิก

- m_num มีชนิดเป็นจำนวนอินทิเจอร์ บอกจำนวนของชื่อไฟล์มัลติมีเดียที่จะดึง ในการคิวรีแต่ละครั้ง
- m_pos มีชนิดเป็นจำนวนอินทิเจอร์ บอกตำแหน่งของไฟล์มัลติมีเดียในขณะที่ทำการดึงค่าจะเป็น -1 หากได้ดึงมาครบแล้ว

เมธอด

- ContentIter()

Syntax: ContentIter(int pos, int num)

พารามิเตอร์ pos จะกำหนดตำแหน่งของรายการที่ต้องการจะดึงเมื่อเริ่มทำการคิวรี ส่วนพารามิเตอร์ num จำนวนไฟล์มัลติมีเดียที่ต้องการในการคิวรีแต่ละครั้ง

2.3.3 คลาส ContentException

Throw Exception ที่เกิดขึ้นขณะทำการคิวรีชื่อไฟล์มัลติมีเดียจาก Server

ค่าคงที่

ค่าคงที่ทุกตัวจะเป็นจำนวนอินทิเจอร์มีดังนี้

- ContentException.EX_BADPARAM ค่าพารามิเตอร์ที่ส่งในเมธอดคิวรีผิด
- ContentException.EX_INTERNAL เป็นข้อผิดพลาดที่เกิดภายในตัวคลาสเอง
- ContentException.EX_BADSTATE เมธอดคิวรีโดนเรียกขณะอยู่ในสถานะที่ผิด
- ContentException.EX_NOTIMPL ทำการเรียกฟังก์ชันที่ไม่มีในเวอร์ชันปัจจุบันของไลบรารี
- ContentException.EX_ERROR เป็นข้อผิดพลาดอื่น ๆ ที่นอกเหนือจาก ContentException

ตัวแปรสมาชิก

- m_code มีชนิดเป็นจำนวนอินทิเจอร์ รหัสข้อผิดพลาดของที่ออราเคิลกำหนดขึ้นมา
- m_type มีชนิดเป็นจำนวนอินทิเจอร์ บอกชนิดของข้อผิดพลาดที่เกิดขึ้นเป็นค่าคงที่
- m_msg มีชนิดเป็นจำนวนอินทิเจอร์ รหัสของสตริงที่เป็นข้อมูลของข้อผิดพลาดที่เกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมธอด

- toString() ทำการข้อมูลของข้อผิดพลาด

Syntax: *String* toString()

พารามิเตอร์ *m_type* ชนิดของข้อผิดพลาดที่เป็นจำนวนอินทิจอร์และจะผ่านสตริงที่เป็นข้อมูลของข้อผิดพลาดกลับมา



ภาคผนวก จ

ไฟล์มัลติมีเดีย

ในส่วนนี้จะอธิบายถึงรูปแบบและความหมายของการระบุตำแหน่งของไฟล์มัลติมีเดียเพื่อทำการควบคุมการเล่น (Control Playback)

1. รูปแบบของการระบุตำแหน่งของไฟล์มัลติมีเดีย

ในการที่จะใช้งานไฟล์มัลติมีเดียเหล่านี้จะต้องทำการเรียกคำสั่ง `Player.Load (mediafile, null)` ซึ่ง `mediafile` จะมีรูปแบบการระบุดังนี้

$$\text{mediafile} = \text{protocol}://\text{server.port}/\text{media_resource}$$

จากรูปแบบจะเห็นว่า การระบุตำแหน่งของไฟล์มัลติมีเดียจะเหมือนกับการกำหนด URL โดยความหมายของรูปแบบในการระบุมีดังนี้

- Protocol เป็นรูปแบบของการส่งข้อมูล ซึ่ง Oracle Video Client จะรองรับการกำหนดรูปแบบการส่งข้อมูล 2 ชนิดคือ
 1. vsudp กำหนดรูปแบบการส่งข้อมูลเป็นแบบ UDP ซึ่งเป็นค่าที่ตั้งไว้แล้วหากไม่ได้รับการกำหนด การทำงานของการส่งข้อมูลแบบ UDP คือจะไม่มีการส่งแพ็กเกจซ้ำในกรณีที่มีการตกหล่นของแพ็กเกจ สำหรับไฟล์สกุล AVI และ WAV นั้นจะไม่ได้ออกแบบมาเพื่อจัดการกับการตกหล่นของข้อมูล ดังนั้นหากต้องการใช้รูปแบบการส่งแบบนี้กับไฟล์ทั้งสองสกุล จึงควรมีการแปลงไฟล์ทั้งสองสกุลให้เป็นรูปแบบของออราเคิลเสียก่อนคือ OSF
 2. vstcp กำหนดรูปแบบการส่งข้อมูลเป็นแบบ TCP ซึ่งการส่งข้อมูลแบบนี้จะดีกว่าแบบ UDP คือ จะมีการส่งแพ็กเกจซ้ำหากมีการตกหล่นของแพ็กเกจ และต้องทำการเชื่อมต่อระหว่างไคลเอนต์และเซิร์ฟเวอร์ก่อนที่จะส่งข้อมูล ทำให้การส่งข้อมูลมีความถูกต้อง ดังนั้นไฟล์สกุล AVI และ MOV จึงไม่จำเป็นต้องแปลงเป็นไฟล์ในรูปแบบของออราเคิลเลย
- Server เป็นการกำหนดเลขที่อยู่ของเซิร์ฟเวอร์ที่ไคลเอนต์ต้องการจะดึงไฟล์มัลติมีเดียมาใช้งาน เช่น 161.246.6.115 เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Port กำหนดช่องทางการติดต่อระหว่างไคลเอนต์และเซิร์ฟเวอร์ซึ่งอราเคิลได้กำหนดช่องทางการติดต่อไว้ที่ 5000 หากผู้ใช้ไม่ได้กำหนดที่ตั้งเลขที่อยู่ของเซิร์ฟเวอร์และช่องทางการติดต่อไว้ ก็จะนำค่าที่กำหนดโดย Oracle Video Client Setting กำหนดไว้มาใช้
- Media_Resource เป็นการกำหนดไฟล์ MDS หรือ Logical Content ที่ต้องการจะใช้จากเซิร์ฟเวอร์

ตัวอย่างการระบุตำแหน่งของไฟล์มีคติมิตยชื่อ ending.mpi ที่เก็บอยู่บนเซิร์ฟเวอร์ที่มีเลขที่อยู่เท่ากับ 161.246.6.115

`vstcp://161.246.6.115:5000/vscontsrv : ending.mpi`



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ฉ

โปรแกรม

```

import java.awt.*;
import java.io.*;
import java.sql.*;
import oracle.jdbc.driver.*;
import oracle.ovc.PlayerFactory;
import oracle.ovc.PlayerException;
import oracle.ovc.Player;
import oracle.ovc.*;

public class OVCPlayer extends Frame
{
    static Player player = null;
    static Component playerUI = null;
    static String m_MediaFile = "";
    static StmOpts optStream = null;
    static CheckboxMenuItem lowVol,midVol,hiVol,normScr,fullScr,autoStart,autoLoop;
    static TextField showStatus;
    static List list,clipList;
    static boolean load,fromServer;
    static String m_Name,protocol,serverName,portName;
    static Dialog n;

    public OVCPlayer()
    {
        setTitle("Movie Player");
        setIconImage(getToolkit().getImage("OvcIcon.jpg"));
        MenuBar menubar = new MenuBar();
        Menu menu = new Menu("File");
        menu.add(new MenuItem("Load Content"));
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

menu.add(new MenuItem("Unload"));
menu.addSeparator();
menu.add(new MenuItem("Quit"));
menubar.add(menu);
menu = new Menu("Playback");
menu.add(new MenuItem("Play"));
menu.add(new MenuItem("Pause"));
menu.add(new MenuItem("Stop"));
menu.addSeparator();
Menu videoSize =new Menu("Video Size");
normScr = new CheckboxMenuItem("Normal");
normScr.setState(true);
videoSize.add(normScr);
fullScr = new CheckboxMenuItem("Full Screen");
videoSize.add(fullScr);
menu.add(videoSize);
Menu volume = new Menu("Volume");
lowVol = new CheckboxMenuItem("Low");
lowVol.setState(true);
volume.add(lowVol);
midVol = new CheckboxMenuItem("Medium");
volume.add(midVol);
hiVol = new CheckboxMenuItem("High");
volume.add(hiVol);
menu.add(volume);
menubar.add(menu);
menu=new Menu("Preferences");
autoStart = new CheckboxMenuItem("Auto Start");
autoLoop = new CheckboxMenuItem("Auto Loop");
menu.add(autoStart);
menu.add(autoLoop);
menu.addSeparator();
menu.add(new MenuItem("Network Setting"));
menubar.add(menu);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

menu=new Menu("Help");
menu.add(new MenuItem("About"));
menubar.add(menu);
setMenuBar(menubar);
setBackground(Color.black);
resize(450,400);
optStream = new StmOpts();
optStream.m_autoStart = false;
optStream.m_loop = false;
showStatus = new TextField();
add("South",showStatus);
clipList = new List(4,false);
n = new NetworkSet(this);
}

public static void main(String args[])
{
    OVPlayer app = new OVPlayer();
    app.addPlayer();
    app.add(playerUI);
    app.show();
}

public boolean handleEvent(Event evt)
{
    if(evt.id==Event.WINDOW_DESTROY && evt.target == this)
    {
        showStatus.setText("Shutting down the java client ...");
        if (player != null)
        {
            try
            {
                showStatus.setText("Player state before exiting: " + player.getState());
                player.unload ();
            }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        player.term ();
    }
    catch (PlayerException e) {
        showStatus.setText("exception while unloading: " + e);
    }
}
System.exit(0);
}
if (evt.id==Event.MOUSE_DOWN)
{
    if (fullScr.getState())
    {
        normScr.setState(true);
        fullScr.setState(false);
        try {
            player.setFullScreen(false);
        }
        catch (PlayerException e) {
            showStatus.setText("exception while set normal screen : " + e);
        }
    }
}
return super.handleEvent(evt);
}

public boolean action(Event evt, Object arg)
{
    if (evt.target == normScr)
    {
        normScr.setState(true);
        fullScr.setState(false);
        try {
            player.setFullScreen(false);
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

catch (PlayerException e) {
    showStatus.setText("exception while set normal screen : " + e);
}
}
else
if (evt.target == fullScr)
{
    normScr.setState(false);
    fullScr.setState(true);
    try {
        player.setFullScreen(true);
    }
    catch (PlayerException e) {
        showStatus.setText("exception while set full screen : " + e);
    }
}
else
if (evt.target == lowVol)
{
    lowVol.setState(true);
    midVol.setState(false);
    hiVol.setState(false);
    try {
        player.setVol(33);
    }
    catch (PlayerException e) {
        showStatus.setText("exception while set low volume : " + e);
    }
}
else
if (evt.target == midVol)
{
    lowVol.setState(false);
    midVol.setState(true);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        hiVol.setState(false);
    try {
        player.setVol(66);
    }
    catch (PlayerException e) {
        showStatus.setText("exception while set medium volume : " + e);
    }
}
else
if (evt.target == hiVol)
{
    lowVol.setState(false);
    midVol.setState(false);
    hiVol.setState(true);
    try {
        player.setVol(100);
    }
    catch (PlayerException e) {
        showStatus.setText("exception while set high volume : " + e);
    }
}
else
if (evt.target == autoStart)
{
    if (autoStart.getState())
    {
        optStream.m_autoStart=true;
    }
    else
    {
        optStream.m_autoStart=false;
    }
}
else

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (evt.target == autoLoop)
{
    if (autoLoop.getState())
    {
        optStream.m_loop=true;
    }
    else
    {
        optStream.m_loop=false;
    }
}
else
if (evt.target instanceof MenuItem)
{
    if (arg.equals("Load Content"))
    {
        Dialog d = new LoadDialog(this);
        d.show();
        if (load)
        {
            try {
                player.unload ();
            }
            catch (PlayerException e) {
                showStatus.setText("exception while unloading: " + e);
            }
        }
        loadMediaFile();
    }
}
if (arg.equals("Unload"))
{
    if (player != null)
    {
        try {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        showStatus.setText("Player state before unloading: " + player.getState());
        player.unload ();
        load=false;
    }
    catch (PlayerException e) {
        showStatus.setText("exception while unloading: " + e);
    }
}

if (arg.equals("Quit"))
{
    showStatus.setText("Shutting down the java client ...");
    if (player != null)
    {
        try {
            showStatus.setText("Player state before exiting: " + player.getState());
            player.unload ();
            player.term ();
        }
        catch (PlayerException e) {
            showStatus.setText("exception while unloading: " + e);
        }
    }
    System.exit(0);
}

if (arg.equals("Play"))
{
    if (load)
    {
        try {
            if (player.getState() == Player.ST_PAUSED) player.resume ();
            else player.play();
        }
        catch (PlayerException e) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        showStatus.setText("exception while play : " + e);
    }
}

if (arg.equals("Pause"))
{
    if (load)
    {
        try {
            player.pause ();
        }
        catch (PlayerException e) {
            showStatus.setText("exception while pause : " + e);
        }
    }
}

if (arg.equals("Stop"))
{
    if (load)
    {
        try {
            player.stop ();
        }
        catch (PlayerException e) {
            showStatus.setText("exception while stop : " + e);
        }
    }
}

if (arg.equals("Network Setting"))
{
    n.show();
}

if (arg.equals("About"))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        {
            Dialog d = new AboutDialog(this);
            d.show();
        }
    }
    else return false;
    return true;
}

```

```
public void addPlayer()
```

```

{
    try {
        player = PlayerFactory.getPlayer ();
    }
    catch ( UnsatisfiedLinkError e ) {
        showStatus.setText(e.getMessage());
    }
    catch ( PlayerException e ) {
        showStatus.setText("Unable to create a Player object: " + e);
    }
    try {
        playerUI = player.getPlayerUI ();
    }
    catch ( PlayerException e ) {
        showStatus.setText("exception in getPlayerUI: " + e);
    }
}

```

```
public void loadMediaFile()
```

```

{
    try {
        player.load(m_MediaFile,optStream);
    }
    catch ( PlayerException e ) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        showStatus.setText("Could not load content: " + e);
    }
}
}

```

```
class NetworkSet extends Dialog
```

```

{
    Checkbox    tcp,udp;
    TextField   t1,t2;
    public NetworkSet(Frame parent)
    {
        super(parent,"Network Setting",true);
        Panel p = new Panel();
        p.setBackground(Color.lightGray);
        resize(280,150);
        setResizable(false);
        p.setLayout(null);
        Label label1 = new Label("Enter Server Address : Port");
        label1.reshape(5,5,200,20);
        p.add(label1);
        t1 = new TextField("161.246.6.118",11);
        t1.reshape(20,30,120,20);
        p.add(t1);
        t2 = new TextField("5000",5);
        t2.reshape(140,30,60,20);
        p.add(t2);
        Button ok = new Button("OK");
        ok.reshape(210,40,50,20);
        p.add(ok);
        Button preview = new Button("Cancel");
        preview.reshape(210,70,50,20);
        p.add(preview);
        Label label2 = new Label("Transport Protocol");
        label2.reshape(10,60,120,20);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

p.add(label2);
tcp = new Checkbox("TCP",null,false);
tcp.reshape(20,70,60,50);
p.add(tcp);
udp = new Checkbox("UDP",null,true);
udp.reshape(130,70,80,50);
p.add(udp);
add("Center",p);
OVCPlayer.protocol="vsudp://";
OVCPlayer.serverName= "161.246.6.118"+":";
OVCPlayer.portName= "5000"+"vscontsrv:";
}

public boolean action(Event evt,Object arg)
{
    if (evt.target instanceof Button)
    {
        if (arg.equals("OK"))
        {
            if (udp.getState())
            {
                OVCPlayer.protocol="vsudp://";
            }
            else
            {
                OVCPlayer.protocol="vstcp://";
            }
            OVCPlayer.serverName= t1.getText()+":";
            OVCPlayer.portName= t2.getText()+"/vscontsrv:";
            dispose();
        }

        if (arg.equals("Cancel"))
        {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        dispose();
    }
}
else
if (evt.target instanceof Checkbox)
{
    if (evt.target.equals(tcp))
    {
        tcp.setState(true);
        udp.setState(false);
    }
    else
    if (evt.target.equals(udp))
    {
        tcp.setState(false);
        udp.setState(true);
    }
}
else return false;
return true;
}
}

class AboutDialog extends Dialog
{
    public AboutDialog(Frame parent)
    {
        super(parent,"About the Auther",true);
        Panel p= new Panel();
        p.setBackground(Color.lightGray);
        resize(200,150);
        setResizable(false);
        p.setLayout(null);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Label label1 = new Label("Written By");
label1.reshape(65,15,60,15);
p.add(label1);
Label label2 = new Label("Mr.Sivanath Massurang");
label2.reshape(25,35,140,15);
p.add(label2);
Label label3 = new Label("Mr.Wirat Wasinpiyamongkol");
label3.reshape(25,55,140,15);
p.add(label3);
Button ok = new Button("OK");
ok.reshape(65,90,50,20);
p.add(ok);
add("Center",p);
}

public boolean action(Event evt, Object arg)
{
    if (arg.equals("OK")) dispose();
    else return false;
    return true;
}
}

class LoadDialog extends Dialog
{
    Checkbox server,local;
    public LoadDialog(Frame parent)
    {
        super(parent,"Load Content",true);
        Panel p = new Panel();
        p.setBackground(Color.lightGray);
        resize(320,220);
        setResizable(false);
        p.setLayout(null);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Label label1 = new Label("Enter Content URL : ");
label1.reshape(5,5,120,20);
p.add(label1);
Panel p1 = new Panel();
OVCPlayer.list = new List(4,false);
OVCPlayer.list.reshape(10,30,200,100);
p1.add("South",OVCPlayer.list);
p1.reshape(10,30,200,100);
p.add(p1);
Button ok = new Button("Fullview");
ok.reshape(245,45,50,20);
p.add(ok);
Button preview = new Button("Preview");
preview.reshape(245,75,50,20);
p.add(preview);
Button cancel = new Button("Cancel");
cancel.reshape(245,105,50,20);
p.add(cancel);
Button browse = new Button("Browse");
browse.reshape(245,155,50,20);
p.add(browse);
Label label2 = new Label("Content Query ");
label2.reshape(10,140,120,20);
p.add(label2);
server = new Checkbox("Server",null,true);
server.reshape(20,150,60,50);
p.add(server);
local = new Checkbox("Local Disk",null,false);
local.reshape(130,150,80,50);
p.add(local);
add("Center",p);
}

```

```
public boolean action(Event evt, Object arg)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    if (evt.target instanceof Button)
    {
        if (arg.equals("Fullview"))
        {
            if (OVCPlayer.fromServer)
            {
                OVCPlayer.m_MediaFile=OVCPlayer.protocol+
                OVCPlayer.serverName+OVCPlayer.portName+
                OVCPlayer.list.getSelectedItem();
            }
            else
            {
                OVCPlayer.m_MediaFile=OVCPlayer.list.getSelectedItem();
            }
            dispose();
            OVCPlayer.load=true;
        }
        if (arg.equals("Preview"))
        {
            if (OVCPlayer.fromServer)
            {
                OVCPlayer.m_MediaFile=OVCPlayer.protocol+
                OVCPlayer.serverName+OVCPlayer.portName+
                OVCPlayer.clipList.getItem(OVCPlayer.list.getSelectedIndex());
            }
            else
            {
                OVCPlayer.m_MediaFile=OVCPlayer.list.getSelectedItem();
            }
            dispose();
            OVCPlayer.load=true;
        }
        if (arg.equals("Cancel"))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        dispose();
        OVCPlayer.load=false;
    }
    if (arg.equals("Browse"))
    {
        OVCPlayer.list.clear();
        OVCPlayer.clipList.clear();
        if (local.getState())
        {
            LoadFileDialog loadFile = new LoadFileDialog();
            if (OVCPlayer.m_Name != null)
            {
                OVCPlayer.list.addItem(OVCPlayer.m_MediaFile);
                OVCPlayer.showStatus.setText(OVCPlayer.m_Media
                File);
                OVCPlayer.fromServer=false;
            }
            if (server.getState())
            {
                try
                {
                    ConnectDB connec = new ConnectDB();
                }
                catch (SQLException e){}
                catch (IOException e) {}
                OVCPlayer.fromServer=true;
            }
        }
        OVCPlayer.list.select(0);
    }
}
else
    if (evt.target instanceof Checkbox)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        if (evt.target.equals(server))
        {
            server.setState(true);
            local.setState(false);
        }
        else
        if (evt.target.equals(local))
        {
            server.setState(false);
            local.setState(true);
        }
    }
    else return false;
    return true;
}
}

class LoadFileDialog
{
    LoadFileDialog()
    {
        FileDialog fd = new FileDialog(new Frame(), null, FileDialog.LOAD);
        fd.setDirectory("D:\\MyDocument\\Project");
        fd.setFile("*.mpg");
        fd.setVisible(true);
        OVCPlayer.m_MediaFile = fd.getDirectory() + fd.getFile();
        OVCPlayer.m_Name = fd.getFile();
    }
}

class ConnectDB
{
    ConnectDB() throws SQLException, IOException

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());
    Connection conn =
    DriverManager.getConnection("jdbc:oracle:thin:@dbserver:1521:orcl","ovs","oraclevideo
server");
    Statement stmt = conn.createStatement ();
    String sql = "select content,clips from preview";
    ResultSet rset = stmt.executeQuery (sql);
    while (rset.next ())
    {
    OVCPlayer.list.addItem(rset.getString(1));
    OVCPlayer.clipList.addItem(rset.getString(2));
    }
    conn.close();

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

หนังสืออ้างอิง

1. Gary Cornell : Cay S.Horstmann, “Core JAVA”, Sunsoft press,1996
2. Mitchell Waite , “Object Oriented Programming in JAVA”, Wait Group Inc,1997
3. ดร.วีระศักดิ์ ชิงกาवर , “ Fundamental of JAVA Programming ”, ซีเอ็ด,1999

เอกสารอ้างอิง

1. “Oracle Video Server Installation Guide”, Oracle Corporation, 1998
Part No. A60125-01
2. Nancy Baltz : Christian Bedford : Aaron Hoffmeyer, “Oracle Video Server Administrator’s Guide and Command Reference”, Oracle Corporation, 1998
Part No. A60126-01
3. John Dowden, “Getting Started with Oracle Video Server Manager”, Oracle Corporation, 1998
Part No. A55979-02
4. Brian Linden : Matt Prather, “Introducing Oracle Video Server”, Oracle Corporation, 1998
Part No. A53956-02
5. Sheryl Maring, “Oracle Media Net Developer's Guide, 3.3”, Oracle Corporation,1998
Part No. A53599-01
6. Gordon Furbush : Denise Stone, “Oracle Video Server Developer’s Guide, Release 3.0”, Oracle Corporation,1998
Part No. A53960-02
7. Rick Herrick, “Oracle Video Client Developer’s Guide”, Oracle Corporation,1998
Part No: 3.0 A53949-02

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้