

**สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง**

ระบบการจัดการทรัพย์สินเทคโนโลยีสารสนเทศ  
IT ASSET MANAGEMENT SYSTEM



เลขหมู่.....  
เลขทะเบียน.....**62331**  
วัน,เดือน,ปี.....**16 ส.ค. 2549**

b.....**11621229**  
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
ภาควิชาวิศวกรรมสารสนเทศ  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# IT ASSET MANAGEMENT SYSTEM



BY

MR. RATCH

RUAYLARP

MS. RACHAPOL

THAISITTIVECH

A THESIS SUBMITTED IN PARTIAL FULILLMENT OF

THE REQUIREMENT FOR THE DEGREE OF

BACHELOR IN DEPARTMENT OF INFORMATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2005

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	ระบบการจัดการทรัพย์สินเทคโนโลยีสารสนเทศ	
TITLE	IT ASSET MANAGEMENT SYSTEM	
ชื่อนักศึกษา	นาย รัชต์ รวยลาภ	รหัสประจำตัว 45010639
	นาย รัชพล ไทยสิทธิเวช	รหัสประจำตัว 45010641
อาจารย์ผู้ควบคุมปริญญานิพนธ์	อาจารย์ นิจจารีย์ สัตยารักษ์	
ระดับการศึกษา	ปริญญาวิศวกรรมศาสตรบัณฑิต	
	สาขาวิชาวิศวกรรมสารสนเทศ	
ภาควิชา	วิศวกรรมสารสนเทศ	
ปีการศึกษา	2548	

ปริญญานิพนธ์ฉบับนี้ได้รับความเห็นชอบจากอาจารย์ผู้ควบคุมปริญญานิพนธ์เป็นที่เรียบร้อยแล้ว



.....  
(อาจารย์ นิจจารีย์ สัตยารักษ์)  
อาจารย์ผู้ควบคุมปริญญานิพนธ์

ลิขสิทธิ์ของ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**หัวข้อปริญญานิพนธ์**

ระบบการจัดการทรัพย์สินเทคโนโลยีสารสนเทศ

IT ASSET MANAGEMENT SYSTEM

**รื่อนักศึกษา**นาย **รัชต์ รวยฉาบ**

รหัสนักศึกษา 45010639

นาย **รัชพล ไทยอิทธิเวช**

รหัสนักศึกษา 45010641

**อาจารย์ผู้ควบคุมปริญญานิพนธ์**อาจารย์ **นิจจารีย์ สัตยารักษ์****ระดับการศึกษา**

ปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมสารสนเทศ

**ภาควิชา**

วิศวกรรมสารสนเทศ

**ปีการศึกษา**

2548

**บทคัดย่อ**

โครงการนี้เป็นการคิดค้นวิธีในการจัดการทรัพย์สินเทคโนโลยีสารสนเทศ ในปัจจุบันนี้เราจะเห็นว่าไม่ใช่นักที่จะทำการจัดสรรทรัพย์สินขององค์กรให้มีความเหมาะสมและคอยดูแลสิทธิของซอฟต์แวร์ทั้งหมดที่มีอยู่ โดยผลที่ได้จากระบบการจัดการทรัพย์สินเทคโนโลยีสารสนเทศจะแสดงผลได้อย่างถูกต้องแม่นยำ รวมถึงการใช้ฮาร์ดแวร์ให้มีประโยชน์สูงสุด ด้วยวิธีการนี้ประกอบด้วยมาตรฐานหลายอย่างรวมกัน เช่น CIM (Common Information Model), COM+ (Component Object Model), .NET Framework, Win32 API ซึ่งวิธีนี้ทำให้สะดวกต่อการจัดการ และยังเพิ่มประสิทธิภาพให้กับองค์กรอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Thesis Title** IT ASSET MANAGEMENT SYSTEM  
**Student** **Mr. Ratch Ruaylarp** **ID. 450106939**  
**Mr. Rachapol Thaisittivech** **ID. 45010641**  
**Advisor** Miss Nitcharee Sattayarak  
**Graduate Level** Bachelor Degree of Information Engineering  
**Department** Information Engineering  
**Academic Year** 2005

## ABSTRACT

The proposal of this project is implementation the solution to manage its IT assets. Nowadays, we realize that it is not always easy for the corporate entity to accurately create and maintain adequate record of all software licenses, system inventories, utilization that IT asset management enables you to monitor more accurately, utilize hardware economically and software products with combination of many standards such as CIM (Common Information Model), COM+ (Component Object Model), .NET Framework, Win32 API and this solution is going to bring ease of management and increasing ROI (Return On Investment) to your organization.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้คงมีอาจสำเร็จไปได้เลย หากปราศจากความช่วยเหลือและความร่วมมือจากหลายฝ่ายด้วยกัน บุคคลสำคัญได้แก่ อาจารย์นิจจารีย์ ศักยารักษ์ อาจารย์ผู้ควบคุมปริญญาานิพนธ์ ผู้คอยให้ความช่วยเหลือ แนะนำ ให้คำปรึกษา และเอาใจใส่เป็นอย่างดีตลอดระยะเวลาทั้งหมดที่ทำปริญญาานิพนธ์ ซึ่งขอขอบพระคุณเป็นอย่างยิ่ง

ขอขอบคุณแนวความคิดต่างๆ รวมถึงให้กำลังใจและคอยช่วยเหลือเมื่อประสบปัญหา ทั้งนี้ รวมถึงเพื่อนๆ ร่วมรุ่นทุกคนที่คอยให้ความช่วยเหลือ ห่วงใย เป็นกำลังใจด้วยดีเสมอมา

สุดท้ายนี้คณะผู้จัดทำต้องขอกราบขอบพระคุณคณาจารย์ทุกท่านที่ได้ประสิทธิ์ประสาทความรู้ วิชาการต่างๆ แก่ศิษย์ จนเกิดเป็นปริญญาานิพนธ์ฉบับนี้ รวมถึงบุคคลสำคัญที่สุด คือ บิดา มารดา ผู้เป็นที่เคารพรักยิ่ง ผู้เป็นกำลังใจ และคอยสนับสนุนในทุกๆ ด้าน จนทำให้คณะผู้จัดทำมีวันนี้ จึงขอกราบขอบพระคุณมา ณ ที่นี้

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

เรื่อง	หน้า
บทคัดย่อไทย	ก
บทคัดย่ออังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญรูปภาพ	ช
<b>บทที่ 1 บทนำ</b>	
1.1 แนวคิดและที่มาของปัญหา	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตของโครงการ	1
1.4 สถาปัตยกรรมของระบบ	2
1.5 ขั้นตอนการดำเนินงาน	3
<b>บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง</b>	
2.1 Windows Management Instrumentation : WMI	4
2.1.1 แนะนำ WMI	4
2.1.2 โครงสร้างของ WMI	4
2.2 Microsoft .NET	6
2.2.1 แนะนำ Microsoft .NET	6
2.2.2 Microsoft .NET Framework	6
2.2.3 ประโยชน์และข้อดีของ .NET Framework	7
2.3 Windows Socket	7
2.3.1 แนะนำ Windows Socket	7
2.3.2 ระดับของ Winsock ในส่วนของ TCP/IP	8
2.3.3 Client/Server	9
2.3.4 การสร้าง Client/Server โดยการใช้ Winsock	9
2.3.5 การทำงานของ Server	10
2.3.6 การทำงานของไคลเอนท์	10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

	หน้า
2.5.19 Activating Client Object	30
<b>บทที่ 3 การวิเคราะห์และการออกแบบ</b>	
3.1 Domain Model	31
3.2 Class Diagram	35
3.3 Sequence Diagram	72
3.4 ER - Diagram	74
3.5 Data Dictionary	75
<b>บทที่ 4 การทดลองและผลจากการทดลอง</b>	
4.1 การใช้งานบนฝั่งเซิร์ฟเวอร์	86
4.2 การใช้งานบนฝั่งไคลเอนท์	87
4.3 การใช้งานส่วนคุมระบบ	88
4.3.1 ส่วนคำสั่งโปรแกรม	89
4.3.1.1 การเข้าสู่ระบบ	89
4.3.1.2 การออกจากระบบ	91
4.3.1.3 แสดงการออกจากโปรแกรม	92
4.3.2 ส่วนการจัดการระบบ	92
4.3.2.1 ส่วนการจัดการผู้ใช้งานระบบ	92
4.3.2.1.1 การเพิ่มผู้ใช้งานระบบ	93
4.3.2.1.2 การลบผู้ใช้งานระบบ	95
4.3.2.1.3 การเปลี่ยนรหัสผ่านผู้ใช้งานระบบ	97
4.3.2.1.4 การเปลี่ยนสิทธิผู้ใช้งานระบบ	98
4.3.2.1.5 การกำหนดสิทธิการดูแลเครื่อง	99
4.3.2.2 แสดงการจัดการระบบหลัก	101
4.3.2.2.1 การดูรายละเอียดของแต่ละระบบ	101
4.3.2.2.2 การจับหน้าจอภาพของระบบ	102
4.3.2.2.3 การดูคุณสมบัติของระบบ	103
4.3.2.2.4 การเรียกดูรายงาน	105
4.3.3 ส่วนการจัดหน้าต่างของระบบ	108

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

	หน้า
4.3.3.1 การจัดการหน้าต่างแบบкасเคด	109
4.3.3.2 การจัดการหน้าต่างตามแนวขวาง	110
4.3.3.3 การจัดการหน้าต่างตามแนวตั้ง	134
<b>บทที่ 5 สรุปผลการทดลอง</b>	112
5.1 สรุปผลการดำเนินโครงการ	112
5.2 ปัญหาที่พบในระหว่างการดำเนินโครงการ	112
5.3 แนวทางการพัฒนาโครงการต่อ	112
5.4 สรุปผล	112
<b>บรรณานุกรม</b>	113



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

	หน้า
2.4 Windows Services	10
2.4.1 แนะนำ Windows Services	10
2.4.2 โครงสร้างของวินโดวส์เซอร์วิส	11
2.4.2.1 Service Program	12
2.4.2.2 Service Control Manager	12
2.4.2.3 Main Function, Service Main and Handler	13
2.4.3 Service Control Program	14
2.4.4 Service Configuration Program	14
2.5 .NET Remoting	14
2.5.1 .NET Remoting คืออะไร	14
2.5.2 ชนิดของแอปพลิเคชันและโปรโตคอล	15
2.5.3 CLR Object Remoting	15
2.5.4 .NET Remoting Overview	15
2.5.5 Context	18
2.5.6 Activation	19
2.5.7 Communication Between Contexts	20
2.5.8 Remote Objects, Clients and Servers	20
2.5.9 Remote Object	21
2.5.10 โครงสร้างของ .NET Remoting	22
2.5.10.1 แชนเนล	22
2.5.11 Setting Channel Property	25
2.5.12 Pluggability of Channel	26
2.5.13 Formatter	26
2.5.14 Channel Service Remoting and Configuration	27
2.5.15 Server for Client-Activated Object	28
2.5.16 Object Activation	29
2.5.17 Application URL	29
2.5.18 Activating Well-Known Object	29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูปภาพ

หน้า

รูปที่ 1-1 สถาปัตยกรรมของระบบ	2
รูปที่ 1-2 ขั้นตอนการดำเนินโครงการ	3
รูปที่ 2-1 การทำงานร่วมกันของคอมพิวเตอร์ต่างๆ	5
รูปที่ 2-2 การสร้างช่องทางการสื่อสารของ Winsock	8
รูปที่ 2-3 การส่งข้อมูลของ Winsock	8
รูปที่ 2-4 การทำงานบนระบบ Client/Server	9
รูปที่ 2-5 การสร้าง Client/Server โดยใช้ฟังก์ชันของ Winsock	9
รูปที่ 2-6 Services Administration Tool	11
รูปที่ 2-7 การทำงานของ SCM	12
รูปที่ 2-8 ตัวอย่างของ Service Control Program	13
รูปที่ 2-9 ภาพรวมของส่วนประกอบต่างๆ	17
รูปที่ 2-10 รายละเอียด Server	18
รูปที่ 2-11 คลาสหลักของแอปพลิเคชัน	21
รูปที่ 2-12 คลาสของรีโมทออบเจกต์	22
รูปที่ 2-13 โค้ดที่ใช้ในการสร้างเซิร์ฟเวอร์	22
รูปที่ 2-14 การสร้าง HTTP และ TCP	23
รูปที่ 2-15 การแสดงข้อมูลของเซิร์ฟเวอร์	24
รูปที่ 2-16 เมธอดในการสร้างเซิร์ฟเวอร์	25
รูปที่ 2-17 ออบเจกต์ในการรับส่งข้อมูล	25
รูปที่ 2-18 ผลลัพธ์ออกมาทางคอนโซล	26
รูปที่ 2-19 เซิร์ฟเวอร์ในการรีจิสเตอร์ของเซิร์ฟเวอร์ HTTP และ TCP	27
รูปที่ 2-20 การจัดการเซิร์ฟเวอร์	27
รูปที่ 2-21 ตัวอย่างโค้ดของฝั่งเซิร์ฟเวอร์	28
รูปที่ 2-22 การตั้งค่าแบบ Activated Object	29
รูปที่ 2-23 แสดง URL	29
รูปที่ 2-24 รายละเอียดของการ Activate	29
รูปที่ 2-25 ออบเจกต์ของรีโมทเซิร์ฟเวอร์	30
รูปที่ 2-26 เมธอดในการสร้างรีโมทออบเจกต์	30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูปภาพ (ต่อ)

หน้า

รูปที่ 2-27 คลาส URL Attribute	31
รูปที่ 2-28 ออบเจกต์ที่ใช้แทนคลาส Activator	31
รูปที่ 3-1 Domain Model ในส่วนที่เกี่ยวกับรีโมทออบเจกต์	32
รูปที่ 3-2 Domain Model ในส่วนที่เกี่ยวกับการเก็บข้อมูลจากเครื่องไคลเอนท์	33
รูปที่ 3-3 Domain Model ในส่วนที่เกี่ยวกับการดักจับจอภาพไคลเอนท์	34
รูปที่ 3-4 Domain Model ในส่วนที่เกี่ยวกับส่วนควบคุมระบบ	34
รูปที่ 3-5 Class Diagram ในส่วนที่เกี่ยวกับรีโมทออบเจกต์	35
รูปที่ 3-6 Class Diagram ในส่วนที่เกี่ยวกับการเก็บข้อมูลจากเครื่องไคลเอนท์	36
รูปที่ 3-7 Class Diagram ในส่วนที่เกี่ยวกับการดักจับภาพไคลเอนท์	37
รูปที่ 3-8 Class Diagram ในส่วนที่เกี่ยวกับส่วนควบคุมของระบบ	38
รูปที่ 3-9 Class ASM.Client.Main Class	39
รูปที่ 3-10 Class ASM.Client.WindowsService	39
รูปที่ 3-11 Class.Component.BaseBoard	39
รูปที่ 3-12 Class ASM.Component.BIOS	40
รูปที่ 3-13 Class ASM.Component.CacheMemory	41
รูปที่ 3-14 Class ASM.Component.CDRomDrive	42
รูปที่ 3-15 Class ASM.Component.ComputerSystem	43
รูปที่ 3-16 Class ASM.Component.Desktop	44
รูปที่ 3-17 Class ASM.Component.DesktopMonitor	45
รูปที่ 3-18 Class ASM.Component.LocalTime	45
รูปที่ 3-19 Class ASM.Component.DiskDrive	46
รูปที่ 3-20 Class ASM.Component.Environment	47
รูปที่ 3-21 Class ASM.Component.FloppyDrive	47
รูปที่ 3-22 Class ASM.Component.Group	47
รูปที่ 3-23 Class ASM.Component.LogicalDisk	48
รูปที่ 3-24 Class ASM.Component.NetworkAdaptor	49
รูปที่ 3-25 Class ASM.Component.NetworkAdaptorConfiguration	50
รูปที่ 3-26 Class ASM.Component.NetworkLoginProfile	51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูปภาพ (ต่อ)

	<b>หน้า</b>
รูปที่ 3-27 Class ASM.Component.OperatingSystem	52
รูปที่ 3-28 Class ASM.Component.PageFile	53
รูปที่ 3-29 Class ASM.Component.PageFileUsage	54
รูปที่ 3-30 Class ASM.Component.PhysicalMemory	54
รูปที่ 3-31 Class ASM.Component.POTSModem	55
รูปที่ 3-32 Class ASM.Component.Printer	55
รูปที่ 3-33 Class ASM.Component.Process	56
รูปที่ 3-34 Class ASM.Component.processor	57
รูปที่ 3-35 Class ASM.Component.ScreenCapturing	58
รูปที่ 3-36 Class ASM.Component.StartupCommand	58
รูปที่ 3-37 Class ASM.Component.SystemEnclosure	58
รูปที่ 3-38 Class ASM.Component.TimeZone	58
รูปที่ 3-39 Class ASM.Component.SystemSlot	59
รูปที่ 3-40 Class ASM.Component.UserAccount	59
รูปที่ 3-41 Class ASM.Component.VideoController	60
รูปที่ 3-42 Class ASM.Component.Win32	61
รูปที่ 3-43 Class ASM.Console.FormUserManagement	61
รูปที่ 3-44 Class ASM.Console.FormSystemSummary	62
รูปที่ 3-45 Class ASM.Console.FormProperties	62
รูปที่ 3-46 Class ASM.Console.FormImageRotator	62
รูปที่ 3-47 Class ASM.Console.FormImage	63
รูปที่ 3-48 Class ASM.Console.FormChangePassword	63
รูปที่ 3-49 Class ASM.Console.TextPrintDocument	63
รูปที่ 3-50 Class ASM.DataAccess.Client	64
รูปที่ 3-51 Class ASM.DataAccess.ComputerSystem	65
รูปที่ 3-52 Class ASM.DataAccess.Event	66
รูปที่ 3-53 Class ASM.DataAccess.EventAdmin	66
รูปที่ 3-54 Class ASM.DataAccess.Screen	66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูปภาพ (ต่อ)

หน้า

รูปที่ 3-55 Class ASM.DataAccess.User	67
รูปที่ 3-56 Class ASM.RemoteObject.Authenticator	67
รูปที่ 3-57 Class ASM.RemoteObject.Client	68
รูปที่ 3-58 Class ASM.RemoteObject.ClientController	69
รูปที่ 3-59 Class ASM.RemoteObject.EventManager	69
รูปที่ 3-60 Class ASM.RemoteObject.Computer	70
รูปที่ 3-61 Class ASM.RemoteObject.PicsManager	71
รูปที่ 3-63 Class ASM.Server.MainClass	71
รูปที่ 3-64 Sequence Diagram ในส่วนที่เกี่ยวกับการเก็บข้อมูลจากเครื่องไคลเอนท์	72
รูปที่ 3-65 Sequence Diagram ในส่วนที่เกี่ยวกับการดึงข้อมูลจากเครื่องไคลเอนท์	73
รูปที่ 3-66 ER-Diagram รวมของระบบ	74
รูปที่ 4-1 การทำงานเมื่อเริ่มต้นการทำงานของเซิร์ฟเวอร์	86
รูปที่ 4-2 การปรับแต่งค่าการทำงานของเซิร์ฟเวอร์	86
รูปที่ 4-3 วินโดว์เซอร์วิสทางฝั่งไคลเอนท์	87
รูปที่ 4-4 การปรับแต่งค่าการทำงานของไคลเอนท์	87
รูปที่ 4-5 หน้าหลักของส่วนควบคุมระบบ	88
รูปที่ 4-6 เมนูการเปิดหน้าจอเข้าสู่ระบบ	89
รูปที่ 4-7 หน้าจอการเข้าสู่ระบบ	89
รูปที่ 4-8 หน้าจอเมื่อเข้าสู่ระบบเรียบร้อยแล้ว	90
รูปที่ 4-9 การเข้าสู่ระบบผิดพลาด	90
รูปที่ 4-10 การออกจากระบบ	91
รูปที่ 4-11 การออกจากโปรแกรม	91
รูปที่ 4-12 การจัดการผู้ใช้งานระบบ	92
รูปที่ 4-13 หน้าจอการจัดการผู้ใช้งานระบบ	92
รูปที่ 4-14 การเพิ่มผู้ใช้งานระบบ	93
รูปที่ 4-15 หน้าจอการเพิ่มผู้ใช้งานระบบ	93
รูปที่ 4-16 การเพิ่มผู้ใช้งานเสร็จสมบูรณ์	94
รูปที่ 4-17 หน้าจอผู้ใช้งานระบบหลังจากเพิ่มผู้ใช้งานเรียบร้อยแล้ว	94

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูปภาพ (ต่อ)

	หน้า
รูปที่ 4-18 การลบผู้ใช้งานระบบ	95
รูปที่ 4-19 การลบผู้ใช้งานเสร็จสมบูรณ์	95
รูปที่ 4-20 หน้าจอผู้ใช้งานระบบหลังจากลบผู้ใช้งานระบบเรียบร้อยแล้ว	96
รูปที่ 4-21 การเปลี่ยนรหัสผ่านผู้ใช้งานระบบ	96
รูปที่ 4-22 หน้าจอการเปลี่ยนรหัสผ่านผู้ใช้งานระบบ	97
รูปที่ 4-23 การเปลี่ยนสิทธิของผู้ใช้งานระบบ	97
รูปที่ 4-24 หน้าจอระบบหลังจากการเปลี่ยนสิทธิผู้ใช้งานเรียบร้อยแล้ว	98
รูปที่ 4-25 การจัดการระบบหลัก	98
รูปที่ 4-26 การกำหนดสิทธิการดูเครื่อง	99
รูปที่ 4-27 กำหนดสิทธิการดูเครื่อง	99
รูปที่ 4-28 กำหนดสิทธิเรียบร้อยแล้ว	100
รูปที่ 4-29 แสดงเฉพาะเครื่องที่กำหนด	100
รูปที่ 4-30 การจัดการระบบหลัก	101
รูปที่ 4-31 หน้าจอของการจัดการระบบหลัก	101
รูปที่ 4-32 การดูรายละเอียดระบบ	102
รูปที่ 4-33 รายละเอียดของระบบ	102
รูปที่ 4-34 การจับหน้าจอภาพของระบบ	103
รูปที่ 4-35 ภาพหน้าจอที่จับได้	103
รูปที่ 4-36 การดูคุณสมบัติของระบบ	104
รูปที่ 4-37 คุณสมบัติต่างๆ ของระบบ	104
รูปที่ 4-38 ตัวอย่างรายละเอียดคุณสมบัติ	105
รูปที่ 4-39 การเรียกดูรายงาน โพรเซสเซอร์	105
รูปที่ 4-40 แสดงรายงาน โพรเซสเซอร์	106
รูปที่ 4-41 การเรียกดูรายงานระบบปฏิบัติการ	106
รูปที่ 4-42 แสดงรายงานระบบปฏิบัติการ	107
รูปที่ 4-43 การเรียกดูรายงานซอฟต์แวร์	107
รูปที่ 4-44 แสดงรายงานซอฟต์แวร์	108
รูปที่ 4-45 การจัดการหน้าต่างของระบบ	108

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูปภาพ (ต่อ)

	หน้า
รูปที่ 4-46 การจัดการหน้าต่างแบบคาสเคด	109
รูปที่ 4-47 ผลการจัดการหน้าต่างแบบคาสเคด	109
รูปที่ 4-48 การเลือกการจัดการหน้าต่างตามแนวขวาง	110
รูปที่ 4-49 ผลการจัดการหน้าต่างตามแนวขวาง	110
รูปที่ 4-50 การเลือกการจัดการหน้าต่างตามแนวตั้ง	111
รูปที่ 4-51 ผลการจัดการหน้าต่างตามแนวตั้ง	111



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

### 1.1 แนวคิดและที่มาของปัญหา

แนวความคิดของระบบการจัดการทรัพย์สินเทคโนโลยีสารสนเทศ (IT Asset Management System) นั้นมาจากการที่บริษัทหรือองค์กรต่างๆ มีจำนวนเครื่องคอมพิวเตอร์ในระบบเป็นจำนวนมาก ทำให้เป็นการยากที่จะคอยดูแลและควบคุมการใช้งานโดยบุคคลทำให้เกิดปัญหาต่างๆ ในระบบ อาทิ เช่น ไม่ทราบจำนวนคอมพิวเตอร์ในระบบ จำนวนซอฟต์แวร์ (Software) ความถี่ของการใช้งานในแต่ละเครื่อง เป็นต้น

โดยที่ระบบการจัดการทรัพย์สินเทคโนโลยีสารสนเทศนั้นมีหลักการอยู่ว่า สามารถจัดการเกี่ยวกับส่วนของฮาร์ดแวร์ (Hardware) และซอฟต์แวร์ที่เหมาะสมกับการใช้งาน และยังหาวิธีแก้ปัญหาไปในทิศทางที่เหมาะสมที่สุดสำหรับองค์กร รวมถึงช่วยให้องค์กรสามารถจัดการซอฟต์แวร์และฮาร์ดแวร์เป็นไปอย่างมีประสิทธิภาพ

### 1.2 วัตถุประสงค์

- 1.2.1 เพื่อพัฒนาระบบ โครงข่ายคอมพิวเตอร์ในองค์กรให้มีประสิทธิภาพมากขึ้น
- 1.2.2 เพื่อให้การปฏิบัติงานภายในองค์กรมีระเบียบมากขึ้น
- 1.2.3 เพื่อตรวจสอบการใช้งานในเครื่องคอมพิวเตอร์แต่ละเครื่อง
- 1.2.4 เพื่อให้ข้อมูลภายในเครื่องคอมพิวเตอร์แต่ละเครื่องมีความถูกต้องและสมบูรณ์

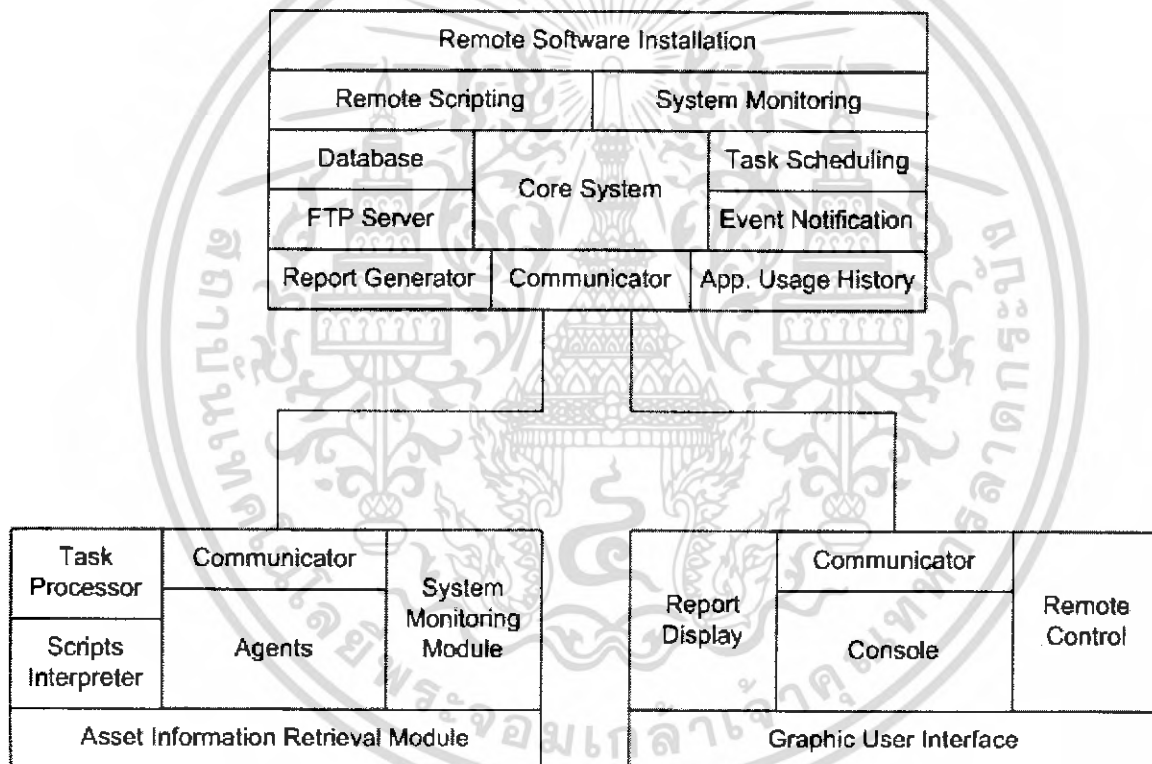
### 1.3 ขอบเขตของโครงการ

- 1.3.1 สามารถทำงานได้บนระบบปฏิบัติการตระกูลวินโดวส์
- 1.3.2 รองรับระบบการจัดกลุ่มคอมพิวเตอร์เป็นกลุ่ม
- 1.3.3 สามารถดูข้อมูลเกี่ยวกับฮาร์ดแวร์ได้ (Motherboard Serial Number, Motherboard Model, Sound Card, Memory Banks, BIOS Version etc.)
- 1.3.4 สามารถดูข้อมูลเกี่ยวกับซอฟต์แวร์ที่ถูกติดตั้งได้
- 1.3.5 สามารถดูข้อมูลทั่วไปของระบบได้ (General System Information) เช่น OS Version, MAC Address, System Path และอื่นๆ
- 1.3.6 สามารถสั่งจับหน้าจอภาพได้
- 1.3.7 สามารถเฝ้าดูประสิทธิภาพของหน่วยประมวลผลกลางได้ (CPU Performance Monitoring)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1.3.8 สามารถเฝ้าดูการใช้งานหน่วยความจำได้ (Memory Usage Monitoring)
- 1.3.9 สามารถเฝ้าดูโปรเซสการทำงานของระบบได้ (Process Monitoring)
- 1.3.10 สามารถเฝ้าดูการทำงานของโปรแกรมที่กำลังทำงานอยู่ได้
- 1.3.11 สามารถสั่งยกเลิกการติดตั้งได้
- 1.3.12 สามารถแสดงรายงานเกี่ยวกับฮาร์ดแวร์ได้
- 1.3.13 สามารถแสดงรายงานเกี่ยวกับซอฟต์แวร์ได้

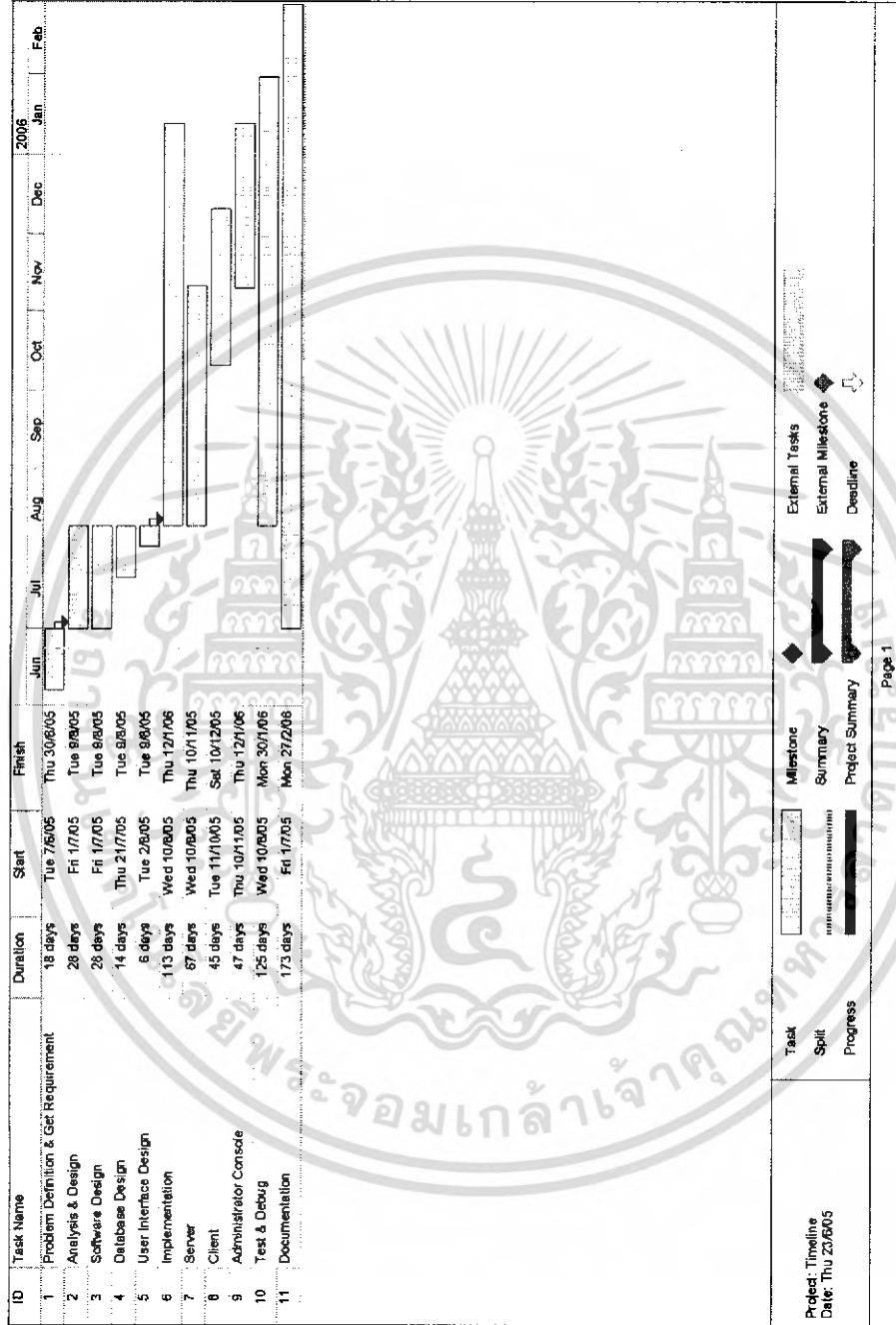
#### 1.4 สถาปัตยกรรมของระบบ



รูปที่ 1-1 สถาปัตยกรรมของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.5 ขั้นตอนการดำเนินงาน



รูปที่ 1-2 ขั้นตอนการดำเนินงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### ทฤษฎีที่เกี่ยวข้อง

#### 2.1 Windows Management Instrumentation: WMI

##### 2.1.1 แนะนำ WMI

Windows Management Instrumentation (WMI) เป็นโครงสร้างพื้นฐานของการจัดการระบบ ซึ่งมีความแน่นอน มีมาตรฐาน มีความยืดหยุ่น และมีตัวประสานงานเชิงวัตถุ WMI จัดเตรียมมาตรฐานการจัดการระบบ ซึ่งอยู่ภายใต้ WMI API และ WMI ยังเป็นพื้นฐานของระบบการจัดการพัฒนาแอปพลิเคชันอีกด้วย และผู้ดูแลระบบสามารถเข้าถึงและจัดการระบบการจัดการข้อมูลได้

WMI สามารถสร้างอุปกรณ์เพื่อใช้ในการบริหารและจัดการระบบข้อมูล ดังนั้นผู้ดูแลระบบหรือผู้จัดการระบบ สามารถดูการทำงานต่างๆ บนระบบได้อย่างละเอียด ตัวอย่างเช่น คุณสามารถพัฒนาแอปพลิเคชัน โดยใช้ WMI ได้ แม้ Web Server จะไม่สามารถทำงานได้ก็ตาม

##### 2.1.2 โครงสร้างของ WMI

โครงสร้างของ WMI ประกอบไปด้วย

- โพรไวเดอร์ (Provider)
- คอนซูเมอร์ (Consumers)
- Common Information Model (CIM) Repository
- Common Information Model Object Manager (CIMOM)

ส่วนประกอบต่างๆ นี้สร้างขึ้นโดยใช้ CIM ซึ่งมีลักษณะเป็นเชิงวัตถุ เป็นอิสระต่อระบบ และแอปพลิเคชัน ซึ่งจะแบ่งออกเป็นชั้นๆ เพื่อให้กำหนดขอบเขตในระบบการจัดการ โดย CIM จะกำหนดขอบเขตผ่านคลาส โดยที่แต่ละคลาสนั้นจะประกอบไปด้วยเมธอดเพื่ออธิบายการทำงาน และคุณสมบัติเพื่ออธิบายข้อมูลบางขอบเขตที่มีอยู่ใน CIM อาทิเช่น แอปพลิเคชัน โครงข่าย เครื่องพิมพ์ และไคร์ฟเวอร์

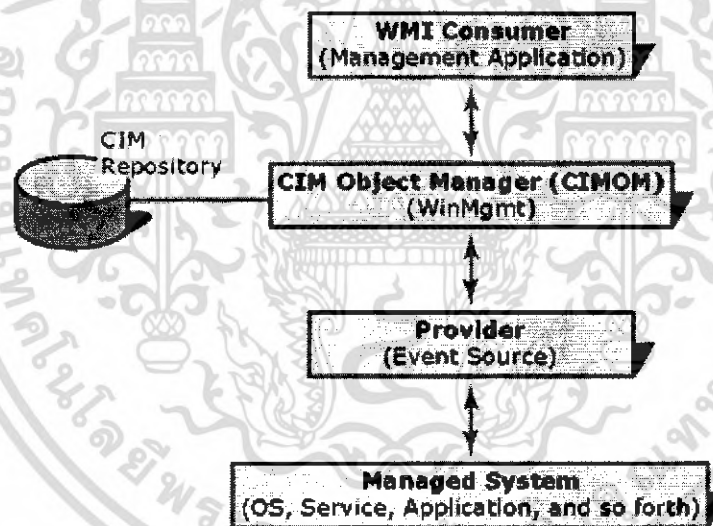
โพรไวเดอร์เป็นตัวกลางระหว่างระบบกับ CIMOM เพื่อการจัดการ (เช่น ระบบปฏิบัติการ บริการ แอปพลิเคชัน ไคร์ฟเวอร์อุปกรณ์) โพรไวเดอร์ เหมือนเป็นเหตุการณ์ที่เกิดขึ้นใน Visual Studio Analyzer มีจุดประสงค์เพื่อลดการจัดการข้อมูลออกจากระบบ โดยใช้ตัวประสานงานแสดงการจัดการแทน ในส่วนต่างๆ ของส่วนติดต่อ CIM โพรไวเดอร์สามารถจำหน่ายข้อมูลไปให้ WMI ตามคำสั่งจากโปรแกรมการจัดการระบบ และสร้างคำเตือนเหตุการณ์ขึ้น โพรไวเดอร์ยังวางแผนการจัดการข้อมูลและส่วนติดต่อผ่านคลาสวัตถุ ซึ่งจะถูกกำหนดขึ้นและจัดเก็บในส่วนของ CIM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอนซูเมอร์ เป็นแอปพลิเคชันการจัดการระบบ เช่น Microsoft Systems Management Server (SMS) หรือแอปพลิเคชันอื่น ๆ หรือสคริปต์ โดยคอนซูเมอร์ต้องการรู้แค่คลาสของออบเจกต์ที่ต้องการ ข้อมูลต่างๆ ว่าข้อมูลเหล่านั้นอยู่ที่ไหน และมีวิธีการเก็บข้อมูลอย่างไร และคอนซูเมอร์ยังสามารถรับข้อมูลเกี่ยวกับคอมพิวเตอร์, ระบบปฏิบัติการ, แอปพลิเคชัน และอุปกรณ์โดยจะได้รับข้อมูลผ่าน โปรโตคอล

CIM เป็นฐานข้อมูลออบเจกต์ที่คอยกำหนดออบเจกต์ เช่นเดียวกับคลาส Static ที่ใช้ในการเข้าถึง และจัดการระบบการจัดการข้อมูลที่ถูกเก็บไว้ CIM จะถูกจัดการโดย CIMOM ซึ่งทำหน้าที่เป็น ตัวกลางสำหรับการร้องขอออบเจกต์ ซึ่ง CIMOM จะทำหน้าที่หาว่าคลาสที่ถูกเรียกขึ้นมาอยู่ในโพรไวเดอร์ไหน ซึ่ง CIMOM และ CIM จะถูกแสดงโดยการบริหารระบบ ที่เรียกว่า WinMgmt และเป็นการเข้าถึง ส่วนหนึ่งของส่วนติดต่อ CIM

รูปภาพข้างล่างนี้แสดงถึงการทำงานร่วมกันของคอมโพเนนต์ต่างๆ



รูปที่ 2-1 การทำงานร่วมกันของคอมโพเนนต์ต่างๆ

เมื่อ CIMOM ได้รับการร้องขอจากคอนซูเมอร์ของ WMI แล้ว CIMOM จะแปลงการร้องขอ แล้วระบุว่าโพรไวเดอร์ไหนมีข้อมูลอยู่ แล้วจะส่งข้อมูลนั้นกลับไปยังคอนซูเมอร์ ซึ่งคอนซูเมอร์จะร้องขอเฉพาะข้อมูลที่ต้องการเท่านั้น ซึ่ง CIMOM จะไม่รู้ Source ของข้อมูลหรือรายละเอียดต่าง ๆ ของวิธีการเอาข้อมูลออกมาจาก API

## 2.2 Microsoft .NET

### 2.2.1 แนะนำ Microsoft .Net

Microsoft .NET (อ่านว่า คีอทเนต) เป็นซอฟต์แวร์ที่ไมโครซอฟท์พัฒนาขึ้นมา โดย .NET ตัวนี้ไม่ได้เกี่ยวข้องกับโดเมนเนมของเว็บใดๆทั้งสิ้น แต่ .NET ตัวนี้ หมายถึง การนำเอาอุปกรณ์ทุกอย่างบนโลกมาเชื่อมโยงต่อกันเหมือนดาข่าย (net = ดาข่าย) ดังนั้น Microsoft .Net จึงมีหน้าที่ทำให้อุปกรณ์ต่างๆ ที่ถูกออกแบบมาต่างหากมาเชื่อมโยงกัน ดังนั้นไมโครซอฟท์จึงคิดค้นสิ่งที่เป็นมาตรฐานขึ้นมาเพื่อให้อุปกรณ์ทุกๆ ชนิดทั่วโลกติดต่อสื่อสารกันได้อย่างรู้เรื่อง

### 2.2.2 Microsoft .Net Framework

อย่างที่ได้อธิบายไปแล้วในตอนต้นว่าไมโครซอฟท์ต้องการที่จะสร้างอะไรที่เป็นมาตรฐานขึ้นมาเพื่อให้ทุกสิ่งทุกอย่างสามารถติดต่อสื่อสารกันได้หมด โดยคิดค้นระบบซึ่งจะให้มันเป็นระบบมาตรฐาน นั่นก็คือ .NET Framework ซึ่งระบบนี้ไม่ใช่ระบบปฏิบัติการ (OS) แต่เปรียบเสมือนโปรแกรมหนึ่งที่สามารถสร้างสถานะแวดล้อมหนึ่ง ซึ่งสามารถทำงานในระบบ .NET นี้ได้

โดย .NET Framework นั้นมี ส่วนประกอบภายในแบ่งออกเป็น 3 ชั้นใหญ่ๆคือ

**1. Programming Language:** เป็นรูปแบบของภาษาที่ออกแบบมาเพื่อให้สามารถทำงานในสถานะที่เป็น .NET ได้โดยในปัจจุบัน ภาษาหลักๆที่จะใช้พัฒนามบน .NET นี้ 3 ภาษา

- C# เป็นภาษาใหม่ที่ไม่โครซอฟท์พัฒนามาจาก C++ กับ JAVA เป็นหลัก
- VB.NET เป็นภาษาที่พัฒนามาจาก Visual Basic ในเวอร์ชัน 6.0
- C++ มีทั้งแบบ Managed Code และ Unmanaged Code

**2. Base Classes Library:** ไบเบรารี (Library) นั้นเปรียบเสมือนชุดคำสั่งสำเร็จรูปย่อยๆ ที่เพิ่มเข้ามา ซึ่งส่วนใหญ่จะเป็นชุดคำสั่งที่ต้องใช้งานอยู่เป็นประจำ ดังนั้นจึงมีผู้คิดค้นเพื่ออำนวยความสะดวกในการเขียนโปรแกรม ซึ่งไลเบรารีในภาษาต่างๆ ส่วนใหญ่จะอยู่ในรูปแบบไฟล์ Include แต่ถ้าเป็น ASP สิ่งที่เป็นไลเบรารีก็คือ คอมโพเนนต์ต่างๆ นั่นเอง ซึ่งภายในระบบ .NET จะสร้างสิ่งที่เรียกว่าเป็นไลเบรารีพื้นฐานขึ้น ทำให้ไม่ว่าจะใช้ภาษาใดในการพัฒนาโปรแกรมก็สามารถที่จะเรียกใช้ไลเบรารีที่เป็นตัวเดียวกันได้หมด

**3. Common Language Runtime (CLR):** แทบจะเป็นสิ่งสำคัญที่สุดของระบบ .NET นี้ก็ว่าได้ เพราะ CLR ที่ว่านี้มีหน้าที่ทำให้โปรแกรมที่เขียนขึ้นมาด้วยภาษาต่างๆกัน กลายเป็นภาษารูปแบบมาตรฐานเดียวกัน ทั้งหมด เราเรียกภาษาที่ว่านี้ว่า Intermediate Language (IL) ซึ่งเมื่อต้องการที่จะรันโปรแกรมใด CLR ที่ว่านี้จะตรวจสอบเครื่องที่รันว่ามีสถานะแวดล้อมการทำงานเช่นใดหลังจากนั้นก็จะมีคอมไพล์เป็นโปรแกรมที่เหมาะสมต่อการทำงานของเครื่องนั้น ทำให้เราสามารถใช้งานโปรแกรมต่างๆ ได้อย่างมีประสิทธิภาพสูงสุดในแต่ละเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.2.3 ประโยชน์และข้อดีของ .NET Framework

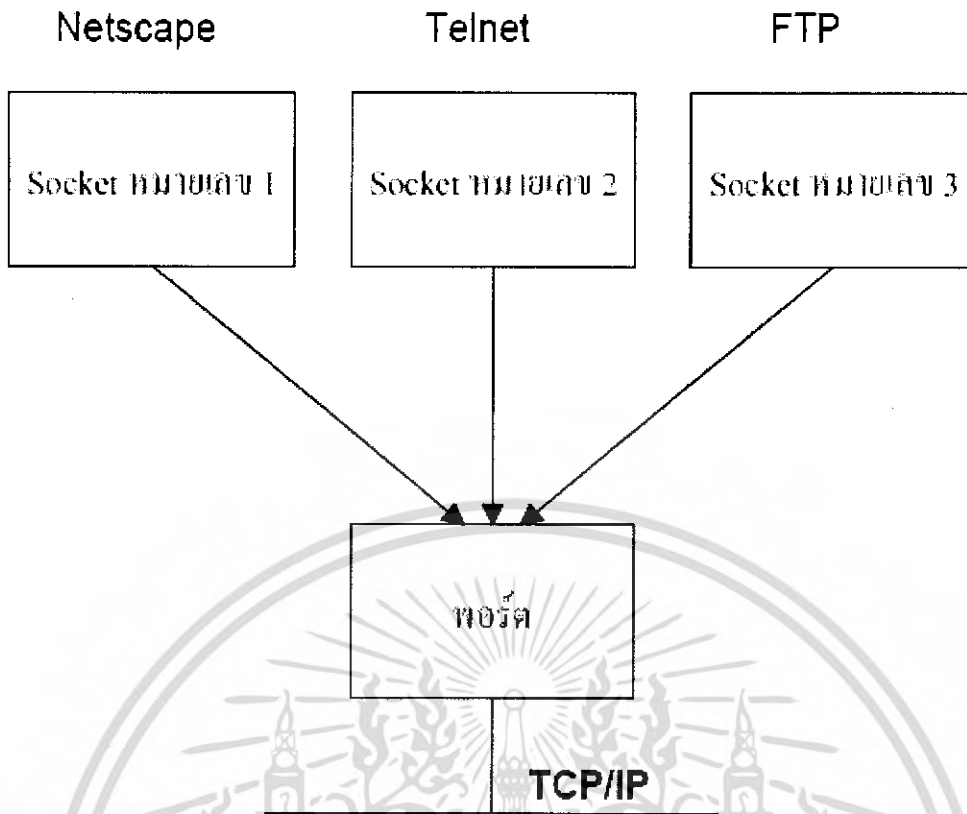
ประโยชน์และข้อดีของ .NET Framework นั้นมีดังนี้

1. **เป็นระบบที่มีไลบรารีที่เป็นมาตรฐานเดียวกัน:** เนื่องจากมีไลบรารีที่เป็นมาตรฐานเดียวกันทั้งหมดทำให้เราไม่ต้องกังวล ว่าภาษาที่ใช้เขียนนั้นมีไลบรารีตัวนั้นตัวนี้หรือไม่ รวมทั้งไม่ต้องคอยกังวลว่าถ้าใช้ไลบรารีของภาษาหนึ่งแล้วอีกภาษาหนึ่งจะไม่มีไลบรารีตัวนั้น
2. **ไม่ขึ้นกับระบบปฏิบัติการ (OS):** เนื่องจากระบบปฏิบัติการ ที่แต่ละบุคคลหรือองค์กรใช้นั้นย่อมไม่เหมือนกัน แต่ภายใน .NET Framework จะไม่มีปัญหานี้ของเพียงแค่มีระบบ .NET Framework ก็จะทำให้สามารถใช้งานโปรแกรมต่างๆได้ ซึ่งเป็นข้อดีตรงที่เราจะสามารถใช้โปรแกรมต่างๆ ได้บนทุกระบบปฏิบัติการที่สนับสนุน .NET Framework
3. **ใช้ในการพัฒนาได้ทุกภาษา:** ทำให้เราไม่ต้องคอยมาศึกษาภาษาใหม่ๆ เมื่อต้องการสร้างโปรแกรมในแต่ละครั้ง นอกจากนั้นเรายังสามารถเลือก ใช้ภาษาที่เราถนัดที่สุดในการพัฒนาโปรแกรมต่างๆ ได้ด้วย
4. **มีการควบคุมสิ่งแวดล้อมในการทำงานเป็นอย่างดี :** เนื่องจากเป็นระบบที่เป็นมาตรฐานทำให้การควบคุมจัดสรรระบบต่างๆ ทำได้ง่ายขึ้น ไม่ว่าจะเป็นการจัดสรรหน่วยความจำด้านการใช้งานเครื่อง ก็มีความรวดเร็วมากขึ้นตลอด โอกาสที่เครื่องจะแฮงค์ก็เป็นอย่างดี
5. **ความปลอดภัยที่มีมากขึ้น :** .NET Framework สามารถกำหนดสิทธิการใช้งานหรือ permission ของผู้ใช้งานได้มากขึ้นทำให้สามารถกำหนดว่า จะให้โปรแกรมในส่วนใดใช้งานได้หรือไม่ได้ แล้วแต่เฉพาะบุคคล

## 2.3 Windows Socket

### 2.3.1 แนะนำ Windows Socket

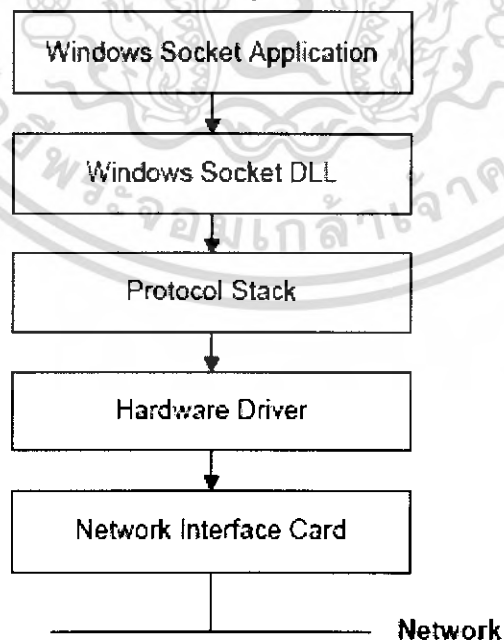
การติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์บนระบบอินเทอร์เน็ต เช่นการ Telnet เข้าไปในระบบ UNIX ระบบจะต้องสามารถรองรับการทำงานแบบ Multi-User ได้ นั่นหมายความว่า ระบบจะต้องสามารถมีการติดต่อสื่อสารกับผู้ใช้งาน (User) ได้พร้อมๆ กัน หรือเมื่อมีการใช้งานโปรแกรมที่เกี่ยวข้องกับอินเทอร์เน็ต เช่น Netscape, Telnet, FTP ซึ่งโปรแกรมต่างๆ เหล่านี้จะต้องทำงานแยกกันโดยอิสระ ดังนั้นจึงได้เกิด Winsock ขึ้นมา นั่นก็คือ Winsock สามารถที่จะสร้างช่องทางสื่อสารขึ้นมาได้หลายๆ ช่องทาง (ขึ้นอยู่กับเวอร์ชันของ Winsock) และแต่ละช่องทางสื่อสารสามารถที่จะส่งข้อมูลได้โดยไม่ขึ้นกับช่องทางสื่อสารอื่นๆ ดังรูป



รูปที่ 2-2 ช่องทางการสื่อสารของ Winsock

### 2.3.2 ระดับของ Winsock ในส่วนของ TCP/IP

Winsock จะทำหน้าที่เป็นตัวกลางในการรับส่งข้อมูลระหว่างแอปพลิเคชันกับ TCP/IP จากนั้น TCP/IP จึงส่งข้อมูลลงไปบนระบบอินเทอร์เน็ต ดังรูป

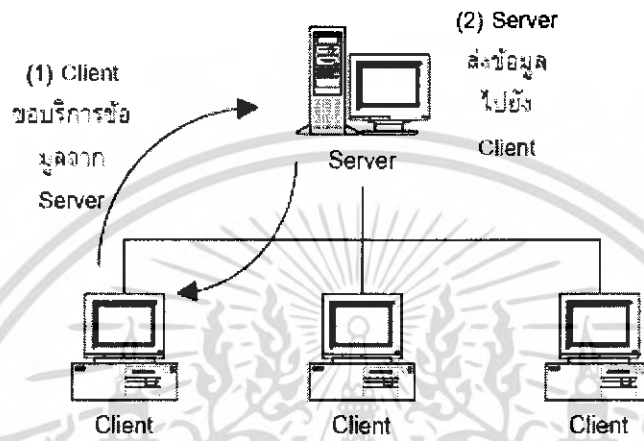


รูปที่ 2-3 การส่งข้อมูลของ Winsock

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

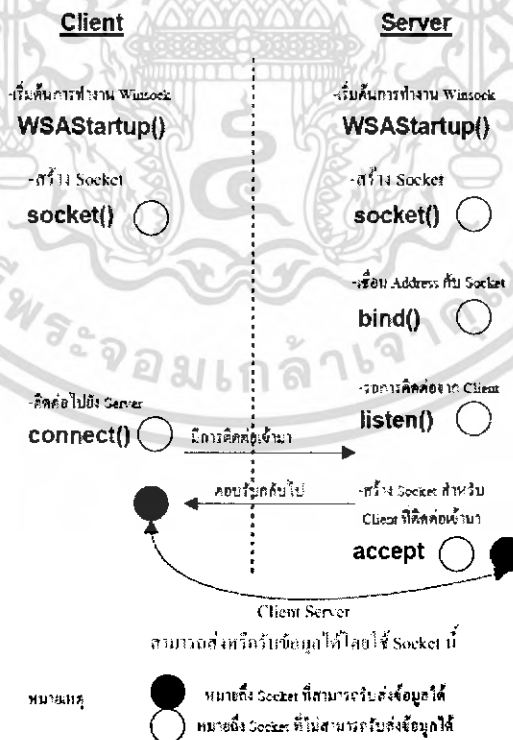
### 2.3.3 Client/Server

บนระบบอินเทอร์เน็ตมีการให้บริการต่างๆมากมาย เครื่องที่ให้บริการ จะเรียกว่าเซิร์ฟเวอร์ (Server) เครื่องที่ทำหน้าที่ขอบริการ จะเรียกว่าไคลเอนท์ (Client) บริการอาจจะหมายถึง การรับส่งข้อความ ไฟล์รูปภาพ ไฟล์เสียง ไฟล์โปรแกรม หรือ เป็นการสั่งให้เซิร์ฟเวอร์ทำงาน หรือประมวลผล ในสิ่งที่ต้องการ ก็ได้ เช่น WEB Server หมายถึงเซิร์ฟเวอร์ที่ให้บริการรับส่งข้อมูลเกี่ยวกับโปรโตคอล HTTP หรือ HOMEPAGE



รูปที่ 2-4 การทำงานบนระบบ Client/Server

### 2.3.4 การสร้าง Client/Server โดยใช้ Winsock



รูปที่ 2-5 การสร้าง Client/Server โดยใช้ฟังก์ชันของ Winsock

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.5 การทำงานของเซิร์ฟเวอร์

เริ่มต้นการทำงานของ Winsock โดยใช้ฟังก์ชัน WSAStartup() จากนั้นทำการสร้าง Socket ขึ้นมาโดยใช้ฟังก์ชัน socket() เชื่อมต่อ Address เข้ากับ Socket ที่สร้างขึ้นมาโดยใช้ฟังก์ชัน bind() และรอการเชื่อมต่อจากไคลเอนท์โดยใช้ฟังก์ชัน listen()

### 2.3.6 การทำงานของ Client

เริ่มต้นการทำงานของ Winsock โดยใช้ฟังก์ชัน WSAStartup() จากนั้นทำการสร้าง Socket ขึ้นมาโดยใช้ฟังก์ชัน socket() และเชื่อมต่อไปยังเซิร์ฟเวอร์โดยใช้ฟังก์ชัน connect()

หากมีไคลเอนท์ติดต่อเข้ามา เซิร์ฟเวอร์จะรับรู้และตอบกลับไปที่ไคลเอนท์และสร้าง Socket ขึ้นมาใหม่เพื่อใช้ในการสื่อสารกับไคลเอนท์นั้นๆ ที่ติดต่อเข้ามาโดยใช้ฟังก์ชัน accept() เมื่อถึงขั้นตอนนี้ หมายความว่าไคลเอนท์และเซิร์ฟเวอร์สามารถที่จะสื่อสารกันได้แล้ว โดยใช้ฟังก์ชัน send() หรือ recv()

## 2.4 Windows Services

วินโดวส์เซอร์วิส เป็นโปรแกรมที่จะเริ่มต้นโดยอัตโนมัติเมื่อเริ่มการทำงานของระบบปฏิบัติการ โดยไม่จำเป็นต้องมีการริจิสเตอร์เข้าสู่เครื่อง

### 2.4.1 แนะนำวินโดวส์เซอร์วิส

วินโดวส์เซอร์วิส เป็นแอปพลิเคชันที่สามารถเริ่มต้นโดยอัตโนมัติ เมื่อระบบปฏิบัติการเริ่มทำงาน ซึ่งสามารถทำงานได้โดยปราศจากผู้ใช้ที่ริจิสเตอร์เข้าสู่ระบบ และยังสามารถกำหนดการทำงานของวินโดวส์ตามความสำคัญของผู้ใช้งานที่ริจิสเตอร์เข้ามาในระบบ

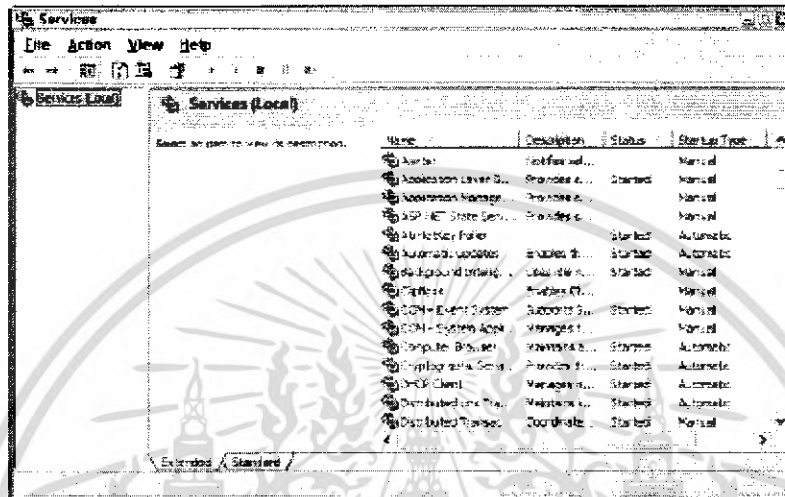
วินโดวส์เซอร์วิสไม่รองรับระบบปฏิบัติการ Windows 95, 98 และ ME แต่จะรองรับระบบปฏิบัติการ Windows NT 4, Windows 2000, Windows XP และ Windows Server 2003

### ตัวอย่างของวินโดวส์เซอร์วิส

- เซอร์วิสของ TCP/IP เป็นโปรแกรมเซอร์วิสที่โอสที่มีขนาดเซิร์ฟเวอร์ของ TCP/IP เล็ก เช่น Echo, Day-time, Quote และอื่นๆ
- เซอร์วิสที่ใช้กันทั่วโลก คือ Internet Information Server (IIS)
- Event Log เป็นเซอร์วิสที่ใช้เข้าข้อความในระบบ
- Microsoft Search เป็นเซอร์วิสที่สร้างดัชนีของข้อมูลบนดิสก์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยใช้ Services Administration Tool ดังแสดงในรูปภาพ 2-6 สำหรับคูเซอรวีตทั้งหมดที่มีอยู่ในระบบใน Windows 2000 Server โปรแกรมนี้สามารถเข้าถึงโดยเลือก Start → Programs → Administrative → Tools → Services แต่ใน Windows 2000 Professional และ Windows XP โปรแกรมนี้สามารถเข้าถึงผ่าน Settings → ControlPanel → AdministrativeTools → Services



รูปที่ 2-6 Services Administration Tool

## 2.4.2 โครงสร้างของวินโดวเซอร์วิส

ในวินโดวเซอร์วิสจะประกอบไปด้วย 3 ส่วนหลักๆ คือ

- A service program
- A service control program
- A service configuration program

โดยที่ Service Program มีหน้าที่หาข้อมูลที่เราต้องการ ซึ่งจะควบคุมไปกับ Service Control Program ซึ่งมันจะร้องขอไปที่เซอรวีต ยกตัวอย่างเช่น Start, Stop, Pause และ Continue แล้วยังมี Service Configuration Program เป็นเซอรวีตที่สามารถติดตั้งได้ โดยจะคัดลอกระบบไฟล์ แล้วเขียนทับลงในรีจิสทรี และก็จะทำงานเหมือนเป็นเซอรวีต ขณะที่ .NET คอมโพเนนท์สามารถติดตั้งได้ด้วย xcopy โดยที่ไม่ต้องการรีจิสทรี แต่การติดตั้งสำหรับเซอรวีตนั้นต้องการ Registry Configuration ซึ่งจะนำมาใช้ในการเปลี่ยนแปลงเซอรวีตได้ในภายหลัง

### 2.4.2.1 Service Program

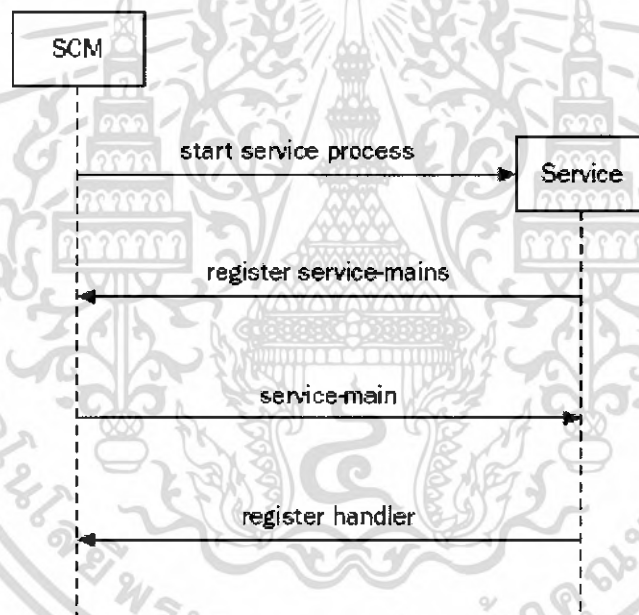
Service Program ประกอบไปด้วย 3 ส่วนคือ

- A main function
- A service-main function
- A handler

ก่อนที่จะรู้จักกับส่วนประกอบต่างๆเหล่านี้ จะต้องศึกษาเกี่ยวกับ Service Control Manager (SCM) ก่อน

### 2.4.2.2 Service Control Manager

SCM คือส่วนที่ระบบปฏิบัติการติดต่อกับเซอร์วิส ตามรูปภาพ 2-7 เป็นแผนภาพ UML อธิบายว่าการติดต่อสื่อสารอย่างไร



รูปที่ 2-7 การทำงานของ SCM

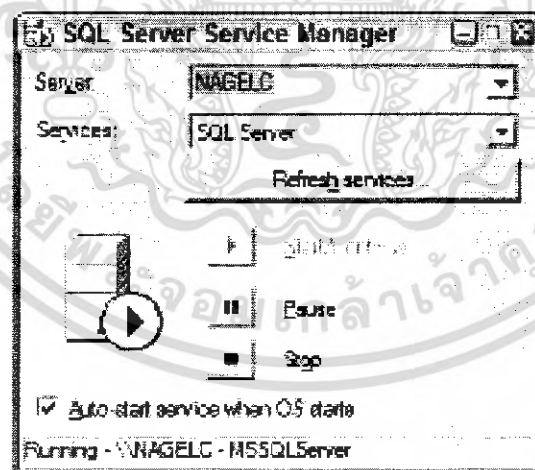
ขณะบูทแต่ละ โพรเซสสำหรับแต่ละเซอร์วิสจะกำหนดให้เริ่มต้นโดยอัตโนมัติ และ ดังนั้นหน้าที่หลักของ โพรเซสนี้คือได้รับ Called เซอร์วิสมีหน้าที่ในการริจิสเตอร์ฟังก์ชันหลัก ของแต่ละเซอร์วิส จากภาพเอ็นทรี่พอยท์สำหรับฟังก์ชันหลักของเซอร์วิสต้องถูกริจิสเตอร์ด้วย SCM

### 2.4.2.3 Main function, service-main, and handlers

Main Function ของเซอร์วิสอาจจะมีมากกว่าหนึ่ง Service Main Function โดยที่แต่ละเซอร์วิสก็ต้องมี Service Main Function ของแต่ละเซอร์วิส และในหนึ่งโปรแกรมอาจจะมีเซอร์วิสมากกว่าหนึ่งเซอร์วิสตัวอย่างเช่น <windows>\system32\services.exe เป็นเซอร์วิสที่ประกอบด้วย Alerter, Application Management, Computer Browser และไคลเอนท์ DHCP โดย SCM มีหน้าที่เรียก Service Main Function ของแต่ละเซอร์วิสให้ทำงาน ซึ่งใน Service Main Function จะเป็นหน้าที่ต่างๆ ของเซอร์วิส หน้าที่ความสำคัญของ Service Main Function คือ ลงทะเบียน Handler ด้วย SCM

Handler คือส่วนที่สามของเซอร์วิส โดย Handler ต้องตอบสนองเหตุการณ์จาก SCM ซึ่ง Service สามารถสั่งให้ Stopped, Suspended และ Resumed ได้ โดย Handler ต้องขั้นตอนต่างๆ ดังนี้

เมื่อ Handler มีลงทะเบียนกับ SCM เรียบร้อยแล้ว Service Control Program จะส่ง Requests ไปยัง SCM ให้ Stop, Suspend และ Resume โดยที่ Service Control program จะเป็นอิสระจาก SCM และจะทำงานด้วยตัวเอง ในระบบปฏิบัติการมี Service Control Program อยู่จำนวนมาก หนึ่งในนั้นคือ MMC เป็นเซอร์วิสที่จะแสดงสิ่งที่เราเห็นล่าสุด และยังสามารถเขียน Service Control Program ขึ้นมาเองก็ได้ ตัวอย่างเช่น SQL Server Service Manager ดังแสดงในรูปภาพ 2-8



รูปที่ 2-8 ตัวอย่าง Service Control Program

### 2.4.3 Service Control Program

มีหน้าที่ในการควบคุมเซอร์วิส สำหรับ Stopping, Suspending และ Resuming ของเซอร์วิส โดยส่ง รหัสควบคุม (Control Code) ไปที่เซอร์วิส แล้ว Handler ก็จะตอบสนองเหตุการณ์นั้น และยังสามารถถามสถานะที่แท้จริงของเซอร์วิส เพื่อที่จะกำหนด Handler เพื่อตอบสนองรหัสควบคุมได้

### 2.4.4 Service Configuration Program

ไม่สามารถใช้ xcopy ในการติดตั้งเซอร์วิสได้ เพราะว่าการบริการจำเป็นต้องแก้ไขใน Registry ซึ่งสามารถตั้งประเภทการเริ่มต้นเป็นแบบอัตโนมัติ, ด้วยมือ หรือระงับ ตัวอย่างเช่น เซอร์วิสนั้น จำเป็นต้องเริ่มต้นก่อนที่อีกเซอร์วิสจะเริ่มได้ การกำหนดค่าเหล่านี้ทั้งหมดจะถูกกำหนดใน Service Configuration Program ในการติดตั้ง โปรแกรมนั้นสามารถใช้ Service Configuration Program ในการแก้ไขเซอร์วิส เว้นแต่โปรแกรมนี้จะสามารถใช้เปลี่ยน Service Configuration Parameters ได้

## 2.5 .NET Remoting

### 2.5.1 .Net Remoting คืออะไร

แอปพลิเคชันบางตัวไม่สามารถทำงานตัวเดียวบนระบบที่แอปพลิเคชันนั้นทำงานอยู่ แต่มันจะใช้เทคโนโลยีการติดต่อสื่อสารระหว่างโครงข่าย เพื่อทำการติดต่อไปยังรีโมทเซิร์ฟเวอร์ (Remote Server) นี่ก็คือชื่อของ .Net Remoting

สำหรับแอปพลิเคชันที่ใช้ในการติดต่อระหว่างไคลเอนท์ (Client) และเซิร์ฟเวอร์ (Server) ซึ่งมีหลายเทคโนโลยีที่นำมาใช้ เช่น ควบคุมแอปพลิเคชันโดยใช้ Socket หรือจะใช้คลาสจาก System.Net ซึ่งจะช่วยให้สะดวกต่อการควบคุมด้วยโปรโตคอล (Protocol), หมายเลขไอพี (IP addresses), และหมายเลขพอร์ต (Port Numbers) เทคโนโลยีเหล่านี้นำมาใช้เพื่อส่งข้อมูลข้ามโครงข่าย ข้อมูลที่ส่งไปโดยโปรโตคอลนั้นจะถูกแปลงที่เซิร์ฟเวอร์ ดังนั้นเซิร์ฟเวอร์จะรู้ว่าใช้วิธีไหนในการจัดการ

นอกเหนือจากการส่งข้อมูลแล้ว ยังสามารถจัดการข้ามเน็ตเวิร์ค (Network) โดยใช้เซิร์ฟเวอร์แอปพลิเคชัน นั่นก็คือ XML Web Service และ .NET Remoting โดยที่ XML Web Service จะใช้โปรโตคอล SOAP เพื่อให้มั่นใจว่าจะสามารถทำงานร่วมกันได้ระหว่างแพลตฟอร์ม (Platform) ที่ต่างกัน ซึ่งต่างกับ .NET Remoting

เป้าหมายของ .NET Remoting นั้นสามารถดูได้จากแอปพลิเคชันและโปรโตคอลที่รองรับ หรือจะดูที่ค่าของ CLR Object Remoting

### 2.5.2 ชนิดของแอปพลิเคชันและโปรโตคอล

จุดเด่นหลักของ .NET Remoting นั้นคือ .NET Remoting นั้นสามารถใช้แอปพลิเคชันได้ทุกประเภทบนทุกเส้นทาง ซึ่งแสดงให้เห็นว่า .NET Remoting มีโครงสร้างที่ยืดหยุ่น

การใช้ SOAP และ HTTP ร่วมกันนั้นเป็นอีกวิธีหนึ่งสำหรับการรีโมท ช่องทางการส่งสามารถเปลี่ยนแปลงและแทนที่ได้ โดยที่ .NET 1.1 ช่องสัญญาณของ HTTP และ TCP จะแสดงโดยคลาส HttpChannel และ TcpChannel โดยสามารถสร้างช่องทางการส่งที่ใช้ UDP, IPX, SMTP, Shared Memory หรือ Message Queuing ขึ้นอยู่กับผู้ใช้งาน

.NET Remoting ไม่ได้จำกัดว่าจะต้องใช้บนเซิร์ฟเวอร์ในทุกๆ แอปพลิเคชัน .NET Remoting สามารถใช้ได้ทุกที่ โดยไม่จำเป็นต้องสร้างคอนโซล (Console) หรือ วินโดวส์แอปพลิเคชัน, วินโดวส์เซอร์วิส หรือ COM+ คอมโพเนนต์ต่างๆ .NET Remoting เป็นเทคโนโลยีที่เหมาะสมสำหรับการสื่อสารแบบ Peer – to – Peer

### 2.5.3 CLR Object Remoting

CLR Object Remoting เป็นส่วนประกอบสำคัญของ .NET Remoting Constructor, Delegate , Interface Method, Properties และ Field สามารถใช้กับการรีโมทได้ .NET Remoting ขยาย CLR Object Remoting ข้ามโครงข่าย CLR Object Remoting ประกอบด้วย Activation, Distributed Identities, Lifetime และ Call Context

นี่เป็นข้อแตกต่างจาก XML Web Service เพราะใน XML Web Service นั้นไคลเอนต์ไม่จำเป็นต้องรู้จัก Object Type ของเซิร์ฟเวอร์ ซึ่งต่างจาก .NET Remoting

### 2.5.4 .NET Remoting Overview

.NET Remoting ใช้สำหรับการเข้าถึงออบเจกต์ในแอปพลิเคชัน โดเมนอื่นๆ และยังสามารถทำงานได้ไม่ว่าออบเจกต์จะอยู่ในโปรเซสเดียวกันหรือจะต่างโปรเซสกัน หรือจะอยู่ต่างระบบกันก็ตาม

Remote Assembly ทำงานในแอปพลิเคชัน โดเมนหรือเป็นส่วนหนึ่งของรีโมทแอปพลิเคชัน ถ้า Assembly เป็นส่วนหนึ่งของของรีโมทแอปพลิเคชัน แล้วไคลเอนต์จะได้รับพร็อกซีสำหรับการติดต่อแทนออบเจกต์ที่แท้จริง ซึ่งพร็อกซีเปรียบเป็นตัวแทนของรีโมทออบเจกต์ในไคลเอนต์โปรเซส ใช้ในการเรียกเมธอด เมื่อไคลเอนต์เรียกเมธอดในพร็อกซีแล้วพร็อกซีจะส่งข้อความเข้าไปในช่องสัญญาณผ่านรีโมทออบเจกต์

.NET แอปพลิเคชัน ทำงานอยู่ภายในแอปพลิเคชัน โดเมน ซึ่งโดเมนนี้จะพบได้ในซิปโปรเซสของโปรเซสนั้นๆ โดยปกติแล้วโปรเซสแต่ละโปรเซสจะทำงานแยกกัน โดยแอปพลิเคชันที่รันอยู่ในแต่ละโปรเซสจะไม่สามารถเข้าถึงและทำลายหน่วยความจำของโปรเซสอื่นได้ สำหรับแอปพลิเคชันที่ใช้

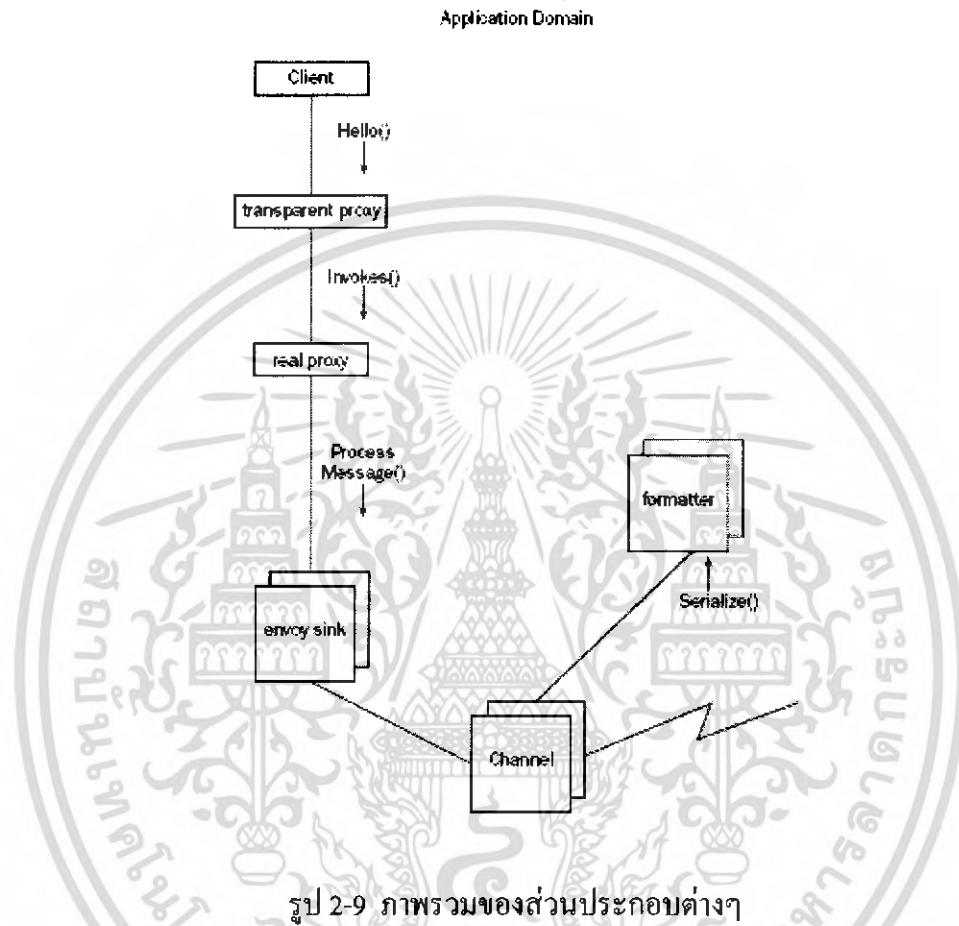
ติดต่อระหว่างกันจำเป็นสำหรับการติดต่อข้ามโปรเซส โดยใน .NET นั้นแอปพลิเคชันโดเมนจะอยู่ในส่วนที่ปลอดภัยภายในโปรเซส เพราะว่า MSIL code เป็นโค้ดที่ปลอดภัยและสามารถพิสูจน์ได้ แอปพลิเคชันที่ต่างกันสามารถรันบนโปรเซสเดียวกันได้แต่ต่างแอปพลิเคชันโดเมนกัน ออบเจกต์ในแอปพลิเคชันโดเมนเดียวกันจะตอบโต้กันได้ ส่วนพรีอ็อกซ์ที่ต้องการคำสั่งในการเข้าถึงออบเจกต์ที่อยู่ต่างแอปพลิเคชันโดเมนกัน

### รายละเอียดของส่วนประกอบต่างๆ ของโครงสร้าง

- **Remote Object** เป็นออบเจกต์ที่ทำงานบนเซิร์ฟเวอร์ โคลเอนที่ไม่สามารถเรียกเมธอดบนออบเจกต์ได้โดยตรง แต่จะใช้พรีอ็อกซ์แทน โดย .NET นั้นสามารถแฮกิริโมทออบเจกต์จากไกลออบเจกต์ได้ กลาสทุกกลาสนั้นได้มาจาก MarshalByRefObject ซึ่งจะไม้ออกมาจากแอปพลิเคชันโดเมนของมันเอง โคลเอนที่สามารถเรียกเมธอดของรีโมทออบเจกต์ผ่านพรีอ็อกซ์ได้
- **Channel** ใช้สำหรับติดต่อระหว่างโคลเอนกับเซิร์ฟเวอร์ ซึ่ง .NET Framework 1.1 เสนอแชนเนลอยู่ 2 ประเภทคือผ่าน TCP หรือ HTTP หรือจะสร้างแชนเนลสำหรับการติดต่อโดยใช้โปรโตคอลตัวอื่นก็ได้
- **Message** ถูกส่งเข้าไปในแชนเนล ข้อความนั้นถูกสร้างขึ้นมาเพื่อใช้ในการติดต่อสื่อสารระหว่างโคลเอนกับเซิร์ฟเวอร์ โดยข้อความเหล่านี้จะเก็บข้อมูลเกี่ยวกับรีโมทออบเจกต์ รวมถึงชื่อเมธอดที่ถูกเรียก และการโต้ตอบต่างๆ
- **Formatter** จะเป็นตัวกำหนดว่าข้อความจะถูกส่งเข้าไปในแชนเนลอย่างไร ใน .NET Framework 1.1 จะมี SOAP และ Binary Formatter โดย SOAP Formatter จะใช้ติดต่อกับเว็บเซอร์วิสที่ไม่ขึ้นอยู่กับ .NET Framework ส่วน Binary Formatter จะทำงานได้เร็วและมีประสิทธิภาพกว่าในสภาพแวดล้อมอินทราเน็ต และยังสามารถสร้าง Formatter ขึ้นมาเองก็ได้
- **Formatter provider** ใช้ร่วมกับ Formatter เกี่ยวกับแชนเนล การสร้างแชนเนลสามารถกำหนดได้ว่าจะใช้ Format provider และกำหนด Formatter สำหรับใช้ในการส่งข้อมูลผ่านแชนเนล
- โคลเอนที่จะเรียกเมธอดบนพรีอ็อกซ์แทนที่จะใช้รีโมทออบเจกต์ มีพรีอ็อกซ์อยู่ 2 ประเภท คือ Transparent Proxy และ Real Proxy ในโคลเอนที่นั้น Transparent Proxy ทำหน้าที่เหมือนรีโมทออบเจกต์ และบน Transparent Proxy นั้นโคลเอนที่จะเรียกเมธอดซึ่งรองรับโดยรีโมทออบเจกต์ ซึ่ง Transparent Proxy จะเรียกเมธอด Invoke() ใน Real Proxy แล้วเมธอด Invoke() จะทำการส่งข้อความเข้าไปในแชนเนล
- **Message Sink** เป็นออบเจกต์ Interceptor ซึ่งใช้ทั้งฝั่งโคลเอนและเซิร์ฟเวอร์ Sink จะทำงานร่วมกับแชนเนล ดังนั้น Sink สามารถทำการ Interception ก่อนที่ข้อความจะเข้าไปในแชนเนล ขึ้นอยู่กับว่า Sink จะถูกใช้ที่ไหน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ไคลเอนท์จะใช้ Activator ในการสร้างรีโมทออบเจกต์บนเซิร์ฟเวอร์หรือจะเรียกพรีอ็อบเจกต์ของ Server-activated Object
- **RemotingConfiguration** เป็นคลาสที่ใช้ประโยชน์ในการรีโมทระหว่างเซิร์ฟเวอร์กับไคลเอนท์
- **ChannelServices** เป็นคลาสที่ใช้ในการบันทึกแชนเนลแล้วก็ส่งข้อความไปให้



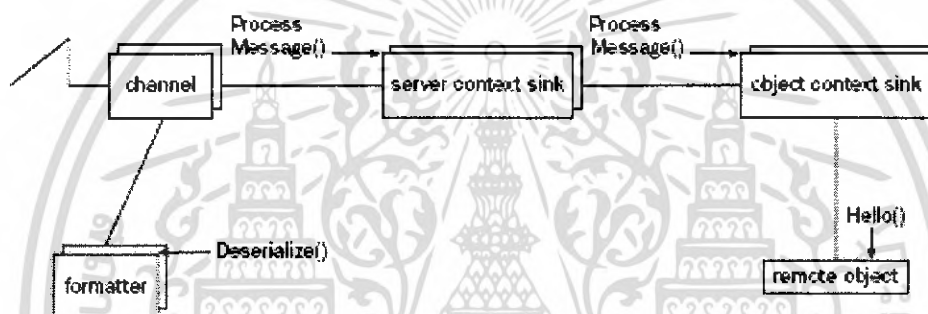
รูป 2-9 ภาพรวมของส่วนประกอบต่างๆ

เมื่อไคลเอนท์เรียกเมธอดในรีโมทออบเจกต์ แต่จริงๆ แล้วจะเรียกเมธอดจาก Transparent Proxy แทนซึ่ง Transparent Proxy จะคล้ายกับออบเจกต์จริงตรงที่มีเมธอดสาธารณะเหมือนออบเจกต์จริง Transparent Proxy จะเห็นเมธอดสาธารณะของออบเจกต์จริง โดยการใส่กลไกการสะท้อนเพื่ออ่าน Metadata จาก Assembly

ในกรณีที่ Transparent Proxy เรียก Real Proxy แล้ว Real Proxy จะทำหน้าที่ส่งข้อความไปยังแชนเนล Real Proxy เป็นประเภทเชื่อมต่อได้ทันที คุณสามารถจัดวางมันไว้ในที่ที่กำหนด ค่าพื้นฐานของ Real Proxy ที่รวบรวมอยู่ใน Envoy Sink และส่งข้อความไปยัง Envoy Sink อันดับแรก Envoy Sink สามารถจัดวางและเปลี่ยนแปลงข้อความ ตัวอย่างเช่น Debugging Sink, Security Sink และ Synchronization Sink

Envoy Sink สุดท้ายจะส่งข้อความเข้าไปในแชนเนล วิธีการส่งข้อความผ่านสายขึ้นอยู่กับรูปแบบของข้อมูล SOAP และ Binary Formatters ใช้ได้กับ .NET Framework 1.1 แชนเนลมีหน้าที่ทั้งการเชื่อมต่อเพื่อฟัง Socket จากเซิร์ฟเวอร์หรือการส่งข้อมูลที่ Format แล้ว ถ้าเป็นแชนเนลที่กำหนดเองจะสามารถทำสิ่งที่แตกต่างกันได้ เช่น การปรับแต่งโค้ดเพื่อให้สามารถส่งข้อมูลไปยังอีกฝั่งได้ รายละเอียดฝั่งด้านเซิร์ฟเวอร์ ดังภาพ 2-10 เมื่อแชนเนลรับข้อความมาจากไคลเอนท์แล้วใช้รูปแบบของข้อมูลไม่ว่าจะเป็น SOAP หรือ Binary Data เข้าไปในข้อความ แล้วแชนเนลก็จะเรียก Server-Context Sink

- Server-Context Sink จะเป็นส่วนหนึ่งของ Chain of Sink แล้ว Sink ตัวสุดท้ายก็จะติดต่อไปยัง Chain of object-context sinks อย่างต่อเนื่อง
- Object-context sink ตัวสุดท้ายก็จะเรียกเมธอดที่อยู่ในรีโมทออบเจกต์



รูปที่ 2-10 รายละเอียดเซิร์ฟเวอร์

Object Context Sink จะจำกัดเนื้อหาของออบเจกต์ และ Server Context Sink ก็จะจำกัดเนื้อหาของเซิร์ฟเวอร์ Single Server Context Sink เพียงอันเดียวสามารถเข้าถึง Object Sink ได้มากกว่า 1 ตัว

### 2.5.5 Context

ก่อนที่จะใช้ .NET Remoting ในการสร้างเครื่องเซิร์ฟเวอร์และเครื่องไคลเอนท์ที่ติดต่อกันข้ามเครือข่าย และในกรณีที่แชนเนลคือต้องการแอปพลิเคชันโคมมอน เช่น การเรียกออบเจกต์ผ่านทางข้อความ

ถ้าเคยเขียน COM+ คอมโพเนนท์มาก่อน ก็จะรู้เกี่ยวกับเนื้อหาของ COM+ เนื้อหาใน .NET มีลักษณะคล้ายกัน เนื้อหาจะถูกเก็บรวบรวมไว้ในออบเจกต์ ภายในเนื้อหาของ COM+ นั้นถ้าออบเจกต์ต้องการสิ่งเดียวกันก็จะถูกจำกัดโดย Context Attributes

แอปพลิเคชัน โดเมน จะมีเนื้อความที่แตกต่างกัน แต่ละเนื้อความจะใช้จัดกลุ่มออบเจกต์ที่มีความต้องการเหมือนกัน เนื้อความประกอบด้วย กลุ่มของคุณสมบัติและใช้สำหรับแบ่งเนื้อความ เมื่อ Context-bound Object ถูกเข้าถึงด้วยเนื้อความที่ต่างกัน Interceptor ก็จะทำงานได้บางส่วนก่อนที่ออบเจกต์จะถูกเรียก ตัวอย่างเช่น กรณีที่ใช้สำหรับ Thread Synchronization, Transaction และ Security Management

สิ่งที่คลาสได้รับจาก MarshalByRefObject คือขอบเขตของแอปพลิเคชัน โดเมน ถ้าภายนอกแอปพลิเคชัน โดเมน Proxy ต้องการที่จะเข้าถึงออบเจกต์ คลาสจะได้รับจาก ContextBoundObject ว่าสิ่งที่ได้รับมาจาก MarshalByRefObject คือขอบเขตของเนื้อความ

Context-bound object จะมี Context Attribute ถ้า Context-bound object ปรากฏจาก Context Attribute จะถูกสร้างในเนื้อความของผู้สร้าง ส่วน Context-bound object ที่มี Context Attribute คือถูกสร้างในเนื้อความใหม่หรือในเนื้อความของผู้สร้าง ถ้าให้ Attribute เข้ากันได้

สิ่งที่จำเป็นในการทำความเข้าใจในส่วนต่อๆ ไป

- การสร้างแอปพลิเคชัน โดเมนจะสร้างเนื้อความพื้นฐานในแอปพลิเคชัน โดเมน ถ้าออบเจกต์ใหม่ต้องการอะไรที่แตกต่างจากที่มีอยู่ก็จะสร้างเนื้อความใหม่ขึ้นมา
- Context attributes จะถูกกำหนดไปที่คลาสที่มาจาก ContextBoundObject และยังสามารถสร้างคลาสขึ้นมาเองโดยส่วนติดต่อ IContextAttribute ถ้า .NET Framework มีคลาสเดียวใน namespace จะเป็น System.Runtime.Remoting.Contexts : SynchronizationAttribute
- Context attributes ได้รับ Context Properties ที่ต้องการออบเจกต์ โดยเนื้อความจะต้องมีส่วนติดต่อของ IContextProperty คลาส ContextAttribute จะสร้างทั้ง IContextProperty และ IContextAttribute และสามารถใช้เป็นพื้นฐานของการสร้างคลาส
- Message sink เป็น Interceptor สำหรับเรียกเมธอดและ Message Sink Method Calls สามารถถูกสกัดได้แต่ก็ยังมี Properties คอยสนับสนุน Message Sinks

### 2.5.6 Activation

เนื้อความใหม่ถูกสร้างขึ้นถ้าคลาสต้องการเนื้อความที่มีความแตกต่างจากเนื้อความที่มีอยู่ Attributes Classes จะติดต่อกับคลาสเป้าหมาย โดยคำว่า Property ทั้งหมดของเนื้อความที่มีอยู่ยอมรับได้หรือไม่ ถ้ามี Property ใดที่ไม่ยอมรับก็จะถาม property ทั้งหมดที่ต้องการแล้วก็สร้างเนื้อความใหม่ ต่อมาก็จะหาว่า Sink ไหนต้องการติดตั้ง คลาสสามารถเพิ่มส่วนติดต่อของ IContributeXXXSink เพื่อรองรับออบเจกต์

คุณสมบัติของเนื้อความจะถูกจำกัดโดย Class Attribute ต้องขยายส่วนติดต่อ IContextAttribute ถ้าเป็นเนื้อความที่กำหนดเองจะได้รับจาก ContextAttribute เพราะว่าคลาสนี้เป็นค่าพื้นฐานของส่วนติดต่อนี้

คลาส .NET Framework 1.1 คือ System.Runtime.Remoting.Contexts.SynchronizationAttribute โดย Synchronization Attribute ถูกกำหนดโดย Synchronization Requirement ซึ่งจะกำหนด Synchronization Property ที่ต้องการโดยออบเจกต์ ซึ่ง Attribute กำหนดว่าถ้ามีหลาย Thread จะไม่สามารถเข้าถึงออบเจกต์ได้พร้อมกัน แต่ Thread ในการเข้าถึงออบเจกต์สามารถเปลี่ยนแปลงได้

**Attribute** มีค่าได้ 4 ค่า

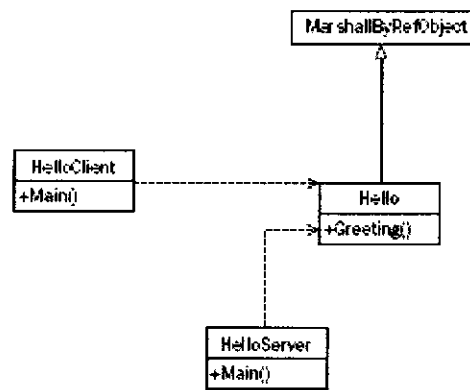
- **NOT\_SUPPORTED** ในกรณีที่คลาสไม่มีค่าในเนื้อความตามที่ Synchronization กำหนด
- **REQUIRED** ในกรณีที่มีการเรียก Synchronization Context
- **REQUIRES\_NEW** ในกรณีที่มีการสร้างเนื้อความใหม่
- **SUPPORTED** ในกรณีที่ยอมรับได้ตามที่ออบเจกต์กำหนด

### 2.5.7 Communication between Contexts

การติดต่อระหว่างเนื้อความทำโดย ไคลเอนท์จะใช้พรีอ็อกซ์แทน Real Object และยังสร้างข้อความที่จะส่งไปที่เซิร์ฟเวอร์ และ Sink ถ้าเป็นกลไกเดียวกันจะใช้สำหรับการติดต่อข้ามแอปพลิเคชัน โดเมนหรือระบบที่แตกต่างกัน TCP หรือ HTTP เซิร์ฟเวอร์ไม่ต้องการการติดต่อข้ามเนื้อความ แต่ช่องสัญญาณจะถูกใช้ที่นี้ด้วย CrossContextChannel ใช้ใน Virtual Memory เดียวกันทั้งฝั่งไคลเอนท์และเซิร์ฟเวอร์ของเซิร์ฟเวอร์และ Formatter ไม่ต้องการข้ามเนื้อความ

### 2.5.8 Remote Objects, Clients, and Servers

รูปภาพ 16-3 แสดงคลาสหลักของคลาสทั้งหมดในแอปพลิเคชันของไคลเอนท์และเซิร์ฟเวอร์ รีโมทออบเจกต์จะเรียก Hello และ HelloServer เป็นคลาสหลักของแอปพลิเคชันบนเซิร์ฟเวอร์ และ HelloClient สำหรับไคลเอนท์



รูปที่ 2-11 คลาสหลักของแอปพลิเคชัน

### 2.5.9 Remote Objects

รีโมทออบเจกต์เป็นสิ่งจำเป็นสำหรับการทำงานของระบบ ออบเจกต์จะเรียกโดยการรีโมทจากระบบที่แตกต่างกันซึ่งจะได้รับจาก System.MarshalByRefObject.MarshalByRefObject ออบเจกต์จะถูกจำกัดโดยแอปพลิเคชันโดเมนที่เป็นตัวสร้าง แสดงให้เห็นว่ามันไม่เคยส่งข้ามทางแอปพลิเคชันโดเมน และใช้พรีอ็อกซี่ ออบเจกต์ในการเข้าถึงรีโมทออบเจกต์แทนแอปพลิเคชัน โดเมนอื่น แอปพลิเคชัน โดเมนนั้น อาจอยู่ใน โปรเซสเดียวกัน, ต่างโปรเซสกัน หรือต่างระบบกัน

รีโมทออบเจกต์จะมีเอกลักษณ์เป็นของตนเอง ตรงที่ว่าการอ้างอิงวัตถุสามารถผ่านไปยังไคลเอนท์อื่นๆ ได้และยังคงเข้าถึงออบเจกต์เดิม ซึ่งพรีอ็อกซี่จะรู้เกี่ยวกับเอกลักษณ์ของรีโมทออบเจกต์

คลาส MarshalByRefObject มีหน้าที่รับเมธอดต่อจากคลาสวัตถุ ซึ่งเป็นเมธอดที่ทำการเริ่มต้นและรองรับ Lifetime Service ซึ่งบ่งบอกถึงระยะเวลาของรีโมทออบเจกต์

.NET Remoting จะสร้างคลาสไลบรารีสำหรับรีโมทออบเจกต์ อย่างคลาส Hello ที่ได้รับจาก System.MarshalByRefObject ใน Constructor และ Destructor ข้อความจะถูกเขียนที่คอนโซลซึ่งจะบ่งบอกถึงข้อมูลเกี่ยวกับ object lifetime นอกจากนี้ยังเพิ่มเมธอด Greeting() ซึ่งจะ เรียกจากไคลเอนท์

เพื่อที่จะสามารถจำแนกระหว่าง assembly และ คลาสในส่วนที่กล่าวต่อไปนี้ โดยให้ชื่ออาร์กิวเมนต์ของเมธอดแตกต่างกัน ชื่อของ assembly คือ RemoteHello ส่วนในคลาสคือ Hello

```

using System;

namespace Mvx.FooSharp.Remoting
{
    public class Hello : System.MarshalByRefObject
    {
        public Hello()
        {
            Console.WriteLine("Constructor called");
        }
        ~Hello()
        {
            Console.WriteLine("Destructor called");
        }

        public string Greeting(string name)
        {
            Console.WriteLine("Greeting called");
            return "Hello, " + name;
        }
    }
}

```

รูปที่ 2-12 คลาสของรีโมทออบเจกต์

### 2.5.10 โครงสร้างของ .NET Remoting

ส่วนประกอบหลักของ .NET Remoting Architecture มีดังต่อไปนี้

#### 2.5.10.1 Channels

แชนเนลถูกใช้ในการติดต่อสื่อสารระหว่าง .NET ไคลเอนท์และเซิร์ฟเวอร์ .NET Framework 1.1 ส่งด้วยคลาสแชนเนลซึ่งติดต่อโดยใช้ TCP, HTTP หรือจะสร้างโปรโตคอลขึ้นมาเองก็ได้

แชนเนล HTTP จะถูกใช้โดยเว็บเซอร์วิส ที่ใช้โปรโตคอล HTTP ในการติดต่อเพราะโดยปกติแล้วไฟร์วอลล์ จะเปิดพอร์ต 80 ดังนั้นไคลเอนท์ก็จะสามารถเข้าถึงเว็บเซิร์ฟเวอร์ .NET Remoting เว็บเซอร์วิสตอบรับพอร์ต 80 ได้ จึงใช้งานได้ง่ายด้วยไคลเอนท์เหล่านี้

สามารถใช้แชนเนล TCP บนอินเทอร์เน็ต แต่เพราะว่ามีไฟร์วอลล์ดังนั้นไคลเอนท์จะเข้าถึงพอร์ตที่กำหนดได้โดยใช้แชนเนล TCP ทั้งยังทำงานได้ดีกว่าแชนเนล HTTP ในสถานะอินเทอร์เน็ต

เมื่อมีการเรียกเมธอดบนรีโมทออบเจกต์ แล้วออบเจกต์แชนเนลไคลเอนท์จะส่งข้อความไปยังออบเจกต์แชนเนลรีโมท

ทั้งเซิร์ฟเวอร์และไคลเอนท์แอปพลิเคชันต้องสร้างแชนเนล ซึ่งมีโค้ดดังแสดงข้างล่าง

```

using System.Runtime.Remoting.Channels.Tcp;
...
TcpServerChannel channel = new TcpServerChannel(8086);

```

รูปที่ 2-13 โค้ดที่ใช้ในการสร้างแชนเนล

พอร์ตบน TCP Socket แต่ละพอร์ตจะทำตามคำสั่งที่ได้รับมา แชนเนลเซิร์ฟเวอร์ต้องกำหนดพอร์ตขึ้นมาและไคลเอนท์ก็จะเข้าถึงเซิร์ฟเวอร์ด้วยพอร์ตนี้ โดยจะสร้าง TcpClientChannel บนไคลเอนท์ ซึ่งไม่จำเป็นต้องเป็นพอร์ตเดียวกันเสมอ เพียงแต่ต้องเป็นพอร์ตที่เมื่อส่งไปแล้วเซิร์ฟเวอร์สามารถส่งข้อมูลกลับมาหาไคลเอนท์ได้

แชนเนล HTTP ใช้งานคล้ายๆ กับแชนเนล TCP สามารถกำหนดพอร์ตเพื่อให้เซิร์ฟเวอร์สร้าง listening socket ได้

เซิร์ฟเวอร์สามารถฟังหลายแชนเนลได้ ตัวอย่างการสร้าง HTTP และ TCP

```
using System;
using System.Runtime.Remoting;
using System.Runtime.Remoting.Channels;
using System.Runtime.Remoting.Channels.Tcp;
using System.Runtime.Remoting.Channels.Http;

namespace Krox.ProCSharp.Remoting
{
    public class HelloServer
    {
        public static void Main(string[] args)
        {
            TcpServerChannel tcpChannel = new TcpServerChannel(8085);
            HttpServerChannel httpChannel = new HttpServerChannel(8085);

            // register the channels
            ChannelServices.RegisterChannel(tcpChannel);
            ChannelServices.RegisterChannel(httpChannel);
            //...
        }
    }
}
```

#### รูปที่ 2-14 การสร้าง HTTP และ TCP

คลาสแชนเนลนั้นต้องมีส่วนติดต่อ IChannel ซึ่งประกอบไปด้วย 2 ส่วนคือ

- **ChannelName** เป็นไฟล์แบบ Read-only ซึ่งจะเปลี่ยนไปตามชื่อของแชนเนลซึ่งขึ้นอยู่กับประเภทของแชนเนล เช่น HTTP แชนเนล ชื่อว่า HTTP
- **ChannelPriority** เป็นไฟล์แบบ read-only ซึ่งการติดต่อระหว่างไคลเอนท์กับเซิร์ฟเวอร์ priority นั้นใช้จัดลำดับของแชนเนล ฟังไคลเอนท์แชนเนลที่มี priority สูงกว่าจะถูกเลือกให้ติดต่อกับเซิร์ฟเวอร์ก่อน

นอกจากนี้ interface จะขึ้นอยู่กับแชนเนลด้วยว่าเป็นไคลเอนท์แชนเนลหรือเป็นเซิร์ฟเวอร์แชนเนล ถ้าเป็นเซิร์ฟเวอร์ interface จะเป็น IChannelReceiver แต่ถ้าเป็นไคลเอนท์จะเป็น IChannelSender

คลาส `HttpChannel` และ `TcpChannel` สามารถใช้ได้ทั้งฝั่งไคลเอนท์และฝั่งเซิร์ฟเวอร์ โดยจะเป็นตัวสนับสนุน `IChannelSender` และ `IChannelReceiver` ซึ่งเป็นส่วนหนึ่งของ `IChannel`

`IChannelSender` ทางฝั่งไคลเอนท์จะมีการเรียก method ที่ชื่อว่า `CreateMessageSink()` โดยจะคืนค่า object ที่ Implement มาจาก `IMessageSink` โดย `IMessageSink` ซึ่งเป็น Interface สามารถที่จะถูกใช้เป็นตัว `Synchronous` หรือที่รู้จักกันว่าเป็น `Asynchronous Message` เข้าไปใน Channel ซึ่ง `IChannelReceiver` ทางฝั่งเซิร์ฟเวอร์ ซึ่งจะสามารถเริ่ม `listen mode` โดยใช้คำสั่ง `StartListening()` และหยุดได้โดยใช้คำสั่ง `StopListening()` โดย Property `ChannelData` จะสามารถถูกใช้สำหรับการเข้าถึงข้อมูล

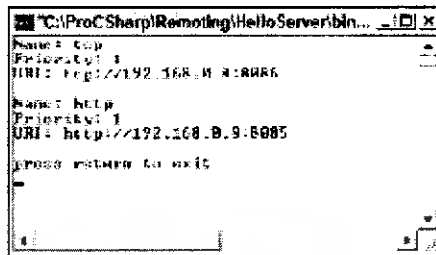
ถ้าต้องการข้อมูลเกี่ยวกับ configuration ของ Channel โดยใช้ Property หลายๆ ตัวของ Channel คลาส สำหรับทั้งสอง Channel (คือ `Http` และ `Tcp`) จะใช้ properties ต่อไปนี้ `ChannelName`, `ChannelPriority` และ `ChannelData` ซึ่ง `ChannelData` สามารถถูกใช้เป็นตัวรับข้อมูลเกี่ยวกับ URIs ที่ซึ่งถูกเก็บค่าในคลาส `ChannelDataStore` ซึ่ง `HttpChannel` จะมี property ที่ชื่อว่า `Scheme` ด้วย โดยตัวอย่างจะแสดงการแสดงผลข้อมูล

```
protected static void ShowChannelProperties(IChannelReceiver channel)
{
    Console.WriteLine("Name: " + channel.ChannelName);
    Console.WriteLine("Priority: " + channel.ChannelPriority);
    if (channel is HttpChannel)
    {
        HttpChannel httpChannel = channel as HttpChannel;
        Console.WriteLine("Scheme: " + httpChannel.ChannelScheme);
    }
    ChannelDataStore data = (ChannelDataStore)channel.ChannelData;
    foreach (string uri in data.ChannelUris)
    {
        Console.WriteLine("URI: " + uri);
    }
    Console.WriteLine();
}
```

รูปที่ 2-15 การแสดงผลข้อมูลของแชนเนล

เมธอด `ShowChannelProperties()` จะทำการเรียกหลังจากสร้าง Channel ใน `Main()` method การเริ่มต้น server จะได้ผลลัพธ์ที่แสดงออกมาจากทาง Console ที่ซึ่งจะถูกแสดงในรูป 16.5 ดังที่แสดงให้เห็น ตามปกติแล้วชื่อของ `TcpServerChannel` คือ `tcp` และ `Http Channel` ถูกเรียกว่า `http` ตามปกติแล้ว Channel ทั้งสองจะมีค่า `priority` เป็น 1 ซึ่งพอร์ตจะถูกเขตค่าที่ Constructor ซึ่งจะเห็นได้ใน URI โดย URI ของ Channel จะแสดงโปรโตคอล, ไอพีแอดเดรส และหมายเลขพอร์ต

```
TcpServerChannel tcpChannel = new TcpServerChannel(8086);
ShowChannelProperties(tcpChannel);
HttpServerChannel httpChannel = new HttpServerChannel(8085);
ShowChannelProperties(httpChannel);
```



รูปที่ 2-16 เมธอดในการสร้างแชนเนล

### 2.5.11 Setting channel Properties

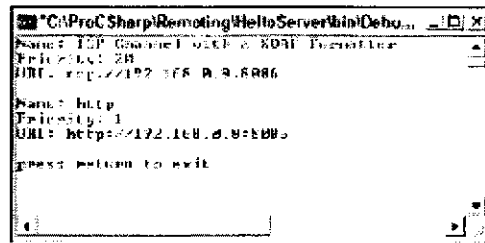
การตั้งค่า Properties ทั้งหมดของ Channel ใน list จะสามารถทำได้โดยใช้ Constructor TcpServerChannel(IDictionary , IServerChannelSinkProvider) ซึ่งคลาส Hashtable จะทำการ Implement IDictionary โดยจะสามารถตั้ง Name, Priority และ port โดยการใช้ class IDictionary เพื่อที่จะใช้คลาส Hashtable จะต้องทำการประกาศการใช้งาน namespace System.Collections

ด้วย constructor ของคลาส TcpServerChannel จะสามารถผ่านค่า object ที่ซึ่ง implement interface IServerChannelSinkProvider ซึ่งจะแทนที่ BinaryServerFormatterSinkProvider ที่ซึ่งเป็นค่าของ TcpServerChannel โดยปกติแล้วการ implement ของคลาส SoapServerFormatterSinkProvider จะถูกใช้ร่วมกับคลาส SoapServerFormatterSink กับ channel ที่ใช้ object SoapFormatter เพื่อที่จะทำการเปลี่ยนข้อมูลสำหรับการรับส่งข้อมูล

```
IDictionary properties = new Hashtable();
properties["name"] = "TCP Channel with a SOAP Formatter";
properties["priority"] = "20";
properties["port"] = "8086";
SoapServerFormatterSinkProvider sinkProvider =
    new SoapServerFormatterSinkProvider();
TcpServerChannel tcpChannel =
    new TcpServerChannel(properties, sinkProvider);
ShowChannelProperties(tcpChannel);
```

รูปที่ 2-17 ออบเจกต์ในการรับส่งข้อมูล

ซึ่งผลลัพธ์ที่ได้จะถูกแสดงออกมาทาง console ดังรูป 16-6



รูปที่ 2-18 ผลลัพธ์ออกมาทางคอนโซล

### 2.5.12 Pluggability of a channel

แชนเนลที่สร้างขึ้นมาสามารถส่งข้อความโดยการใช่โปรโตคอลที่นอกเหนือจาก HTTP หรือ TCP หรือจะ ขยายแชนเนล

- ด้านส่งต้องเพิ่มส่วนติดต่อ `IchannelSender` ส่วนที่สำคัญที่สุดคือ เมธอด `CreateMessageSink()` ที่ใช้ในกรณีที่ไคลเอนท์ส่ง URL และมีการเชื่อมต่อไปที่เซิร์ฟเวอร์ และ Message Sink ก็จะถูกสร้างขึ้นที่นี้ แล้ว proxy ก็จะใช้ในการส่งข้อความไปที่แชนเนล
- ด้านรับต้องเพิ่มส่วนติดต่อ `IchannelReceiver` จนกว่าจะเริ่มมีการรับ property จาก `ChannelData` จากนั้นก็รองรับข้อมูลจากไคลเอนท์ หลังจากที่มีการแบ่งข้อความแล้ว ก็จะใช้ `ChannelServices.SyncDispatchMessage()` ส่งข้อความไปที่ออบเจกต์

### 2.5.13 Formatter

.NET Framework มีคลาสอยู่ 2 รูปแบบคือ

- `.System.Runtime.Serialization.Formatters.Binary.BinaryFormatter`
- `.System.Runtime.Serialization.Formatters.Soap.SoapFormatter`

Formatters จะทำงานร่วมกับแชนเนลผ่าน formatter sink object และ formatter sink provider คลาสทั้งสองรูปแบบนี้จะมีส่วนติดต่อ `System.Runtime.Remoting.Messaging.IRemotingFormatter` โดยจะใช้เมธอด `Serialize()` และ `Deserialize()` เพื่อรับและส่งข้อมูลจากแชนเนล

คลาส formatter ที่กำหนดขึ้นมาเอง จะสามารถทำงานร่วมกับแชนเนลที่ต้องการใช้ ซึ่งทำได้โดยใช้ sink formatter และ sink formatter โปรไวเดอร์ ตัวอย่างเช่น `SoapServerFormatterSinkProvider` สามารถส่งอาร์กิวเมนต์ เมื่อมีการสร้างแชนเนลใหม่ ซึ่ง sink formatter โปรไวเดอร์ เพิ่มส่วนติดต่อ `IServerChannelSinkProvider` สำหรับฝั่งเซิร์ฟเวอร์ และ `IClientChannelSinkProvider` สำหรับฝั่งไคลเอนท์ และทั้งสองส่วนติดต่อนี้จะใช้ เมธอด `CreateSink()` เมื่อ sink formatter ถูกส่งคืน ส่วน `SoapServerFormatterSinkProvider` ก็จะส่งคลาส `SoapServerFormatterSink`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฝั่งไคลเอนต์คลาส SoapClientFormatterSink จะใช้เมธอด SyncProcessMessage() และ AsyncProcessMessage() ของคลาส SoapFormatter เพื่อจัดลำดับข้อความ และใช้คลาส SoapServerFormatterSink ในการแยกลำดับข้อความ แล้วก็ใช้คลาส SoapFormatter อีกครั้ง

โดย Sink และคลาสโพรไวเดอร์ทั้งหมดนี้สามารถเปลี่ยนแปลงได้ตามที่กำหนด

#### 2.5.14 Channel Services and Remoting Configuration

คลาส ChannelServices ถูกใช้สำหรับรีจิสเตอร์แชนเนลเข้าไปใน .NET Remoting ซึ่งคลาสนี้ยังสามารถเข้าถึงได้ทุกแชนเนลที่รีจิสเตอร์แล้ว

แชนเนลจะทำการรีจิสเตอร์โดยใช้เมธอด ChannelServices.RegisterChannel()

โหนดของเซิร์ฟเวอร์ที่ใช้ในการรีจิสเตอร์ของแชนเนล HTTP และ TCP

```
TcpChannel tcpChannel = new TcpChannel(5088);
HttpChannel httpChannel = new HttpChannel(5085);
ChannelServices.RegisterChannel(tcpChannel);
ChannelServices.RegisterChannel(httpChannel);
```

รูปที่ 2-19 เซิร์ฟเวอร์ในการรีจิสเตอร์ของแชนเนล HTTP และ TCP

คลาส ChannelServices ใช้ในการส่งข้อความทั้งแบบ synchronous และ asynchronous และยกเลิกการรีจิสเตอร์ของช่องสัญญาณได้ ทุกแชนเนลที่รีจิสเตอร์แล้วจะส่ง RegisteredChannels กลับไปในอาร์เรย์ของ IChannel และใช้เมธอด GetChannel() ในการเรียกแชนเนลที่ต้องการโดยใช้ชื่อของแชนเนล และยังสามารถเขียน utility เองเพื่อใช้ในการจัดการกับแชนเนล ดังตัวอย่างข้างล่าง

```
HttpServerChannel channel = (HttpServerChannel)ChannelServices.GetChannel("http");
channel.StopListening(null);
```

รูปที่ 2-20 การจัดเก็บแชนเนล

ส่วนคลาส RemotingConfiguration ก็เป็นเครื่องมืออีกตัวหนึ่งของคลาส .NET Remoting บนเซิร์ฟเวอร์ใช้สำหรับการรีจิสเตอร์ชนิดของรีโมทออบเจกต์ สำหรับ server-activated object และ เรียบเรียงรีโมทออบเจกต์ที่อ้างถึงคลาส ObjRef.ObjRef ซึ่งจะทำหน้าที่จัดลำดับการนำเสนอออบเจกต์ที่ส่งฝ่ายสายนมา สำหรับฝั่งไคลเอนต์ RemotingServices ใช้เพื่อแบ่งแยกรีโมทออบเจกต์เพื่อสร้าง proxy จากออบเจกต์อ้างอิง

ข้างล่างนี้เป็นตัวอย่างโหนดของฝั่งเซิร์ฟเวอร์

```

RemotingConfiguration.RegisterWellKnownServiceType(
    typeof(Hello), // Type
    "Hi", // URI
    WellKnownObjectMode.SingleCall); // Mode

```

## รูปที่ 2-21 ตัวอย่างโค้ดของฝั่งเซิร์ฟเวอร์

อาร์กิวเมนต์แรกของ RegisterWellKnownServiceType() คือ typeof(Hello) ใช้ในการกำหนดชนิดของรีโมทออบเจกต์ ส่วนอาร์กิวเมนต์ที่สอง คือ "Hi" เป็นรูปแบบมาตรฐานของรีโมทออบเจกต์ที่ไคลเอนต์ใช้ในการเข้าถึงรีโมทออบเจกต์ และอาร์กิวเมนต์สุดท้ายเป็นโหมดของรีโมทออบเจกต์ โดยโหมดจะเป็นค่าของ WellKnownObjectMode อาทิเช่น SingleCall หรือ Singleton

- **SingleCall** หมายถึงออบเจกต์ที่ไม่มีสถานะ และเมื่อมีการเรียกรีโมทออบเจกต์จะมีค่าใหม่ทุกครั้ง โดยออบเจกต์แบบ single-call จะถูกสร้างมาจากฝั่งเซิร์ฟเวอร์โดยใช้เมธอด RemotingConfiguration.RegisterWellKnownServiceType() และอาร์กิวเมนต์ WellKnownObjectMode.SingleCall และนี่ถือเป็นข้อดีของเซิร์ฟเวอร์ตรงที่ว่าถ้ามีไคลเอนต์หลายเครื่องเซิร์ฟเวอร์ไม่จำเป็นต้องเก็บค่าไว้
- **Singleton** เป็นออบเจกต์ที่ถูกแชร์สำหรับทุกไคลเอนต์บนเซิร์ฟเวอร์ ออบเจกต์บางตัวสามารถใช้ในการแชร์ข้อมูลระหว่างไคลเอนต์ได้ ชนิดวัตถุนั้นสามารถ ลักษณะแบบนี้ไม่เป็นปัญหาสำหรับข้อมูลประเภท read-only แต่ถ้าข้อมูลแบบ read-write ออบเจกต์นี้จะถูกสร้างโดยเซิร์ฟเวอร์ด้วยเมธอด RemotingConfiguration.RegisterWellKnownServiceType() และอาร์กิวเมนต์ WellKnownObjectMode.Singleton

### 2.5.15 Server for Client-Activated Object

ถ้ารีโมทออบเจกต์จะเก็บสถานะของไคลเอนต์ที่ต้องการ จะต้องใช้ออบเจกต์ Client-activated และต่อไปจะ ศึกษาว่า Server-activated และ Client-activated ทำงานต่างกันอย่างไรทั้งฝั่งเซิร์ฟเวอร์และฝั่งไคลเอนต์

จะต้องเรียกเมธอด RemotingConfiguration.RegisterActivatedServiceType() แทนการเรียก RemotingConfiguration.RegisterWellKnownType() ซึ่งการใช้เมธอดนี้กำหนดเฉพาะชนิดเท่านั้นและไม่ต้องใช้ URI เป็นเพราะว่า ไคลเอนต์บางไคลเอนต์มีชนิดของออบเจกต์ที่ต่างกันแต่อาจจะมี URI ซึ่ง URI ของ client-activated object ทั้งหมดจะถูกกำหนดโดย RemotingConfiguration.ApplicationName

```
RemotingConfiguration.ApplicationName = "HelloServer";
RemotingConfiguration.RegisterActivatedServiceType(typeof(Hello));
```

## รูปที่ 2-22 การตั้งค่าแบบ Activated Object

### 2.5.16 Object Activation

ไคลเอนต์สามารถใช้และสร้างคลาสรีโมท Activator ก็ส่ง proxy ไปที่ Server-activated หรือถ้าเป็นรีโมทออบเจกต์ที่เคยเจอกันแล้วก็จะใช้เมธอด GetObject() และเมธอด CreateInstance() จะส่ง proxy กลับไปที่ client-activated remote object

โอเพอร์เรเตอร์ตัวใหม่จะใช้ activated remote object แทนการใช้คลาส Activator ส่วนรีโมทออบเจกต์ภายในไคลเอนต์จะใช้คลาส RemotingConfiguration

### 2.5.17 Application URL

ในทุก activation จะต้องกำหนด URL ที่รีโมทออบเจกต์ โดย URL นี้จะเป็น URL เดียวกับที่ใช้ในการค้นหาที่ Web Browser ในส่วนแรกจะระบุโปรโตคอลแล้วตามด้วย ชื่อเซิร์ฟเวอร์ หรือหมายเลขไอพี, หมายเลขพอร์ต และ URI ที่ระบุว่ามีกรีจิสเตอร์ของรีโมทออบเจกต์ในเซิร์ฟเวอร์เมื่อไหร่ซึ่งจะอยู่ในรูปแบบดังนี้

**protocol://server:port/URI**

ตัวอย่าง URL ข้างล่าง ซึ่งจะระบุโปรโตคอลว่าเป็น http หรือ tcp ชื่อเซิร์ฟเวอร์เป็น localhost หมายเลขพอร์ต เป็น 8085 และ 8086 ตามลำดับ ส่วน URI เป็น Hi ดังภาพ

```
http://localhost:8085/Hi
tcp://localhost:8086/Hi
```

## รูปที่ 2-23 แสดง URL

### 2.5.18 Activating well-known object

รายละเอียดของการ activate เมื่อเป็น well-known object ดังตัวอย่าง

```
Hello obj = (Hello)Activator.GetObject(typeof(Hello),
    "tcp://localhost:8086/Hi");
```

## รูปที่ 2-24 รายละเอียดของการ Activate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตัวอย่างข้างต้นเมธอด `GetObject()` เป็นของคลาส `System.Activator` ซึ่งจะเรียก `RemotingServices.Connect()` เพื่อส่ง proxy object ไปที่รีโมทออบเจกต์ โดยอาร์กิวเมนต์แรกของเมธอดนี้ จะกำหนดชนิดของรีโมทออบเจกต์ ส่วนอาร์กิวเมนต์ที่สองเป็น URL ของรีโมทออบเจกต์ ก็คือ `tcp://localhost:8086/Hi` แสดงให้เห็นว่าใช้ TCP เป็นโปรโตคอล `localhost:8086` คือ hostname และหมายเลขพอร์ต และสุดท้าย `Hi` คือ URI ของออบเจกต์ที่กำหนดให้ใช้ `RemotingConfiguration.RegisterWellKnownServiceType()` แทนที่จะใช้ `Activator.GetObject()` อาจจะใช้ `RemotingServices.Connect()` โดยตรงเลยก็ได้

```

Hello obj = (Hello)RemotingServices.Connect(typeof(Hello),
                                           "tcp://localhost:8086/Hi");

```

### รูปที่ 2-25 ออบเจกต์ของรีโมทเซอร์วิส

ถ้าต้องการใช้โอเปอเรเตอร์ `new` เพื่อ activate well-known รีโมทออบเจกต์ โดยรีโมทออบเจกต์สามารถรีจิสเตอร์ที่ฝั่งไคลเอนท์โดยใช้ `RemotingConfiguration.RegisterWellKnownClientType()` และก็ใช้อาร์กิวเมนต์เช่นเดียวกันซึ่งประกอบด้วย ชนิดของรีโมทออบเจกต์ และ URI ซึ่งโอเปอเรเตอร์ตัวนี้จะไม่สร้างรีโมทออบเจกต์ตัวใหม่ แต่มันจะส่ง proxy ที่คล้ายกับ `Activator.GetObject()` แทน แต่ถ้ารีโมทออบเจกต์ถูกรีจิสเตอร์ด้วย flag `WellKnownObjectMode.SingleCall` และรีโมทออบเจกต์ก็จะถูกสร้างขึ้นใหม่ทุกครั้งที่มีเมธอดเรียก

```

RemotingConfiguration.RegisterWellKnownClientType(typeof(Hello),
                                                  "tcp://localhost:8086/Hi");
Hello obj = new Hello();

```

### รูปที่ 2-26 เมธอดในการสร้างรีโมทออบเจกต์

#### 2.5.19 Activating client-activated objects

รีโมทออบเจกต์จะคงสถานะของไคลเอนท์ไว้แล้ว `Activator.CreateInstance()` จะสร้าง Client-activated remote object การใช้เมธอด `Activator.GetObject()` รีโมทออบเจกต์จะถูกสร้างบนเมธอด และจะถูกทำลายเมื่อเมธอดจบการทำงาน ออบเจกต์จะไม่คงสถานะบนเซิร์ฟเวอร์ ซึ่งต่างกันตรงที่ `Activator.CreateInstance()` กับเมธอด `CreateInstance()` จะเริ่มด้วยการสร้างรีโมทออบเจกต์ และออบเจกต์นี้จะทำงานจนกระทั่งหมดอายุงาน

บางครั้งที่มีเมธอด `Activator.CreateInstance()` เกิด โอเวอร์โหลด จะสามารถสร้างได้แค่ออบเจกต์ภายใน ถ้าต้องการสร้างรีโมทออบเจกต์เมธอดจะต้องการที่ซึ่งรองรับการส่งผ่านได้ ดังตัวอย่างข้างล่างนี้ เมธอดรับสตริงพารามิเตอร์ 2 ส่วน ส่วนแรกคือ ชื่อของแอสเซมบลี และส่วนที่สองคือชนิด และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พารามิเตอร์ที่สามคือ อาร์เรย์ของออบเจกต์ ชื่อของเซิร์ฟเวอร์และออบเจกต์จะถูกกำหนดในออบเจกต์ อาร์เรย์ด้วยความช่วยเหลือของ `UrlAttribute` และใช้คลาส `UrlAttribute` ใน `System.Runtime.Remoting.Activation`

```
object[] attrs = {new UrlAttribute("tcp://localhost:8086/HelloServer") };
ObjectHandle handle = Activator.CreateInstance(
    "RemoteHello", "Wrox.ProCSharp.Remoting.Hello", attrs);
if (handle == null)
{
    Console.WriteLine("could not locate server");
    return;
}
Hello obj = (Hello)handle.Unwrap();
Console.WriteLine(obj.Greeting("Christian"));
```

### รูปที่ 2-27 คลาส URL Attribute

สำหรับ Client-activated object ถ้าใช้โอเปอเรเตอร์ `new` แทนคลาส `Activator` โดยการทำแบบนี้จะต้องรีจิสเตอร์โดยใช้ `RemotingConfiguration.RegisterActivatedClientType()` ในโครงสร้างของ client-activated object โอเปอเรเตอร์ `new` ไม่แค่ส่ง proxy กลับไปแต่ยังสร้างรีโมทออบเจกต์อีกด้วย

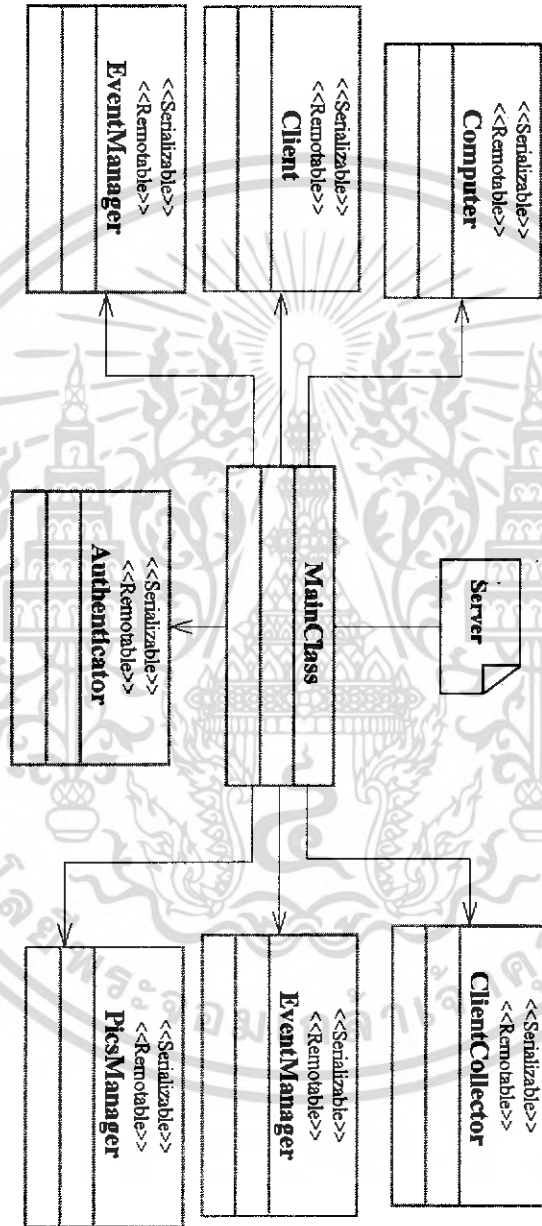
```
RemotingConfiguration.RegisterActivatedClientType(typeof(Hello),
    "tcp://localhost:8086/HelloServer");
Hello obj = new Hello();
```

### รูปที่ 2-28 ออบเจกต์ที่ใช้แทนคลาส Activator

# บทที่ 3

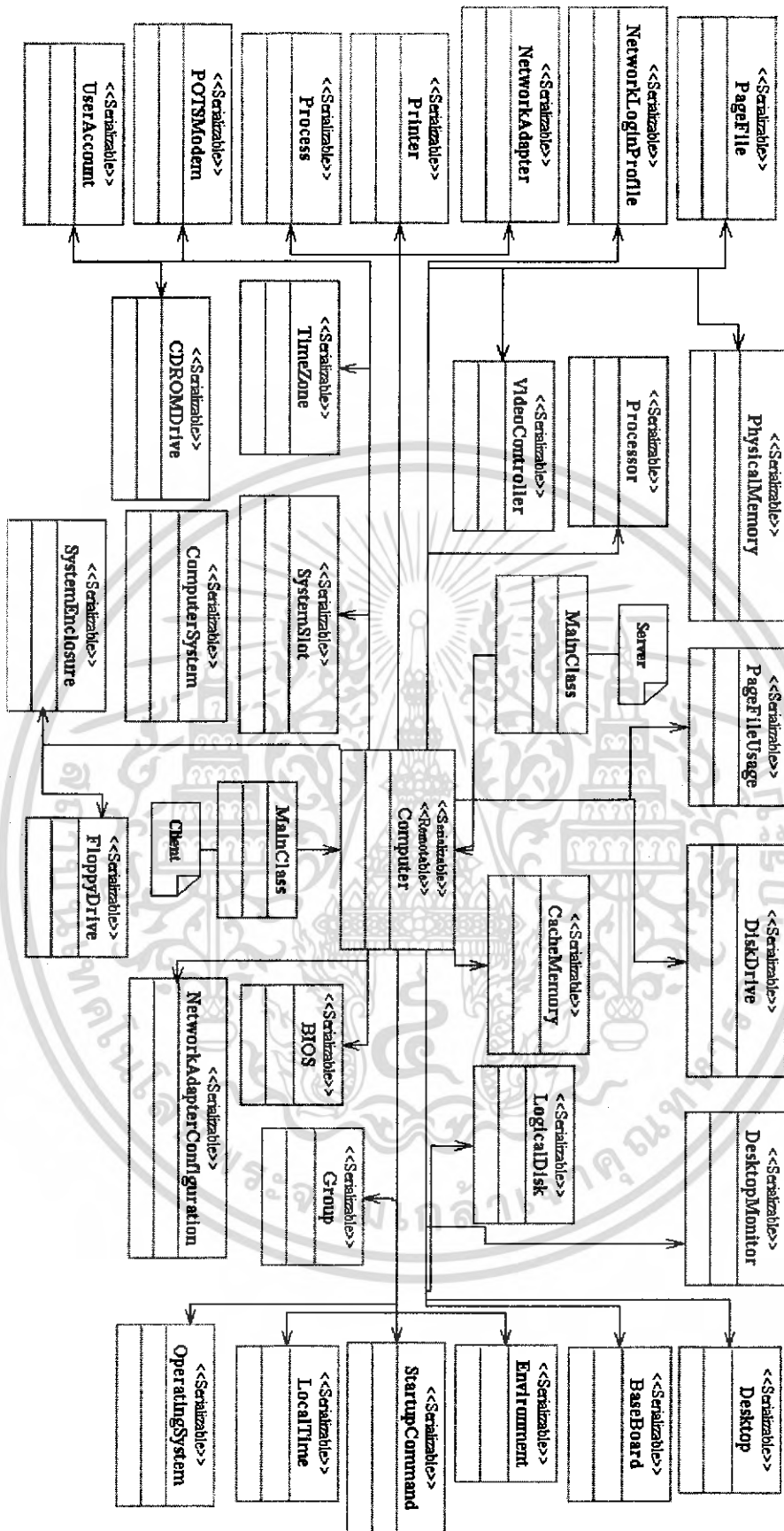
## การวิเคราะห์และการออกแบบ

### 3.1 Domain Model



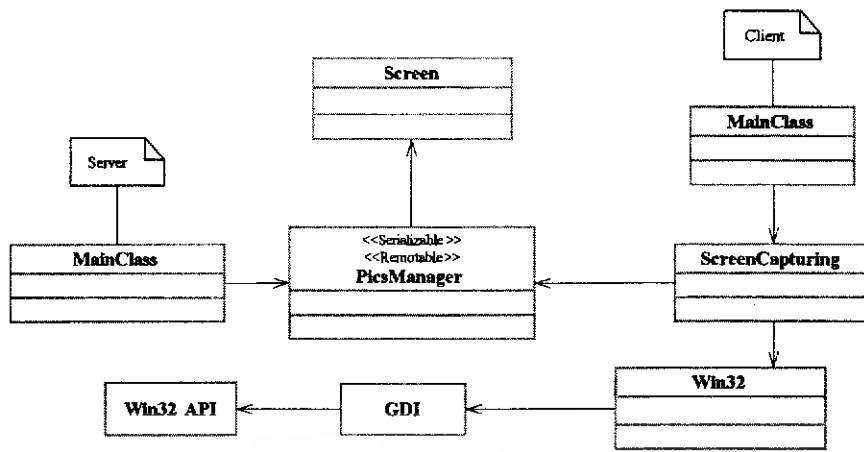
รูปที่ 3-1 Domain Model ในส่วนที่เกี่ยวข้องกับรีโมตออบเจกต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

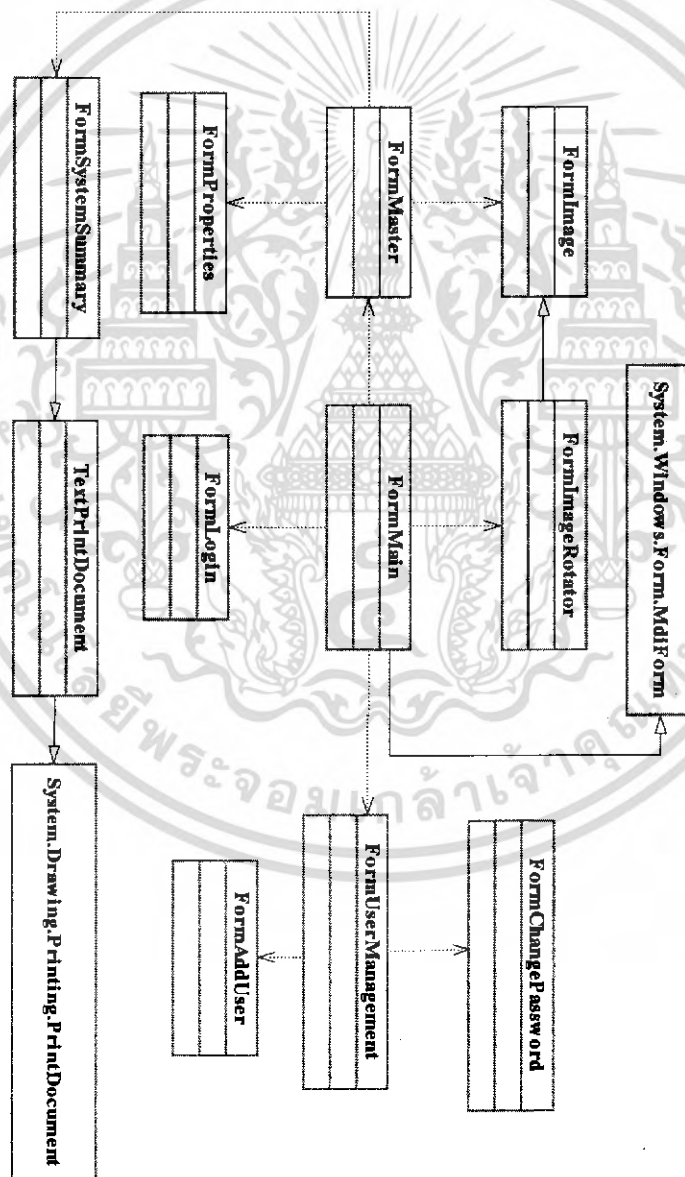


รูปที่ 3-2 Domain Model ในส่วนที่เกี่ยวกับการเก็บข้อมูลจากเครื่องไคลเอนท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



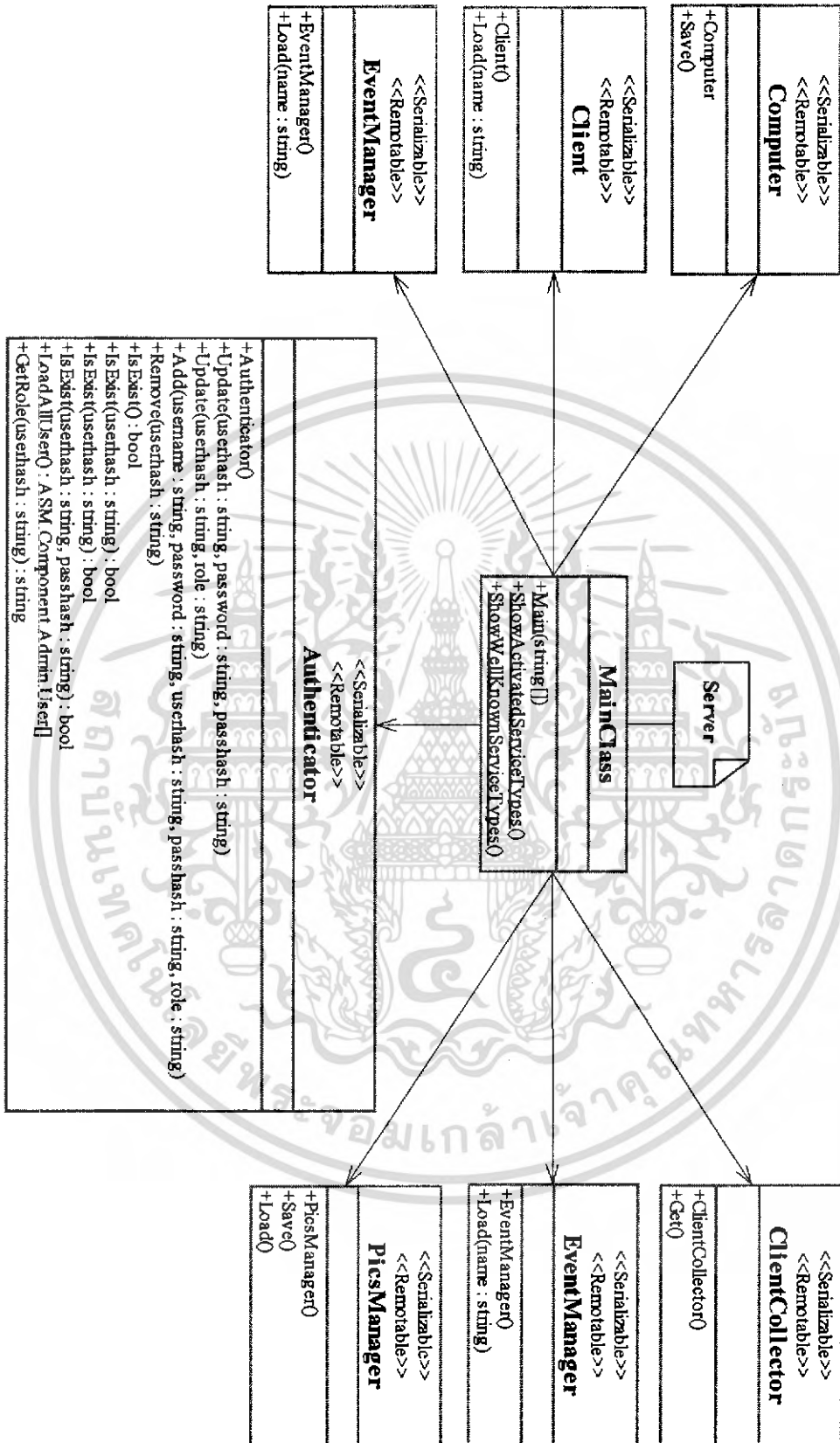
รูปที่ 3-3 Domain Model ในส่วนที่เกี่ยวกับการค้จ้งบอภาพจากไคลเอนท์



รูปที่ 3-4 Domain Model ในส่วนที่เกี่ยวกับส่วนควบคุมระบบ

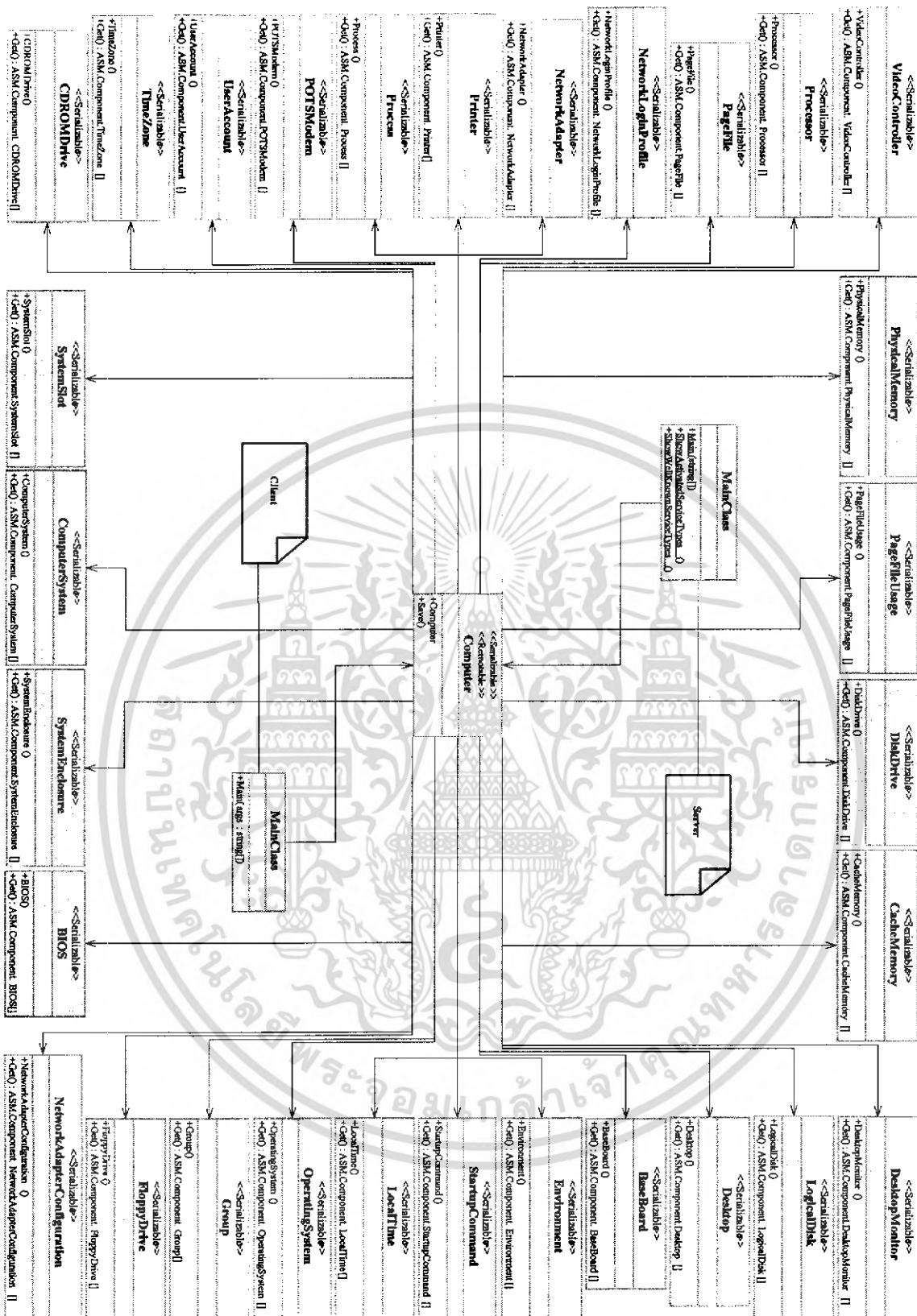
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2 กลยุทธ์โคอะแกรม



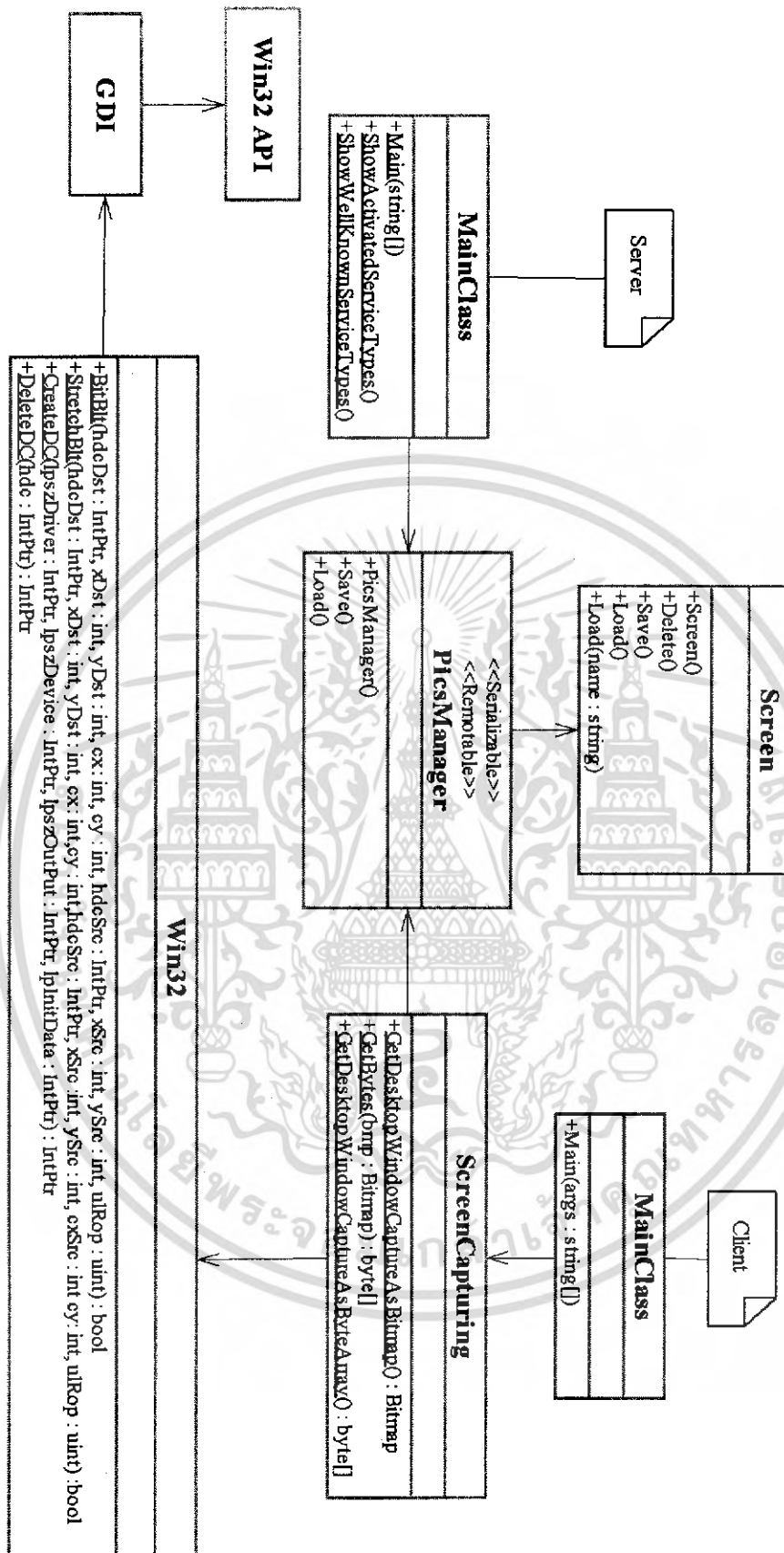
รูปที่ 3-5 Class Diagram ในส่วนที่เกี่ยวข้องกับริโมตออปเจก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



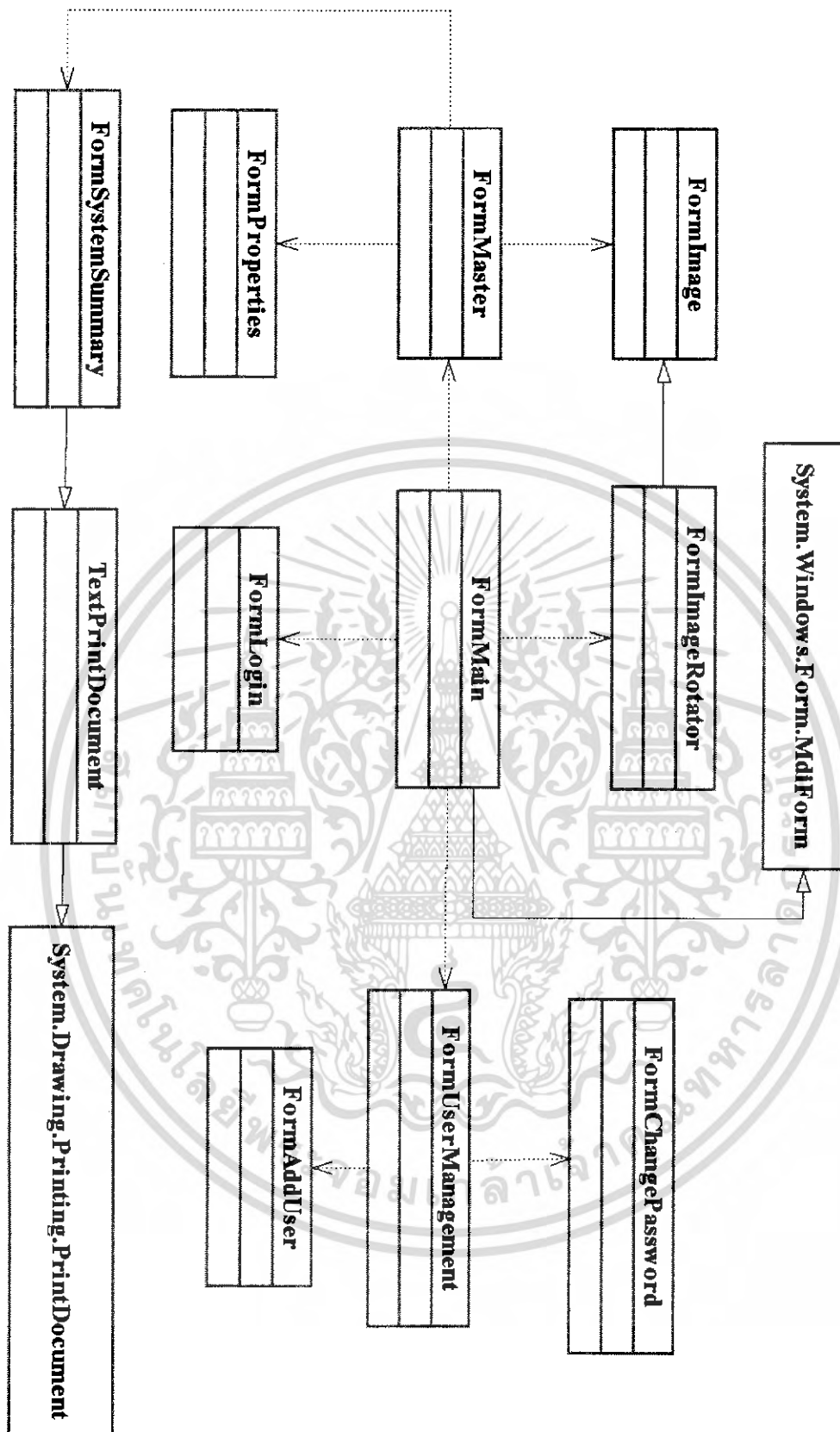
รูปที่ 3-6 Class Diagram ในส่วนที่เกี่ยวกับการเก็บข้อมูลจากเครื่องโคลน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-7 Class Diagram ในส่วนที่เกี่ยวกับการดักจับจอภาพจากไคลเอนท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-8 Class Diagram ในส่วนที่เกี่ยวกับส่วนควบคุมระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<b>MainClass</b>
+Main(args : string[])

รูปที่ 3-9 Class ASM.Client.MainClass

<b>WindowsService</b>
-attrs : object[] = {new UrlAttribute("tcp://LNOKITEL:13011/Server")} -ClientName : string -timerComputer : Timer -timerScreen : Timer -eventLog1 : EventLog -components : Container = null
+Client() -InitializeComponent() #Dispose(disposing : bool) -Main() #OnStart(args : string[]) #OnStop() #OnShutdown() -timerComputer_Elapsed(sender : object,e : ElapsedEventArgs) -timerScreen_Elapsed(sender : object,e : ElapsedEventArgs) -Scan() -ScreenCapture() -EventManager()

รูปที่ 3-10 Class ASM.Client.WindowsService

<<Serializable>> <b>BaseBoard</b>
-hotSwappable : string -manufacturer : string -product : string -serialNumber : string -version : string +HotSwappable : string +Manufacturer : string +Product : string +SerialNumber : string +Version : string
+BaseBoard () +Get() : ASM.Component. BaseBoard []

รูปที่ 3-11 Class ASM.Component.BaseBoard

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<<Serializable>> <b>BIOS</b>
-currentLanguage : string -installableLanguage : int -listOfLanguage : string -manufacturer : string -name : string -primaryBIOS : string -releaseDate : string -serialNumber : string -smbiosBIOSVersion : string -smbiosMajorVersion : int -smbiosMinorVersion : int -smbiosPresent : string -softwareElementState : int -version : string +CurrentLanguage : string +InstallableLanguages : int +ListOfLanguages : string +Manufacturer : string +Name : string +PrimaryBIOS : string +ReleaseDate : string +SerialNumber : string +SMBIOSBIOSVersion : string +SMBIOSBIOSMajorVersion : int +SMBIOSBIOSMinorVersion : int +SMBIOSPresent : string +SoftwareElementState : int +Version : string
+BIOS() +Get() : ASM.Component. BIOS[]

รูปที่ 3-12 Class ASM.Component.BIOS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<<Serializable>> <b>CacheMemory</b>
-associativity : int -blockSize : int -cacheType : int -currentSRAM : int -deviceID : string -installedSize : int -level : int -location : int -maxCacheSize : int -numberOfBlocks : int -purpose : string -supportedSRAM : int -writePolicy : int +Associativity : int +BlockSize : int +CacheType : int +CurrentSRAM : int +DeviceID : string +InstalledSize : int +Level : int +Location : int +MaxCacheSize : int +NumberOfBlocks : int +Purpose : string +SupportedSRAM : int +WritePolicy : int +CacheMemory () +Get() : ASM.Component.CacheMemory []

รูปที่ 3-13 Class ASM.Component.CacheMemory

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<<Serializable>> <b>CDROMDrive</b>
-drive :string -fileSystemFlagsEx : string -maximumComponentLength : int -mediaLoaded : string -name : string -scsiBus : int -scsiLogicalUnit : int -scsiPort : int -size : long -transferRate : double -volumnName : string -volumnSerialNumber : string +Drive : string +FileSystemFlagsEx : string +MaximumComponentLength : int +MediaLoaded : string +Name : string +SCSIBus : int +SCSILogicalUnit : int +SCSIPort : int +SCSITargetId : int +Size : long +TransferRate : double +VolumnName : string +VolumnSerialNumber : string +CDROMDrive () +Get() : ASM.Component.CDROMDrive []

รูปที่ 3-14 Class ASM.Component.CDROMDrive

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



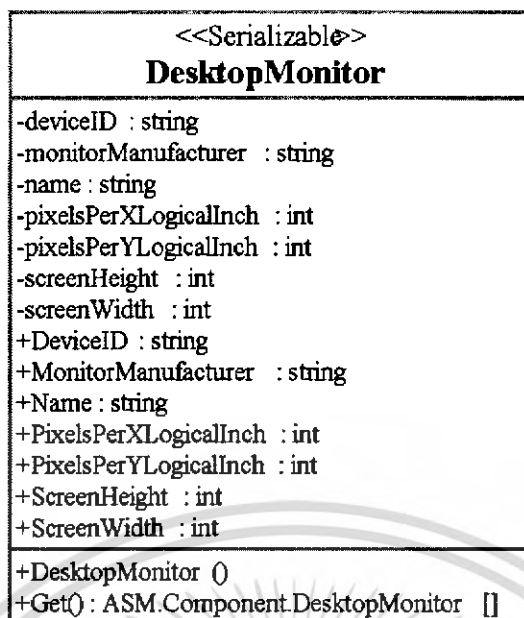
รูปที่ 3-15 Class ASM.Component.ComputerSystem

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

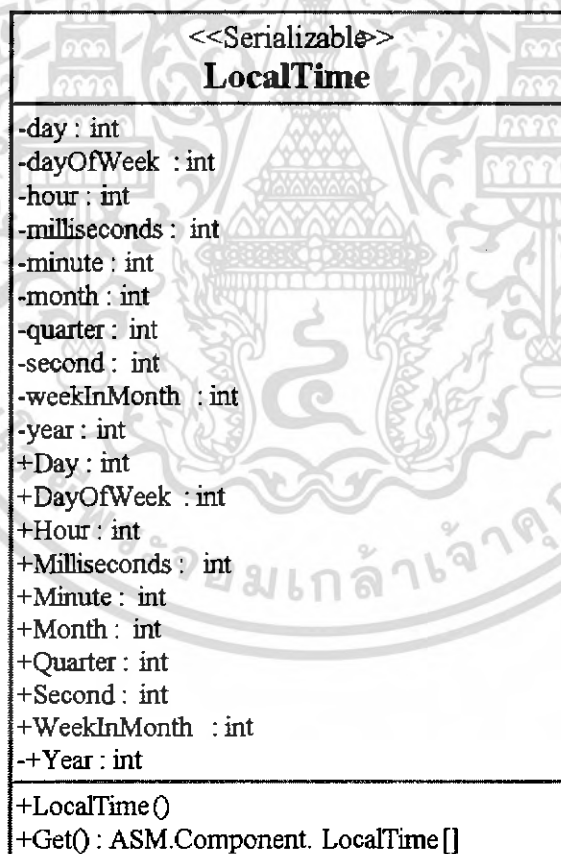
<<Serializable>> <b>Desktop</b>
-borderWidth : int -coolsWitch : string -cursorBlinkRate : int -dragFullWindows : string -gridGranularity : int -iconSpacing : int -iconTitileFaceName : string -iconTitleSize : int -iconTitleWrap : string -name : string -screenSaverActive : string -screenSaverExecutable : string -screenSaverSecure : string -screenSaverTimeout : int -wallPaper : string -wallPaperStretched : string -wallPaperTiled : string +BorderWidth : int +CoolsWitch : string +CursorBlinkRate : int +DragFullWindows : string +GridGranularity : int +IconSpacing : int +IconTitileFaceName : string +IconTitleSize : int +IconTitleWrap : string +Name : string +ScreenSaverActive : string +ScreenSaverExecutable : string +ScreenSaverSecure : string +ScreenSaverTimeout : int +WallPaper : string -WallPaperStretched : string +WallPaperTiled : string
+Desktop () +Get() : ASM.Component.Desktop []

รูปที่ 3-16 Class ASM.Component.Desktop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-17 Class ASM.Component.DesktopMonitor



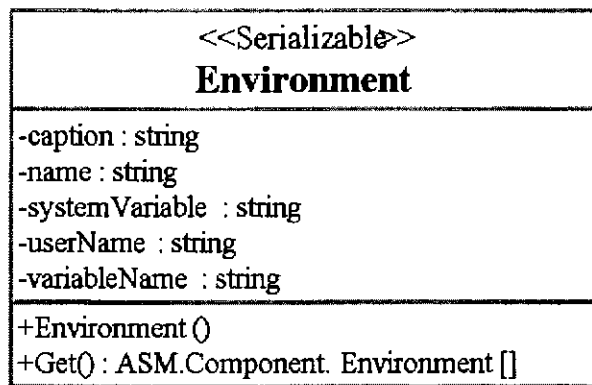
รูปที่ 3-18 Class ASM.Component.LocalTime

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

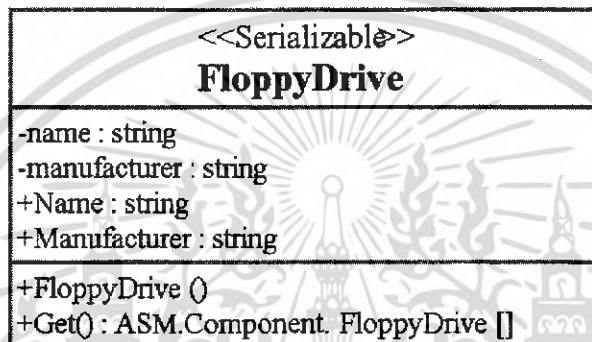
<<Serializable>> <b>DiskDrive</b>
-bytesPerSector : int -caption : string -index : int -interfaceType : string -manufacturer : string -mediaType : string -model : string -name : string -partitions : int; -scsiBus : int; -scsiLogicalUnit : int -scsiPort : int -scsiTargetId : int -sectorsPerTrack : int -signature : long -size : long -totalCylinders : long -totalHeads : long -totalSectors : long -totalTracks : long -tracksPerCylinder : int +BytesPerSector : int +Caption : string +Index : int +InterfaceType : string +Manufacturer : string +MediaType : string +Medel : string +Name : string +Partitions : int +SCSIBus : int +SCSILogicalUnit : int +SCSIPort : int +SCSITargetId : int +SectorsPerTrack : int +Size : long +TotalCylinders : long +TotalHeads : long +TotalSectors : long +TotalTracks : long +TracksPerCylinder : int
+DiskDrive () +Get() : ASM.Component.DiskDrive []

รูปที่ 3-19 Class ASM.Component.DiskDrive

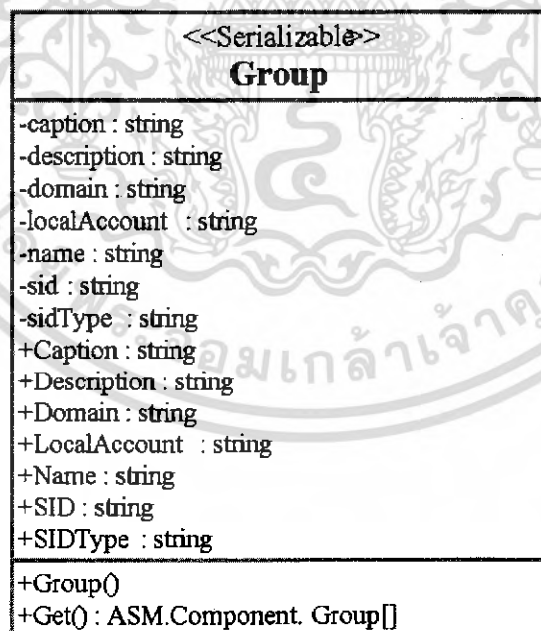
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-20 Class ASM.Component.Environment



รูปที่ 3-21 Class ASM.Component.FloppyDrive



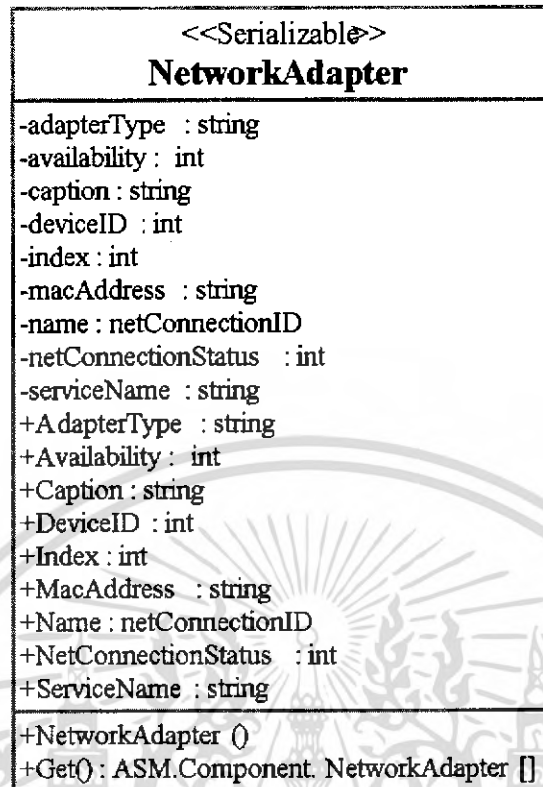
รูปที่ 3-22 Class ASM.Component.Group

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<<Serializable>> <b>LogicalDisk</b>
-compressed : string -description : string -driveType : int -fileSystem : string -freeSpace : long -maximumComponentLength : int -mediaType : int -name : string -size : long -supportsFileBasedCompression : string -volumnName : string -volumnSerialNumber : string +Compressed : string +Description : string +DriveType : int +FileSystem : string +FreeSpace : long +MaximumComponentLength : int +MediaType : int +Name : string +Size : long +SupportsFileBasedCompression : string +VolumnName : string +VolumnSerialNumber : string +LogicalDisk () +Get() : ASM.Component. LogicalDisk []

รูปที่ 3-23 Class ASM.Component.LogicalDisk

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-24 Class ASM.Component.NetworkAdapter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<<Serializable>> <b>NetworkAdapterConfiguration</b>	
-caption	: string
-databasePath	: string
-defaultIPGateway	: string
-description	: string
-dhcpEnabled	: string
-dhcpServer	: string
-dhcpDomain	: string
-dnsDomainSuffixSearchOrder	: string
-dnsEnabledForWINSResolution	: string
-dnsHostName	: string
-dnsServerSearch	: string
-fullDNSRegistrationEnabled	: string
-gatewayCostMetric	: string
-igmpLevel	: string
-index	: int
-ipAddress	: string
-ipConnectionMetric	: int
-ipEnabled	: string
-ipFilterSecurityEnabled	: string
-ipPortSecurityEnabled	: string
-ipsecPermitIPProtocols	: int
-ipsecPermitTCPPorts	: int
-ipsecPermitUDPPorts	: int
-ipSubnet	: string
-ipxAddress	: string
-ipxEnabled	: string
-ipxFrameType	: string
-ipxMediaType	: string
-ipxVirtualNetNumber	: string
-macAddress	: string
-tcpipNetbiosOptions	: int
-tcpWindowsSize	: int
-winsEnableLMHostsLookup	: string
+Caption	: string
+DatabasePath	: string
+DefaultIPGateway	: string
+Description	: string
+DHCPEnabled	: string
+DHCPServer	: string
+DHCPDomain	: string
+DNSDomainSuffixSearchOrder	: string
+DNSEnabledForWINSResolution	: string
+DNSHostName	: string
+DNSServerSearch	: string
+FullDNSRegistrationEnabled	: string
+GatewayCostMetric	: string
+IGMPLevel	: string
+Index	: int
+IPAddress	: string
+IPConnectionMetric	: int
+IPEnabled	: string
+IPFilterSecurityEnabled	: string
+IPPortSecurityEnabled	: string
+IPSecPermitIPProtocols	: int
+IPSecPermitTCPPorts	: int
+IPSecPermitUDPPorts	: int
+IPSubnet	: string
+IPXAddress	: string
+IPXEnabled	: string
+IPXFrameType	: string
+IPXMediaType	: string
+IPXVirtualNetNumber	: string
+MACAddress	: string
+TcpipNetbiosOptions	: int
+TcpWindowsSize	: int
+WINSEnableLMHostsLookup	: string
+NetworkAdapterConfiguration ()	
+Get() : ASM.Component. NetworkAdapterConfiguration []	

รูปที่ 3-25 Class ASM.Component.NetworkAdapterConfiguration

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-26 Class ASM.Component.NetworkLoginProfile

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<<Serializable>> OperatingSystem
<pre> -BootDevice : string -BuildNumber : int -Caption : string -CodeSet : int -CountryCode : int -CSDVersion : string -CSName : string -CurrentTimeZone : int -Debug : string -Distributed : string -EncryptionLevel : int -FreePhysicalMemory : long -FreeSpaceInPagingFiles : long -FreeVirtualMemory : long -LocalDateTime : string -LastBootupTime : string -Locale : string -Manufacturer : string -MaxNumberOfProcesses : long -Name : string -NumberOfProcesses : int -NumberOfUsers : int -Organization : string -OSLanguage : int -OSType : int -Primary : string -RegisteredUser : string -SerialNumber : string -ServicePackMajorVersion : int -ServicePackMinorVersion : int -SizeStoredInPagingFile : long -SystemDevice : string -SystemDirectory : string -TotalSwapSpaceSize : long -TotalVirtualMemorySize : long -TotalVisibleMemorySize : long -Version : string -WindowsDirectory : string +BootDevice : string +BuildNumber : int +Caption : string +CodeSet : int +CountryCode : int +CSDVersion : string +CSName : string +CurrentTimeZone : int +Debug : string +Distributed : string +EncryptionLevel : int +FreePhysicalMemory : long +FreeSpaceInPagingFiles : long +FreeVirtualMemory : long +LocalDateTime : string +LastBootupTime : string +Locale : string +Manufacturer : string +MaxNumberOfProcesses : long +Name : string +NumberOfProcesses : int +NumberOfUsers : int +Organization : string +OSLanguage : int +OSType : int +Primary : string +RegisteredUser : string +SerialNumber : string +ServicePackMajorVersion : int +ServicePackMinorVersion : int +SizeStoredInPagingFile : long +SystemDevice : string +SystemDirectory : string +TotalSwapSpaceSize : long +TotalVirtualMemorySize : long +TotalVisibleMemorySize : long +Version : string +WindowsDirectory : string  +OperatingSystem () +Get() : ASM.Component.OperatingSystem [] </pre>

รูปที่ 3-27 Class ASM.Component.OperatingSystem

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<<Serializable>> <b>PageFile</b>
-archive : string -compressed : string -creationDate : string -drive : string -eightDotThreeFileName : string -encrypted : string -encryptionMethod : string -extension : string -fileName : string -fileSize : long -fileType : string -fsName : string -hidden : string -initialSize : long -inUseCount : long -maximumSize : long -name : string -path : string -readable : string -system : string -writeable : string +Archive() : string +Compressed : string +Drive : string +EightDotThreeFileName : string +Encrypted : string +EncryptionMethod : string +Extension : string +FileName : string +FileSize : long +FileType : string +FSName : string +Hidden : string +InitialSize : long +InUseCount : long +MaximumSize : long +Name : string +Path : string +Readable : string +System : string +Writeable : string +PageFile() +Get() : ASM.Component.PageFile[]

รูปที่ 3-28 Class ASM.Component.PageFile

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<<Serializable>> <b>PageFileUsage</b>
-allocatedBaseSize : long -currentUsage : long -name : string -peakUsage : long -tempPageFile : string +AllocatedBaseSize : long +CurrentUsage : long +Name : string +PeakUsage : long TempPageFile : string
+PageFileUsage() +Get() : ASM.Component.PageFileUsage[]

รูปที่ 3-29 Class ASM.Component.PageFileUsage

<<Serializable>> <b>PhysicalMemory</b>
-bankLabel : string -capacity : long -dataWidth : int -deviceLocator : string -formFactor : int -manufacturer : string -memoryType : int -positionInRow : string -tag : string -totalWidth : int -typeDetail : int +BankLabel : string +Capacity : long +DataWidth : int +DeviceLocator : string +FormFactor : int +Manufacturer : string +MemoryType : int +PositionInRow : string +Tag : string +TotalWidth : int +TypeDetail : int
+PhysicalMemory() +Get() : ASM.Component.PhysicalMemory[]

รูปที่ 3-30 Class ASM.Component.PhysicalMemory

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-31 Class ASM.Component.POTSModem



รูปที่ 3-32 Class ASM.Component.Printer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<<Serializable>> <b>Process</b>
-commandLine : string
-executablePath : string
-executionState : int
-handle : long
-handleCount : int
-kernelModeTime : long
-maximumWorkingSetSize : long
-minimumWorkingSetSize : long
-name : string
-otherOperatingCount : long
-otherTransferCount : long
-pageFaults : int
-pageFileUsage : long
-parentProcessId : int
-peakPageFileUsage : long
-priority : int
-privatePageCount : long
-processId : long
-quotaNonPagedPoolUsage : long
-quotaPagedPoolUsage : long
-quotaPeakNonPagePoolUsage : long
-quotaPeakPagedPoolUsage : long
-readOperationCount : long
-readTransferCount : long
-sessionId : int
-userModeTime : long
-virtualSize : long
-windowsVersion : string
-writeOperationCount : long
+CommandLine : string
+ExecutablePath : string
+ExecutionState : int
+Handle : long
+HandleCount : int
+KernelModeTime : long
+MaximumWorkingSetSize : long
+MinimumWorkingSetSize : long
+Name : string
+OtherOperationCount : long
+OtherTransferCount : long
+PageFileUsage : long
+ParentProcessId : int
+PeakPageFileUsage : long
+Priority : int
+PrivatePageCount : long
+ProcessId : long
+QuotaNonPagedPoolUsage : long
+QuotaPagedPoolUsage : long
+QuotaPeakPagedPoolUsage : long
+ReadOperationCount : long
+ReadTransferCount : long
+SessionId : int
+UserModeTime : long
+VirtualSize : long
+WindowsVersion : string
+WriteOperationCount : long
+WriteTransferCount : long
+Process()
+Get() : ASM.Component.Process[]

รูปที่ 3-33 Class ASM.Component.Process

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-34 Class ASM.Component.Processor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<b>ScreenCapturing</b>
+GetDesktopWindowCaptureAsBitmap () : Bitmap
+GetBytes (bmp : Bitmap) : byte[]
+GetDesktopWindowCaptureAsByteArray () : byte[]

รูปที่ 3-35 Class ASM.Component.ScreenCapturing

<<Serializable>> <b>StartupCommand</b>
-command : string
-location : string
-name : string
-user : string
+Command : string
+Location : string
+Name : string
+User : string
+StartupCommand ()
+Get() : ASM.Component.StartupCommand []

รูปที่ 3-36 Class ASM.Component.StartupCommand

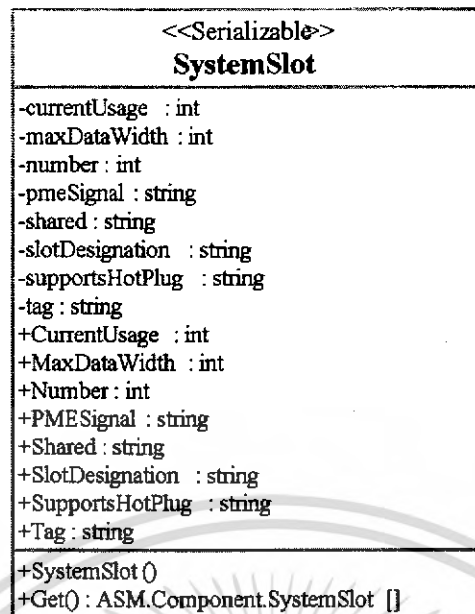
<<Serializable>> <b>SystemEnclosure</b>
-chasisTypes : int
+ChassisTypes : int
+SystemEnclosure ()
+Get() : ASM.Component.SystemEnclosure []

รูปที่ 3-37 Class ASM.Component.SystemEnclosure

<<Serializable>> <b>TimeZone</b>
-caption : string
-standardName : string
+Caption : string
+StandardName : string
+TimeZone ()
+Get() : ASM.Component.TimeZone []

รูปที่ 3-38 Class ASM.Component.TimeZone

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-39 Class ASM.Component.SystemSlot



รูปที่ 3-40 Class ASM.Component.UserAccount

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<<Serializable>> <b>VideoController</b>
-adapterCompatibility : string -adapterDACType : string -adapterRAM : int -caption : string -currentBitsPerPixel : int -currentHorizontalResolution : int -currentNumberOfColors : int -currentRefreshRate : long -currentScanMode : int -currentVerticalResolution : int -description : string -deviceID : string -driverDate : string -driverVersion : string -installedDisplayDrivers : string -maxRefreshRate : int -minRefreshRate : int -monoChrome : string -name : string -numberOfColorPlanes : int -videoArchitecture : int -videoMemoryType : int -videoModeDescription : string -videoProcessor : string +AdapterCompatibility : string +AdapterDACType : string +AdapterRAM : int +Caption : string +CurrentBitsPerPixel : int +CurrentHorizontalResolution : int +CurrentNumberOfColors : int +CurrentRefreshRate : long +CurrentScanMode : int +CurrentVerticalResolution : int +Description : string +DeviceID : string +DriverDate : string +DriverVersion : string +InstalledDisplayDrivers : string +MaxRefreshRate : int +MinRefreshRate : int +MonoChrome : string +Name : string +NumberOfColorPlanes : int +VideoArchitecture : int +VideoMemoryType : int +VideoModeDescription : string +VideoProcessor : string +VideoController () +Get() : ASM.Component. VideoController []

รูปที่ 3-41 Class ASM.Component.VideoController

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Win32
+BinaryRasterOperations : enum
+TernaryRasterOperations : enum
+BitBlt(hdcDst : IntPtr, xDst : int, yDst : int, cx : int, cy : int, hdcSrc : IntPtr, xSrc : int, ySrc : int, ulRop : uint) : bool
+StretchBlt(hdcDst : IntPtr, xDst : int, yDst : int, cx : int, cy : int, hdcSrc : IntPtr, xSrc : int, ySrc : int, cxSrc : int, cySrc : int, ulRop : uint) : bool
+CreateDC(lpszDriver : IntPtr, lpszDevice : IntPtr, lpszOutput : IntPtr, lpInitData : IntPtr) : IntPtr
+DeleteDC(hdc : IntPtr) : IntPtr

รูปที่ 3-42 Class ASM.Component.Win32

FormUserManagement
+fromMain : FromMain
+UserHash : string
-Role : string
-listView1 : ListView
-contextMenu1 : ContextMenu
-columnUsername : ColumnHeader
-columnRole : ColumnHeader
-components : IContainer
-menuAdd : MenuItem
-menuRemove : MenuItem
-menuChangePassword : MenuItem
-menuRoleAdmin : MenuItem
-menuRoleGuest : MenuItem
-menuRole : MenuItem
-contextMenu2 : ContextMenu
-menuAdd2 : MenuItem
-users : ASM.Component.Admin.User[]
+FormUserManagement()
#Dispose(disposing : bool)
-InitializeComponent()
-listView1_SelectedIndexChanged(sender : object, e : EventArgs)
+LoadUser()
+GetRole()
-menuRemove_Click(sender : object, e : EventArgs)
-menuChangePassword_Click(sender : object, e : EventArgs)
-menuRoleAdmin_Click(sender : object, e : EventArgs)
-menuRoleGuest_Click(sender : object, e : EventArgs)
-menuAdd_Click(sender : object, e : EventArgs)
+CheckRole()
-Hash(stringHash : string) : string
-menuAdd2_Click(sender : object, e : EventArgs)

รูปที่ 3-43 Class ASM.Console.FormUserManagement

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<b>FormSystemSummary</b>
+ClientName : string -client : ASM.RemoteObject.Admin.Client -components : Container = null #menuItem22 : MenuItem #menuItem18 : MenuItem #menuItem17 : MenuItem #menuItem16 : MenuItem #printDialog1 : PrintDialog #textArea : RichTextBox #printMenu : MenuItem #pageSetupMenu : MenuItem #menuItem1 : MenuItem #mainMenu : MainMenu #printPreviewMenu : MenuItem -storedPageSettings : PageSetting = null
+FormSystemSummary() #Dispose(disposing : bool) -InitializeComponent() -PrintMenu_Click(sender : object, e : EventArgs) -PageSetupMenu_Click(sender : object, e : EventArgs) -PrintPreviewMenu_Click(sender : object, e : EventArgs) +LoadSummary() +PrintSummary()

รูปที่ 3-44 Class ASM.Console.FormSystemSummary

<b>FormProperties</b>
+ClientName : string -myClient : ASM.RemoteObject.Client -treeView : TreeView -components : Container = null
+FormProperties() #Dispose(disposing : bool) -InitializeComponent() -FormProperties_SizeChanged(sender : object, e : EventArgs) +LoadClient() +RefreshTree()

รูปที่ 3-45 Class ASM.Console.FormProperties

<b>FormImageRotator</b>
-index : int -myClient : string[] -timer1 : Timer -components : IContainer = null
+FormImageRotator() #Dispose(disposing : bool) -InitializeComponent() +LoadImage() +LoadClient() -timer1_Tick(sender : object, e : EventArgs)

รูปที่ 3-46 Class ASM.Console.FormImageRotator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<b>FormImage</b>
+pictureBox : PictureBox -components : Container = null =bmp : Bitmap =scale : double
+FormImage() #Dispose(disposing : bool) -InitializeComponent () -FormImage_SizeChanged (sender : object, e : EventArgs ) +LoadImage ()

รูปที่ 3-47 Class ASM.Console.FormImage

<b>FormChangPassword</b>
+ClientName : string -label1 : Label -label2 : Label -label3 : Label -textOldPass : TextBox -textNewPass : TextBox -textConfirmPass : TextBox -buttonEnter : Button -buttonClear : Button -buttonCancel : Button -components : Container = null
+FormChangePassword () #Dispose(disposing : bool) -InitializeComponent () -buttonCancel_Click(sender : object, e : EventArgs ) -buttonClear_Click(sender : object, e : EventArgs ) -Hash(stringHash : string) : string -buttonEnter_Click(sender : object, e : EventArgs )

รูปที่ 3-48 Class ASM.Console.FormChangePassword

<b>TextPrintDocument</b>
-printFont : Font = null -streamToPrint : StringReader = null -overflowText : string =null
+TextPrintDocument(streamToPrint : StringReader, printFont : Font) #OnBeginPrint(ev : PrintEventArgs) #OnPrintPage(ev : PrintPageEventArgs) -PrintLine(e : PrintPageEventArgs, text : string, yStartPos : float) : int

รูปที่ 3-49 Class ASM.Console.TextPrintDocument

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Client
-connectionString : string -connection : SqlConnection
+Client() +LoadClient() : string[] +LoadClient() : string[] +LoadBaseBoard (name : string) : ASM.Component.BaseBoard [] +LoadBIOS(name : string) : ASM.Component.BIOS [] +LoadCacheMemory (name : string) : ASM.Component.CacheMemory [] +LoadCDROMDrive (name : string) : ASM.Component.CDROMDrive [] +LoadComputerSystem (name : string) : ASM.Component.ComputerSystem [] +LoadDesktop (name : string) : ASM.Component.Desktop [] +LoadDesktopMonitor (name : string) : ASM.Component.DesktopMonitor [] +LoadDiskDrive (name : string) : ASM.Component.DiskDrive [] +LoadEnvironment (name : string) : ASM.Component.Environment [] +LoadFloppyDrive (name : string) : ASM.Component.FloppyDrive [] +LoadGroup(name : string) : ASM.Component.Group [] +LoadLocalTime (name : string) : ASM.Component.Group [] +LoadLogicalDisk (name : string) : ASM.Component.LogicalDisk [] +LoadNetworkAdapter (name : string) : ASM.Component.NetworkAdapter [] +LoadNetworkAdapterConfiguration (name : string) : ASM.Component.NetworkAdapterConfiguration [] +LoadNetworkLoginProfile (name : string) : ASM.Component.NetworkLoginProfile [] +LoadOperatingSystem (name : string) : ASM.Component.OperatingSystem [] +LoadPageFile (name : string) : ASM.Component.PageFile [] +LoadPageFileUsage (name : string) : ASM.Component.PageFileUsage [] +LoadPhysicalMemory (name : string) : ASM.Component.PhysicalMemory [] +LoadPOTSModem (name : string) : ASM.Component.POTSModem [] +LoadPrinter (name : string) : ASM.Component.Printer [] +LoadProcess (name : string) : ASM.Component.Process [] +LoadProcessor (name : string) : ASM.Component.Processor [] +LoadService (name : string) : ASM.Component.Service [] +LoadShare (name : string) : ASM.Component.Share [] +LoadSoftware (name : string) : ASM.Component.Software [] +LoadStartupCommand (name : string) : ASM.Component.StartupCommand [] +LoadSystemEnclosure (name : string) : ASM.Component.SystemEnclosure [] +LoadSystemSlot (name : string) : ASM.Component.SystemSlot [] +LoadTimeZone (name : string) : ASM.Component.TimeZone [] +LoadUserAccount (name : string) : ASM.Component.UserAccount [] +LoadVideoController (name : string) : ASM.Component.VideoController []

รูปที่ 3-50 Class ASM.DataAccess.Client

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<b>ComputerSystem</b>
<pre> -connectionString : string -connection : SqlConnection -name : string -adminPasswordStatus : int -automaticResetBootOption : string -automaticResetCapability : string -bootROMSupported : string -bootupState : string -chassisBootupState : string -currentTimeZone : string -domain : string -domainRole : int -enabledDaylightSavingsTime : string -frontPanelResetStatus : int -networkServerModeEnabled : string -numberOfProcessors : int -pauseAfterReset : long -powerState : int -powerSupplyState : int -primaryOwnerName : string -roles : string -systemStartupOptions : string -systemStartupSetting : int -systemType : string -thermalState : int -totalPhysicalMemory : long -userName : string -wakeUpType : int -workgroup : string -baseboard : ASM.Component.BaseBoard -bios : ASM.Component.BIOS -localTime : ASM.Component.LocalTime -operatingSystem : ASM.Component.OperatingSystem -timeZone : ASM.Component.TimeZone -systemEnclosure : ASM.Component.SystemEnclosure -cacheMemory : ASM.Component.CacheMemory [] -cdromDrive : ASM.Component.CDROMDrive [] -desktop : ASM.Component.Desktop [] -desktopMonitor : ASM.Component.DesktopMonitor [] -diskDrive : ASM.Component.DiskDrive [] -environment : ASM.Component.Environment [] -floppyDrive : ASM.Component.FloppyDrive [] -group : ASM.Component.Group [] -logicalDisk : ASM.Component.LogicalDisk [] -networkAdapter : ASM.Component.NetworkAdapter [] -networkAdapterConfiguration : ASM.Component.NetworkAdapterConfiguration [] -networkLoginProfile : ASM.Component.NetworkLoginProfile [] -pageFile : ASM.Component.PageFile [] -pageFileUsage : ASM.Component.PageFileUsage [] -physicalMemory : ASM.Component.PhysicalMemory [] -potsModem : ASM.Component.POTSModem [] -printer : ASM.Component.Printer [] -process : ASM.Component.Process [] -processor : ASM.Component.Processor [] -service : ASM.Component.Service [] -share : ASM.Component.Share [] -software : ASM.Component.Software [] -startupCommand : ASM.Component.StartupCommand [] -systemSlot : ASM.Component.SystemSlot [] -userAccount : ASM.Component.UserAccount [] -videoController : ASM.Component.VideoController []  +Client() +LoadClient() : string[] +Load() +Add() +Delete() </pre>

รูปที่ 3-51 Class ASM.DataAccess.ComputerSystem

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<b>Event</b>
-connectionString : string -connection : SqlConnection -watcher : ASM.Component.WatcherType[] +WatcherType : ASM.Component.WatcherType[]
+Event() +Load(name : string)

รูปที่ 3-52 Class ASM.DataAccess.Event

<b>EventAdmin</b>
-connectionString : string -connection : SqlConnection -watcher : ASM.Component.WatcherType[] +WatcherType : ASM.Component.WatcherType[]
+EventAdmin() +Add(watcherType : ASM.Component.WatcherType)

รูปที่ 3-53 Class ASM.DataAccess.EventAdmin

<b>Screen</b>
-connectionString : string -connection : SqlConnection -name : string -picture : byte[] -time : DateTime +Name : string +Picture : byte[] +Time : DateTime
+Screen() +Delete() +Save() +Load() +Load(name : string)

รูปที่ 3-54 Class ASM.DataAccess.Screen

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<b>User</b>
-connectionString : string -connection : SqlConnection
+User() +Add(username : string, password : string, userhash : string, passhash : string, role : string) +Remove(userhash : string) +Update(userhash : string, password : string, passhash : string) +Update(userhash : string, role : string) +IsExist(userhash : string, passhash : string) : bool +IsExist(userhash : string) : bool +LoadAll() : ASM.Component.Admin.User[] +GetRole(userhash : string) : string

รูปที่ 3-55 Class ASM.DataAccess.User

<<Serializable>> <<Remotable>> <b>Authenticator</b>
-username : string -password : string -userhash : string -passhash : string -role : string +Username : string +Password : string +UserHash : string +Password : string +Role : string
+Authenticator() +Update( userhash : string, password : string, passhash : string) +Update( userhash : string, role : string) +Add(username : string, password : string, userhash : string, passhash : string, role : string) +Remove(userhash : string) +IsExist() : bool +IsExist(userhash : string) : bool +IsExist(userhash : string) : bool +IsExist(userhash : string, passhash : string) : bool +LoadAllUser () : ASM.Component.Admin.User [] +GetRole(userhash : string) : string

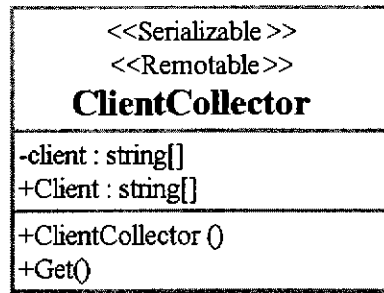
รูปที่ 3-56 Class ASM.RemoteObject.Authenticator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

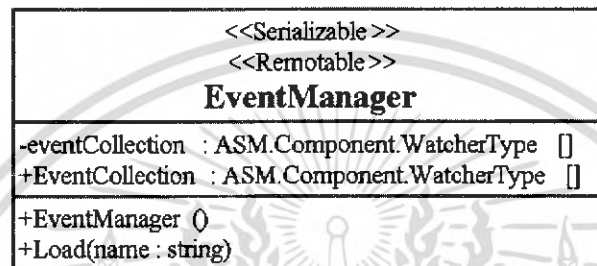
<<Serializable >> <<Remotable >> <b>Client</b>
-baseBoard : ASM.Component.BaseBoard [] -bios : ASM.Component.BIOS [] -cacheMemory : ASM.Component.CacheMemory [] -cdromDrive : ASM.Component.CDROMDrive [] -computerSystem : ASM.Component.ComputerSystem [] -desktop : ASM.Component.Desktop [] -desktopMonitor : ASM.Component.DesktopMonitor [] -diskDrive : ASM.Component.DiskDrive [] -environment : ASM.Component.Environment [] -floppyDrive : ASM.Component.FloppyDrive [] -group : ASM.Component.Group [] -localTime : ASM.Component.LocalTime [] -logicalDisk : ASM.Component.LogicalDisk [] -networkAdapter : ASM.Component.NetworkAdapter [] -networkAdapterConfiguration : ASM.Component.NetworkAdapterConfiguration [] -networkLoginProfile : ASM.Component.NetworkLoginProfile [] -operatingSystem : ASM.Component.OperatingSystem [] -pageFile : ASM.Component.Pagefile [] -pageFileUsage : ASM.Component.PageFileUsage [] -physicalMemory : ASM.Component.PhysicalMemory [] -potsModem : ASM.Component.POTSModem [] -printer : ASM.Component.Printer [] -process : ASM.Component.Process [] -processor : ASM.Component.Processor [] -service : ASM.Component.Service [] -share : ASM.Component.Share [] -software : ASM.Component.Software [] -startupCommand : ASM.Component.StartupCommand [] -systemEnclosure : ASM.Component.SystemEnclosure [] -systemSlot : ASM.Component.SystemSlot [] -timeZone : ASM.Component.TimeZone [] -userAccount : ASM.Component.UserAccount [] -videoController : ASM.Component.VideoController [] +BaseBoard : ASM.Component.BaseBoard [] +BIOS : ASM.Component.BIOS [] +CacheMemory : ASM.Component.CacheMemory [] +CDROMDrive : ASM.Component.CDROMDrive [] +ComputerSystem : ASM.Component.ComputerSystem [] +Desktop : ASM.Component.Desktop [] +DesktopMonitor : ASM.Component.DesktopMonitor [] +DiskDrive : ASM.Component.DiskDrive [] +Environment : ASM.Component.Environment [] +FloppyDrive : ASM.Component.FloppyDrive [] +Group : ASM.Component.Group [] +LocalTime : ASM.Component.LocalTime [] +LogicalDisk : ASM.Component.LogicalDisk [] +NetworkAdapter : ASM.Component.NetworkAdapter [] +NetworkAdapterConfiguration : ASM.Component.NetworkAdapterConfiguration [] +NetworkLoginProfile : ASM.Component.NetworkLoginProfile [] +OperatingSystem : ASM.Component.OperatingSystem [] +PageFile : ASM.Component.PageFile [] +PageFileUsage : ASM.Component.PageFileUsage [] +PhysicalMemory : ASM.Component.PhysicalMemory [] +POTSModem : ASM.Component.POTSModem [] +Printer : ASM.Component.Printer [] +Process : ASM.Component.Process [] +Processor : ASM.Component.Processor [] +Service : ASM.Component.Service [] +Share : ASM.Component.Share [] +Software : ASM.Component.Software [] +StartupCommand : ASM.Component.StartupCommand [] +SystemEnclosure : ASM.Component.SystemEnclosure [] +SystemSlot : ASM.Component.SystemSlot [] +TimeZone : ASM.Component.TimeZone [] +UserAccount : ASM.Component.UserAccount [] +VideoController : ASM.Component.VideoController [] +Client() +Load(name : string)

รูปที่ 3-57 Class ASM.RemoteObject.Client

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-58 Class ASM.RemoteObject.ClientCollector



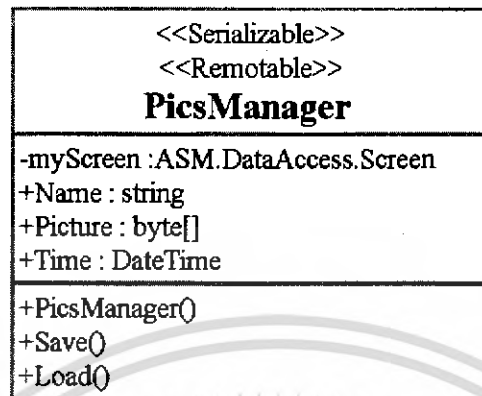
รูปที่ 3-59 Class ASM.RemoteObject.EventManager

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

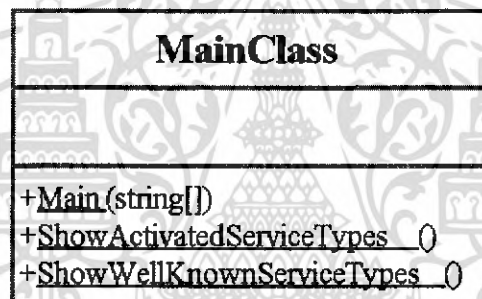
<<Serializable>> <<Remotable>> <b>Computer</b>
-baseBoard : ArrayList -bios : ArrayList -cacheMemory : ArrayList -cdromDrive : ArrayList -computerSystem : ArrayList -desktop : ArrayList -desktopMonitor : ArrayList -diskDrive : ArrayList -environment : ArrayList -floppyDrive : ArrayList -group : ArrayList -localTime : ArrayList -logicalDisk : ArrayList -networkAdapter : ArrayList -networkAdapterConfiguration : ArrayList -networkLoginProfile : ArrayList -operatingSystem : ArrayList -pageFile : ArrayList -pageFileUsage : ArrayList -physicalMemory : ArrayList -potsModem : ArrayList -printer : ArrayList -process : ArrayList -processor : ArrayList -service : ArrayList -share : ArrayList -software : ArrayList -startupCommand : ArrayList -systemEnclosure : ArrayList -systemSlot : ArrayList -timeZone : ArrayList -userAccount : ArrayList -videoController : ArrayList +BaseBoard : ASM.Component.BaseBoard[] +BIOS : ASM.Component.BIOS[] +CacheMemory : ASM.Component.CacheMemory[] +CDROMDrive : ASM.Component.CDROMDrive[] +ComputerSystem : ASM.Component.ComputerSystem[] +Desktop : ASM.Component.Desktop[] +DesktopMonitor : ASM.Component.DesktopMonitor[] +DiskDrive : ASM.Component.DiskDrive[] +Environment : ASM.Component.Environment[] +FloppyDrive : ASM.Component.FloppyDrive[] +Group : ASM.Component.Group[] +LogicalDisk : ASM.Component.LogicalDisk[] +LocalTime : ASM.Component.LocalTime[] +NetworkAdapter : ASM.Component.NetworkAdapter[] +NetworkAdapterConfiguration : ASM.Component.NetworkAdapterConfiguration[] +NetworkLoginProfile : ASM.Component.NetworkLoginProfile[] +OperatingSystem : ASM.Component.OperatingSystem[] +PageFile : ASM.Component.PageFile[] +PageFileUsage : ASM.Component.PageFileUsage[] +PhysicalMemory : ASM.Component.PhysicalMemory[] +POTSModem : ASM.Component.POTSModem[] +Printer : ASM.Component.Printer[] +Process : ASM.Component.Process[] +Processor : ASM.Component.Processor[] +Service : ASM.Component.Service[] +Share : ASM.Component.Share[] +Software : ASM.Component.Software[] +StartupCommand : ASM.Component.StartupCommand[] +SystemEnclosure : ASM.Component.SystemEnclosure[] +SystemSlot[] : ASM.Component.SystemSlot[] +TimeZone[] : ASM.Component.TimeZone[] +UserAccount[] : ASM.Component.UserAccount[] +VideoController[] : ASM.Component.VideoController[]
+Computer +Save()

รูปที่ 3-60 Class ASM.RemoteObject.Computer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



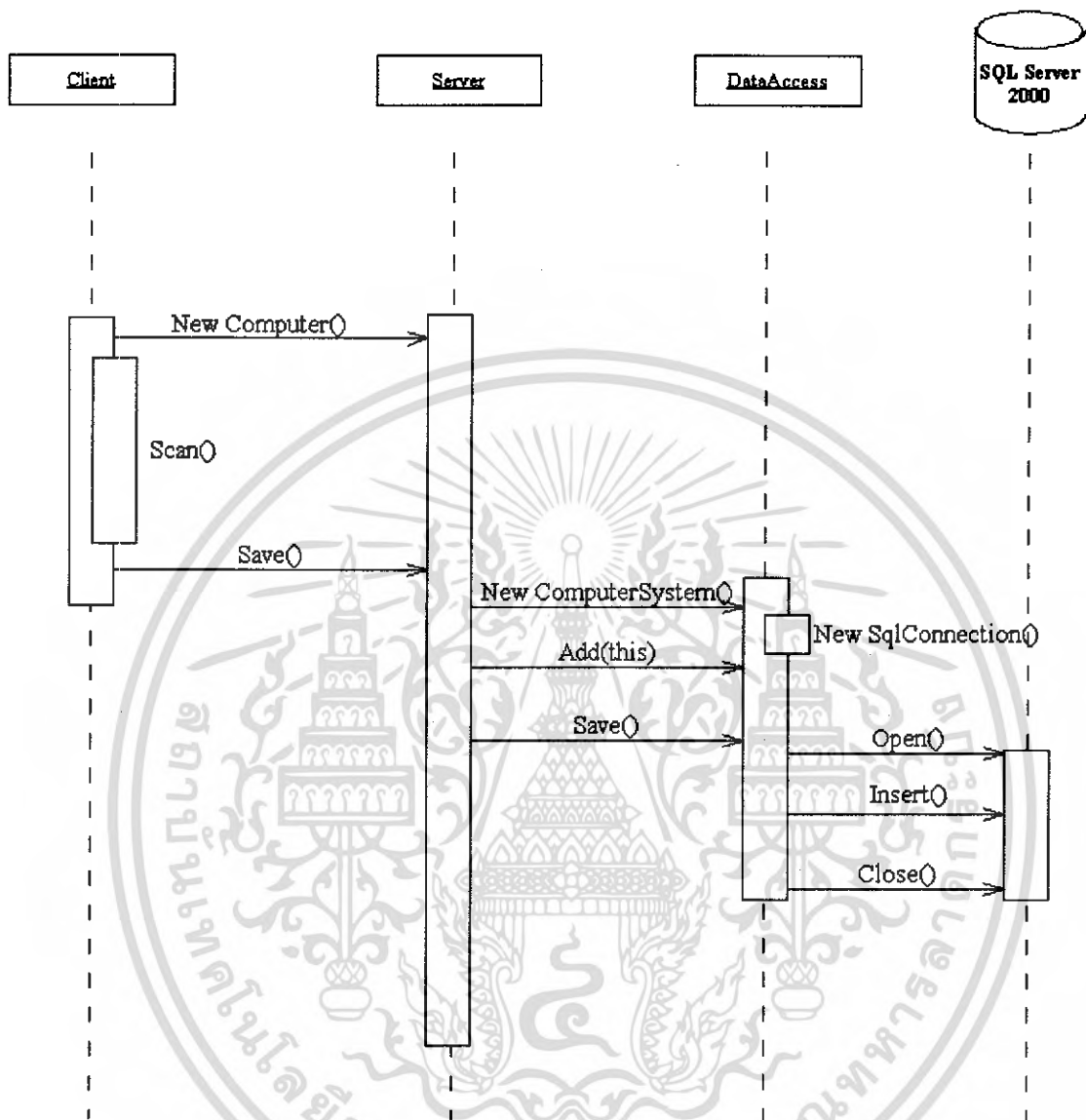
รูปที่ 3-62 Class ASM.RemoteObject.PicsManager



รูปที่ 3-63 Class ASM.Server.MainClass

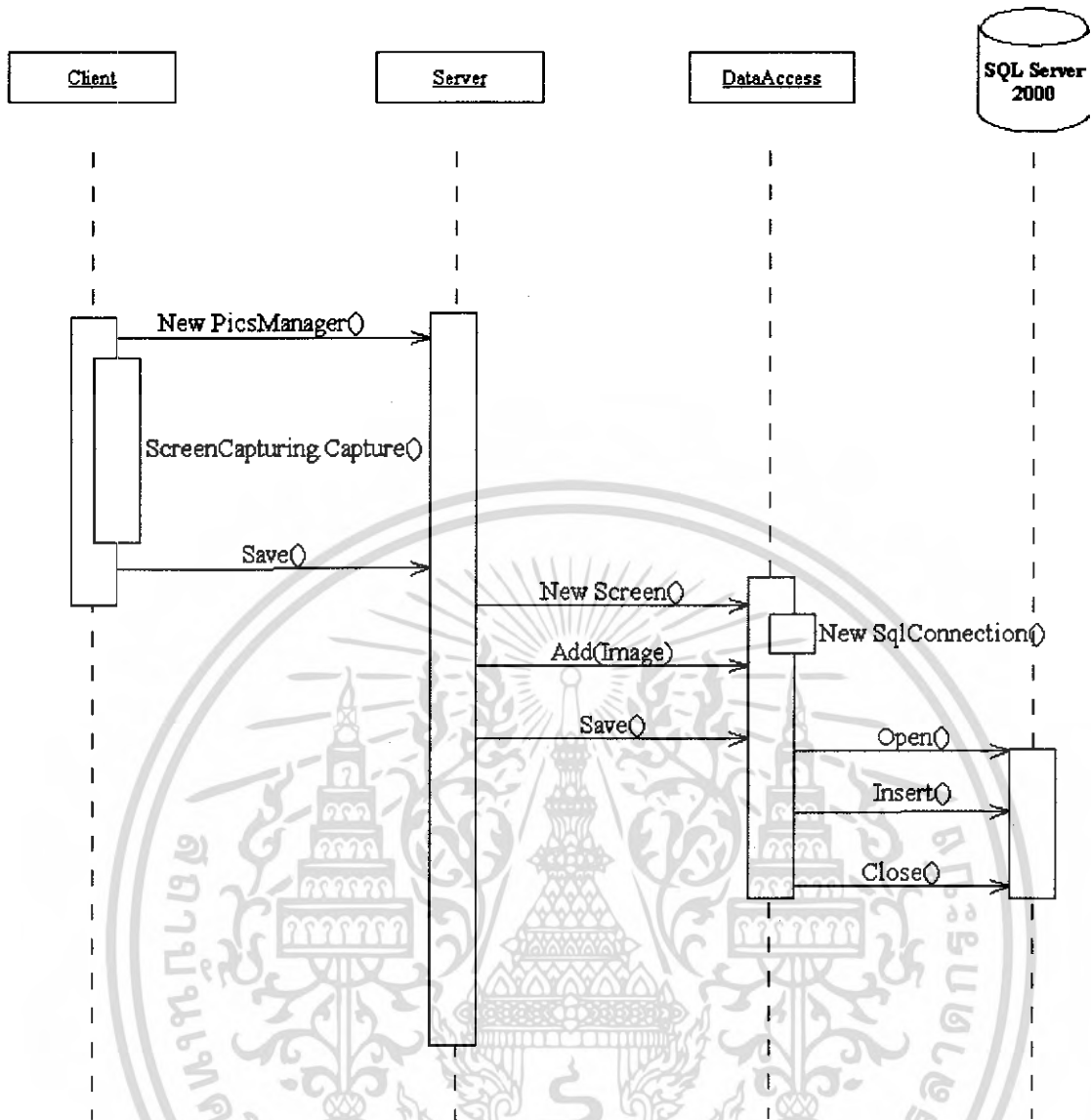
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3 Sequence Diagram



รูปที่ 3-64 Sequence Diagram ในส่วนที่เกี่ยวกับการเก็บข้อมูลจากเครื่องไคลเอนท์

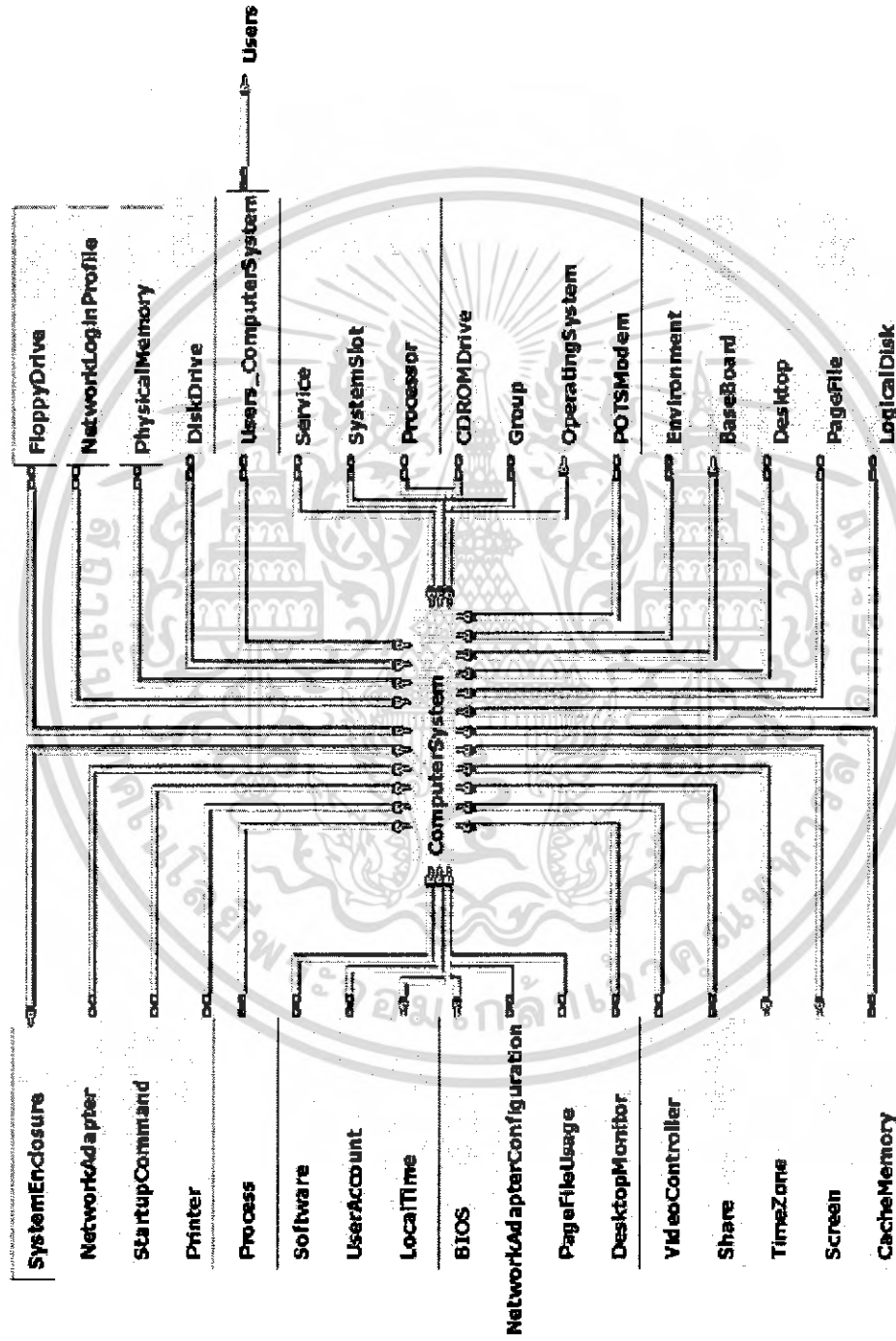
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-65 Sequence Diagram ในส่วนที่เกี่ยวกับการดักจับภาพจากเครื่องไคลเอนท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4 ER - Diagram



รูปที่ 3-66 ER-Diagram รวมทั้งระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.5 Data Dictionary

Table	Field	Data Type	Key
BaseBoard	Name HotSwappable Manufacturer Product SerialNumber Version	Char(255) Char(10) VarChar(255) VarChar(255) VarChar(255) VarChar(255)	P
BIOS	Name CurrentLanguage InstallableLanguage ListOfLanguage Manufacturer BIOS_Name PrimaryBIOS ReleaseDate SerialNumber SMBIOSBIOSVersion SMBIOSMajorVersion SMBIOSMinorVersion SMBIOSPresent SoftwareElementState Version	Char(255) VarChar(255) int VarChar(255) VarChar(255) VarChar(255) VarChar(255) VarChar(255) VarChar(255) VarChar(255) int int int Char(10) int VarChar(255)	P
CacheMemory	Name CacheMemoryID Associativity BlockSize CacheType CurrentSRAM DeviceID InstalledSize Levels Location MaxCacheSize NumberOfBlocks Purpose SupportedSRAM WritePolicy	Char(255) int int int int int VarChar(255) int int int int int int VarChar(255) int int	P P
CDROM	Name CDROMDrive_Id Drive FileSystemFlagEx MaximumComponentLength MediaLoaded CDROMDrive_Name SCSIBus SCSILogicalUnit SCSIPort SCSITargetId Sizes TransferRate	Char(255) int VarChar(255) VarChar(255) int Char(10) VarChar(255) int int int int bigint real	P P

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table	Field	Data Type	Key
	VolumeName	VarChar(255)	
	VolumeSerialNumber	VarChar(255)	
ComputerSystem	Name	Char(255)	P
	AdminPasswordStatus	int	
	AutomaticResetBootOption	Char(10)	
	AutomaticResetCapability	Char(10)	
	BootROMSupported	Char(10)	
	BootupState	VarChar(255)	
	ChassisBootupState	int	
	CurrentTimeZone	int	
	Domains	Char(10)	
	DomainRole	int	
	EnableDaylightSavingsTime	Char(10)	
	FrontPanelResetStatus	int	
	NetworkServerModeEnabled	Char(10)	
	NumberOfProcessors	int	
	PauseAfterReset	bigint	
	PowerState	int	
	PowerSupplyState	int	
	PrimaryOwnerName	VarChar(255)	
	Roles	VarChar(255)	
	SystemStartupDelay	int	
	SystemStartupOptions	VarChar(255)	
	SystemStartupSetting	int	
	SystemType	VarChar(255)	
	ThermalState	int	
	TotalPhysicalMemory	int	
	UserName	bigint	
	WakeUpType	int	
	Workgroup	VarChar(255)	
Desktop	Name	Char(255)	P
	DesktopID	int	P
	BorderWidth	int	
	CoolSwitch	Char(10)	
	CursorBlinkRate	int	
	DragFullWindows	Char(10)	
	GridGranularity	int	
	IconSpacing	int	
	IconTitleFaceName	VarChar(255)	
	IconTitleSize	int	
	IconTitleWrap	Char(10)	
	Desktop_Name	VarChar(255)	
	ScreenSaverActive	Char(10)	
	ScreenSaverExecutable	VarChar(255)	
	ScreenSaverSecure	VarChar(255)	
	ScreenSaverTimeout	int	
	WallPaper	VarChar(255)	
	WallPaperStretched	Char(10)	
	WallPaperTiled	Char(10)	
DesktopMonitorID	Name	Char(255)	P
	DesktopMonitorID	int	P
	DeviceID	VarChar(255)	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table	Field	Data Type	Key
	MonitorManufacturer DesktopMonitor_Name PixelsPerXLogicalInch PixelsPerYLogicalInch ScreenHeight ScreenWidth	VarChar(255) VarChar(255) int int int int	
DiskDrive	Name DiskDrive_Id BytesPerSector Caption Indexs InterfaceType Manufacturer MediaType Model DiskDrive_Name Partitions SCSIBus SCSILogicalUnit SCSIPort SCSITargetId SectorPerTrack Signature Sizes TotalCylinders TotalHeads TotalSectors TotalTracks TracksPerCylinder	Char(255) int int VarChar(255) int VarChar(255) VarChar(255) VarChar(255) VarChar(255) VarChar(255) int int int int int int bigint bigint bigint bigint bigint bigint int	P P
Environment	Name EnvironmentID Caption Environment_Name SystemVariable UserName VariableName	Char(255) int VarChar(255) VarChar(255) Char(10) VarChar(255) VarChar(512)	P P
Event	Name Event_Id Class EventType EventParameter Days Hours Minutes Seconds	Char(255) int VarChar(255) VarChar(255) VarChar(1024) int int int int	P P
FloppyDrive	Name FloppyDrive_Id FloppyDrive_Name Manufacturer	Char(255) int VarChar(255) VarChar(255)	P P
Group	Name GroupID Caption	Char(255) int VarChar(255)	P P

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table	Field	DataType	Key
	Description Domains LocalAccount Group_Name SID SIDType	VarChar(255) VarChar(255) Char(10) VarChar(255) VarChar(255) VarChar(255)	
LocalTime	Name Days DayOfWeek Hours Milliseconds Minutes Months Quarter Seconds WeekInMonth Years	Char(255) int int int int int int int int int int	P
LogicalDisk	Name LogicalDisk_Id Compressed Description DriveType FileSystem FreeSpace MaximumComponentLength MediaType LogicalDisk_Name Sizes SupportsFileBasedCompression VolumnName VolumnSerialNumber	Char(255) int Char(10) VarChar(255) int VarChar(255) bigint int int VarChar(255) bigint Char(10) VarChar(255) VarChar(255)	P P
NetworkAdapter	Name NetworkAdapter_Id AdapterType Availability Caption DeviceID Indexs MACAddress Manufacturer NetworkAdapter_Name NetConnectionID NetConnectionStatus ServiceName	Char(255) int VarChar(255) int VarChar(255) int int VarChar(255) VarChar(255) VarChar(255) VarChar(255) int VarChar(255)	P P
NetworkAdapterConfiguration	Name NetworkAdapterConfiguration_Id Caption DatabasePath DefaultIPGateway Description DHCPEnabled DHCPSEServer	Char(255) int VarChar(255) VarChar(255) VarChar(255) VarChar(255) Char(10) VarChar(255)	P P

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table	Field	Data Type	Key
	DNSDomain DNSDomainSuffixSearchOrder DNSEnabledForWINSResolution DNSHostName DNSServerSearchOrder FullDNSRegistrationEnabled GatewayCostMetric IGMPLevel Indexs IPAddress IPConnectionMetric IPEnabled IPFilterSecurityEnabled IPPortSecurityEnabled IPsecPermitIPProtocols IPsecPermitTCPPorts IPsecPermitUDPPorts IPSubnet IPXAddress IPXEnabled IPXFrameType IPXMediaType IPXVirtualNetNumber MACAddress TcipNetbiosOptions TcpWindowSize WINSEnabledLMHostsLookup	VarChar(255) VarChar(255) Char(10) VarChar(255) VarChar(255) Char(10) VarChar(255) VarChar(255) int VarChar(255) int Char(10) Char(10) Char(10) int int int VarChar(255) VarChar(255) Char(10) VarChar(255) Char(10) VarChar(255) VarChar(255) int int Char(10)	
NetworkLoginProfile	Name NetworkLoginProfile_Id AccountExpires AuthorizationFlags BadPasswordCount Caption Description Flags LastLogoff LastLogon LogonServer MaximunStorage NetworkLoginProfile_Name NumberOfLogons PasswordAge PasswordExpires PrimaryGroupld Privillegess UnitsPerWeek UserComment UserId UserType	Char(255) int VarChar(255) int int VarChar(255) VarChar(255) bigint VarChar(255) VarChar(255) VarChar(255) bigint VarChar(255) int VarChar(255) VarChar(255) int int int VarChar(255) int VarChar(255)	P P
OperatingSystem	Name BootDevice BuildNumber	Char(255) VarChar(255) int	P

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table	Field	Data Type	Key
	Caption	VarChar(255)	
	CodeSet	int	
	CountryCode	int	
	CSDVersion	VarChar(255)	
	CSName	VarChar(255)	
	CurrentTimeZone	int	
	Debug	Char(10)	
	Distributeds	Char(10)	
	EncryptionLevel	int	
	FreePhysicalMemory	bigint	
	FreeSpaceInPagingFiles	bigint	
	FreeVirtualMemory	bigint	
	LocalDateTime	VarChar(255)	
	LastBootupTime	VarChar(255)	
	Locale	VarChar(255)	
	Manufacturer	VarChar(255)	
	MaxNumberOfProcesses	bigint	
	OperatingSystem_Name	VarChar(255)	
	NumberOfProcesses	int	
	NumberOfUsers	int	
	Organization	VarChar(255)	
	OSLanguage	int	
	OSType	int	
	Primarys	Char(10)	
	RegisteredUser	VarChar(255)	
	SerialNumber	VarChar(255)	
	ServicePackMajorVersion	int	
	ServicePackMinorVersion	int	
	SizeStoredInPagingFiles	bigint	
	SuiteMask	int	
	SystemDevice	VarChar(255)	
	SystemDirectory	VarChar(255)	
	TotalSwapSpaceSize	bigint	
	TotalVirtualMemorySize	bigint	
	TotalVisibleMemorySize	bigint	
	Version	VarChar(255)	
	WindowsDirectory	VarChar(255)	
PageFile	Name	Char(255)	P
	PageFileID	int	P
	Archive	Char(10)	
	Compressed	Char(10)	
	CreationDate	VarChar(255)	
	Drive	VarChar(255)	
	EightDotThreeFileName	VarChar(255)	
	Encrypted	Char(10)	
	EncryptionMethod	VarChar(255)	
	Extension	VarChar(255)	
	FileName	VarChar(255)	
	FileSize	bigint	
	FileType	VarChar(255)	
	FSName	VarChar(255)	
	Hidden	Char(10)	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table	Field	Data Type	Key
	InitialSize InUseCount MaximumSize PageFile_Name Path Readable System Writeable	bigint bigint bigint VarChar(255) VarChar(255) Char(10) Char(10) Char(10)	
PageFileUsage	Name PageFileUsage_Id AllocatedBaseSize CurrentUsage PageFileUsage_Name PeakUsage TempPageFile	Char(255) int bigint bigint VarChar(255) bigint Char(10)	P P
PhysicalMemory	Name PhysicalMemory_Id BankLabel Capacity DataWidth DeviceLocator FormFactor Manufacturer MemoryType PositionInRow Tag TotalWidth TypeDetail	Char(255) int VarChar(255) bigint int VarChar(255) int VarChar(255) int VarChar(255) VarChar(255) int int int	P P
POTSModem	Name POTSModem_Id AttachedTo CompressionOff CompressionOn CountrySelected DeviceID DeviceType Indexs MaxBaudRateToPhone MaxBaudRateToSerialPort Model Name RingsBeforeAnswer	Char(255) int VarChar(255) VarChar(255) VarChar(255) VarChar(255) VarChar(255) VarChar(255) int int int VarChar(255) VarChar(255) int	P P
Printer	Name Printer_Id DeviceID DriverName HorizontalResolution JobCountSinceLastReset Printer_Name PortName PrinterStatus SharedName	Char(255) int VarChar(255) VarChar(255) int int VarChar(255) VarChar(255) int VarChar(255)	P P

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table	Field	DataType	Key
	VerticalResolution	int	
Process	Name	Char(255)	P
	Process_Id	int	P
	CommandLine	VarChar(255)	
	ExecutablePath	VarChar(255)	
	ExecutableState	int	
	Handle	bigint	
	HandleCount	int	
	KernelModeTime	bigint	
	MaximumWorkingSetSize	bigint	
	MinimumWorkingSetSize	bigint	
	Process_Name	VarChar(255)	
	OtherOperationCount	bigint	
	OtherTransferCount	bigint	
	PageFaults	int	
	PageFileUsage	bigint	
	ParentProcessId	int	
	PeakPageFileUsage	bigint	
	Priority	int	
	PrivatePageCount	bigint	
	ProcessId	bigint	
	QuotaNonPagedPoolUsage	bigint	
	QuotaPagedPoolUsage	bigint	
	QuotaPeakNonPagedPoolUsage	bigint	
	QuotaPeakPagedPoolUsage	bigint	
	ReadOperationCount	bigint	
	ReadTransferCount	bigint	
	SessionId	int	
	UserModeTime	bigint	
	VirtualSize	bigint	
	WindowsVersion	VarChar(255)	
	WriteOperationCount	bigint	
	WriteTransferCount	bigint	
Processor	Name	Char(255)	P
	Processor_Id	int	P
	AddressWidth	int	
	Architecture	int	
	Caption	VarChar(255)	
	CpuStatus	int	
	CurrentClockSpeed	int	
	DataWidth	int	
	DeviceID	VarChar(255)	
	ExtClock	int	
	Family	int	
	Levels	int	
	LoadPercentage	int	
	Manufacturer	VarChar(255)	
	MaxClockSpeed	int	
	Processor_Name	VarChar(255)	
	ProcessorId	VarChar(255)	
	ProcessorType	int	
	Revision	int	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table	Field	DataType	Key
	Role SocketDesignation Stepping Version	VarChar(255) VarChar(255) int VarChar(255)	
Screen	Name ScreenShot ScreenTime	Char(255) image datetime	P
Service	Name ServiceID AcceptPause AcceptStop Description DesktopInteract DisplayName ErrorControl ExitCode Service_Name PathName ProcessId ServiceSpecificExitCode ServiceType Started StartMode StartName State	Char(255) int Char(10) Char(10) VarChar(255) Char(10) VarChar(255) VarChar(255) int int VarChar(255) VarChar(255) int int VarChar(255) Char(10) VarChar(255) VarChar(255) VarChar(255)	P P
Shares	Name Shares_Id AccessMask AllowMaximum Caption Description MaximumAllowed Share_Name Path Type	Char(255) int VarChar(255) Char(10) VarChar(255) VarChar(255) VarChar(255) VarChar(255) VarChar(255) VarChar(255) bigint	P P
Software	Name Software_Id Software_Name Version InstallDate InstallLocation Publisher QuietUninstallString UninstallString	Char(255) int VarChar(255) VarChar(255) VarChar(255) VarChar(255) VarChar(255) VarChar(255) VarChar(255)	P P
StartupCommand	Name StartupCommandID Command Location StartCommand_Name Users	Char(255) int VarChar(255) VarChar(255) VarChar(255) VarChar(255)	P P
SystemEnclosure	Name	Char(255)	P

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table	Field	Data Type	Key
	ChassisTypes	int	
SystemSlot	Name SystemSlot_Id CurrentUsage MaxDataWidth Number PMESignal Shared SlotDesignation SupportHotPlug Tag	Char(255) int int int int Char(10) Char(10) VarChar(255) Char(10) VarChar(255)	P P
TimeZone	Name Caption StandardName	Char(255) VarChar(255) VarChar(255)	P
UserAccount	Name UserAccount_Id AccountType Caption Description Disabled Domains FullName LocalAccount LockOut UserAccount_Name PasswordChangeable PasswordExpired PasswordRequired SID SIDType	Char(255) int int VarChar(255) VarChar(255) Char(10) VarChar(255) VarChar(255) Char(10) Char(10) VarChar(255) Char(10) Char(10) Char(10) VarChar(255) int	P P
Users	UserName Password UserHash PassHash Role	VarChar(255) VarChar(255) VarChar(255) VarChar(255) VarChar(50)	P
Users_ComputerSystem	Name UserName isRead isCommand	Char(255) VarChar(255) Char(10) Char(10)	P P
VideoController	Name VideoController_Id AdapterCompatibility AdapterDACType AdapterRAM Caption CurrentBitsPerPixel CurrentHorizontalResolution CurrentNumberOfColors CurrentRefreshRate CurrentScanMode CurrentVerticalResolution	Char(255) int VarChar(255) VarChar(255) int Char(255) int int int bigint int int	P P

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table	Field	DataType	Key
	Description	VarChar(255)	
	DeviceID	VarChar(255)	
	DriverDate	VarChar(255)	
	DriverVersion	VarChar(255)	
	InstalledDisplayDrivers	VarChar(255)	
	MaxRefreshRate	int	
	MinRefreshRate	int	
	MonoChrome	Char(10)	
	VideoController_Name	VarChar(255)	
	NumberOfColorPlanes	int	
	VideoArchitecture	int	
	VideoMemoryType	int	
	VideoModeDescription	VarChar(255)	
	VideoProcessor	VarChar(255)	



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### สรุปผลการทดลอง

#### 5.1 สรุปผลการดำเนินการโครงการ

การดำเนินโครงการ ได้ดำเนินการตามขั้นตอนตั้งแต่ศึกษาระบบการบริหารทรัพยากรสารสนเทศ โยธินสารสนเทศ และทำการรวบรวมข้อมูลเกี่ยวกับเทคโนโลยีที่สามารถจะนำมาใช้แก้ปัญหาได้ รวมถึงรวบรวมข้อมูลเกี่ยวกับความต้องการของอาจารย์และผู้ดูแลระบบ มาทำการวิเคราะห์และออกแบบระบบฐานข้อมูล โปรแกรมประยุกต์ ส่วนติดต่อกับผู้ใช้ (User Interface) ของระบบโดยคำนึงถึงความเข้ากันได้ของระบบ (Compatible) ความยืดหยุ่น (flexible) และความปลอดภัย

นอกจากนี้ ยังได้ทำการเขียนส่วนย่อยต่างๆ ที่ใช้ในระบบ เช่น ส่วนติดต่อ Win32 API ส่วนติดต่อ WMI ระบบการเชื่อมต่อ การรับรองสิทธิและส่วนการออกรายงาน

#### 5.2 ปัญหาที่พบในระหว่างการดำเนินโครงการ

1. เพื่อที่จะพัฒนาระบบให้รองรับระบบปฏิบัติการที่มีความหลากหลาย ข้อมูลบางส่วนของบางระบบปฏิบัติการจึงหายไป
2. ทางผู้ผลิตอุปกรณ์ไม่ได้เก็บข้อมูลไว้ในฮาร์ดแวร์ตามมาตรฐาน CIM
3. การใช้งานระบบผ่านเครือข่ายอินเทอร์เน็ตความเร็วต่ำ อาจทำให้การใช้งานระบบมีปัญหาความผิดพลาดเกิดขึ้นได้
4. การพัฒนาระบบใช้เทคโนโลยีใหม่ๆ ทำให้มีแหล่งข้อมูลที่ใช้ค้นคว้าน้อย
5. ปัญหาเรื่องสถานที่การดำเนินโครงการไม่เอื้ออำนวย

#### 5.3 แนวทางการพัฒนาโครงการต่อ

1. เพิ่มความสามารถของระบบในส่วนของความปลอดภัย โดยสามารถรองรับ โปรโตคอลที่เกี่ยวข้องกับการเข้ารหัสของข้อมูลได้มากขึ้น
2. เพิ่มความสามารถในส่วนของแสดงผลข้อมูลและระบบการออกรายงานได้หลากหลายยิ่งขึ้น
3. เพิ่มรายละเอียดในการเก็บข้อมูลของให้มีความละเอียดยิ่งขึ้น
4. เพิ่มความยืดหยุ่นและความคงทนของระบบมากขึ้น รวมถึงความง่ายในการติดตั้งระบบ

#### 5.4 สรุปผล

โครงการนี้ได้ให้ความสำคัญกับการบริหารระบบทรัพยากรสารสนเทศ โยธินสารสนเทศ ซึ่งเน้นความเข้ากันได้ของระบบ ความยืดหยุ่น เพื่อสามารถนำไปใช้ได้กับระบบที่หลากหลาย ซึ่งจะทำให้ระบบการบริหารทรัพยากรสารสนเทศ โยธินสารสนเทศเป็นไปได้อย่างมีประสิทธิภาพมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

Simon Robinson .2004. “Advance .NET Programming” Wrox Press

ดร. สนั่น ศรีสุข. “Internet Programming With Winsock”

<http://www.codeproject.com>

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vbcon/html/vbconintroductionontserviceapplications.asp>

[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnanchor/html/anch\\_wmi.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnanchor/html/anch_wmi.asp)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้