

# การพัฒนาระบบคลังอุปกรณ์คอมพิวเตอร์

## PC Hardware Inventory System



\*H001794\*

โดย

นางสาวนวรรณ์ อักกะวินต

รหัส 42067168

อาจารย์ที่ปรึกษา

ดร.โชติพัชร์ ภรณ์วลัย

วัน เดือน ปี.....	10 ส.ค. 2550
เลขทะเบียน.....	01794
เลขเรียกหนังสือ.....	วท. ๙๖๒๙๓๓ ๒๕๕๔
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."	

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน  
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 1 ปีการศึกษา 2544

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวข้อ	การพัฒนาระบบคลังอุปกรณ์คอมพิวเตอร์
นักศึกษา	นางสาวนวรรตน์ อักกะวินิต
อาจารย์ที่ปรึกษา	อาจารย์โชติพัทธ์ ภรณวลัย
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2544

### บทคัดย่อ

เป็นการเสนอการพัฒนาระบบเพื่อตรวจสอบอุปกรณ์คอมพิวเตอร์ขององค์กร สำหรับตรวจดูว่าในแต่ละเครื่องคอมพิวเตอร์ที่ต่ออยู่ในระบบเครือข่ายติดตั้งอุปกรณ์ใดบ้าง เพื่อเฝ้าต่อการควบคุม และตรวจสอบ มิให้ก่อให้เกิดการทุจริต เช่นการเปลี่ยนแปลงอุปกรณ์บางอย่าง ทำให้การควบคุมอุปกรณ์คอมพิวเตอร์ขององค์กรมีประสิทธิภาพมากขึ้น โดยในระบบประกอบด้วย Software ที่ทำหน้าที่เป็น Manager มีหน้าที่ในการเก็บข้อมูลรวมของอุปกรณ์คอมพิวเตอร์ที่ติดตั้งไว้ที่เครื่องลูกข่าย รวมทั้งที่อยู่ใน Store และ Software ที่ทำหน้าที่เป็น Agent ที่ติดตั้งเป็น Service ฟังไว้ ณ เครื่องลูกข่ายเพื่อคอยตรวจสอบอุปกรณ์คอมพิวเตอร์ของเครื่องตัวเอง และส่งให้ Manager ตรวจสอบ ถ้าพบว่ามีสิ่งผิดปกติ หรือมีการเปลี่ยนแปลงเกิดขึ้น Manager ต้องมีหน้าที่ในการจัดการกับเครื่องลูกข่ายนั้นๆ

ในการติดต่อสื่อสารระหว่าง Manager กับ Agent นั้น ใช้ Protocol TCP/IP โดยติดต่อกันผ่านทาง Socket ใน Port ที่กำหนด

<b>Title</b>	Development of PC Hardware Inventory System
<b>Student</b>	Ms.Nawarat Akkawinet
<b>Advisor</b>	Chotipat Pornawalai
<b>Level of Study</b>	Master of Science in Information Technology
<b>Major</b>	Information Science
<b>Academic Year</b>	2001

### ABSTRACT

Development Of Hardware Control System is the development of software for control hardwares are installed in each computer. So the system can support installed hardware collection in each computers and hardwares in the store to add more efficient in hardware control in organization. The system consist of manager which has main function for controlling the database of installed hardware and hardware in store and compare installed hardware in each computer with the installed hardware list in database as well. Manager also manage the client that has incorrect data of installed hardware by sending warning message or shutting down client computer. The system also consist of the agents that acting as embed application at any client computer. The main function of agent is checking the installed hardware of own and send that data to manager.

Manager and agent establish connection by TCP/IP network so it use socket to communicate.

## กิตติกรรมประกาศ

โครงการนี้ประสบความสำเร็จไปได้ด้วยดีนั้น ผู้พัฒนาระบบควบคุมอุปกรณ์คอมพิวเตอร์ ขอขอบคุณบุคคลต่างๆ ที่มีส่วนในการสนับสนุนให้การจัดทำโครงการสำเร็จตามวัตถุประสงค์ที่ต้องการ ได้แก่

◆ บิดา-มารดา ผู้ซึ่งเล็งเห็นความสำคัญของการศึกษาของลูกๆ ทุกคน  
รวม

◆ ดร.โชติพัชร ภรณ์วลัย ทั้งพี่น้องที่คอยเป็นกำลังใจที่สำคัญตลอดมา  
อาจารย์ที่ปรึกษาโครงการ ผู้ให้คำปรึกษา และแนะนำ  
แนว

ทางการจัดทำโครงการ รวมทั้งให้ความช่วยเหลือ  
ตลอดเวลาในการศึกษา

นางสาวนวรรตน์ อักกะวิเนต  
ผู้จัดทำ

# สารบัญ



หน้า

บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	V
สารบัญภาพ.....	VI
บทที่	
1. บทนำ.....	1
1.1 วัตถุประสงค์ของการพัฒนาโครงการ.....	1
1.2 แนวทางการพัฒนา.....	1
1.3 เครื่องมือที่ใช้ในการพัฒนา.....	2
1.4 เงื่อนไขและขอบเขตในการพัฒนา.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
2. ความรู้พื้นฐานในการพัฒนาระบบ.....	3
2.1 Windows Registry.....	3
2.2 พื้นฐานเกี่ยวกับฟังก์ชันวินโดวส์ API-32 บิต.....	8
2.3 การติดต่อสื่อสารด้วย TCP/IP บน Visual Basic.....	18
3. การออกแบบโครงสร้างระบบ.....	21
3.1 การออกแบบการทำงานของ Manager.....	23
3.2 การออกแบบการทำงานของ Manager.....	26
3.3 การออกแบบการติดต่อสื่อสารระหว่าง Agent และ Manager.....	28
3.4 การออกแบบฐานข้อมูล.....	29
4. การพัฒนาระบบ.....	34
4.1 การพัฒนา Manager.....	34
4.2 การพัฒนา Agent.....	43
5. บทสรุปและข้อเสนอแนะ.....	45
5.1 สรุปผล.....	45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ข้อควรปรับปรุงเพิ่มเติม.....	45
5.3 ปัญหาในการพัฒนาระบบ.....	46



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

หน้า

### ตารางที่

2.1 รายละเอียดไฟล์ .dll.....	9
2.2 ชนิดข้อมูลในภาษา C.....	11
2.3 การประกาศพารามิเตอร์ สำหรับข้อมูลชนิดตัวเลข 8 บิต.....	13
2.4 การประกาศพารามิเตอร์ สำหรับข้อมูลชนิดตัวเลข 16 บิต.....	13
2.5 การประกาศพารามิเตอร์ สำหรับข้อมูลชนิดตัวเลข 32 บิต.....	14
2.6 การประกาศพารามิเตอร์ สำหรับข้อมูลชนิดจำนวนจริง 32 บิต.....	15
2.7 การประกาศพารามิเตอร์ สำหรับข้อมูลชนิดจำนวนจริง 64 บิต.....	15
2.8 การประกาศพารามิเตอร์ สำหรับข้อมูลชนิด handle สำหรับวินโดวส์ 32 บิต.....	16
2.9 การประกาศฟังก์ชันวินโดวส์ API.....	17
3.1 Service Code ที่ใช้ในการร้องขอและให้บริการของ Manager.....	28
3.2 Manage Code ที่ใช้ในการจัดการกับ Agent.....	29
3.3 Table : Workgroup ข้อมูลส่วนงาน.....	31
3.4 Table : Computer ข้อมูลคอมพิวเตอร์.....	31
3.5 Table : InstalledHardware ข้อมูลอุปกรณ์คอมพิวเตอร์ที่ติดตั้งที่คอมพิวเตอร์.....	31
3.6 Table : Hardware ข้อมูลอุปกรณ์คอมพิวเตอร์ใน Store.....	32
3.7 Table : StockIn_Header ข้อมูลการรับของเข้า Store.....	32
3.8 Table : StockIn_Item รายละเอียดอุปกรณ์คอมพิวเตอร์ที่รับเข้ามาใน Store.....	32
3.9 Table : StockOut_Header ข้อมูลการนำของออกไปจาก Store.....	33
3.10 Table : StockOut_Item รายละเอียดอุปกรณ์คอมพิวเตอร์ที่นำออกไปจาก Store.....	33
3.11 Table : Log เก็บประวัติและรายละเอียด error ต่างๆ.....	33

# สารบัญภาพ

หน้า

ภาพที่

2.1 โครงสร้าง Registry.....	4
2.2 การเชื่อมโยงไฟล์ไลบรารีชนิดสแตติก และชนิดไดนามิกส์.....	11
3.1 ภาพรวมของระบบ.....	21
3.2 Context Diagram ของระบบ.....	22
3.3 Data Flow Diagram Level 1 ของระบบ.....	23
3.4 Flow Chart แสดงการทำงานของ Manager.....	25
3.5 Entity Relationship Diagram ของระบบ.....	30
4.1 การทำงานของ Manager.....	34
4.2 การจัดการข้อมูล Workgroup.....	35
4.3 การจัดการข้อมูล Computer.....	36
4.4 การ Query ข้อมูลคอมพิวเตอร์ด้วยหมายเลข IP.....	36
4.5 การจัดการข้อมูลอุปกรณ์คอมพิวเตอร์ที่ติดตั้งในแต่ละเครื่อง.....	37
4.6 การ Query ข้อมูลด้วย Output และเงื่อนไขที่กำหนดได้เอง.....	38
4.7 การ Query ข้อมูลอุปกรณ์ที่ติดตั้งด้วยหมายเลข IP.....	38
4.8 ข้อมูล Log โดยจำแนกตามหมายเลข IP.....	39
4.9 ข้อมูล Log โดยจำแนกตามวันที่.....	39
4.10 การลบข้อมูล Log ตามเงื่อนไขที่กำหนดได้เอง.....	40
4.11 การจัดการ Agent.....	41
4.12 การควบคุม Agent.....	41
4.13 การจัดการข้อมูล Store.....	42
4.14 การจัดการข้อมูล Stock In.....	43
4.15 การจัดการข้อมูล Stock Out.....	43
4.16 การทำงานของ Agent.....	44
4.17 รายการอุปกรณ์คอมพิวเตอร์ใน Agent.....	44

# บทที่ 1

## บทนำ

ในปัจจุบัน บริษัท หรือองค์กรมีจำนวนคอมพิวเตอร์เพิ่มมากขึ้น มักประสบปัญหาของการรวบรวมข้อมูลเกี่ยวกับอุปกรณ์คอมพิวเตอร์ นอกจากนี้ยังประสบปัญหาการขาดการควบคุมอุปกรณ์คอมพิวเตอร์ที่ติดตั้งในแต่ละเครื่องคอมพิวเตอร์ โดยเฉพาะอุปกรณ์บางชนิดที่มีขนาดเล็กแต่มีราคา ซึ่ง User มักจะถอด หรือเปลี่ยนแปลงอุปกรณ์ภายในเครื่อง โดยที่ไม่มีใครรู้ เพื่อจุดประสงค์ใดๆ ก็ตาม ทำให้สร้างความยุ่งยากให้กับผู้ดูแลระบบ เพื่อให้มีการใช้งานที่เหมาะสมและมีประสิทธิภาพสูงสุด ดังนั้น จึงเกิดแนวคิดในการพัฒนาระบบควบคุมอุปกรณ์คอมพิวเตอร์ที่ทำหน้าที่ในทำทะเบียนคอมพิวเตอร์ การตรวจสอบอุปกรณ์ต่างๆ ที่ติดตั้งอยู่ภายในเครื่องคอมพิวเตอร์ที่ต่อกันผ่านเครือข่าย ซึ่งเป็นข้อมูลส่วนใหญ่ของระบบ ยังรวมไปถึงการทำข้อมูลคลังของอุปกรณ์คอมพิวเตอร์ทั้งหมดใน store เพื่อเป็นข้อมูลรวมของอุปกรณ์คอมพิวเตอร์ภายในองค์กร

### 1.1 วัตถุประสงค์ของการพัฒนาโครงการ

1. ศึกษาและวิเคราะห์ปัญหาที่เกี่ยวข้องกับการจัดเก็บข้อมูลอุปกรณ์คอมพิวเตอร์
2. ออกแบบระบบการควบคุมอุปกรณ์คอมพิวเตอร์
3. พัฒนาระบบทะเบียนคอมพิวเตอร์
4. พัฒนาเครื่องมือที่ช่วยในการอ่านค่าและตรวจสอบอุปกรณ์คอมพิวเตอร์ที่ติดตั้งไว้ที่เครื่องคอมพิวเตอร์ รวมทั้งมีวิธีการจัดการกับเครื่องคอมพิวเตอร์ที่พบว่ารายการอุปกรณ์คอมพิวเตอร์ที่ติดตั้งไม่ตรงกับข้อมูลที่มีอยู่
5. พัฒนาระบบคลังของอุปกรณ์คอมพิวเตอร์ที่สำรองไว้ใน store ด้วย

### 1.2 แนวทางการพัฒนา

1. พัฒนา Agent ฟังก์ชัน Client เพื่อทำหน้าที่ในการอ่านค่าอุปกรณ์คอมพิวเตอร์ที่ติดตั้งไว้ที่เครื่อง Client แต่ละเครื่องในระบบเครือข่าย โดยอ่านค่าผ่าน Registry ของระบบ หรือผ่าน Function Window API แล้วรายงานผลมายัง Manger ที่เป็นศูนย์กลาง เพื่อบันทึกข้อมูล นอกจากนี้ยังตรวจสอบค่าที่อ่านได้กับข้อมูลที่มีอยู่ ถ้าไม่ตรงกันจะรายงานไปยัง Manager เช่นกัน
2. พัฒนา Manager เพื่อทำหน้าที่เป็นศูนย์กลางในการทำทะเบียนคอมพิวเตอร์ รวมถึงรวบรวมและบันทึกอุปกรณ์คอมพิวเตอร์ที่ได้ติดตั้งไว้ในแต่ละ Client ซึ่งถูกส่งมาจาก Agent ที่

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประจำอยู่ในแค่ Client และบันทึกข้อมูลของ Client ที่มีรายการอุปกรณ์คอมพิวเตอร์ที่ติดตั้ง ไม่ตรงกับข้อมูลที่มีอยู่ รวมทั้งมีการจัดการกับ Client เหล่านั้นด้วย นอกจากนี้ Manager ยังควบคุมการเข้า-ออกของอุปกรณ์คอมพิวเตอร์ที่สำรองไว้ใน store ด้วย

### 3. Agent และ Manager ติดต่อกันผ่านเครือข่าย TCP/IP

#### 1.3 เครื่องมือที่ใช้ในการพัฒนา

เครื่องมือที่ใช้ในการพัฒนาระบบนี้ ประกอบไปด้วย

1. Microsoft Visual Basic 6.0 ใช้สำหรับพัฒนา Agent ที่ทำงานฝั่งตัวอยู่ในเครื่อง Client และใช้สำหรับพัฒนา Manager ที่เป็นศูนย์กลาง โดยใช้ Microsoft Winsock Control ในการติดต่อสื่อสารระหว่าง Manager และ Client
2. เครื่อง Computer ที่เชื่อมต่อกันเป็นเครือข่าย ที่ติดตั้ง Microsoft Windows 95/98

#### 1.4 เงื่อนไข และขอบเขตในการพัฒนา

เงื่อนไข และขอบเขตในการพัฒนานั้น ได้กำหนดไว้ดังนี้

1. ระบบควบคุมอุปกรณ์คอมพิวเตอร์ มีขอบเขตการใช้งานบนเครื่อง PC Computer ที่เชื่อมต่อกันเป็นเครือข่าย และได้ติดตั้ง Microsoft Windows 95/98 หรือสูงกว่า และใช้ Registry เป็นที่เก็บข้อมูลระบบ
2. การเชื่อมต่อเครือข่ายระหว่าง Agent และ Manager ใช้เครือข่าย TCP/IP เท่านั้น เนื่องจาก การติดต่อระหว่าง Agent และ Manager นั้น ใช้หลักการของ Socket ในการติดต่อสื่อสาร

#### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถเพิ่มประสิทธิภาพในการควบคุมอุปกรณ์คอมพิวเตอร์ที่ติดตั้งในแต่ละเครื่องขององค์กร และง่ายต่อการสืบค้นข้อมูล เพื่อประโยชน์ต่อการ install program หรือการ upgrade เครื่องในแต่ละส่วนงานให้เกิดความเหมาะสม
2. สร้างระบบทะเบียนคอมพิวเตอร์ขององค์กร
3. ควบคุมการสำรองอุปกรณ์คอมพิวเตอร์ในองค์กร

## บทที่ 2

### ความรู้พื้นฐานในการพัฒนาระบบ

เนื่องจากในขั้นตอนของการพัฒนานั้น จำเป็นจะต้องเกี่ยวข้องกับการ Access ข้อมูลใน Windows Registry ผ่านทาง Function Windows API โดยใช้ Microsoft Visual Basic ในการพัฒนา รวมทั้งการติดต่อสื่อสารกันผ่านระบบเครือข่ายด้วย ดังนั้นจึงจำเป็นที่จะต้องศึกษาความรู้พื้นฐานที่เกี่ยวข้องก่อนเพื่อประโยชน์ในการพัฒนาระบบให้มีประสิทธิภาพมากขึ้น

#### 2.1 Windows Registry

Registry คือฐานข้อมูลกลางของระบบปฏิบัติการ Windows ที่ใช้เก็บข้อมูลการกำหนดค่าต่างๆ ทั้งในส่วนของฮาร์ดแวร์ ซอฟต์แวร์ , การปรับหน้าจอ รวมถึง Configuration ทั้งหมด ของ Windows ซึ่งจำเป็นต่อการบริหารการทำงานของคอมพิวเตอร์ทั้งเครื่อง

ก่อนจะมาเป็น Registry อย่างใน Windows95 และเวอร์ชันต่อมา Windows (เวอร์ชัน – 3.1x ลงไป) เก็บข้อมูลเหล่านี้ไว้ในไฟล์ .INI สำหรับแต่ละโปรแกรมแยกกันกระจายไปทั่วฮาร์ดดิสก์ โดยตัว Windows เองเก็บข้อมูลที่ต้องใช้ไว้ในไฟล์ WIN.INI และ SYSTEM.INI ซึ่งการจัดเก็บไฟล์ .INI อย่างนี้ทำให้สะดวกต่อการแก้ไขอยู่บ้าง แต่เมื่อความสามารถของ Windows เพิ่มมากขึ้น รองรับงานที่ซับซ้อนขึ้น ทำให้ต้องปรับเปลี่ยนวิธีการจัดเก็บข้อมูลเสียใหม่ให้เหมาะสมกว่าเดิม ผลลัพธ์จึงได้ออกมาเป็น Registry อย่างนี้เอง

ข้อมูลใน Registry อาจแบ่งได้เป็น 4 ส่วนใหญ่ๆ คือข้อมูลเกี่ยวกับฮาร์ดแวร์ที่มีอยู่ในเครื่องคอมพิวเตอร์ ข้อมูลการทำงานของซอฟต์แวร์แต่ละตัวซึ่งรวมถึงซอฟต์แวร์แบบ 16 บิตด้วย ข้อมูลการกำหนดค่าและการทำงานต่างๆ ของ Windows และสุดท้ายข้อมูลการกำหนดค่าของผู้ใช้แต่ละคน ทั้ง 4 ส่วนนี้ถูกจัดเก็บไว้ในไฟล์ 2 ไฟล์คือ System.dat และ User.Dat ภายใน Directory หลักของ Windows โดยไฟล์ System.Dat ทำหน้าที่เก็บข้อมูลฮาร์ดแวร์,ซอฟต์แวร์ ส่วนไฟล์ User.Dat เก็บข้อมูลการกำหนดค่าสำหรับผู้ใช้แต่ละคน เช่น หน้าตาของ Desktop ,ลักษณะรูป Cursor ที่ใช้ เป็นต้น ไฟล์ .Dat เหล่านี้เก็บข้อมูลผสมกันทั้งที่เป็น Binary และ String ไม่ใช่เท็กซ์ไฟล์เหมือนอย่างไฟล์ .INI อีกต่อไป อีกส่วนที่สำคัญมากและถือเป็นจุดเด่นหนึ่งของ Registry ก็คือข้อมูลบางอย่างใน Registry จะถูกปรับแก้ตลอดเวลาในขณะที่เปิดเครื่องอยู่หรือเปลี่ยนแปลงได้ทุกครั้งที่เปิดเครื่องใหม่ เช่นข้อมูลของการตั้งค่าต่างๆและอุปกรณ์ต่อพ่วง เป็นต้น และเพื่อให้สามารถแก้ไขได้อย่างรวดเร็ว ในระหว่างการทำงานนั้นข้อมูลดังกล่าวจึงถูกโหลดเข้ามาอยู่

ในหน่วยความจำหลัก โดย Windows จะดำเนินการตรวจสอบอุปกรณ์ทุกชิ้นที่มีอยู่ในระบบและไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

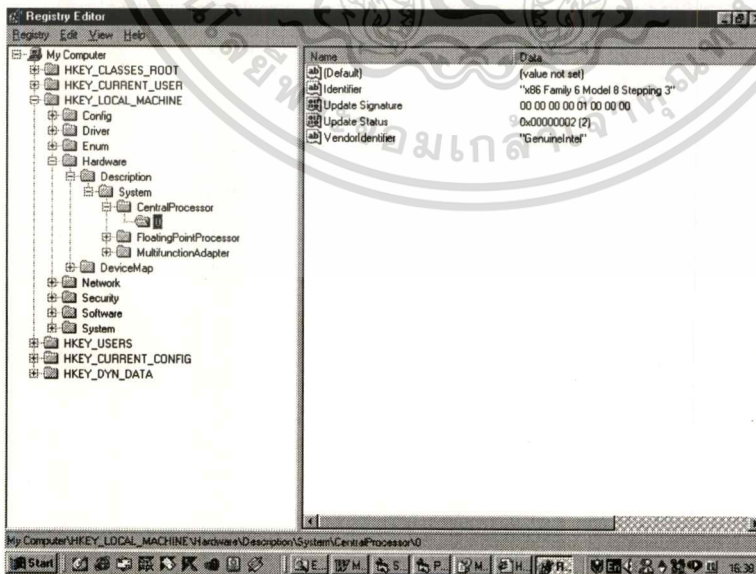
ปรับปรุงค่าต่างๆ ทุกครั้งที่เปิดเครื่องใหม่ เพื่อให้ใช้งานอุปกรณ์เหล่านี้ได้ตามคุณสมบัติของระบบปฏิบัติการแบบ Plug and Play (PnP)

แต่ด้วยความจำเป็น เพื่อให้โปรแกรมเก่าๆยังคงทำงานได้ (Backward Compatibility) ทำให้ยังคงมีไฟล์ WIN.INI และ SYSTEM.INI อยู่เพื่อให้ซอฟต์แวร์รุ่นเก่าสามารถกำหนดค่าบางอย่างให้กับ Windows ได้เช่นเดิม ซึ่งหากเปิดดูก็จะพบว่า มี เนื้อหาน้อยลงมาก รวมทั้งค่าไฟล์ AUTOEXEC.BAT และ CONFIG.SYS ก็ยังมีความจำเป็นต่อการกำหนดสภาพแวดล้อมบางอย่างของ DOS ใน Windows ด้วย

โครงสร้างโดยทั่วไปภายใน Registry ประกอบด้วยส่วนหลักๆที่เรียกว่า คีย์ (key) ซึ่งมีความสัมพันธ์ต่อกันในลักษณะของต้นไม้แบบลำดับชั้น (Hierarchical Tree) เริ่มจากคีย์ที่เป็น Root (root key) แล้วแตกลงไปเป็นคีย์ย่อย (subkey หรือ Hive) ต่างๆ ส่วนที่ใช้เก็บข้อมูลจริงนั้นเป็น Entry ภายในคีย์เรียกว่า Value Entry หรือสั้นๆว่า Entry โดยแต่ละคีย์ใน Registry สามารถจัดเก็บ Entry ไว้ได้ไม่จำกัดจำนวน ไม่ว่าจะ เป็นคีย์ในระดับใดก็ตาม ซึ่งข้อมูลที่เก็บนั้นแบ่งแยกออกได้เป็น 3 ประเภทคือ

1. ข้อมูลแบบ DWORD (ค่าตัวเลขแบบ Double word ขนาด 32 บิต)
2. ข้อมูลแบบ String (ข้อความ)
3. ข้อมูลแบบ Binary

โครงสร้างแบบต่างๆ แสดงได้ดังรูป



รูปที่ 2.1 โครงสร้าง Registry

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การที่แบ่งแยกเป็น System.Dat กับ User.Dat นี้ เพื่อให้เกิดความชัดเจนในการจัดการมากยิ่งขึ้น โดย System.Dat จะเก็บข้อมูลเฉพาะส่วนที่เกี่ยวข้องกับคอมพิวเตอร์เครื่องนั้นๆ ทั้งซอฟต์แวร์และฮาร์ดแวร์ ส่วนไฟล์ User.Dat ก็จะเก็บข้อมูลส่วนที่กำหนดแตกต่างกันสำหรับผู้ใช้แต่ละคน เช่น Icon บน Desktop , การกำหนดรูปภาพพื้นหลังบน Desktop ข้อดีของการเก็บข้อมูลแยกกันสามารถสรุปได้ดังนี้

1. ผู้ใช้แต่ละคนสามารถกำหนดลักษณะหน้าจอของตัวเองบนจอคอมพิวเตอร์เครื่องเดียวกัน เลือก Icon บน Desktop เลือก Wallpaper ตามที่ตัวเองต้องการได้
2. ในระบบเครือข่ายสามารถกำหนดให้ผู้ใช้เห็นหน้าตา Desktop ของตัวเองได้เสมอไม่ว่าจะเข้าสู่ระบบจากคอมพิวเตอร์เครื่องใดก็ตาม โดยกำหนดให้ใช้ไฟล์ User.Dat ที่เครื่อง Server และ การปรับแก้ก็สามารถส่งจากเครื่องที่กำลังใช้อยู่ได้ด้วย ในขณะที่ไฟล์ System.Dat ต้องมันอยู่ในเครื่องคอมพิวเตอร์แต่ละเครื่องเสมอ

## คีย์และค่าใน Registry ที่สำคัญ

### 2.1.1 HKEY\_CURRENT\_USER และ HKEY\_USERS

HKEY\_USER เป็นศูนย์รวมของ Configuration ต่างๆที่เกี่ยวข้องกับผู้ใช้ ในระบบที่มีการกำหนดผู้ใช้ไว้หลายคนก็จะมีคีย์ย่อยของผู้ใช้แต่ละคน ส่วนในคีย์ HKEY\_CURRENT\_USER จะเป็นคอนฟิกูเรชันของผู้ใช้คนปัจจุบัน ซึ่งการให้แก้ไขค่าในคีย์ HKEY\_CURRENT\_USER จะมีผลต่อค่าในคีย์ย่อยของผู้ใช้คนนั้นๆ ภายในคีย์ HKEY\_USER ด้วย เนื่องจาก KEY\_CURRENT\_USER เป็นเพียงจุดที่ Registry ยกขึ้นมาเพื่อความสะดวกในการเข้าถึงค่า Configuration ต่างๆของผู้ใช้คนปัจจุบันเท่านั้น แท้ที่จริงก็ชี้ไปที่ข้อมูลชุดเดียวกันนั่นเอง

คีย์ย่อยที่อยู่ภายใต้คีย์ทั้งสอง มีดังต่อไปนี้

### Control Panel

Desktop หรือ desktop คีย์นี้บรรจุเอ็นทรีที่เกี่ยวข้องกับการกำหนดวอลเปเปอร์และสกรีนเซฟเวอร์ไว้หลายเอ็นทรี รวมทั้งอีก 4 เอ็นทรีที่น่าสนใจ คือ

- CursorBlinkRate กำหนดอัตราการกระพริบของเคอร์เซอร์สำหรับป้อนตัวอักษร
- MenuShowDelay กำหนดเวลาที่ถ่วงไว้ก่อนแสดงเมนูย่อยออกมา
- DoubleClickHeight และ DoubleClickWeidth กำหนดระยะเพื่อการขยับของเมาส์ในระหว่างการดับเบิลคลิก หากการดับเบิลคลิกของคุณยังไม่นิ่งพอ เพื่อช่วยให้ Windows ตัดสินใจได้ว่าเหตุการณ์แบบนี้เป็นการดับเบิลคลิก หรือการลากเมาส์ในระยะทางสั้นๆ

desktop\WindowMetrics คีย์นี้เก็บค่าเกี่ยวกับขนาดของวินโดว, ขนาดตัวอักษร, ระยะห่างระหว่างไอคอน และอีกหลายๆ อย่าง รวมทั้งวิธีแสดงการ Minimize และ Maximize ของวินโดว ให้มีแอนิเมชันหรือไม่

desktop\Keyboard คีย์นี้กำหนดการใช้งานคีย์บอร์ด เช่นช่วงเวลาหลังจากการกดปุ่มบนคีย์บอร์ดค้างไว้ก่อนถือเป็นการพิมพ์ตัวอักษรซ้ำ หรือกำหนดอัตราเร็วในการพิมพ์ซ้ำตัวอักษรเมื่อพิมพ์ค้างไว้ เป็นต้น

desktop\Mouse คีย์นี้กำหนดการใช้งานเมาส์ เช่นกำหนดช่วงเวลาการคลิกเมาส์สองครั้งติดกัน โดยไม่ถือเป็นการดับเบิลคลิก, กำหนดอัตราเร็วเริ่มต้น, กำหนดปุ่มให้เมาส์ เป็นต้น

#### InstallLocationsMRU

คีย์นี้ เป็นการเก็บรายการ (Path) ที่เป็นตำแหน่งค้นหาทางการติดตั้งซอฟต์แวร์ต่างๆ ไว้ โดยมีการบอกลำดับก่อนหลังของพารามิเตอร์

#### keyboard layout\toggle

คีย์นี้เก็บเอ็นทรีที่เกี่ยวข้องกับการสลับภาษาของคีย์บอร์ด

#### RemoteAccess\Profile

คีย์นี้เก็บ Configuration ของการติดต่อทางโมเด็มแต่ละรายการที่ถูกสร้างขึ้นในโพลเดอร์ Dial-Up Networking

#### Software

คีย์นี้ประกอบไปด้วยคีย์ย่อยที่เก็บค่า Configuration ซึ่งซอฟต์แวร์ต่างๆ สร้างขึ้นสำหรับผู้ใช้แต่ละคนในระบบ

Microsoft\Internet เก็บค่า Configuration แต่ละประเภทของ Internet Explorer

Microsoft\Microsoft Office เก็บค่า Configuration สำหรับโปรแกรมในชุด Office

Microsoft\telnet เก็บค่าเซตติ้งของโปรแกรม Telnet

Microsoft\Windows\CurrentVersion เก็บค่า Configuration จากทุกๆ ส่วนของ Windows ซึ่งเป็นค่าของระบบเอง และค่าของโปรแกรมที่ติดตั้งมาพร้อมกับ Windows ด้วย

### 2.1.2. HKEY\_LOCAL\_MACHINE

เป็นคีย์ที่เก็บ Configuration ของเครื่องทั้งในส่วนฮาร์ดแวร์และซอฟต์แวร์ โดยไม่ขึ้นกับผู้ใช้แต่ละคน มีคีย์ย่อยที่น่าสนใจดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Software

คีย์นี้มีโครงสร้างเดียวกันกับคีย์ Software ใน HKEY\_USER ตามที่ได้กล่าวไปแล้ว แต่ซอฟต์แวร์ต่างๆ จะใช้คีย์นี้ในลักษณะของข้อมูลที่เป็นส่วนรวม ไม่ขึ้นกับผู้ใช้คนใดคนหนึ่ง ยกเว้นคีย์ย่อย Classes ที่บรรจุข้อมูลการกำหนดประเภทของ Object ต่างๆภายในเครื่องไว้เช่น ไฟล์, โพลเดอร์, ประเภทของไฟล์ และ Object แบบ OLE เป็นต้น ซึ่งส่วนหนึ่งก็คือ File Associations ที่เราสามารถกำหนดได้จาก Tab file Types ใน Folder Options นั่นเอง และคีย์นี้จะถูกดึงออกมาเป็นคีย์หลักที่ชื่อว่า HKEY\_CLASSES\_ROOT เพื่ออำนวยความสะดวกในการเข้าถึงคีย์ต่างๆที่อยู่ภายใน

คีย์ย่อยที่น่าสนใจภายในคีย์ Software นี้ก็จะเป็นคีย์ของ Windows เป็นหลักคีย์อื่นๆจะขึ้นอยู่กับว่ามีซอฟต์แวร์ใดถูกติดตั้งไว้ในเครื่องนั้นบ้าง

Microsoft\Windows\CurrentVersion\App Paths เป็นตำแหน่งที่ Windows ใช้ในการค้นหาโปรแกรมเมื่อมีการป้อนคำสั่งลงในช่อง Open ของ Dialogue Box Run โดยไม่ได้ระบุ Path (ลักษณะคล้ายๆตัวแปร Path ของ Dos) แต่ละคีย์ย่อยเก็บ Path เต็มของไฟล์โปรแกรมไว้ที่ค่า Default และเก็บ Path ที่เป็น Directory ทำงานของโปรแกรมไว้ที่ Entry Path

Microsoft\Windows\CurrentVersion\controls Folder ภายในเป็นคีย์ย่อยที่เก็บรายละเอียดต่างๆ เกี่ยวกับส่วนเพิ่มเติม Shell (Shell Extension) ของ Applet ใน Control Panel เช่นการเพิ่ม Tab สำหรับกำหนดค่า แต่ละ Applet ใน Control Panel ก็จะต้องมาลงทะเบียนไว้ที่นี้ด้วยเช่นกัน

Microsoft\Windows\CurrentVersion\explorer\xxx\NameSpace xxx ที่เขียนไว้นั้นสื่อถึงพวก Virtual โพลเดอร์ต่างๆ เช่น Desktop , My Computer เป็นต้นภายในคีย์ของ Virtual โพลเดอร์เหล่านี้จะมีคีย์ย่อยชื่อ NameSpace

คีย์ NameSpace คือตำแหน่งที่ จะได้พบกับคีย์ย่อยที่มีชื่อเป็นหมายเลข CLSID (เปิดหัวท้ายด้วยเครื่องหมายปีกกา) ของ Object ใน Virtual โพลเดอร์นั้นอีกทีหนึ่ง (เมื่อพิจารณาจากโครงสร้าง Name Space )จากรูปข้อมูลใน Registry บอกให้เราเห็นว่า โพลเดอร์ Dial-Up Networking เป็น Object ที่อยู่ภายใต้โพลเดอร์ My Computer นั่นเอง

Microsoft\Windows\CurrentVersion\explorer\Shell Folders เหมือนกับคีย์ Shell Folder ใน HKEY\_CURRENT\_USER แต่คีย์นี้ใช้เก็บรายชื่อโพลเดอร์พิเศษที่ใช้ร่วมกันในคอมพิวเตอร์เครื่องนั้นๆ

Microsoft\Windows\CurrentVersion\Run... คีย์ Run ใช้เก็บชื่อ Path เต็มของไฟล์โปรแกรม ซึ่งจะถูกรันทุกครั้งที Log On เข้าสู่ Windows โปรแกรมที่มีพาธเต็มอยู่ในคีย์ Run เหล่านี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะถูกเรียกใช้ในขั้นตอนสุดท้ายๆ ของกระบวนการบู๊ต นอกจากนี้ ยังมีคีย์อื่นๆที่อยู่ในระดับเดียวกัน ได้แก่ RunOnce, Runservices และ RunServicesOnce

คีย์ RunOnce ถูกใช้ในการติดตั้งซอฟต์แวร์ที่ต้องมีการ Restart Windows แล้วจึงติดตั้งต่อจนเสร็จ วิธีการโปรแกรมติดตั้งต่อหลังจากบู๊ตเครื่องใหม่ทำได้โดยบันทึก Path เต็มของโปรแกรมที่จะรันไว้ในคีย์นี้ เมื่อรันเรียบร้อยแล้ว Windows จะจัดการลบเอ็นทรีนั้นออกไปจาก Registry ทันที

Microsoft\Windows\CurrentVersion\Setup เก็บข้อมูลเกี่ยวกับการติดตั้งโปรแกรมในชุดของ Windows ได้แก่ข้อมูลการติดตั้งโปรแกรมย่อย, ข้อมูลการอัปเดตโปรแกรมรุ่นใหม่ๆ ของ Windows และเก็บชื่อ directory หลัก

Microsoft\Windows\CurrentVersion\Uninstall ภายในคีย์นี้ จะพบคีย์ย่อยที่เก็บข้อมูลสำหรับใช้ยกเลิกการติดตั้งซอฟต์แวร์ต่างๆ และจะถูกแสดงเป็นรายการใน Tab Install/Uninstall ของ Applet Add/Remove Programs

### System

เป็นคีย์ที่สำคัญในการบู๊ต ลักษณะเดียวกับไฟล์ AUTOEXEC.BAT และ CONFIG.SYS ของ MS-DOS ข้อมูลของไดรเวอร์ที่ถูกโหลดระหว่างการบู๊ตจะอยู่ในคีย์นี้

### 2.1.3 HKEY\_DYN\_DATA

เป็นคีย์ที่ข้อมูลถูกเก็บไว้ในหน่วยความจำอยู่ตลอดเวลา ข้อมูลส่วนใหญ่จะถูกสร้างขึ้นในขณะที่บู๊ตเครื่องและสามารถถูกแก้ไขได้ตลอดเวลาที่ใช้งานคอมพิวเตอร์ มีคีย์ย่อยที่น่าสนใจดังนี้

#### Config Manager

เป็นข้อมูลเกี่ยวกับคอนฟิกูเรชันของฮาร์ดแวร์ที่ติดตั้งและกำลังใช้งานอยู่ข้อมูลภายในคีย์นี้ถูกสร้างขึ้นเมื่อบู๊ตเครื่องและสามารถเปลี่ยนแปลงแก้ไขได้ตลอดเวลา คีย์นี้เองที่ทำให้สามารถติดตั้งอุปกรณ์ฮาร์ดแวร์ต่างๆ ได้อย่างสะดวกโดยใช้ สมบัติ Plug and Play ของอุปกรณ์นั้น

#### PerfStats

เก็บข้อมูลประสิทธิภาพการทำงานในแต่ละส่วนของคอมพิวเตอร์ ซึ่งจะถูกนำไปแสดงโดยโปรแกรม System Monitor

## 2.2 พื้นฐานเกี่ยวกับฟังก์ชันวินโดวส์ API – 32 บิต

ฟังก์ชันวินโดวส์ API ของวินโดวส์ หมายถึง ฟังก์ชันงานทั้งหมดที่อยู่ในไฟล์ไลบรารี .dll ของวินโดวส์ เช่น User32.dll Gdi32.dll หรือ Kernel32.dll เป็นต้น ซึ่งไฟล์ไลบรารี .dll เหล่านี้จะ

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นแกนหลักของระบบปฏิบัติการวินโดวส์ เพราะการทำงานของวินโดวส์ จะอาศัยความสามารถของฟังก์ชันที่ปรากฏในไฟล์ไลบรารี .dll

### 2.2.1 ไฟล์ไลบรารี .dll ของวินโดวส์ 32 บิต

เป็นไฟล์ที่โดยปกติจะถูกกำหนดให้มีนามสกุล .dll และจะถูกจัดเก็บเอาไว้ใน Directory ย่อย \System ของวินโดวส์ และการทำงานของวินโดวส์ก็ต้องอาศัยฟังก์ชันวินโดวส์ API ที่มีในแต่ละไฟล์ไลบรารี .dll นั่นเอง สำหรับวินโดวส์ 32 บิต (วินโดวส์ 98) สนับสนุนฟังก์ชันวินโดวส์ API มาตรฐานมากกว่า 1,500 ฟังก์ชัน โดยที่จะแยกจัดเก็บไว้ในไฟล์ไลบรารีที่ถือได้ว่าเป็นแกนหลักของวินโดวส์ 32 บิต ดังต่อไปนี้

ตารางที่ 2.1 รายละเอียดไฟล์ .dll

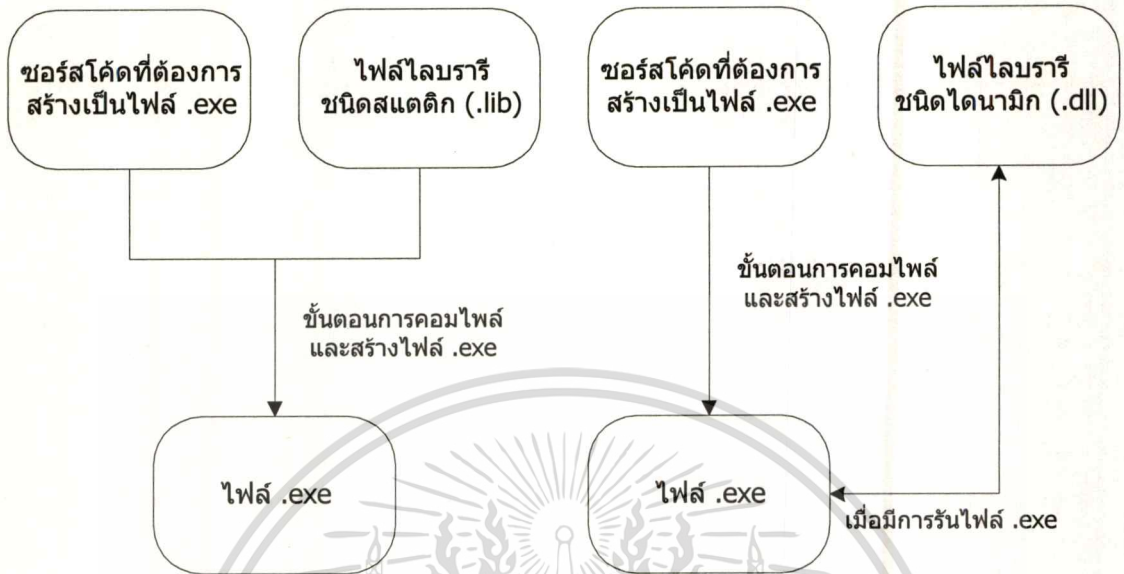
ไฟล์ .dll	รายละเอียด
Gdi32.dll	ไลบรารีสำหรับการติดต่อสื่อสารกับดีไวซ์กราฟิก (Graphic Device Interface) ซึ่งประกอบด้วยฟังก์ชันต่างๆสำหรับงานด้าน อุปกรณ์แสดงผล การวาดกราฟิก ไฟล์เมตต้า (.wmf) ฟอนต์ และการกำหนดตำแหน่งโคออร์ดิเนต
Comdlg32.dll	ไลบรารีสำหรับสนับสนุนการสร้างและแสดงผล Dialog box ซึ่ง Visual Basic 6.0 ก็อาศัยความสามารถจากไลบรารีดังกล่าวนี้ ร่วมกับคอนโทรล Comdlg32.ocx ในการแสดงผล Dialog box มาตรฐาน เช่น Open / Save As หรือ Font เป็นต้น และสำหรับการปรับแต่ง Form สำหรับ Visual Basic 6.0
Version.dll	ไลบรารีสำหรับการสนับสนุนการควบคุม และอ่านหมายเลขเวอร์ชันจากไฟล์ .exe .dll .ocx ฯลฯ ซึ่งจะมีประโยชน์อย่างมากในขณะทำการติดตั้งไฟล์ต่างๆ เนื่องจากวินโดวส์ 32 บิต สนับสนุนการตรวจสอบหมายเลขเวอร์ชันของไฟล์เพื่อป้องกันความไม่เข้ากันของ Application และคอนโทรล ActiveX หรือไฟล์ไลบรารีต่างๆ
Dllz32.dll	ไลบรารีสำหรับการสนับสนุนการขยายขนาดของไฟล์ที่ถูกบีบอัด อย่างเช่น ในการสร้างแผ่น setup นั้น ไฟล์ต่างๆจะถูกบีบอัดด้วยโปรแกรม compress.exe ซึ่งโดยปกติไฟล์ที่ถูกบีบอัดก็จะมีนามสกุลตัวสุดท้ายเป็นเครื่องหมาย underscore ( ) โดยในขณะที่ทำการติดตั้ง ไฟล์เหล่านี้ก็จะถูกขยายขนาดอัตโนมัติโดยวินโดวส์ ซึ่ง

เอกสารนี้เป็นเอกสารที่จะอาศัยฟังก์ชันวินโดวส์ API จากไฟล์ไลบรารี นั่นเอง ดึงนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟล์ .dll	รายละเอียด
Kernel32.dll	ไลบรารีที่ประกอบด้วยฟังก์ชันสำหรับการติดต่อในระดับต่ำ (low-level) กับระบบปฏิบัติการ การจัดการงาน การควบคุมทรัพยากรของระบบ และการจัดการระบบการใช้งานหน่วยความจำ เป็นต้น
User32.dll	ไลบรารีที่ประกอบด้วยฟังก์ชันสำหรับการจัดการกับเมนู Cursor, Caret การจัดการกับเวลาของระบบ การจัดการกับการส่ง message ระหว่าง Application หรือระบบปฏิบัติการ เป็นต้น
Netapi32.dll	ไลบรารีที่ประกอบด้วยฟังก์ชันสำหรับการจัดการด้านระบบเครือข่าย
Odbc32.dll	ไลบรารีที่ประกอบด้วยฟังก์ชันสำหรับการควบคุมการติดต่อสื่อสารระหว่างระบบฐานข้อมูล ที่รูปแบบต่างๆ กัน โดยอาศัยเทคโนโลยีที่เรียกว่า Open Database Connectivity (ODBC) ซึ่งจะช่วยให้การแลกเปลี่ยนข้อมูลระหว่างฐานข้อมูล ที่ถูกจัดการโดยเซิร์ฟเวอร์ต่างๆกัน เป็นไปได้โดยสะดวกและง่ายมากขึ้น
Winmm.dll	ไลบรารีที่ประกอบด้วยฟังก์ชันสำหรับการจัดการด้านมัลติมีเดีย เช่น การจัดการหรือควบคุมการเล่นกับไฟล์ .wav .mid หรือแม้กระทั่งการจัดการกับ CD-Audio เป็นต้น

การที่ไม่โครซอฟท์เรียกไฟล์ไลบรารีประเภทนี้ว่าเป็นชนิดไดนามิกส์ (dynamic) เพราะเงื่อนไขการใช้งานที่เป็นแบบลักษณะไดนามิกนั่นเอง โดยขอให้ทำการเปรียบเทียบกับไฟล์ไลบรารีแบบสแตติก (static) โดยปกติจะมีนามสกุล .lib ในการสร้างไฟล์ .exe หรือที่เรียกว่ากระบวนการคอมไพล์ซอร์สโค้ดนั้น คอมไพเลอร์ต้องทำการลิงก์ฟังก์ชันต่างๆ จากไฟล์ไลบรารีแบบสแตติกในขณะที่ทำการสร้างไฟล์ .exe ดังนั้นฟังก์ชันต่างๆ จากไฟล์ไลบรารีแบบสแตติกจะถูกรวมเอาไว้ในไฟล์ .exe แต่สำหรับการคอมไพล์ซอร์สโค้ด ภายใต้ระบบปฏิบัติการวินโดวส์ ฟังก์ชันต่างๆ จากไฟล์ไลบรารีแบบไดนามิกส์จะไม่ถูกนำมารวมไว้กับไฟล์ .exe ที่สร้างขึ้นมาใหม่ แต่เมื่อมีการรันไฟล์ .exe วินโดวส์จะทำการลิงก์หรือค้นหา และเรียกใช้ฟังก์ชันจากไฟล์ไลบรารีแบบไดนามิกส์ให้เองโดยอัตโนมัติ โดยการอ่านไฟล์ไลบรารีนั้นๆ มาเก็บไว้ในหน่วยความจำ ด้วยเหตุนี้ จึงเรียกไฟล์ไลบรารี .dll ว่า dynamic library link โดยในรูป 2.2 เปรียบเทียบการเชื่อมโยง (link) ไฟล์ไลบรารีชนิดสแตติก และไดนามิกส์



รูปที่ 2.2 การเชื่อมโยงไฟล์ไลบรารีชนิดสแตติก และชนิดไดนามิกส์

### 2.2.2 การเขียนฟังก์ชันวินโดวส์ API ด้วย Visual Basic

เนื่องจากฟังก์ชันวินโดวส์ API ส่วนใหญ่จะถูกเขียนขึ้นมาเพื่อสนับสนุนภาษา C หรือ Visual C++ ดังนั้นชนิดของข้อมูลบางส่วนจึงไม่ Compatible กับชนิดข้อมูลของ Visual Basic 6.0 ซึ่งในการส่งผ่านตัวแปรให้กับแต่ละพารามิเตอร์ของฟังก์ชันวินโดวส์ API จึงต้องมีการตรวจสอบความเข้ากันได้ของชนิดข้อมูลเสียก่อนทุกครั้ง และความหมายของชนิดข้อมูลสำหรับภาษา C หรือ Visual C++ มีดังต่อไปนี้

ตารางที่ 2.2 ชนิดของข้อมูลในภาษา C

ชนิดข้อมูล	รายละเอียด
BOOL	ข้อมูลชนิด Boolean ขนาด 32 บิต โดยที่ 0 หมายถึง False และค่าไม่เท่ากับ 0 หมายถึง True
BYTE	ข้อมูลชนิดไบต์ ขนาด 8 บิต
Char	ข้อมูลชนิดตัวเลขเต็ม unsigned ขนาด 8 บิต
FARPROC	ข้อมูลชนิด Pointer ไปยัง Procedure หรือฟังก์ชัน ขนาด 32 บิต
HANDLE	ข้อมูลชนิดเลขจำนวนเต็ม unsigned ขนาด 32 บิต เพื่อเป็น handle ไปยัง Object

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชนิดข้อมูล	รายละเอียด
Int	ข้อมูลชนิดเลขจำนวนเต็ม signed ขนาด 32 บิต
Short	ข้อมูลชนิดเลขจำนวนเต็ม signed ขนาด 16 บิต
Long	ข้อมูลชนิดเลขจำนวนเต็ม signed ขนาด 32 บิต
LPINT	ข้อมูลชนิด Pointer ขนาด 32 บิต ซึ่งไปยังข้อมูลชนิดเลขจำนวนเต็ม signed ขนาด 32 บิต
LPSTR	ข้อมูลชนิด Pointer ขนาด 32 บิต ซึ่งไปยังข้อมูลชนิดสตริงชนิด ASCII
UINT	ข้อมูลชนิดเลขจำนวนเต็ม unsigned ขนาด 32 บิต
WORD	ข้อมูลชนิดเลขจำนวนเต็ม unsigned ขนาด 16 บิต
DWORD	ข้อมูลชนิดเลขจำนวนเต็ม unsigned ขนาด 32 บิต
Flags	ข้อมูลชนิด Flag ขนาด 32 บิต
Float	ข้อมูลชนิดเลขจำนวนจริง ขนาด 32 บิต
Double	ข้อมูลชนิดเลขจำนวนจริง ขนาด 32 บิต
LPUNKNOWN	Pointer ขนาด 32 บิต ไปยัง Interface lunknown OLE 2.0

จากข้างต้นจะเห็นว่าชนิดข้อมูลของภาษา C หรือ Visual C++ นั้นมีความหลากหลายมาก โดยเฉพาะชนิดชนิดตัวเลขเพราะภาษา C สนับสนุนทั้งข้อมูลตัวเลขที่มีเครื่องหมายและไม่มีเครื่องหมาย แต่สำหรับ Visual Basic 6.0 จะสนับสนุนเฉพาะข้อมูลตัวเลขที่มีเครื่องหมายเท่านั้น ยกเว้นข้อมูลชนิด byte ของ Visual Basic 6.0 เท่านั้น ที่สนับสนุนข้อมูลชนิดไม่มีเครื่องหมาย ดังนั้นเพื่อให้สามารถเข้าใจถึงการเปรียบเทียบชนิดข้อมูลระหว่างภาษาซี และ Visual Basic 6.0 ได้ง่ายขึ้น ก็สามารถพิจารณาได้ดังนี้

### 1. ข้อมูลชนิดตัวเลขขนาด 8 บิต

ข้อมูลชนิดตัวเลขขนาด 8 บิต สำหรับภาษาซีนั้นมีอยู่ด้วยกัน 2 ประเภท ดังนี้ char และ BYTE ซึ่งจะ Compatible กับข้อมูลชนิด byte ของ Visual Basic 6.0 โดยจะต้องมีการประกาศใช้กับ พารามิเตอร์ ดังนี้

### ตารางที่ 2.3 การประกาศพารามิเตอร์ สำหรับข้อมูลชนิดตัวเลข 8 บิต

C, C++	Visual Basic 6.0
Char posval	ByVal posval As Byte
BYTE posval	ByVal posval As Byte

สำหรับข้อมูลชนิด char นั้นสามารถเก็บค่าได้ตั้งแต่ -128 ถึง 127 แต่ข้อมูลชนิด byte ของ Visual Basic 6.0 เป็นข้อมูลชนิดเลขจำนวนเต็ม unsigned ดังนั้นถ้าหากค่าของข้อมูลชนิด char มีค่าเป็นลบ ต้องทำการเขียน Code เพื่อแปลงค่าของข้อมูลเอง ส่วนข้อมูลชนิด BYTE นั้น Compatible กับข้อมูลชนิด byte ของ Visual Basic 6.0 แต่เนื่องจากวินโดวส์ 32 บิต เน้นหนักด้านการระบบปฏิบัติการระบบ 32 บิต ดังนั้นข้อมูลชนิด char และ BYTE มักจะไม่มีการใช้งานในฟังก์ชันวินโดวส์ API

### 2. ข้อมูลชนิดตัวเลขขนาด 16 บิต

สำหรับภาษาซีนั้นมีอยู่ด้วยกัน 2 ประเภท ดังนี้ short และ WORD ซึ่งจะ Compatible กับข้อมูลชนิด integer ของ Visual Basic 6.0 โดยจะต้องมีการประกาศให้กับพารามิเตอร์ ดังนี้

### ตารางที่ 2.4 การประกาศพารามิเตอร์ สำหรับข้อมูลชนิดตัวเลข 16 บิต

C, C++	Visual Basic 6.0
Short posval	ByVal posval As Integer
WORD posval	ByVal posval As Integer

สำหรับข้อมูลชนิด short นั้นเป็นข้อมูลชนิดตัวเลข signed ขนาด 16 บิต ซึ่งจะเหมือนกับข้อมูลชนิด integer ของ Visual Basic 6.0 ทุกประการ แต่สำหรับข้อมูลชนิด WORD นั้นเป็นข้อมูลชนิดตัวเลข unsigned ขนาด 16 บิต ซึ่งสามารถจัดเก็บค่าตัวเลขได้ตั้งแต่ 0 ถึง 65,535 ดังนั้นจึงไม่เหมือนกับข้อมูลชนิด Integer ของ Visual Basic 6.0 แต่เนื่องจากขนาดของข้อมูลเท่ากัน จึงมีความ Compatible กัน และในการส่งค่าตัวเลขหรืออ่านค่ากลับจากฟังก์ชันวินโดวส์ API สำหรับข้อมูลชนิดตัวเลข unsigned ขนาด 16 บิต ที่มีค่าตั้งแต่ 32,768 ถึง 65,535 จึงต้องอาศัยเทคนิคในการแปลงข้อมูลตัวเลขที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3. ข้อมูลชนิดตัวเลขขนาด 32 บิต

สำหรับภาษาซีนั้นมีอยู่ด้วยกัน 5 ประเภทดังนี้ BOOL, DWORD, int, UINT และ LONG ซึ่งจะ Compatible กับข้อมูลชนิด long ของ Visual Basic 6.0 โดยจะต้องมีการประกาศให้กับพารามิเตอร์ ดังนี้

ตารางที่ 2.5 การประกาศพารามิเตอร์ สำหรับข้อมูลชนิดตัวเลข 32 บิต

C, C++	Visual Basic 6.0
BOOL posval	ByVal posval As Long
DWORD posval	ByVal posval As Long
Int posval	ByVal posval As Long
LONG posval	ByVal posval As Long
UINT posval	ByVal posval As Long

ข้อมูลชนิด int และ LONG นั้นเป็นข้อมูลชนิดตัวเลข Signed ขนาด 32 บิต ซึ่งจะเหมือนกับข้อมูลชนิด long ของ Visual Basic 6.0 ทุกประการ แต่สำหรับข้อมูลชนิด DWORD และ UINT นั้นเป็นข้อมูลชนิดตัวเลข unsigned ขนาด 32 บิต และสามารถจัดเก็บค่าตัวเลขได้ตั้งแต่ 0 ถึง 4,294,967,295 ดังนั้นจึงไม่เหมือนกับข้อมูลชนิด long ของ Visual Basic 6.0 แต่เนื่องจากขนาดของข้อมูลเท่ากัน จึงมีความ Compatible กัน และในการส่งค่าตัวเลขหรืออ่านค่ากลับจากฟังก์ชันวินโดวส์ API สำหรับข้อมูลชนิดตัวเลข unsigned ขนาด 32 บิต ที่มีค่าตั้งแต่ 2,147,483,647 ถึง 4,294,967,295 นั้นจึงต้องมีเทคนิคในการแปลงข้อมูล

### 4. ข้อมูลชนิดบูลีน (BOOL) นั้น

สำหรับ Visual Basic 6.0 จะมีอยู่เพียง 2 ความหมายดังนี้ 0 หมายถึง False ส่วนค่าอื่นๆ ที่ไม่เท่ากับ 0 หรือโดยปกติค่าของ -1 จะหมายถึง True แต่สำหรับภาษาซีมีข้อแตกต่างกับ Visual Basic 6.0 ตรงที่ค่าของ True นั้นจะเท่ากับ 1 ส่วนค่าของ False นั้นจะเท่ากับ 0 เช่นกัน ดังนั้นในการแปลงค่าจาก True มาเป็น False สำหรับค่าตัวเลขที่ได้จากภาษาซีหรือฟังก์ชันวินโดวส์ API นั้น จึงไม่สามารถกระทำได้โดยใช้ตัวปฏิบัติการบูลีน Not (สำหรับ Visual Basic 6.0 นั้น True = Not False หรือ -1 = Not 0 ซึ่งแตกต่างจากภาษาซีเพราะ True มีค่าเท่ากับ 1) แต่ในการตรวจสอบ

ค่าตัวเลขที่ได้จากภาษาซีหรือฟังก์ชันวินโดวส์ API แนะนำให้ใช้เทคนิคดังนี้ไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตรวจสอบค่าที่เป็นจริง (True)

If IsWindowVisible(hWnd%) Then

...

End If

ตรวจสอบค่าที่เป็นเท็จ (False)

If IsWindowVisible(hWnd%) = False Then

...

End If

### 5. ข้อมูลชนิดตัวเลขจำนวนจริงขนาด 32 บิต

สำหรับภาษาซีนั้นมีอยู่ด้วยกันเพียงประการเดียว คือ float ซึ่งจะเหมือนกับข้อมูลชนิด single ของ Visual Basic 6.0 ทุกประการ ดังนั้นจึงไม่จำเป็นต้องมีการแปลงข้อมูลที่ได้

ตารางที่ 2.6 การประกาศพารามิเตอร์ สำหรับข้อมูลชนิดจำนวนจริง 32 บิต

C, C++	Visual Basic 6.0
float posval	ByVal posval As Single

### 6. ข้อมูลชนิดตัวเลขจำนวนจริงขนาด 64 บิต

สำหรับภาษาซีนั้นมีอยู่ด้วยกันเพียงประการเดียว คือ double ซึ่งจะเหมือนกับข้อมูลชนิด double ของ Visual Basic 6.0 ทุกประการ ดังนั้นจึงไม่จำเป็นต้องมีการแปลงข้อมูลที่ได้ ซึ่งสามารถประกาศให้กับพารามิเตอร์

ตารางที่ 2.7 การประกาศพารามิเตอร์ สำหรับข้อมูลชนิดจำนวนจริง 64 บิต

C, C++	Visual Basic 6.0
Double posval	ByVal posval As Double

## 7. ข้อมูลชนิด handle สำหรับวินโดวส์ 32 บิต

จะมีขนาดเท่ากับ 32 บิต สำหรับใช้กำหนดหรือ อ้างอิง (reference) ชนิดของ Object ของ วินโดวส์ ดังนั้นถ้าหากมีการกำหนดตัวแปร สำหรับ รับหรือส่งข้อมูลชนิด handle นี้ ก็ต้องใช้ข้อมูลชนิด long ของ ของ Visual Basic

ตารางที่ 2.8 การประกาศพารามิเตอร์ สำหรับข้อมูลชนิด handle สำหรับวินโดวส์ 32 บิต

C, C++	Visual Basic 6.0
HWND posval	ByVal posval As Long

## 8. ข้อมูลชนิดสตริง (string)

ที่สนับสนุนโดย C หรือ C++ นั้นเป็น null-terminated C (ASCIIZ) ซึ่งแตกต่างจากสตริงของ Visual Basic 6.0 โดยสิ้นเชิง ดังนั้น จึงไม่สามารถส่งผ่านสตริงของ Visual Basic 6.0 ให้กับ ฟังก์ชันวินโดวส์ API โดยตรง แต่ต้องทำการแปลงสตริง ของ Visual Basic 6.0 ให้อยู่ในรูปแบบของ ASCIIZ เสียก่อน โดยการนำหน้าพารามิเตอร์ชนิดสตริงด้วย ByVal จะทำให้ Visual Basic 6.0 แปลงสตริงให้อยู่ในรูปแบบ ASCIIZ ก่อนที่จะส่งให้กับฟังก์ชัน และสตริงดังกล่าวก็สามารถที่จะถูกแก้ไขได้โดยฟังก์ชันวินโดวส์ API ดังนั้น สตริงที่ส่งให้กับฟังก์ชันวินโดวส์ API จะต้องถูกกำหนดขนาดให้เพียงพอกับจำนวนตัวอักษรที่ผู้อ่านต้องการรับจากฟังก์ชันวินโดวส์ API บวกกับ 1 สำหรับ ASCIIZ

### 2.2.3 การประกาศฟังก์ชันวินโดวส์ API

ก่อนที่จะสามารถใช้งานฟังก์ชันวินโดวส์ API ในการเขียนโปรแกรม จะต้องทำการประกาศฟังก์ชันนั้นๆ เสียก่อน เพื่อเป็นการกำหนด Prototype ในการติดต่อระหว่างฟังก์ชันวินโดวส์ API กับ Visual Basic 6.0 สำหรับการประกาศฟังก์ชันวินโดวส์ API ซึ่งมีอยู่ด้วยกัน 2 รูปแบบ คือการประกาศ Procedure ประเภท Sub Routine และ การประกาศ Procedure ประเภท ฟังก์ชัน

#### รูปแบบที่ 1

**[Public | Private] Declare Sub name Lib "libname" [ Alias "aliasname"][(arglist)]**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบที่ 2

[Public | Private] Declare Function name Lib "libname" [ Alias "aliasname"][(arglist)][As type]

ตารางที่ 2.9 การประกาศฟังก์ชันวินโดวส์ API

รูปแบบ	คำอธิบาย
Public	กำหนดให้ Procedure ที่ถูกประกาศสามารถถูกเรียกใช้ได้จาก Module หรือ Form อื่นๆ ใน Project ปัจจุบัน แต่การประกาศ Procedure ด้วย Public จะต้องการกระทำใน Module มาตรฐาน (.bas) เท่านั้น
Private	กำหนดให้ Procedure ที่ถูกประกาศสามารถถูกเรียกใช้ได้ เฉพาะจาก Module หรือ Form ที่ประกาศ Procedure นี้เท่านั้น
Sub	กำหนดให้ Procedure เป็นชนิด Routine ซึ่งจะไม่มีคำสั่งกลับ
Function	กำหนดให้ Procedure เป็นชนิดฟังก์ชัน ซึ่งจะมีคำสั่งกลับ ดังนั้น จึงสามารถนำค่าที่ได้จากฟังก์ชัน ไปใช้ในลักษณะของตัวแปรหนึ่งในนิพจน์ (expression) เช่น นิพจน์การคำนวณ
Name	กำหนดชื่อของ Procedure สำหรับ Visual Basic 6.0 ยอมให้ชื่อของ Procedure ความยาวทั้งหมดเพียง 40 ตัวอักษรเท่านั้น
Lib	กำหนดให้สตริงที่ตาม ("libname") จะบ่งบอกถึงชื่อของไฟล์ไลบรารี .dll ซึ่งในการประกาศฟังก์ชันวินโดวส์ API จะต้องมีการกำหนดชื่อของไฟล์ไลบรารี .dll ทุกครั้ง
Alias	กำหนดให้ Procedure ชื่อ "aliasname" สามารถถูกเรียกใช้ได้โดยใช้ชื่ออื่นๆ ตามที่กำหนดโดย name ทั้งนี้เนื่องจากในบางกรณีชื่อของฟังก์ชันวินโดวส์ API นั้น อาจจะซ้ำกับประโยคหรือคำสั่งของ Visual Basic 6.0 หรืออาจจะประกอบด้วยตัวอักษรที่ Visual Basic 6.0 ไม่ยอมรับในการกำหนดเป็นชื่อของ Procedure
Arglist	รายการของพารามิเตอร์ที่ส่งให้กับ Procedure
ByVal	กำหนดให้พารามิเตอร์มีการส่งค่าแบบ passed by value แต่ถ้าหากนำไปใช้ประกาศฟังก์ชันวินโดวส์ API ก็จะมี ความหมายตามแต่ชนิดข้อมูลของพารามิเตอร์ตามที่เคยกล่าวรายละเอียดมาแล้วข้างต้น เช่น Byte StrVal\$ หมายถึง ASCIIZ
ByRef	กำหนดให้พารามิเตอร์มีการส่งค่าแบบ passed by reference

รูปแบบ	คำอธิบาย
ParamArray	ใช้กำหนดให้พารามิเตอร์ดังกล่าวเป็น array ของข้อมูลชนิด variant ซึ่งจะต้องเป็นพารามิเตอร์ตัวสุดท้ายของ Procedure เท่านั้น โดยทั่วไปส่วนนี้มักจะไม่สามารถใช้ได้กับฟังก์ชันวินโดวส์ API มาตรฐาน ยกเว้นฟังก์ชันถูกออกแบบเฉพาะกับ Visual Basic 6.0 เท่านั้น

### ตัวอย่างการประกาศ Function Windows API

```
Public Declare Function SendMessage Lib "User32.dll" Alias_
    "SendMessageA" (ByVal hwnd As Long, ByVal wParam As Long,
    ByVal lParam As Long, ByVal wMsg As Long) As Long
```

การประกาศฟังก์ชันวินโดวส์ SendMessage ตามตัวอย่างข้างบน กำหนดให้เป็นฟังก์ชันที่มีการส่งค่ากลับเป็นข้อมูลชนิด long integer (4 ไบต์) โดยที่เป็นฟังก์ชัน ชื่อ SendMessage ซึ่งชื่อฟังก์ชันเดิมที่ถูกจัดเก็บในไฟล์ไลบรารีของวินโดวส์ได้แก่ SendMessageA และฟังก์ชันของไฟล์ไลบรารี User32.dll โดยที่ฟังก์ชันนี้จะประกอบด้วยพารามิเตอร์ทั้งหมด 4 ตัว ดังมีรายละเอียดดังต่อไปนี้

- hwnd หมายเลข handle ของออบเจกต์ ซึ่งเป็นข้อมูลชนิด long
- wParam หมายเลข message ของวินโดวส์ ซึ่งเป็นข้อมูลชนิด long
- lParam ข้อมูลสำหรับการทำงานของฟังก์ชัน ซึ่งเป็นข้อมูลชนิด Long
- lParam ข้อมูลสำหรับการทำงานของฟังก์ชัน ซึ่งเป็นข้อมูลชนิด Long

### 2.3 การติดต่อสื่อสารด้วย TCP/IP บน Visual Basic

การทำงานเกี่ยวกับเครือข่ายคอมพิวเตอร์นั้นเป็นเรื่องค่อนข้างซับซ้อนจนทำให้มีการสร้างข้อกำหนดต่างๆ ขึ้นมาเพื่อให้วิธีการสื่อสารระหว่างกันเป็นไปได้โดยราบรื่นแล้วก็ทำให้เกิดเป็นโพรโตคอล(Protocol) มากมายหลายชนิดมาใช้งานร่วมกับเครือข่าย ซึ่งมีการแบ่ง Protocol ต่างๆ ออกเป็นระดับชั้น เพื่อให้ง่ายต่อการใช้งาน และง่ายต่อการทำความเข้าใจ

นานวันเข้าโพรโตคอลก็มีเพิ่มมากขึ้นจนกลายเป็นความยุ่งยาก ต่างบริษัทก็ต่างมีผลิตภัณฑ์ที่สนับสนุนโพรโตคอลต่างๆ ไม่เหมือนกัน ทำให้การใช้งานร่วมกันทำได้ยากขึ้น ทำให้ต้องหาวิธีใช้งานโพรโตคอลต่างๆ ร่วมกันโดยใช้มาตรฐานอันเดียวที่ทุกคนยอมรับ ซึ่ง Winsock ก็เป็นหนึ่งในมาตรฐานเหล่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบปฏิบัติการยูนิกซ์จะมีความสามารถอย่างหนึ่งที่เรียกว่า Socket ซึ่งก็คือ การที่โปรแกรมต่างๆ สามารถสื่อสารข้อมูลระหว่างกันได้ โดยไม่จำเป็นต้องรันอยู่บนคอมพิวเตอร์เครื่องเดียวกัน

Socket เองก็มาจากแนวความคิด Client/Server โดยส่วนที่ทำหน้าที่เป็นเซิร์ฟเวอร์ จะสร้าง Socket (ช่องทางการติดต่อ) จะมีจำนวนเท่าใดก็ได้แล้วแต่ จากนั้นเมื่อมี Client ที่ต้องการสื่อสารด้วยก็จะติดต่อมาที่ Socket ที่ Server ได้เตรียมไว้ให้ เมื่อเชื่อมต่อกันเสร็จก็พร้อมจะแลกเปลี่ยนข้อมูลระหว่างกัน โดยเป็นการสื่อสารแบบ 2 ทางชนิดเต็มรูปแบบ (2 Way Full-Duplex)

### 2.3.1 การใช้งาน Winsock กับระบบปฏิบัติการตระกูล Windows

ระบบปฏิบัติการตระกูล Windows อันได้แก่ Windows 95/98 และ windows NT ก็ได้เตรียมการทำงานในลักษณะ Socket โดยตั้งชื่อว่า Winsock โดยมีหลักการคล้ายคลึงกับ Socket ในยูนิกซ์ แม้จะไม่ Compatible กัน 100 % ก็ตาม

Winsock เป็น API (Application Programming Interface) หรือชุดฟังก์ชันมาตรฐาน ซึ่งออกแบบมาเพื่อทำให้การเขียนโปรแกรมกับอินเตอร์เน็ตเป็นเรื่องง่าย อีกทั้ง Winsock ยังถูกกำหนดให้เป็น API ที่มีลักษณะเปิดกว้าง และสามารถ port ไปยัง Socket ของยูนิกซ์ได้อย่างไม่ยากนัก

### 2.3.2 โหมดการทำงานของ Winsock Control

การทำงานของ Winsock Control กับโปรโตคอล TCP/IP นี้แบ่งการทำงานออกเป็น 2 โหมดคือ TCP กับ UDP

TCP (Transmission Control Protocol) TCP เป็นการทำงานในลักษณะ Connection-Based เปรียบเหมือนกับการทำงานของโทรศัพท์ที่จะมีการตรวจสอบการทำงานจากทั้งสองฝั่งมีการโต้ตอบระหว่างกัน ทำให้ได้การสื่อสารที่มีประสิทธิภาพ TCP จะเหมาะกับการส่งข้อมูลที่ต้องการการทำงานที่ถูกต้องแม่นยำของข้อมูล เช่นการส่งภาพหรือเสียงผ่านเครือข่าย ซึ่งจะมีการตรวจสอบความถูกต้องของข้อมูลที่มีการส่งถึงกัน ทำให้มีการใช้ทรัพยากรของระบบค่อนข้างสูง

UDP (User Datagram Protocol) UDP เป็นการทำงานในลักษณะตรงกันข้ามกับ TCP นั่นคือมีการทำงานในลักษณะ Connectionless เปรียบเหมือนการกระจายเสียงวิทยุ UDP จะเหมาะกับการส่งข้อมูลที่ไม่สำคัญนัก เช่นส่งข้อมูลแสดงสถานะการทำงานซึ่งก็มักจะเป็นข้อมูลขนาดเล็กซึ่งทำให้ UDP กินทรัพยากรของระบบน้อยกว่า TCP

### Property ที่สำคัญของ Winsock Control

เอกสารนี้เป็น Protocol ที่สงวนไว้เป็นการเลือกโปรโตคอลสำหรับการทำงาน อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LocalPort	เป็นการกำหนดหมายเลขพอร์ตของคอมพิวเตอร์ที่จะใช้งานกับ Winsock
RemoteHost	เป็นการกำหนดชื่อของคอมพิวเตอร์ที่เราจะติดต่อด้วย อาจจะเป็น
IP Address	หรือเป็นชื่อคอมพิวเตอร์ที่เป็นชื่อที่ง่ายต่อการจดจำ
RemotePort	เป็นการกำหนดหมายเลขพอร์ตของคอมพิวเตอร์ที่เราจะติดต่อด้วย
ByteReceive	เป็นจำนวนข้อมูลที่ได้รับเข้ามาเก็บใน Buffer จากเมธอด GetData

### Method ที่สำคัญของ Winsock Control

Listen เป็นเมธอดที่ใช้สร้าง Socket แล้ว รอการ Connect ทำให้คอมพิวเตอร์เครื่องอื่นๆ สามารถติดต่อเข้ามาได้

Connect เป็นเมธอดที่ใช้สร้างการติดต่อแบบ Socket ไปยังคอมพิวเตอร์เครื่องอื่น โดยจะต้องระบุ Socket Address (IP Address กับหมายเลขพอร์ตที่กำหนดให้ใช้กับ Socket)

Accept Request เป็นเมธอดที่ใช้รับ Request จากคอมพิวเตอร์ที่ติดต่อเข้ามา

SendData เป็นเมธอดที่ใช้ส่งข้อมูลไปยังคอมพิวเตอร์เครื่องอื่นที่เราติดต่อแบบ Socket

GetData เป็นเมธอดที่ใช้รับข้อมูลจากบัฟเฟอร์เข้ามาเก็บในตัวแปรที่เรากำหนดให้ โดยสามารถกำหนดชนิดตัวแปร และความยาวของข้อมูลที่จะนำมาเก็บได้

Close เป็นเมธอดที่ใช้ยกเลิกการติดต่อ

### Event ที่สำคัญของ Winsock Control

ConnectionRequest เป็นเหตุการณ์ที่เกิดเมื่อ คอมพิวเตอร์เครื่องอื่น มีการ Request เข้ามา ซึ่งจะมีการกำหนด ID ให้กับแต่ละค่าของ Request ที่เข้ามา

Data Arrival เป็นเหตุการณ์ที่เกิดเมื่อมีข้อมูลชุดใหม่เข้ามาเก็บใน Buffer ซึ่งเราสามารถตรวจสอบขนาดข้อมูลได้จาก Property ByteReceive

SendProgress เป็นเหตุการณ์ที่เกิดขณะที่มีการส่งข้อมูลระหว่างกัน ซึ่งจะมีพารามิเตอร์แสดงจำนวนข้อมูลที่ส่งมาแล้ว และข้อมูลที่ยังคงเหลืออยู่

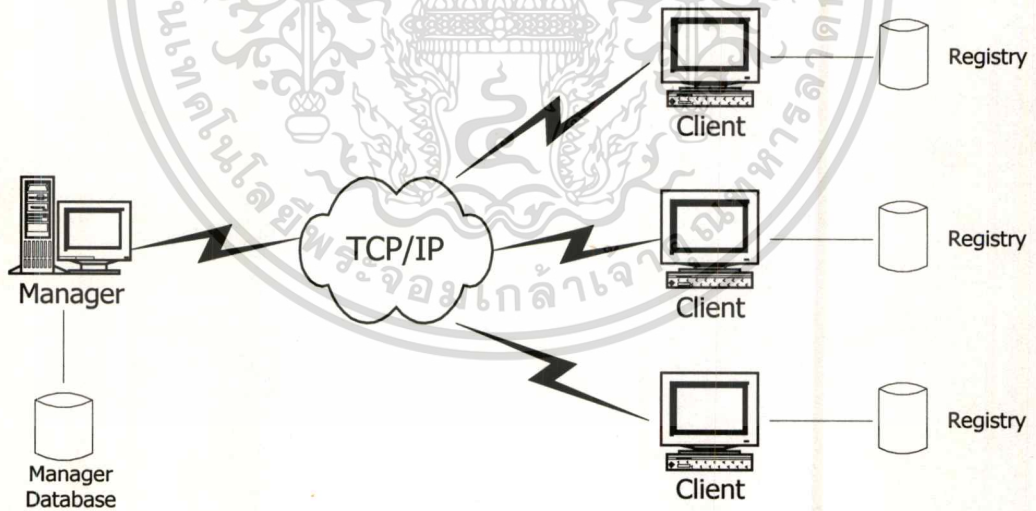
SendComplete เป็นเหตุการณ์ที่เกิดเมื่อการส่งข้อมูลเสร็จสิ้นสมบูรณ์

Error เป็นเหตุการณ์ที่เกิดเมื่อมีความผิดพลาดเกิดขึ้น ซึ่งจะแสดงหมายเลขของความผิดพลาด, คำอธิบาย และรายละเอียดอื่นๆ สำหรับการจัดการกับข้อผิดพลาดที่เกิดขึ้น

### บทที่ 3

#### การออกแบบโครงสร้างระบบ

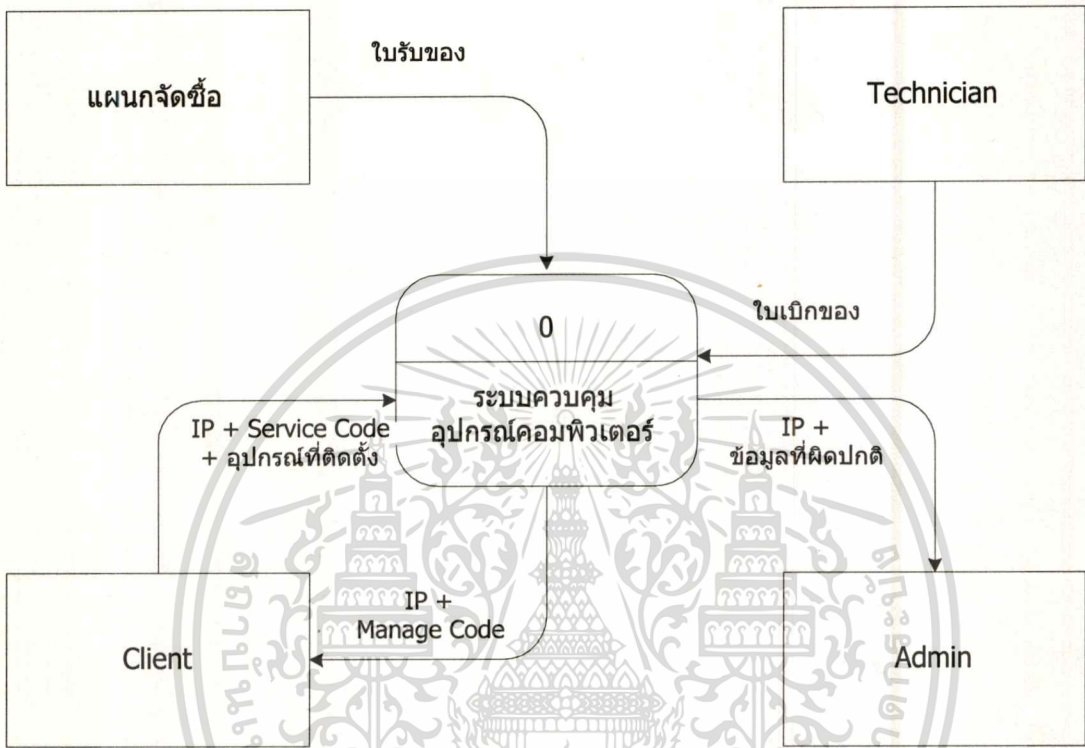
ในขั้นการตอนการออกแบบได้ออกแบบโดยแบ่งการทำงานออกเป็น 2 ส่วนตามลักษณะหน้าที่การทำงาน โดยในระบบประกอบไปด้วย Manager ที่ทำหน้าที่เป็นศูนย์กลางในการเก็บรวบรวมข้อมูลคอมพิวเตอร์ ข้อมูลการติดตั้งอุปกรณ์คอมพิวเตอร์ ที่ได้ทำการติดตั้งไว้ตามเครื่อง Client พร้อมทั้งจัดการกับเครื่อง Computer ที่พบว่าได้ติดตั้งอุปกรณ์คอมพิวเตอร์ที่แตกต่างไปจากข้อมูลเดิม และทำข้อมูลคงคลังของอุปกรณ์คอมพิวเตอร์ที่อยู่ใน Store และในระบบยังประกอบด้วย Agent ที่ได้ติดตั้งไว้ในเครื่อง Client ทุกเครื่องในระบบ โดยมีหน้าที่หลักในการตรวจสอบอุปกรณ์คอมพิวเตอร์ของตนเอง และส่งข้อมูลไปยัง Manager เพื่อบันทึกข้อมูล และยังทำหน้าที่ในการวิเคราะห์และเปรียบเทียบกับข้อมูลที่จัดเก็บอยู่ ทั้งนี้ Manager และ Agent ได้ติดต่อกันผ่านเครือข่าย TCP/IP



รูปที่ 3.1 ภาพรวมของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพรวมของระบบสามารถนำมาออกแบบ และเขียน Context Diagram และแผนภาพการไหลของข้อมูล (Data Flow Diagram) ได้ดังนี้



รูปที่ 3.2 Context Diagram ของระบบ

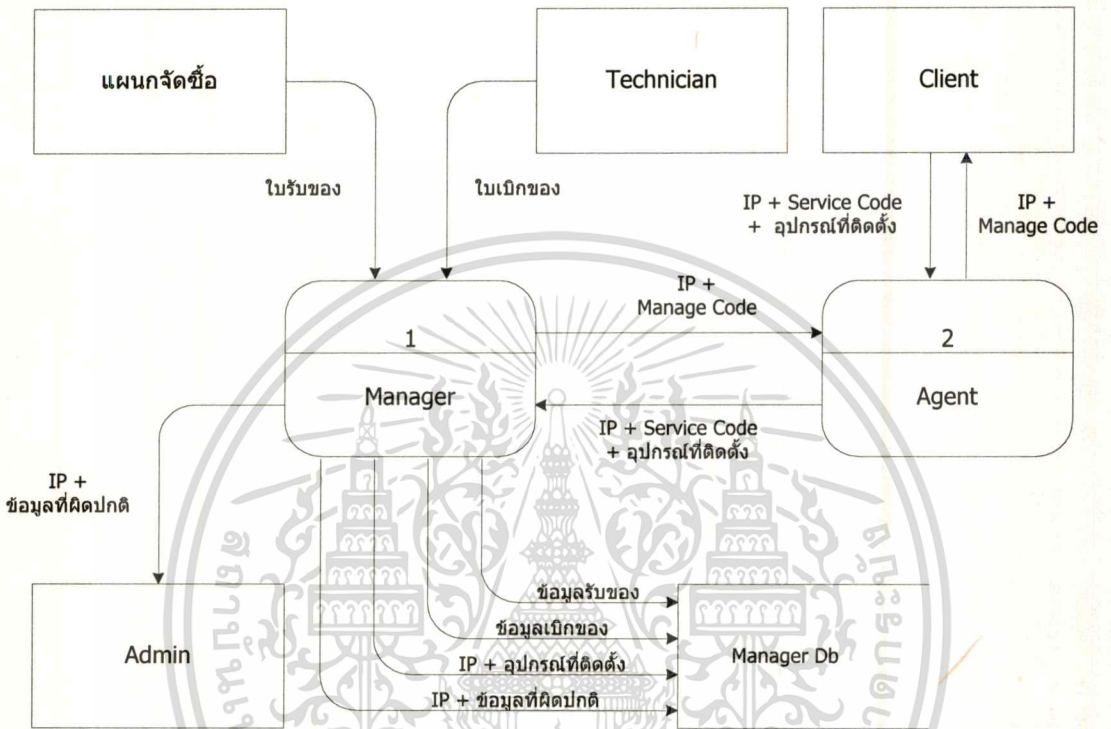
จากรูป 3.2 สามารถอธิบายได้ว่า ระบบควบคุมอุปกรณ์คอมพิวเตอร์นั้นรับข้อมูลจากภายนอกระบบด้วยกัน 3 แหล่งคือ

- ข้อมูลอุปกรณ์คอมพิวเตอร์ที่ติดตั้งในเครื่องจาก Client
- ข้อมูลการรับของจากแผนก Purchase
- ข้อมูลการเบิกของจาก Technician

เมื่อพบความผิดปกติของอุปกรณ์คอมพิวเตอร์ที่ติดตั้งที่เครื่อง Client ระบบจะแจ้งให้ Administrator รับทราบ

ส่วนการทำงานของระบบ แบ่งการทำงานออกเป็น 2 ส่วนคือ Manager ทำหน้าที่เป็นศูนย์กลางในการเก็บรวบรวมข้อมูลคอมพิวเตอร์ ข้อมูลรายการอุปกรณ์คอมพิวเตอร์ที่ติดตั้งที่เครื่อง Client และควบคุมรายการอุปกรณ์คอมพิวเตอร์ที่อยู่ใน Store และส่วน Agent ที่ทำงานอยู่ตาม Client เพื่อส่งข้อมูลอุปกรณ์คอมพิวเตอร์ให้ Manager ทราบเพื่อบันทึกข้อมูล และตรวจสอบข้อมูล เอกสารนี้เป็นเอกสารที่ส่งงานไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาติเห็นใบเช็คประเขยชนตามการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รวมทั้งการทำงานอื่นๆตามที่ Manager ต้องการเพื่อจัดการกับ Client ที่ติดตั้งอุปกรณ์คอมพิวเตอร์ที่แตกต่างไปจากข้อมูลที่มีอยู่ สามารถเขียนเป็น Data Flow Diagram ได้ดังนี้



รูปที่ 3.3 Data Flow Diagram Level 1 ของระบบ

### 3.1 การออกแบบการทำงานของ Manager

เนื่องจาก Manager เป็นส่วนที่ทำหน้าที่เป็นศูนย์กลางในการทำทะเบียนคอมพิวเตอร์ในองค์กร บันทึกข้อมูลอุปกรณ์คอมพิวเตอร์ที่ติดตั้งไว้ที่ Client ต่างๆ และรวมทั้งมีหน้าที่ในการบันทึกข้อมูล Client ที่ผิดปกติที่ Agent ส่งมาให้ และดำเนินการจัดการกับเครื่อง Client เหล่านั้น รวมถึงการจัดการข้อมูลเข้า-ออกของอุปกรณ์คอมพิวเตอร์ใน Store ด้วย ดังนั้นสามารถออกแบบ Module หลักๆ ได้ดังนี้

1. Module ข้อมูลทะเบียนคอมพิวเตอร์ เพื่อเก็บข้อมูลพื้นฐานที่สนับสนุนการทำงานของระบบ ได้แก่ข้อมูล IP Address, ชื่อเครื่อง, ส่วนงาน, ชื่อเจ้าของเครื่อง และสถานที่ตั้ง
2. Module บันทึกข้อมูลการติดตั้งอุปกรณ์คอมพิวเตอร์ โดยผ่านการทำงานของ Agent ที่ทำงานอยู่ประจำเครื่องนั้นๆ โดยเมื่อมีการติดตั้งเครื่อง Client ใหม่ จะมีการส่งข้อมูลอุปกรณ์ของ

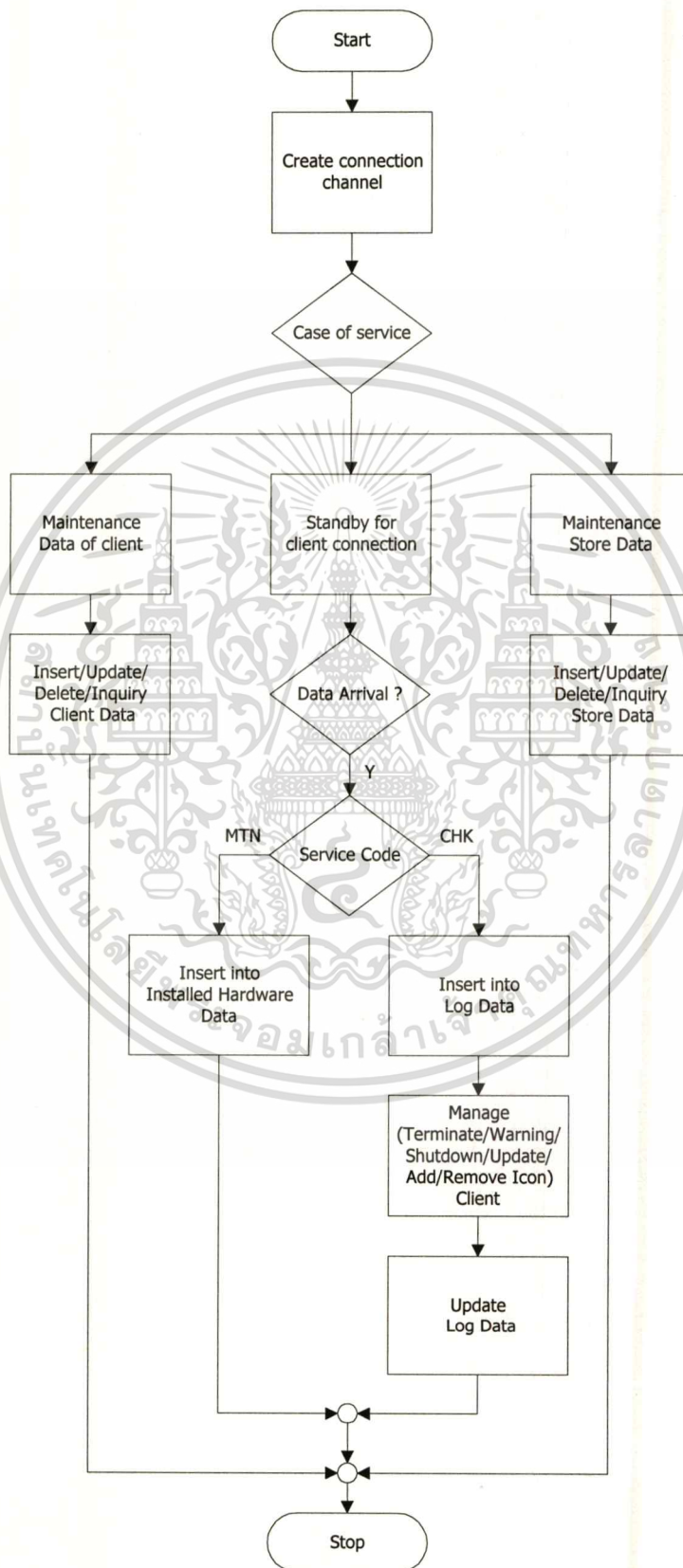
เครื่อง Client นั้นๆ มาให้ Manager เพื่อทำการบันทึกข้อมูลลงในฐานข้อมูลไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Module บันทึกข้อมูลที่ผิดปกติ โดย Agent จะเปรียบเทียบข้อมูลอุปกรณ์คอมพิวเตอร์ที่ติดตั้งที่อ่านได้ กับข้อมูลที่บันทึกไว้ ถ้าไม่ตรงกัน จะแจ้งให้ Manager ทราบ แล้ว Manager จะบันทึกข้อมูลนั้นไว้
4. Module จัดการกับเครื่อง Client ที่ได้ติดตั้งอุปกรณ์คอมพิวเตอร์ที่แตกต่างออกไปจากข้อมูลบันทึกไว้ โดยมีระดับของการจัดการเป็น 4 ระดับ คือ
  - 4.1 Update config เพื่อ Update ข้อมูลอุปกรณ์คอมพิวเตอร์ที่ติดตั้ง
  - 4.2 Terminate เพื่อ Terminate โปรแกรมที่ฝังตัวอยู่ที่ Agent
  - 4.3 Message Warning โดยแจ้งข้อความเตือนขอให้เจ้าของเครื่อง Client รับทราบ
  - 4.4 Shutdown เครื่อง Client นั้นๆ ถ้าพบว่าไม่มีการแก้ไขหลังจากที่ได้แจ้งด้วย Message warning
5. Module จัดการคงคลังขององค์กร ในส่วนของ Store ที่มีการรับของเข้า และมีการจ่ายของออกไป

จากข้อมูลข้างต้น สามารถสรุปลักษณะการทำงานของ Manager เป็นขั้นตอนได้ ดังนี้

1. สร้าง Socket เพื่อสร้างช่องทางการติดต่อกับ Agent
2. Manager รับข้อมูลอุปกรณ์คอมพิวเตอร์ที่ติดตั้งที่เครื่อง Client จาก Agent โดยอ้างอิงผ่าน IP Address และ Service Code ว่าเป็นการ Maintenance หรือการตรวจสอบ ผ่าน Port ที่ให้บริการ
3. สามารถจำแนกงานออกเป็น 2 ลักษณะงาน คือ
  - 3.1 สำหรับการ Maintenance เครื่อง Manager จะจัดเก็บหรือแก้ไขข้อมูลลงฐานข้อมูลที่มีอยู่
  - 3.2 สำหรับการตรวจสอบเครื่อง Manager จะบันทึกข้อมูลที่ผิดปกติเอาไว้ เพื่อให้ผู้ดูแลระบบทราบ และให้ผู้ดูแลระบบจัดการกับ Client นั้นๆ โดยส่ง Manage Code ไปยัง Agent ของ Client นั้นๆ
4. เมื่อมีการนำอุปกรณ์คอมพิวเตอร์เข้ามา หรือการเบิกรูอุปกรณ์คอมพิวเตอร์ไปจาก Store ต้องมีการบันทึกทุกครั้ง เพื่อเป็นการปรับยอดข้อมูลใน Store  
สามารถเขียนแผนภาพ Flow Chart การทำงานของ Manager ได้ดังนี้



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

รูปที่ 3.4 Flow Chart แสดงการทำงานของ Manager

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2 การออกแบบการทำงานของ Agent

Agent เป็นส่วนที่ทำงานอยู่ที่เครื่อง Client โดยมีหน้าที่หลักในการตรวจสอบอุปกรณ์คอมพิวเตอร์ที่ติดตั้งไว้ ณ เครื่อง Client ที่ Agent ทำงานอยู่ และส่งข้อมูลไปให้ Manager เพื่อทำการบันทึก หรือตรวจสอบกับข้อมูลที่มีอยู่จริง นอกจากนี้ตัว Agent ยังให้บริการอื่นตามที่ Manager ต้องการเช่น การแสดงข้อความแจ้งเตือน, การ shutdown ระบบ รวมทั้งการซ่อน/แสดง Icon ของ Agent บน System Tray ดังนั้นสามารถออกแบบการทำงานของได้ดังนี้

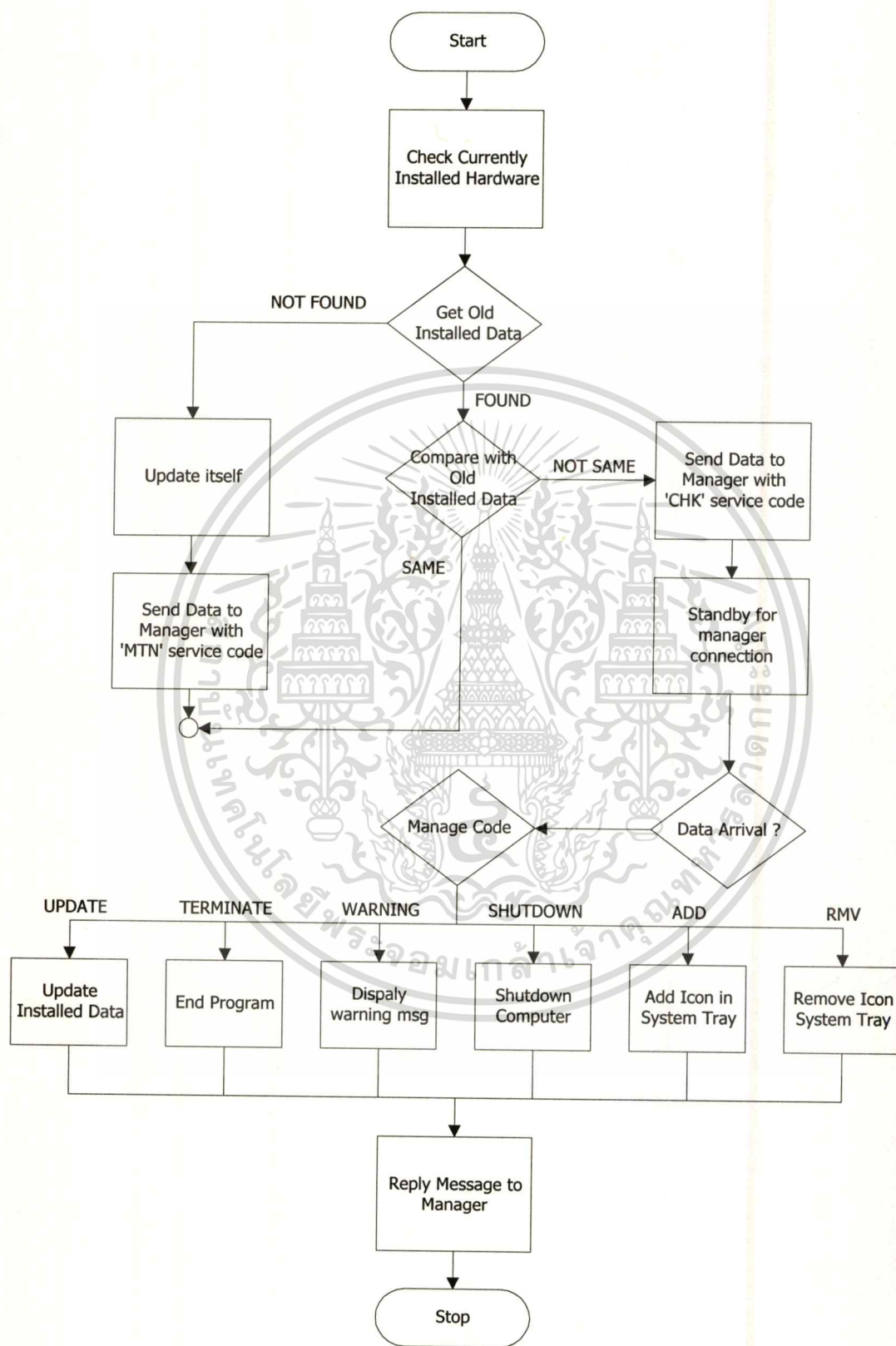
1. Module Hardware Servey โดยการตรวจสอบอุปกรณ์คอมพิวเตอร์ที่ติดตั้งไว้ ณ เครื่องตัวเอง
2. Module Update Config ทำหน้าที่ Update ข้อมูลอุปกรณ์คอมพิวเตอร์ที่เครื่องตนเอง
3. Module Terminate ทำหน้าที่ Terminate โปรแกรม
4. Module Message Warning ทำหน้าที่แสดง Warning Message เมื่อได้รับจากการร้องขอจาก Manager
5. Module Shutdown ระบบทำหน้าที่ในการ Shutdown ระบบเมื่อได้รับ การร้องขอจาก Manager
6. Module การซ่อน/แสดง Icon ของ Agent บน System Tray

ลักษณะการทำงานของ Agent เป็นลักษณะการทำงานแบบ Service ที่ Run ฝังตัวอยู่ที่เครื่อง Client โดยในส่วนของการตรวจสอบอุปกรณ์คอมพิวเตอร์ จะทำงานทุกครั้งเมื่อเปิดเครื่องคอมพิวเตอร์ ถ้าเป็นเครื่องใหม่ที่ไม่มีการบันทึกข้อมูลมาก่อน Agent จะส่งข้อมูลไปให้ Manager เพื่อบันทึกข้อมูล แต่ถ้าเป็นเครื่องที่มีการบันทึกข้อมูลแล้ว Agent จะตรวจสอบว่าตรงกับข้อมูลที่มีอยู่หรือไม่ ถ้าตรงกัน จะปิดตัวเองลงไปทันที แต่ถ้าไม่ตรงกัน ผู้ดูแลระบบต้องมีการจัดการกับ Client นั้นๆ ทำให้เครื่อง Client ต้อง Stand by โดยการเปิด Socket รอการ Connect จาก Manager ได้ตลอดเวลา โดยสามารถสรุปเป็นขั้นตอนการทำงานของ Agent ได้ดังนี้

1. เริ่ม Start เครื่อง Agent จะทำการตรวจสอบอุปกรณ์คอมพิวเตอร์ในเครื่องตัวเอง
2. ถ้าเป็นการติดตั้งเครื่องใหม่ หรือเครื่องเก่าที่พบอุปกรณ์คอมพิวเตอร์ที่อ่านค่าได้ไม่ตรงกับข้อมูลที่มีอยู่ จะทำงานในขั้นตอนต่อไป แต่ถ้าไม่ใช่ ก็จะสิ้นสุดการทำงานของ Agent
3. สร้าง Socket เพื่อเชื่อมต่อการสื่อสารกับ Manager
4. สำหรับเครื่องใหม่ที่ไม่มีข้อมูลมาก่อน จะส่ง Service Code และข้อมูลอุปกรณ์คอมพิวเตอร์ไปยัง Manager เพื่อบันทึกข้อมูล เป็นอันสิ้นสุดการทำงาน สำหรับเครื่องเก่า จะส่ง Service Code และข้อมูลอุปกรณ์ที่มีค่าที่อ่านได้ ไม่ตรงกับค่าที่มีอยู่ไปยัง Manager เพื่อบันทึกข้อมูลเช่นกัน แต่ต้องรอการตอบกลับจาก Manager
5. ในกรณีที่มีการ Connect จาก Manager ก็ทำการรับ Connection พร้อมทั้งทำการรับ Message มาทำการตรวจสอบว่าเป็น Manager Code แบบไหน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษายเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 Flow Chart แสดงการทำงานของ Agent

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. Agent ทำตาม Manage Code ที่ทาง Manager ต้องการ
7. ในกรณีที่สมารถทำได้สำเร็จ Agent ก็ทำการส่ง Output กลับไปยัง Manager และกรณีที่ไม่สามารถทำได้สำเร็จ ทาง Agent ทำการส่ง Error Message ไปบอก Manager สามารถเขียนแผนภาพ Flow Chart การทำงานของ Agent ได้ดังรูปที่ 3.5

### 3.3 การออกแบบการติดต่อสื่อสารระหว่าง Agent และ Manager

ในการติดต่อสื่อสารระหว่าง Manager และ Agent ได้ใช้การติดต่อสื่อสารผ่านเครือข่าย TCP/IP โดยใช้ Protocol TCP ในการติดต่อ และหมายเลข Port 2708 สำหรับ Manager และ 2709 สำหรับ Agent และได้เปิด Socket รอการเชื่อมต่อ เมื่อ Agent ต้องการส่งข้อมูลไปให้ Manager ทำได้โดยระบุหมายเลข IP Address ของ Manager และหมายเลข Service Port ของ Application พร้อมทั้ง Service Code และข้อมูลที่ต้องการส่งไปให้ Manager เช่นเดียวกัน โดยมี Service Code ดังนี้

ตารางที่ 3.1 Service Code ที่ใช้ในการร้องขอและให้บริการของ Manger

Code	Type	Description
MTN	Service	เพิ่มข้อมูลบนฐานข้อมูล
CHK	Service	ตรวจสอบข้อมูล

โดยที่มีรูปแบบ Message ในการส่งดังนี้

1. รูปแบบ Message สำหรับ เพิ่มเติมข้อมูลบนฐานข้อมูล

**รูปแบบ**      *Service Code | IP : IP Address | OS : OS Information | Hardware Type : Hardware Information | Hardware Type : Hardware Information | ...*

เช่น *MTN | IP : 127.0.0.1 | OS : Window 98 Version 4.10 A | Processor : Speed 600 MHz GenuineIntel Celeron | RAM : Size 127 MB | HD : Size 6 GB Serial 13F9-171A | LAN : MAC Address 00-60-08-CF-4F-49 | VGA : ATI 3D Rang Pro (atir3) | Monotor : XJ770 | Keyboard : Standard 101/102-Key | Mouse : PS/2 Compatible Mouse Port*

2. รูปแบบ Message สำหรับ ตรวจสอบข้อมูล

**รูปแบบ**      *Service Code | IP : IP Address | Hardware Type : Error Information*

เช่น *CHK | IP : 127.0.0.1 | RAM : Size 127 MB -> Size 64 MB*

เมื่อ Manager ต้องการติดต่อกับ Agent ที่อยู่ที่ Client ทำได้โดยการระบุหมายเลข IP ของเครื่อง Client ที่ต้องการจัดการพร้อมทั้งระบุหมายเลข Service Port ของ Application พร้อมทั้งบอก Manage Code ของการจัดการแล้วส่งไปยัง Agent ณ เครื่อง Client ที่ต้องการติดต่อด้วย สำหรับ Service Code และ Manage Code ของการให้บริการและการจัดการสามารถกำหนดได้ดังนี้

**ตารางที่ 3.2** Manage Code ที่ใช้ในการจัดการกับ Agent

Code	Type	Description
Update	Manage	แก้ไขข้อมูลบนที่ Client
Terminate	Manage	หยุดการทำงานของโปรแกรม
Warning	Manage	แสดงข้อความแจ้งเตือน
Shutdown	Manage	ทำการ Shutdown เครื่อง Client
Add	Manage	แสดง Agent Icon บน System tray
Remove	Manage	ซ่อน Agent Icon บน System tray

โดยที่มีรูปแบบ Message ในการส่งเป็น

รูปแบบ *Manager Code | Manage Information*

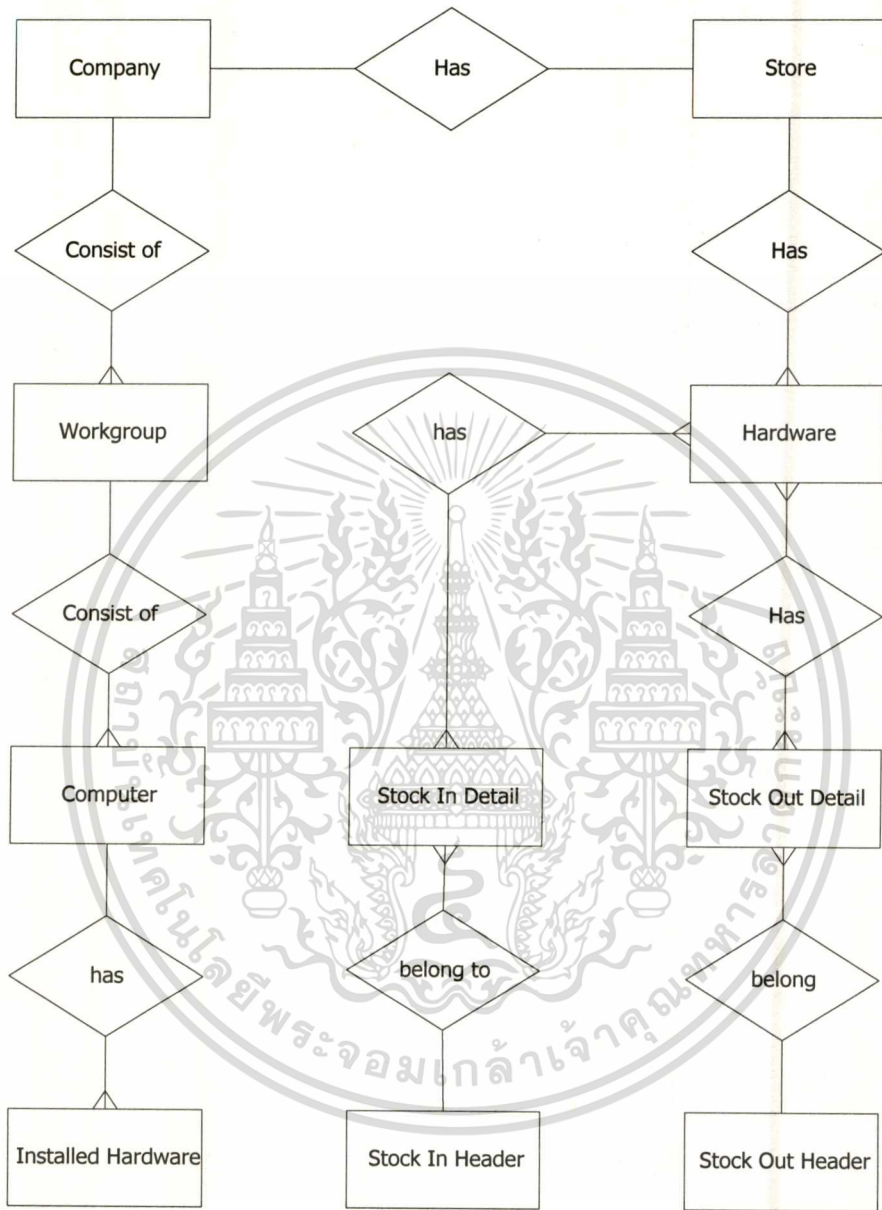
โดยที่ เฉพาะ Manage Code ที่เป็น Warning เท่านั้นที่มี Manage Information เป็นข้อความสำหรับแจ้งเตือน ส่วน Manage Code อื่น ส่งเฉพาะ Manage Code อย่างเดียว

### 3.4 การออกแบบฐานข้อมูล

จากการออกแบบการทำงานของ Manager ซึ่งมีหน้าที่หลักในการเก็บรวบรวมข้อมูลอุปกรณ์คอมพิวเตอร์ทั้งหมด สามารถนำมาออกแบบข้อมูลโดยการเขียน Entity Relationship Diagram และออกแบบตารางจัดเก็บข้อมูลได้ดังรูปที่ 3.6

จาก Entity Relationship Diagram สามารถอธิบายได้ว่าในองค์กรนั้นประกอบไปด้วยหลายส่วนงาน (Workgroup) ซึ่งในแต่ละส่วนงานนั้น ประกอบด้วยเครื่องคอมพิวเตอร์ (Computer) จำนวนหลายเครื่อง ในแต่ละเครื่องนั้นได้ติดตั้งอุปกรณ์คอมพิวเตอร์หลายชนิด และในองค์กรยังต้องมี Store สำหรับสำรองเก็บอุปกรณ์คอมพิวเตอร์ ซึ่งการนำเข้ามาหรือนำออกไปจาก Store ต้องผ่านเอกสารใบนำเข้า หรือใบเบิกของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 Entity Relationship Diagram ของระบบ

จาก Entity Relationship Diagram นี้ สามารถนำมาออกแบบตารางการจัดเก็บข้อมูลได้ดังนี้

นี้

ตารางที่ 3.3 Table : Workgroup ข้อมูลส่วนงาน

Field Name	Field Type	Key	Description
WGCode	Char (5)	PK	Workgroup Code
WGName	Char (50)	-	Workgroup Name
WGLoc	Char (50)	-	Workgroup Location
PersonInCharge	Char (50)	-	Contact Person

ตารางที่ 3.4 Table : Computer ข้อมูลคอมพิวเตอร์

Field Name	Field Type	Key	Description
CompIP	Char (15)	PK	IP Address Number
CompName	Char (30)	-	Computer Name
CompGroupCode	Char (5)	-	Computer Group
CompLoc	Char (50)	-	Computer Location
CompOwner	Char (50)	-	Computer Owner

ตารางที่ 3.5 Table : InstalledHardware ข้อมูลอุปกรณ์คอมพิวเตอร์ที่ติดตั้งที่คอมพิวเตอร์

Field Name	Field Type	Key	Description
CompIP	Char (15)	PK	IP Address Number
OS	Char (50)	-	OS Description
Processor	Char (50)	-	Processor Description
RAM	Char (50)	-	RAM Description
HardDisk	Char (50)	-	Harddisk Description
LANCard	Char (50)	-	LAN Card Description
VGACard	Char (50)	-	VGA Card Description

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์ทางการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Field Name	Field Type	Key	Description
Monitor	Char (50)	-	Monitor Description
Keyboard	Char (50)	-	Keyboard Description
Mouse	Char (50)	-	Mouse Description

ตารางที่ 3.6 Table : Hardware ข้อมูลอุปกรณ์คอมพิวเตอร์ใน Store

Field Name	Field Type	Key	Description
HWCode	Char (10)	PK	Hardware Code
HWDesc	Char (50)	-	Hardware Description
Quantity	Number (3)	-	Quantity

ตารางที่ 3.7 Table : StockIn Header ข้อมูลการรับของเข้า Store

Field Name	Field Type	Key	Description
StockInNo	Char (10)	PK	Number of Document
TxDate	Date	-	Transaction Date
Name	Char (50)	-	Name

ตารางที่ 3.8 Table : StockIn\_Item ข้อมูลรายละเอียดอุปกรณ์คอมพิวเตอร์ที่รับเข้ามาใน Store

Field Name	Field Type	Key	Description
StockInNo	Char (10)	PK	Number of Document
HWCode	Char (5)	PK	Hardware Code
Quantity	Number (3)	-	Quantity

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.9 Table : StockOut\_Header ข้อมูลการนำของออกไปจาก Store

Field Name	Field Type	Key	Description
StockOutNo	Char (10)	PK	Number of Document
TxDate	Date	-	Transaction Date
Name	Char (50)	-	Name

ตารางที่ 3.10 Table : StockOut\_Item ข้อมูลรายละเอียดอุปกรณ์คอมพิวเตอร์ที่นำออกไปจาก Store

Field Name	Field Type	Key	Description
StockOutNo	Char (10)	PK	Number of Document
HWCode	Char (5)	PK	Hardware Code
Quantity	Number (3)	-	Quantity

ตารางที่ 3.11 Table : Log เก็บประวัติและรายละเอียด error ต่างๆ

Field Name	Field Type	Key	Description
LogDate	Date	PK	Log Date
CompIP	Char (15)	PK	Computer IP Address
ErrType	Char (25)	-	Error Type
ErrMsg	Char (100)	-	Error Message
Remark	Char (100)	-	Remark

## บทที่ 4

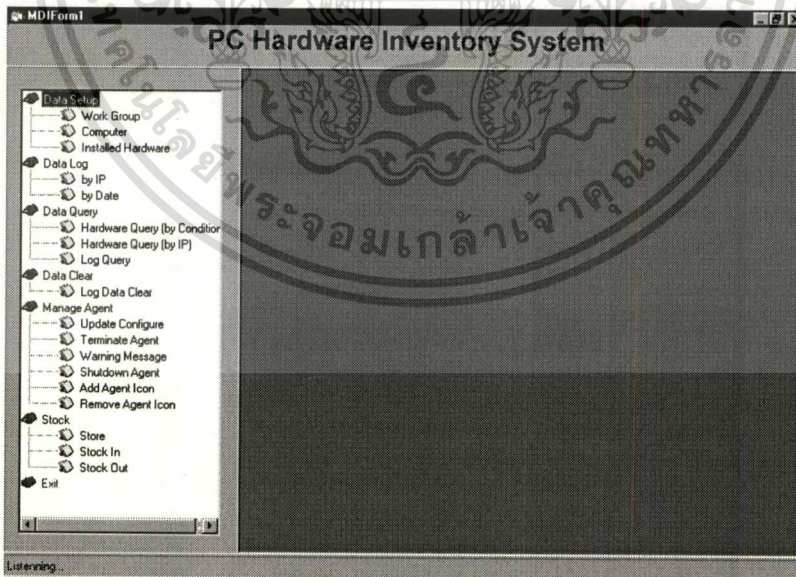
### การพัฒนาระบบ

ในขั้นตอนของการพัฒนานี้ระบบ นั้น ได้พัฒนาส่วนงานต่างๆ ( Module ) ตามที่ได้ออกแบบเอาไว้ โดยใช้เครื่องมือในการพัฒนาดังนี้

1. เครื่อง Computer PC ที่มีติดตั้ง Windows 95/98 พร้อมทั้งสามารถติดต่อกับเครือข่าย TCP/IP
2. Microsoft Visual Basic 6.0 และ Component ต่างๆที่จำเป็นในการพัฒนาส่วนงานด้านการติดต่อสื่อสารกับเครือข่าย และ Component ต่างๆที่จำเป็นในการติดต่อกับ Database
3. Microsoft Office 97 สำหรับจัดทำเอกสารประกอบการพัฒนาระบบ และทำส่วนการ Presentation

#### 4.1 การพัฒนาส่วนการทำงานของ Manager

การทำงานในส่วน Manager สามารถแสดงได้ ดังรูป



รูปที่ 4.1 การทำงานของ Manager

การพัฒนาส่วนการทำงานของ Manager เป็นการพัฒนาส่วนที่มีหน้าที่ในการเก็บข้อมูลคอมพิวเตอร์ ข้อมูลรายการอุปกรณ์คอมพิวเตอร์ที่ได้ติดตั้งไว้ ณ เครื่อง Client ต่างๆ รวมทั้งไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บันทึกข้อมูลที่มีความผิดปกติ ที่ Client ตรวจพบ และทำหน้าที่ในการจัดการกับเครื่อง Client ที่พบว่าได้เปลี่ยนแปลงอุปกรณ์ไปจากข้อมูลที่มีอยู่ นอกจากนี้ในส่วนของ Manager ยังทำหน้าที่ในการจัดการข้อมูลใน Store ด้วย

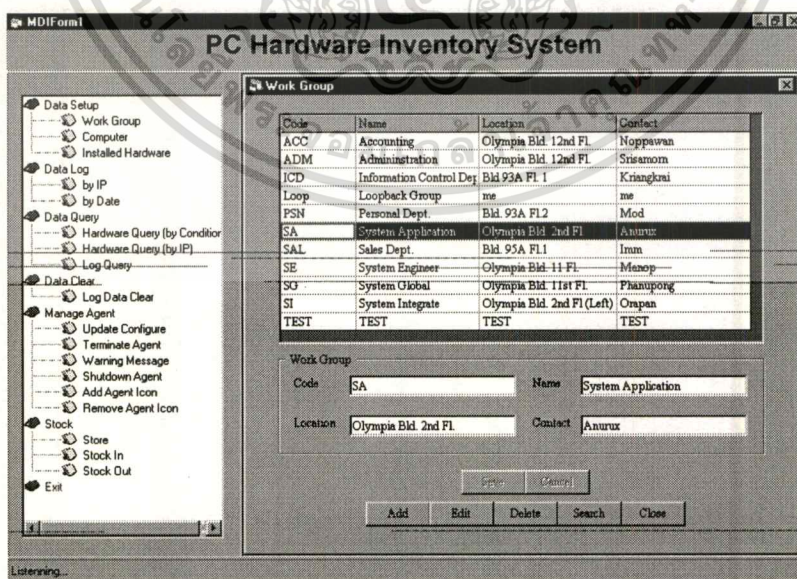
#### 4.1.1 การพัฒนาส่วนการจัดการข้อมูล

เป็นส่วนที่ใช้ในการจัดการกับข้อมูลของ Manager เพื่อประกอบในการพัฒนาระบบ เป็นการเก็บข้อมูลพื้นฐานของคอมพิวเตอร์ ในการจัดการข้อมูลแต่ละส่วนนั้นจะประกอบไปด้วย Function ที่เหมือนกัน ดังนี้

- Add เป็น Function สำหรับการเพิ่มรายการใหม่ลงไปในฐานะข้อมูล
- Edit เป็น Function สำหรับการแก้ไขรายการข้อมูลในฐานะข้อมูล
- Delete เป็น Function สำหรับการลบรายการข้อมูลออกจากรฐานข้อมูล
- Search เป็น Function สำหรับการค้นหารายการในฐานะข้อมูล

โดยที่ข้อมูลที่จะบันทึกประกอบไปด้วยข้อมูลต่างๆดังนี้

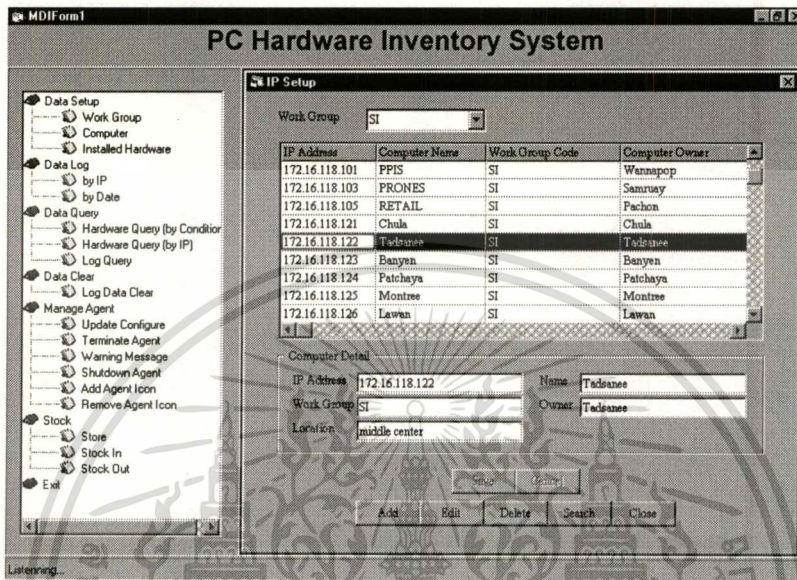
1. ข้อมูล Workgroup เป็นข้อมูลของส่วนงาน ในระบบคอมพิวเตอร์ เราจะจัดกลุ่มคอมพิวเตอร์ ออกเป็นกลุ่มหน่วยงานที่ทำงานในกลุ่มเดียวกัน เพื่อให้ง่ายแก่การควบคุม โดยจะเก็บข้อมูลของ รหัส ชื่อ ที่ตั้ง และผู้ดูแลรับผิดชอบ ซึ่งสามารถแสดงได้ ดังรูป



รูปที่ 4.2 การจัดการข้อมูล Workgroup

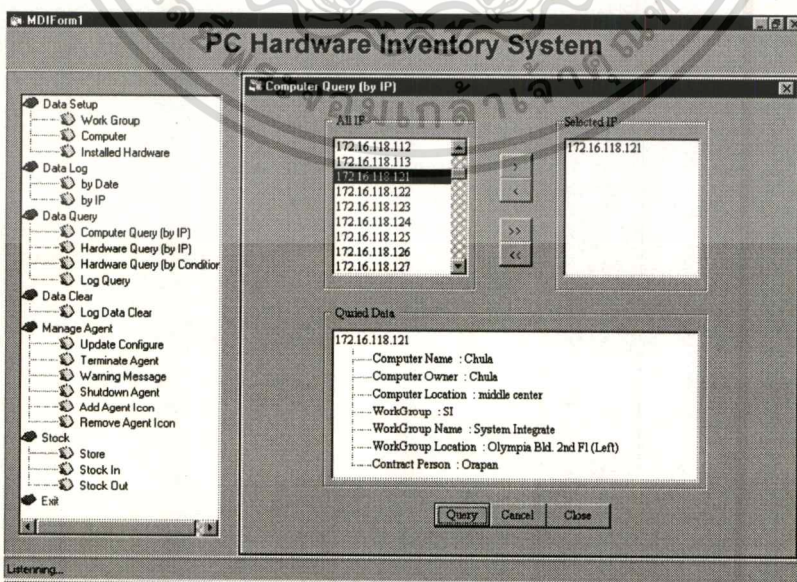
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ข้อมูล Client แต่ละเครื่อง เช่น หมายเลข IP ชื่อคอมพิวเตอร์ กลุ่มการทำงาน ชื่อเจ้าของเครื่อง และที่ตั้งซึ่งสามารถแสดงได้ดังรูป



รูปที่ 4.3 การจัดการข้อมูลคอมพิวเตอร์

นอกจากนี้ ยังมีส่วนสำหรับ Query ข้อมูลของคอมพิวเตอร์ ซึ่งสามารถแสดงได้ดังรูป



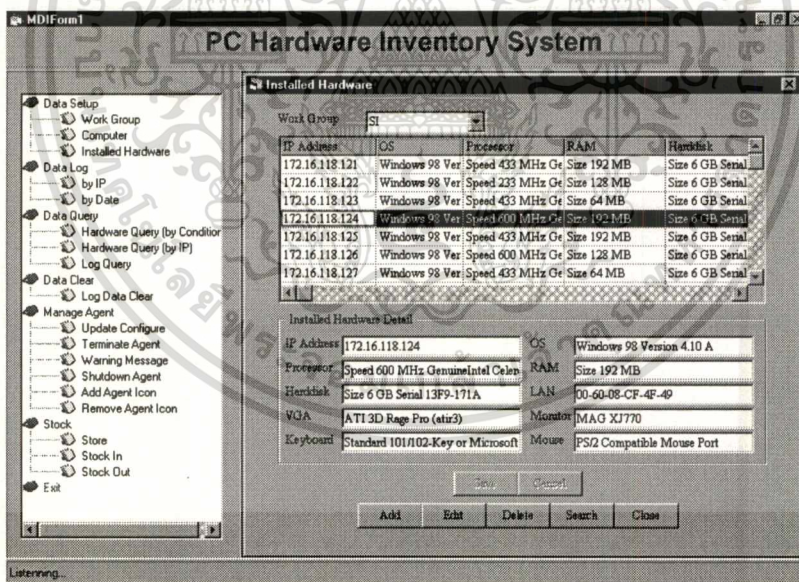
รูปที่ 4.4 การ Query ข้อมูลคอมพิวเตอร์ด้วยหมายเลข IP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญตเห็นนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.1.2 การพัฒนาส่วนการรวบรวมรายการอุปกรณ์คอมพิวเตอร์

ในการใช้งานจริงนั้น เมื่อมีการติดตั้งคอมพิวเตอร์ ณ เครื่องคอมพิวเตอร์ใดๆ เมื่อเครื่องคอมพิวเตอร์อ่านค่าอุปกรณ์ติดตั้งแล้ว จะเก็บข้อมูลไว้ที่เครื่องตัวเองเพื่อใช้เป็นข้อมูลในการอ้างอิงครั้งต่อไป และต้องส่งข้อมูลอุปกรณ์คอมพิวเตอร์ที่ติดตั้งมายัง Manager เพื่อบันทึกเพิ่มข้อมูลลงในฐานข้อมูล และเป็นการเก็บรวบรวมข้อมูลของอุปกรณ์คอมพิวเตอร์ไว้ด้วยกันทั้งองค์กร โดยส่ง Service Code เป็น Maintenance (MTN) โดยมีหลักการทำงานคือ เมื่อ Manager ได้รับข้อมูลจะบันทึกข้อมูลนั้นไว้ในฐานข้อมูลของตนเอง ซึ่งข้อมูลอุปกรณ์ที่เก็บจะได้แก่ IP ของเครื่อง, OS ที่ใช้, ข้อมูลของ Processor (ความเร็ว, ผู้ผลิต และรุ่น), ข้อมูลของ RAM (Size), ข้อมูลของ Harddisk (Serial No และ Size), ข้อมูลของ LAN Card (MAC Address), ข้อมูลของ VGA Card (Model), ข้อมูลของจอภาพ, คีย์บอร์ด และเมาส์ ซึ่งสามารถดูได้ดังรูปที่ 4.5 และเพื่อให้การเรียกดูอุปกรณ์ที่ติดตั้งในแต่ละเครื่องคอมพิวเตอร์มีความละเอียดมากขึ้น จึงได้เพิ่มการทำงานในส่วน query ข้อมูล เพื่อให้ง่าย และตรงตามวัตถุประสงค์ของผู้ใช้มากที่สุดดังรูปที่ 4.6 และ 4.7

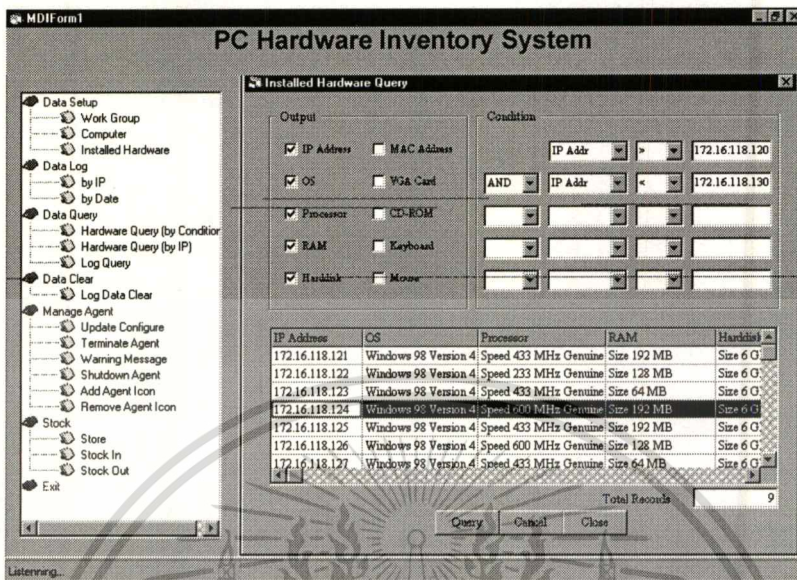


รูปที่ 4.5 การจัดการข้อมูลอุปกรณ์คอมพิวเตอร์ที่ติดตั้งในแต่ละเครื่อง

การ Query ข้อมูล สามารถแบ่งงานได้ดังนี้ คือ

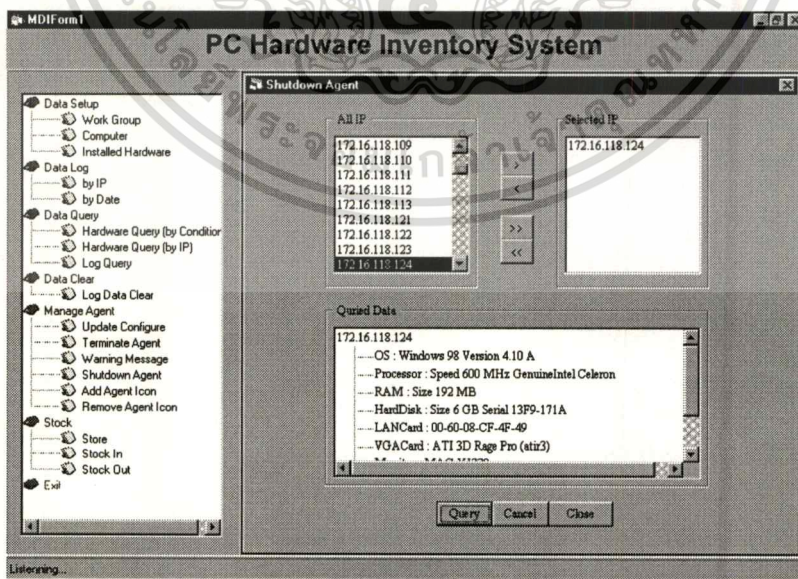
- กรณีที่ต้องการเลือกดูข้อมูลอุปกรณ์บางตัว และภายใต้เงื่อนไขที่สนใจเท่านั้น สามารถแสดงได้ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 การ Query ข้อมูล ด้วย Output และ เงื่อนไขที่กำหนดได้เอง

- กรณีที่ต้องการเลือกดูข้อมูลอุปกรณ์คอมพิวเตอร์ เฉพาะบางเครื่องคอมพิวเตอร์ที่สนใจเท่านั้น สามารถแสดงได้ดังรูป

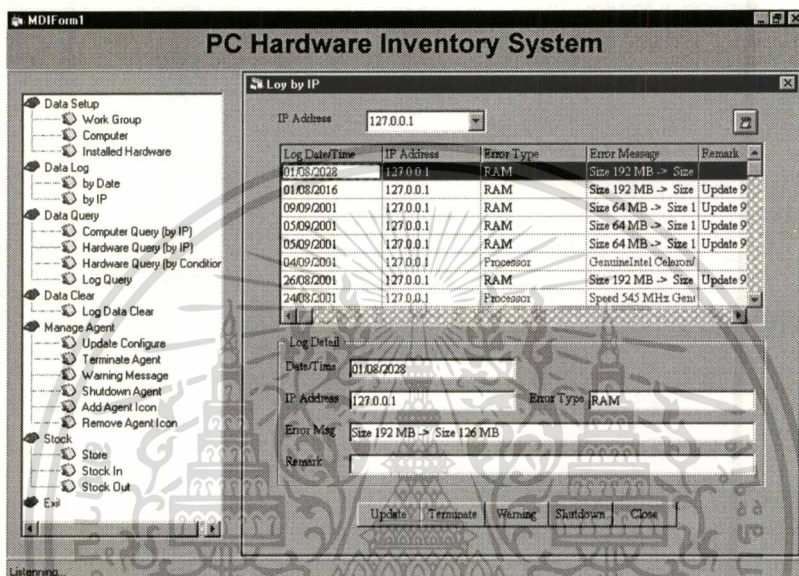


รูปที่ 4.7 การ Query ข้อมูลอุปกรณ์ที่ติดตั้งด้วยหมายเลข IP

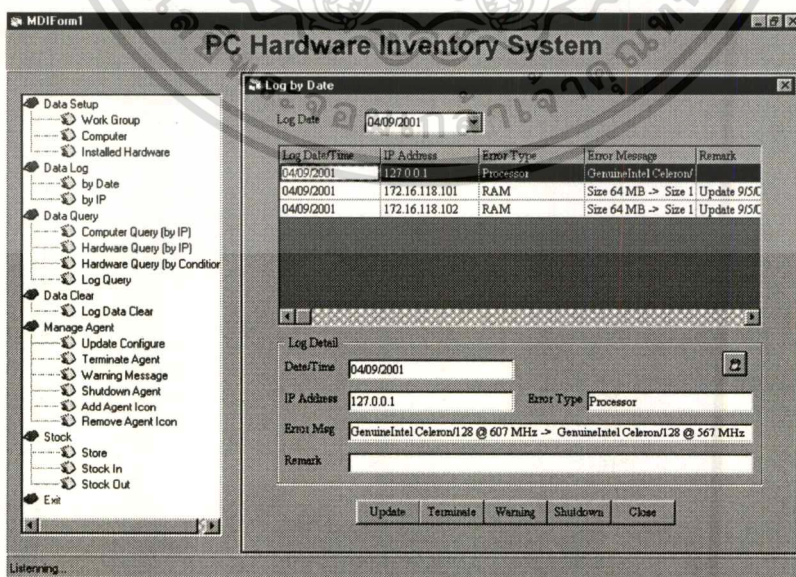
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.1.3 การพัฒนาส่วนการบันทึกข้อมูลที่ผิดพลาด

เป็นส่วนที่ใช้ในการบันทึกรายการอุปกรณ์คอมพิวเตอร์ที่ผิดพลาดที่ได้จาก Client ทุกครั้งของการติดตั้งเครื่องใหม่หรือเมื่อพบว่าข้อมูลที่เก็บ กับข้อมูลที่อ่านได้ไม่ตรงกัน ด้วย Service Code ที่มีค่าเป็น Checking (CHK) โดยมีหลักการทำงานคือ เมื่อ Manager ได้รับข้อมูล จะบันทึก



รูปที่ 4.8 ข้อมูล Log โดยจำแนกตาม IP Address

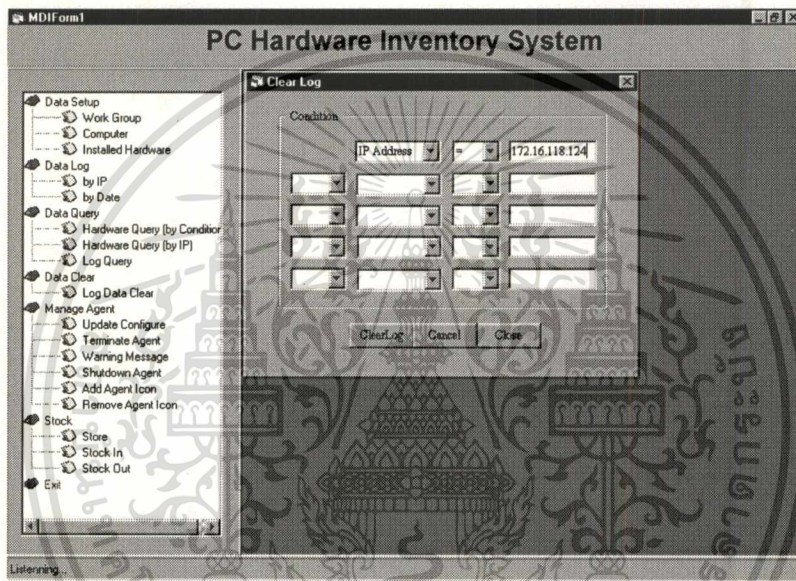


รูปที่ 4.9 ข้อมูล Log โดยจำแนกตามวันที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไว้ใน Log ซึ่งสามารถดูข้อมูลเหล่านี้ ได้ใน 2 ลักษณะคือ จำแนกตาม IP Address และจำแนกตามวันที่ Manager สามารถที่จะจัดการกับ Client ที่ผิดปกติเหล่านี้ ด้วย Manage Code ต่างๆ ตามที่กำหนด สามารถแสดงได้ ดังรูปที่ 4.8 และ 4.9

จะเห็นว่า ข้อมูล Log จะมีขนาดเพิ่มขึ้นเรื่อยๆ จึงต้องมีการทำงานสำหรับลบข้อมูลเก่าที่ไม่ได้ใช้งานออก เพื่อเพิ่มประสิทธิภาพของระบบ โดยกำหนดให้มีการ Clear Log ซึ่งสามารถกำหนดเงื่อนไขการลบข้อมูลได้ ดังแสดงไว้ ดังรูป



รูปที่ 4.10 การลบข้อมูล Log ตามเงื่อนไข

#### 4.1.4 การพัฒนาส่วน Agent Management

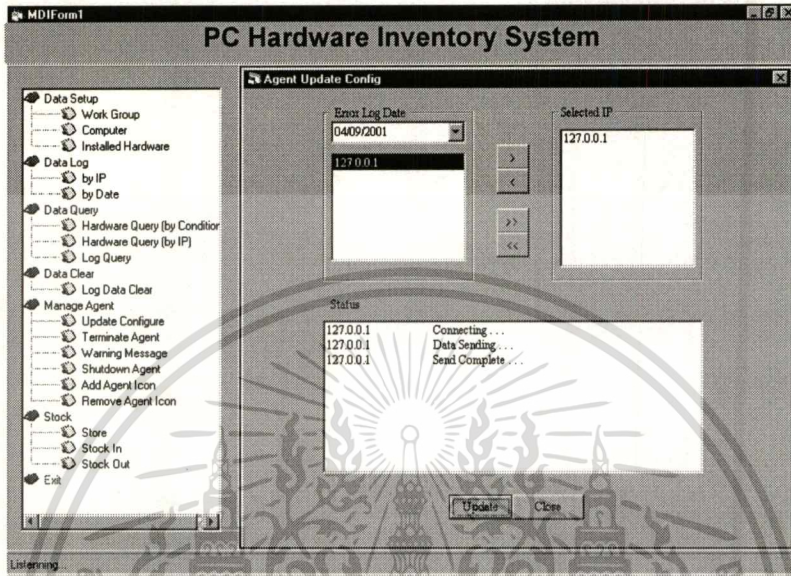
สามารถแบ่งได้เป็น 2 ส่วน คือ ส่วนที่ใช้จัดการ Agent ที่พบว่ามีข้อมูลผิดปกติ และส่วนที่ใช้ควบคุม Agent ซึ่งสามารถแบ่งงานได้ คือ

- ส่วนที่ใช้จัดการ Agent ที่พบว่ามีข้อมูลผิดปกติ ซึ่งอาจเป็นการ Update ข้อมูลหรือ Terminate โปรแกรม ที่ Agent เอง ในกรณีที่ข้อมูลที่ผิดปกติ นั้น เกิดจากการดูแลของผู้รับผิดชอบ แต่ถ้าไม่ใช่ มีการจัดการกับ Agent เหล่านั้น ซึ่งมีได้ตั้งแต่การเตือนเป็น Error Message และการ Shutdown เครื่อง สามารถแสดงได้ ดังรูป 4.11

ส่วนจัดการ Agent ทุกประเภทของการทำงาน จะมีลักษณะเหมือนกัน โดยที่จะมีฟังก์ชันการทำงานที่ต่างกัน เป็น Update, Terminate, Warning หรือ Shutdown ตามลักษณะการ

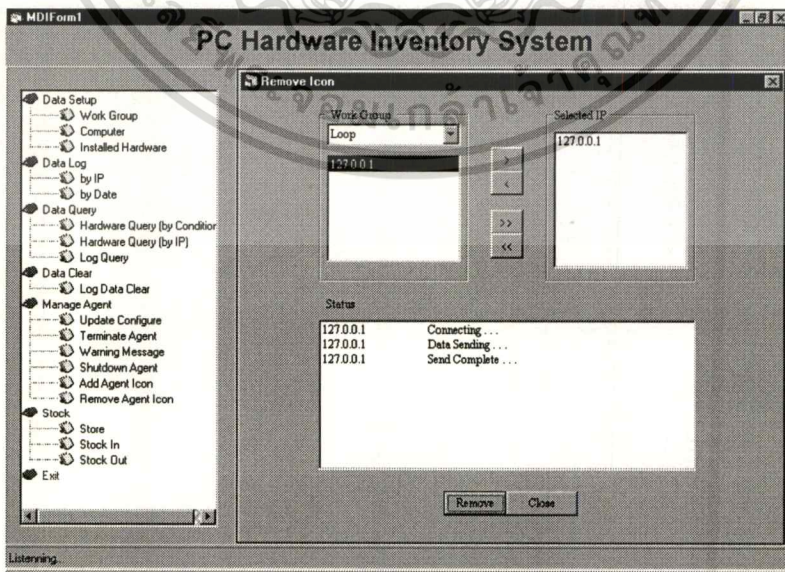
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงานนั้นๆ โดยข้อมูลคอมพิวเตอร์ที่แสดง จะเป็นข้อมูลที่มีความผิดปกติ ที่ถูกเก็บไว้เป็น Log ซึ่งสามารถแสดงการติดต่อให้เห็นได้ที่ Status



รูปที่ 4.11 การจัดการ Agent

- ส่วนที่ใช้ควบคุม Agent ได้แก่การ Show หรือ Remove Icon ของ Agent ใน System Tray



รูปที่ 4.12 การควบคุม Agent

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

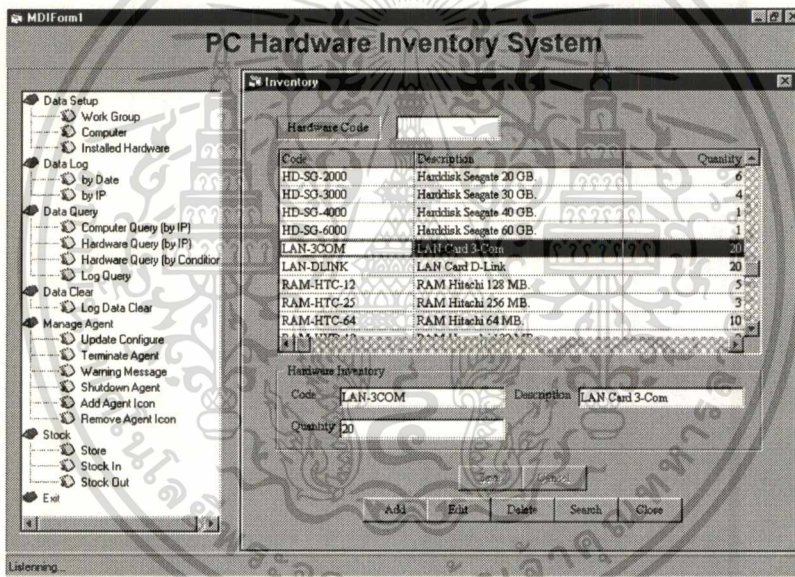
#### 4.1.5 การพัฒนาส่วนการควบคุมจำนวนอุปกรณ์คงคลังใน Store

เป็นการพัฒนาส่วนที่ใช้ควบคุมจำนวนอุปกรณ์คอมพิวเตอร์ที่อยู่ใน Store ซึ่งจะมีผลมาจาก การนำของเข้ามา หรือการเบิกของออกไปจาก Store ซึ่งข้อมูลเหล่านี้จะถูกนำไปตัดยอดกับจำนวนใน Store เพื่อเป็นการควบคุมการทำ Repair part

ในส่วนนี้ สามารถแบ่งการควบคุมได้เป็น 3 ส่วน คือ

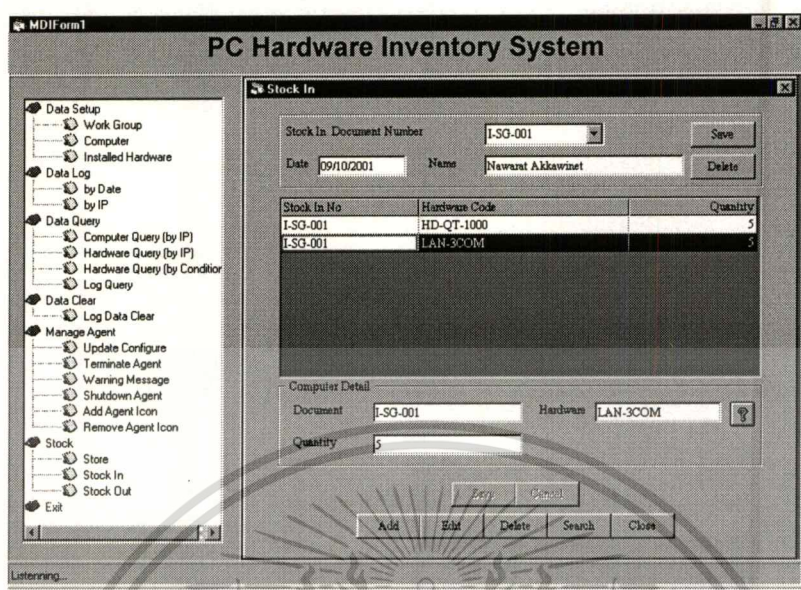
- ส่วน Inventory เป็นส่วนที่เก็บข้อมูลของอุปกรณ์คอมพิวเตอร์ที่มีอยู่ใน Store ซึ่งเก็บข้อมูลรหัสของอุปกรณ์, คำอธิบาย, จำนวน และหน่วย เพราะอุปกรณ์คอมพิวเตอร์แต่ละชิ้น ใช้ลักษณะการเก็บหน่วยไม่เหมือนกัน ขึ้นอยู่กับชนิดของอุปกรณ์นั้นๆ สามารถแสดงได้ ดังรูปที่

4.13

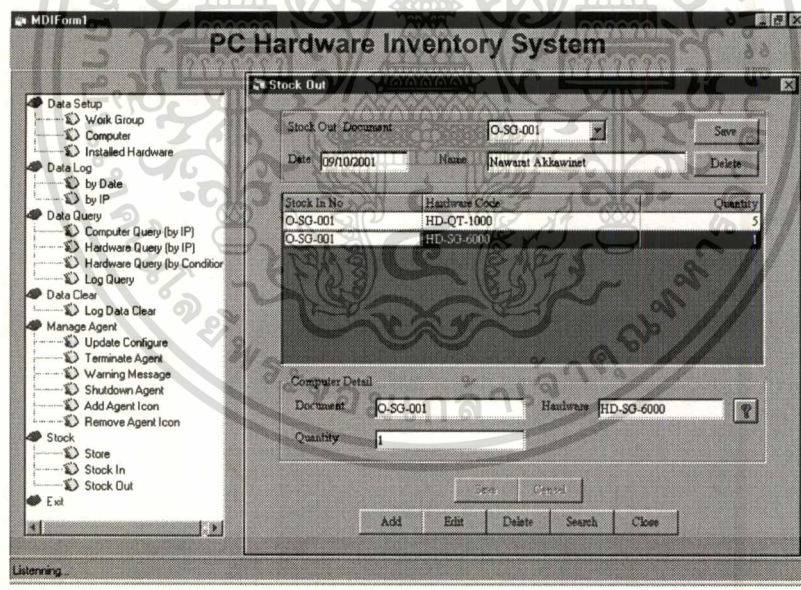


รูปที่ 4.13 การจัดการข้อมูล Store

- ส่วน Stock In เป็นส่วนที่เก็บข้อมูลการนำของเข้ามา ซึ่งต้องเพิ่มค่าจำนวนของอุปกรณ์นั้นๆ ของข้อมูลใน Store โดยอัตโนมัติ ซึ่งสามารถดูข้อมูลได้ ดังรูปที่ 4.14
- ส่วน Stock Out เป็นส่วนที่เก็บข้อมูลการนำของเข้ามา ซึ่งต้องลดค่าจำนวนของอุปกรณ์นั้นๆ ของข้อมูลใน Store โดยอัตโนมัติ ซึ่งสามารถดูข้อมูลได้ ดังรูปที่ 4.15



รูปที่ 4.14 การจัดการข้อมูล Stock In



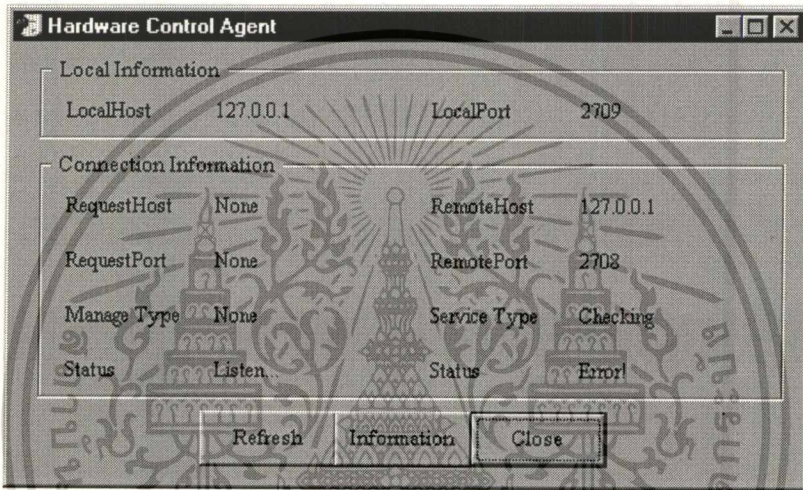
รูปที่ 4.15 การจัดการข้อมูล Stock Out

### 4.2 การพัฒนา Agent

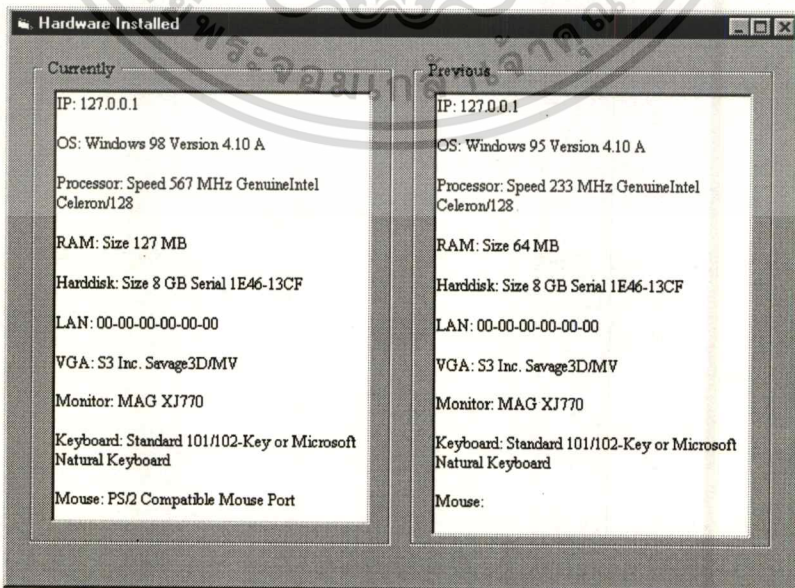
ในการพัฒนา Agent นั้น ได้พัฒนาเป็น Service ที่ Run ฝั่งตัวไ้ ณ Client โดยจะแสดง Icon ของ Agent ไว้ใน System Tray โดยลักษณะการทำงานเริ่มต้น จะตรวจสอบข้อมูลในเครื่อง ของตัวเองที่เป็นอยู่จริง แล้วเปรียบเทียบกับข้อมูลที่เคยถูกบันทึกไว้ ถ้าข้อมูลตรงกัน จะ เอกสารนี้เป็นเอกสารที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Terminate ตัวเองไปโดยอัตโนมัติ แต่ถ้าพบว่าไม่ตรงกัน จะติดต่อไปยัง Manager เพื่อบันทึกลงใน Log ว่า Client ตัวนี้ มีการเปลี่ยนแปลงอุปกรณ์บางอย่าง หลังจากนั้น จะ Stand by รอการ Connect จาก Manager และเมื่อได้รับ Request จาก Manager แล้ว ทาง Agent จะทำการตรวจสอบว่าเป็น Manage Code แบบใด ก็จะไปเรียก Module นั้นมาทำงาน การทำงานของ Agent สามารถแสดงได้ ดังรูป 4.16 และในตัว Agent เอง สามารถที่จะแสดงรายการอุปกรณ์คอมพิวเตอร์ของตัวเอง ณ ขณะนั้น และแสดงรายการอุปกรณ์ที่เคยถูกบันทึกไว้ ซึ่งแสดงได้ดังรูป 4.17



รูปที่ 4.16 การทำงานของ Agent



รูปที่ 4.17 รายการอุปกรณ์คอมพิวเตอร์ใน Agent

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการอบรมเพื่อความรู้เท่านั้น ไม่ควรนำเอกสารนี้ไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### สรุปผลการทดสอบระบบ

จากโครงการที่ได้ทำการพัฒนาระบบควบคุมอุปกรณ์คอมพิวเตอร์ สืบเนื่องมาจากเป็นระบบงานที่มีความสำคัญกับองค์กรในการที่จะควบคุมอุปกรณ์คอมพิวเตอร์ในองค์กร โดยอ่านและบันทึกค่าอุปกรณ์ที่ติดตั้งที่คอมพิวเตอร์แต่ละเครื่อง ผ่านระบบ Network ซึ่งเป็นข้อมูลส่วนใหญ่ขององค์กร ยังรวมไปถึงการรวบรวมอุปกรณ์คอมพิวเตอร์อีกส่วนหนึ่งที่อยู่ใน Store ด้วย ซึ่งในปัจจุบันมี Software ที่สามารถอ่านค่าอุปกรณ์ที่ติดตั้งในคอมพิวเตอร์จำหน่ายอยู่จำนวนมาก แต่พบว่าไม่ได้รองรับการทำงานในแบบที่ต้องการ เพียงแต่เป็นการอ่านค่าอุปกรณ์ในเครื่องเท่านั้น จากจุดนี้ ทำให้เกิดแนวความคิดในการพัฒนาระบบนี้ขึ้นมา

#### 5.1 สรุปผล

ระบบที่พัฒนาแล้วสามารถตอบสนองงานได้ดังนี้

1. ส่วนทะเบียนคอมพิวเตอร์ และอุปกรณ์ที่ติดตั้ง ซึ่งจัดเก็บข้อมูลในฐานข้อมูลรวม ซึ่งเป็นข้อมูลส่วนใหญ่ขององค์กร
2. ส่วนตรวจสอบและบันทึกให้ผู้ดูแลระบบทราบได้ เมื่อมีการเปลี่ยนแปลงอุปกรณ์บางชนิด
3. ส่วนจัดการกับ Client ที่พบว่ามี การติดตั้งอุปกรณ์ที่ไม่ถูกต้องกับข้อมูลที่บันทึกไว้ โดยอาจจะแจ้งเตือน โดยส่งเป็น Warning Message หรือการ Shutdown เครื่องไม่ให้ผู้ใช้ได้ใช้งาน
4. ส่วน Store ซึ่งเป็นส่วนหนึ่งที่เก็บอุปกรณ์คอมพิวเตอร์เช่นกัน เพื่อใช้เป็นการควบคุมการทำ Spare part

#### 5.2 ข้อควรปรับปรุงเพิ่มเติม

1. ระบบยังไม่สามารถอ่านค่าอุปกรณ์คอมพิวเตอร์ได้ถึง Serial Number ของอุปกรณ์นั้นๆ รวมถึงยังไม่สามารถอ่านค่าอุปกรณ์ที่นอกเหนือไปจากที่กำหนด
2. ระบบเก็บข้อมูลอุปกรณ์คอมพิวเตอร์ได้ตามที่กำหนดเท่านั้น ยังไม่ยืดหยุ่นต่อการเก็บอุปกรณ์ที่มากกว่าเดิม รวมถึงยังไม่ยืดหยุ่นต่อการเก็บรายละเอียดของข้อมูลที่มากกว่าที่กำหนด
3. ระบบยังไม่สามารถแจ้งเตือนให้ผู้ดูแลระบบทราบโดยตรง เมื่อพบว่า Client บางตัวได้มีการเปลี่ยนแปลงอุปกรณ์บางอย่าง ระบบแค่บันทึกข้อมูลไว้เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ระบบต้องใช้กับองค์กรที่กำหนด IP Address ของเครื่องไว้แน่นอนเท่านั้น ไม่ยึดหยุ่นกับองค์กรที่ใช้ DHCP

### 5.3 ปัญหาในการพัฒนาระบบ

1. ในการอ่านค่าอุปกรณ์ที่ติดตั้ง ซึ่งน่าจะอ่านได้จาก Window Registry แต่อุปกรณ์บางชนิดไม่สามารถทำได้ ต้องใช้ฟังก์ชันวินโดว API ของ Visual Basic เข้าช่วย และอุปกรณ์บางชนิดไม่สามารถอ่านได้ทั้งจาก Window Registry รวมทั้ง Function API ต้องอาศัยการเขียนโปรแกรมภาษา C หรือ Assembly ซึ่งผู้เขียนยังไม่มีความรู้เพียงพอ
2. ในการอ่านค่าอุปกรณ์บางตัว จะไม่ตรงกับอุปกรณ์ที่ต่ออยู่จริง เพราะเป็นการอ่านจาก Driver ที่ Installed เท่านั้น ซึ่ง Driver นั้นสามารถ compatible กับอุปกรณ์ชนิดนั้น ทำให้เราไม่สามารถทราบข้อมูลอุปกรณ์ที่ใช้งานจริง
3. ผู้เขียนได้ทำการทดสอบโดยการใช้หมายเลข IP Address เป็น Loop Back Address (127.0.0.1) เป็นส่วนใหญ่ ยังขาดการทดสอบภายในองค์กรจริงๆ

## บรรณานุกรม

พร้อมเลิศ หล่อวิจิตร และ ศุภพงษ์ อารีประเสริฐกุล. 2543. เจาะลึกเคล็ดลับ Windows.

กรุงเทพฯ : โปรวิชั่น

สมทธิศักดิ์ พงศ์ธนาพานิช. 2542 . การเขียนโปรแกรมด้วย Visual Basic 6.0 ระดับสูง. กรุงเทพฯ :

โรงพิมพ์จุฬาลงกรณ์มหาวิทยาลัย.

Craig John Clark and Webb Jeff. 1998. **Microsoft Visual Basic 6.0 Developer Workshop.**

Microsoft Press.

Roman Steven. Jan 2000. **Win32 API Programming With Visual Basic.** O'reilly.

