

การพัฒนาระบบเฝ้าตรวจสอบซอฟต์แวร์
Development of Software Monitoring System

โดย

นายอานวย อูทัยรัมย์

รหัส 42067071

อาจารย์ที่ปรึกษา

อาจารย์ อัครินทร์ คุณกิตติ



H001729

วัน เดือน ปี..... 0 8 2550

เลขทะเบียน..... 0 8 1 7 2 9

เลขเรียกหนังสือ..... ๑๗ ๒ ๒๘๕๓ ๕๕๓

"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
ภาคเรียนที่ 2 ปีการศึกษา 2543
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวข้อ	การพัฒนาระบบเฝ้าตรวจซอฟต์แวร์
นักศึกษา	นายอำนาจ อุทัยรัมย์
อาจารย์ที่ปรึกษา	อาจารย์อักรินทร์ คุณกิตติ
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2543

บทคัดย่อ

รายงานฉบับนี้เป็นการนำเสนอการพัฒนา Software ที่ใช้สำหรับเฝ้าตรวจว่าในแต่ละเครื่องของ Computer ที่ต่ออยู่ในเครือข่ายได้ติดตั้ง Software ประเภทใดบ้าง เพื่อเฝ้าตรวจการควบคุมและตรวจสอบกับลิขสิทธิ์ที่ทางบริษัทหรือองค์กรมีอยู่ทำให้การควบคุมการติดตั้ง Software ขององค์กรมีประสิทธิภาพมากขึ้น โดยในระบบประกอบด้วย Software ที่หน้าที่เป็น Manager มีหน้าที่หลักในการเก็บรวบรวม Software ต่างๆที่ได้ทำการติดตั้งไว้ ณ เครื่อง Computer ลูกข่าย (Client) พร้อมทั้งนำมาวิเคราะห์เปรียบเทียบกับรายการ Software ที่ทางองค์กรมีอยู่ รวมทั้งมีหน้าที่ในการจัดการกับเครื่อง Computer ลูกข่ายที่ได้ทำการติดตั้ง Software ที่ละเมิดลิขสิทธิ์เช่น ระบบการแจ้งเตือน (Warning Message) ระบบการยกเลิกการติดตั้งจากระยะไกล (Remote UnInstallation) รวมทั้งการสั่ง Shutdown เครื่อง Computer ที่ยังฝ่าฝืนติดตั้ง Software ที่ละเมิดลิขสิทธิ์ นอกจากนี้ในระบบยังประกอบไปด้วย Software ที่ทำหน้าที่เป็นตัว Agent ที่ได้ติดตั้งเป็น Service ที่ Run ฝังตัวไว้ ณ เครื่อง Computer ลูกข่าย (Client) ซึ่งมีหน้าที่หลักในการให้บริการต่างๆตามที่ทาง Manager ได้ทำการร้องขอ (Request) ซึ่งส่งมาในลักษณะของ Service Code โดยบริการที่สำคัญตัว Agent ให้บริการได้แก่ การตรวจหา Software ที่ได้ทำการติดตั้งไว้ใน Computer ลูกข่าย (Client) โดยตรวจสอบจาก Windows Registry ของระบบ การทำการแสดงข้อความแจ้งเตือน (Warning Message) รวมทั้งการ Shutdown เครื่อง Computer เมื่อได้รับการร้องขอจาก Manager

ในการติดต่อสื่อสารระหว่าง Manager และ Agent นั้นได้ใช้ Protocol TCP/IP ซึ่งอาศัยการสร้าง Socket ในการสร้างการเชื่อมต่อ โดยทาง Agent ซึ่งทำหน้าที่เป็นผู้ให้บริการนั้นได้ใช้หมายเลข Port ในการติดต่อสื่อสารเป็น 3007 ในการให้บริการ

Title Development of Software Monitoring System
Student Mr.Amnuay Uthairungsri
Advisor Akharin Khunkitti
Level of Study Master of Science in Information Technology
Major Information Science
Academic Year 2000

ABSTRACT

Development Of Software Monitoring System is the development of software for monitor software are installed in each computer .So the system can support software installation collection and add more efficient in software installation control in organization. The system consist of manager which has the main function in software collection form client computer and compare with software list in organization. Manager also manage the client that install stall illegal software , such as send warning message ,remote uninstall software , including shutdown client computer. The system also consist of the agent that acting as embed application at any client computer. The main function of agent is to service manager . The agent has many service for manager such as query installed software from client , display warning message , force system to shutdown .

Manager and agent establish connection by TCP/IP network so it use socket to communicate by agent acting as server and has service port no 3007 and stand by for connection request.

กิตติกรรมประกาศ

การที่โครงการนี้ประสบความสำเร็จไปด้วยดีได้นั้น ผู้เขียนต้องขอขอบคุณบุคคลต่าง ๆ ที่มี
ส่วนในการสนับสนุนให้การจัดทำโครงการดังกล่าวสำเร็จตามวัตถุประสงค์ที่ต้องการ ได้แก่

1. คุณพ่อ คุณแม่ ผู้ซึ่งเล็งเห็นความสำคัญของการศึกษาของลูกๆทุกคน รวมทั้งพี่น้องที่คอย
เป็นกำลังใจที่สำคัญตลอดมา
2. อาจารย์อัครินทร์ คุณกิตติ อาจารย์ที่ปรึกษาโครงการ ผู้ให้คำปรึกษาและชี้แนวทางในการ
จัดทำโครงการ รวมทั้งให้ความช่วยเหลือตลอดเวลาในการศึกษา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
บทที่	
1. บทนำ.....	1
2. ความรู้พื้นฐานในการพัฒนาระบบเฝ้าตรวจซอฟต์แวร์.....	3
2.1 Windows Registry.....	3
2.2 การติดต่อสื่อสารด้วย TCP/IP บน Visual Basic.....	7
2.3 พื้นฐานเกี่ยวกับฟังก์ชัน API.....	10
3. การออกแบบโครงสร้างระบบ.....	18
3.1 การออกแบบการทำงานของ Manager.....	21
3.2 การออกแบบการทำงานของ Agent.....	26
3.3 การออกแบบการติดต่อสื่อสารระหว่าง Manager และ Agent.....	28
4. การพัฒนาระบบเฝ้าตรวจซอฟต์แวร์.....	29
4.1 การพัฒนาส่วน Manager.....	29
4.2 การพัฒนาส่วน Agent.....	34
5. สรุปผลการทดลอง.....	37
5.1 สรุปผล.....	37
5.2 ข้อควรปรับปรุงเพิ่มเติม.....	37
5.3 ปัญหาในการพัฒนาระบบ.....	38
6. เอกสารอ้างอิง.....	39
7. ภาคผนวก.....	40

บทที่ 1

บทนำ

ในปัจจุบัน บริษัท องค์กร ที่มี Computer เป็นจำนวนมาก มักประสบกับปัญหาการควบคุมการใช้ Software ของ End User ซึ่งมักจะนำ Software จากภายนอกองค์กรเข้ามาทำการติดตั้งเพื่อใช้งานเพื่อความบันเทิง หรือจุดประสงค์อื่น ซึ่งสร้างความยุ่งยากให้กับผู้ดูแลระบบ Computer ในการตรวจสอบ เนื่องจาก Software ดังกล่าวมักจะเป็น Software ที่ไม่ถูกต้องตามลิขสิทธิ์ที่ทางองค์กรมีอยู่ ส่งผลให้ทางองค์กรต้องจ่ายค่าปรับในกรณีที่มีการเข้าตรวจจากหน่วยงานภายนอก รวมทั้งส่งผลให้องค์กรเสื่อมเสียชื่อเสียง ดังนั้นจึงมีแนวคิดในการพัฒนาระบบ Software Monitoring ที่ทำหน้าที่ตรวจสอบ Software ที่ทำการติดตั้งไว้ในแต่ละ Client ที่ต่อกันผ่านเครือข่าย

1.1 วัตถุประสงค์ของการพัฒนาโครงการ

เพื่อเป็นการพัฒนาเครื่องมือที่ช่วยในการเฝ้าตรวจสอบการติดตั้ง Software ที่ได้ทำการติดตั้งไว้ ณ เครื่อง Computer ลูกข่าย (Client) เพื่อทำการวิเคราะห์เปรียบเทียบกับรายการ Software ที่ทางองค์กรมีลิขสิทธิ์อยู่ รวมทั้งมีวิธีการจัดการกับเครื่อง Computer ที่ทำการติดตั้ง Software ที่ละเมิดลิขสิทธิ์ ทำให้การควบคุมและตรวจสอบการใช้ Software ขององค์กรมีประสิทธิภาพมากยิ่งขึ้น

1.2 แนวทางการพัฒนา

1. พัฒนา Agent ฝังไว้ที่ Client เพื่อทำหน้าที่ Monitoring Software ที่ได้ทำการ Install ไว้ที่เครื่อง Client แต่ละเครื่องในระบบเครือข่าย โดยการตรวจสอบใน Registry ของระบบ แล้วรายงานผลมายังศูนย์กลาง
2. พัฒนา Manager ซึ่งทำหน้าที่รวบรวม Software ที่ถูกติดตั้งในแต่ละ Client ซึ่งถูกส่งมาจาก Agent ที่ประจำอยู่ในแต่ละ Client พร้อมทั้งเปรียบเทียบกับ Software List ที่ถูกต้องตามลิขสิทธิ์ที่ทางบริษัทมีอยู่ รวมทั้งมีการแจ้งเตือนในกรณีที่เจ้าของเครื่องมี software ที่ไม่มี ลิขสิทธิ์
3. Agent และ Manager ติดต่อกันผ่านเครือข่าย TCP/IP

1.3 เครื่องมือที่ใช้ในการพัฒนา

เครื่องมือที่ใช้ในการพัฒนาระบบ Software Monitoring System นั้นประกอบไปด้วย

1. Microsoft Visual Basic 6.0 ใช้สำหรับพัฒนาตัว Agent ที่ทำงานฝั่งตัวอยู่ในเครื่อง Client และใช้สำหรับพัฒนาตัว Manager โดยใช้ Microsoft Winsock Control ในการติดต่อสื่อสารระหว่าง Manager และ Agent
2. เครื่อง Computer PC ที่เชื่อมต่อกันเป็นเครือข่าย พร้อมทั้งได้ทำการติดตั้ง Microsoft Windows 95/98 ไว้แล้ว

1.4 ขอบเขตการพัฒนา

ส่วนเงื่อนไขและขอบเขตของการพัฒนานั้น ได้กำหนดขอบเขตของการพัฒนาไว้เป็นดังนี้

1. ระบบ Software Monitoring มีขอบเขตการใช้งานบนเครื่อง PC Computer ที่เชื่อมต่อกันเป็นเครือข่าย และได้ทำการติดตั้ง Microsoft windows 95/98 หรือ สูงกว่าและมีการใช้ Registry เป็นที่เก็บข้อมูลระบบ
2. การเชื่อมต่อเครือข่ายระหว่าง Agent และ Manager ใช้เครือข่าย TCP/IP เท่านั้น เนื่องจาก การติดต่อระหว่าง Agent และ Manager ได้ใช้หลักการของ Socket ในการติดต่อสื่อสาร

บทที่ 2

ความรู้พื้นฐานในการพัฒนาระบบเฝ้าตรวจสอบซอฟต์แวร์

เนื่องจากในขั้นของการพัฒนานั้น จำเป็นจะต้องเกี่ยวข้องกับการ Access ข้อมูลใน Windows Registry ผ่านทาง Function ของ windows API โดยใช้ Microsoft Visual Basic ในการพัฒนา รวมทั้งในระบบยังเกี่ยวข้องเครือข่ายด้วย ดังนั้นจึงจำเป็นต้องศึกษาความรู้พื้นฐานที่เกี่ยวข้องของก่อนเพื่อประโยชน์ในการพัฒนาระบบให้มีประสิทธิภาพมากขึ้น

2.1 Windows Registry

Registry คือฐานข้อมูลกลางของระบบปฏิบัติการ Windows ที่ใช้เก็บข้อมูลการกำหนดค่าต่างๆ ทั้งในส่วนของฮาร์ดแวร์ ซอฟต์แวร์, การปรับหน้าจอ รวมถึง Configuration ทั้งหมดของ Windows ซึ่งจำเป็นต่อการบริหารการทำงานของคอมพิวเตอร์ทั้งเครื่อง

ข้อมูลใน Registry อาจแบ่งได้เป็น 4 ส่วนใหญ่ๆ คือข้อมูลเกี่ยวกับฮาร์ดแวร์ที่มีอยู่ในเครื่องคอมพิวเตอร์ ข้อมูลการทำงานของซอฟต์แวร์แต่ละตัวซึ่งรวมถึงซอฟต์แวร์แบบ 16 บิตด้วย ข้อมูลการกำหนดค่าและการทำงานต่างๆ ของ Windows และสุดท้ายข้อมูลการกำหนดของผู้ใช้แต่ละคน 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 เช่นข้อมูลของการตั้งค่าและอุปกรณ์ต่อพ่วง เป็นต้น และเพื่อให้สามารถแก้ไขได้อย่างรวดเร็ว ในระหว่างการทำงานนั้นข้อมูลดังกล่าวจึงถูกโหลดเข้ามาอยู่ในหน่วยความจำหลัก โดย Windows จะดำเนินการตรวจสอบอุปกรณ์ทุกชิ้นที่มีอยู่ในระบบและปรับปรุงค่าต่างๆ ทุกครั้งที่เปิดเครื่องใหม่ เพื่อให้ใช้งานอุปกรณ์เหล่านี้ได้ตามคุณสมบัติของระบบปฏิบัติการแบบ Plug and Play (PnP) แต่ด้วยความจำเป็น เพื่อให้โปรแกรมเก่าๆยังคงทำงานได้ (Backward Compatibility) ทำให้ยังคงมีไฟล์ WIN.INI และ SYSTEM.INI อยู่เพื่อให้ซอฟต์แวร์รุ่นเก่าสามารถกำหนดค่าบางอย่างให้กับ Windows ได้เช่นเดิม ซึ่งหากเปิดดูก็จะพบว่ามี เนื้อหาน้อยลงมาก รวมทั้งค่าไฟล์ AUTOEXEC.BAT และ CONFIG.SYS ก็ยังมีความจำเป็นต่อการกำหนดสภาพแวดล้อมบางอย่างของ DOS ใน Windows ด้วย

โครงสร้างโดยทั่วไปภายใน Registry ประกอบด้วยส่วนหลักๆที่เรียกว่า คีย์ (key) ซึ่งมีความสัมพันธ์ต่อกันในลักษณะของต้นไม้แบบลำดับชั้น (Hierarchical Tree) เริ่มจากคีย์ที่เป็น Root การค้นหาว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(root key) แล้วแตกลงไปในคีย์ย่อย (subkey หรือ Hive) ต่างๆ ส่วนที่ใช้เก็บข้อมูลจริงนั้นเป็น Entry ภายในคีย์เรียกว่า Value Entry หรือสั้นๆว่า Entry โดยแต่ละคีย์ใน Registry สามารถจัดเก็บ Entry ไว้ได้ไม่จำกัดจำนวน ไม่ว่าจะ เป็นคีย์ในระดับใดก็ตาม ซึ่งข้อมูลที่เก็บนั้นแบ่งแยกออกได้ เป็น 3 ประเภทคือ

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

การที่แบ่งแยกเป็น System.Dat กับ User.Dat นี้ เพื่อให้เกิดความชัดเจนในการจัดการมากยิ่งขึ้น โดย System.Dat จะเก็บข้อมูลเฉพาะส่วนที่เกี่ยวข้องกับคอมพิวเตอร์เครื่องนั้นๆ ทั้ง ซอฟต์แวร์และฮาร์ดแวร์ ส่วนไฟล์ User.Dat ก็จะมีเก็บข้อมูลส่วนที่กำหนดแตกต่างกันสำหรับผู้ใช้แต่ละคน 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 การปรับแก้ก็สามารถตั้งจากเครื่องที่กำลังใช้อยู่ได้ด้วย 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 จะมีผลต่อค่าในคีย์ย่อยของผู้ใช้นั้นๆ ภายในคีย์ HKEY_USER ด้วย เนื่องจาก HKEY_CURRENT_USER เป็นเพียงจุดที่ Registry ยกขึ้นมาเพื่อความสะดวกในการเข้าถึงค่า Configuration ต่างๆของผู้ใช้คนปัจจุบันเท่านั้น แท้ที่จริงก็ชี้ไปที่ข้อมูลชุดเดียวกันนั่นเอง คีย์ย่อยที่อยู่ภายใต้คีย์ทั้งสองมีดังต่อไปนี้

Control Panel\Desktop\CursorBlinkRate กำหนดอัตราการกระพริบของ caret สำหรับ ป้อนตัวอักษร หน่วยเป็นมิลลิวินาที (1/1000วินาที) ค่าของ Entry นี้สามารถกำหนดได้จากแอปพลิเคชัน Keyboard ใน Control Panel

Control Panel\Desktop\MenuShowDelay ค่าของ Entry นี้กำหนดช่วงเวลาที่ต้องรอไว้ก่อน แสดงเมนูย่อยออกมา (เช่นเมื่อเลื่อนเมาส์ไปชี้ที่เมนู Programs ของเมนูปุ่ม Start แล้วจะปรากฏเมนูย่อยของเมนู Programs ออกมา)มีหน่วยเป็นวินาที

Control Panel\Desktop\DoubleClickHeight และ DoubleClickWidth ผู้ใช้มือใหม่ที่ไม่คุ้นเคยกับเมาส์อาจเจอเหตุการณ์แบบนี้ก็ได้ คือขณะที่ ต้องการ Double Click แต่ด้วยความไม่ถนัดมือ การกดปุ่มแต่ละครั้งกลับทำให้เมาส์ขยับไปด้วยในระหว่างที่ครั้งแรกกับครั้งที่สอง ผลที่ได้จึงเป็นการลากเมาส์ในระยะทางสั้นๆแทนการดับ Double Click ที่ ต้องการ Entry ทั้งสองนี้จะช่วยให้ Windows ตัดสินใจแทน ว่าเหตุการณ์แบบนี้เป็นการ Double Click หรือเปล่า โดย จะต้องกำหนดระยะเพื่อการขยับของเมาส์ในระหว่างการขยับของเมาส์ในระหว่างการดับเบิ้ลคลิกไว้เป็นตัวเลขแบบค่าสตริงในหน่วยพิกเซลให้มากขึ้น หากการ Double Click คลิกของ ยังไม่เพียงพอ

Control Panel\Desktop\Desktop\ คีย์นี้เก็บ Entry ที่นอกเหนือจากคีย์ย่อย Colors ในคีย์ Appearance โดยเป็นค่าเกี่ยวกับขนาดของวินโดว์,ขนาดของตัวอักษร,ระยะห่างระหว่างไอคอน และอีกหลายๆอย่างที่ไม่ใช่ค่าเกี่ยวกับสี อีก Entry หนึ่งที่ Windows ไม่ได้สร้างไว้ให้หลังการติดตั้งเสร็จใหม่ๆก็คือ MinAnimate ซึ่งใช้กำหนดวิธีการแสดงการ Minimize และ Maximize ของวินโดว์ว่าจะให้ Animation แบบค่อยๆย่อ ค่อยๆขยายหรือไม่ ถ้าค่านี้ไม่มีหรือมีค่าเป็น 1 จะแสดง Animationในการย่อ-ขยาย แต่ถ้าเป็น 0 จะไม่แสดง

Control Panel\Desktop\Desktop\Keyboard คีย์ นี้เก็บ Entry ที่เกี่ยวข้องกับคีย์บอร์ดไว้ 2 Entry คือ KeyboardDelay กำหนดช่วงเวลาหลังจากการกดปุ่มบนคีย์บอร์ดค้างไว้ก่อนถือเป็นการพิมพ์ตัวอักษรซ้ำและ KeyboardSpeed กำหนดอัตราเร็วในการพิมพ์ซ้ำตัวอักษรเมื่อกดปุ่มค้างไว้ ค่าสูงสุดที่กำหนดได้คือ 31

Control Panel\Desktop\Keyboard layout\toggle คีย์นี้เก็บ Entry ที่เกี่ยวข้องกับสลับภาษาของคีย์บอร์ด ปกติสามารถแก้ไขได้จาก Tab Language ใน Sheet Keyboard Properties โดยค่า Default ของคีย์จะหมายถึงปุ่มที่ใช้สลับภาษา ค่าที่กำหนดได้คือ 1,2และ 3 ซึ่งสอดคล้องกับตัวเลือก Left Alt+Shift, Ctrl+Shift และ None ใน Tab Language ตามลำดับ

Control Panel\Desktop\RemoteAccess\Profile คีย์นี้จะมีคีย์ย่อยที่เก็บคอนฟิกูเรชันของการติดต่อทางโมเด็มแต่ละรายการที่ สร้างขึ้นมาในโพลเดอร์ Dial-Up networking

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2. HKEY_LOCAL_MACHINE

เป็นคีย์ที่เก็บ Configuration ของเครื่องทั้งในส่วนฮาร์ดแวร์และซอฟต์แวร์ ไม่ขึ้นกับผู้ใช้แต่ละคน มีคีย์ย่อยที่น่าสนใจดังต่อไปนี้

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 ที่เขียนไว้บนนั้นสื่อถึงพวก Virtual โพลเดอร์ต่างๆ เช่น Desktop , My Computer เป็นต้นภายในคีย์ของ Virtual โพลเดอร์เหล่านี้จะมีคีย์ย่อยชื่อ NameSpace

คีย์ NameSpace คือตำแหน่งที่ จะ ได้พบกับคีย์ย่อยที่มีชื่อเป็นหมายเลข CLSID (ปิดหัวท้ายด้วยเครื่องหมายปีกกา) ของ Object ใน Virtual โพลเดอร์นั้นอีกทีหนึ่ง (เมื่อพิจารณาจากโครงสร้าง Name Space)จากรูปข้อมูลใน Registry บอกให้เราทราบว่าโพลเดอร์ Dial-Up Networking เป็น Object ที่อยู่ภายใต้โพลเดอร์ My Computer นั่นเอง

Microsoft\Windows\CurrentVersion\Run... คีย์ Run ใช้เก็บชื่อ Path เต็มของไฟล์โปรแกรมซึ่งจะถูกรันทุกครั้งที่ Log On เข้าสู่ Windows คีย์อื่นๆในระดับเดียวกันได้แก่ RunOnce, Runservices และ RunServicesOnce คีย์ RunOnce ถูกใช้ในการติดตั้งซอฟต์แวร์ที่ต้องมีการ Restart Windows แล้วจึงติดตั้งต่อจนเสร็จ วิธีรันโปรแกรมติดตั้งต่อหลังจากบูตเครื่องใหม่ทำได้โดยบันทึก Path เต็มของโปรแกรมที่จะรันไว้ในคีย์นี้ เมื่อรันเรียบร้อยแล้ว

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

2.1.3 HKEY_DYN_DATA

เป็นคีย์ที่ข้อมูลถูกเก็บไว้ในหน่วยความจำอยู่ตลอดเวลา ข้อมูลส่วนใหญ่จะถูกสร้างขึ้นในขณะที่บูตเครื่องและสามารถถูกแก้ไขได้ตลอดเวลาที่ใช้งานคอมพิวเตอร์ มีคีย์ย่อยที่น่าสนใจดังนี้

Config Manager เป็นข้อมูลเกี่ยวกับคอนฟิกูเรชันของฮาร์ดแวร์ที่ติดตั้งและกำลังใช้งานอยู่ ข้อมูลภายในคีย์นี้ถูกสร้างขึ้นเมื่อบูตเครื่องและสามารถเปลี่ยนแปลงแก้ไขได้ตลอดเวลา คีย์นี้เองที่ทำให้สามารถติดตั้งอุปกรณ์ฮาร์ดแวร์ต่างๆ ได้อย่างสะดวกโดยใช้ สมบัติ Plug and Play ของอุปกรณ์นั้น

PerfStats เก็บข้อมูลประสิทธิภาพการทำงานในแต่ละส่วนของคอมพิวเตอร์ ซึ่งจะถูกนำไปแสดงโดยโปรแกรม System Monitor

2.2 การติดต่อสื่อสารด้วย TCP/IP บน Visual Basic

การทำงานเกี่ยวกับเครือข่ายคอมพิวเตอร์นั้นเป็นเรื่องค่อนข้างซับซ้อนจนทำให้มีการสร้างข้อกำหนดต่างๆ ขึ้นมาเพื่อให้วิธีการสื่อสารระหว่างกันเป็นไปได้โดยราบรื่นแล้วก็ทำให้เกิดเป็น โพรโตคอล(Protocol) มากมายหลายชนิดมาใช้งานร่วมกับเครือข่าย ซึ่งมีการแบ่ง Protocol ต่างๆ ออกเป็นระดับชั้น เพื่อให้ง่ายต่อการใช้งาน และง่ายต่อการทำความเข้าใจ

นานวันเข้าโพรโตคอลก็มีเพิ่มมากขึ้นจนกลายเป็นความยุ่งยาก ต่างบริษัทก็ต่างมีผลิตภัณฑ์ที่สนับสนุนโพรโตคอลต่างๆ ไม่เหมือนกัน ทำให้การใช้งานร่วมกันทำได้ยากขึ้น ทำให้ต้องหาวิธีใช้งานโพรโตคอลต่างๆ ร่วมกันโดยใช้มาตรฐานอันเดียวที่ทุกคนยอมรับ ซึ่ง Winsock ก็เป็นหนึ่งในมาตรฐานเหล่านั้น

ระบบปฏิบัติการยูนิกซ์จะมีความสามารถอย่างหนึ่งที่เรียกว่า Socket ซึ่งก็คือ การที่โปรแกรมต่างๆ สามารถสื่อสารข้อมูลระหว่างกันได้ โดยไม่จำเป็นต้องรออยู่บนคอมพิวเตอร์เครื่องเดียวกัน

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

2.2.1 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 ซึ่งออกแบบมาเพื่อทำให้การเขียนโปรแกรมกับอินเทอร์เน็ตเป็นเรื่องง่าย อีกทั้ง Winsock ยังถูกกำหนดให้เป็น API ที่มีลักษณะเปิดกว้าง และสามารถ port ไปยัง Socket ของยูนิกซ์ได้อย่างไม่ยากนัก

2.2.2 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 นั้นคือมีการทำงานในลักษณะ Connectionless เปรียบเหมือนการกระจายเสียงวิทยุ UDP จะเหมาะกับการส่งข้อมูลที่ไม่สำคัญนัก เช่นส่งข้อมูลแสดงสถานะการทำงานซึ่งก็มักจะเป็นข้อมูลขนาดเล็กซึ่งทำให้ UDP กินทรัพยากรของระบบน้อยกว่า TCP

Property ที่สำคัญของ Winsock Control

Protocol เป็นการเลือกโปรโตคอลสำหรับการทำงาน

LocalPort เป็นการกำหนดหมายเลขพอร์ตของคอมพิวเตอร์ที่จะใช้งานกับ Winsock

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ในการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RemoteHost	เป็นการกำหนดชื่อของคอมพิวเตอร์ที่เราจะติดต่อด้วย อาจจะเป็น
IP Address	หรือเป็นชื่อคอมพิวเตอร์ที่เป็นชื่อที่ง่ายต่อการจดจำ
RemotePort	เป็นการกำหนดหมายเลขพอร์ตของคอมพิวเตอร์ที่เราจะติดต่อด้วย
ByteReceive	เป็นจำนวนข้อมูลที่รับเข้ามาเก็บใน 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 เป็นเหตุการณ์ที่เกิดขึ้นเมื่อ คอมพิวเตอร์เครื่องอื่น มีการ Request เข้ามา ซึ่งจะมีการกำหนด ID ให้กับแต่ละค่าของ Request ที่เข้ามา

Data Arrival เป็นเหตุการณ์ที่เกิดขึ้นเมื่อมีข้อมูลชุดใหม่เข้ามาเก็บใน Buffer ซึ่งเราสามารถตรวจสอบขนาดข้อมูลได้จาก Property ByteReceive

SendProgress เป็นเหตุการณ์ที่เกิดขึ้นที่มีการส่งข้อมูลระหว่างกัน ซึ่งจะมีพารามิเตอร์แสดงจำนวนข้อมูลที่ส่งมาแล้ว และข้อมูลที่ยังคงเหลืออยู่

SendComplete เป็นเหตุการณ์ที่เกิดขึ้นเมื่อการส่งข้อมูลเสร็จสิ้นสมบูรณ์

Error เป็นเหตุการณ์ที่เกิดขึ้นเมื่อมีความผิดพลาดเกิดขึ้น ซึ่งจะแสดงหมายเลขของความผิดพลาด, คำอธิบาย และรายละเอียดอื่นๆ สำหรับการจัดการกับข้อผิดพลาดที่เกิดขึ้น

2.3 พื้นฐานเกี่ยวกับฟังก์ชันวินโดวส์ API – 32 บิต

ฟังก์ชันวินโดวส์ API ของวินโดวส์ หมายถึง ฟังก์ชันงานทั้งหมดที่อยู่ในไฟล์ไลบรารี .dll ของวินโดวส์ เช่น User32.dll Gdi32.dll หรือ Kernel32.dll เป็นต้น ซึ่งไฟล์ไลบรารี .dll เหล่านี้จะ เป็นแกนหลักของระบบปฏิบัติการวินโดวส์ เพราะการทำงานของวินโดวส์ จะอาศัยความสามารถ ของฟังก์ชันที่ปรากฏในไฟล์ไลบรารี .dll

2.3.1 ไฟล์ไลบรารี .dll ของวินโดวส์ 32 บิต

เป็นไฟล์ที่โดยปกติจะถูกกำหนดให้มีนามสกุล .dll และจะถูกจัดเก็บเอาไว้ใน Directory ย่อย \System ของวินโดวส์ และการทำงานของวินโดวส์ก็ต้องอาศัยฟังก์ชันวินโดวส์ API ที่มีในแต่ละไฟล์ไลบรารี .dll นั้นเอง สำหรับวินโดวส์ 32 บิต (วินโดวส์ 98) สนับสนุน ฟังก์ชันวินโดวส์ API มาตรฐานมากกว่า 1,500 ฟังก์ชัน โดยที่จะแยกจัดเก็บไว้ในไฟล์ไลบรารีที่ถือ ได้ว่าเป็นแกนหลักของวินโดวส์ 32 บิต ดังต่อไปนี้

ไฟล์ .dll	รายละเอียด
Gdi32.dll	ไลบรารีสำหรับการติดต่อสื่อสารกับดีไวซ์กราฟิก (Graphic Device Interface) ซึ่ง ประกอบด้วยฟังก์ชันต่างๆสำหรับงานด้าน อุปกรณ์แสดงผล การวาดกราฟิก ไฟล์ เมตา (.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 และ สำหรับการปรับแต่ง Form สำหรับ Visual Basic 6.0
Version.dll	ไลบรารีสำหรับการสนับสนุนการควบคุม และอ่านหมายเลขเวอร์ชันจากไฟล์ .exe .dll .ocx ฯลฯ ซึ่งจะมีประโยชน์อย่างมากในขณะทำการติดตั้งไฟล์ต่างๆ เนื่องจากวินโดวส์ 32 บิต สนับสนุนการตรวจสอบหมายเลขเวอร์ชันของไฟล์ เพื่อป้องกันความไม่เข้ากันของ Application และคอนโทรล ActiveX หรือไฟล์ ไลบรารีต่างๆ
Dllz32.dll	ไลบรารีสำหรับการสนับสนุนการขยายขนาดของไฟล์ที่ถูกบีบอัด อย่างเช่น ในการสร้างแผ่น setup นั้น ไฟล์ต่างๆจะถูกบีบอัดด้วยโปรแกรม compress.exe ซึ่ง โดยปกติไฟล์ที่ถูกบีบอัดก็จะมีนามสกุลตัวสุดท้ายเป็นเครื่องหมาย underscore () โดยในขณะที่ทำการติดตั้ง ไฟล์เหล่านี้ก็จะถูกขยายขนาดอัตโนมัติโดยวินโดวส์ ซึ่ง จะอาศัยฟังก์ชันวินโดวส์ API จากไฟล์ไลบรารี นั้นเอง

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่ควรนำข้อมูลไปใช้โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

2.3.2 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

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
Int	ข้อมูลชนิดเลขจำนวนเต็ม signed ขนาด 32 บิต

นี่เป็นเอกสารที่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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 ASCIIZ
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 IUnknown 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 โดยจะต้องมีการประกาศให้กับ พารามิเตอร์ ดังนี้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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 จึงต้องอาศัยเทคนิคในการแปลงข้อมูลตัวเลข

3. ข้อมูลชนิดตัวเลขขนาด 32 บิต สำหรับภาษาซีนั้นมีอยู่ด้วยกัน 5 ประเภทดังนี้ BOOL DWORD int UINT และ LONG ซึ่งจะ Compatible กับข้อมูลชนิด long ของ Visual Basic 6.0 โดยจะต้องมีการประกาศให้กับพารามิเตอร์ ดังนี้

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 และในการส่งค่าตัวเลขหรืออ่านค่ากลับจากฟังก์ชัน วินโดวส์ 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

5. ข้อมูลชนิดตัวเลขจำนวนจริงขนาด 32 บิต สำหรับภาษาซีนั้นมีอยู่ด้วยกันเพียงประการเดียว คือ float ซึ่งจะเหมือนกับข้อมูลชนิด single ของ Visual Basic 6.0 ทุกประการ ดังนั้นจึงไม่จำเป็นต้องมีการแปลงข้อมูลที่ได้

6. ข้อมูลชนิดตัวเลขจำนวนจริงขนาด 64 บิต สำหรับภาษาซีนั้นมีอยู่ด้วยกันเพียงประการเดียว คือ double ซึ่งจะเหมือนกับข้อมูลชนิด double ของ Visual Basic 6.0 ทุกประการ ดังนั้นจึงไม่จำเป็นต้องมีการแปลงข้อมูลที่ได้ ซึ่ง สามารถประกาศให้กับพารามิเตอร์

7. ข้อมูลชนิด handle สำหรับวินโดวส์ 32 บิต จะมีขนาดเท่ากับ 32 บิต สำหรับใช้กำหนดหรือ อ้างอิง (reference) ชนิดของ Object ของวินโดวส์ ดังนั้นถ้าหากมีการกำหนดตัวแปรสำหรับ รับหรือส่งข้อมูลชนิด handle นี้ ก็ต้องใช้ข้อมูลชนิด long ของ Visual Basic

8. ข้อมูลชนิดสตริง (string) ที่สนับสนุนโดย C หรือ C++ นั้นเป็น null-terminated ซึ่งแตกต่างจากสตริงของ Visual Basic 6.0 โดยสิ้นเชิง ดังนั้น จึงไม่สามารถส่งผ่านสตริงของ Visual Basic 6.0 ให้กับฟังก์ชันวินโดวส์ API โดยตรง แต่ ต้องการแปลงสตริงของ Visual Basic 6.0 ให้อยู่ในรูปแบบของ null-terminated

2.3.3 การประกาศฟังก์ชันวินโดวส์ API

ก่อนที่จะสามารถใช้งานฟังก์ชันวินโดวส์ API ในการเขียนโปรแกรม จะต้องทำการประกาศฟังก์ชันนั้นๆเสียก่อน เพื่อเป็นการกำหนด Prototype ในการติดต่อระหว่างฟังก์ชันวินโดวส์ API กับ Visual Basic 6.0 สำหรับการประกาศฟังก์ชันวินโดวส์ API ซึ่งมีอยู่ด้วยกัน 2 รูปแบบ คือ การประกาศ Procedure ประเภท Sub Routine และ การ Procedure ประเภทฟังก์ชัน

รูปแบบที่ 1

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

รูปแบบที่ 2

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

Public	กำหนดให้ Procedure ที่ถูกประกาศสามารถถูกเรียกใช้ได้จาก Module หรือ Form อื่นใน 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 ซึ่งจะไม่มีการส่งค่ากลับ
Function	กำหนดให้ Procedure เป็นชนิดฟังก์ชัน ซึ่งจะมีค่าส่งกลับ ดังนั้น จึงสามารถนำค่าที่ได้จากฟังก์ชันไปใช้ในลักษณะของตัวแปรหนึ่งในนิพจน์ (expression) เช่น นิพจน์การคำนวณ
Name	กำหนดชื่อของ Procedure สำหรับ Visual Basic 6.0 ขอมให้ชื่อของ Procedure ความยาวทั้งหมดเพียง 40 ตัวอักษรเท่านั้น
Lib	กำหนดให้สตริงที่ตาม ("libname") จะบ่งบอกถึงชื่อของไฟล์ไลบรารี .dll ซึ่งในการประกาศฟังก์ชันวินโดวส์ API จะต้องมีการกำหนดชื่อของไฟล์ไลบรารี .dll ทุกครั้ง
Alias	กำหนดให้ Procedure ชื่อ "aliasname" สามารถถูกเรียกใช้ได้โดยใช้ชื่ออื่นๆ ตามที่กำหนดโดย name ทั้งนี้เนื่องจากในบางกรณีชื่อของฟังก์ชันวินโดวส์ API นั้นอาจจะซ้ำกับประโยคหรือคำสั่งของ Visual Basic 6.0 หรืออาจจะประกอบด้วยตัวอักษรที่ Visual Basic 6.0 ไม่ยอมรับในการกำหนดเป็นชื่อของ Procedure
Arglist	รายการของพารามิเตอร์ที่ส่งให้กับ Procedure
ByVal	กำหนดให้พารามิเตอร์มีการส่งค่าแบบ passed by value แต่ถ้าหากนำไปใช้ประกาศฟังก์ชันวินโดวส์ API ก็จะมีความหมายตามแต่ชนิดข้อมูลของพารามิเตอร์ตามที่เคยกล่าวรายละเอียดมาแล้วข้างต้น เช่น Byte StrVal\$ หมายถึง ASCIIZ
ByRef	กำหนดให้พารามิเตอร์มีการส่งค่าแบบ passed by reference
ParamArray	ใช้กำหนดให้พารามิเตอร์ดังกล่าวเป็น array ของข้อมูลชนิด variant ซึ่งจะต้องเป็นพารามิเตอร์ตัวสุดท้ายของ Procedure เท่านั้น โดยทั่วไปส่วนนี้มักจะไม่สามารถ

เอกสารนี้เป็นเอกสารที่...
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้ได้กับฟังก์ชันวินโดวส์ 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, lParam 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

wMsg หมายเลข message ของวินโดวส์ ซึ่งเป็นข้อมูลชนิด long

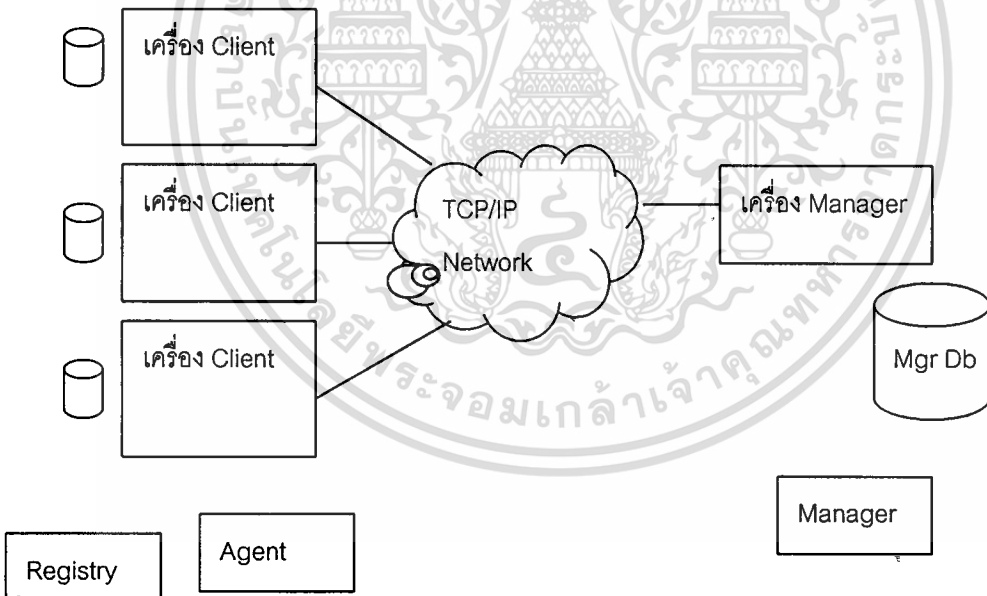
wParam ข้อมูลสำหรับการทำงานของฟังก์ชัน ซึ่งเป็นข้อมูลชนิด Long

lParam ข้อมูลสำหรับการทำงานของฟังก์ชัน ซึ่งเป็นข้อมูลชนิด Long

บทที่ 3

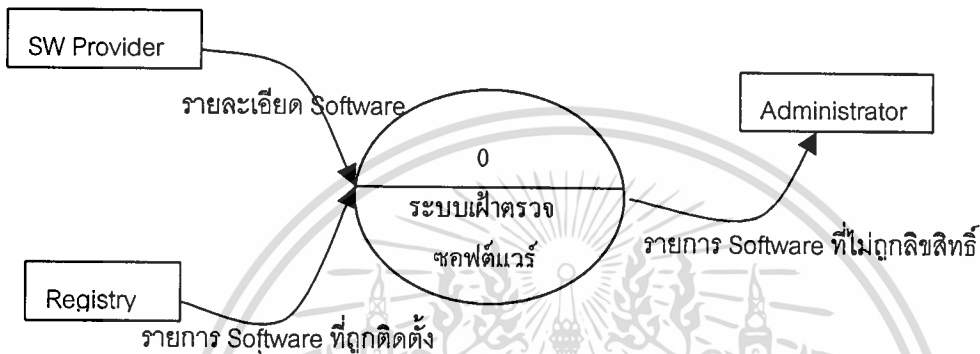
การออกแบบโครงสร้างระบบ

ในขั้นการตอนการออกแบบได้ออกแบบโดยแบ่งการทำงานออกเป็น 3 ส่วนตามลักษณะหน้าที่การทำงาน โดยในระบบประกอบไปด้วย Manager ที่ทำหน้าที่เป็นศูนย์กลางในการเก็บรวบรวมข้อมูลการติดตั้ง Software ที่ได้ทำการติดตั้งไว้ตามเครื่อง Client Computer และยังทำหน้าที่ในการวิเคราะห์และเปรียบเทียบกับ Software ที่ทาง Manager จัดเก็บอยู่ พร้อมทั้งจัดการกับเครื่อง Computer ที่ได้ติดตั้ง Software ที่ละเมิดลิขสิทธิ์ และในระบบยังประกอบด้วย Agent ที่ได้ติดตั้งไว้ในเครื่อง Client Computer ทุกเครื่องในระบบ โดยมีหน้าที่หลักในการบริการทำงานตามที่ทาง Manager ร้องขอมา ทั้งนี้ Manager และ Agent ได้ติดต่อกันผ่านเครือข่าย TCP/IP โดยที่ Agent ทำหน้าที่เป็นผู้ให้บริการ ดังนั้นสามารถออกแบบภาพรวมของระบบได้ดังนี้



รูปที่ 3.1 แสดงภาพรวมของระบบ

จากภาพรวมของระบบสามารถนำมาออกแบบ และเขียน Context Diagram และแผนภาพการไหลข้อมูล(Data Flow Diagram) ได้ดังนี้



รูปที่ 3.2 แสดง Context Diagram ของระบบเฝ้าตรวจซอฟต์แวร์

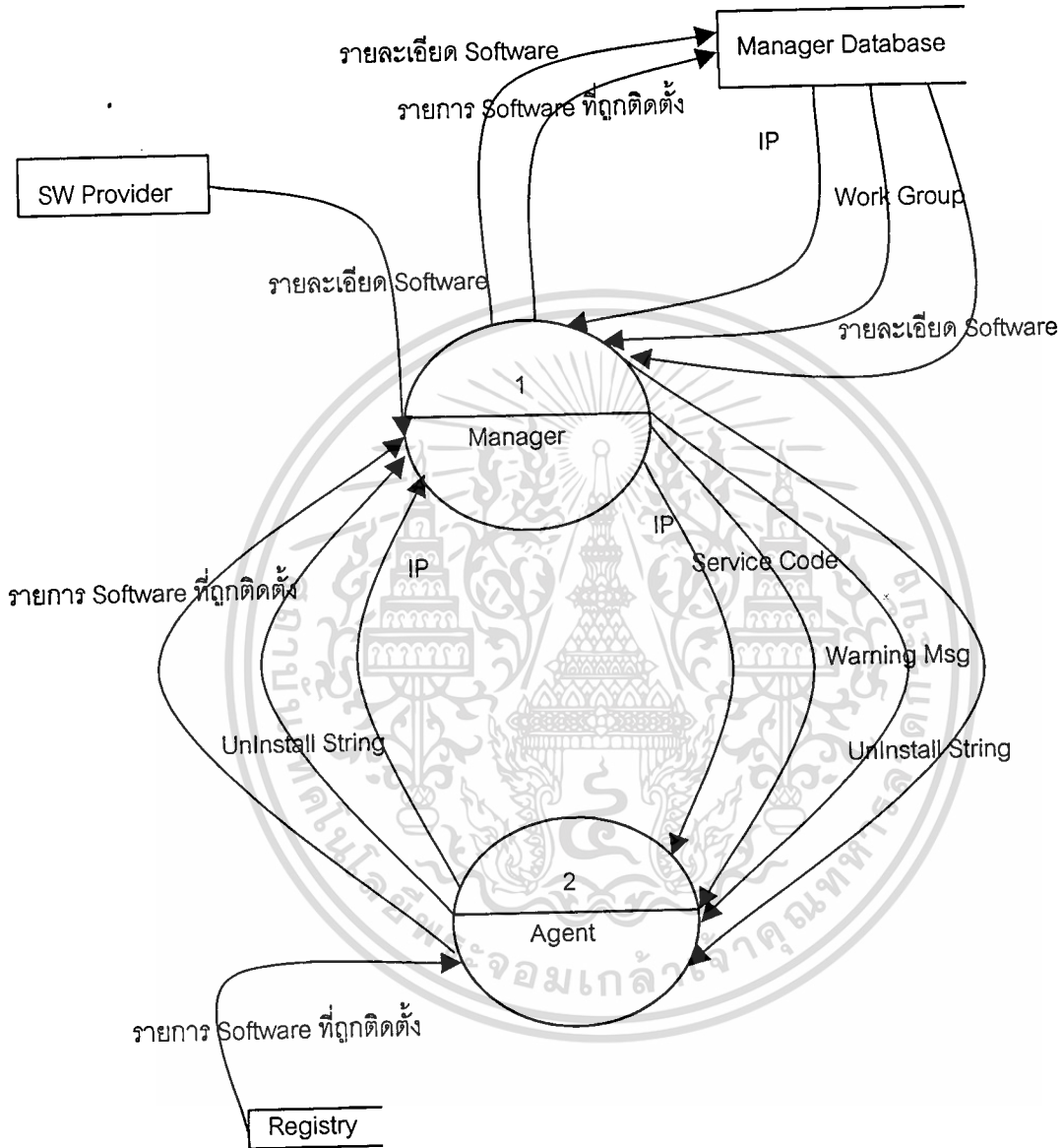
จากรูปที่ 3.2 สามารถอธิบายได้ว่าในระบบเฝ้าตรวจซอฟต์แวร์นั้นต้องรับข้อมูลจากภายนอกระบบ 2 แหล่งด้วยกันคือ

- Software Provider เป็นผู้ที่จำหน่ายหรือบริการ Software ซึ่งเป็นผู้ที่ให้รายละเอียดเกี่ยวกับ Software เช่น ราคา จำนวนลิขสิทธิ์ หมายเลขลิขสิทธิ์(License Number)
- Registry เป็นแหล่งเก็บข้อมูลของรายการ Software ที่ถูกติดตั้งไว้ในแต่ละเครื่องของ Computer ในบริษัทหรือองค์กร

ดังนั้นระบบเฝ้าตรวจ Software จะนำข้อมูลจากทั้ง 2 แหล่งนั้นมาทำการเปรียบเทียบกันว่า Software ที่ได้ทำการติดตั้งไว้ในแต่ละเครื่องของบริษัทหรือองค์กรนั้นเป็น Software ที่ถูกต้องตามลิขสิทธิ์ที่ทางองค์กรมีอยู่หรือไม่

ส่วนการออกแบบการทำงานของระบบเฝ้าตรวจ Software นั้นสามารถแบ่งการทำงานออกเป็น 2 ส่วนคือ Manager ที่ทำงานเป็นศูนย์กลางในการเก็บรวบรวมรายการ Software ที่ติดตั้งตามแต่ละเครื่อง และส่วน Agent ที่ทำงานอยู่ตามแต่ละเครื่องมีหน้าที่ในการคอยรับคำสั่งร้องขอจาก Manager ในการส่งข้อมูลรายการ Software ที่ถูกติดตั้งไว้ในเครื่องที่ Agent ทำงานอยู่ รวมทั้งการทำงานอื่นๆ ตามที่ Manager ได้ร้องขอเพื่อเป็นการประกอบการทำงาน of ระบบเฝ้าตรวจ Software

ให้สมบูรณ์มากขึ้น การออกแบบการทำงานของระบบเฝ้าตรวจ Software สามารถเขียนเป็น Data Flow Diagram (DFD level 1) ได้ดังนี้



รูปที่ 3.3 แสดง DFD Level 1 ของระบบเฝ้าตรวจซอฟต์แวร์

3.1 การออกแบบการทำงานของ Manager

เนื่องจาก Manger เป็นส่วนที่ทำหน้าที่เป็น ศูนย์กลางในการเก็บรวบรวม ข้อมูลจาก Agent ที่ติดตั้งไว้ตาม Client ต่างๆ รวมทั้งมีหน้าที่ในการเปรียบเทียบกับ Software List ที่ตัว Manger มีอยู่ และจัดการกับเครื่อง Client ที่ติดตั้ง Software ที่ละเมิดลิขสิทธิ์ ดังนั้นสามารถออกแบบ Module หลักๆ ได้ดังนี้

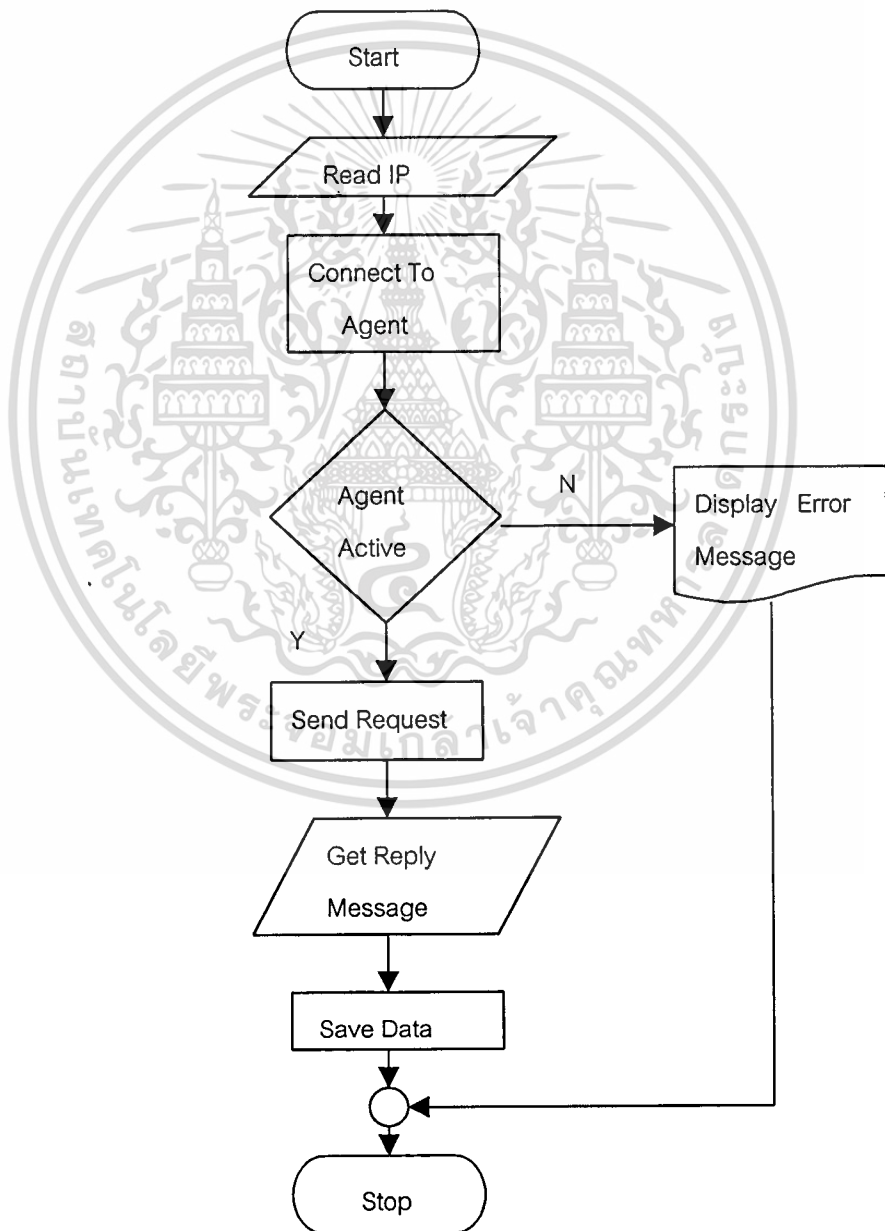
1. Module การเก็บรวบรวมข้อมูลการติดตั้ง Software ต่างๆที่ได้ติดตั้งไว้ที่ Client ต่างๆ โดยผ่านการทำงานของ Agent ที่ทำงานอยู่ประจำเครื่องนั้นๆ โดยการส่ง Request ที่ระบุ Service Code เป็นการขอ Query Software หลังจากได้รับคำตอบจากเครื่อง Agent แล้วก็ทำการแสดงผล พร้อมทั้งสามารถจัดเก็บเป็นข้อมูลสำหรับการวิเคราะห์เปรียบเทียบกับรายการ Software ที่องค์กรมีอยู่ในโอกาสต่อไป
2. Module การเปรียบเทียบ software ที่ได้จากการเก็บรวบรวมจากแต่ละ Client กับรายการ Software ที่ทางองค์กรมีอยู่ พร้อมทั้งแสดงผลให้ผู้ดูแลระบบทราบเพื่อที่ได้ดำเนินการต่อไป
3. Module Client Management เป็น Module สำหรับจัดการกับเครื่อง Client ที่ได้ติดตั้ง software ที่ละเมิดลิขสิทธิ์โดยมีระดับของการจัดการเป็น 3 ระดับคือ
 - 3.1 แจ้งข้อความเตือนขอให้เจ้าของเครื่อง Client ทำการ Uninstall Software ออกจากเครื่องด้วยตัวเอง
 - 3.2 ทำการยกเลิกการติดตั้ง Software ที่ละเมิดลิขสิทธิ์โดย Manager (Remote Uninstall) ใช้ในกรณีที่เจ้าของเครื่อง Client ที่ได้ติดตั้ง Software ละเมิดลิขสิทธิ์ยังเพิกเฉยต่อข้อความแจ้งเตือน โดยไม่มีเหตุผลอันควร
 - 3.3 ทำการ Shutdown เครื่อง Client ที่ติดตั้ง Software ที่ละเมิดลิขสิทธิ์ ในกรณีที่ฝ่าฝืนนโยบายขององค์กรเกี่ยวกับการละเมิดลิขสิทธิ์หลายครั้ง
4. Module เกี่ยวกับการจัดการระบบสารสนเทศของ Manager เช่นการจัดการเก็บรายการ Software ที่ทางองค์กรมีอยู่ การจัดการรายการ Software ที่ติดตั้งไว้ใน Client แต่ละเครื่อง รวมทั้งการจัดการจัดเก็บรายละเอียดของเครื่อง Client ทุกเครื่องที่ได้มีการเชื่อมต่ออยู่ในระบบ เช่น หมายเลข IP หน่วยงานที่ใช้ ที่ตั้งของตัวเครื่อง

ลักษณะการทำงานของ Manager เป็นลักษณะการทำงานแบบ Client Application กล่าวคือ จะเป็นการส่ง Request ร้องขอข้อมูลจาก agent แล้วนำมาเก็บไว้ในฐานข้อมูลเพื่อนำมาใช้ในการวิเคราะห์และประมวลในขั้นตอนอื่นต่อไป จากลักษณะการทำงานดังกล่าวสามารถสรุปเป็นขั้นตอนได้ดังนี้

1. Manager เตรียมการร้องขอโดย Manager จะต้องบอกหมายเลข IP Address ของเครื่องที่
- เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ดูแลระบบใช้เอกสารนี้ในการดำเนินการใดๆ ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้องการจะร้องขอ และ ระบุ Service Code ที่ต้องการร้องขอ พร้อมทั้งบอกหมายเลข Port การให้บริการของ agent เป็น 3007

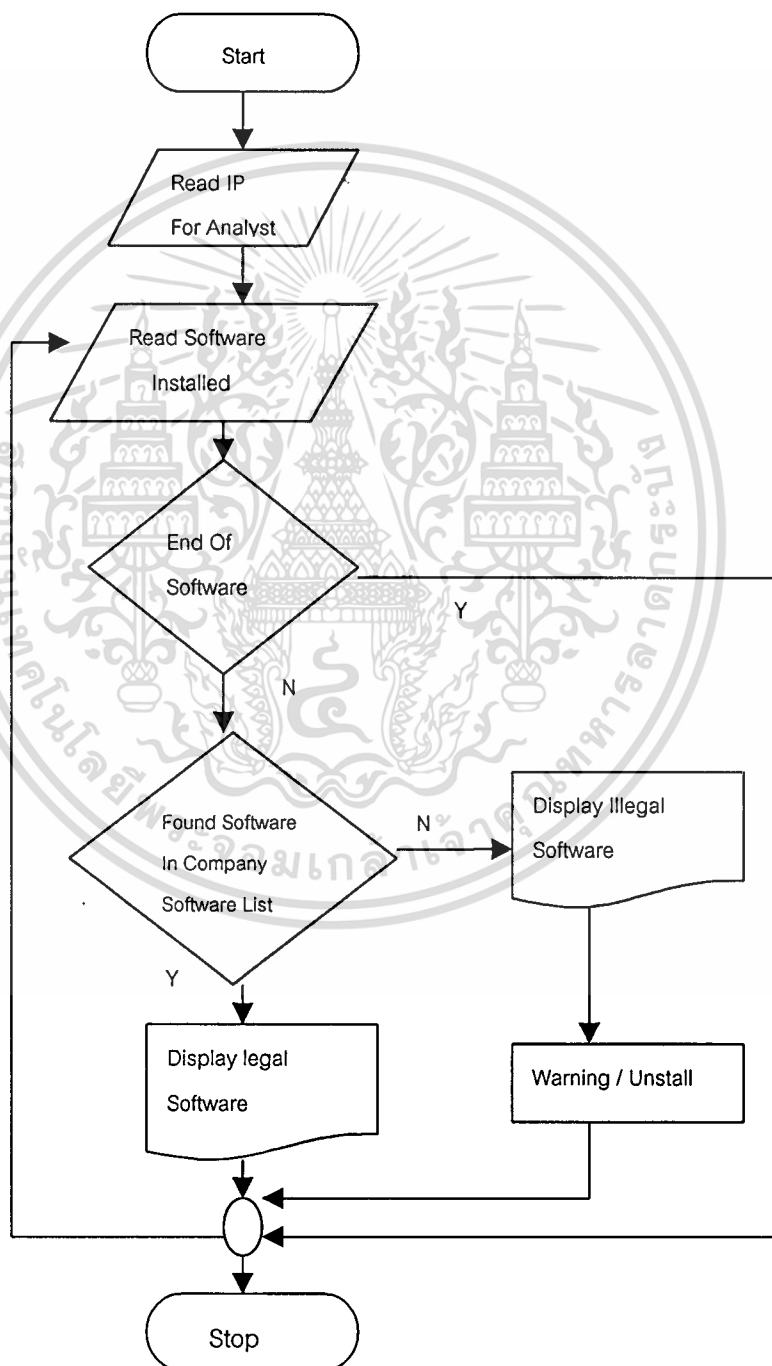
2. Application จะทำการตรวจสอบว่าเครื่อง Computer ดังกล่าวกำลังให้บริการอยู่หรือไม่ ซึ่งเป็นไปได้ว่าเครื่องดังกล่าวอาจมีได้เปิด (Power off) หรือ เปิดแบบ Offline
 3. ในกรณีที่เครื่อง ที่ agent ทำงานอยู่สามารถเปิดให้บริการได้ ก็ทำการส่ง request ไปขอข้อมูล
 4. เมื่อได้รับข้อมูลแล้วทาง manager ทำการ Save ข้อมูลลงฐานข้อมูล
- จากกระบวนการทำงานของ Manager ดังกล่าวสามารถเขียนเป็น Flow Chart ได้ดังนี้



รูปที่ 3.4 Flow Chart แสดงการทำงานของ Manager ในการทำ Software Collection

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่ที่สำคัญอีกอย่างหนึ่งของ Manager คือ การวิเคราะห์ Software ว่าถูกต้องตามลิขสิทธิ์ที่ทางบริษัทหรือองค์กรมีอยู่หรือไม่ ซึ่งสามารถทำได้โดย นำรายการ Software ที่ได้จากการเก็บรวบรวมในแต่ละเครื่องมาทำการค้นหาและเปรียบเทียบกับรายการ Software ที่ทางองค์กรหรือบริษัทมีอยู่ สามารถสรุปกระบวนการทำงานในการวิเคราะห์ความถูกต้องลิขสิทธิ์ของ Software เป็น Flow Chart ได้ดังนี้

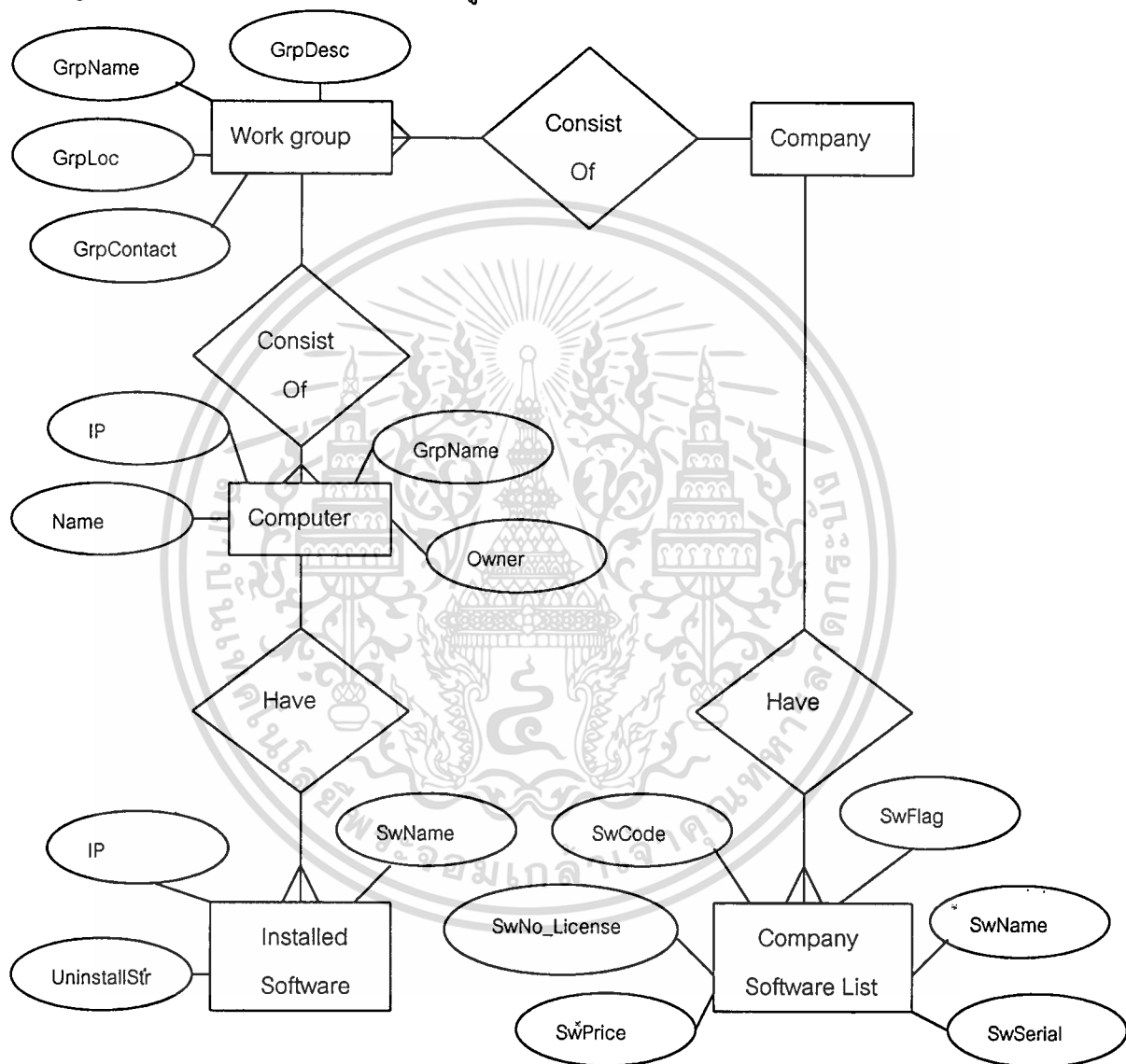


รูปที่ 3.5 Flow Chart แสดงการทำงานของ Manager ในการวิเคราะห์ลิขสิทธิ์

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการออกแบบการทำงานของ Manager ซึ่งมีหน้าที่หลักในการเก็บรวบรวมรายการ Software ที่ได้ติดตั้งไว้ในแต่ละเครื่อง และทำหน้าที่ในการวิเคราะห์ความถูกต้องของลิขสิทธิ์นั้น สามารถนำมาออกแบบการจัดเก็บข้อมูลในฐานข้อมูลโดยการเขียน Entity Relationship Diagram (ER Diagram) และออกแบบตารางจัดเก็บข้อมูลได้ดังนี้



รูปที่ 3.6 Entity Relation shipของระบบเฝ้าตรวจ Software

จาก Entity Relationship ดังกล่าวสามารถอธิบายได้ว่าในระบบเฝ้าตรวจ Software นั้นได้แบ่งหน่วยงานต่างๆที่มีหน้าที่การทำงานเหมือนกันให้อยู่ในกลุ่มการทำงานเดียวกัน (Workgroup) ซึ่งในแต่ละกลุ่มการทำงานนั้นประกอบไปด้วยเครื่อง Computer ที่ใช้ทำงานเป็นจำนวนหลายเครื่อง และในแต่ละเครื่องนั้นได้ทำการติดตั้ง Software ต่างๆไว้เป็นจำนวนหลายตัวซึ่งแต่ละตัวนั้น อาจ

เป็น Software ที่ถูกต้องตามลิขสิทธิ์ที่ทางองค์กรมีอยู่หรืออาจจะเป็น Software ที่ไม่สอดคล้องกับรายการ Software ที่ทางองค์กรมีอยู่

ดังนั้นจาก Entity Relationship สามารถนำมาออกแบบการจัดเก็บข้อมูลในฐานข้อมูลได้ดังนี้

1. ตาราง workgroup เก็บรายละเอียดกลุ่มการทำงาน

Filed Name	Field Type	Field Length	Description
GrpName	Text	25	Group Name
GrpDesc	Text	50	Group Description
GrpLoc	Text	50	Group Location
GrpContact	Text	50	Contact Person

2. ตาราง IP เก็บรายละเอียดเครื่อง Computer ต่างๆ

Filed Name	Field Type	Field Length	Description
Ip	Text	15	IP Address Number
GrpName	Text	25	Group Belong To
Name	Text	50	Computer Name
Owner	Text	50	Computer Owner

3. ตาราง Software รายละเอียด Software ขององค์กร

Filed Name	Field Type	Field Length	Description
SwCode	Text	15	Software Code
SwName	Text	225	Software Name
SwPrice	Numeric	10.2	Software Price
SwFlag	Boolean	1	Have License ?
Sw_No_license	Integer	10	Number of License
Sw_Serial	Text	50	Software Serial

4. ตาราง IP_Software เก็บรายละเอียด Software ที่ได้ทำการติดตั้งไว้ในแต่ละเครื่อง

Filed Name	Field Type	Field Length	Description
IP	Text	15	IP Address Number
SwName	Text	225	Software Name
UnInstallStr	Text	225	Uninstall String

3.2 การออกแบบการทำงานของ Agent

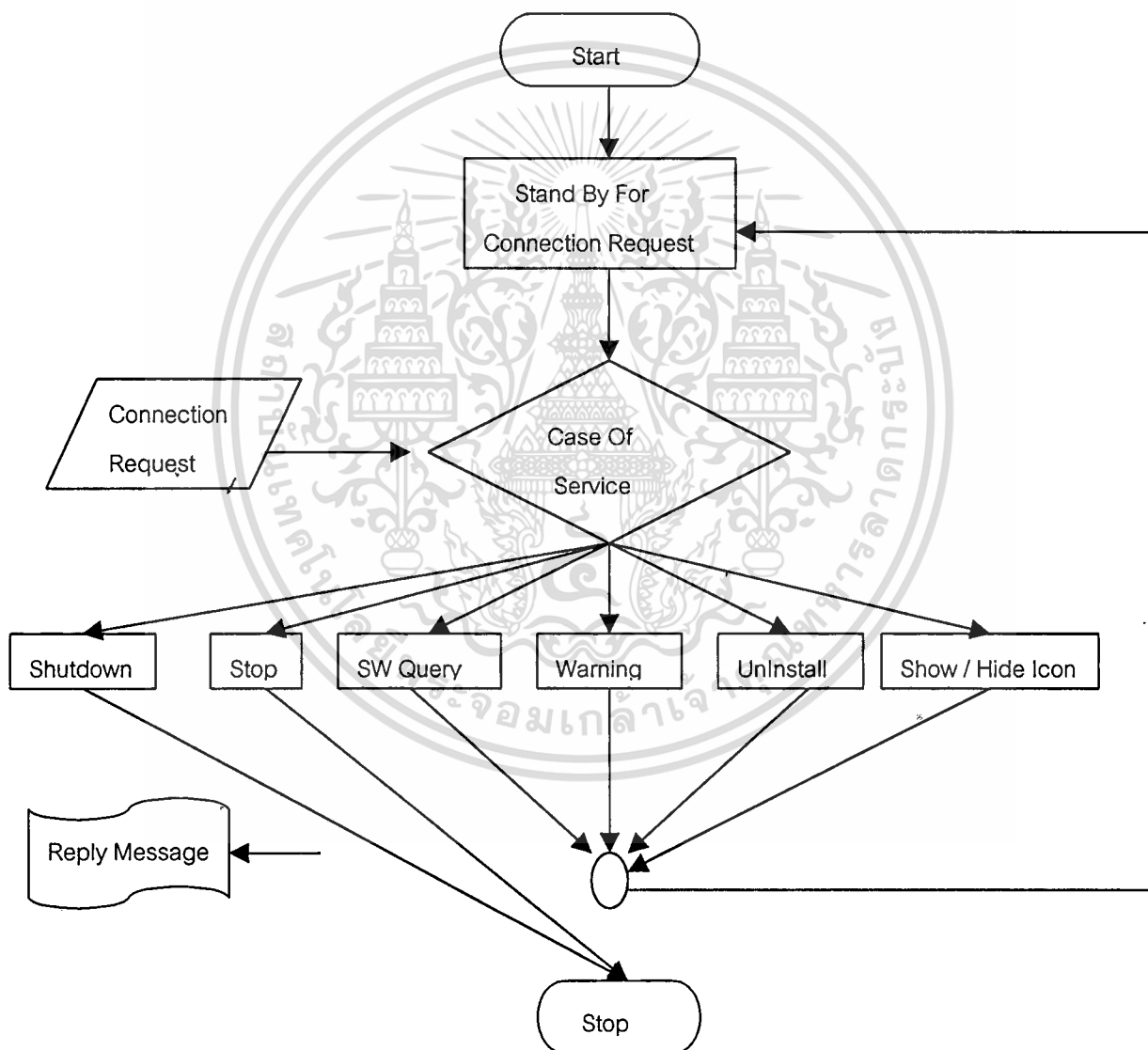
Agent เป็นส่วนที่ใช้ทำงานอยู่ที่เครื่อง Client โดยมีหน้าที่หลักในการให้บริการค้นหารายการ Software ที่ได้ทำการติดตั้งไว้ ณ เครื่อง Client ที่ Agent ทำงานอยู่ นอกจากนี้ตัว Agent ยังให้บริการอื่นตามที่ Manager ต้องการเช่น การแสดงข้อความแจ้งเตือน การทำการ Uninstall Software ที่ละเมิดลิขสิทธิ์ การ shutdown ระบบ รวมทั้งการซ่อน/แสดง Icon ของ Agent บน System Tray ดังนั้นสามารถออกแบบการทำงานของได้ดังนี้

1. Module Software Query โดยการสืบค้นรายการ Software ที่ได้ทำการติดตั้งไว้จาก ฐานข้อมูล Windows Registry ของเครื่อง Client ในแต่ละเครื่อง
2. Module Display Warning Message ทำหน้าที่รับ Warning Message ที่ได้รับจากการร้องขอจาก Manager
3. Module การ Shutdown ระบบทำหน้าที่ในการ Shutdown ระบบเมื่อได้รับ การร้องขอจาก Manager
4. Module การซ่อน/แสดง Icon ของ Agent บน System Tray

เนื่องจากลักษณะการทำงานของ Agent เป็นลักษณะการทำงานแบบ Service ที่ Run ฝังตัวอยู่ที่ เครื่อง Client ซึ่งสามารถให้บริการได้ตลอดเวลาที่ Run อยู่ ดังนั้นจึงจำเป็นต้องเปิด Socket รอการ Connect ไว้ตลอดเวลา สามารถสรุปเป็นขั้นตอนการทำงานของ Agent ได้ดังนี้

1. สร้าง Socket และ ใช้ Local Port ในการติดต่อสื่อสารเป็น 3007 แล้วทำการ Stand By (Listen) รอการ Connect จาก Manager
2. ในกรณีที่มีการ Connect จาก Manager ก็ทำการรับ Connection (Connection Establish) พร้อมทั้งทำการรับ Message มาทำการตรวจสอบว่าเป็น Service Code แบบไหน
3. Agent ทำตาม Service Code ที่ทาง Manager ต้องการ

4. ในกรณีที่สำเร็จ Agent ก็ทำการส่ง Output กลับไปยัง Manager และกรณีที่ไม่สามารถทำได้สำเร็จ ทาง Agent ทำการส่ง Error Message ไปบอก Manager
 5. Agent กลับมาอยู่ในสถานะ Standby รอรับการ Connect ในคราวต่อไป
- จากขั้นตอนการทำงานของ Agent ในระบบเฝ้าตรวจ Software ดังกล่าวสามารถเขียนสรุปเป็นแผนผังการทำงาน (Flow Chart) ได้ดังนี้



รูปที่ 3.7 Flow Chart แสดงการทำงานของ Agent

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การออกแบบการติดต่อสื่อสารระหว่าง Agent และ Manager

ในการติดต่อสื่อสารระหว่าง Manager และ Agent ได้ใช้การติดต่อสื่อสารผ่านเครือข่าย TCP/IP โดยที่ทาง Agent ได้ใช้ หมายเลข Port ในการให้บริการเป็น 3007 และได้เปิด Socket รอการเชื่อมต่อจากทาง Manager ดังนั้นการที่ Manager ต้องการที่ใช้ จาก Client เครื่องใดก็สามารถทำได้โดยการระบุหมายเลข IP ของเครื่อง Client ที่ต้องการใช้บริการพร้อมทั้งระบุหมายเลข Service Port ของ Agent Application เป็น 3007 พร้อมทั้งบอก Service Code ของการร้องขอ บริการแล้วทำการส่งไปยัง Agent ณ เครื่อง Client ที่ต้องการติดต่อด้วย สำหรับ Service Code ของการให้บริการสามารถกำหนดได้ดังนี้

Service Code	Description
SwQry	ทำการ Query Software ที่ได้ติดตั้งไว้
Warn	แสดงข้อความแจ้งเตือน
Shutdown	ทำการ Shutdown เครื่อง Client
StopAgent	หยุดการทำงานของ Agent
UninstSw	ทำการยกเลิกการติดตั้ง Software
AddIcon	แสดง Agent Icon บน System tray
RemIcon	ซ่อน Agent Icon บน System tray

รูปที่ 3.8 ตาราง Service Code ที่ใช้ในการร้องขอและให้บริการของ Manger และ Agent

บทที่ 4

การพัฒนาระบบเฝ้าตรวจซอฟต์แวร์

ในขั้นตอนของการพัฒนานั้นระบบเฝ้าตรวจ Software นั้น ก็ได้พัฒนา ส่วนงานต่างๆ (Module) ตามที่ได้ออกแบบเอาไว้ โดยใช้เครื่องมือในการพัฒนาดังนี้

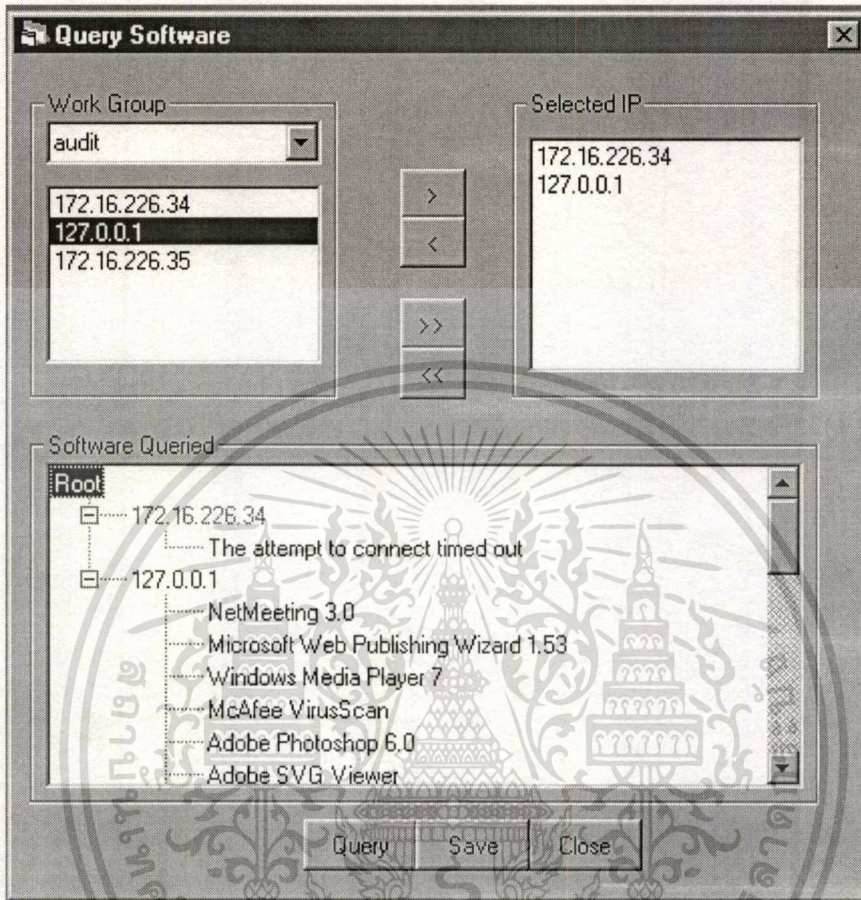
1. เครื่อง Computer PC ที่มีติดตั้ง Windows 95/98 พร้อมทั้งสามารถติดต่อกับเครือข่าย TCP/IP
2. Microsoft Visual Basic 6.0 และ Component ต่างๆที่จำเป็นในการพัฒนาส่วนงานด้านการติดต่อสื่อสารกับเครือข่าย และ Component ต่างๆที่จำเป็นในการติดต่อกับ Database
3. Microsoft Office 97 สำหรับจัดทำเอกสารประกอบการพัฒนาระบบ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

4.1 การพัฒนาส่วนการทำงานของ Manager

เป็นการพัฒนาส่วนที่มีหน้าที่ในการเก็บรวบรวมรายการ Software ที่ได้ทำการติดตั้งไว้ ณ เครื่อง Client ต่างๆ พร้อมทั้งทำการเปรียบเทียบกับรายการ Software ที่ทางองค์กรมีอยู่ และทำหน้าที่ในการจัดการกับเครื่อง Client ที่ได้ติดตั้ง Software ที่ละเมิดลิขสิทธิ์ นอกจากนี้ในส่วนของ Manager ยังทำหน้าที่ในการจัดการข้อมูลต่างๆของระบบเช่น รายการเครื่อง Computer หมายเลข IP รายการ Software

4.1.1 การพัฒนาส่วนการเก็บรวบรวมรายการ Software

เนื่องจากการใช้งานจริงนั้น ทางผู้ดูแลระบบสามารถเก็บรวบรวมรายการการติดตั้ง Software จากเครื่อง Client ได้จากหลายๆเครื่องพร้อมกัน โดยทำการระบุหมายเลข IP ของเครื่องที่ต้องการ พร้อมทั้งระบุ Remote Port เป็น 3007 แล้วทำการส่ง Service Code เป็น “SwQry” ออกไปยังเครื่อง Client แต่ละเครื่องที่ต้องการ จากนั้นก็ทำการรอกว่า Agent ที่เครื่อง Client ได้ตอบกลับมา หรือถ้าไม่สามารถติดต่อกับ Agent ที่เครื่อง Client ก็จะทำการแสดงเป็น Error Message ออกมาให้ทางผู้ดูแลระบบรับทราบ เมื่อได้รับคำตอบของรายการ Software จาก agent ที่ประจำแต่ละเครื่องแล้วทางผู้ดูแลระบบสามารถทำการบันทึกรายการ Software ดังกล่าวไว้ในฐานข้อมูล เพื่อที่จะสามารถนำมาเปรียบเทียบกับรายการ software ที่ทางองค์กรมีอยู่ได้ในภายหลังได้ จากการทำงานดังกล่าวสามารถสร้าง Screen ในการทำงานส่วนการเก็บรวบรวมรายการ software ได้ดังรูปที่ 4.1



รูปที่ 4.1 แสดงการพัฒนาส่วนการเก็บรวบรวมข้อมูล

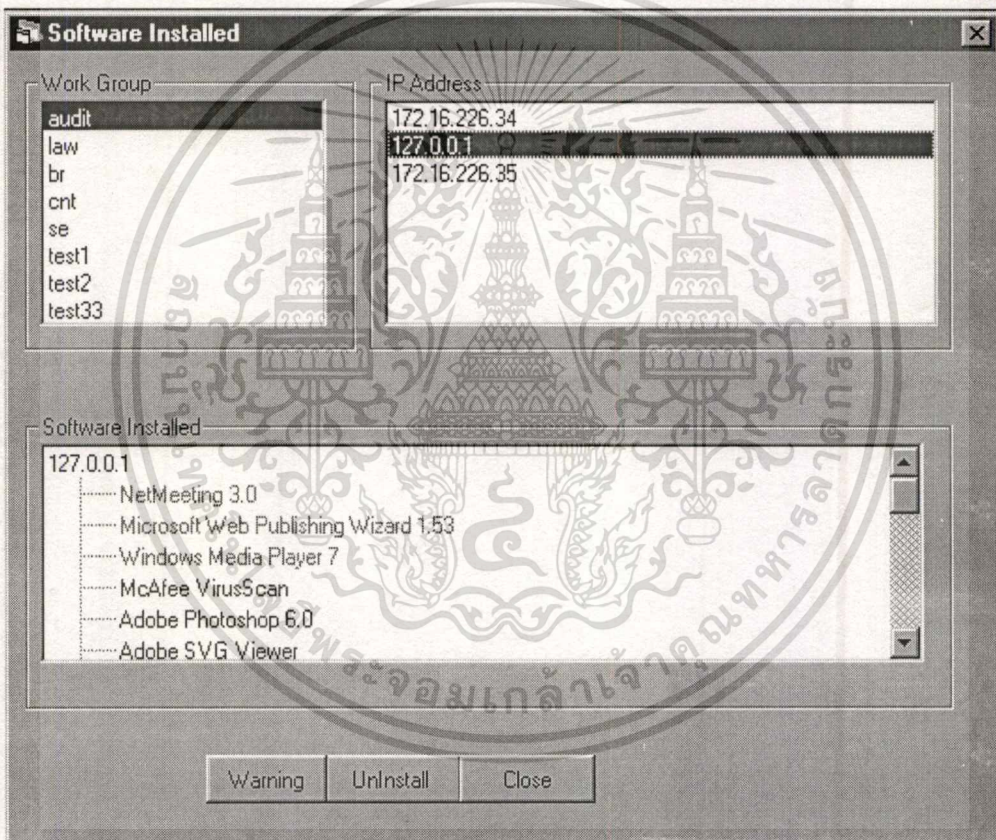
วิธีการใช้งานส่วนการเก็บรวบรวมข้อมูล

1. เลือก Work Group และหมายเลข IP ของเครื่องที่ต้องการจะเก็บรวบรวมข้อมูล จาก Combo Box Workgroup แล้ว Add เข้าไปยัง List Box ที่ชื่อ Selected IP ทำเช่นนี้ไปเรื่อยๆจนกว่าจะได้ IP ครบทุกเครื่องตามที่ต้องการ
2. กดปุ่ม Query เพื่อเริ่มต้นการส่งข้อมูลไปยัง Agent ที่ทำงานอยู่ ณ เครื่อง Client ทุกเครื่อง
3. ทาง Agent จะทำการ Reply ข้อมูลมาและปรากฏอยู่ใน List Box ชื่อ software queried แยกตามหมายเลข IP โดยแสดงเป็นลักษณะของ Tree กรณีที่มี Error เกิดขึ้นก็จะแสดงเป็น Error Message ให้ทางผู้ดูแลระบบทราบ
4. ทางผู้ดูแลระบบสามารถจัดเก็บข้อมูลดังกล่าวได้โดยกดปุ่ม Save
5. ออกจากส่วนการเก็บรวบรวมข้อมูลโดยการกดปุ่ม Close

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 การพัฒนาส่วนการเปรียบเทียบรายการ Software และการจัดการ Client

เป็นส่วนที่ใช้ในการเปรียบเทียบรายการ Software ที่ได้จากการเก็บรวบรวมจาก Client กับรายการ Software ที่มีอยู่ในระบบ โดยมีหลักการทำงานก็คือ นำรายการ Software ที่ได้จากเครื่อง Client มาทำการ Matching กับรายการ software ที่ในระบบมีอยู่ หากไม่สามารถ Match กันได้ หรือสามารถ Match กันกับ software ที่ไม่มีลิขสิทธิ์ ก็ถือว่าเป็น software ที่ไม่ถูกต้อง การพัฒนาส่วนการเปรียบเทียบ และ จัดการ Client สามารถแสดงได้ดังรูป



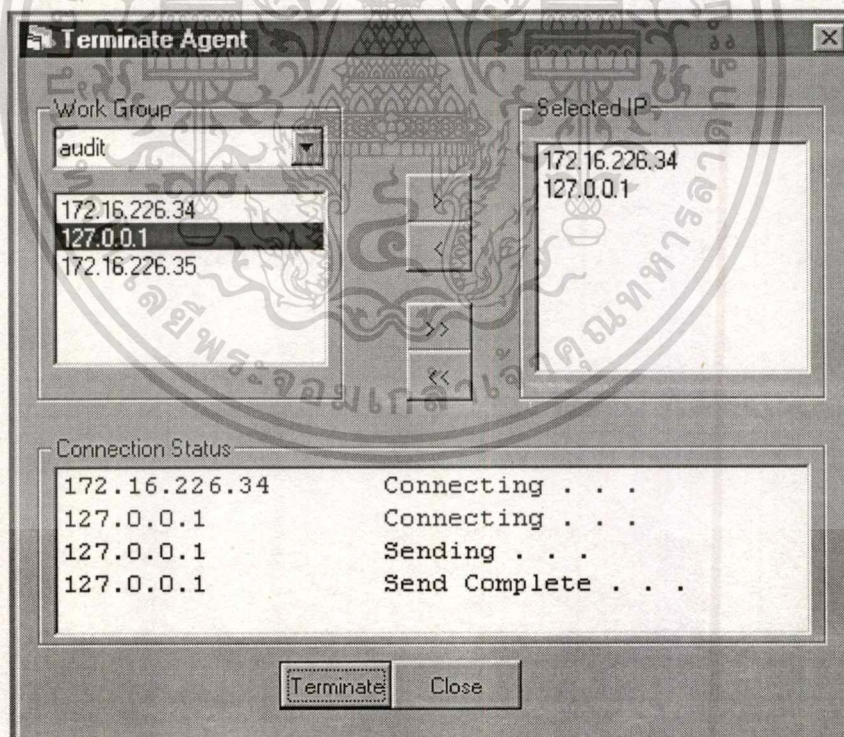
รูปที่ 4.2 แสดงการพัฒนาส่วนที่เปรียบเทียบและจัดการ Client

วิธีการทำงานของส่วนการเปรียบเทียบและการจัดการ Client ที่ติดตั้ง Software ละเมิดลิขสิทธิ์

1. จาก Main Menu เลือก Open ก็จะเข้าสู่ส่วนการดูและเปรียบเทียบ Software
2. เลือก Workgroup และหมายเลข IP ของเครื่อง ที่ต้องการเปรียบเทียบ ตัวโปรแกรม จะทำการ Query รายการ Software ที่ทำการติดตั้ง ณ เครื่องนั้นมา พร้อมทั้งนำไป Matching กับรายการ Software ที่ทางระบบมีอยู่แล้วทำการแสดงผลใน List Box Install software โดยที่ Software ที่มีลิขสิทธิ์ถูกต้องจะแสดงเป็นสีน้ำเงิน และ software ที่ไม่ถูกต้องจะแสดงเป็น สีแดง
3. ผู้ดูแลระบบสามารถใช้ Mouse Click ไปยัง Software ไม่ถูกต้องแล้วสามารถกดปุ่ม Warning หรือ Uninstall ได้

4.1.3 การพัฒนาส่วน Agent Management

เป็นการพัฒนาส่วนที่ใช้ควบคุมการทำงานของ Agent โดย Manager เช่นการ ซ่อน/แสดง Icon ของ Agent การหยุดการทำงานของ Agent การ Shutdown เครื่อง Agent เป็นต้น การพัฒนาส่วนการทำงาน Agent Management สามารถแสดงได้ดังรูป



รูปที่ 4.3 การพัฒนาส่วนของ agent management

วิธีการใช้งานส่วนงาน Agent Management

1. จาก Main menu เลือก Menu Agent ภายในประกอบด้วย menu ย่อยต่างๆในการจัดการ Agent เช่น การ Terminate ,Add/Remove Icon , shutdown agent ในที่นี้จะขอยกตัวอย่างการ Terminate Agent
2. เลือกหมายเลข IP ของเครื่องที่ต้องการจะ Manage พร้อมทั้ง add ไปยัง List box Select IP
3. กดปุ่ม ตาม Function ที่ต้องการทำงาน เช่น ปุ่ม Terminate ปุ่ม Add Icon To system Tray หรือ ปุ่ม shutdown ของแต่ละ Screen
4. ใน list box จะแสดง status ของแต่ละ connection ของแต่ละ Client ทำให้ผู้ดูแลระบบทราบ Status ของแต่ละ Connection

4.1.4 การพัฒนาส่วนการจัดการข้อมูล

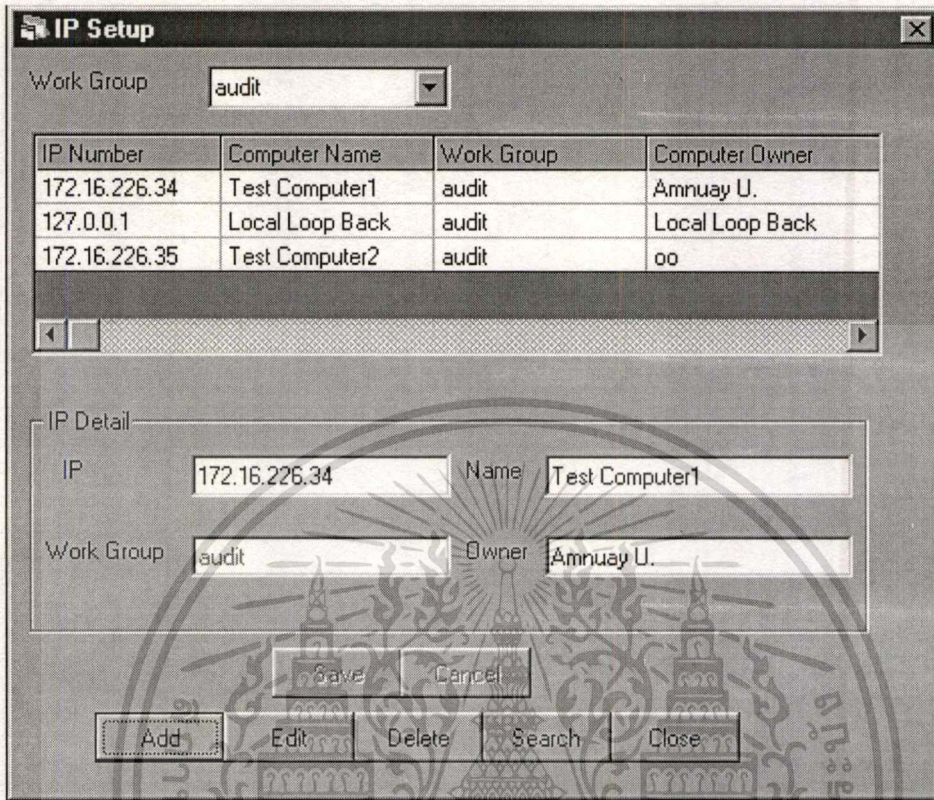
เป็นส่วนที่ใช้ในการจัดการกับข้อมูลของ Manager ข้อมูลที่จะจัดการประกอบไปด้วยข้อมูลต่างๆดังนี้

1. ข้อมูล Workgroup เป็นการจับกลุ่มหน่วยงานที่ทำงานในกลุ่มเดียวกัน
2. ข้อมูล Client แต่ละเครื่องเช่น หมายเลข IP ,workgroup,ชื่อ Computer
3. ข้อมูลรายการ software ที่ทางองค์กรมีอยู่เช่นชื่อ software ,Provider, ราคาที่ซื้อมา ,status การมี License ,จำนวน License

ในการจัดการข้อมูลแต่ละส่วนนั้นจะประกอบไปด้วย Function ที่เหมือนกัน ดังนี้

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

ดังนั้นสามารถพัฒนาส่วนการจัดการข้อมูลได้รูปที่ 4.4 ดังนี้



รูปที่ 4.4 แสดงการพัฒนาส่วนการจัดการข้อมูล

4.2 การพัฒนา Agent

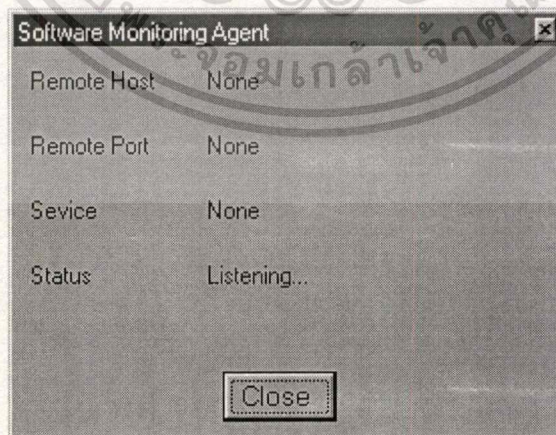
ในการพัฒนา Agent นั้น ได้พัฒนาเป็น Service ที่ Run ฝั่งตัวไ้ ณ Client โดยจะแสดง Icon ของ Agent ไว้ใน System Tray โดยลักษณะการทำงานจะเป็น Service ที่ Stand by รอการ Connect จาก Manager และหลังจากการได้รับ Request จาก Manager แล้ว ทาง Agent จะทำการตรวจสอบว่าเป็น Service Code แบบใด ก็จะไปเรียก Module นั้นมาทำงาน โดยที่ Agent มี Module การทำงานทั้งหมดดังนี้

1. Software Query เป็น Module หลักที่สำคัญของ agent ทำหน้าที่ในการ Software ที่ได้ติดตั้งไว้ในระบบ มีหลักการทำงานคือเข้าไปสืบค้นรายการ software ใน windows registry ในหมวดของ HKEY_LOCAL_MACHINE โดยไปที่ subkey ที่ ชื่อ Software\Microsoft\Windows\CurrentVersion\Uninstall ซึ่งเป็นที่เดียว ที่ Software มาตรฐาน ทำการ Register ไว้ในขั้นตอนการ Install เพื่อประโยชน์ในการทำเอกสาร Uninstall ภายหลัง โดยทำการเข้าถึงรายการ Software ทุกรายการแล้วทำการเก็บค่าใน Key ที่การค่าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อ Displayname Key และ ค่าใน Key UninstallString เก็บใส่ ตัวแปรแบบ Array และทำการ Reply กลับไปยัง Manager

2. UnInstall Software มีหน้าที่ในการ Uninstall software โดยรับ Request จาก Manger พร้อมรับ Uninstall String จากนั้นทาง Agent ก็เพียงแค่ Execute Uninstall String ที่ Client
3. Add Icon To System Tray เป็นการเพิ่ม Icon ของ Agent เข้าไปใน System Tray เพื่อให้สามารถ Click เข้าไปดู Status ของ Agent ได้ ในวิธีการ Add สามารถทำได้โดยเรียกใช้ Function ของ Windows API
4. Remove Icon From system Tray เป็น Module ที่มีหน้าที่ในการซ่อนการแสดง Icon ของ agent ในกรณีที่ไม่ต้องให้เจ้าของเครื่องดู status ของ Agent ได้ วิธีการ Remove สามารถทำได้โดยเรียกใช้ Function ของ Windows API
5. Terminate Agent เป็นการจบการทำงานของ Agent ณ เครื่อง Client
6. Shutdown Computer เป็นสั่งให้ Agent ทำการ Shutdown เครื่อง Computer โดยทำการเรียกใช้ Function ของระบบ ผ่านทาง Function ของ Windows API
7. Display Warning Message ทำหน้าที่ในการแสดง Warning Message เมื่อได้รับการร้องขอจาก Manager

ส่วนการพัฒนาหน้าจอของ Agent สามารถแสดงได้ดังรูป



รูปที่ 4.5 แสดงหน้าจอของ Agent

จากรูปแสดงหน้าจอของ Agent สามารถอธิบายได้ดังนี้

1. Remote Host เมื่อมีการ Connect เข้ามาจะแสดงหมายเลข IP ของเครื่องที่ได้ Connect เข้ามา
2. Remote Port เมื่อมีการ Connect เข้ามาจะแสดงหมายเลข Port ของเครื่องที่ได้ Connect เข้ามา
3. Service เมื่อมีการ Connect เข้ามาจะแสดง Service Code ของเครื่องที่ได้ Connect เข้ามา
4. Status เป็นช่องที่ใช้แสดง Status ของ Agent ซึ่ง Status ที่เป็นไปได้ดังนี้
 - 4.1 Listen จะแสดงเมื่อ agent Standby รอการ connect
 - 4.2 Connection request จะแสดงเมื่อมีการร้องขอใช้บริการ
 - 4.3 Sending จะแสดงเมื่อ agent มีการส่งข้อมูล Reply กลับไปยัง Manager
 - 4.4 Close จะแสดงเมื่อมีการ Close Connection



บทที่ 5

สรุปผลการทดสอบระบบ

จากการพัฒนา Software ที่ใช้สำหรับการเฝ้าตรวจการติดตั้ง Software ต่างๆ ของเครื่อง Computer ในองค์กร และได้ทำการทดสอบโดยการให้หมายเลข IP เป็น Loop Back Address (127.0.0.1) และ ใช้หมายเลข IP ของเครื่องอื่นๆ ในเครือข่าย ทดลอง Query Software จากเครื่องต่างๆ ดังกล่าวมาจัดเก็บไว้ในฐานข้อมูลของ Manager แล้วทำการ Make ข้อมูลให้ Software บางตัวมีลิขสิทธิ์ และบางตัวไม่มีลิขสิทธิ์และทำการเปรียบเทียบสามารถสรุปผลได้ดังนี้

5.1 ผลการทดสอบ

1. ระบบสามารถเปรียบเทียบ Software ที่มีลิขสิทธิ์และไม่มีลิขสิทธิ์ได้ โดยทำการแสดงผลเป็นสีน้ำเงิน (มีลิขสิทธิ์) และสีแดง (ไม่มีลิขสิทธิ์) ได้ถูกต้อง
2. ระบบสามารถจัดการกับเครื่อง Client ที่ติดตั้ง software ที่ไม่มีลิขสิทธิ์โดยการแจ้งเตือน การทำ Remote Uninstall รวมทั้งการ Shutdown ได้อย่างถูกต้อง
3. ในกรณีที่ไม่สามารถติดต่อกับเครื่อง Client บางเครื่องได้ ระบบสามารถระบุ Error ได้อย่างถูกต้อง ทำให้ผู้ดูแลระบบรับทราบได้
4. ในการจัดการเรื่องข้อมูลของ Manager พบว่าสามารถทำงานได้อย่างสมบูรณ์

5.2 ข้อควรปรับปรุงเพิ่มเติม

1. เนื่องจากข้อจำกัดของเวลาในการพัฒนาทำให้ Source Code ของโปรแกรมในส่วนงานต่างๆ ยังไม่มีความกระชับ กะทัดรัด ดังนั้นจึงควรมีการทำ Optimize Source Code ของโปรแกรมในแต่ละส่วนงาน เพื่อให้ระบบมีประสิทธิภาพมากขึ้น
2. ในการสร้าง Connection ระหว่าง agent และ manager นั้น ในกรณีที่ไม่สามารถติดต่อ Client ได้ยังคงใช้เวลานานกว่าที่ทางผู้ดูแลระบบจะทราบได้
3. ควรมีการสร้าง Report ที่สำคัญบาง Report เช่น รายการ Software ที่มีลิขสิทธิ์ที่ไม่มีลิขสิทธิ์ไม่ถูกต้อง
4. ควรมีการออกแบบและพัฒนาส่วนการวิเคราะห์และเปรียบเทียบ ให้มีรูปแบบการใช้งานที่มีประสิทธิภาพมากขึ้น

5.3 ปัญหาในการพัฒนาระบบเฝ้าตรวจ Software

1. ตัว Agent ไม่สามารถทราบค่าของ Software Serial No. ได้ เนื่องจาก Software ที่มี Provider ที่แตกต่างกันมักจะลงทะเบียนใน Windows Registry ใน Subkey ที่แตกต่างกัน รวมทั้งใช้ค่า Key ที่แตกต่างกัน เช่นบาง software ใช้ Key =“License No” ในขณะที่บาง software ใช้ Key =“Product No” หรือ “Serial No” ทำให้การเปรียบเทียบ software ต้องใช้ชื่อในการเปรียบเทียบ แทนที่จะใช้ License No ซึ่งน่าจะมีประสิทธิภาพมากกว่า
2. การทำ Remote UnInstall ไม่สามารถทำได้อย่างสมบูรณ์เนื่องจากในขั้นตอนการทำงานนั้น ได้ส่ง UnInstall String ไปให้ Agent ทำการ Execute ที่เครื่อง Client ซึ่งในระหว่างทำการ Uninstall นั้น Software ส่วนใหญ่มักมีการถามยืนยัน (Confirm Message) ก่อนเสมอ ทำให้เจ้าของเครื่องสามารถทำการยกเลิกได้

เอกสารอ้างอิง

- [1] Roman Steven. Jan 2000. **Win32 API Programming With Visual Basic** . : O'reilly.
- [2] Craigg John Clark . 1998 . **Microsoft Visual Basic 6.0 Developer Workshop** .:Microsoft Press.
- [3] สุทธิศักดิ์ พงษ์ธนาพานิช. 2542 . **การเขียนโปรแกรมด้วย Visual Basic 6.0 ระดับสูง** .กรุงเทพฯ : โรงพิมพ์จุฬาลงกรณ์มหาวิทยาลัย.



ภาคผนวก

1. การทำงานของ Agent

1.1 Public Module Source Code For Agent เป็น Module กลางที่ใช้ประกอบการทำงานของ Agent ซึ่งประกอบด้วย sub Module ดังต่อไปนี้

```
' For Retrieve Software Installed
' Programmed By Amnuay Uthairungsri
'Id 42067071 IS7.IT.KMITL
Option Explicit
```

```
Type FILETIME
    dwLowDateTime As Long
    dwHighDateTime As Long
End Type
```

```
Public Const HKEY_CLASSES_ROOT = &H80000000
Public Const HKEY_CURRENT_CONFIG = &H80000005
Public Const HKEY_CURRENT_USER = &H80000001
Public Const HKEY_DYN_DATA = &H80000006
Public Const HKEY_LOCAL_MACHINE = &H80000002
Public Const HKEY_USERS = &H80000003
```

```
Public Const REG_OPTION_NON_VOLATILE = 0
Public Const REG_SZ = 1
Public Const REG_CREATED_NEW_KEY = &H1
Public Const REG_OPENED_EXISTING_KEY = &H2
```

```
Public Const READ_CONTROL = &H20000
```

```
Public Const SYNCHRONIZE = &H100000
Public Const STANDARD_RIGHTS_ALL = &H1F0000
Public Const STANDARD_RIGHTS_EXECUTE = (READ_CONTROL)
Public Const STANDARD_RIGHTS_READ = (READ_CONTROL)
Public Const STANDARD_RIGHTS_REQUIRED = &HF0000
Public Const STANDARD_RIGHTS_WRITE = (READ_CONTROL)
```

```
Public Const KEY_CREATE_LINK = &H20
Public Const KEY_CREATE_SUB_KEY = &H4
Public Const KEY_ENUMERATE_SUB_KEYS = &H8
Public Const KEY_EVENT = &H1
Public Const KEY_NOTIFY = &H10
Public Const KEY_QUERY_VALUE = &H1
Public Const KEY_READ = ((STANDARD_RIGHTS_READ Or KEY_QUERY_VALUE Or
KEY_ENUMERATE_SUB_KEYS Or KEY_NOTIFY) And (Not SYNCHRONIZE))
Public Const KEY_SET_VALUE = &H2
Public Const KEY_WRITE = ((STANDARD_RIGHTS_WRITE Or KEY_SET_VALUE Or
KEY_CREATE_SUB_KEY) And (Not SYNCHRONIZE))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```
lpcbData As Long) As Long
```

```
Public Function SwQry(ByRef totalSw As Integer) As String()
```

```
' Search the registry for Data Providers
```

```
Const BUF_LEN As Long = 2048
```

```
Const MAX_SW As Integer = 255
```

```
Dim lret As Long
```

```
Dim hCLSIDKey As Long
```

```
Dim hClassKey As Long
```

```
Dim hClassSubKey As Long
```

```
Dim bufKeyName As String * BUF_LEN
```

```
Dim bufKeyName2 As String * BUF_LEN
```

```
Dim lbufValue As Long
```

```
Dim bufValue As String * BUF_LEN
```

```
Dim lValueType As Long
```

```
Dim ft As FILETIME
```

```
Dim lxKey As Long, lxKey2 As Long
```

```
Dim SubkeyStr As String
```

```
Dim swTmp(MAX_SW) As String
```

```
Dim bProvider As Boolean
```

```
totalSw = 0
```

```
SubkeyStr = "Software\Microsoft\Windows\CurrentVersion\Uninstall"
```

```
' Open CLSID key for read-only
```

```
lret = RegOpenKeyEx(HKEY_LOCAL_MACHINE, SubkeyStr, 0, KEY_READ, hCLSIDKey)
```

```
If lret <> ERROR_SUCCESS Then
```

```
    MsgBox "Cannot open Subkey key", vbCritical
```

```
    Exit Function
```

```
End If
```

```
lxKey = -1 ' Key index
```

```
Do
```

```
    ' Next subkey index
```

```
    lxKey = lxKey + 1
```

```
    ' Set up buffer for key name
```

```
    bufKeyName = String(BUF_LEN, 0)
```

```
    ' Get next subkey of CLSID
```

```
    lret = RegEnumKeyEx(hCLSIDKey, lxKey, bufKeyName, BUF_LEN, _  
        0, vbNullString, 0, ft)
```

```
If lret <> ERROR_SUCCESS Then Exit Do
```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยราชภัฏวชิรเวศน์บุรีรัมย์ การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

' Open the key
Iret = RegOpenKeyEx(HKEY_LOCAL_MACHINE, _
    SubkeyStr & "\" & Trim0(bufKeyName), _
    0, KEY_QUERY_VALUE, hClassSubKey)

If Iret = ERROR_SUCCESS Then
    lbufValue = 0
    bufValue = String(BUF_LEN, 0)

    ' Get the default value
    ' First get the size of the data and the value type
    Iret = RegQueryValueEx(hClassSubKey, "DisplayName", 0&, _
        lValueType, 0&, lbufValue)

    ' Is the data a string?
    If lValueType = REG_SZ Then

        ' Get the actual string data

        Iret = RegQueryValueExStr(hClassSubKey, "DisplayName", 0&, _
            lValueType, bufValue, lbufValue)

        'Debug.Print Trim0(bufValue)
        totalSw = totalSw + 1
        swTmp(totalSw) = Trim0(bufValue)

    'RegCloseKey hClassSubKey
End If
'add uninstall string
' then you must begin qry here
' Open the key
Iret = RegOpenKeyEx(HKEY_LOCAL_MACHINE, _
    SubkeyStr & "\" & Trim0(bufKeyName), _
    0, KEY_QUERY_VALUE, hClassSubKey)

If Iret = ERROR_SUCCESS Then
    lbufValue = 0
    bufValue = String(BUF_LEN, 0)

    ' Get the default value
    ' First get the size of the data and the value type
    Iret = RegQueryValueEx(hClassSubKey, "UnInstallString", 0&, _
        lValueType, 0&, lbufValue)

    ' Is the data a string?
    If lValueType = REG_SZ Then

        ' Get the actual string data

        Iret = RegQueryValueExStr(hClassSubKey, "UnInstallString", 0&, _
            lValueType, bufValue, lbufValue)

```

```

'Debug.Print Trim0(bufValue)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
swTmp(totalSw) = swTmp(totalSw) & "=" & Trim0(bufValue)
```

```
'RegCloseKey hClassSubKey
```

```
End If
```

```
End If
```

```
RegCloseKey hClassSubKey
```

```
'and end here
```

```
End If
```

```
Loop
```

```
RegCloseKey hCLSIDKey
```

```
SwQry = swTmp
```

```
End Function
```

```
Public Function Trim0(sName As String) As String
```

```
' Keep left portion of string sName up to first 0. Useful with Win API null terminated strings.
```

```
Dim X As Integer
```

```
X = InStr(sName, Chr$(0))
```

```
If X > 0 Then Trim0 = Left$(sName, X - 1) Else Trim0 = sName
```

```
End Function
```

```
Public Sub ShutDown()
```

```
MsgBox "System Shutdown Now", vbOKOnly + vbExclamation
```

```
ExitWindowsEx EWX_SHUTDOWN, 0
```

```
End Sub
```

```
Public Sub Warn(ByVal dMsg As String)
```

```
MsgBox dMsg, vbOKOnly + vbExclamation, "Software Monitoring"
```

```
End Sub
```

```
Public Function getMyIP() As String
```

```
Const BUF_LEN As Long = 2048
```

```
Const SubKeyIP = "System\CurrentControlSet\Services\Class\NetTrans\0000"
```

```
Dim lret, hCLSIDKey, lValueType, lbufValue As Long
```

```
Dim bufValue As String
```

```
'Open Reg Key and point by hCLSIDKey
```

```
lret = RegOpenKeyEx(HKEY_LOCAL_MACHINE, SubKeyIP, 0, KEY_READ, hCLSIDKey)
```

```
'Set Buff
```

```
lbufValue = 0
```

```
bufValue = String(BUF_LEN, 0)
```

```
'Test lValueType First
```

```
lret = RegQueryValueEx(hCLSIDKey, "IPAddress", 0&, _
```

```
lValueType, 0&, lbufValue)
```

```
'And Then Get the key
```

```
Iret = RegQueryValueExStr(hCLSIDKey, "IPAddress", 0&, _
    IValueType, bufValue, lbufValue)
```

```
'Assign to Function
getMyIP = Trim0(bufValue)
End Function
```

```
Public Function getMyName() As String
Const BUF_LEN As Long = 2048
Const SubKeyName = "System\CurrentControlSet\Control\ComputerName\ComputerName"
Dim Iret, hCLSIDKey, IValueType, lbufValue As Long
Dim bufValue As String
```

```
'Open Reg Key and point by hCLSIDKey
Iret = RegOpenKeyEx(HKEY_LOCAL_MACHINE, SubKeyName, 0, KEY_READ, hCLSIDKey)
```

```
'Set Buff
lbufValue = 0
bufValue = String(BUF_LEN, 0)
'Test lvalutype First
Iret = RegQueryValueEx(hCLSIDKey, "ComputerName", 0&, _
    IValueType, 0&, lbufValue)
'And Then Get the key
Iret = RegQueryValueExStr(hCLSIDKey, "ComputerName", 0&, _
    IValueType, bufValue, lbufValue)
```

```
'Assign to Function
getMyName = Trim0(bufValue)
End Function
```

```
Sub Uninstall(sw As String)
Dim retVal
On Error GoTo errhdl
retVal = Shell(sw, 1)
errhdl:
End Sub
```

1.2 Screen Source Code เป็น Module ที่ควบคุมการทำงานของหน้าจอ Agent (Agent Screen)

ประกอบด้วย Sub Module ต่างๆ ดังต่อไปนี้

```
Private Sub Form_Load()
```

```
tcpAgent.Protocol = sckTCPProtocol
tcpAgent.LocalPort = 3007
tcpAgent.Listen
```

```
lbStatus = "Listening..."
lbRmtPort = "None"
lbRmtIP = "None"
lbSrvType = "None"
```

```
frmAgent.Show
Me.Visible = False
AddIcon
```

```
End Sub
```

```
Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
'Event occurs when the mouse pointer is within the rectangular
'boundaries of the icon in the taskbar status area.
Dim msg As Long
Dim sFilter As String
'Left-click constants.
```

```
msg = X / Screen.TwipsPerPixelX
Select Case msg
Case WM_LBUTTONDOWN
```

```
Me.Visible = True
Case WM_LBUTTONDOWNBLCLK
Me.Visible = True
```

```
End Select
```

```
End Sub
```

```
Private Sub Form_Terminate()
```

```
RemIcon
End Sub
```

```
Private Sub tcpAgent_ConnectionRequest _
(ByVal requestID As Long)
```

```
If tcpAgent.State <> sckClosed Then
tcpAgent.Close
End If
```

```
lbStatus = "Connection Request "
lbRmtIP = tcpAgent.RemoteHostIP
```

```

lbRmtPort = tcpAgent.RemotePort
tcpAgent.Accept requestID
End Sub

```

```

Private Sub tcpAgent_DataArrival _
(ByVal bytesTotal As Long)
' Declare a variable for the incoming data.
' Invoke the GetData method and set the Text
' property of a TextBox named txtOutput to
' the data.

```

```

Dim recData As String
Dim trfDataBuff() As String
Dim trfDataIdx As Integer
Dim i As Integer
Dim myIP As String

```

```

tcpAgent.GetData recData
lbStatus = "Data Arrival... "

```

```

lbSrvType = Left$(recData, InStr(recData, " ") - 1)
recData = Mid(recData, InStr(recData, " ") + 1, Len(recData))

```

```

Select Case lbSrvType

```

```

Case "SwQry":
    trfDataBuff = SwQry(trfDataIdx)

```

```

    lbStatus = "Sending..."

```

```

    myIP = getMyIP

```

```

    For i = 1 To trfDataIdx

```

```

        tcpAgent.SendData trfDataBuff(i) & "|"

```

```

        lbStatus = "Sending..." & trfDataBuff(i)

```

```

        lbRmtIP = tcpAgent.RemoteHostIP

```

```

        lbRmtPort = tcpAgent.RemotePort

```

```

    Next i

```

```

Case "Warn":

```

```

    Warn (recData)

```

```

Case "Shutdown":

```

```

    ShutDown

```

```

Case "AddIcon":

```

```

    AddIcon

```

```

Case "RemIcon":

```

```

    RemIcon

```

```

Case "UninstSw":

```

```

    Uninstall (recData)

```

```

Case "StopAgent":

```

```

    Unload Me

```

```

End Select

```

```

End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub tcpAgent_close()
```

```
tcpAgent.Close  
tcpAgent.Listen
```

```
lbStatus = "Listening..."  
lbRmtPort = "None"  
lbRmtIP = "None"  
lbSrvType = "None"
```

```
End Sub
```

```
'Private Sub tcpAgent_SendComplete()  
'tcpAgent.Close  
'tcpAgent.Listen
```

```
'lbStatus = "Listening..."  
'lbRmtPort = "None"  
'lbRmtIP = "None"  
'lbSrvType = "None"  
'End Sub
```



2. Manager Source Code

2.1 Software Query เป็น Module ส่วนที่ใช้ในการติดต่อกับ Agent เพื่อขอให้ agent ทำการ query รายการ software ที่ได้ทำการติดตั้งไว้ และเมื่อได้ข้อมูลแล้วก็ทำการจัดเก็บข้อมูลลงฐานข้อมูลด้วย ดังรายละเอียดการทำงานดังต่อไปนี้

```
Dim Conn As ADODB.Connection
Dim nodX As Node
Dim rsIP As ADODB.Recordset
Private Type SW_TRAN
    ip As String * 15
    sw As String
    uninstall As String
End Type
```

```
Dim rsWrkGrp As ADODB.Recordset
Dim rsSw As ADODB.Recordset
Dim svcCode As String
Const SEP = " | "
Const DB_PROVIDER = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source="
Dim dbFile As String
Dim errCnt As Integer
Dim swTran() As SW_TRAN
Dim swtranIndex As Integer
Dim listImg As ListImage
Private Sub BtnClose_Click()
MDIMain.MDIForm_Load
Unload Me
End Sub
```

```
Private Sub btnMv_Click(Index As Integer)
Dim i As Integer
Select Case Index
Case 0 ' add one ip to selected listbox
If listIP1.ListIndex >= 0 Then
listIP2.AddItem listIP1.List(listIP1.ListIndex)
End If
Case 1 ' Remove active ip from selected listbox
If listIP2.ListIndex >= 0 Then
listIP2.RemoveItem listIP2.ListIndex
End If
Case 2 ' add all ip to selected list
For i = 0 To listIP1.ListCount - 1
listIP2.AddItem listIP1.List(i)
Next i
Case 3 ' Remove all from select ip list
listIP2.Clear
tvSw.Nodes.Clear
End Select
```

```
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub btnQry_Click()
Dim i, Cnt As Integer
On Error Resume Next
Cnt = 0
svcCode = "SwQry "
tvSw.Nodes.Clear
Set nodX = tvSw.Nodes.Add(, , "R", "Root")
tvSw.Nodes("R").Expanded = True
tvSw.Style = tvwTreelinesPlusMinusPictureText
'tvSw.ImageList = imgList

If tcpMgr.Count > 1 Then
  For i = 1 To tcpMgr.Count - 1
    tcpMgr(i).Close
    Unload tcpMgr(i)
  Next i
End If
If listIP2.ListCount > 1 Then
  For i = 1 To listIP2.ListCount - 1
    Load tcpMgr(i)
  Next i
End If
swtranIndex = 0
For i = 0 To listIP2.ListCount - 1
  Set nodX = tvSw.Nodes.Add("R", tvwChild, Trim$(listIP2.List(i)), Trim$(Trim$(listIP2.List
(i))))
  nodX.ForeColor = vbRed
  ' nodX.Image = 1
  DoEvents
  tcpMgr(i).RemoteHost = listIP2.List(i)
  tcpMgr(i).RemotePort = "3007"
  tcpMgr(i).Connect

```

```

Next i
End Sub

```

```

Private Sub btnSave_Click()
saveData
End Sub

```

```

Private Sub cboWrkGrp_Click()
GetIp cboWrkgrp.List(cboWrkgrp.ListIndex)
End Sub

```

```

Private Sub Form_Load()
'Load Data
dbFile = CurDir() & "\sw.mdb"
Set Conn = New ADODB.Connection
errCnt = 0
With Conn
  .ConnectionString = DB_PROVIDER & dbFile
  .Open
End With

```

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Set rsWrkGrp = Conn.Execute("select grpName from wrkgrp")
```

```
cboWrkgrp.Clear
```

```
listIP1.Clear
```

```
listIP2.Clear
```

```
Do While Not rsWrkGrp.EOF
```

```
    cboWrkgrp.AddItem rsWrkGrp("grpname")
```

```
    rsWrkGrp.MoveNext
```

```
Loop
```

```
If cboWrkgrp.ListCount > 0 Then
```

```
    cboWrkgrp.ListIndex = 0
```

```
    GetIp cboWrkgrp.List(0)
```

```
End If
```

```
End Sub
```

```
Sub GetIp(ByVal wg As String)
```

```
    Dim sqlStr As String
```

```
    sqlStr = "select ip from ip where wrkgrp = " & wg & """
```

```
    listIP1.Clear
```

```
    Set rsIP = Conn.Execute(sqlStr)
```

```
    Do While Not rsIP.EOF
```

```
        listIP1.AddItem rsIP("ip")
```

```
        rsIP.MoveNext
```

```
    Loop
```

```
    If listIP1.ListCount > 0 Then
```

```
        listIP1.ListIndex = 0
```

```
    End If
```

```
End Sub
```

```
Private Sub tcpMgr_Connect(Index As Integer)
```

```
    tcpMgr(Index).SendData svcCode
```

```
End Sub
```

```
Public Sub saveData()
```

```
    Dim Tmp, sqlStr As String
```

```
    Dim rsSw As ADODB.Recordset
```

```
    Tmp = ""
```

```
    For i = 1 To UBound(swTran)
```

```
        If swTran(i).ip <> Tmp Then
```

```
            sqlStr = "delete from swtran where trim(ip) = " & Trim(swTran(i).ip) & """
```

```
            Set rsSw = Conn.Execute(sqlStr)
```

```
            Tmp = swTran(i).ip
```

```
        End If
```

```
    Next i
```

```
    For i = 1 To UBound(swTran)
```

```
        With swTran(i)
```

```
            sqlStr = "insert into swtran values (" & .ip & ", " & .sw & ", " & .uninstall & ")"
```

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Set rsSw = Conn.Execute(sqlStr)
Next i
MsgBox "Save Complete", vbOKOnly, "Save Transaction"
Set rsSw = Nothing
Set Conn = Nothing

```

```
End Sub
```

```

Private Sub tcpMgr_DataArrival(Index As Integer, ByVal bytesTotal As Long)
Dim recData As String
Dim dataBuff(255) As String
Dim i As Integer
Dim eqSign As String
i = 0
If bytesTotal = 0 Then
Exit Sub
End If
tcpMgr(Index).GetData recData

Do While Len(recData) > 0
swtranIndex = swtranIndex + 1
ReDim Preserve swTran(swtranIndex)
i = i + 1
dataBuff(i) = Left(recData, InStr(recData, "|") - 1)
recData = Mid(recData, InStr(recData, "|") + 1, Len(recData))
swTran(swtranIndex).ip = tcpMgr(Index).RemoteHost
eqSign = InStr(dataBuff(i), "=")
If eqSign > 0 Then
swTran(swtranIndex).sw = Mid(dataBuff(i), 1, eqSign - 1)
swTran(swtranIndex).uninstall = Mid(dataBuff(i), eqSign + 1, Len(dataBuff(i)))
Else ' no Uninstall Str
swTran(swtranIndex).sw = dataBuff(i)
swTran(swtranIndex).uninstall = ""
End If
swTran(swtranIndex).ip = Trim$(tcpMgr(Index).RemoteHost)

' listQry.AddItem tcpMgr(Index).RemoteHost & SEP & i & SEP & dataBuff(i)
Set nodX = tvSw.Nodes.Add(tcpMgr(Index).RemoteHost, tvwChild, swTran
(swtranIndex).ip & swTran(swtranIndex).sw, swTran(swtranIndex).sw)
nodX.Parent.ForeColor = vbBlue
'nodX.Parent.Image = 2

Loop
tcpMgr(Index).Close
If Index > 0 Then 'cant unload master obj

Unload tcpMgr(Index)
End If
End Sub

```

```

Sub tcpMgr_Error(Index As Integer, ByVal Number As Integer, Description As String, ByVal
Scode As Long, ByVal Source As String, ByVal HelpFile As String, ByVal HelpContext As Long,
CancelDisplay As Boolean)

```

เอกสารนี้เป็นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Set nodX = tvSw.Nodes.Add(tcpMgr(Index).RemoteHost, twwChild, errCnt & Description,  
Description)  
If Index > 0 Then 'cant unload master obj  
    Unload tcpMgr(Index)  
End If  
  
End Sub
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 Software Compare เป็น Module ส่วนที่ใช้ในการเปรียบเทียบรายการ software ที่ได้ query มาจาก Agent แล้ว มาทำการเปรียบเทียบกับรายการ Software ที่ทางองค์กรมีอยู่ซึ่งเก็บในฐานข้อมูล ประกอบด้วย Module การทำงานดังต่อไปนี้

```

Dim rsIP As ADODB.Recordset
Dim rsWrkGrp As ADODB.Recordset
Dim rsSw As ADODB.Recordset
Const DB_PROVIDER = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source="
Dim Conn As ADODB.Connection
Dim dbFile As String
Dim svcCode As String
Dim UswArray(500) As String
Dim curNode As Integer

Private Sub cmdClose_Click()
Unload Me
End Sub

Private Sub cmdUninstall_Click()
If curNode > 0 Then
    svcCode = "UninstSw " & UswArray(curNode)
    tcpMgr.RemoteHost = Trim$(tvSw.Nodes(curNode).Parent.Text)
    listStat.AddItem "Connection Request To " & tcpMgr.RemoteHost & " For Uninstall. . ."
    tcpMgr.RemotePort = "3007"
    tcpMgr.Connect
End If
End Sub

Private Sub cmdWarn_Click()
Dim WarnMsg, DefMsg As String
If curNode > 1 Then
    DefMsg = "Please Uninstall " & tvSw.Nodes(curNode).Text & listSw.Text
    WarnMsg = InputBox("Enter Warning Message", "Software Monitoring Manager", DefMsg)
    If Trim$(WarnMsg) = "" Then
        Exit Sub
    End If
    svcCode = "Warn " & WarnMsg
    tcpMgr.RemoteHost = Trim$(tvSw.Nodes(curNode).Parent.Text)
    tcpMgr.RemotePort = "3007"
    tcpMgr.Connect
    listStat.AddItem "Connection Request To " & tcpMgr.RemoteHost & " For Warning. . ."
End If
End Sub

Private Sub tcpMgr_Connect()
listStat.AddItem "Connecting To " & tcpMgr.RemoteHost & " Complete. . ."
tcpMgr.SendData svcCode
listStat.AddItem "Send To " & tcpMgr.RemoteHost & " Complete. ... "
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub Form_Load()
'Load Data
dbFile = CurDir() & "\sw.mdb"
Set Conn = New ADODB.Connection
With Conn
.ConnectionString = DB_PROVIDER & dbFile
.Open
End With
Set rsWrkGrp = Conn.Execute("select grpname from wrkgrp")

listWrkgrp.Clear

' listSw.Clear
tvSw.Nodes.Clear
Do While Not rsWrkGrp.EOF
listWrkgrp.AddItem rsWrkGrp("grpname")
rsWrkGrp.MoveNext
Loop
End Sub
Sub GetIp(ByVal wg As String)
Dim sqlStr As String
sqlStr = "select ip from ip where wrkgrp = " & wg & """"
listIP.Clear
Set rsIP = Conn.Execute(sqlStr)
Do While Not rsIP.EOF
listIP.AddItem rsIP("ip")
rsIP.MoveNext
Loop
If listIP.ListCount > 0 Then
listIP.ListIndex = 0
GetSw listIP.List(0)
End If

End Sub
Sub GetIp2(ByVal wg As String)
Dim sqlStr As String
Dim iptmp As String
Dim i As Integer
sqlStr = "select * from swtran where swtran.ip in (select ip.ip from ip where ip.wrkgrp = " &
wg & """)"
listStat.Clear
Set rsIP = Conn.Execute(sqlStr)

tvSw.Nodes.Clear
Set nodX = tvSw.Nodes.Add(, , "R", wg)
tvSw.Nodes("R").Expanded = True

iptmp = ""
' On Error Resume Next
i = 0
If rsIP.EOF Then
Set nodX = tvSw.Nodes.Add("R", tvwChild, "x", "No Ip Address")
Else
Do While Not rsIP.EOF

```

```

If iptmp <> Trim(rsIP("ip")) Then 'new ip
    Set nodX = tvSw.Nodes.Add("R", tvwChild, Trim(rsIP("ip")), rsIP("ip"))
    iptmp = Trim(rsIP("ip"))
End If
rsIP.MoveNext
Loop
rsIP.MoveFirst
End If
Do While Not rsIP.EOF
    iptmp = Trim$(rsIP("ip")) & rsIP("sw")
    Set nodX = tvSw.Nodes.Add(Trim(rsIP("ip")), tvwChild, Trim(rsIP("ip")) & rsIP("sw"), rsIP
("sw"))
    If Not IsNull(rsIP("uninst")) Then
        UswArray(nodX.Index) = rsIP("uninst") ' Add uninstall string to array
    End If
    sqlStr = "select * from sw where swname = '" & rsIP("sw") & "'"
    Set rsChkSw = Conn.Execute(sqlStr)
    If rsChkSw.EOF Then
        nodX.ForeColor = vbRed
    Else
        If Not rsChkSw("swflag") Then
            nodX.ForeColor = vbRed
        Else
            nodX.ForeColor = vbBlue
        End If
    End If
    rsIP.MoveNext
Loop
'If listIP.ListCount > 0 Then
'listIP.ListIndex = 0
'GetSw listIP.List(0)
'End If
End Sub

```

```

Private Sub listIP_Click()
GetSw listIP.List(listIP.ListIndex)
End Sub

```

```

Private Sub listWrkgrp_Click()
GetIp2 listWrkgrp.List(listWrkgrp.ListIndex)
End Sub

```

```

Sub GetSw(ByVal ip As String)
Dim sqlStr As String
Dim i As Integer
Dim rsChkSw As ADODB.Recordset
sqlStr = "select * from swtran where ip = '" & ip & "'"
tvSw.Nodes.Clear
Set nodX = tvSw.Nodes.Add(, "R", ip)
tvSw.Nodes("R").Expanded = True
tvSw.Style = tvwTreelinesPlusMinusPictureText
' listSw.Clear
Set rsSw = Conn.Execute(sqlStr)

```

```

cmdWarn.Enabled = False
cmdUninstall.Enabled = False
Else
  cmdWarn.Enabled = True
  cmdUninstall.Enabled = True
End If
Do While Not rsSw.EOF
  i = i + 1
  Set nodX = tvSw.Nodes.Add("R", tvwChild, ip & rsSw("sw"), rsSw("Sw"))
  UswArray(i) = rsSw("uninst")
  sqlStr = "select * from sw where swname = " & rsSw("sw") & ""
  Set rsChkSw = Conn.Execute(sqlStr)
  If rsChkSw.EOF Then
    nodX.ForeColor = vbRed
  Else
    If Not rsChkSw("swflag") Then
      nodX.ForeColor = vbRed
    Else
      nodX.ForeColor = vbBlue
    End If
  End If
  rsSw.MoveNext
Loop
curNode = -1
' If listSw.ListCount > 0 Then
' listSw.ListIndex = 0
' End If
End Sub

Private Sub tcpMgr_Error(ByVal Number As Integer, Description As String, ByVal Scode As Long, ByVal Source As String, ByVal HelpFile As String, ByVal HelpContext As Long, CancelDisplay As Boolean)
  listStat.AddItem "Connection error to " & tcpMgr.RemoteHost & Description
  MsgBox "Can't Connect To " & tcpMgr.RemoteHost, vbOKOnly + vbExclamation, "Connecting Alert"
  tcpMgr.Close
End Sub

Private Sub tcpMgr_SendComplete()
  tcpMgr.Close
End Sub

Private Sub tvSw_NodeClick(ByVal Node As MSComctlLib.Node)
  If Node.Index > 1 Then
    curNode = Node.Index
  Else
    curNode = -1
  End If
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 Agent Management เป็นตัวอย่าง Module หนึ่งที่ใช้ในการ terminate การทำงานของ agent จาก Manager ประกอบด้วยการทำงานดังนี้

```
Dim Conn As ADODB.Connection
```

```
Dim rsIP As ADODB.Recordset
Private Type SW_TRAN
    ip As String * 15
    sw As String
    uninstall As String
End Type
```

```
Dim rsWrkGrp As ADODB.Recordset
Dim rsSw As ADODB.Recordset
Dim svcCode As String
Const SEP = " | "
Const DB_PROVIDER = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source="
Dim dbFile As String
Dim errCnt As Integer
Dim swTran() As SW_TRAN
Dim listImg As ListImage
Private Sub cmdClose_Click()
    MDIMain.MDIForm_Load
    Unload Me
End Sub
```

```
Private Sub btnMv_Click(Index As Integer)
    Dim i As Integer
    Select Case Index
    Case 0 ' add one ip to selected listbox
        If listIP1.ListIndex >= 0 Then
            listIP2.AddItem listIP1.List(listIP1.ListIndex)
        End If
    Case 1 ' Remove active ip from selected listbox
        If listIP2.ListIndex >= 0 Then
            listIP2.RemoveItem listIP2.ListIndex
        End If
    Case 2 ' add all ip to selected list
        For i = 0 To listIP1.ListCount - 1
            listIP2.AddItem listIP1.List(i)
        Next i
    Case 3 ' Remove all from select ip list
        listIP2.Clear
    End Select
End Sub
```

```
Private Sub cmdterm_Click()
    Dim i, Cnt As Integer
    On Error Resume Next
    Cnt = 0
    svcCode = "StopAgent"
```

```

If tcpMgr.Count > 1 Then
    For i = 1 To tcpMgr.Count - 1
        tcpMgr(i).Close
        Unload tcpMgr(i)
    Next i
End If
For i = 1 To listIP2.ListCount - 1
    Load tcpMgr(i)
Next i
For i = 0 To listIP2.ListCount - 1

    tcpMgr(i).RemoteHost = listIP2.List(i)
    tcpMgr(i).RemotePort = "3007"
    tcpMgr(i).Connect
    listStat.AddItem tcpMgr(i).RemoteHost & Space(20 - Len(tcpMgr(i).RemoteHost)) &
"Connecting . . ."
Next i
End Sub

Private Sub cboWrkGrp_Click()
    GetIp cboWrkgrp.List(cboWrkgrp.ListIndex)
End Sub

Private Sub Form_Load()
    'Load Data
    dbFile = CurDir() & "\sw.mdb"
    Set Conn = New ADODB.Connection
    errCnt = 0
    With Conn
        .ConnectionString = DB_PROVIDER & dbFile
        .Open
    End With

    Set rsWrkGrp = Conn.Execute("select grpName from wrkgrp")

    cboWrkgrp.Clear
    listIP1.Clear
    listIP2.Clear

    Do While Not rsWrkGrp.EOF
        cboWrkgrp.AddItem rsWrkGrp("grpname")
        rsWrkGrp.MoveNext
    Loop
    If cboWrkgrp.ListCount > 0 Then
        cboWrkgrp.ListIndex = 0
        GetIp cboWrkgrp.List(0)
    End If
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Sub GetIp(ByVal wg As String)
  Dim sqlStr As String
  sqlStr = "select ip from ip where wrkgrp = " & wg & ""
  listIP1.Clear
  Set rsIP = Conn.Execute(sqlStr)
  Do While Not rsIP.EOF
    listIP1.AddItem rsIP("ip")
    rsIP.MoveNext
  Loop
  If listIP1.ListCount > 0 Then
    listIP1.ListIndex = 0
  End If

End Sub

```

```

Private Sub tcpMgr_Connect(Index As Integer)
  tcpMgr(Index).SendData svcCode
  listStat.AddItem tcpMgr(Index).RemoteHost & Space(20 -
  Len(tcpMgr(Index).RemoteHost)) & "Sending . . ."
End Sub

```

```

Sub tcpMgr_Error(Index As Integer, ByVal Number As Integer, Description As String, ByVal
  Scode As Long, ByVal Source As String, ByVal HelpFile As String, ByVal HelpContext As Long,
  CancelDisplay As Boolean)
  errCnt = errCnt + 1

  If Index > 0 Then 'cant unload master obj
    Unload tcpMgr(Index)
  End If
End Sub

```

```

Private Sub tcpMgr_SendComplete(Index As Integer)
  listStat.AddItem tcpMgr(Index).RemoteHost & Space(20 - Len(tcpMgr(Index).RemoteHost)) &
  "Send Complete . . ."
  tcpMgr(Index).Close
End Sub

```

```

Private Sub Form_Load()
  'Load Data
  dbFile = CurDir() & "\sw.mdb"
  Set Conn = New ADODB.Connection
  errCnt = 0
  With Conn
    .ConnectionString = DB_PROVIDER & dbFile
    .Open
  End With

```

```

Set rsWrkGrp = Conn.Execute("select grpName from wrkgrp")

```

```

cboWrkGrp.Clear
listIP1.Clear
listIP2.Clear

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Do While Not rsWrkGrp.EOF
    cboWrkGrp.AddItem rsWrkGrp("grpname")
    rsWrkGrp.MoveNext
Loop
If cboWrkGrp.ListCount > 0 Then
    cboWrkGrp.ListIndex = 0
    GetIp cboWrkGrp.List(0)

End If

End Sub

Sub GetIp(ByVal wg As String)
    Dim sqlStr As String
    sqlStr = "select ip from ip where wrkgrp = '" & wg & "'"
    listIP1.Clear
    Set rsIP = Conn.Execute(sqlStr)
    Do While Not rsIP.EOF
        listIP1.AddItem rsIP("ip")
        rsIP.MoveNext
    Loop
    If listIP1.ListCount > 0 Then
        listIP1.ListIndex = 0
    End If

End Sub

Private Sub tcpMgr_Connect(Index As Integer)
    tcpMgr(Index).SendData svcCode
    listStat.AddItem tcpMgr(Index).RemoteHost & Space(20 -
Len(tcpMgr(Index).RemoteHost)) & "Sending . . ."
End Sub

Sub tcpMgr_Error(Index As Integer, ByVal Number As Integer, Description As String, ByVal
Scode As Long, ByVal Source As String, ByVal HelpFile As String, ByVal HelpContext As Long,
CancelDisplay As Boolean)
    errCnt = errCnt + 1

If Index > 0 Then 'cant unload master obj
    Unload tcpMgr(Index)
End If
End Sub

Private Sub tcpMgr_SendComplete(Index As Integer)
listStat.AddItem tcpMgr(Index).RemoteHost & Space(20 - Len(tcpMgr(Index).RemoteHost)) &
"Send Complete . . ."
tcpMgr(Index).Close
End Sub

```

2.4 IP Management เป็นตัวอย่างการทำงานในการ Setup ข้อมูลในฐานข้อมูล โดยเป็นตัวอย่าง Source Code การทำงานของการ setup ข้อมูล IP ในการ setup ข้อมูลอื่น ๆ ก็มีการทำงานในลักษณะเดียวกัน ดังตัวอย่างดังต่อไปนี้

```
Dim Conn As ADODB.Connection
Dim rsGrp As ADODB.Recordset
Dim rsIP As ADODB.Recordset
Dim sqlCmd As ADODB.Command
Const DB_PROVIDER = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source="
Const MAX_COL = 4
Dim dbFile As String
Dim Act As String
Private Sub cmdip_Click(Index As Integer)
```

```
If Index > 4 Then
'disable save btn
cmdIP(0).Enabled = True
cmdIP(1).Enabled = True
cmdIP(2).Enabled = True
cmdIP(3).Enabled = True
cmdIP(4).Enabled = True
cmdIP(5).Enabled = False
cmdIP(6).Enabled = False
cmdIP(5).Caption = "Save"
Else
cmdIP(0).Enabled = False
cmdIP(1).Enabled = False
cmdIP(2).Enabled = False
cmdIP(3).Enabled = False
cmdIP(4).Enabled = False
cmdIP(5).Enabled = True
cmdIP(6).Enabled = True
If Index = 0 Or Index = 1 Then
cmdIP(5).Caption = "Save"
End If
If Index = 2 Then
cmdIP(5).Caption = "Delete"
End If
If Index = 3 Then
cmdIP(5).Caption = "Search"
End If
End If
```

```
Select Case Index
Case 0: Act = "Add"
cmdClr
Case 1: Act = "Edit"
Case 2: Act = "Delete"
Case 3: Act = "Search"
cmdClr
Case 4:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MDIMain.MDIForm_Load
Unload Me
```

```
Case 5:
```

```
Select Case Act
Case "Add": cmdAdd
Case "Edit": cmdEdit
Case "Delete": cmdDel
Case "Search": cmdSearch
End Select
```

```
Case 6: gridip_Click
End Select
```

```
If Index <> 4 Then
gridIP.SetFocus
End If
```

```
End Sub
```

```
Sub cmdAdd()
```

```
Dim i As Integer
```

```
Dim valStr As String
```

```
Dim Grp As String
```

```
Grp = cboWrkgrp.List(cboWrkgrp.ListIndex)
```

```
On Error GoTo errTrap
```

```
valStr = "(" & txtIP(0) & "," & txtIP(1) & "," & Grp & "," & txtIP(3) & ")"
```

```
sqlCmd.CommandText = " Insert into ip values " & valStr
```

```
sqlCmd.Execute
```

```
With gridIP
```

```
.AddItem ""
```

```
.Row = .Rows - 1
```

```
'update grid
```

```
For i = 0 To MAX_COL - 1
```

```
.TextMatrix(.Row, i) = txtIP(i)
```

```
Next i
```

```
End With
```

```
errTrap:
```

```
If Err.Number <> 0 Then
```

```
MsgBox Err.Description, vbExclamation + vbOKOnly, "SWM Error"
```

```
End If
```

```
End Sub
```

```
Sub cmdEdit()
```

```
Dim valStr As String
```

```
Dim oldKey As String
```

```
Dim i As Integer
```

```
On Error GoTo errTrap
```

```
With gridIP
```

```
oldKey = .TextMatrix(.Row, .Col)
```

```
End With
```

```
valStr = "ip=" & txtIP(0) & "," & "name=" & txtIP(1) & ","
```

เอกสารนี้เป็นลิขสิทธิ์ของสำนักงานส่งเสริมการค้าในต่างประเทศ ณ นครเชียงใหม่
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
& "wrkgrp= " & txtIP(2) & ", " & "owner = " & txtIP(3) & "" _
& " where ip = " & oldKey & ""
```

```
sqlCmd.CommandText = "Update ip Set " & valStr
sqlCmd.Execute
```

```
With gridIP
For i = 0 To MAX_COL - 1
.TextMatrix(.Row, i) = txtIP(i)
Next i
End With
errTrap:
If Err.Number <> 0 Then
MsgBox Err.Description, vbExclamation + vbOKOnly, "SWM Error"
End If
End Sub
```

```
Sub cmdDel()
Dim valStr As String
Dim oldKey As String

On Error GoTo errTrap
With gridIP
oldKey = .TextMatrix(.Row, .Col)
End With
valStr = "ip = " & oldKey & ""
sqlCmd.CommandText = "delete from ip where " & valStr
sqlCmd.Execute
```

```
With gridIP
.RemoveItem .Row
.Col = 0
.Row = 1
gridip_Click
```

```
End With
errTrap:
If Err.Number <> 0 Then
MsgBox Err.Description, vbExclamation + vbOKOnly, "SWM Error"
End If
```

```
End Sub
Sub cmdSearch()
Dim schFld As Integer
Dim i As Integer
schFld = -1
For i = 0 To 3
If txtIP(i).Text <> "" And i <> 2 Then
schFld = i
Exit For
End If
Next i
If schFld >= 0 Then
With gridIP
```

```

For i = 1 To .Rows - 1
    .Row = i
    .Refresh
    If Trim$(UCase(txtIP(schFld).Text)) = Trim$(UCase(.TextMatrix(i, schFld))) Then
        gridip_Click
        Exit For
    End If
Next i
If i = .Rows Or schFld = 2 Then
    MsgBox "Data Not Found", vbExclamation + vbOKOnly, "SWM"
End If
End With
Else
    MsgBox "Please Enter Criteria", vbOKOnly + vbExclamation, "SWM Error"
End If

```

```

End Sub
Sub cmdClr()
    Dim i As Integer
    For i = 0 To MAX_COL - 1
        If i <> 2 Then
            txtIP(i).Text = ""
        End If
    Next i
End Sub

```

```

Private Sub cboWrkGrp_Click()
    RefWrkGrp
End Sub

```

```

Private Sub Form_Load()
    Dim icol As Integer
    'Load Data
    dbFile = CurDir() & "\sw.mdb"
    Set Conn = New ADODB.Connection

```

```

With Conn
    .ConnectionString = DB_PROVIDER & dbFile
    .CursorLocation = adUseClient
    .Mode = adModeReadWrite
    .Open
End With

```

```

Set sqlCmd = New ADODB.Command
Set rsGrp = New ADODB.Recordset
With sqlCmd
    .CommandText = "Select grpname From wrkgrp"
    .CommandType = adCmdText
    .ActiveConnection = Conn
End With

```

```

Set rsGrp = sqlCmd.Execute
With cboWrkGrp

```

```

.Clear
rsGrp.MoveFirst
While Not rsGrp.EOF
    cboWrkgrp.AddItem rsGrp.Fields(0)
    rsGrp.MoveNext
Wend
.ListIndex = 0
End With
End Sub
Sub RefWrkGrp()
sqlCmd.CommandText = "select * from ip where wrkgrp = " & cboWrkgrp.List
(cboWrkgrp.ListIndex) & ""
Set rsIP = sqlCmd.Execute

With gridIP
.Rows = 1
.Cols = rsIP.Fields.Count
For icol = 0 To rsIP.Fields.Count - 1
    .Col = icol
    .Text = rsIP.Fields(icol).Name
Next icol

While Not rsIP.EOF
.Rows = .Rows + 1
.Row = .Rows - 1
    For icol = 0 To rsIP.Fields.Count - 1
        .Col = icol
        .Text = rsIP(icol) & ""
    Next icol
    rsIP.MoveNext
Wend
End With
gridIP.FormatString = "<IP Number      |<Computer Name      |<Work Group
|<Computer Owner
'disable save btn
cmdIP(5).Enabled = False
cmdIP(6).Enabled = False
'default location
If gridIP.Rows > 1 Then
    gridIP.Col = 0
    gridIP.Row = 1
    gridIP_Click
End If
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Set Conn = Nothing
    Set sqlCmd = Nothing
    Set rsIP = Nothing
End Sub

```

```
Private Sub gridip_Click()  
With gridIP  
    For i = 0 To MAX_COL - 1  
        txtIP(i).Text = .TextMatrix(.Row, i)  
    Next i  
End With  
End Sub
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้