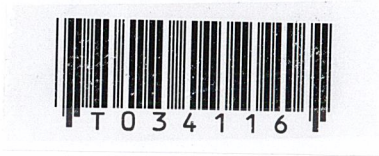


สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การพัฒนาเว็บแอปพลิเคชัน โดยใช้เอกทีฟเอ็กซ์เทคโนโลยี

WEB APPLICATION BY ACTIVEX



โดย
นางสาวเพียงเดือน สัตยารักษ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541

เอกสารนี้เป็นเอกสารทสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
เลขหม.....
เลขทะเบียน.....
วัน, เดือน, ปี- 5 ต.ค. 2542

การพัฒนาเว็บแอปพลิเคชัน โดยใช้แอคทีฟเอ็กซ์เทคโนโลยี

WEB APPLICATION BY ACTIVEX



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2541

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาเว็บแอปพลิเคชันโดยใช้เอกทีพีเอ็กซ์เทคโนโลยี

WEB APPLICATION BY ACTIVEX

ผู้จัดทำ

1.นางสาวเพียงเดือน สัตยารักษ์ รหัสประจำตัว 37014303



อาจารย์ที่ปรึกษา

(ดร. วรวัฒน์ ลิมโกกา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาเว็บแอปพลิเคชันโดยใช้แอททิฟเอ็กซ์เทคโนโลยี

นางสาวเพียงเดือน สัตยรักษ์ 37014303
ดร.วรวิวัฒน์ ลิ้มโกคา อาจารย์ที่ปรึกษา
ปีการศึกษา 2541

บทคัดย่อ

ในปัจจุบันการติดต่อสื่อสารข้อมูลผ่านระบบอินเทอร์เน็ตได้พัฒนาไปอย่างรวดเร็ว ทั้งในด้านความเร็วและความสวยงาม หลายบริษัทได้แนะนำเทคโนโลยีใหม่ ๆ ออกมาเพื่อใช้ในการปรับปรุงรูปแบบของเว็บแอปพลิเคชันให้ดีขึ้น แอททิฟเอ็กซ์ก็เป็นอีกเทคโนโลยีหนึ่งที่ถูกนำเสนอเพื่อให้การสร้างเว็บแอปพลิเคชัน โดยรูปแบบของแอททิฟเอ็กซ์จะเน้นทั้งความเร็วที่จะเสียเวลาโหลดคอนโทรลเพียงครั้งเดียว การพัฒนาเว็บแอปพลิเคชันก็เป็นไปโดยง่ายเพราะมีเครื่องมือมากมายที่สามารถใช้เพื่อสร้างแอททิฟเอ็กซ์ได้ รวมทั้งการนำไปใช้ที่สามารถนำไปใช้ที่อื่น ๆ ก็ได้ไม่จำเป็นต้องเป็นบนเว็บ

ปริญญานิพนธ์ฉบับนี้เป็นการศึกษาวิธีการพัฒนาเว็บแอปพลิเคชัน โดยใช้แอททิฟเอ็กซ์เทคโนโลยี โดยจะศึกษาถึงที่มา,เทคนิคการสร้าง,การนำไปใช้ และผลที่ได้รับ โดยจะทดลองสร้างคอนโทรลที่สามารถเรียกใช้ฐานข้อมูลผ่านทางอินเทอร์เน็ต โดยที่ฐานข้อมูลที่ถูกเรียกใช้ไม่จำเป็นต้องเป็นฐานข้อมูลชนิดเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Web Application by ActiveX

Peangduen Satayarak

Dr.Worawat Limpoka Adviser

Abstract

Recently, communication pass internet system was developed rapidly both speed and graphic. Many company introduced new technology for developed web application form. ActiveX was technology that presented for build web application. ActiveX technology was fast by only one download control and easy to used to develop because have variant tool. Not only on web that control can used. It can use on other place such as Windows application.

This thesis was studied about how to develop web application by ActiveX technology. It was studied about why it come, how to develop and result form it by build control that can call database pass internet system. This database can be from variant program.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ หากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลาย ๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงก็คือ อาจารย์ วรวัฒน์ ลิ้มโกคา อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก

และต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดาที่ช่วยเลี้ยงดูมาตลอด ขอกราบขอบพระคุณมา ณ ที่นี้

อีกทั้งเพื่อน ๆ ทุกคนที่คอยช่วยเหลือและเป็นกำลังใจให้เสมอ ก็ขอขอบพระคุณท่านทั้งหลายมา ณ ที่นี้ด้วย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพ	V
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎี และการออกแบบ	2
2.1 ส่วนประกอบของเทคโนโลยี ActiveX	2
2.2 พัฒนาการของ ActiveX Control	7
2.3 ความปลอดภัยของ ActiveX บนอินเทอร์เน็ต	9
2.4 ความรู้เบื้องต้นเกี่ยวกับ โพรโทคอล TCP/IP	11
2.5 สถาปัตยกรรม COM (Component Object Model)	16
2.6 การออกแบบอินเทอร์เฟซและคอมโพเนนท์ที่เป็น COM (Component Object Model)	19
2.7 ODBC (Open Database Connectivity)	27
2.8 การเชื่อมฐานข้อมูลแบบออบเจกต์เข้ากับโปรแกรม	30
2.9 การออกแบบการทำงานของโปรแกรม Database on Internet	32
2.10 การสร้าง ActiveX ด้วย VBCCE (Visual Basic Custom Control Edition)	34
2.11 การใช้ VBScript ควบคุมการทำงานของ ActiveX Control	38
2.12 การนำ Winsock Control ไปใช้เพื่อการติดต่อระหว่างไคลเอนต์/เซิร์ฟเวอร์	41
2.13 การ ActiveX Control Pad เพื่อช่วยในการทำเว็บเพจจาก ActiveX Control	43
2.14 ฐานข้อมูลเบื้องต้น	49
บทที่ 3 การพัฒนาแอปพลิเคชัน	51
3.1 จุดประสงค์ของแอปพลิเคชัน	51
3.2 การพัฒนาแอปพลิเคชัน	51
3.3 การทำงานของโปรแกรม Database on Internet ส่วนเซิร์ฟเวอร์	55
3.4 การทำงานของโปรแกรม Database on Internet ส่วนไคลเอนต์	59
บทที่ 4 การใช้งานโปรแกรม	60
บทที่ 5 บทสรุป และวิจารณ์	64
ภาคผนวก ก ต้นฉบับโปรแกรมในส่วนเซิร์ฟเวอร์	66
ภาคผนวก ข ต้นฉบับโปรแกรมในส่วนไคลเอนต์	88
บรรณานุกรม	125

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

หน้า

รูปที่ 1 แสดงส่วนต่าง ๆ ของเทคโนโลยี ActiveX	2
รูปที่ 2 แสดงเลขเอร์ของ TCP/IP	11
รูปที่ 3 หมายเลข IP แอดเดรส	13
รูปที่ 4 แสดงคลาส,จำนวนเครือข่าย และจำนวนโฮสในแต่ละคลาส	13
รูปที่ 5 ตัวอย่าง IP แอดเดรส	13
รูปที่ 6 ความสัมพันธ์ระหว่างชุดโปรโตคอล	14
รูปที่ 7 Interface ของ COM object	16
รูปที่ 8 แสดงสถาปัตยกรรมของ COM ที่สามารถแบ่งการทำงานตามระดับการใช้งานได้	17
รูปที่ 9 แสดงการทำงานของ COM	18
รูปที่ 10 ฐานข้อมูลตามแบบธรรมดาเดิม	28
รูปที่ 11 เปรียบเทียบลักษณะรวมศูนย์กับลักษณะไคลเอนต์เซิร์ฟเวอร์	29
รูปที่ 12 ODBC Data Resource Administrator	30
รูปที่ 13 Creat New Data Source	31
รูปที่ 14 Oracle ODBC Driver Setup	32
รูปที่ 15 แสดงการส่งข้อมูลผ่าน Winsock คอนโทรล	34
รูปที่ 16 เริ่ม Visual Basic Custom Control Edition	34
รูปที่ 17 แสดงการวาง object ลงบนฟอร์ม	35
รูปที่ 18 เพิ่ม Procedure ให้คอนโทรล	36
รูปที่ 19 Dialog สำหรับการสร้าง ActiveX Control จากโค้ดที่ได้เขียนมา	37
รูปที่ 20 เริ่ม ActiveX Control Pad	43
รูปที่ 21 ActiveX Control Pad Layout	44
รูปที่ 22 แสดงการวาง object ลงบน Layout	45
รูปที่ 23 แสดงการนำ Layout ไปใส่ในเพจ	45
รูปที่ 24 แสดงเพจที่ได้	46
รูปที่ 25 แสดงการเลือก object บน Layout เพื่อเขียนสคริปต์	47
รูปที่ 26 แสดง Script Wizard เป็นส่วนที่ช่วยในการเขียนสคริปต์	47
รูปที่ 27 แสดงความสามารถของคอนโทรลเมื่อวางลงบนเว็บเพจ	48
รูปที่ 28 แสดงการทำงานของโปรแกรมในส่วนเซิร์ฟเวอร์	49
รูปที่ 29 แสดงการทำงานของโปรแกรมในการเปิดปิดฐานข้อมูล	50
รูปที่ 30 แสดงการทำงานของโปรแกรมในส่วนที่จัดการข้อมูลตาม Request	51
รูปที่ 31 แสดงการทำงานของโปรแกรมในส่วนไคลเอนต์	52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น ผู้จัดทำมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 32 หน้าจอเมื่อเรียกใช้โปรแกรม Server.exe	53
รูปที่ 33 เลือกฐานข้อมูลที่ต้องการ	53
รูปที่ 34 แสดงรายชื่อฐานข้อมูลที่ได้เปิดไว้	54
รูปที่ 35 ใสรหัสก่อนถึงจะเข้าได้	54
รูปที่ 36 เมื่อเรียกใช้บริการก็จะมีรายชื่อของตารางข้อมูลที่สามารถเรียกดูได้มาให้เลือก	54
รูปที่ 37 แสดงการเลือกรายการจากตารางที่เลือกในขั้นต้น	55
รูปที่ 38 ข้อมูลที่ได้จากตาราง	55
รูปที่ 39 แสดงการค้นหาข้อมูลจากค่าของข้อมูลในรายการหนึ่ง	55
รูปที่ 40 แสดงข้อมูลจากอีกตารางหนึ่งที่มีความสัมพันธ์กับตารางแรก	56



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

อินเทอร์เน็ตเป็นเครือข่ายที่ก้าวเข้ามามีบทบาทในพื้นฐานชีวิตของคนทั่วไปมากขึ้นทุกที มันเป็นส่วนสำคัญที่ใช้ในการเผยแพร่ข้อมูลข่าวสารของแต่ละองค์กรให้คนภายนอกรู้ และได้พัฒนารูปแบบของข้อมูลข่าวสารให้ดูหน้าสนใจยิ่งขึ้นด้วยภาพเคลื่อนไหว และในปัจจุบันอินเทอร์เน็ตได้มีการพัฒนาขึ้นไปอีกขั้น จากที่ผู้รับเคยรับได้ข้อมูลและข่าวสารเพียงอย่างเดียวเป็นผู้รับสามารถส่งข้อมูลและข่าวสารต่าง ๆ กลับมาตอบโต้กับทางผู้ส่งได้อีกด้วย ทำให้ผู้ส่งสามารถให้ผู้รับแก้ไขข้อมูลและข่าวสารต่างให้ทันสมัยได้อีกด้วย

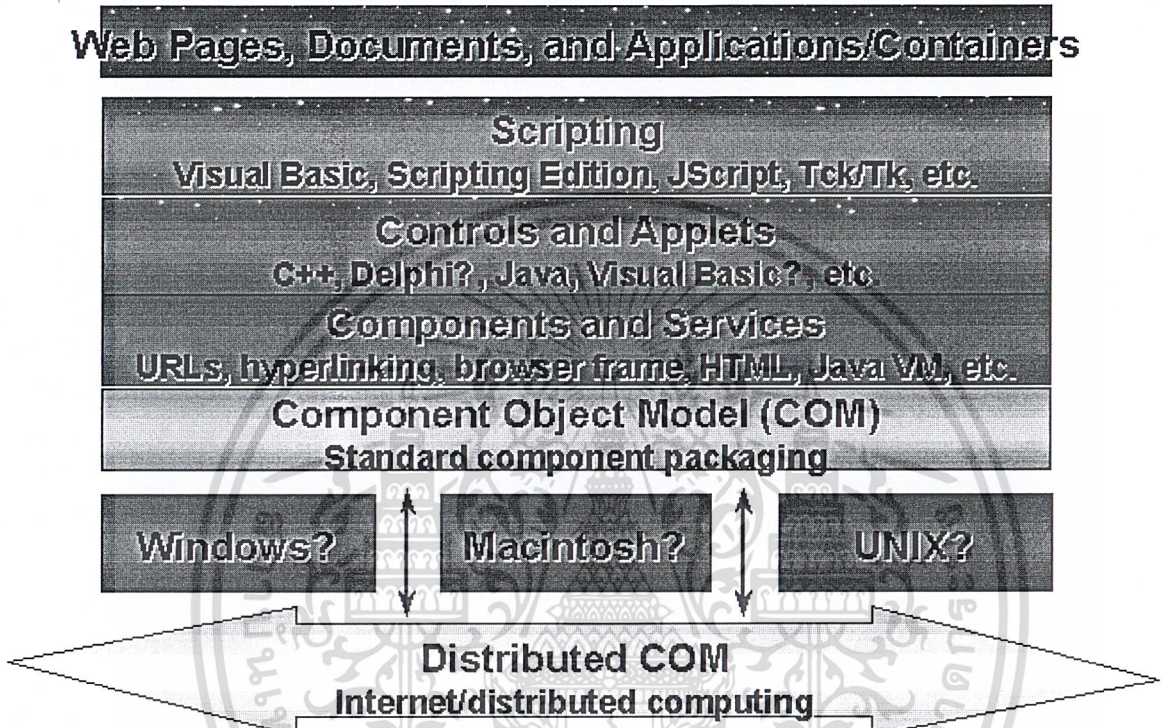
และเพื่อให้อินเทอร์เน็ตมีความสามารถมากขึ้นการเขียนเพจโดยใช้ภาษา HTML เริ่มไม่สามารถพัฒนาให้เพิ่มความสามารถของเพจได้มากนัก จึงได้มีหลายแนวทางที่ถูกพัฒนาเพื่อมาใช้ในการนี้ เช่นการพัฒนาภาษาจาวามาใช้ และทางไมโครซอฟท์ก็ได้พัฒนามาตรฐานจากบนวินโดว์มาใช้บนอินเทอร์เน็ตอันได้แก่ ActiveX

เนื่องจากที่เทคโนโลยี ActiveX นั้นมีรากฐานมาจากเทคโนโลยี OLE ของวินโดว์ ทำให้ ActiveX มีคุณสมบัติของการ Linking และ Embedding เช่นเดียวกับ OLE แต่ได้เพิ่มคุณสมบัติของโพรโตคอลที่ OLE ไม่มีเพื่อให้สามารถใช้ได้บนระบบเน็ตเวิร์ค และด้วยคุณสมบัติเหล่านี้ทำให้เพจที่ใช้เทคโนโลยี ActiveX มีลักษณะการทำงาน และคุณสมบัติต่างๆ ที่คล้ายกับ แอปพลิเคชันบนวินโดว์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎี

2.1 ส่วนประกอบของเทคโนโลยี ActiveX



รูปที่ 1 แสดงส่วนต่างๆ ของเทคโนโลยี ActiveX

ActiveX Scripting

Netscape ได้เพิ่มลักษณะเด่นที่มีประโยชน์มากให้แก่ Navigator เรียกว่าจาวาสคริปต์โดยปกติแล้วเว็บเบราว์เซอร์จะมีตัวตีความที่ทำหน้าที่เป็นแปลสำหรับสคริปต์ชนิดนี้ บริการใช้สคริปต์ ActiveX ที่มีสำหรับ Microsoft Internet Explorer 3.0 นั้นก็คล้ายคลึงกันแต่จะเป็นบริการที่ธรรมดากว่า กล่าวคือทำให้สามารถเพิ่มด้านสคริปต์และความสามารถในการลักษณะเดียวกับ OLE ให้กับโปรแกรมได้ สคริปต์ของ ActiveX จะจัดหาแพลตฟอร์มสำหรับผู้พัฒนาภาษา แต่สำหรับ ตัวภาษาสคริปต์, ไวยากรณ์ และ รูปแบบการทำงานนั้นจะหลากหลายแตกต่างกันไป ขึ้นอยู่กับการออกแบบของนักพัฒนา

ActiveX Scripting มีองค์ประกอบ 2 ชนิด คือ

1 ActiveX Scripting Hosts

ActiveX scripting host จะจัดหาแพลตฟอร์มไว้เพื่อให้ ActiveX Scripting Engines ทำงานจะเห็นได้ว่าเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ทางการค้า ActiveX scripting host หลักที่สำคัญคือ Microsoft Internet Explorer อย่างไรก็ตามยังมี scripting host อื่นที่มีศักยภาพภายใต้ ActiveX อีกเช่นกัน ได้แก่

- Web browser อื่นๆ (เช่น Netscape)
- Internet authoring tools
- Web servers (server-based scripting)

2 ActiveX Scripting Engines

โดยพื้นฐานแล้ว ActiveX scripting engine ก็คือภาษาที่สามารถปฏิบัติการบน ActiveX scripting host ได้ ActiveX scripting engine ชนิดแรกได้แก่ VBScript (เป็นส่วนหนึ่งของ Visual Basic 4.0 ซึ่งเป็นที่นิยมมาก) อย่างไรก็ตามยังมี environment อื่นๆ อีกเช่น Perl, Lisp, Delphi, Scheme

ActiveX components

เนื่องจากได้มีการปรับปรุงเทคโนโลยีในการผลิตเบราว์เซอร์ ให้ดีขึ้น ทำให้สามารถใช้ ActiveX คอมโพเนนต์ซึ่งเป็นเทคโนโลยีที่มีพื้นฐานมาจาก Microsoft's Component Object Model (COM) ในการเพิ่มแพลตฟอร์มต่างๆ ให้กับเบราว์เซอร์ได้อย่างอิสระ ขอบข่ายของ ActiveX คอมโพเนนต์ จะเป็นได้ตั้งแต่ fancy controls อย่างเช่น spinner และ slider ไปจนถึง nonvisual components จึงทำให้เกิดการเข้าถึงข้อมูลและเพิ่มประสิทธิภาพให้กับการส่งไปรษณีย์อิเล็กทรอนิกส์ (E-mail) โดยคอมโพเนนต์เหล่านี้จะทำให้เพจใน Internet Explorer มีทั้งความน่าสนใจและมีฟังก์ชันใช้งานหลากหลาย แต่จะไม่สามารถใช้ได้ในระบบที่ไม่สนับสนุน ActiveX อย่างเช่น Netscape Navigator

ActiveX คอมโพเนนต์ถูกเพิ่มเข้าไปในเว็บเพจโดยใช้ <OBJECT> tag ซึ่งเป็น การการกำกับลักษณะเฉพาะของคอมโพเนนต์ให้กับเบราว์เซอร์ โค้ดต่อไปนี้ ใช้ <OBJECT> tag เพื่อเพิ่มคอนโทรลลาเบล (label control) ที่เป็น ActiveX ให้กับเว็บเพจ

```
<OBJECT ID="IeLabel3" WIDTH=148 HEIGHT=40
CLASSID="CLSID:99B42120-6EC7-11CF-A6C7-00AA0A47DD2">
CODEBASE="http://www.microsoft.com/activex/controls/ielabel.ocx">
<PARAM NAME="_ExtentX" VALUE="3916">
<PARAM NAME="_ExtentY" VALUE="1058">
<PARAM NAME="Caption" VALUE=" Label3">
<PARAM NAME="Angel" VALUE="0">
<PARAM NAME="Alignment" VALUE="4">
<PARAM NAME="Mode" VALUE="1">
<PARAM NAME="FillStyle" VALUE="0">
<PARAM NAME="ForeColor" VALUE="#000000">
<PARAM NAME="BackColor" VALUE="#C0C0C0">
<PARAM NAME="FontName" VALUE="Arial">
<PARAM NAME="FontSize" VALUE="12">
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในการศึกษาวิจัยเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ผู้เขียนคือผู้รับผิดชอบไปทั้งหมดของเอกสารทุกครั้งที่มีการนำไปใช้

```

<PARAM NAME="FontItalic" VALUE="12">
<PARAM NAME="FontBold" VALUE="0">
<PARAM NAME="FontUndeline" VALUE="0">
<PARAM NAME="FontStrikeout" VALUE="0">
<PARAM NAME="TopPoints" VALUE="0">
<PARAM NAME="BotPoints" VALUE="0">

```

</OBJECT>

โดยภายใน <OBJECT> tag จะประกอบไปด้วยส่วนที่กำหนดคุณสมบัติต่างๆ ที่ทำให้พอทราบได้ว่า ActiveX คอมโพเนนท์จะแสดงบนเพจได้อย่างไร ID attribute เป็นส่วนที่ใช้ระบุชื่อของ ActiveX คอนโทรลซึ่งในการกำหนดชื่อแต่ละครั้งจะทำให้สามารถเข้าถึงคอนโทรลดังกล่าวได้จากโคดสคริปต์ และสามารถเรียกใช้คุณสมบัติ (properties), เหตุการณ์ (events) และการทำงาน (methods) ของ control เพื่อแสดงการใช้งานบนเพจได้ ลักษณะของ attribute แบ่งออกเป็น 2 ประเภท คือ CLSID และ CODEBASE

1. CLSID attribute

คือจำนวนที่เป็นตัวอักษร, ตัวเลข, เครื่องหมาย และสัญลักษณ์ที่เรียงกัน (alphanumeric serial number) เพื่อแทนลักษณะเฉพาะของ ActiveX คอมโพเนนท์ จำนวนเหล่านี้เรียกว่า "Globally Unique Identifier" หรือ "GUID" ใน Internet Explorer ใช้ GUID เพื่อหาตำแหน่งเก็บ ActiveX คอมโพเนนท์ พร้อมทั้งสร้างไว้บนเพจและการเรียกใช้ GUID บนระบบปฏิบัติการสามารถทำโดยผ่านส่วนบันทึก (registry) หรือฐานข้อมูลส่วนกลาง (centralized database) ที่จะรับผิดชอบในการเก็บข้อมูลเกี่ยวกับ software object ที่ถูกใช้โดยแอปพลิเคชันต่าง ๆ เมื่อ Internet Explorer ได้รับ <OBJECT> tag แล้วจะไปในส่วนบันทึกเพื่อหา CLSID attribute ที่เข้าคู่กันกับ GUID ที่ได้รับมา เมื่อจัดเก็บ GUID แล้ว ส่วนบันทึกจะทำการจัดเตรียมส่วนข้อมูลเพิ่มเติมที่ใช้เก็บเพิ่มข้อมูลไว้สำหรับ ActiveX คอนโทรล

2. CODEBASE attribute

ถูกใช้ใน Internet Explorer เพื่อเป็นตำแหน่งในการจัดเก็บ ActiveX คอนโทรลบนเครื่องผู้ให้บริการ (server machine) ในกรณีที่ไม่มี ActiveX คอนโทรลบนเครื่องผู้ใช้บริการ (client machine) ซึ่งเพิ่มข้อมูลเหล่านั้นจะถูกดาวน์โหลด โดยผ่าน CODEBASE attribute แล้ว ActiveX คอมโพเนนท์จะถูกติดตั้งลงบนเครื่องไคลเอนต์ โดยจะ install ทุกครั้งที่ IE ต้องการใช้งานคอนโทรล การใช้ <OBJECT> tag ไม่ถูกจำกัดไว้เพียงเป็นส่วนควบคุมเท่านั้น แต่ยังสามารถเข้าถึง ActiveX คอมโพเนนท์ใดใดหรือใช้กระตุ้น (activate) การทำงานของ ActiveX คอมโพเนนท์ ซึ่งรวมถึงคอมโพเนนท์ที่สร้างด้วยโปรแกรมภาษาอื่นๆ เช่น Visual Basic, J++ , C++ และ Microsoft FoxPro ได้ด้วย ดังนั้นการเขียน ActiveX คอมโพเนนท์ขึ้นมาเองจะทำให้สามารถเพิ่มฟังก์ชันการใช้งานให้กับเครื่องไคลเอนต์ และการดาวน์โหลดคอมโพเนนท์ต่าง ๆ ให้กับบราวเซอร์

นอกจากนี้ในส่วน <OBJECT> tag ยังมี <PARAM> tag ที่จัดเตรียมข้อมูลเพื่อใช้ในการสร้างคอมโพเนนท์โดยจะประกอบไปด้วยส่วนของ NAME และ VALUE attribute เพื่อที่จะผ่านค่าคุณสมบัติเริ่มต้น (initial properties) ให้กับ ActiveX คอมโพเนนท์ที่สร้างเป็นครั้งแรกในเว็บเพจ และหลังจากที่กำหนดค่าเริ่มต้นแล้ว คุณสมบัตินี้จะสามารถเปลี่ยนได้ในขณะที่โปรแกรมทำงาน โดยอาศัยการเรียกใช้ของสคริปต์คำสั่งทุกครั้งที่มีการนำไปใช้

ActiveX documents

ActiveX documents เป็นชุดเครื่องมือ (tools) ที่ใช้ในการสร้าง dynamic content ซึ่งจะสนับสนุนการสร้าง ActiveX คอนโทรลได้ดีพอๆ กับ developer ชนิดอื่น เช่น Visual Basic 5.0 และ ActiveX documents เป็น software object ที่จะถูกดาวน์โหลด และทำงานภายใน ActiveX container อย่างเช่น "Internet Explorer" ได้ ดังนั้น ActiveX documents จึงนับว่าเป็นเครื่องมือที่ผู้เร่ร่อนนักพัฒนาได้อย่างมากในการสร้าง application ทางอินเทอร์เน็ต โดย ActiveX documents จะกำหนดส่วนที่ใช้ในการเข้าถึง (access) สมบัติ (properties), ฟังก์ชัน (methods) ต่างๆ ที่สร้างขึ้นภายใน developer ไว้ในรูปแบบที่สามารถดาวน์โหลดได้ ผลที่ได้ก็คือทำให้ผู้ใช้สามารถดูและแก้ไข document ที่ไม่ใช่ HTML ผ่านทางบราวเซอร์ได้

ActiveX Control

ActiveX คอนโทรลได้ถูกแนะนำเพื่อการรวมสองส่วนที่แยกกันวิวัฒนาการของเทคโนโลยีคอมพิวเตอร์ Custom Control กับส่วนที่คิดใหม่ของโอเคียพื้นฐานเกี่ยวกับ OLE และ OOP

คอนโทรลตัวใหม่นี้เป็นคอนโทรล OLE ที่อยู่บนพื้นฐานของ DCOM (Distributed Component Object Model) ตัวแรก ดังที่เคยกล่าวมาแล้วว่าการติดต่อพื้นฐานของ OLE อยู่บนพื้นฐานของ COM เพราะฉะนั้น ActiveX คือคอนโทรล OLE อย่างแท้จริงตัวแรก

ใน DCOM นักเขียนโปรแกรมไม่ต้องเขียนโปรแกรมสำหรับคอมพิวเตอร์ที่แยกกันในแต่ละตัวแต่ละเขียนโปรแกรมสำหรับสภาพแวดล้อมหรือเน็ตเวิร์กที่คอมพิวเตอร์นั้นทำงานอยู่ ในระบบโมเดล COM แบบเก่า นักเขียนโปรแกรมต้องการเพียงติดตามข้อบังคับว่าทำอะไรคอมพิวเตอร์กับผู้ใช้จึงได้ตอบได้ กับ DCOM นักเขียนโปรแกรมจะต้องจัดหาข้อบังคับที่คอมพิวเตอร์จะได้ออกกับเครื่องอื่นในระบบอินเทอร์เน็ตหรืออินทราเน็ต

การใช้ ActiveX คอนโทรลบนอินเทอร์เน็ต

เนื่องจาก content ต่างๆ ในอินเทอร์เน็ตกลายเป็น dynamic มากขึ้น จึงทำให้เว็บเบราว์เซอร์ต้องจัดการกับงานต่างๆ ของ content เหล่านั้นมากขึ้น Netscape นำ Navigator 2.0 ซึ่งสนับสนุนจาวาแอฟเฟลตออกสู่ตลาด เพื่อเป็นการเผยแพร่จาวาไปในตัว เมื่อใช้จาวาจะทำให้เกิดพัฒนาเว็บมีชุดเครื่องมือที่ใช้ในการโต้ตอบกับเครื่องของผู้ใช้หรือลูกค้า หลังจากนั้น Microsoft จึงออก Microsoft Internet Explorer 3.0 ซึ่งสนับสนุนการทำงานของจาวา พร้อมทั้งตัวเลือกใหม่ที่ถูกเรียกว่า ActiveX คอนโทรลซึ่งพัฒนามาจาก OLE 2.0 ActiveX คอนโทรลใช้กับเว็บเพจในลักษณะเดียวกับที่ VBX และ OCX ใช้กับวินโดวส์แอปพลิเคชัน ในอนาคตคุณอาจจะใช้ ActiveX คอนโทรลแทนที่ตำแหน่งที่ใส่จาวาแอฟเฟลตอยู่ในปัจจุบัน เนื่องจาก ActiveX คอนโทรลมีข้อได้เปรียบเหนือกว่าจาวาแอฟเฟลตดังต่อไปนี้

1. สร้างโดยใช้เครื่องมือที่รู้จัก

ActiveX คอนโทรลไม่ได้ถูกสร้างขึ้นโดยใช้ Microsoft Foundation Classes แต่ Microsoft สร้าง ActiveX Template Library (ATL) ขึ้นมาเพื่อช่วยนักพัฒนาในการสร้าง ActiveX คอนโทรลในขณะที่การใช้จาวา

จะต้องเรียนรู้ภาษาใหม่ ถึงแม้ว่าจาวาและ C++ จะมีไวยากรณ์ที่คล้ายคลึงกันก็ตาม แต่เมื่อใช้จาวาจะต้องเรียนรู้ลำดับขั้นการถ่ายถอดความสัมพันธ์ใหม่ทั้งหมด อย่างไรก็ตามการใช้ C++ และการสร้าง OCX จะทำให้รู้ถึงหลักการพื้นฐานของจาวา

2. ง่ายต่อการรวมเข้ากับแอปพลิเคชันอื่นๆ

เนื่องจาก ActiveX คอนโทรลมีวิวัฒนาการมาจาก OLE 2.0 จึงทำให้มีความสามารถในการโต้ตอบกับแอปพลิเคชันอื่นๆ (ตัวอย่างเช่น word processor, spread-sheet และ presentations) ซึ่งต่างกับจาวาที่โดยแท้จริงแล้วยังไม่มีความสามารถในการโต้ตอบกับแอปพลิเคชันอื่นๆที่มีอยู่ ถึงแม้ว่าจาวาจะสามารถสร้างแอปพลิเคชันที่ทำงานได้เพียงลำพังได้ก็ตาม

3. ง่ายต่อการใช้งานใน Containers

Internet ActiveX Control Pack แสดงให้เห็นว่าสามารถใช้งาน ActiveX ใน development environments อื่นๆ ได้ เช่น Visual C++, Delphi, Visual Basic และ MS-Access เนื่องจาก Microsoft ใช้ OLE 2.0 เป็นต้นแบบในการสร้าง ActiveX คอนโทรลต่างจากจาวาแอปเพลตที่ถูกสร้างขึ้นมาเพื่อปฏิบัติงานบนเวบบราวเซอร์เท่านั้น

อย่างไรก็ตามยังมีปัญหาบางประการในการใช้ ActiveX คอนโทรลกล่าวคือ

1. ยังไม่ถูกสนับสนุนใน Netscape

ถ้า Netscape ยอมรับสนับสนุน ActiveX คอนโทรลจะทำให้เป็นบราวเซอร์ที่ได้เปรียบทางด้านตลาด แต่เนื่องจาก Microsoft ถือว่าเป็นบริษัทคู่แข่งที่สำคัญ และ Netscape ได้สนับสนุนจาวาอยู่แล้ว ทำให้ Netscape ยังไม่ยอมที่จะ สนับสนุน ActiveX Controls ดังนั้นจึงมีบริษัทอื่นที่ผลิต plug-in (สามารถดาวน์โหลดได้จาก www.microsoft.com) ขึ้นมาเพิ่มการทำงานของเว็บเบราว์เซอร์เพื่อให้สนับสนุน ActiveX ได้ ทำให้ปัญหาดังกล่าวไม่มีผลกระทบต่อ ActiveX มากนัก

2. ยังไม่ถูกสนับสนุนบนแพลตฟอร์ม UNIX

อินเทอร์เน็ตเจเรจิวเติบโตจากแนวความคิดที่ต้องการแพลตฟอร์มสนับสนุนได้โดยไม่จำกัด จึงทำให้แอปพลิเคชันที่สำคัญส่วนใหญ่ของอินเทอร์เน็ตสามารถทำงานได้บนแพลตฟอร์มที่หลากหลาย ในปัจจุบันเครื่องที่ใช้ระบบ UNIX เป็นที่ยอมรับและมีผู้ใช้งานจำนวนมาก แต่ที่ผ่านมารบริษัท Microsoft ยังไม่ให้ความสำคัญในการขยายเทคโนโลยีของตนลงบนแพลตฟอร์ม UNIX ดังนั้นการที่ ActiveX คอนโทรลไม่ถูกสนับสนุน โดย UNIX ทำให้เกิดปัญหาด้านการยอมรับของตลาด เป็นผลให้บริษัท Microsoft ออกมารับรองว่าในอนาคต ActiveX คอนโทรลจะถูกสนับสนุน โดย UNIX environment

ActiveX คอนโทรลจะถูกใช้ในการทำ data binding ซึ่งเป็นการเชื่อมการทำงานระหว่างคุณสมบัติ (properties) ของคอนโทรลกับ field ข้อมูลที่จะใช้บนฐานข้อมูล หรือเป็นการรวมคอนโทรลในส่วน DataSource เข้ากับ DataField นอกจากนี้ IE 4.0 จะใช้ ActiveX คอนโทรลในการสนับสนุนลักษณะ multimedia ต่างๆ ที่จะทำได้งานที่น่าพอใจมากที่สุด เพราะฉะนั้นผู้พัฒนา Internet Explorer 4.0 มีความตั้งใจให้ multimedia Controls สร้างได้รวดเร็ว, สามารถใช้ รูปและระบบเสียงช่วยในการสร้างคอมโพเนนท์ และใช้เวลาในการดาวน์โหลดไม่นาน โดยเฉพาะอย่างยิ่งการใช้งานร่วมกับ multimedia Web content

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 พัฒนาการของ ActiveX Control

ActiveX Control มีการพัฒนามาจากเทคโนโลยี OLE (Object Linking and Embedding) มันทำให้ผู้ใช้สามารถใช้ข้อมูลจากแหล่งต่าง ๆ ร่วมกันได้ผ่านอินเทอร์เน็ตหรืออินทราเน็ตขององค์กร ActiveX คือเทคโนโลยีทั้งหลายของกระบวนการที่ทำงานผ่านสภาพแวดล้อมแบบกระจาย (distributed environment) เช่นในอินเทอร์เน็ต เป็นต้น

ด้วย OLE ข้อมูลจากแอปพลิเคชันอย่างใน Excel หรือ Access สามารถนำไปใช้เป็นส่วนหนึ่งของเอกสารใน Word ได้ จากเดิมที่ต้องทำงานในสามแอปพลิเคชันถึงจะประมวลผลข้อมูลมารายงานได้

OLE (Object Linking and Embedding)

การที่จะเข้าใจในความหมายของ OLE เราต้องลองทำความเข้าใจแต่ละคำ

Object - ทุก "สิ่ง" ที่อยู่บนระบบคอมพิวเตอร์รวมทั้งคอนโซล, แอปพลิเคชัน, ไลบรารีและอื่น ๆ

Linking - บางครั้งการเชื่อมจะถูกสร้างจากสอง Object ในระบบซึ่งจะต้องมีการแบ่งปันข้อมูลและทรัพยากรต่าง ๆ ซึ่งกันและกัน เช่นถ้ามีรายชื่อของส่งจดหมายกับฟอร์มของจดหมาย ก็จะต้องสร้างการเชื่อมโยงให้ฟอร์มของจดหมายได้ส่งไปยังทุกคนที่มีรายชื่อ

Embedding - เมื่อได้สร้างเอกสารในแอปพลิเคชันหนึ่ง ๆ บางครั้งแอปพลิเคชันนั้นไม่เหมาะกับการทำงานอื่น ๆ กับเอกสารนั้น จึงต้องนำเอกสารจากแอปพลิเคชันนั้นไปฝังไว้ในแอปพลิเคชันอื่น ๆ เช่นการสร้างกราฟด้วย Excel แล้วจึงนำกราฟนั้นไปวางบนเอกสารใน Word เพื่อแสดงผลออกมาพร้อมกับส่วนอื่นในเอกสาร

OLE ก็คือการสามารถที่นำสิ่งที่อยู่บนระบบคอมพิวเตอร์ที่สร้างบนแอปพลิเคชันหนึ่งไปเชื่อมหรือวางรวมไว้ที่แอปพลิเคชันอื่น ๆ นั่นเอง

ประวัติของ OLE

OOP (Object Oriented Programming) ได้ถูกพัฒนาเมื่อประมาณกลางศตวรรษที่ผ่านมาในเวลานั้นมันถูกจำกัดให้พัฒนาในองค์กรทหาร, ห้องวิจัย, มหาวิทยาลัย, และบริษัทใหญ่ ๆ OLE เป็นเทคโนโลยีที่ได้รับการยอมรับไม่ได้ถูกนำเสนอต่อสาธารณชนจนกระทั่งบริษัทแอปเปิลได้ออกผลิตภัณฑ์ Macintosh ในแมคใช้ GUI (Graphical User Interface) และการโปรแกรมของมันก็ใช้หลัก Object Oriented

ในไม่กี่ปีต่อมามันก็ได้รับการยอมรับอย่างกว้างขวางเมื่อบริษัทไมโครซอฟท์ได้นำ OLE และ OOP มาใช้ใน NTServer และ Windows95 ปัจจุบันมันเป็นศาสตร์แห่งการเขียนโปรแกรมหลังจากเกือบ 50 ปีที่มันได้ริเริ่มขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนประกอบ OLE

การใช้ส่วนใหญ่ของ OLE อยู่บนข้อกำหนดของ COM (Component Object Model) COM แสดงคำสั่งและกฎเพื่อการพัฒนาโปรแกรมที่ใช้หลัก object-orient เมื่อนักเขียนโปรแกรม OOP ของวินโดวส์ได้พัฒนาแอปพลิเคชันของเขา โครงสร้างของ COM จะถูกนำมาใช้เป็นหลักในการเขียนโปรแกรม

เมื่อสร้าง ActiveX Control จะต้องแสดงส่วนติดต่อ IUnknown ซึ่งจะต้องสามารถลงบรรทัดได้เอง คอนโทรลจะส่งผ่านกลวิธี AddrOf() และ Release() โดยทันที หรือไม่ทันทีไปในภาษาที่ใช้เขียนโปรแกรมที่ใช้

OLE ได้เข้ามาเพิ่มผลประโยชน์อย่างมากให้ผู้ผลิตซอฟต์แวร์ผู้ซึ่งต้องการเพิ่มคุณค่าของผลิตภัณฑ์ของตัวเองด้วยการทำให้แอปพลิเคชันอื่นสามารถเข้าถึงการทำงานของซอฟต์แวร์ของพวกเขา ผู้ผลิตได้เปิดเผย API (Application Programming Interface) สำหรับผลิตภัณฑ์ และการพัฒนาจะอ้างอิงที่เอกสารเพื่อให้เห็นว่าทำอย่างไรในการที่จะโปรแกรมให้โต้ตอบกับผลิตภัณฑ์ได้ การใช้ API ที่ซับซ้อนนี้นักเขียนโปรแกรมจะต้องพัฒนา front ends จากซอฟต์แวร์ของผู้อื่นและคัดแปลงมันให้เข้ากับการใช้งานเฉพาะอย่างของลูกค้าของเขา

VBX 16 บิต Custom Control

VBX คือ Custom Control ตัวแรกที่มีใช้ มันรวมส่วนหลัก ๆ ของ API เข้าไว้เป็นตัวเองซึ่งจะถูกเพิ่มเข้าไปในโปรแกรม VBX นี้จะทำให้นักเขียนโปรแกรมสามารถเข้าถึงส่วนการทำงานหลักที่มันมีโดยไม่ต้องรู้ส่วนที่ซับซ้อนภายใน OLE แต่คอนโทรลจะทำให้ผู้ใช้สามารถโต้ตอบกับคอมโพเนนต์พิเศษของ OLE หรือกลุ่มของคอมโพเนนต์

ระบบการเขียนโปรแกรมแบบ Visual (เหมือนใน Visual Basic หรือใน Access) ซึ่งเป็น แอปพลิเคชันที่บรรจุ OLE จะสามารถเข้าถึงส่วนที่ซับซ้อนของ Object ภายนอกด้วยโค้ดเพียงไม่กี่บรรทัด แม้ว่ากระบวนการที่ VBX ใช้จะไม่ตรงกับ OLE เลยทีเดียวแต่มันเป็น OLE ล้อเลียน (Mock OLE) ที่มีประโยชน์มากที่สุด

การพัฒนาของ Custom Control นำไปสู่ผลประโยชน์ที่เพิ่มมากขึ้นของนักเขียนโปรแกรมซึ่งในหลายคนเหล่านี้ทำงานด้วยภาษาสำหรับเขียนโปรแกรมซึ่งไม่ใช่ OOP โดยแท้จริง แม้ว่า VBX จะไม่ใช่ OLE อย่างแท้จริงแต่มันก็ทำลายกำแพงของนักเขียนโปรแกรมที่จะเข้าถึงการทำงานของภายในคอมโพเนนต์ของโปรแกรมจากส่วนนอก

OCX 16/32 บิต Object Control

คอนโทรล OCX ถูกพัฒนาขึ้นเพื่อตอบสนองที่ต้องการความง่ายแต่ยังคงประสิทธิภาพที่เข้าถึงคอนโทรลและการทำงานของ OLE ซึ่ง VBX ไม่สามารถทำได้ แม้ว่าส่วนใหญ่ของคอนโทรลเหล่านี้จะยังคงไม่เป็น OLE คอนโทรลอย่างแท้จริง แต่ส่วนใหญ่ของมรรวมการทำงานของ OLE ที่ซับซ้อนมากขึ้น (และเป็นส่วนที่ซับซ้อน) การเพิ่มความสามารถของ OLE ล้อเลียนของ OCX ตัวแรกได้เพิ่มความต้องการเหล่านี้

Custom control เช่น VBX และ OCX ได้จัดหาทางที่จะรวมส่วนที่ซับซ้อนเข้าเป็นแพ็คเกจที่ใช้ง่าย ด้วยเทคโนโลยีซึ่งเพิ่มคุณค่านักเขียนโปรแกรมเริ่มต้องการอุปกรณ์มากขึ้น การต้องการที่นักพัฒนาร่วมส่วนของ OLE จริง ๆ เข้าไป OCX ตอบสนองความต้องการเหล่านี้ได้ดีแต่ยังคงเป็นสิ่งที่สับสนและถกเถียงเกี่ยวกับการที่ OCX เป็น OLE คอนโทรลที่แท้จริง

2.3 ความปลอดภัยของ ActiveX บนอินเทอร์เน็ต

เทคโนโลยี ActiveX ที่ใช้อยู่บนเครื่องนั้นสามารถนำมาใช้บนอินเทอร์เน็ตได้ในเว็บเพจ สามารถที่จะนำคอนโทรลต่าง ๆ หรือแอปพลิเคชันทั้งอันที่เป็น ActiveX โปรแกรมไปวางบนเว็บเพจได้ เนื่องจากมีคอนโทรล ActiveX มากมายให้ใช้บนอินเทอร์เน็ต และเมื่อคิดที่จะใช้มันก็ต้องดาวน์โหลดมัน แล้วคอนโทรลก็จะลงทะเบียนกับระบบปฏิบัติการ และมันก็จะสามารถถูกใช้โดยแอปพลิเคชันอื่น ๆ ที่อยู่บนระบบด้วยได้ นอกเหนือจากใช้บนเบราว์เซอร์ กระบวนการนี้เรียกว่า การดาวน์โหลดคอมโพเนนต์ (component downloading)

อย่างไรก็ตาม ยังมีข้อจำกัดบางประการและคำถามสำคัญที่มีอยู่ซึ่งจะต้องถามเพื่อป้องกันในแต่ละการดาวน์โหลดคอนโทรล ActiveX มายังเครื่อง

- คอนโทรล ActiveX บางตัวไม่สามารถนำมาใช้หรือเผยแพร่ได้ถ้าไม่มีใบอนุญาตจากนักพัฒนาของคอนโทรลนั้น ใบอนุญาตที่ทำให้สามารถใช้คอนโทรลนั้นในเว็บเพจของตัวเอง ผู้ใช้สามารถดาวน์โหลดคอนโทรล และลงทะเบียนคอนโทรลนั้นกับระบบของผู้ใช้ แต่ผู้ใช้ไม่อาจใช้มันได้ถ้าไม่มีใบอนุญาตสำหรับมัน

เหตุผลอื่นที่คอนโทรล ActiveX ต้องมีใบอนุญาตเพื่อเป็นทางหนึ่งของการแจ้งว่าคอนโทรลนั้นมีความปลอดภัยที่จะใช้บนเว็บเพจ ตัวอย่างของคอนโทรลมักเป็นเช่นนั้นซึ่งทำงานภายใต้เบราว์เซอร์ในการสร้างไฟล์ของผู้ใช้และการทำอื่นที่คล้ายกัน

- ผู้ใช้จะรู้ได้อย่างไรว่าโค้ดที่ใช้สร้างคอนโทรลนั้นปลอดภัย และมาจากแหล่งที่น่าเชื่อถือ ? ขณะที่ยังที่จะเชื่อบริษัทที่สามารถกระจายคอมโพเนนต์ ActiveX ผู้ที่คิดร้ายก็สามารถกระจายคอนโทรลได้เช่นกัน ดังนั้นอะไรจะเป็นตัววัดว่าจะป้องกันการทำลายและอันตรายของคอนโทรลซึ่งอาจบรรลุไว้สได้ ? คำตอบได้ถูกวางไว้กับการเซ็นรหัส (code signing) และเทคโนโลยีเอาท์เทนต์โคด (the Authenticode technology) ที่เสนอโดยไมโครซอฟท์

การเซ็นรหัส (Code Signing)

หลักการของการเซ็นรหัส หรือความสัมพันธ์ของคอนโทรลกับบุคคลหรือบริษัทที่พัฒนามันถูกใช้เพื่ออธิบายความแท้ของคอนโทรล ActiveX ที่พัฒนาโดยผู้พัฒนารายย่อยหรือองค์กร หลักการนี้รับกับความต้องการในปัจจุบันเพราะว่ามีข้อมูลอยู่มากมายหลากหลายบนอินเทอร์เน็ตโดยเฉพาะที่อยู่ในรูปของคอนโทรล ActiveX คอนโทรล ActiveX มีความสามารถมากในการใช้บนอินเทอร์เน็ตแต่ก็แสดงถึงการทำลายสูงต่อระบบของเอกสารนี้เป็นเอกสารที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าคอมพิวเตอร์

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่เดิมมีข้อมูลไม่กี่ชนิดที่สามารถดาวน์โหลด(เทกซ์ไฟล์,ไฮเปอร์เทกซ์ไฟล์,รูป, และเสียง) ผู้ใช้จะถูกเตือนถ้ามีไฟล์ที่แตกต่างจากนี้แต่ขณะนี้เวลาของบราวเซอร์และเว็บที่นิ่งนี้ไม่ใช่หลักการของคอนโทรล ActiveX

ในความเป็นจริงซึ่งผู้ใช้อินเทอร์เน็ตที่ไม่รู้ว่าใครสร้าง หรือไม่ได้คิดแปลงเมื่อไร หรือบรรจิวีรัส ซึ่งทั้งหมดเป็นเหตุผลที่มี ดิจิตอลซิกเนเจอร์(digital signatures) หรือโคดที่ถูกต้องโดยเจ้าของมันในการให้ดาวน์โหลดคอมโพเนนท์ ActiveX เพื่อใช้ดิจิตอลซิกเนเจอร์ผู้พัฒนาซอฟต์แวร์บนอินเทอร์เน็ตต้องทำให้คอมโพเนนท์เหล่านี้ต้องบรรจุใบรับรองดิจิตอลจาก Certificate Authority(CA) ซึ่งนักพัฒนาสามารถใช้เพื่อเซ็นคอมโพเนนท์ที่กระจายบนอินเทอร์เน็ตทั้งหมดที่เขาจำหน่าย

กระบวนการสำหรับการเซ็นรหัสและทำดิจิตอลซิกเนเจอร์มีดังต่อไปนี้ :

- เริ่มจากนักพัฒนาบรรจุใบรับรองดิจิตอลสำหรับการใช้โปรแกรมที่ทำงานบนอินเทอร์เน็ตทั้งหมดที่ได้พัฒนา เกร็ดเหล่านี้จะรวมทั้งชื่อผู้พัฒนา,บริษัท(ถ้ามี),เว็บไซต์ของผู้พัฒนา,อีเมล แอดเดรส และข้อมูลอื่น ๆ สิ่งเหล่านี้จะอธิบายให้ผู้รู้ว่าโคดที่กำลังดาวน์โหลดอยู่นั้นมาจากแหล่งที่น่าเชื่อถือ มันจะให้ความมั่นใจ และรับประกันกับผู้ใช้มากขึ้น เพราะเขาจะรู้ว่าสิ่งที่กำลังดาวน์โหลด

- ขั้นต่อไป อัลกอริทึมแฮชทางเดียวจะทำงานและสร้างตัวต่อที่มีขนาดคงที่ ด้วยอันนี้สร้างมาจากข้อมูลที่ถูกรวบรวมจากตัวทำงานเดิมซึ่งได้ถูกออกแบบขึ้น ชิ้นส่วนทั้งหลายที่ระบุสำหรับแต่ละการทำงานจะถูกใส่ตรงซึ่งเป็นการแสดงที่เป็นไปได้อย่างไม่มีการจำกัดของตัวต่อ ถึงนี้เป็นสิ่งสำคัญมากในการเข้ารูปเป็นกุญแจลับสำหรับโปรแกรม ซึ่งอยู่ในขั้นตอนต่อไป

- กุญแจลับจะถูกสร้างจากตัวต่อนี้และเข้ารหัสโดยใช้เทคโนโลยีการเข้ารหัสความปลอดภัยข้อมูลของ RSA การสร้างกุญแจลับที่เหมือนกันสองอันเป็นไปได้ กุญแจลับสร้างบนพื้นฐานจากตัวต่อที่ถูกใช้เป็นตัวเดิม,ไม่เข้าไปเกี่ยว,ไม่แก้ไขสแตทของโปรแกรม กุญแจลับนี้จะถูกนำไปรวมกับใบรับรองที่บรรจุโดย Certificate Authority เพื่อสร้างดิจิตอลซิกเนเจอร์สำหรับโปรแกรม

- ขั้นสุดท้ายนักพัฒนาต้องรวมดิจิตอลซิกเนเจอร์เข้ากับโปรแกรม และพร้อมที่จะเผยแพร่ในอินเทอร์เน็ต โปรแกรมนี้จะวางบนเว็บเพจในแท็ก <OBJECT> หรือโดยวิธีอื่น และพร้อมที่จะถูกดาวน์โหลดโดยผู้ใช้ ตอนนี้ผู้ใช้ก็สามารถตัดสินใจในการที่จะติดตั้งหรือไม่ติดตั้งคอมโพเนนท์ตัวนั้น เมื่อดาวน์โหลดจากอินเทอร์เน็ตด้วยบราวเซอร์เช่น Microsoft Internet Explorer 3.0 ใบรับรองก็จะปรากฏขึ้นและอธิบายว่าโคดนั้นได้ถูกเซ็นรับรองและแสดงว่าใครเป็นเจ้าของโปรแกรม

กุญแจลับนี้จะรวมอยู่ในโปรแกรมและถูกเปรียบเทียบกับกุญแจสาธารณะที่ถูกสร้างโดยระบบของผู้ใช้ขณะที่ทำงาน ถ้ากุญแจนั้นเหมือนกันทุกอย่างแล้วใบรับรองก็จะปรากฏขึ้นและแสดงว่าโคดนี้เป็นโคดที่ถูกต้อง ถ้ากุญแจไม่เหมือนกันแล้วโปรแกรมจะเตือนว่าโปรแกรมที่ถูกต้องกับโปรแกรมที่ดาวน์โหลดมานั้นไม่เหมือนกัน และบางทีมันอาจมีไวรัส ถ้าไม่มีใบรับรองแล้วบราวเซอร์ก็จะเตือนผู้ใช้และแนะนำว่าไม่ควรติดตั้ง และตอนนี้ผู้ใช้ก็เลือกที่จะติดตั้งหรือไม่ก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาตรฐานเอาทีเทนติโคด (Authenticode Standard)

เอาทีเทนติโคดนี้เป็นการเสนอมาตรฐานสำหรับการแสดงการเซ็นรหัสและดิจิทัลซิกเนเจอร์ มาตรฐานเอาทีเทนติโคดนี้ถูกสนับสนุนโดยมากกว่า 40 บริษัทใหญ่ และ องค์กรรับรองก็จะออกใบรับรอง เอาทีเทนติโคดถูกใช้เป็นขั้นตอนพื้นฐานสำหรับการสร้างดิจิทัลซิกเนเจอร์แต่ตั้งขึ้นสำหรับคนที่บรรจุและใช้ใบรับรอง ใบรับรองนี้ถูกตั้งขึ้นมาเพื่อเป็นแนวทางซึ่งนักพัฒนาเห็นด้วย

มีการแยกประเภทของนักพัฒนาไว้สองประเภทคือนักพัฒนารายย่อยกับนักพัฒนาที่อยู่ในรูปของบริษัทหรือองค์กร สภาพการบรรจุใบรับรองก็ต่างกัน ค่าใช้จ่ายรายปีสำหรับใบอนุญาตทางการค้าสำหรับบริษัทจะแพงกว่าสำหรับนักพัฒนารายย่อย การต่ออายุรายปีสำหรับใบอนุญาตนี้ก็ถูกกว่า ใบอนุญาตทางการค้าต้องการบริษัทเพื่อจะได้มี Dun and Bradstreet (เอเจนซีเรตติ้งเครดิต) เครดิตเช็ค ในางเพิ่มขึ้นใบอนุญาตสามารถใช้ได้ไม่มีจำกัดกับโปรแกรมซึ่งบริษัทนั้นได้เลือกที่จะเผยแพร่ผ่านอินเทอร์เน็ต ผู้ใช้มักจะเชื่อผลิตภัณฑ์ของบริษัทมากกว่าของผู้พัฒนารายย่อย เพราะว่าการสามารถมองเห็นได้ง่ายกว่า

ผู้พัฒนารายย่อยที่ผ่านการรับรองจะสามารถบรรจุใบรับรองของเอาทีเทนติโคดด้วยค่าใช้จ่ายเพียงน้อยนิด ปัญหาที่คือเมื่อผู้ใช้ระดับปานกลางจะรู้สึกสับสนแค่ไหนในการดาวน์โหลดคอมโพเนนท์จากผู้ใช้รายย่อย สิ่งนี้ต้องการการตัดสินใจแต่ทั้งระดับย่อยและองค์กรของนักพัฒนาที่ต้องทำใบรับรองจาก CA ที่บอกว่าโปรแกรมของเขาเชื่อถือ และไม่มีโคดที่มุ่งร้าย หรือไวรัส นักพัฒนาต้องอธิบายและชี้แจงผลิตภัณฑ์ นี่เป็นกลไกหลักในการทำให้โคดน่าเชื่อถือขึ้น

2.4 ความรู้เบื้องต้นเกี่ยวกับโปรโตคอล TCP/IP

เปรียบเทียบระหว่างโปรโตคอลTCP/IP และ OSI โมเดล

การออกแบบโปรโตคอล TCP/IP นั้นไม่ได้เป็นรูปแบบของ OSI โมเดล เนื่องจากถูกออกแบบโดยองค์กรขนาดใหญ่ซึ่งใช้เวลานานในการออกแบบตลอดจนการรับรองมาตรฐานต่างกับโปรโตคอล TCP/IP ที่ถูกออกแบบด้วยความต้องการอันเร่งด่วนของรัฐบาลสหรัฐ จึงทำให้การพัฒนาโปรโตคอล TCP/IP มีเงื่อนไขของในด้านความต้องการที่ต่างจาก OSI โมเดล ซึ่งหากเรามองโดยรวมแล้วจะเห็นว่าโปรโตคอล TCP/IP มีการแบ่งเป็นเลเยอร์น้อยกว่า OSI โมเดลคือมี 4 ชั้นเท่านั้นดังรูป 2 โดยแบ่งเป็น

1.Application Layer
2.Transport Layer
3.Internet Layer
4.Physical Layer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอปพลิเคชัน (Application Layer)

ในเลเยอร์นี้ประกอบด้วยโปรแกรมประยุกต์ที่ใช้เครือข่าย และอาจกล่าวได้ว่าเลเยอร์นี้ของโพรโทคอล TCP/IP ก็คือเลเยอร์ในชั้นแอปพลิเคชัน รวมกับชั้นพรีเซนเตชันเลเยอร์ (Presentation Layer) ใน OSI โมเดลนั่นเอง และในเลเยอร์ชั้นนี้ของโพรโทคอล TCP/IP จะกลืนอยู่ในตัวโปรแกรมประยุกต์

ทรานสปอร์ตเลเยอร์ (Transport Layer)

ในชั้นนี้เป็นชั้นที่ให้การส่งข้อมูลจากจุดปลายถึงจุดปลาย หากเปรียบเทียบใน OSI โมเดล ก็สามารถเทียบได้กับชั้นเซสชันเลเยอร์ (Session Layer) รวมกับทรานสปอร์ตเลเยอร์นั่นเอง โดยโพรโทคอล TCP/IP มี ซอกเก็ต (Socket) เป็นจุดปลาย (end-point) ในการสื่อสาร ซึ่งซอกเก็ตนี้ประกอบไปด้วยหมายเลขของคอมพิวเตอร์ และหมายเลขพอร์ต (port) ของเครื่องที่ต้องการส่งข้อมูลไปถึง ในชั้นนี้มีการรับรองการถึงที่หมาย และลำดับของการส่งโดยไม่ซ้ำ และความผิดพลาดข้อมูล

อินเทอร์เน็ตเลเยอร์ (Internet Layer)

เลเยอร์นี้มีการกำหนดค่าแอดเดรส และการหาเส้นทางการส่ง หน้าที่ของเลเยอร์นี้เทียบเท่ากับเน็ตเวิร์กเลเยอร์ (Network Layer) และดาต้าลิงก์เลเยอร์ (Data Link Layer) ของ OSI โมเดล

ฟิสิกคอลลเลเยอร์ (Physical Layer)

โพรโทคอล TCP/IP ไม่ได้กำหนดรูปแบบของการเชื่อมต่อในระดับนี้ไว้ใหม่ แต่ได้ใช้มาตรฐานที่มีอยู่เดิมกำหนดไว้ก่อน เช่น RS232, อีเทอร์เน็ต (Ethernet) เป็นต้น

รูปแบบการกำหนดแอดเดรสของโพรโทคอล TCP/IP

ในหัวข้อนี้จะอธิบายรูปแบบการกำหนดแอดเดรสของโพรโทคอล TCP/IP ซึ่งลักษณะแอดเดรสของโพรโทคอลนี้ค่าของแอดเดรสของเครื่องคอมพิวเตอร์ในระบบเครือข่ายจะไม่ซ้ำกันเลย โดยหมายเลขนี้เรียกว่า IP แอดเดรส (IP Address) เป็นเลข 32 บิตซึ่งแบ่งเป็นคลาส (Class) ตามหลักการพิจารณาที่จะกล่าวต่อไป

การแบ่งเน็ตเวิร์กคลาส (Network Class)

เนื่องจากหมายเลขแอดเดรสของคอมพิวเตอร์เครื่องใด ๆ นั้นจะต้องสามารถบอกถึงความแตกต่างระหว่างตัวเครื่องเองตลอดจนเครือข่ายคอมพิวเตอร์นั้นเชื่อมต่ออยู่ด้วย หมายเลข IP แอดเดรส จึงแยกเป็น 2 ส่วน ได้แก่ ส่วนที่แสดงหมายเลขของคอมพิวเตอร์โฮส และส่วนที่เป็นหมายเลขของเครือข่าย

การแบ่งคลาสของแอดเดรสทำได้โดยพิจารณาจำนวนบิตของ 2 ส่วนประกอบข้างต้น ซึ่งมีการแบ่งออกเป็น 5 คลาส แต่มีการใช้เพียง 3 คลาสแรกเท่านั้นคือ คลาส A, คลาส B, และคลาส C ส่วนคลาส D และคลาส E ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถูกสงวนไว้สำหรับจุดประสงค์พิเศษ โดยปกติแล้วผู้พัฒนาโปรแกรมไม่ต้องสนใจความแตกต่างระหว่างคลาสของ IP แอดเดรส

31	0
Class ID	Network ID
Host ID	

รูปที่ 3 หมายเลข IP แอดเดรส

Network Class	Networks	Host per Network
A	126	16,777,214
B	16382	65,534
C	2,097,150	254

รูปที่ 4 แสดงคลาส, จำนวนเครือข่าย และจำนวนโฮสต์ในแต่ละคลาส

การแทนด้วยเลขฐานสิบ และจุด (Dotted Decimal Notation)

เนื่องจากการแทนหมายเลข IP แอดเดรสเป็นเลขฐาน 2 ซึ่งอ่านค่อนข้างยากไม่สะดวกจึงมีการแทนเลขฐานสองเหล่านี้ในรูปตัวเลขฐานสิบและจุดโดยเลขฐานสิบแต่ละตัวจะแทนเลขฐานสองจำนวน 8 บิต โดยระหว่างเลขฐานสิบแต่ละตัวจะแทรกด้วยจุด ดังนั้นจะต้องใช้เลขฐานสิบ 4 ตัว ในการแทนเลข 32 บิตที่เป็น IP แอดเดรส ดังตัวอย่างตามรูปที่ 5 จะสังเกตว่าหมายเลข IP แอดเดรสนี้จัดอยู่ในคลาส B โดยหมายเลขของเครือข่ายคือ 166.78 และหมายเลขประจำเครื่องคือ 4.139

Dotted Decimal Notation
166.78.4.139
Binary representation
10100110 01001110 00000100 10001011

รูปที่ 5

การทำซับเน็ตติง (Subnetting)

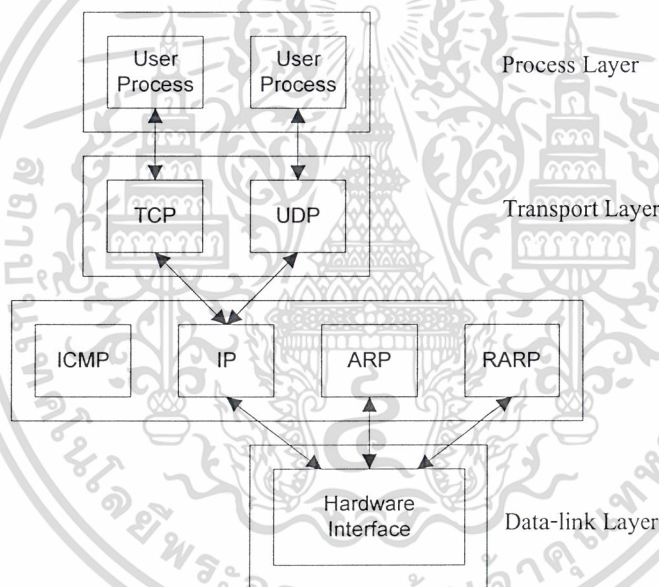
การทำซับเน็ตติงเป็นการเปลี่ยนแปลงการใช้หมายเลขของเครื่องโฮสต์ และหมายเลขของเครือข่ายในระดับท้องถิ่น โดยในทางตรรกคือการเลื่อนเส้นแบ่งที่แยกหมายเลขเครื่องกับหมายเลขของเน็ตเวิร์กที่อยู่ในหมายเลข IP แอดเดรส โดยที่ปริมาณของหมายเลขเครื่องโฮสต์ และหมายเลขเครือข่ายจะแปรผกผันกัน เช่น หากมีเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ปริมาณของเน็ตเวิร์กมากก็จะทำให้เครื่องใด ๆ ที่จะต่อกับระบบเครือข่ายหนึ่ง ๆ น้อยลง เป็นต้น ในทางปฏิบัติการแบ่งการดูแล ทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ ทำซับเน็ตติงทำโดยการนำซับเน็ตมาร์ค (Subnet mark) คือตัวเลขจำนวน 32 บิต มาทำการกระทำตรรก "และ" กับ

หมายเลข IP แอดเดรส ดังตัวอย่างโดยกำหนดหมายเลข IP แอดเดรสคือ 166.78.4.139 และชั้นเน็ตเวิร์กคือ 255.255.255.0 ทำการกระทำตรรก ดังรูป

จะเห็นว่าหากพิจารณาโดยไม่มีการทำชั้นเน็ตเวิร์กแล้วจะได้หมายเลขของเน็ตเวิร์กคือ 166.78 และหมายเลขประจำเครื่องคือ 4.139 แต่ผลที่ได้จากการกระทำตรรก “และ” ในรูปข้างต้น ผลลัพธ์ที่ได้คือ 166.78.4.0 และที่เหลือคือ 139 หากพิจารณาผลลัพธ์นี้คือหมายเลขเน็ตเวิร์กคือ 166.78.4.0 และหมายเลขเครื่องคือ 139 หรืออาจพูดได้ว่าคอมพิวเตอร์นี้มีหมายเลขเครื่องเท่ากับ 139 และอยู่บนเครือข่ายหมายเลข 166.78.4

ชุดโพรโตคอล TCP/IP (TCP/IP Protocol Suite)

ชุดโพรโตคอล TCP/IP นอกจากมีโพรโตคอล TCP และ IP แล้ว ยังมีโพรโตคอลอย่างอื่นอีก ดังรูปที่ 6 แสดงความสัมพันธ์ของชุดโพรโตคอลโดยแบ่งตามเลเยอร์



รูปที่ 6 ความสัมพันธ์ระหว่างชุดโพรโตคอล

โพรโตคอล IP (Internet Protocol)

โพรโตคอล IP เป็นโพรโตคอลแบบคอนเนกชันเลส (Connectionless Protocol) ซึ่งลักษณะของโพรโตคอลชนิดนี้เป็นแบบที่ไม่มีขั้นตอนการสร้างการติดต่อ และขั้นตอนยกเลิกการติดต่อ แต่จะมีขั้นตอนการส่งผ่านข้อมูลเพียงอย่างเดียว โดยข้อมูลเรียกว่าแพคเกจจะถูกส่งจากระบบหนึ่งไปสู่ระบบหนึ่งอย่างเป็นอิสระโดยไม่ขึ้นอยู่กับตัวแพคเกจอื่น ๆ โดยที่โพรโตคอล IP ไม่รับประกันว่าข้อมูลที่ส่งจะไปถึงปลายทางซึ่งแพคเกจ (Packet) ของข้อมูลอาจไปถึงในลักษณะที่ผิดลำดับ, ซ้ำกัน, หรือไปไม่ถึงเลย โดยความน่าเชื่อถือของการส่งจะถูกควบคุมในโพรโตคอลในเลเยอร์ต่าง ๆ ไป การหาเส้นทางของข้อมูลจะทำในระดับของโพรโตคอล IP นี้ โดยพิจารณาแต่ละแพคเกจแยกออกจากกัน และยังมีหน้าที่ในการจัดเรียงข้อมูลที่ใหม่ที่ปลายทางด้วย

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โพรโทคอล ARP (Address Resolution Protocol)

โพรโทคอลนี้ทำหน้าที่จับคู่ระหว่างหมายเลข IP แอดเดรสเข้ากับหมายเลขแอดเดรสทางฮาร์ดแวร์ โดยโพรโทคอลนี้ทำการส่งข้อความไปทั่วเครือข่ายท้องถิ่น ซึ่งข้อความนี้เป็นลักษณะข้อความที่ตรวจสอบว่ามีคอมพิวเตอร์ที่มีหมายเลข IP แอดเดรสตรงกับที่ต้องการหาหรือไม่ หากคอมพิวเตอร์ที่มีหมายเลข IP แอดเดรสตรงกันนั้นได้รับข้อความนี้ก็จะตอบกลับและเป็นที่น่าสังเกตว่าโพรโทคอลนี้ทำงานได้กับระบบเครือข่ายท้องถิ่นเท่านั้น เพราะว่าโครงสร้างหมายเลขทางฮาร์ดแวร์ของเครื่องคอมพิวเตอร์จะขึ้นอยู่กับชนิดของระบบเครือข่ายด้วย

โพรโทคอล ICMP (Internet Control Message Protocol)

เป็นโพรโทคอลที่จัดการเกี่ยวกับข่าวสารความผิดพลาดและการควบคุมเกตเวย์ และเครื่องคอมพิวเตอร์ในระบบเครือข่าย

โพรโทคอล RARP (Reverse Address Resolution Protocol)

เป็นโพรโทคอลที่ทำหน้าที่จับคู่ระหว่างหมายเลขของฮาร์ดแวร์กับหมายเลข IP แอดเดรส หรือทำงานกลับกันกับโพรโทคอล ARP

โพรโทคอล UDP (User Datagram Protocol)

โพรโทคอลนี้เป็นโพรโทคอลที่อยู่ในระดับทรานสปอร์ตเลเยอร์ และมีความสำคัญเพราะว่าเป็นโพรโทคอลที่ผู้พัฒนาโปรแกรมสามารถใช้ได้โดยตรง โพรโทคอลนี้เป็นโพรโทคอลแบบคอนเนกชันเลสมีความน่าเชื่อถือต่ำไม่รับรองว่าข้อมูลที่ส่งไปจะไปถึงปลายทางหรือไม่ และอาจซ้ำซ้อน หรือผิดพลาดก็ได้ แต่ข้อดีของโพรโทคอลนี้คือค่าความสิ้นเปลือง (overhead) ต่ำ

โพรโทคอล TCP (Transmission Control Protocol)

โพรโทคอลนี้อยู่ในระดับทรานสปอร์ตเลเยอร์ เหมือนกับโพรโทคอล UDP แต่มีลักษณะตรงข้ามกันคือ เป็นโพรโทคอลแบบคอนเนกชันออเรียนเต็ด คือต้องมีขั้นการสร้างการติดต่อ (connection establishment), ขั้นการส่งผ่านข้อมูล (data transfer), และขั้นตอนยกเลิกการติดต่อ (connection termination) โดยจะมีความน่าเชื่อถือในการรับส่งข้อมูล และลำดับของข้อมูลจะมีลำดับเหมือนกับต้นทางและเนื้อข้อมูลไม่ผิดพลาด จึงทำให้เกิดความสิ้นเปลืองในการเชื่อมต่อของข้อมูลสูงกว่าโพรโทคอล UDP

หมายเลขพอร์ต

เนื่องจากในเวลาใด ๆ สามารถมีกระบวนการของผู้ใช้ สามารถใช้ UDP หรือ TCP ได้พร้อมกันดังนั้นจึงต้องมีวิธีแยกแยะว่าข้อมูลเป็นของผู้ใช้คนใด ซึ่งวิธีที่ TCP และ UDP ใช้ก็คือการใช้หมายเลขพอร์ต (port number) เมื่อกระบวนการของไคลเอนต์ต้องการจะติดต่อกับเซิร์ฟเวอร์ ไคลเอนต์จะต้องเจาะจงเซิร์ฟเวอร์ที่ต้องการติดต่อแต่ถ้าฝั่งผู้แอดเดรส 32 บิตอย่างเดียวไม่พอ เพราะว่าสามารถติดต่อกับโฮสต์ได้เพียงอย่างเดียวแต่ไม่

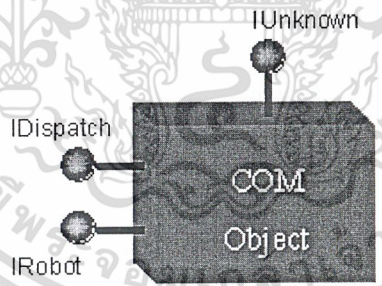
สามารถเจาะจงกระบวนการที่จะทำการติดต่อได้ ดังนั้นเพื่อแก้ปัญหานี้ทั้ง TCP และ UDP ได้มีการกำหนดหมายเลขพอร์ตมาตรฐาน (well-known port) ซึ่งเป็นที่รู้จักกัน

เมื่อ TCP หรือ UDP กำหนดหมายเลขพอร์ตที่ไม่ซ้ำกันให้กระบวนการของผู้ใช้ เราเรียกหมายเลขพอร์ตนี้ว่าหมายเลขพอร์ตชั่วคราว (ephemeral port number) เมื่อไคลเอนต์เลิกใช้หมายเลขพอร์ตนี้แล้ว สามารถกำหนดหมายเลขพอร์ตนี้ให้ไคลเอนต์อื่นได้ กระบวนการที่ได้รับหมายเลขพอร์ตชั่วคราวนี้ไม่สนใจว่ามีค่าเท่าไร แต่เป็นหน้าที่ของอีกกระบวนการหนึ่งที่ต่อกันที่ต้องสนใจ เพราะต้องส่งข้อมูลกลับมาที่พอร์ตนี้ ใน TCP และ UDP นั้นหมายเลขพอร์ตมีตั้งแต่ 1 -1023 เป็นพอร์ตที่สงวนไว้สำหรับหมายเลขพอร์ตมาตรฐาน

2.5 สถาปัตยกรรม COM (Component Object Model)

หลักพื้นฐานของ COM นักออกแบบ COM ได้เพียรพยายามเพื่อ

- เป็นไบนารีคอมโพเนนท์ (Binary Components) ได้แก่ ใช้ภาษาใด ๆ ในการเขียนโปรแกรมก็ได้, ทำงานที่ใด ๆ (กระบวนการเดียวกัน:in-process,ข้ามกระบวนการ:cross-process,ข้ามเครื่อง:cross-machine)ก็ได้, และไม่มีกรรวมอำนาจ
- ไม่ให้มีการสูญเสียประสิทธิภาพการทำงานของระบบภายใน
- รูปแบบเรียบง่ายที่สุดเท่าที่เป็นไปได้ (สามารถเพิ่มเติมและเปลี่ยนแปลงได้)

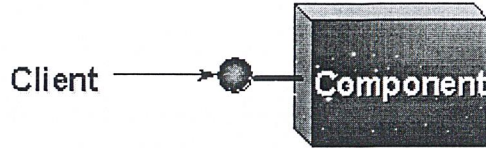


รูปที่ 7 Interface ของ COM Object

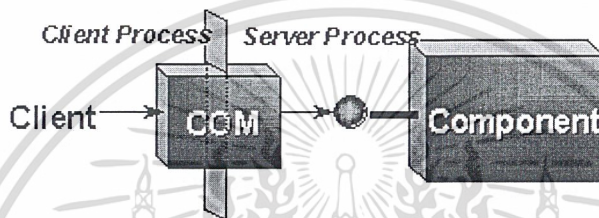
หลักของ COM

- เข้ารหัสอย่างเคร่งครัด
 - ทำรหัสให้เหมือนกล่องดำที่ เครื่องมือ ไม่มีรู้
 - ทุกชิ้นงานต้องสร้างการติดต่อที่เข้มงวด
- Polymorphism

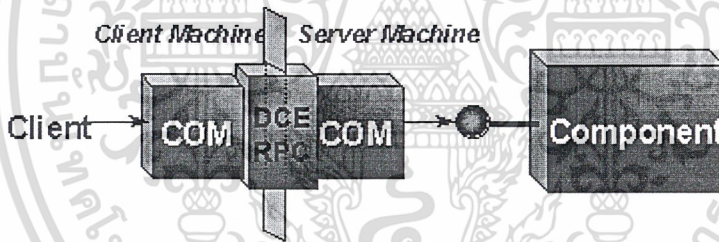
เอกสารนี้เป็นเอกสารที่ติดต่อกับหลายทางต่อหนึ่งคลาส มีการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 มี QueryInterface ให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก) ในกระบวนการเดียวกัน สามารถเรียกทำงานโดยตรง และเร็ว



ข) บนเครื่องเดียวกัน มีความปลอดภัย และ IPC เร็ว



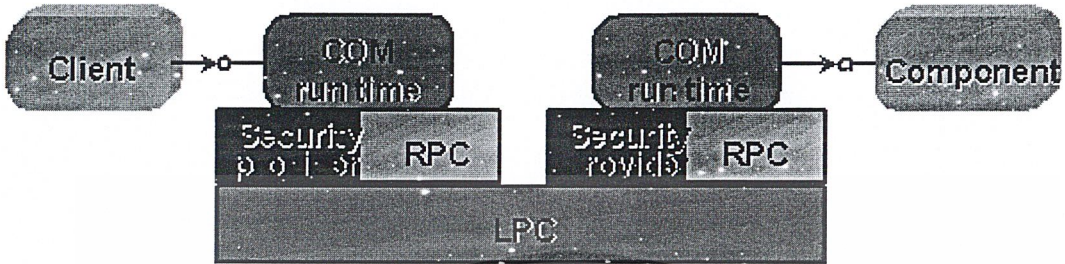
ค) ข้ามเครื่อง มีความปลอดภัย, เชื่อถือได้, และมีความยืดหยุ่น

รูปที่ 8 แสดงสถาปัตยกรรมของ COM ที่สามารถแบ่งทำงานตามระดับการใช้งานได้

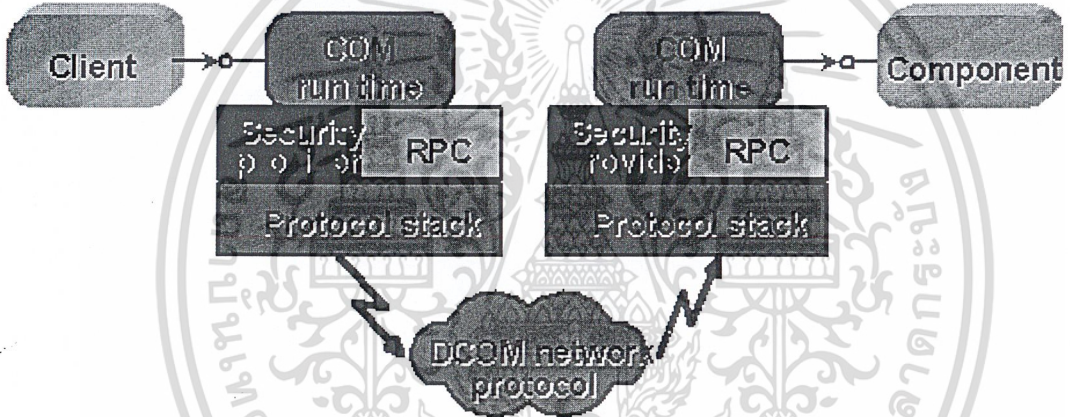
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก) ในกระบวนการเดียวกัน (In-process)



ข) บนเครื่องเดียวกัน (Local)



ค) คนละเครื่อง (Remote)

รูปที่ 9 แสดงการทำงานของ COM

Distributed COM (DCOM)

ถือว่าเป็น COM ที่ใช้ในการติดต่อที่ไกลกัน เพราะว่ามันคือ COM ที่เพิ่มมาตรฐานโพรโตคอลในการส่งข้อมูลเท่านั้นเอง โดยที่คอมโพเนนท์ทั้งหมดสื่อสารในแบบเดียวกัน ทั้งบนเครื่องเดียวกัน(ทั้งกระบวนการภายใน:in-process, ทั้งกระบวนการภายนอก:out-process), ทั้งที่ผ่านเครือข่ายท้องถิ่น, ผ่านเครือข่ายระยะไกล, และผ่านอินเทอร์เน็ต แล้วยังใช้เครื่องมือ, ความรู้, โทดเดียวกัน

บริการหลักของ COM

เอกสารนี้เกี่ยวกับความปลอดภัย Security หรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่สามารถแจ้งการจัดการ Lifecycle Management นี้หาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ชนิดข้อมูล Type Information (Interface Repository)
- Monikers (Naming)
- อัตโนมัตินิ Automaton (Dynamic Invocation)
- การเคลื่อนย้ายข้อมูล Data Transfer
- แบ่งประเภทคอมโพเนนต์ Component Categories
- การลงทะเบียน Registry

2.6 การออกแบบอินเทอร์เฟซและคอมโพเนนต์ที่เป็น COM (Component Object Model)

พื้นฐานขั้นต้นของการสร้างคอมโพเนนต์ที่เป็น COM

ออกแบบก่อน เพราะว่าอินเทอร์เฟซจะไม่สามารถเปลี่ยนแปลงได้จึงต้องทำให้ถูกในครั้งแรก และการเปลี่ยนคลาสอินเทอร์เฟซอาจผลักดันให้ไคลเอนต์เปลี่ยนต้องพยายามทำสิ่งนี้ให้ถูกต้องตั้งแต่แรกด้วย

ปรับปรุงคุณภาพในขั้นสุดท้าย เครื่องมือสร้างคอมโพเนนต์ที่ดีมีเพียงพออยู่แล้ว ใช้ประสิทธิภาพที่ต้องการของคอมโพเนนต์เพื่อเลือกเครื่องมือที่เหมาะสม และในการสร้างสามารถเปลี่ยนได้ทุกเวลา เมื่อปรับการสร้างหนึ่งครั้งอินเทอร์เฟซที่เป็นสาธารณะจะถูกรับ

หลักการออกแบบทั่วไป

- ต้องเข้าใจปัญหาของธุรกิจ
- กำหนดลูกค้า
 - ใครเป็นผู้ที่ใช้บริการของเครือข่าย?
 - ใครเป็นผู้ที่นำคอมโพเนนต์ไปใช้สร้างแอปพลิเคชัน?
- กำหนดข้อบังคับในการออกแบบ
 - ความต้องการด้านประสิทธิภาพของคอมโพเนนต์
 - ความต้องการด้านการใช้ของคอมโพเนนต์
 - ความต้องการด้านความปลอดภัยในการใช้คอมโพเนนต์
- ตัดสินใจว่าคอมคือเทคโนโลยีที่เหมาะสมกับงานหรือเปล่า

การออกแบบอินเทอร์เฟซ

ต้องการออกแบบขั้นพื้นฐานก่อน จากนั้นก็ดูข้อบังคับของอินเทอร์เฟซนั้นเช่นตัวลูกค้า, ประสิทธิภาพ, เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครูใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า, การใช้งาน, และ ความปลอดภัย ความสำเร็จของการออกแบบอย่างไรบ้าง

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พื้นฐาน

- มองหาอินเทอร์เฟซที่มีอยู่เดิมที่ตรงตามความต้องการก่อนถ้าไม่ได้ค่อยสร้างอินเทอร์เฟซใหม่
- ทุกอินเทอร์เฟซ ...
 - ไม่สามารถเปลี่ยนแปลงได้
 - ได้มาจาก IUnknown
 - มี IID ที่ไม่เหมือนใคร
 - คืนค่า HRESULT จากแต่ละเมทอด (method)
- ทุกอินเทอร์เฟซควรจะ...
 - ต้องระวังในการกำหนดความหมาย ของภาวะก่อน/หลัง, แขนงฝั่งสถานะ, ข้อบังคับที่รีเอนทรานซ์ซี (reentrancy)
 - ต้องถูกอธิบายในไทป์ไลบรารี (type library) และ IDL
 - มีชื่อที่มีความหมายอย่างสมเหตุสมผล
 - ต้องใช้ระยะไกลได้
 - ต้องการพารามิเตอร์ชนิดที่เปลี่ยนแปรได้ยาก (strongly type parameter)
 - ใช้ Unicode สำหรับพารามิเตอร์สตริง

พื้นฐาน – แฟกเตอร์ริง (Factoring)

- แบ่งอินเทอร์เฟซไปสู่กลุ่มที่สามารถจัดการด้วยกันได้และมีความเกี่ยวข้องกันของเมทอด
 - เลี่ยง E_NOTIMPL syndrome
 - เลี่ยงแฟลกบอกความสามารถ
 - คิดเกี่ยวกับค่า QI
- เก็บรายการพารามิเตอร์ที่สามารถจัดได้ด้วย
 - คิดเกี่ยวกับราคาของ round trip เทียบกับค่าเรียบเรียง
- ภาษาที่บ่งชนิดไม่ได้สามารถก่อให้เกิดปัญหาได้

พื้นฐาน – ชนิดพารามิเตอร์ (Parameter types)

- ใช้ชนิดที่เปลี่ยนแปรได้ยากไม่ใช่ตัวแปร
- ใช้ชนิดของตัวชี้อินเทอร์เฟซที่แข็งแรง (strong interface pointer types) ได้แก่ชนิดที่ไม่ว่างเปล่า และให้ผู้ใช้ระบุค่า IID เพื่อการคืนค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานานับไปของบุคคลในวงไปใช้ประโยชน์ด้านการค้า
 HRESULT GetInterface([In]REFIID iid, [out, iid_id(riid)]IUnknown**ppUnk);
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น เลี่ยงการทำให้เป็นกลุ่มเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คิดให้ดีเกี่ยวกับ โดยมูลค่า (by value) กับ โดยอ้างอิง (by reference)

พื้นฐาน – HRESULT

- ถ้าเหมาะสมก็ใช้มาตรฐาน COM HRESULT
- กำหนดค่าโค้ด interface-specific failure ใหม่ โดยใช้ FACILITY_ITF
- เลี่ยงการกำหนดโค้ดใช้ใหม่ เพราะว่าบางภาษาไม่สามารถทำได้ และจะมีความสับสนมาสู่นักเขียนโปรแกรมส่วนใหญ่ แต่ถ้าจะทำให้พิจารณาส่วนของข้อบังคับของอินเทอร์เฟซให้ดี

พื้นฐาน – เรียกกลับ (Callbacks)

- คิดเกี่ยวกับการใช้จุดการเชื่อมต่อ (connection points) เป็นการง่ายที่จะใช้เหตุการณ์ (event) ใน VB, Scripting
- ไตร่ตรองข้ออ้างที่วนเวียน ต่อกาทางที่จะหยุดการวนเวียน และ IDataObject [Un] Advise เป็นรูปแบบที่ดี
- คิดเกี่ยวกับข้อบังคับการสร้าง ว่าสามารถเรียกกลับอย่างปลอดภัยหรือเปล่า หรือว่าการเรียกกลับถูกสนับสนุน โดยการส่งแบบ RPC หรือเปล่า

พื้นฐาน – การเรียบเรียง (Marshaling)

- ใช้ [V1_enum] บนทุก enums ต่่ง 32 บิต เก็บแนวโครงสร้าง
- อย่าใช้เกิน [length_is] แต่ [size_is] ก็ควรจะพอ
- เลี่ยงตัวชี้ full [ptr] ยอมให้ปลอม, ยอมให้แพง

พื้นฐาน – การทำรุ่น (Versioning)

- อินเทอร์เฟซไม่สามารถเปลี่ยนแปลงได้ ทั้งไม่เปลี่ยนไวยากรณ์, ไม่เปลี่ยนความหมาย, และไม่เปลี่ยนการเรียบเรียง
- การทำรุ่นเกิดจากการสร้างอินเทอร์เฟซใหม่ ๆ โดยที่จะใช้จากอินเทอร์เฟซเก่าเพียงถ้าใช้ข้อกำหนดจากอินเทอร์เฟซเก่า

ใครเป็นไคลเอนต์ ?

- เป็นแอปพลิเคชันเดี่ยวหรือออกแบบสำหรับการใช้ซ้ำได้ (reuse) ?

- ถ้าเป็นอินเทอร์เฟซที่จะใช้ซ้ำได้จะต้องการการออกแบบที่ระมัดระวังมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครูผู้สอนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังสงวนลิขสิทธิ์และข้อความข้างต้นไว้ของเอกสารทุกครั้งที่มีการนำไปใช้

จัดทำคำสั่งให้เป็นและเพียงพอสำหรับ ไคลเอนต์รู้ว่ามีอยู่แล้ว

- อาจต้องการแฟลคเตอร์ต่าง ๆ สู้หลาย ๆ อินเทอร์เฟซ
- เป็นนักพัฒนาในบ้านหรือข้างนอก
 - ความง่ายในการใช้อาจจะสำคัญกว่าประสิทธิภาพเลิศหรู
- ภาษาที่ไคลเอนต์ใช้ในการพัฒนา
 - เครื่องมือบางตัว (VB,Java) เข้าใจแต่ชนิดที่เป็นอัตโนมัติ
 - ภาษาที่บ่งชนิดไม่ได้ (สคริปต์) เข้าใจแค่ IDispatch โดยมีหนึ่งอินเทอร์เฟซต่อแอปเจค
 - ในภาษา C++ ยากที่จะใช้ IDispatch / ตัวแปร

ใครเป็นผู้สร้าง ?

- ผู้สร้างมีจำนวนเท่าไร ?
 - กำหนดไวยากรณ์อย่างระมัดระวัง
 - ความเข้มงวดของข้อบังคับนั้นมีเท่าใด
 - แฟลคเตอร์อินเทอร์เฟซเพื่อหลีกเลี่ยง E_NOTIMPL
- ทำในบ้านหรือข้างนอก ?
 - ความง่ายในการสร้างอาจจะสำคัญกว่าประสิทธิภาพเลิศหรู
- ภาษาอะไรที่คอมไพเลอร์พัฒนา
 - เครื่องมือบางตัว (VB,Java) เข้าใจแต่ชนิดที่เป็นอัตโนมัติ
 - ตัวแปร หรือ พารามิเตอร์ที่เลือกค่าได้ ยากที่จะทำงานกับ C++

ประสิทธิภาพที่ต้องการ

- การจราจรในเครือข่าย
 - ใช้ การปรับแบบ IDL อย่างสุขุม ใช้ [size_is] และ [length_is]
 - คิดเกี่ยวกับจำนวนข้อมูลต่อแต่ละการเรียก
- ใช้เมททอดดีกว่าใช้พรอพเพอร์ตี้ (property) เวลาใช้ระยะไกล
- เวลาตอบรับ
 - วิธีการแบบซิงโครนัส (synchronous) อาจจะกั้นไคลเอนต์
 - คำนึงถึงการแบ่งงานสู่หลาย ๆ เมททอด
 - คำนึงถึงการใช้การเรียกกลับเพื่อส่งสัญญาณเมื่องานที่ยาว ๆ ได้สำเร็จ
 - คำนึงถึงข้อมูลที่ส่งผ่านหลาย ๆ การเรียก, NT5 COM Pipes
 - มองหาตัวเลือกที่เป็นอะซิงโครนัส (asynchronous) ใน NT5.COM+

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้า

รับประกันการสงหรือเปล่า ?

ไม่ว่ากรณีใดๆ ทั้งสิ้น ก็กำลังทำมันให้ดีขึ้นและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ไม่ อินเทอร์เฟซมีเพียงพารามิเตอร์ [in]

- รับ เลี้ยงความผิดพลาดที่เซิร์ฟเวอร์กำลังยุ่งหรือปิดอยู่ และอินเทอร์เน็ตถูกจำกัดสู่พารามิเตอร์ [in]

การใช้ที่ต้องการ

- ไคลเอนต์/ผู้สร้างอยู่ที่ไหน ?
 - คิดเกี่ยวกับการจราจรระหว่างเครื่องในเครือข่าย
- LAN, WAN, หรืออินเทอร์เน็ต
 - คิดเกี่ยวกับ Latency และ Bandwidth
 - บริการทางอินเทอร์เน็ตของคอมไม่มีการเรียกกลับ
- ไคลเอนต์สามารถเลิกติดต่อได้หรือไม่ ?
 - MTS สนับสนุนชนิดอัตโนมัติ
 - MTS ออปเจกต์ไม่สามารถเก็บสถานะข้ามขอบเขตของการจัดการ ต้องผ่านค่าสถานะไปยังออปเจกต์ในเมททอด
 - MTS ตั้งค่าความปลอดภัยที่ระดับอินเทอร์เน็ตเฟซ ไม่ใช่ต่อแต่ละเมททอด

ความปลอดภัยที่ต้องการ

- ไคลเอนต์ที่ต่างกันจะใช้กลุ่มของคำสั่งที่ต่างกันหรือไม่? แฟคเตอร์อินเทอร์เน็ตเฟซ(factor interfaces)สำหรับชนิดไคลเอนต์ที่ต่างกัน
- พารามิเตอร์ต้องการจะเข้ารหัสหรือไม่? คำนี้ถึงการกำหนดความหมายของอินเทอร์เน็ตเฟซ

การออกแบบคอมโพเนนท์

ขั้นแรกต้องกำหนดคลาสก่อน จากนั้นค่อยกำหนดอินเทอร์เน็ตเฟซที่แสดงออกโดยแต่ละคลาส แล้วตัดสินใจว่าจะรวบรวมเป็นชิ้นอย่างไร สุดท้ายพิจารณาว่าข้อบังคับมีอิทธิพลแค่ไหนกับการออกแบบ

การกำหนดคลาส

- คำนี้ถึงลักษณะทางกายภาพของเครือข่าย ว่าแต่ละการแสดงผลออก, การแสดงผลออกสัมพันธ์กับหลักความจริงของธุรกิจ แต่ต้องไม่ไกลจากหลักความจริงของคุณ
- หมวดหมู่ของอินเทอร์เน็ตเฟซ/คอมโพเนนท์ อาจจะมาจากอะไรที่อินเทอร์เน็ตเฟซแสดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรวบรวมให้เป็นชิ้น

- ตัดสินใจว่าคลาสใดไปด้วยกัน
- DLL, EXE, หรือบริการ? คำนึงถึงการแยกกันของกระบวนการ และก็คำนึงถึงความปลอดภัยด้วย

ใครเป็นไคลเอนต์ ?

- เป็นแอปพลิเคชันเดี่ยวหรือออกแบบสำหรับการใช้ซ้ำได้? การแบ่งคลาสโดยเริ่มจากละเอียดไปหยาบจะทำให้ง่ายกว่าถ้าจะใช้ซ้ำ
- อะไรคือภาษาที่ไคลเอนต์ใช้พัฒนา? ภาษาที่บ่งชนิดไม่ได้ต้องการ IDispatch หนึ่งอันสำหรับทุกคำสั่ง
- ไคลเอนต์ใช้ออปเจกอย่างไร? ต้องสนับสนุนสำหรับอินเทอร์เฟซที่แน่นอน

ใครเป็นผู้สร้าง ?

- ผู้สร้างมีปริมาณเท่าไร? หลาย ๆ ผู้สร้างต่อหนึ่งชนิดคลาสจะต้องแยกหมวดหมู่คอมโพเนนต์ และต้องการการกำหนดที่ชัดเจนว่าอินเทอร์เฟซอะไรถูกสนับสนุน และอินเทอร์เฟซอะไรเป็นตัวเลือก

ประสิทธิภาพที่ต้องการ

- ขนาดฟุตพริ้นท์ (footprint)/ดาวน์โหลด (download) บนแผ่น
- หน่วยความจำโคdexของฟุตพริ้นท์ ต้องคำนึงถึงว่าคลาสจะถูกรวบรวมอย่างไร
- # ไคลเอนต์ที่เห็นชอบ คิดถึงการใช้ MTS สำหรับการสเกล และ MTS ต้องการการรวบรวมคอมโพเนนต์เป็น DLLs
- # ออปเจกที่เห็นชอบ ต้องเป็นตัวเดียว และเป็นออปเจกขนาดเล็ก

การใช้ที่ต้องการ

- ไคลเอนต์อยู่ที่ไหน? ไคลเอนต์ที่อยู่ไกลอาจต้องการข้อมูลรีจิสเตอร์
- คอมโพเนนต์อยู่ที่ไหน? ต้องเก็บคลาสที่จะเข้าถึงข้อมูลให้ใกล้ข้อมูล และเก็บคลาสที่จะจัด

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์และเผยแพร่โดยไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
หากบริการการแสดงผลให้ใกล้ส่วนติดต่อผู้ใช้ของท่านนั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ที่ต้องการแยกกระบวนการหรือไม่? ต้องคำนึงถึง DLL surrogates

การสร้างคอมโพเนนต์

ต้องเลือกเครื่องมือพัฒนาโดยใช้พื้นฐานบนข้อบังคับที่มี แล้วปรับให้ดีขึ้นเท่าที่เป็นไปได้

การเลือกเครื่องมือพัฒนา

- มีเครื่องมือหลายอย่างสำหรับการเขียนคอมโพเนนต์
- เครื่องมือมีความแตกต่างทั้งจุดแข็งและจุดอ่อน เพราะฉะนั้นต้องประเมินอย่างระมัดระวัง และเลือกให้ตรงกับข้อบังคับของการออกแบบมากที่สุด
- ป้องกันการโคดแบบ 'to - the - metal' สำหรับคอมโพเนนต์ที่มีข้อบังคับไม่ธรรมดา

ลักษณะเครื่องมือ

- ข้ามแพลตฟอร์ม
- ประสิทธิภาพ
- ขนาด
- จุดเด่นที่แสดงออกมา ทั้ง dcom และความปลอดภัย
- ความยืดหยุ่น

ความปลอดภัย

- สร้างการป้องกันความปลอดภัยลงในคอมโพเนนต์ อย่าให้ขึ้นอยู่กับการตั้งค่าของผู้ดูแลระบบ ถ้าช่วยได้
- คอมโพเนนต์ เรียก CoInitializeSecurity ที่ ระดับกระบวนการ และใช้ IServerSecurity สำหรับแต่ละข้อมูลการเรียกของไคลเอนต์
- MTS คอมโพเนนต์ ใช้ข้อมูลในคอนเทกซ์ (context) ออปเจก

เธรดดิ้ง (Threading)

- ใช้รูปแบบเธรดเดี่ยวถ้าเป็นไปได้
- คำนึงถึงการซิงโครไนส์ในอพาร์ทเมนต์ ที่ปรากฏการณ์คอขวดอาจเกิดขึ้น
- คำนึงถึงการเรียงเรียงปริมาณที่เกินของการเรียกข้ามอพาร์ทเมนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เฉพาะเพื่อการศึกษานี้เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีให้ดาวน์โหลดเป็นเอกสารและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีคนนำไปใช้
ตัวบรรจุ (Container) จะได้รับผลกระทบ เช่น IE4, IIS, MTS มีเครื่องมือบางตัวบังคับเลือก

การเรียบเรียง

- โดยปกติมาตรฐานการเรียบเรียงคือ MTS ต้องการการเรียบเรียงที่มีมาตรฐาน
- คำนี้ถึงการเรียบเรียงที่คัสตอม (Custom) เพื่อเจอข้อบังคับที่มีประสิทธิภาพ

การทำรุ่น

- ใช้ CLSID และ/หรือ ชื่อ DLL ที่เหมือนกัน ถ้าการสร้างใหม่เข้ากันได้กับ ไคลเอนต์ที่มีอยู่เดิม และผ่านการทดสอบที่ตรงข้ามกับ ไคลเอนต์ที่มีอยู่เดิม
- ใช้ CLSID และ/หรือ ชื่อ DLL ใหม่ ถ้าไม่สามารถรับประกันได้ว่าสามารถเข้ากันได้กับ ไคลเอนต์ที่มีอยู่เดิม

Multiple Inheritance กับ Nested Classes เป็นตัวเลือกการสร้างของ C++

- Multiple Inheritance จะง่ายที่จะสร้าง และใช้ IUnknown เดียว แต่ชื่อคลาสอาจก่อให้เกิดปัญหา
- Nested Classes ตัวพอบรรจุตัวชี้ หรือ สมาชิกของตัวแปร ใช้สำหรับผลรวม, tear-offs

การใช้ซ้ำ

- ใช้ซ้ำคอมโพเนนต์ที่มีอยู่ถ้าเป็นไปได้ ถ้าเป็นตัวแทน หรือเป็นตัวรวม
- ใช้ซ้ำบริการของระบบที่เป็นมาตรฐาน
- ใช้ซ้ำการสร้างต้นฉบับที่มีมาตรฐาน เช่น ALT, MFC, ฯลฯ

การปรับปรุงให้ดีขึ้น

- การเริ่มต้นที่ล่าช้า
- การรวมที่ล่าช้า
- เทียร์ออฟ (tear-off) อินเทอร์เฟซ ไม่ค่อยจะใช้อินเทอร์เฟซ หรืออินเทอร์เฟซที่ใช้ชั่วคราว (เช่น การเริ่มต้น)
- การรวมเทียร์ออฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

- ปรับพฤติกรรม ใช้การสร้างหลาย ๆ เมททอด

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 ODBC (Open Database Connectivity)

เมื่อสิบปีก่อน ความคิดที่จะใช้แอปพลิเคชันเข้าถึงข้อมูล โดยไม่ใช้การโปรแกรมและให้แอปพลิเคชันนั้นเข้าถึงฐานข้อมูลตัวอื่นได้อีก นับเป็นเรื่องที่ยากและแทบจะเป็นไปไม่ได้เลย ODBC ไม่เพียงแต่ทำให้เหตุการณ์ดังกล่าวเป็นไปได้ แต่ยังเป็นอย่างง่ายคายอีกด้วย

ODBC คืออะไร

ODBC ย่อมาจาก Open Database Connectivity เป็นมาตรฐานของแอปพลิเคชันเพื่อที่จะเข้าถึงข้อมูลของระบบจัดการฐานข้อมูลทั้งแบบเชิงสัมพันธ์และไม่ใช้แบบเชิงสัมพันธ์ ด้วย API ของ ODBC แอปพลิเคชันสามารถเข้าถึงข้อมูลที่เก็บอยู่ในเครื่องหลายระดับตั้งแต่เครื่อง PC มีเน็ตเวิร์กพีซีและเมนเฟรม แม้ว่าจัดเก็บในรูปแบบที่ต่างกันและมี API ที่ต่างกันไป เพื่อให้บรรลุจุดประสงค์ดังกล่าว สถาปัตยกรรมของ ODBC ต้องแบ่งระดับระหว่างแอปพลิเคชันและ DBMS โดยแอปพลิเคชันรับรู้เพียงการเรียกผ่าน API ของ ODBC เท่านั้น

ODBC มีพื้นฐานอยู่บนข้อกำหนด Call Level Interface (CLI) ของ X/Open SQL Group ซึ่งสถานะการณ์โดยทั่วไปคือ

- ซอฟต์แวร์ทุกตัวของไมโครซอฟท์สามารถใช้งาน ODBC ได้ และสนับสนุน ODBC
- การสนับสนุน ODBC ไม่จำกัดเพียงไมโครซอฟท์สามารถใช้งาน ODBC ได้ และสนับสนุนด้วย
- ODBC มีอยู่บนหลายแพลตฟอร์ม เช่น UNIX, OS/2 เป็นต้น

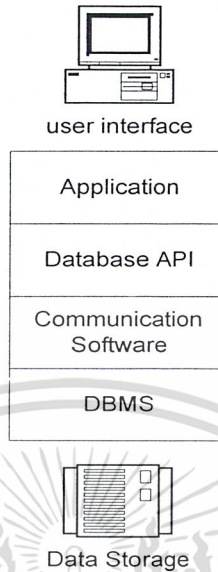
ความต้องการ ODBC เกิดจากปัญหาดังต่อไปนี้

- ผู้ใช้ต้องการความอ่อนตัว และเครื่องมือที่ใช้ทั่วไปไม่จำกัดเฉพาะฐานข้อมูลของผู้ผลิตรายใดรายหนึ่ง
- ผู้ใช้ต้องการแอปพลิเคชันทางธุรกิจที่พัฒนาและแก้ไขได้รวดเร็ว
- ผู้ต้องการผลลัพธ์ที่ทำงานได้และเป็นเทคโนโลยีที่เป็นไปได้
- ผู้ใช้ต้องการผลลัพธ์ที่ใช้งานได้ในทุก DBMS
- ผู้ใช้ต้องการประสิทธิภาพสูงสุด
- ผู้ใช้ต้องการความเรียบง่าย
- ผู้พัฒนาไม่ต้องยึดติดกับ DBMS ตัวใดตัวหนึ่ง
- ผู้พัฒนาต้องการแอปพลิเคชันทางธุรกิจโดยใช้เทคนิคการโปรแกรมแบบ RAD (Rapid Application Development) หรือใช้หลักการของวิศวกรรมโปรแกรมมิ่ง
- ผู้พัฒนาต้องการแอปพลิเคชันที่เข้าถึงข้อมูลได้หลายตัว
- ผู้พัฒนาต้องการฟังก์ชันเต็มรูปแบบและประสิทธิภาพสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถาปัตยกรรมและโมเดลของ ODBC

ฐานข้อมูลตามแบบธรรมดาเดิมประกอบไปด้วยดังรูปที่ 10

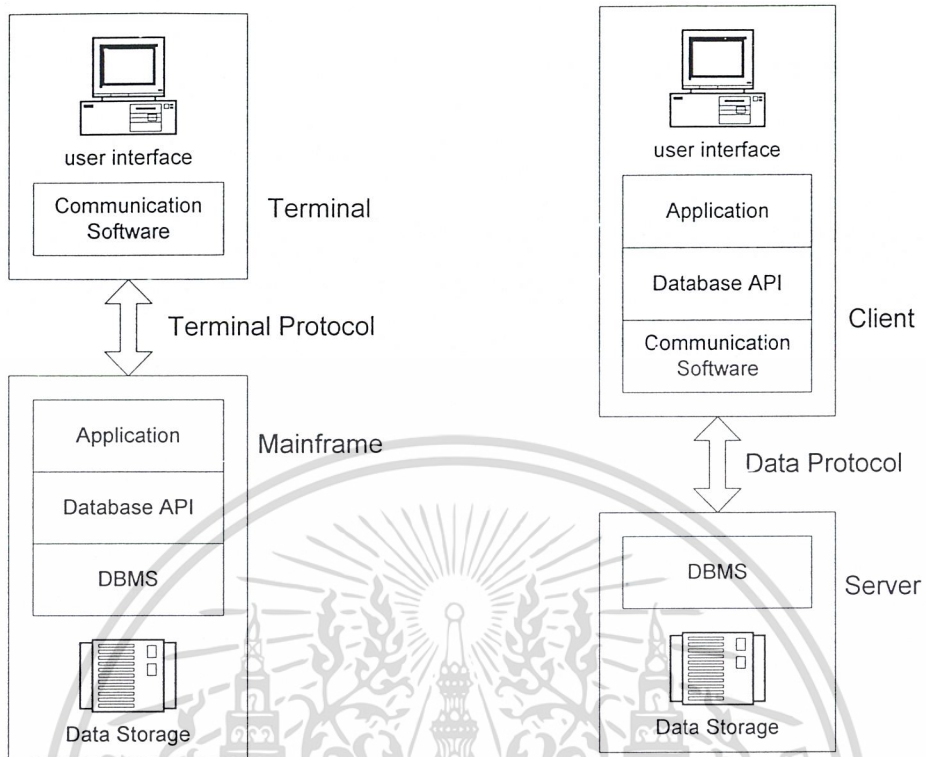


รูปที่ 10 ฐานข้อมูลตามแบบธรรมดาเดิม

- User Interface เป็นสิ่งที่ผู้ใช้สุดท้าย(End User)เห็นบนหน้าจอ
- Application เป็นโปรแกรมที่ทำให้ผู้ใช้สุดท้ายทำงานได้ลุล่วง โดยใช้ข้อมูลที่ต้องการที่ใช้จากเซิร์ฟเวอร์
- Database API เป็นส่วนโค้ดที่จะขอรับการร้องขอและจะให้ผลลัพธ์แก่แอปพลิเคชันเมื่อเข้าถึงฐานข้อมูล
- Network/Communication Software เป็นส่วนที่ทำให้โปรแกรมไคลเอ็นต์ติดต่อกับเซิร์ฟเวอร์ได้และในทำนองกลับกัน
- DBMS เป็นโปรแกรมเซิร์ฟเวอร์เพื่อรองรับความต้องการจากเครื่องไคลเอ็นต์สำหรับเข้าถึงฐานข้อมูล
- Data Storage เป็นที่ที่ข้อมูลแท้จริงเก็บไว้อยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปรียบเทียบระหว่างการรวมศูนย์กับไคลเอนต์เซิร์ฟเวอร์



รูปที่ 11 เปรียบเทียบลักษณะรวมศูนย์ (ซ้าย) กับลักษณะไคลเอนต์/เซิร์ฟเวอร์ (ขวา)

ลักษณะการโปรแกรมโดยใช้ ODBC

ตามปกติการโปรแกรมในระบบเชิงสัมพันธ์ สามารถทำได้โดยใช้

- Embedded SQL เป็นการเขียนโปรแกรมในภาษาเช่น C , C++ , COBOL โดยผนวกรวมคำสั่งของ SQL ลงไปได้ แต่ต้องผ่านพรีคอมไพเลอร์เสียก่อนจึงทำการคอมไพล์ตามปกติได้ โดยแบ่งเป็น Static SQL และ Dynamic SQL
- Call level Interface เป็นการเขียนโปรแกรมเรียกผ่านฟังก์ชันของ Database API เพื่อทำการดังต่อไปนี้
 - เชื่อมต่อไปยังเซิร์ฟเวอร์ฐานข้อมูล
 - ส่งประโยค SQL ไปยังเซิร์ฟเวอร์
 - ประมวลผล ผลลัพธ์ที่ได้จากการตอบประโยค SQL ที่ส่งให้เซิร์ฟเวอร์ไม่ว่าจะเป็นข้อผิดพลาด สถานะข้อมูล หรือข้อมูล

ลักษณะทั่วไปของ CLI คล้าย Dynamic SQL แต่ต่างกันที่ไม่มีการใช้พรีคอมไพเลอร์ซึ่ง ODBC จะเป็นลักษณะของ Call Level Interface

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถาปัตยกรรมของ ODBC

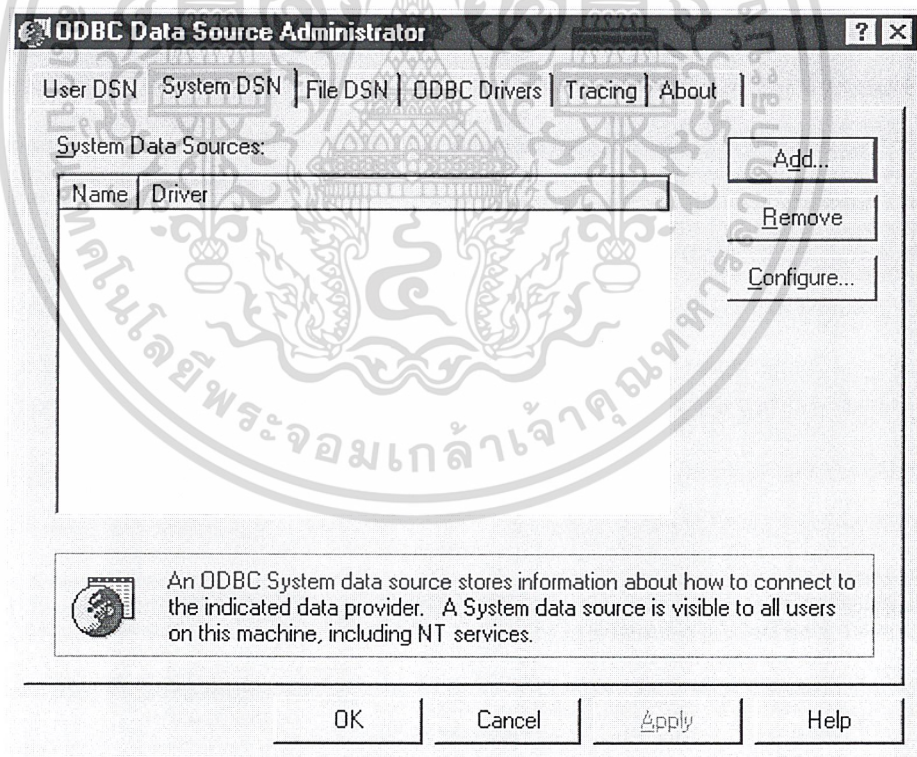
สถาปัตยกรรมของ ODBC ถูกสร้างอยู่บนโมเดลของไคลเอ็นต์เซิร์ฟเวอร์ซึ่ง

- สนับสนุน API มาตรฐาน
- ให้ฟังก์ชันครบครันสำหรับในละ DBMS ที่แตกต่างกัน
- ให้ประสิทธิภาพใกล้เคียงกับ API โดยแท้ของ DBMS

ส่วนประกอบของสถาปัตยกรรม ODBC

- แอปพลิเคชัน
- ไดรเวอร์แมนเนเจอร์ เป็นตัวจัดการไดรเวอร์ทั้งหมดของ ODBC และฐานข้อมูล
- ไดรเวอร์ เป็นตัวกลางในการแปลง ODBC API ให้อยู่ในรูปที่ DBMS เข้าใจ
- ดาต้าซอร์ส เป็นแหล่งข้อมูลที่จะเป็นฐานมูลหลัก

2.8 การเชื่อมฐานข้อมูลแบบออราเคิลเข้ากับโปรแกรม

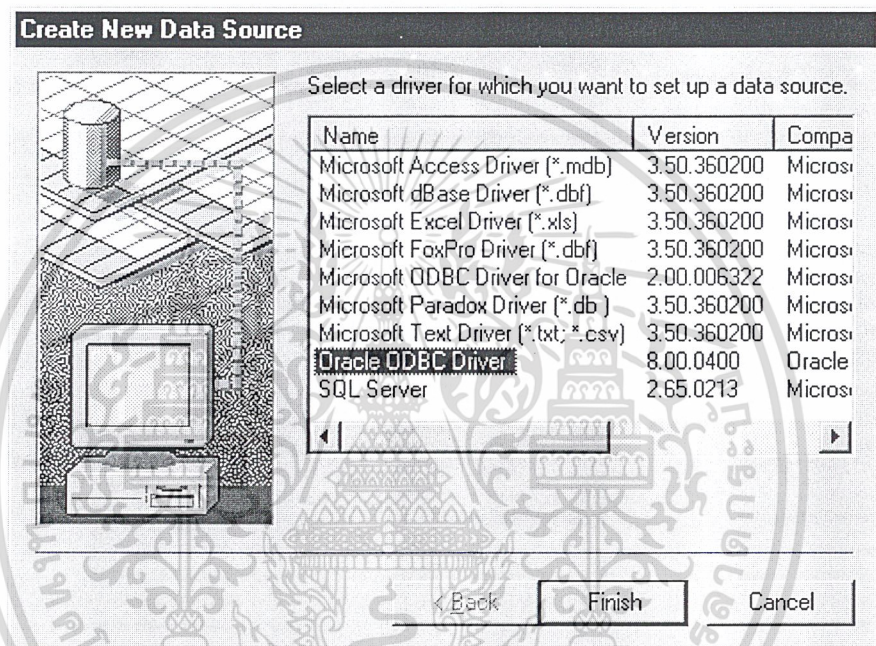


รูปที่ 12

ในทำให้โปรแกรมที่สร้างขึ้นสามารถเข้าถึงข้อมูลที่เป็นข้อมูลของออราเคิลได้นั้น ต้องเครื่อง
 เอกสารที่เลือก ODBC ไดรเวอร์ของออราเคิลจะก่อนจึงจะทำงานได้ ซึ่งการลงไดรเวอร์นั้นจำเป็นสำหรับทุก
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฐานข้อมูลที่เราต้องการติดต่อ หลังจากนั้นเราก็ต้องสร้าง System Data Source Name เพื่อให้เครื่องของเรา
นั้นสามารถติดต่อกับฐานข้อมูลนั้นได้ โดยวิธีการสร้างมีดังต่อไปนี้

1. เลือก Data Source Administrator จาก Control Panel
2. เลือกแท็บ System DSN (มาจาก Data Source Name) จะได้หน้าจอดังรูปที่ 12
3. ให้เลือก Add แล้วจะได้หน้าจอดังรูปที่ 13 เพื่อสร้าง Data source อันใหม่
4. เลือก ODBC ไดรเวอร์ที่ต้องการถ้าเลือกออราเคิลไดรเวอร์จะได้ดังรูปที่ 14



รูปที่ 13

5. จากนั้นก็ตั้งค่าต่าง ๆ ดังต่อไปนี้
 - 5.1 Data Source Name เป็นชื่อซึ่งจะตั้งชื่ออะไรก็ได้ตามความพอใจ
 - 5.2 Discription เป็นคำอธิบายใส่อะไรก็ได้
 - 5.3 Service Name เป็นชื่อเครื่องที่ตั้งของฐานข้อมูลออราเคิล
 - 5.4 UserID เป็นชื่อผู้ใช้ของฐานข้อมูลออราเคิลตรงนี้ทิ้งว่างไว้ก็ได้

เมื่อสร้าง System Data Source Name เสร็จแล้วก็พร้อมที่จะให้โปรแกรมติดต่อกับฐานข้อมูลโดย
ในการติดต่อนั้นต้องใช้ Function OpenDataBase โดยเรียกแบบนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 14

```
sConnect = "ODBC;DSN=" & DSN & ";"
sConnect = sConnect & "UID=" & Name & ";"
sConnect = sConnect & "PWD=" & Password & ";"
Set Db(Co) = OpenDatabase("", 0, 0, sConnect)
```

โดย DSN จะเป็นชื่อ System Data Source Name ส่วน Name เป็นชื่อผู้ใช้ และ Password เป็นรหัสผ่านของผู้ใช้ เสร็จแล้วก็จะสามารถติดต่อกับฐานข้อมูลที่เป็นออราเคิลได้ จะเห็นได้ว่าเราสามารถติดต่อข้อมูลจากเครื่องอื่นที่ไม่ใช่เครื่องที่รันโปรแกรมได้ และยังทำให้สามารถติดต่อกับข้อมูลจากหลาย ๆ เครื่องได้

2.9 การออกแบบการทำงานของโปรแกรม Database on Internet

เป็นการดึงความสามารถในการจัดการระบบฐานข้อมูลให้สามารถจัดการผ่านสื่ออินเทอร์เน็ตได้ โดยใช้ความสามารถของ ActiveX โดยที่ Server จะจัดการกับฐานข้อมูลแล้วส่งข้อมูลของฐานข้อมูลที่ Client ต้องการมายัง Client เพื่อแสดงผลโดยอัตโนมัติ โดยการทำงานของโปรแกรมจะแบ่งเป็น 2 ส่วนที่สำคัญได้แก่ ส่วนที่ทำงานบน Server กับส่วนที่ทำงานบน Client

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่ทำงานบน Server

เป็นส่วนที่ติดต่อกับฐานข้อมูล โดยที่การติดต่อกับฐานข้อมูลในการโปรแกรมโดยใช้ภาษา Visual Basic นั้นสามารถทำได้ 2 วิธีคือการทำผ่านกลุ่มของ Data Access Object (DAO) ที่มีให้ใช้ใน Visual Basic กับการทำผ่าน Data Access control ที่ถูกเขียนขึ้นมาเพื่อให้สามารถทำงานเกี่ยวกับฐานข้อมูลได้ง่ายขึ้น ซึ่งคอนโทรลนี้จะไปติดต่อกับ DAO เองอีกที สำหรับโปรแกรมนี้เราใช้วิธีติดต่อกับ DAO โดยตรงไม่ผ่าน Data Access Control

ส่วนที่ทำงานบน Client

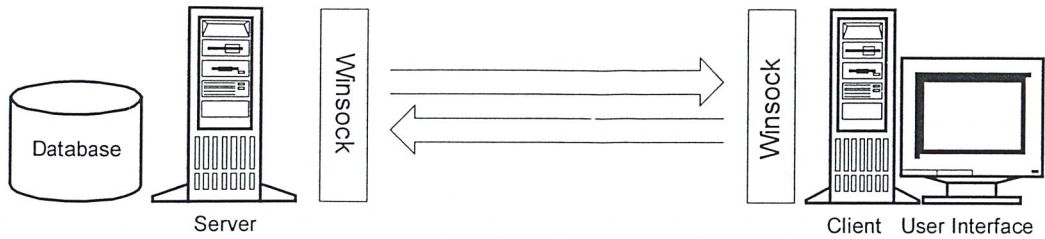
เป็นส่วนที่ใช้ในการติดต่อกับผู้ใช้ โดยจะรับว่าผู้ใช้ต้องการอะไร แล้วส่งไปให้ส่วนที่ทำงานบน Server จัดการ จากนั้นก็รับข้อมูลจากส่วน Server นำมาแสดงผลให้ผู้ใช้ โดยส่วนประกอบหลักของส่วนนี้คือการแสดงผลตามข้อมูลซึ่งเราใช้ User Interface Control ที่มีในภาษา Visual Basic นำมาประกอบกันแล้ว Compile ให้อยู่ในรูปของ ActiveX Control แล้วนำไปวางบนเพจที่เขียนด้วยภาษา HTML โดยเราจะถือว่าคอนโทรลนี้เป็น Object หนึ่งที่วางอยู่บน HTML เท่านั้นไม่สนใจการทำงานภายในคอนโทรล ด้วยวิธีนี้ผู้ใช้ก็สามารถเรียกใช้งานโปรแกรมจากที่ใดก็ได้ที่สามารถติดต่อกับอินเทอร์เน็ตได้ ในการ Compile นี้เราจะได้ไฟล์ .cab ด้วยเพื่อในกรณีที่ผู้ใช้ยังขาด Component ใด ก็สามารถดาวน์โหลดไปได้หากมีครบอยู่แล้วก็ไม่ต้องดาวน์โหลดอีก ซึ่งวิธีนี้จะทำให้การทำงานเร็วขึ้นในการเรียกใช้ครั้งต่อ ๆ มา

การติดต่อกันระหว่างส่วน Server กับส่วน Client

การส่งข้อมูลของฐานข้อมูลจาก Server ไปยัง Client กับความต้องการของผู้ใช้จาก Client ไปยัง Server นั้นเราใช้วิธีส่งผ่าน Winsock Control ซึ่งมีให้ใช้ในภาษา Visual Basic โดยการทำงานของโปรแกรมเริ่มจาก Server ส่งรายชื่อฐานข้อมูลและรายชื่อตารางในฐานข้อมูลไปยัง Client เพื่อให้ผู้ใช้ได้เลือกใช้ฐานข้อมูลและตารางที่ต้องการใช้ เมื่อผู้ใช้เลือกแล้ว Server จะข้อมูลของตารางนั้นให้ผู้ใช้ ถ้าตารางที่ผู้ใช้ดูอยู่มีความสัมพันธ์กับตารางอื่น ก็จะให้ผู้ใช้เลือกดูข้อมูลอื่นของ Record นั้น ๆ ได้

การทำงานของโปรแกรมจะแบ่งเป็น 2 ส่วน คือ ส่วนที่ทำงานบนเครื่อง Server กับส่วนที่ทำงานบนเครื่อง Client โดยที่ส่วนที่ทำงานบนเครื่อง Server นั้นจะเป็นส่วนที่จะจัดการกับฐานข้อมูล ส่วนส่วนที่ทำงานบนเครื่อง Client นั้นจะจัดการกับส่วนที่ติดต่อกับผู้ใช้ โดยส่วน Client จะรับความต้องการจากผู้ใช้ว่าต้องการทำอะไรแล้วจึงส่งความต้องการเหล่านั้นยัง Server โดยผ่าน Winsock Control เมื่อส่วนที่เป็น Server ได้รับแล้วจึงจัดการกับฐานข้อมูลตามนั้น แล้วส่งคำตอบกลับที่ได้ผ่าน Winsock Control เพื่อให้ส่วน Client แสดงผลให้ผู้ใช้ดู ดังรูปที่ 15

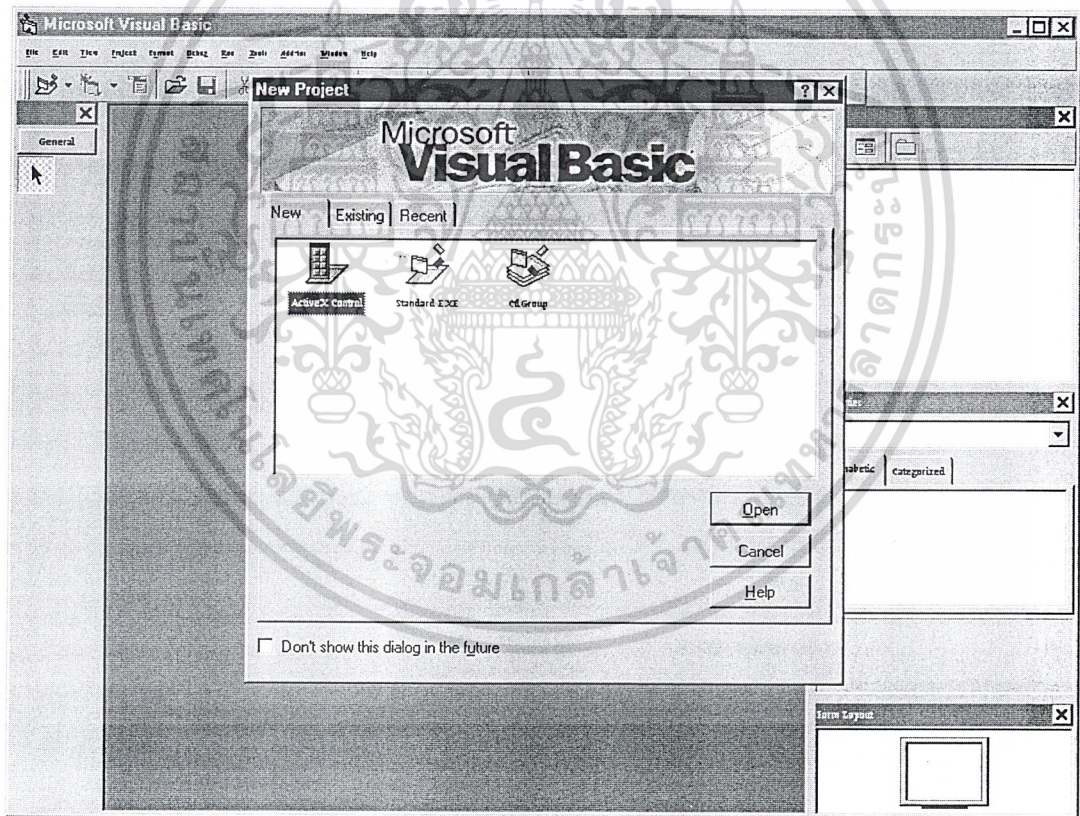
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 15

สำหรับการสร้างนั้นเราจะใช้ Visual Basic 5 เขียนโปรแกรมส่วน Server เหมือนเขียน Application บน Windows ตัวหนึ่ง สำหรับในส่วนของ Client นั้น เราจะต้อง Compile ให้อยู่ในรูปของ Control ก่อนแล้วนำไปวางบนไฟล์ HTML เพื่อให้สามารถใช้บน Internet ได้

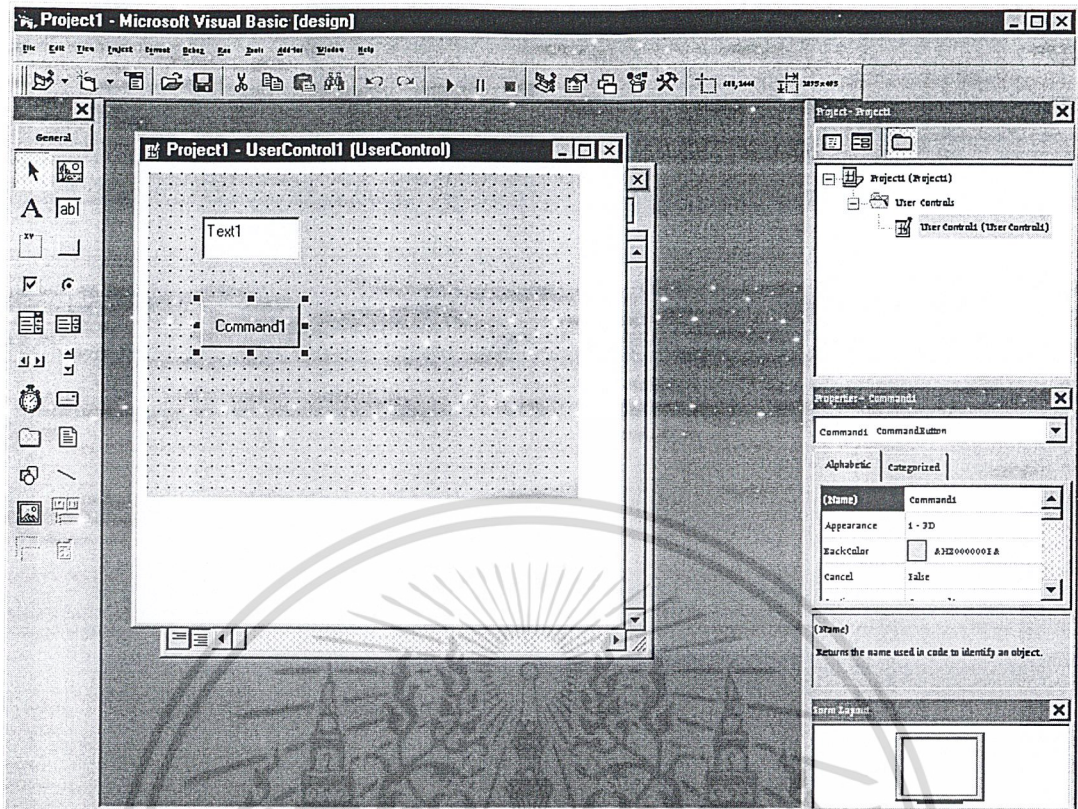
2.10 การสร้าง ActiveX ด้วย VBCCE(Visual Basic Custom Control Edition)



รูปที่ 16 เริ่ม Visual Basic Custom Control Edition

1. เริ่มโปรแกรม VBCCE แล้วมันจะถามถึงชนิดของโปรเจกที่เราจะสร้าง ดังรูปที่ 16
2. เลือก "Active Control" แล้ว VBCCE จะเปิดฟอร์ม UserControl อันใหม่ให้พร้อมสำหรับการออกแบบ
3. ตั้งชื่อให้โปรเจกและฟอร์ม วาง Object ที่ต้องการใช้ลงบนฟอร์ม ตั้งค่า Property ตามต้องการเช่นใน

รูปที่ 17 เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเท่านั้น เมื่อผู้ยืมได้รับเอกสารฉบับนี้ให้นำค่า
ไม่ว่ากรณใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 17 แสดงการวาง Object ลงบนฟอร์ม

การสร้าง Code ใน VBCCE

สำหรับการสร้าง Code ใน VBCCE นั้น ถ้าเราต้องการให้มีอะไรเกิดขึ้นกับ Control เมื่อมีเหตุการณ์ใด ๆ เกิดขึ้นกับ Object ใน Control นั้น ให้ Double Click ที่ Object นั้น หรือถ้าต้องการให้มีอะไรเกิดเมื่อมีเหตุการณ์ใด ๆ เกิดขึ้นกับตัว Control ที่เราสร้างนั่นเองก็ให้ Double Click ที่ ตัวฟอร์มของ Control นั้นเลย เราก็จะได้ Code สำหรับการสร้างโดยมันจะขึ้นว่า

```
Private Sub Command1_Click ()
```

```
End Sub
```

เราก็จะสามารถใส่ Code ที่เราต้องการลงใน Sub ได้เลย

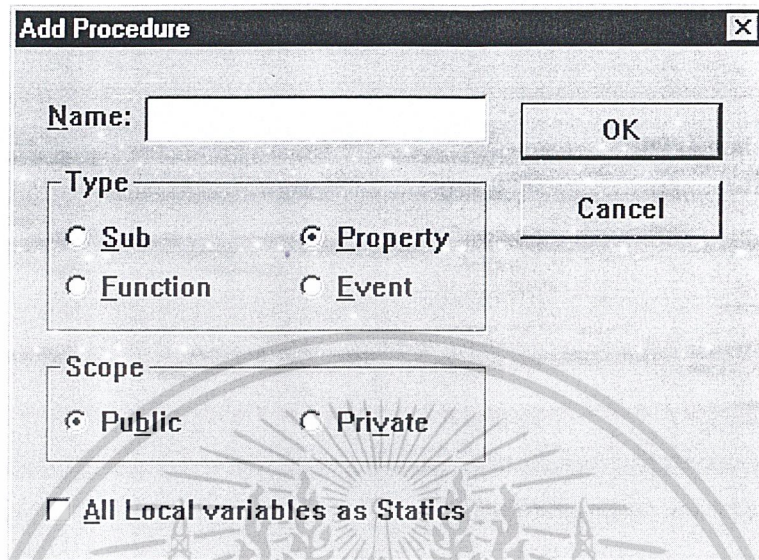
การสร้าง Property ให้ Control

1. เลือก Tools, Add Procedure จะได้ Add Procedure dialog ดังรูปที่ 18
2. หลังจากเปิด Dialog แล้ว ให้ใส่ชื่อ Procedure แล้วเลือก Property เสร็จแล้วกด OK
3. เมื่อเลือกเสร็จแล้วมันจะสร้าง Code ให้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 Public Property Let PropertyName (ByVal vNewValue As Variant)
 ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
 End Property

Public Property Get PropertyName () As Variant

End Property



รูปที่ 18 เพิ่ม Procedure ให้ Control

4. กำหนดตัวแปรเพื่อเก็บค่าของ Property นั้น เช่น

Option Explicit

Private PropertyTemp As Variant

Public Property Let PropertyName (ByVal vNewValue As Variant)

PropertyTemp = vNewValue

End Property

Public Property Get PropertyName () As Variant

PropertyName = PropertyTemp

End Property

5. เมื่อเรากำหนดค่าต่าง ๆ แล้วเราก็จะได้ Property ของ Control ตามต้องการ

การสร้าง Event ให้ Control

1. เลือก Tools, Add Procedure
2. ใส่ชื่อ Procedure แล้วเลือก Event เสร็จแล้วกด OK
3. เมื่อเลือกเสร็จแล้วมันจะสร้าง Code ให้ดังนี้

Public Event EventName

เอกสารนี้เป็นเอกสารทบทวนวิชาสำหรับครูผู้ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

4. เมื่อเราต้องการให้ Event นี้เกิดขึ้นเมื่อไรก็ให้ใส่ Code

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ต่อและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RaiseEvent EventName

ลงไปใน Code ก็จะทำให้ Event นั้น ๆ เกิดขึ้นตรงนั้น

การสร้าง Method ให้ Control

1. เลือก Tools, Add Procedure
2. ใส่ชื่อ Procedure แล้วเลือก Sub เสร็จแล้วกด OK
3. เมื่อเลือกเสร็จแล้วมันจะสร้าง Code ให้ดังนี้

```
Public Sub SubName
```

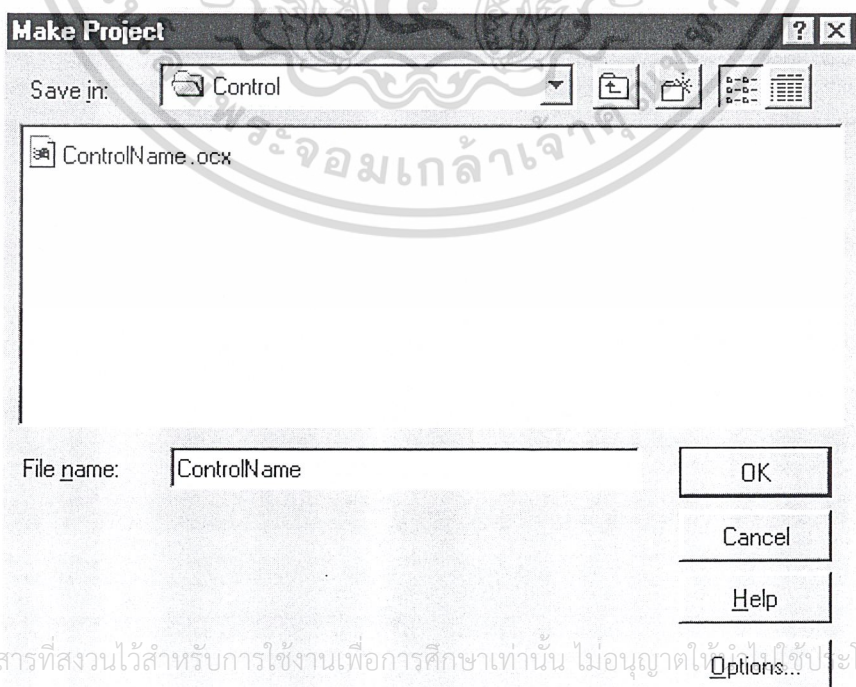
```
End Sub
```

4. ถ้าเราต้องการให้ Control ทำอะไรเมื่อผู้ใช้ได้เรียก Method นี้ก็ให้ใส่ลงไปใน Sub ได้เลย

การ Compile Code ให้เป็น Control

หลังจากที่ได้สร้าง Code ทั้งหมดของ Control เสร็จแล้วเราก็พร้อมที่จะ Compile Code ทั้งหมดให้เป็น Control เพื่อที่เราจะได้เรียกใช้ Control นี้ในงานอื่น ๆ ต่อไป

1. เลือก File ในเมนู จากนั้นเลือก Make ControlName.ocx ใน File เมนูจะได้ Dialog ดังรูปที่ 19
2. เมื่อได้ Compile Active Control เสร็จแล้วโปรแกรมจะทำการ Register เพื่อให้สามารถนำไปใช้งานได้ทั้งใน Visual Basic, Delphi, ActiveX Control Pad และ โปรแกรมอื่น ๆ ที่สนับสนุนการทำงานของ ActiveX



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
รูปที่ 19 Dialog สำหรับการสร้าง ActiveX Control จาก Code ที่ได้เขียนมา

2.11 การใช้ VBScript ควบคุมการทำงานของ ActiveX Control

VBScript เติบโตมาจากภาษา Visual Basic ภาษาที่ใช้กันอย่างแพร่หลายในช่วงหลายปีที่ผ่านมา Visual Basic เป็นพื้นฐานสำหรับภาษาสคริปต์ใน Microsoft Office, Word, Access, Excel, และ PowerPoint Visual Basic เป็นภาษาที่ใช้พื้นฐานของคอมพิวเตอร์ การสร้างโปรแกรม Visual Basic ต้องสร้างโดยการวางคอมโพเนนท์ลงบนฟอร์มแล้วใช้ภาษา Visual Basic เพื่อผูกมันไว้ด้วยกัน จะพบว่า Visual Basic เปรียบเสมือนเป็นปู่ของ ActiveX control จาก Visual Basic control (VBX) Visual Basic control จะแบ่งการติดต่อทั่วไปซึ่งคิดตามมันมาลงบนฟอร์มของ Visual Basic ด้วย นี่คือนี่หนึ่งในหลายการใช้ของซอฟต์แวร์ที่เป็น component-base VBX เป็นการเริ่มไปสู่คอนโทรล OLE (OCX) แล้วกลายมาเป็น ActiveX และ VBScript ก็ได้ต้นแบบมาจาก Visual Basic

สถาปัตยกรรมของสคริปต์

การที่จะศึกษา VBScript จะต้องรู้เรื่องต่อไปนี้

- แพลตฟอร์มของซอฟต์แวร์ซึ่งสนับสนุน VBScript
- ต้นแบบการเขียนโปรแกรมของ VBScript
- ชนิดตัวแปร
- การดำเนินงาน
- ฟังก์ชันการทำงาน
- โครงสร้างการไหลของโปรแกรม
- ส่วนการติดต่อผู้ใช้ของ VBScript

แพลตฟอร์มซอฟต์แวร์ที่สนับสนุน

มีเพียงบราวเซอร์ Internet Explorer ของไมโครซอฟท์เท่านั้นที่สนับสนุนการทำงานของ VBScript นอกจากนั้นก็ยังมีบราวเซอร์หลักได้แก่ Netscape ซึ่งถ้าไม่มีโปรแกรมเสริมก็ไม่สามารถทำงานกับ VBScript หรือใช้ ActiveX object ได้ แต่ถ้าอยากใช้ Netscape ก็สามารถหาโปรแกรมเสริมเพื่อให้สนับสนุนการทำงานของ ActiveX และ VBScript

ในปัจจุบันไมโครซอฟท์เป็นผู้ควบคุมมาตรฐานทั้ง ActiveX และ VBScript อย่างไรก็ตามไมโครซอฟท์มีความตั้งใจที่จะทำให้มาตรฐาน ActiveX เป็นมาตรฐานของอุตสาหกรรม

ถ้าจะต้องการใช้ ActiveX และ VBScript ตอนนี้ก็ต้องใช้ Internet Explorer เป็นแพลตฟอร์มของซอฟต์แวร์ของกลุ่มเป้าหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้นแบบการเขียนโปรแกรมของ Visual Basic

VBScript เป็นภาษาที่ใช้หลัก Procedural สำหรับใครที่ใหม่สำหรับการเขียนโปรแกรมภาษาที่ใช้หลัก Procedural จะใช้การทำงานย่อยเป็นหน่วยพื้นฐาน จะเห็นได้ว่ามีหน่วยพื้นฐานย่อยหลายตัวที่ใช้ทำงานบนเว็บเพจ ตัวอย่างเช่น

```
Sub LoadMe()
    Alert "This Space for Rent"
    Answer = Confirm ("Launch the Shuttle?")
    String1 = Prompt ("Enter Here", "Defaulted Text")
    Alert "string1"
End sub
```

ทุกคำสั่งที่อยู่ระหว่าง Sub กับ end sub คือขั้นตอน (procedure) สำหรับการทำงาน VBScript และ ภาษาที่ใช้หลักทำเป็นขั้นตอนในทั่วไปจะได้เปรียบตรงใ้ง่ายและแสดงผลเร็ว ข้อเสียก็คือถ้าโครงการมีขนาดใหญ่จะยากที่จะจัดการ เพราะว่าเว็บเพจจะต้องทำงานผ่านเครือข่ายมันจึงต้องใส่ไปกับเวลาที่ใ้ใช้ และเนื่องจากโคดมมีขนาดเล็ก การใช้ VBScript จึงเป็นทางเลือกที่เหมาะสม

ชนิดตัวแปร

ตัวแปรที่ใช้ใน VBScript เป็นตัวแปรที่แบ่งชนิดแบบหลวม ๆ (loosely type) สิ่งนี้หมายความว่าเราสามารถเพิ่มค่า int ไปในตัวแปร real โดยไม่ต้องเปลี่ยนชนิดของตัวแปร ในภาษาที่ใช้ชนิดแบบแข็ง (strongly typed) จะไม่ยอมให้มีการเพิ่มค่าที่เป็น int ลงในค่าที่เป็น float ดังนั้นในการใช้ VBScript จึงสามารถเขียนสคริปต์อย่างนี้ได้

```
Sub LoadMe()
    String1 = "42"
    Window.document.open
    Window.document.writeln("<PRE>")
    Window.document.writeln(string1)
    String2 = 2 + String1
    Window.document.writeln(string2)
    Window.document.writeln("</PRE>")
    Window.document.close
End sub
```

ผลจากสคริปต์นี้ก็จะใ้

42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
44
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดีของตัวแปรแบบนี้คือไม่ต้องจำฟังก์ชันเปลี่ยนชนิดตัวแปรทั้งหลาย แต่ข้อเสียของมันก็คือ การที่ไม่รู้ว่าภาษาจะตีความอย่างไร ตัวตีความจะไม่รู้ว่าตัวแปรเหล่านี้ต้องการเปลี่ยนชนิดหรือไม่

การดำเนินงาน

การดำเนินงาน (Operator) จะเป็นตัวอธิบายว่าจะทำอะไรกับตัวแปรและฟังก์ชัน มีชนิดของการดำเนินงานอยู่ 5 ชนิด ได้แก่ การกำหนด (assignment), การคำนวณ (math), ตัวหนังสือ(string), การเปรียบเทียบ (comparison), เป็นสิ่ง (object)

การกำหนด (Assignment Operator)

เราใช้การดำเนินงานโดยกำหนดทุกครั้งที่ต้องการเปลี่ยนค่าในตัวแปร มันเป็นหัวใจของการเขียนโปรแกรม VBScript มีสองวิธีในการกำหนดคือใช้เครื่องหมายเท่ากับ (=) และการใช้คำสั่ง Set

การคำนวณ (Math Operator)

การคำนวณก็เป็นการดำเนินการที่สำคัญอีกอย่างหนึ่ง ถ้าต้องการที่จะรวมค่าของตัวแปรกับค่าตัวแปรเอง หรือค่าคงที่อื่น ๆ เหมาะกับการทำงานกับตัวเลข

ตัวหนังสือ (String)

มีการดำเนินงานเพียงอย่างเดียวที่เกี่ยวกับตัวอักษร ได้แก่การใช้ตัวและ (&) การใช้ตัวดำเนินงานนี้จะทำให้เชื่อมตัวหนังสือจากตัวแปรต่าง ๆ รวมเป็นตัวแปรเดียว

การเปรียบเทียบ (Comparison)

เป็นการดำเนินงานที่ใช้หาความสัมพันธ์ระหว่างตัวแปร มันครอบคลุมทั้งการเทียบระหว่างตัวเลขกับตัวเลข, ตัวเลขกับตัวอักษร, และตัวอักษรกับตัวอักษร

ฟังก์ชันการทำงานที่มากับโปรแกรม

VBScript มีฟังก์ชันที่ผู้สร้างซึ่งเราควบคุมได้อยู่แล้วและฟังก์ชันที่มีอยู่แล้วกับโปรแกรม โดยฟังก์ชันที่มากับโปรแกรมมีหลายเรื่องรวมทั้งฟังก์ชันการเปลี่ยนชนิดตัวแปร, ฟังก์ชันที่ใช้กับวันเวลา, ฟังก์ชันที่ใช้คำนวณ,อาร์เรย์,ตัวหนังสือ, และตัวแปร

ตัวแปลง (Conversion) เป็นฟังก์ชันที่ใช้เพื่อการแปลงค่าของตัวแปร และเพื่อเป็นการกำหนดให้ชัดเจนสำหรับชนิดของตัวแปร

วัน/เวลา (Date/Time) เป็นฟังก์ชันที่ใช้เกี่ยวกับเวลา โดยใช้เพื่อหาเวลา กำหนดเวลา และอื่น ๆ

คำนวณ (Math) เป็นฟังก์ชันที่ใช้เพื่อการคำนวณต่าง ๆ ที่เกิดขึ้น

ฟังก์ชันอาร์เรย์ (Array Function) เป็นการใช้ฟังก์ชันกับตัวแปรที่เป็นอาร์เรย์ โดยจะทำการกำหนดขอบเขตของอาร์เรย์ ลบค่าในอาร์เรย์

ฟังก์ชันสตริง (String Function) เป็นฟังก์ชันที่จัดการตัวแปรที่เป็นตัวหนังสือทั้งค้นหาข้อความตัดต่อข้อความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า และอื่น ๆ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การไหลของโปรแกรม

มีโครงสร้างในการควบคุมการไหลของโปรแกรมอยู่ 5 แบบ

Do...Loop

ถ้าต้องการทำอะไรอย่างน้อยหนึ่งครั้งก่อนค่อยคิดว่าจะทำอีกหรือไม่ก็ควรจะใช้คำสั่ง Do...Loop ตามด้วย Until ถ้าต้องการต้องการคิดก่อนว่าจะทำหรือไม่ก็ต้องเลือกใช้ Do...Loop แล้วใส่ While ตามหลัง Do ใช้ Exit...Do เพื่อออกจากลูป ณ เวลาที่ต้องการได้เลย

For...Next

คำสั่งนี้จะทำให้โปรแกรมทำงานให้ครั้งตามตัวเลขที่กำหนด โดยส่วน For จะกำหนดเลขที่เริ่มต้นนับ ส่วน To จะกำหนดเลขที่สิ้นสุดการนับ โดยจะมีคำสั่ง Step เป็นตัวบอกระยะห่างของแต่ละการนับ

If...Then...Else

ส่วน If จะเป็นตัวเช็คเงื่อนไขว่าจะทำงานตามคำสั่งหลัง Then หรือไม่ หรือจะทำตามคำสั่งหลัง Else Select Case คำสั่งนี้จะเช็คค่าในตัวแปรแล้วจึงเลือกทำตามค่าที่อยู่ในตัวแปรนั้น

While...Wend

คำสั่งนี้เป็นคำสั่งที่มีมาตั้งแต่ Visual Basic เวอร์ชันเก่า มันจะทำคำสั่งที่อยู่ระหว่าง While กับ Wend ครอบไว้ตามที่เงื่อนไขยังเป็นจริงอยู่

ส่วนการติดต่อผู้ใช้

VBScript มีส่วนติดต่อกับผู้ใช้อยู่สองส่วนซึ่งเต็มไปด้วย Alert และ Prompt MsgBox จะใช้ง่ายกว่า Alert มันจะให้กำหนดไตเติลและบอกกลุ่มของปุ่ม มีบางครั้งที่ต้องใช้ Alert หรือ Confirm ก่อนที่จะใช้ MsgBox โดยในขั้นแรก MsgBox จะต้องการการตั้งค่า ตั้งแต่ไม่มีค่าคงที่ใน VBScript จะต้องรู้โค้ดตัวเลขสำหรับปุ่มที่แตกต่างกัน ขั้นที่สองถ้าในสลิปครั้งจะต้องสร้าง MsgBox ซึ่งคล้าย Alert

2.12 การนำ Winsock Control ไปใช้เพื่อการติดต่อระหว่างไคลเอนต์/เซิร์ฟเวอร์

เป็นคอนโทรลที่ทำให้เครื่องคอมพิวเตอร์ใด ๆ บนระบบอินเทอร์เน็ตสามารถติดต่อแลกเปลี่ยนข้อมูลข่าวสารต่าง ๆ ได้ โดยใช้ Internet Protocol (IP) ในการติดต่อ โดยมันจะสามารถใช้ได้สองโหมดในการติดต่อ ได้แก่ Transmission Control Protocol (TCP) กับ User Datagram Protocol (UDP) โดยโหมดแรกนั้นจะทำงานคล้ายกับการทำงานของโทรศัพท์ คือมีเครื่องหนึ่งต้องคอยรับฟังการติดต่อเมื่ออีกเครื่องหนึ่งติดต่อมา เครื่องแรกจะต้องรับสายก่อนถึงจะเริ่มคุยกันและแลกเปลี่ยนข่าวสารได้ โดยก่อนติดต่อนั้นเครื่องแรกไม่จำเป็นต้องรู้ที่ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อยู่ของฝ่ายส่งแต่ฝ่ายส่งต้องรู้ที่อยู่(หรืออาจจะแทนว่าเป็นเบอร์โทรศัพท์) เพื่อจะได้เรียกสายได้ถูกต้อง ส่วน โหมดที่สองนั้นทำงานคล้ายเครื่องอินเทอร์เน็ต คือเชื่อมโยงกันอยู่แล้วหากฝ่ายใดต้องการส่งข้อมูลก็สามารถส่ง ได้เลย ไม่ต้องรอให้มีผู้มารับสาย โดยวิธีนี้ทั้งสองฝ่ายต้องรู้ที่อยู่ซึ่งกันและกันเพื่อจะได้โยงสายถึงกันได้

สมาชิกของ Winsock ก่อนโทรล

คุณสมบัติ (Property) ของ Winsock

- BytesReceived เป็นจำนวนข้อมูลที่รับเข้ามาในแต่ละครั้ง
- LocalHostName แสดงชื่อเครื่องที่ใช้
- LocalIP จะบอกค่า IP address ของเครื่องที่ใช้
- LocalPort จะบ่งถึงช่องทางสำหรับการส่งข้อมูล
- Protocol เป็นตัวเลือกระบบที่จะใช้ในการติดต่อ (TCP หรือ UDP)
- RemoteHost แสดงถึง IP Address ของเครื่องที่เราจะติดต่อกับ
- RemoteHostIP แสดงถึง IP Address ของเครื่องที่ติดต่อเข้ามา
- RemotePort ไว้ใช้เลือกช่องทางในการติดต่อกับเครื่องอื่นๆ
- State สถานะการติดต่อเช่นกำลังติดต่อยู่, เลิกติดต่อไปแล้ว, ยังไม่ได้ทำการติดต่อใดๆ, รอรับฟังการติดต่อ จากอีกด้าน

กระบวนการ (Method) ที่ใช้ Winsock

- Accept แสดงว่ายอมรับการติดต่อจากอีกเครื่อง
- Bind บ่ง LocalPort และ LocalIP ที่จะใช้สำหรับการติดต่อแบบ TCP
- Close ปิดการติดต่อ หรือการรับฟัง
- GetData ไว้รับข้อมูลที่ส่งมาจากอีกเครื่องจากบล็อคของข้อมูลมาใส่ไว้ในตัวแปร
- Listen รอรับฟังการติดต่อจากเครื่องอื่น
- PeekData ไว้รับรายละเอียดของข้อมูลที่ได้รับ
- SendData ไว้ส่งบล็อคข้อมูลไปยังอีกเครื่อง

เหตุการณ์ (Event) ที่เกิดขึ้นใน Winsock

- Close เกิดขึ้นเมื่อได้รับการตัดการติดต่อจากอีกเครื่องหนึ่ง

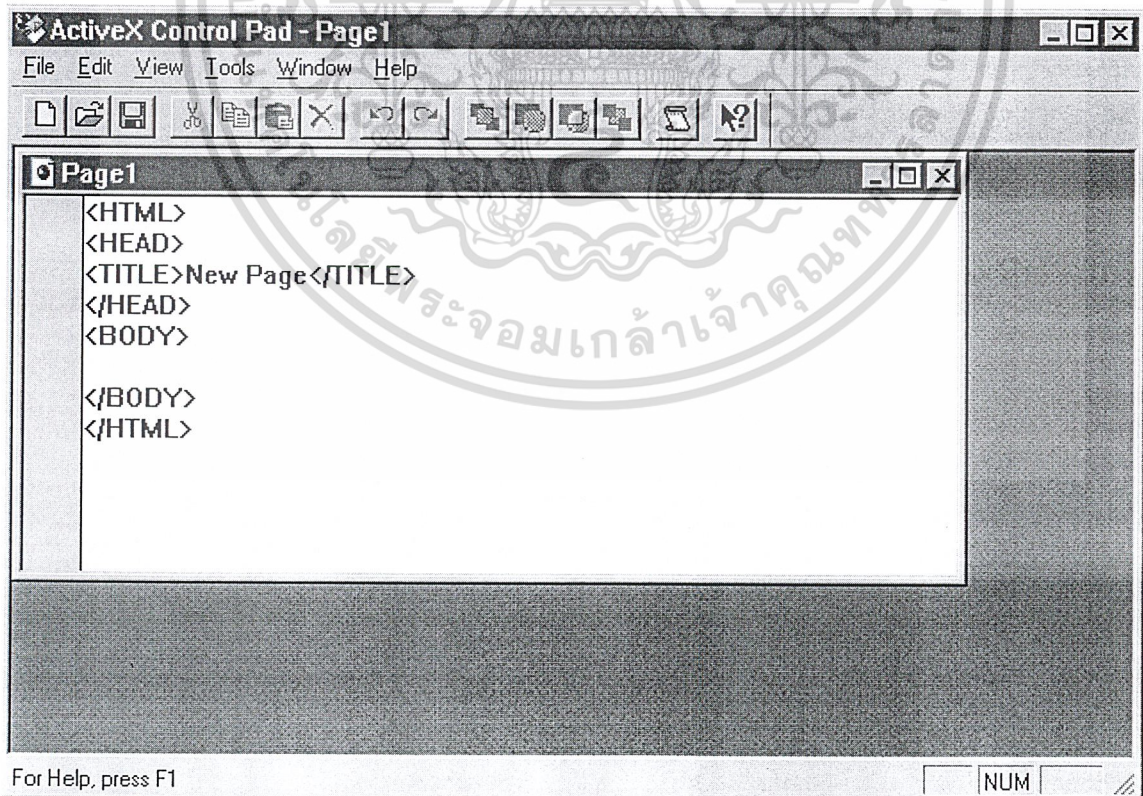
Connect เกิดขึ้นเมื่อได้ทำการติดต่อสำเร็จแล้ว เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ConnectionRequest เกิดขึ้นเมื่อมีการได้รับสัญญาณขอติดต่อกับอีกเครื่อง
- DataArrival เกิดขึ้นเมื่อได้รับสัญญาณการส่งข้อมูลมาจากอีกเครื่อง
- Error เกิดขึ้นเมื่อมีข้อผิดพลาดต่าง ๆ เกิดขึ้นมาระหว่างการทำงาน
- SendComplete เกิดเมื่อข้อมูลได้ทำการส่งไปเรียบร้อยแล้ว
- SendProgress เกิดในขณะที่มีการส่งข้อมูลอยู่

2.13 การActiveX Control Pad เพื่อช่วยในการทำเว็บเพจจาก ActiveX Control

มันเป็นงานที่ง่ายกว่าถ้าเราใช้ ActiveX Control Pad เพื่อช่วยในการเขียนเพจด้วย ActiveX Control และ ActiveX Script ด้วยการช่วยเหลือของ ActiveX Control Pad เราไม่ต้องจำรูปแบบของการเรียกใช้ ActiveX Control ในเพจนั้นเป็นเช่นไร ไม่ต้องจำเบอร์คลาสของคอนโทรลที่จะใช้ และไม่ต้องจำว่าคอนโทรลนั้นมีคุณสมบัติอย่างไรที่ต้องตั้งค่า มีเหตุการณ์อะไรที่เกิดขึ้นได้ ซึ่งสิ่งเหล่านี้จะช่วยในการเขียนสคริปต์ (โดยปกติแล้วสคริปต์ที่ได้นั้นเป็น Vbscript) ซึ่งช่วยให้การเขียนเพจจาก ActiveX Control และ ActiveX Script เป็นไปโดยง่าย

การใช้ ActiveX Control Pad

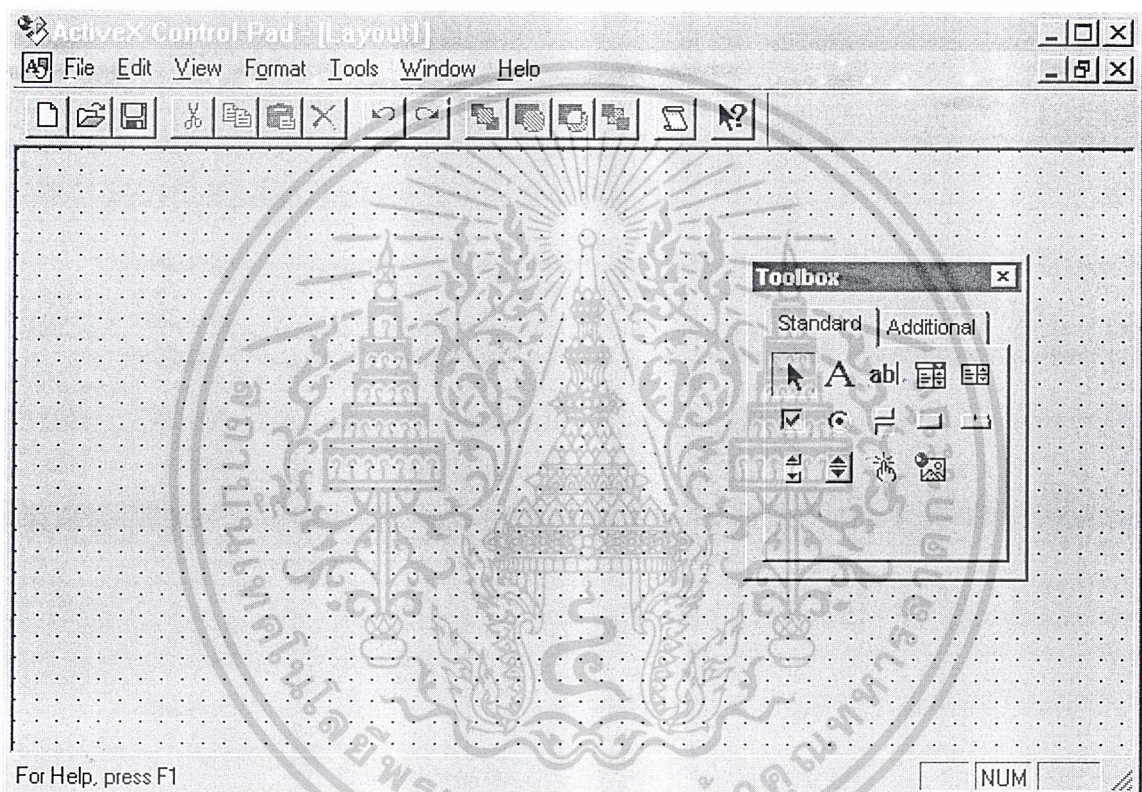


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 20
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังที่เราติดตั้ง ActiveX Control Pad เมื่อเรียกใช้โปรแกรมเราจะได้น้ำจืดรูป 20

การสร้างเพจ

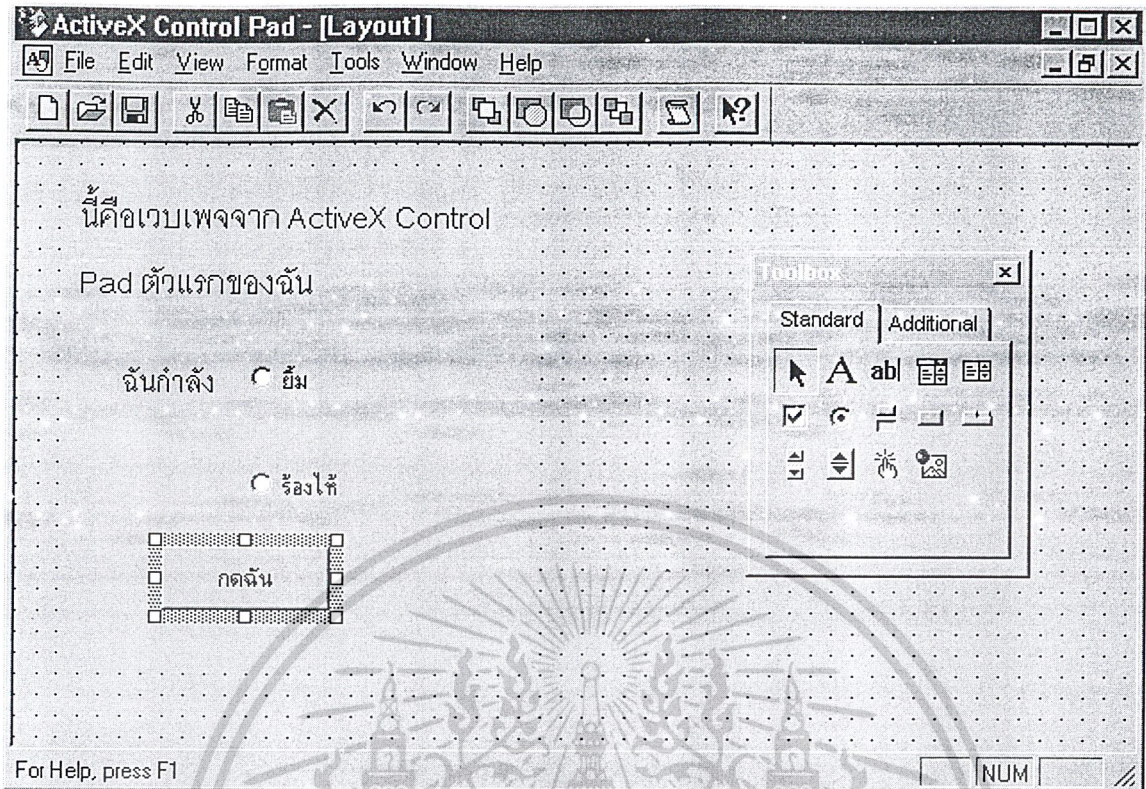
หลังจากที่เราได้เรียกใช้ ActiveX Control Pad แล้ว เราอาจจะสร้างเพจ HTML ธรรมดาคล้ายการใช้ Text Editor ตัวหนึ่ง แต่ ActiveX Control Pad สามารถทำอะไรได้มากกว่านั้น เปิดเมนู File แล้วเลือก New HTML Layout เราจะได้น้ำจืดรูปที่ 21



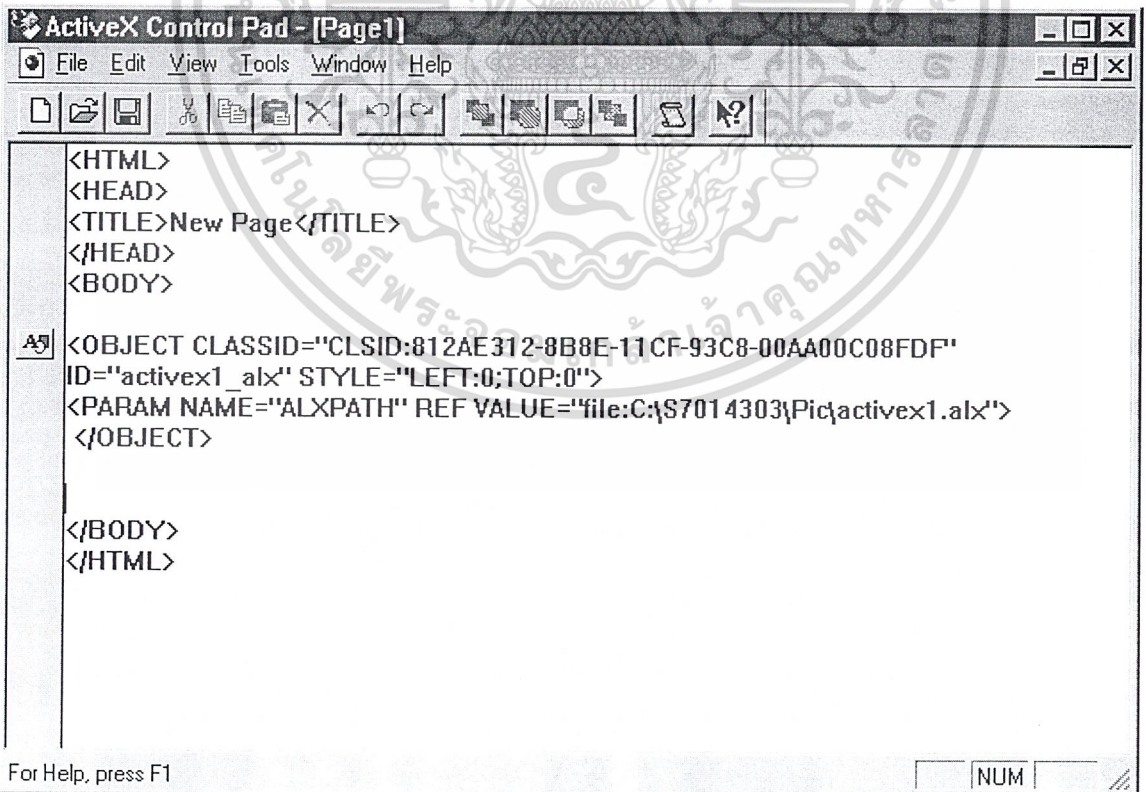
รูปที่ 21

สำหรับบางคนที่เคยใช้ Visual Basic หรือ Delphi ก็คงรู้สึกคุ้นเคยหน้าจอนี้ สิ่งหนึ่งที่สำคัญมากที่ ActiveX Control Pad นำไปสู่ความสามารถในการออกแบบเว็บเพจก็คือการที่สิ่งที่คุณเห็นคือสิ่งที่คุณได้ (What-You-See-Is-What-You-Get: WYSIWYG) มันเป็นการยากที่จะเขียนบรรยายด้วยคำให้เพจออกมาอย่างที่ เราต้องการ ActiveX Control Pad สามารถสร้างฟอร์มโดยการเลือกสิ่งต่าง ๆ มาแล้วก็วางลงในตำแหน่งที่ต้องการ ลองเริ่มสร้างโดยการวาง Label, Radio Button และ Button ลงเราจะได้เพจหน้าตาจืดรูปที่ 22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



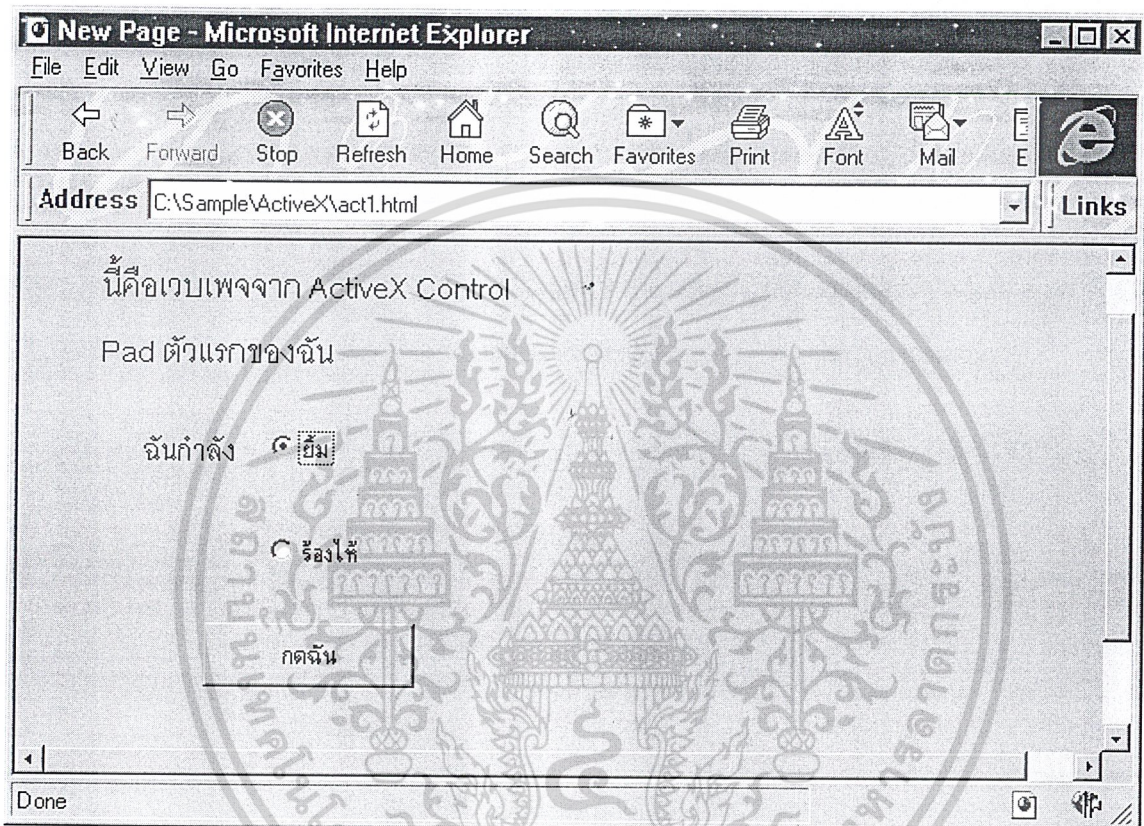
รูปที่ 22



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานที่รูปที่ 23 ึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่นี้ก็ลองเซฟไฟล์ตั้งชื่อเช่น ActiveX1.alx หลังจากนั้นก็ปิดหน้าจอนี้และกลับไปหน้าจอแรก (รูปที่ 16) เลื่อนเคอร์เซอร์ไประหว่างแท็กซ์ <BODY> เรียกเมนู Edit แล้วเลือก Insert HTML Layout เลือกไฟล์ ActiveX1.alx คุณก็จะได้หน้าจอดังรูปที่ 23

ที่นี้ก็เซฟไฟล์ตั้งชื่อไฟล์เช่น Act1.html ที่นี้เราก็พร้อมแล้วที่จะตรวจสอบมันโหลดไฟล์เข้าบราวส์เซอร์จะได้เพจ ดังรูปที่ 24



รูปที่ 24

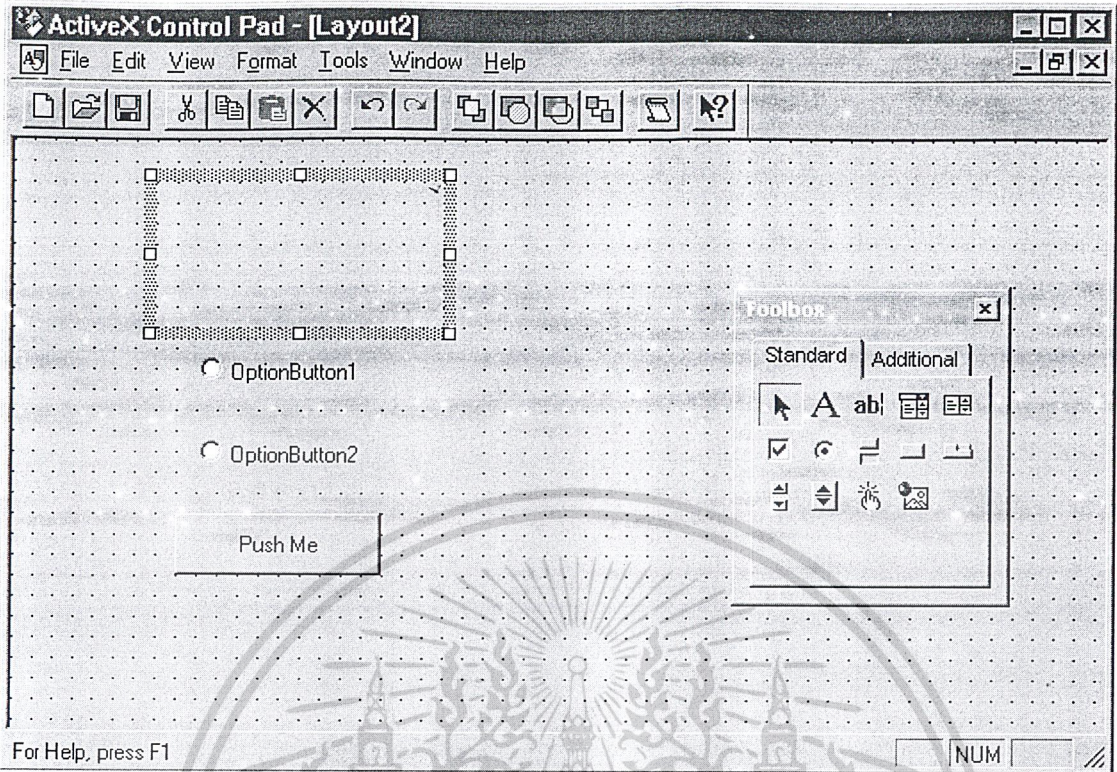
หลังจากนั้นเราก็ได้เพจจากการเขียนด้วย ActiveX โดยใช้ ActiveX Control Pad ที่นี้เราจะมาลองตรวจสอบอะไรบางอย่าง ขึ้นแรกเรากลับไป ActiveX Control Pad ลองใช้ text browser เปิดไฟล์ ActiveX1.alx

ไม่มีอะไรที่เป็นความลึกลับในที่นี้ ที่นี้เราลองมามองกันที่โปรแกรม Act1.html ที่รูปที่ 23 ที่นี้ก็จะเห็นว่า ActiveX1.alx ไม่ได้ถูกส่งไปยังหน้าจอทันทีแต่มันจะถูกเก็บให้อยู่ในรูปของ HTML Layout Control ซึ่งหน้าจอที่วางไว้ใน Layout จะถูกนำมาแสดงตามที่วางไว้

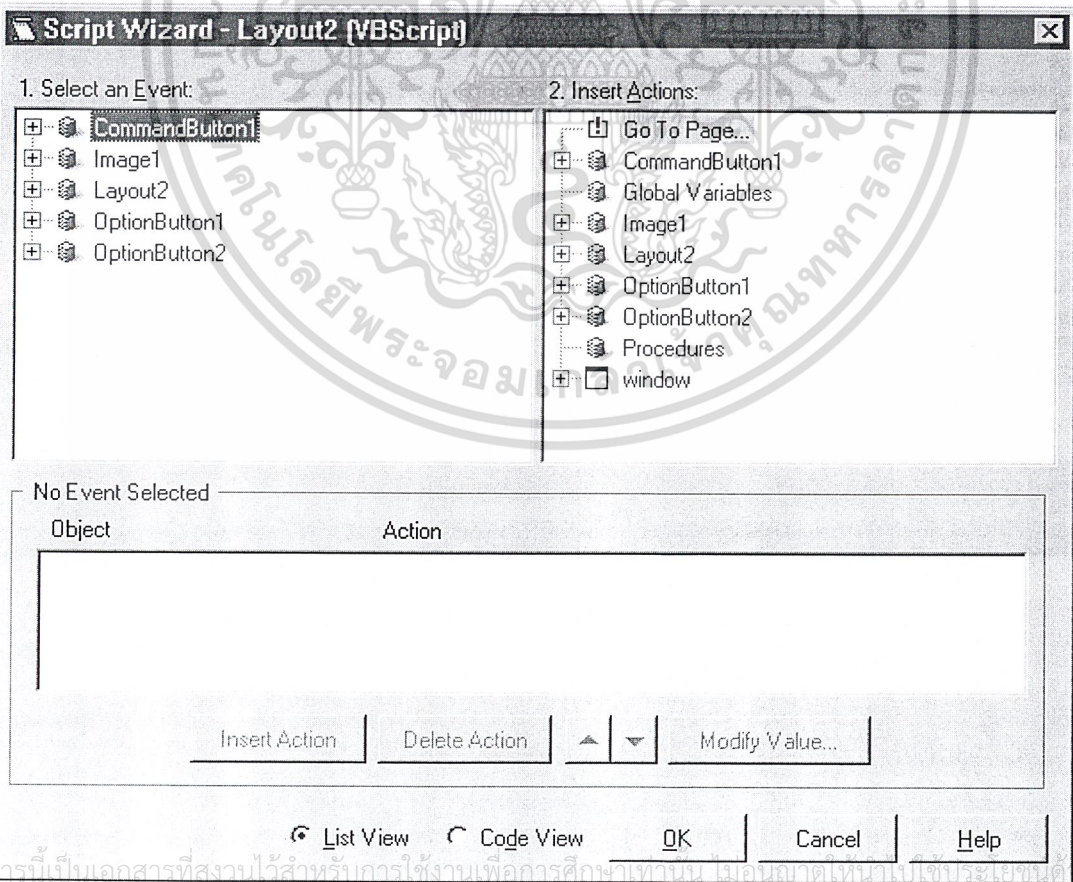
การเขียนสคริปต์ด้วย ActiveX Control Pad

จากวิธีการ WYSIWYG สำหรับการออกแบบเว็บเพจ ActiveX Control Pad สนับสนุนทั้ง JavaScript และ VbScript ออกแบบเลย์เอาท์เรียกมันว่า ActiveX2.alx ใต้ radio button, Image, และ button บนนั้นจะได้ อะไรดังรูปที่ 25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 25



รูปที่ 26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

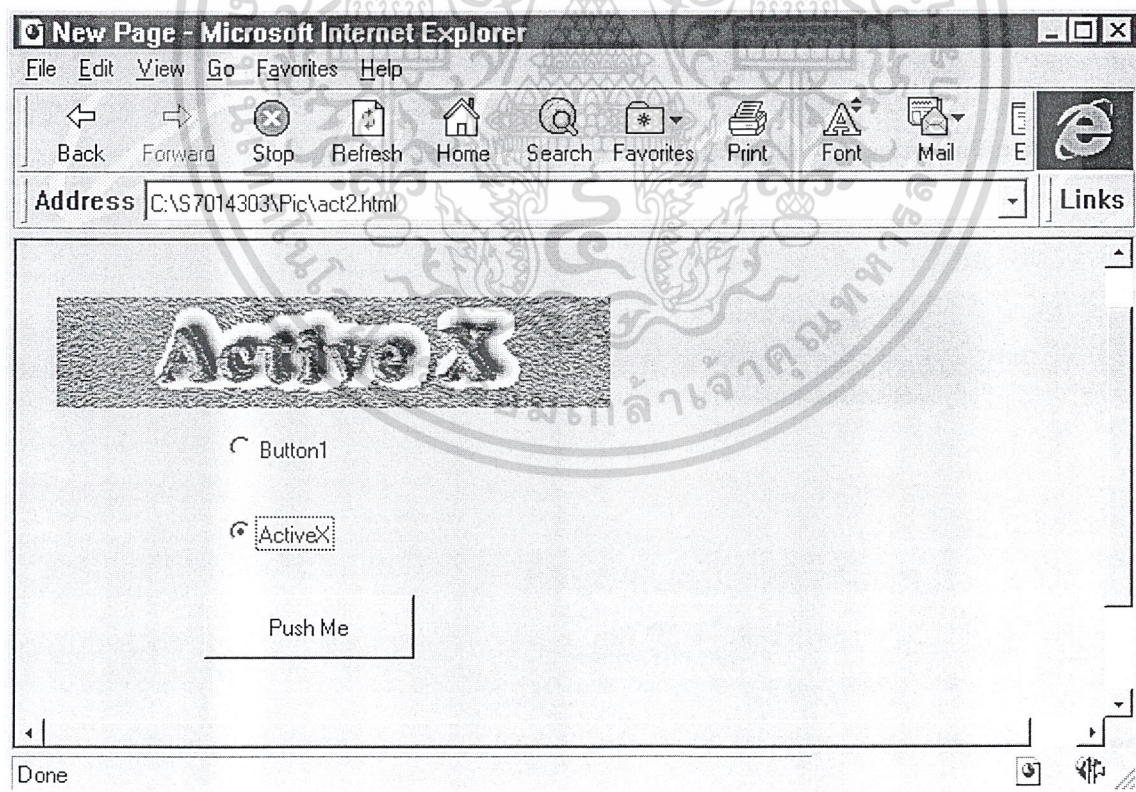
เราต้องการให้โปรแกรมนี้สามารถเปลี่ยนรูปได้ตามการเลือกของ radio button โดยจะทำงานเมื่อกดปุ่ม Push Me เมื่อได้ฟอร์มดังรูปที่ 25 เลือก the Script Wizard จากเมนู Tools แล้วจะเห็นว่า Script Wizard จะรู้ว่าไม่มีสิ่งใดบ้างที่อยู่บนฟอร์ม และมีเหตุการณ์อะไรบ้างที่เกิดขึ้นได้กับแต่ละ Object ทั้งหมดนี้อยู่บน Script Wizard ดังรูปที่ 26

เพิ่มโคดเหล่านี้ลงเมื่อเลือก event ของ commandbutton1

```
if (OptionButton1.Value)
{ image1.picturepath = "cash.gif"
}
else
{ image1.picturepath = "pow.gif"
}
```

หลังจากเปลี่ยน GroupName บน radio button ทั้งสองปุ่มเป็น Choice จะเป็นการเชื่อมปุ่มทั้งสองโดยเมื่อทำเช่นนี้แล้วปุ่มแรกของ radio button ก็จะถูกเลือก

เซฟไฟล์นี้ลงบน ActiveX2.alx แล้วใส่ลงในเพจอันใหม่เซฟเพจนี้ว่า Act2.html หลังจากนั้นก็โหลดเพจนี้มาดู เมื่อดูครั้งแรกจะไม่มีรูปปรากฏแต่เมื่อกดปุ่มก็จะได้รูปตามการเลือกปุ่มดังรูปที่ 27



รูปที่ 27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.14 ฐานข้อมูลเบื้องต้น

นิยามศัพท์เกี่ยวกับฐานข้อมูล

- Table (ตาราง) คือส่วนที่เก็บความสัมพันธ์ของข้อมูล
- Record (เรคคอร์ด) คือส่วนที่เก็บข้อมูลในแต่ละชิ้นข้อมูล อาจจะเรียกว่า Row หรือ Tuple
- Field (ฟิลด์) คือส่วนที่เก็บข้อมูลในรายละเอียดของแต่ละเรคคอร์ด อาจจะเรียกว่า Column หรือ Attribute
- Primary Key คือฟิลด์หนึ่ง ๆ หรือหลายฟิลด์รวมกัน โดยที่ Primary Key มีคุณสมบัติว่าควรมีค่ากระทัดรัด, ไม่เปลี่ยนแปลงโดยง่าย, ไม่มีค่าซ้ำกันในเรคคอร์ดที่ต่างกัน
- Compound Primary Key คือ Primary Key ที่เป็นหลายฟิลด์รวมกัน
- Non Primary Key คือฟิลด์ที่ไม่ใช่ Primary Key
- Foreign Key คือฟิลด์ที่เป็น Primary Key จากตารางหนึ่ง ไปเป็นฟิลด์ที่เป็น Non Primary Key ในอีกตารางหนึ่ง
- Candidate Key คือทุกฟิลด์ที่สามารถจะเป็น Primary Key ได้

Maintaining Database Integrity

คือส่วนที่เป็นข้อบังคับเพื่อให้ฐานข้อมูลนั้น เป็นฐานข้อมูลใช้งานได้ง่ายไม่สับสน แบ่งเป็นข้อบังคับต่าง ๆ ดังนี้

- Entity Integrity
ข้อบังคับของ entity integrity ก็คือค่าของ primary key จะผิดไม่ได้ และทุกเรคคอร์ดต้องมี primary key และ ฟิลด์ที่เป็น primary key จะต้องลงค่าข้อมูลที่สมบูรณ์ รวมทั้งใน compound primary key ด้วย
- Referential Integrity
ข้อมูลจากตารางต่างหากที่มีการอ้างอิงกัน เมื่อมีการเปลี่ยนแปลงในตารางหลักแล้ว ค่าของข้อมูลที่อยู่ในตารางที่อ้างอิงกันนั้น (เป็น foreign key จากตารางแรกที่เปลี่ยนแปลงค่า) จะต้องถูกเปลี่ยนค่าตาม กระบวนการนี้ถูกเรียกว่า cascading update
ถ้าเป็นการลบแล้วจะลบข้อมูลที่อ้างอิงข้อมูลนั้นโดยตารางอื่นก็จะถูกลบไปด้วย โดยจะเรียกกระบวนการนี้ว่า cascading delete

ความสัมพันธ์ระหว่างสองตาราง การที่ตารางมีความสัมพันธ์ต่อกันได้นั้นเกิดจากการที่ Primary Key จากตารางหนึ่ง เป็น Foreign Key ในอีกตารางหนึ่ง โดยจะเรียกว่าตารางแรกนั้นมีความสัมพันธ์แบบ 1:1 ไม่ว่าการใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1 to Many กับอีกตารางหนึ่ง ได้แก่การที่ค่าของฟิลด์ Primary Key ในตารางที่สองที่มีค่าต่างกัน อาจจะมีค่าในฟิลด์ Foreign Key ของตารางที่สองที่เหมือนกันก็ได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การพัฒนาแอปพลิเคชัน

3.1 จุดประสงค์ของแอปพลิเคชัน

สามารถติดต่อกับฐานข้อมูลชนิดต่างๆ แล้วส่งผ่านไปแสดงผลระหว่างเครื่องได้ โดยที่ฐานข้อมูลที่จะติดต่อนั้นไม่จำเป็นต้องรู้ชื่อฟิลด์ จำนวนฟิลด์ จำนวนเรคคอร์ด หรือความสัมพันธ์ระหว่างตารางมาก่อนทำให้ผู้ใช้สามารถเลือกได้ว่าจะดูตารางไหน ดูฟิลด์อะไร และสามารถดูข้อมูลจากตารางที่มีความสัมพันธ์แบบ 1:Many ได้โดยผู้ใช้ไม่จำเป็นต้องรู้มาก่อนว่าต้องรู้มาก่อนว่าตารางทั้งสองนั้นมีความสัมพันธ์กัน

3.2 การพัฒนาแอปพลิเคชัน

สามารถแบ่งการทำงานของแอปพลิเคชันได้สามส่วนใหญ่คือ

1. ส่วนที่ใช้สำหรับการติดต่อกับฐานข้อมูลต่าง ๆ

ขั้นแรกถ้าหากต้องการติดต่อกับฐานข้อมูลต่าง ๆ นั้นแอปพลิเคชันนั้นต้องอ้างอิง Microsoft DAO 3.5 Object Library เพื่อจะได้เรียกใช้ฟังก์ชันการทำงานสำหรับการติดต่อกับฐานข้อมูลได้ สำหรับฐานข้อมูลข้อมูลที่ติดต่อดังนั้นสามารถแบ่งประเภทใหญ่ได้เป็นสองประเภทได้แก่

1.1 ประเภทที่ใครก็สามารถเรียกดูได้

ฐานข้อมูลประเภทนี้ผู้บริการ (ที่เครื่องเซิร์ฟเวอร์) ควรจะเป็นผู้เลือกว่าจะให้ผู้ใช้ (ที่เครื่องไคลเอนต์) สามารถเรียกดูตารางจากไฟล์ใดได้บ้าง สำหรับแอปพลิเคชันนี้เมื่อผู้บริการเริ่มเปิดบริการ (เรียกใช้โปรแกรม) ก็ต้องเลือกไฟล์ฐานข้อมูลต่างๆ โดยฐานข้อมูลชนิดที่เปิดได้มีดังต่อไปนี้

- ฐานข้อมูลชนิดที่เป็น Access (ไฟล์นามสกุล MDB)
- ฐานข้อมูลชนิดที่เป็น Excel (ไฟล์นามสกุล XLS)
- ฐานข้อมูลชนิดที่เป็น Dbase หรือ Foxpro (ไฟล์นามสกุล DBF)
- ฐานข้อมูลชนิดที่เป็น Paradox (ไฟล์นามสกุล DB)

เมื่อผู้บริการ ได้เลือกไฟล์ฐานข้อมูลเหล่านี้โปรแกรมก็จะทำการเปิดฐานข้อมูลเหล่านี้ไว้รอผู้ใช้งานมาเรียกดูเลยโดย Function สำหรับการเปิดฐานข้อมูลก็คือ

Set database = .OpenDatabase (dbname, options, read-only, connect)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น ยกเว้นที่มีเหตุอันสมควร และต้องขออนุญาตจากเจ้าของเอกสารทุกประการ

database เป็นตัวแปรชนิดฐานข้อมูลจะใช้เวลาที่อ้างอิงฐานข้อมูลตัวที่เป็นี้ (Database type)

dbname คือชื่อของฐานข้อมูลโดยถ้าเป็นฐานข้อมูลชนิดที่เป็น Access หรือ Excel ก็คือชื่อและ Path ของไฟล์ ถ้าเป็นฐานข้อมูลชนิดที่เป็น Dbase, Foxpro หรือ Paradox ก็จะเป็นชื่อ Path ที่เก็บไฟล์นั้นเท่านั้น (เมื่อเปิดฐานข้อมูลมันจะเปิดทุกไฟล์ใน Path นั้นเก็บเป็นฐานข้อมูลเดียวกัน โดยที่ไฟล์หนึ่งก็เก็บหนึ่งตาราง)

option จะมีหรือไม่มีก็ได้เป็นการตั้ง option พิเศษสำหรับฐานข้อมูลนั้น

read-only ตั้งว่าจะใช้ดูอย่างเดียวหรือแก้ไข,เพิ่ม,ลบได้ ในที่นี้จะตั้งให้เป็น true คือดูอย่างเดียวเท่านั้น

connect บอกว่าฐานข้อมูลที่จะเปิดนั้นเป็นฐานข้อมูลชนิดไหน

เมื่อเปิดฐานข้อมูลเสร็จแล้วก็พร้อมให้ผู้ใช้มาเรียกใช้งาน

1.2 ประเภทหลายผู้ใช้ต้องใส่ชื่อและรหัสผ่าน

ฐานข้อมูลประเภทนี้คือประเภทที่ต้องใส่รหัสผ่านก่อนจึงจะสามารถเรียกใช้บริการได้ สำหรับแอปพลิเคชันนี้ฐานข้อมูลชนิดนี้ที่สามารถเปิดดูได้คือฐานข้อมูลชนิดออราเคิลโดยเมื่อผู้ใช้เรียกใช้งานแล้วผู้ใช้จะถูกถามรหัสผ่าน สำหรับผู้ใช้ที่ต้องการดูเฉพาะฐานข้อมูลแบบที่หนึ่งก็ให้ใส่ชื่อว่า *guest* ก็จะสามารถผ่านเข้าไปดูได้เลย (การกำหนดเช่นนี้มีข้อเสียคือไม่สามารถเปลี่ยนแปลงรหัสผ่านได้ถ้าใครรู้ก็สามารถผ่านเข้าไปได้ตลอดเวลา การเปลี่ยนรหัสผ่านนั้นต้องใช้วิธีแก้ไขโปรแกรมอย่างเดียวซึ่งในส่วนนี้ต้องการการแก้ไขต่อไป) สำหรับผู้ใช้ที่ต้องการเข้าดูฐานข้อมูลชนิดออราเคิลด้วยก็ต้องใส่ชื่อและรหัสผ่านให้ตรงกับที่เวลาเข้าดูข้อมูลจากออราเคิลจากเครื่องเซิร์ฟเวอร์โดยตรง และยังคงใส่ *Data Source Name* ให้ตรงกับที่เครื่องที่บริการเซิร์ฟเวอร์ตั้งไว้ (การทำเช่นนี้สามารถเลือกติดต่อกับฐานข้อมูลชนิดออราเคิลได้จากหลายแหล่ง โดยการตั้ง *Data Source Name* ตามหัวข้อเรื่องการเชื่อมฐานข้อมูลแบบออราเคิลเข้ากับโปรแกรม) หากรหัสผ่านไม่ตรงทั้งสองอย่างก็ไม่สามารถเข้าดูข้อมูลได้ เมื่อเข้าดูข้อมูลได้ตอนนี้ก็พร้อมที่จะเรียกดูตารางต่าง ๆ ได้ตามต้องการ

สำหรับการติดต่อในการเปิดดูตารางต่าง ๆ ฐานข้อมูลที่ได้เปิดไว้แล้วนั้นใช้ฟังก์ชันนี้

```
Set recordset = object.OpenRecordset (source, type, options, lockedits)
```

```
recordset movelast
```

```
recordset movefirst
```

recordset เป็นตัวแปรชนิด *recordset* จะใช้เวลาที่อ้างถึงฐานข้อมูลตัวที่เปิดนี้

object คือตัวแปรชนิดฐานข้อมูลที่เก็บตารางนั้นไว้

source คือ SQL สตริงที่ใช้เรียกตารางนั้น (*Selec * from TableName*)

type มีหรือไม่มีก็ได้เป็นชนิดของการเรียกใช้ข้อมูล ในที่นี้จะตั้งเป็น *dbOpenSnapshot* เพราะว่า

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ทางการค้า
 ไม่ควรนำเอกสารนี้ไปเผยแพร่ในที่สาธารณะโดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

options มีหรือไม่มีก็ได้เป็นการระบุการส่งผ่านข้อมูลระหว่างฐานข้อมูลกับแอปพลิเคชัน ในที่นี้จะตั้งเป็น *dbSQLPassThrough* โดยจะส่งความต้องการเป็น SQL เพื่อเรียกดูข้อมูล

lockedits มีหรือไม่มีก็ได้ เป็นตัวเลือกการล็อกสำหรับเรคคอร์ดเซต ในที่นี้จะป็น *dbReadOnly* เนื่องจากแอปพลิเคชันนี้จะเรียกดูข้อมูลอย่างเดียว ไม่มีการแก้ไขต่อเติม

ในที่จะเห็นว่าการเรียกดูตารางนั้นจะเปิดฐานข้อมูลให้อยู่ในรูปของเรคคอร์ดเซต ไม่ได้เปิดเป็นตารางโดยตรง สำหรับคำสั่งนี้เมื่อเรียกใช้แล้วควรเรียกคำสั่ง *recordset movelast* และ *recordset movefirst* ต่อทันทีเพื่อให้ข้อมูลต่าง ๆ มารอพร้อม (ถ้าไม่ใช่คำสั่งเหล่านี้แล้วจะเรียกข้อมูลมาดูไม่ได้)

สำหรับการเช็คความสัมพันธ์ระหว่างตารางนั้น จะทำโดยการเช็คจากฟิลด์ที่ชื่อเหมือนกัน โดยจะเช็คจากทุกตารางที่เรียกดูได้ จะทำการเช็คเมื่อรู้ว่าตารางทั้งหมดมีอะไรบ้าง ซึ่งในการเช็คฟิลด์โดยใช้รูปเช็คเช่นนี้จะทำให้โปรแกรมทำงานช้า และอาจจะทำให้ตารางที่ไม่ได้สัมพันธ์กันแต่มีชื่อฟิลด์เหมือนกันมาสัมพันธ์กันได้ ซึ่งถ้าสามารถเรียกดูความสัมพันธ์ที่มีระบุอยู่แล้วในฐานข้อมูลแต่ละชนิด จะทำงานได้เร็วและถูกต้องมากขึ้น แต่มันจะไม่สามารถเช็คความสัมพันธ์ระหว่างตารางคนละชนิดได้

2. ส่วนที่ใช้สำหรับการสื่อสารระหว่างเครื่อง

ในการติดต่อสื่อสารระหว่างเครื่องของแอปพลิเคชันนั้นใช้วิธีสื่อสารกันผ่าน *winsock control* สำหรับรายละเอียดในการใช้งานของ *winsock Control* นั้นอ่านได้ที่หัวข้อการนำ *Winsock Control* ไปใช้เพื่อการติดต่อระหว่างไคลเอนต์/เซิร์ฟเวอร์ในบทที่ 2

โดยในการพัฒนาแอปพลิเคชันนี้ สำหรับโปรแกรมที่ใช้ ณ เครื่องเซิร์ฟเวอร์ต้องใช้ *winsock control* สองตัว โดยตัวแรกนั้นจะทำหน้าที่คอยรับฟังการติดต่อเข้ามาจากเครื่องไคลเอนต์ *winsock control* ตัวที่สองใช้สำหรับการสื่อสารระหว่างตัวเซิร์ฟเวอร์กับไคลเอนต์

เมื่อเริ่มโปรแกรม จะต้องเช็คค่าคุณสมบัติ (*property*) ของ *winsock control* ตัวแรก โดยจะเช็คค่าของ *LocalPort* ควรจะเป็นค่าที่มีค่าเกินหนึ่งพัน ไปนิดหน่อย (ควรเช็คว่าไม่ใช่ *port* ซ้ำกับอันอื่น) แล้วก็เรียกฟังก์ชัน *Listen* เพื่อเช็คให้ *winsock control* ตัวนี้คอยรับฟังการติดต่อจากเครื่องอื่น ๆ คราวนี้เครื่องก็พร้อมสำหรับการติดต่อจากที่ใด ๆ แล้ว

สำหรับโปรแกรมที่รันที่เครื่องเซิร์ฟเวอร์นั้น ก็จะต้องมี *winsock control* อีกตัวหนึ่งเพื่อติดต่อกับ *winsock control* ที่เซิร์ฟเวอร์ วิธีการที่ใช้ในการติดต่อขั้นแรกต้องเช็คค่าของ *RemortPort* ให้ตรงกับที่เช็คค่า *LocalPort* ของ *winsock control* ที่เซิร์ฟเวอร์ ขั้นตอนมาก็ตั้งค่า *RemoteHost* ให้ตรงกับ IP ของเซิร์ฟเวอร์ (เนื่องจากค่า IP ของเซิร์ฟเวอร์นั้นอาจจะเปลี่ยนไปตามเครื่องต่าง ๆ เพราะฉะนั้นในการเขียนแอปพลิเคชันนี้ควรจะสร้างให้ค่าของ *RemortHost* นั้นเปลี่ยนได้ตามการเลือกของผู้ใช้ โดยการเพิ่มค่าคุณสมบัติให้กับคอนโทรลทำให้เมื่อมีผู้มาใช้คอนโทรลก็สามารถตั้งค่าของ *RemortHost* ได้เอง) จากนั้นก็สั่งให้ *winsock control* ตัวนั้นเริ่มทำการติดต่อ โดยเรียกใช้ฟังก์ชันการทำงาน *connect* แล้วรอผลการตอบรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าจากเซิร์ฟเวอร์

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นที่เซิร์ฟเวอร์เมื่อได้รับ request (เกิดเหตุการณ์ ConnectionRequest ที่ winsock control ตัวแรก) ก็ต้องจัดการการติดต่อให้โดยใช้ฟังก์ชัน Accept โดยเรียกใช้ดังนี้

```
Winsock.Accept requestID
```

โดยค่า RequestID คือค่าที่ส่งมาเมื่อเกิดเหตุการณ์ ConnectionRequest จะได้ดังต่อไปนี้

```
Private Sub Winsock1_ConnectionRequest(ByVal requestID As Long)
```

```
Winsock2.Accept requestID
```

```
End Sub
```

หลังจากนั้นทั้งสองเครื่องก็จะสามารถติดต่อกันได้ โดยเมื่อต้องการจะติดต่อก็เรียกใช้ฟังก์ชัน SendData โดยเรียกใช้แบบนี้

```
Winsock2.SendData Str
```

โดย Str แทนข้อความที่ใช้ส่ง โดยเมื่อเครื่องหนึ่งส่งอีกเครื่องก็จะได้รับ โดยจะเกิดเหตุการณ์ DataArrival ก็จะรับข้อความโดยใช้วิธีนี้

```
Private Sub Winsock2_DataArrival(ByVal bytesTotal As Long)
```

```
Dim strData As String
```

```
Winsock2.GetData strData, vbString
```

```
End Sub
```

โดยที่ค่าที่ได้จะอยู่ในตัวแปร strData เท่านั้นก็สามารถส่งข้อความถึงกันได้แล้ว

3. ส่วนที่ใช้สำหรับการแสดงผลให้ผู้ใช้งาน

ส่วนนี้จะเป็นส่วนที่อยู่บนโปรแกรมที่ทำงานบนเครื่องไคลเอนต์ สามารถทำได้โดยการนำ User Interface Control ต่าง ๆ ที่ Visual Basic มาใช้ โดยที่ใช้หลัก ๆ ก็ได้แก่

- *CommandButton* คือปุ่มที่ใช้เหตุการณ์ Click แล้วให้ทำงานต่าง ๆ ตามต้องการเมื่อกดปุ่ม
- *Label* ใช้แสดงผลที่อยู่ในรูปของอักษร ค่าคุณสมบัติที่มีก็จะใช้ได้แก่ *Caption* ซึ่งเอาไว้เก็บ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

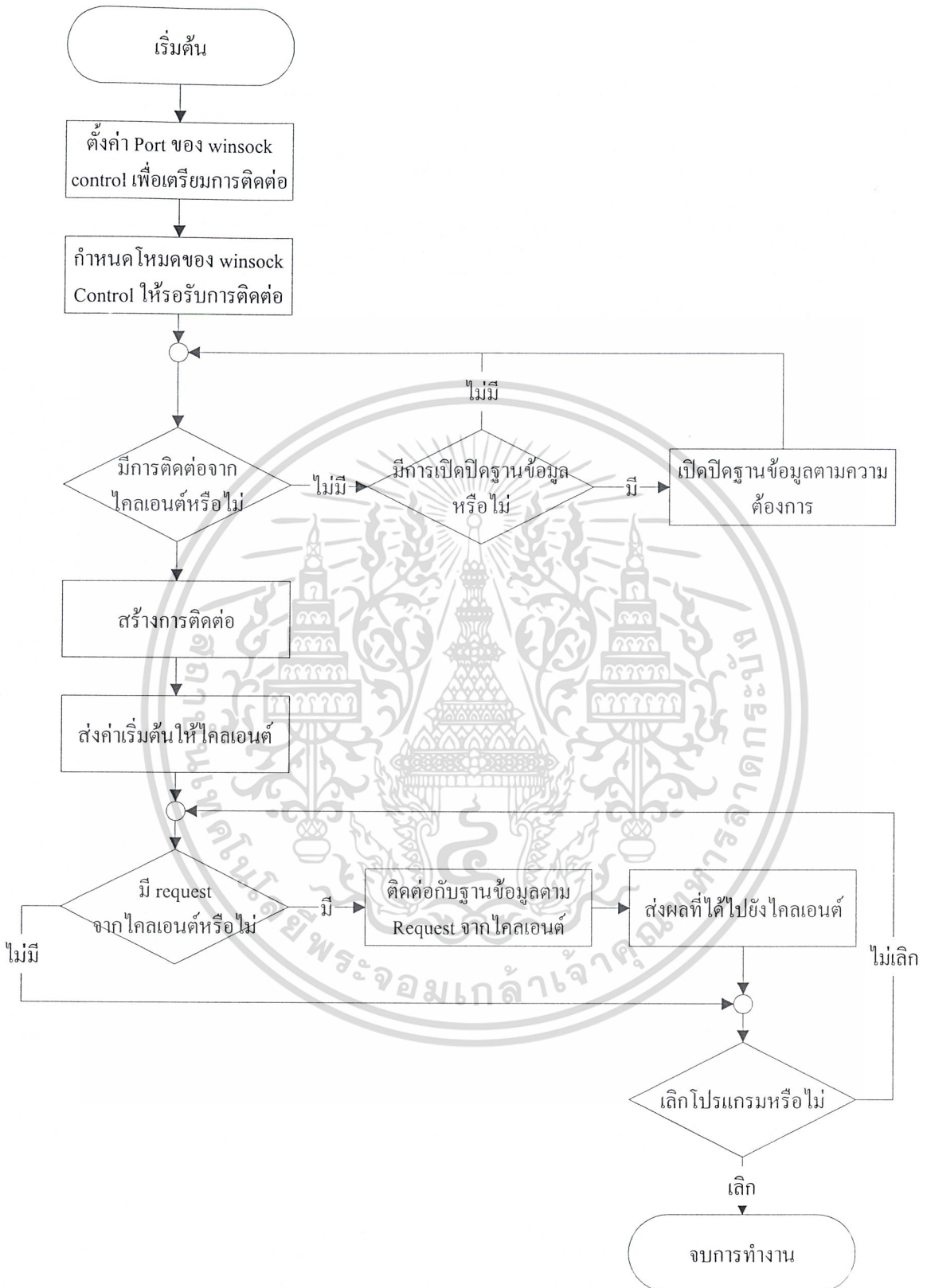
- *ListBox* ใช้เก็บรายชื่อต่าง ๆ เช่นรายชื่อตารางที่เรียกดูได้หรือรายชื่อของฟิลด์ในตาราง โดยเมื่อต้องการเลือกก็ให้กดปุ่มที่รายชื่อที่ต้องเพื่อให้เกิดเหตุการณ์ Click หรือ DbClick เพื่อให้ทำงานต่าง ๆ ตามต้องการได้ และมักจะใช้ฟังก์ชัน AddItem เพื่อเพิ่มสตริงลงในรายการ ใช้ฟังก์ชัน Clear เพื่อลบทุกค่าที่อยู่ในรายการ ใช้คุณสมบัติ ListIndex เพื่อดูว่ารายการนั้นถูกเลือกอยู่ที่ข้อมูลอันไหน ใช้คุณสมบัติ Text เพื่อดูข้อมูลอันที่ถูกเลือกว่าเป็นอะไร
- *PictureBox* ใช้เพื่อให้ง่ายต่อการตั้งค่า เนื่องจากคอนโทรลที่ทำนี้มีหลายหน้าจออยู่บนคอนโทรลเดียว เพื่อให้ง่ายต่อการเลือกใช้หน้าจอไหนก็ได้โดยการจัดให้เป็นหน้าจอหนึ่งต่อหนึ่ง Picture Box โดยวาง object ของหน้านั้นลงบนหนึ่ง Picture Box เมื่อต้องการเรียกใช้หน้าจอไหนก็ให้ Picture Box ที่เก็บหน้าจออันนั้นมองเห็น (เซ็ตที่ค่าคุณสมบัติ visible ถ้าเป็น true ก็มองเห็น ถ้าเป็น false ก็มองไม่เห็น) แล้วทำให้ Picture Box ที่เก็บหน้าจออื่นมองไม่เห็น อาจจะใช้วิธีเดียวกันกับการทำให้หน้าจออื่นเรียกใช้งานได้หรือไม่ (เซ็ตที่คุณสมบัติ enable ถ้าเป็น true ก็เรียกใช้ได้ ถ้าเป็น false ก็เรียกใช้ไม่ได้)

สำหรับการสร้างส่วนแสดงผลนั้นอาจจะใช้ User Interface Control อื่น ๆ อีกเพื่อทำหน้าที่อื่น ๆ ที่นอกเหนือจากนี้ซึ่งเป็นสิ่งที่จะต้องศึกษาเพิ่มเติมต่างหาก

3.3 การทำงานของโปรแกรม Database on Internet ส่วนเซิร์ฟเวอร์

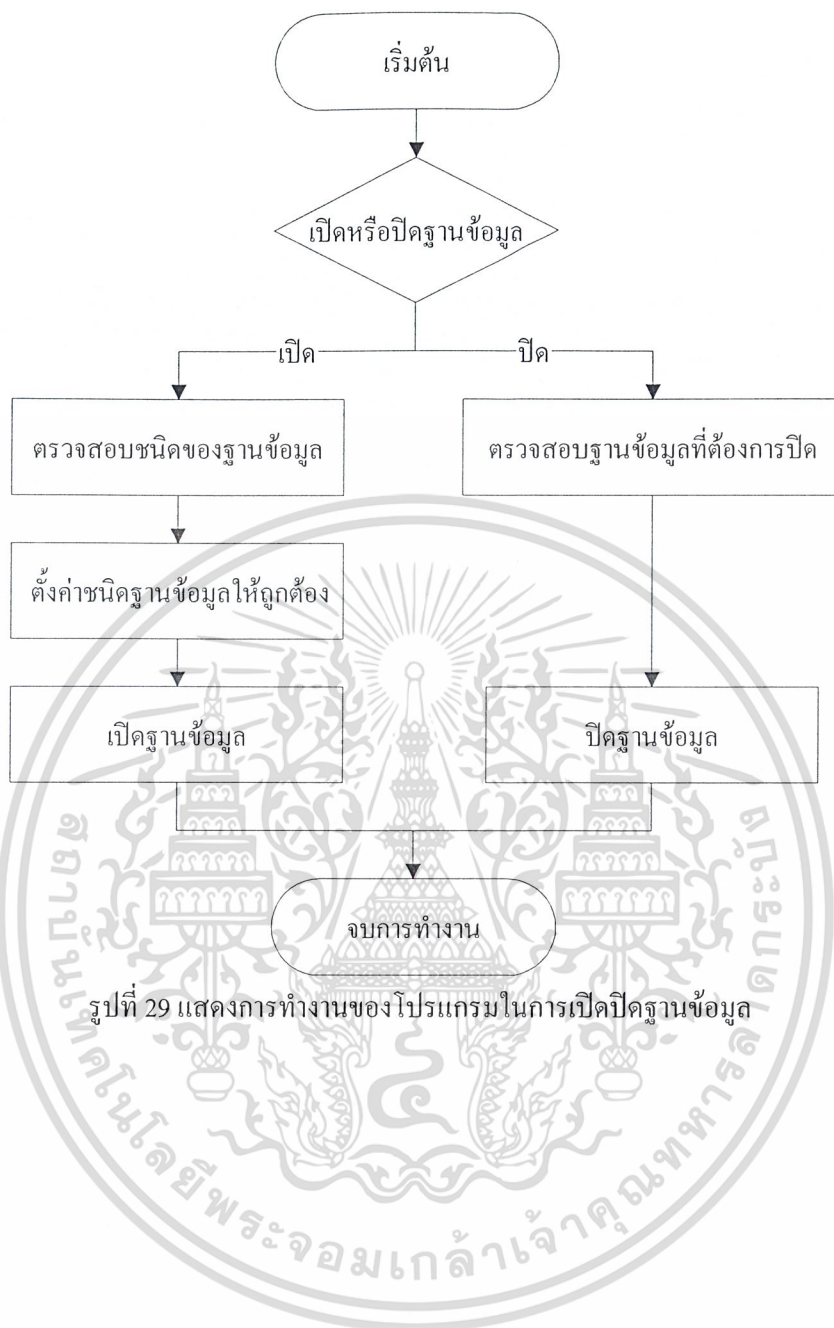
การทำงานของโปรแกรมในส่วนเซิร์ฟเวอร์นั้นจะทำงานตามรูปที่ 28 โดยในการเปิดเปิดไฟล์จะทำตามรูปที่ 29 และติดต่อฐานข้อมูลตาม Request ตามรูปที่ 30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

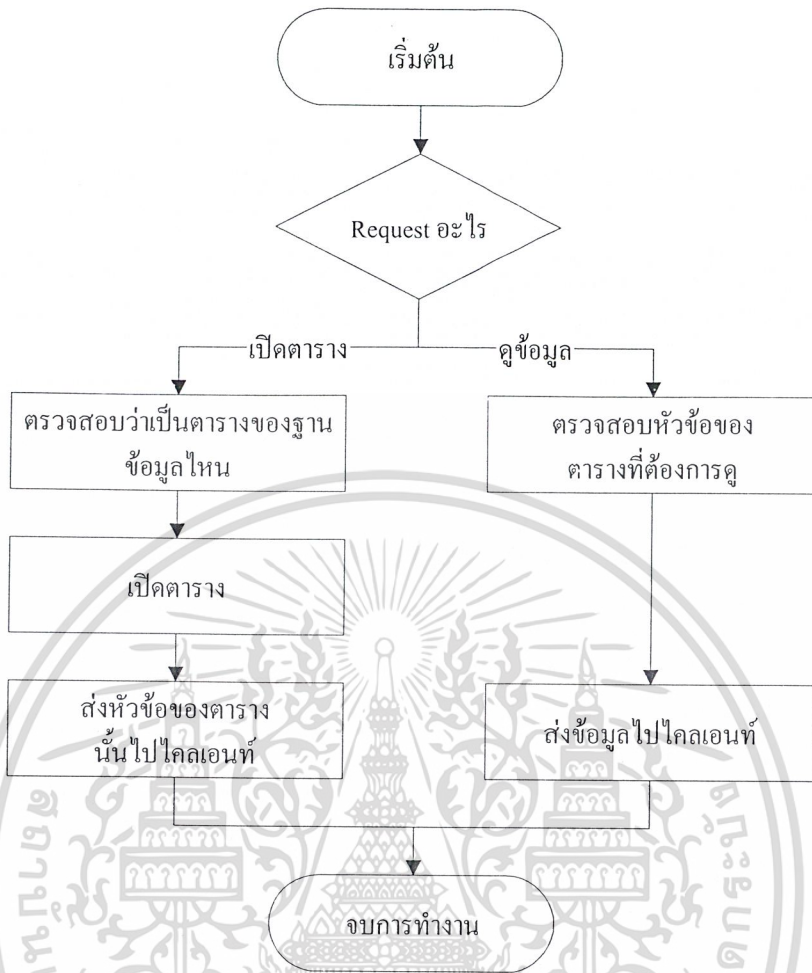


รูปที่ 28 แสดงการทำงานของโปรแกรมในส่วนเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

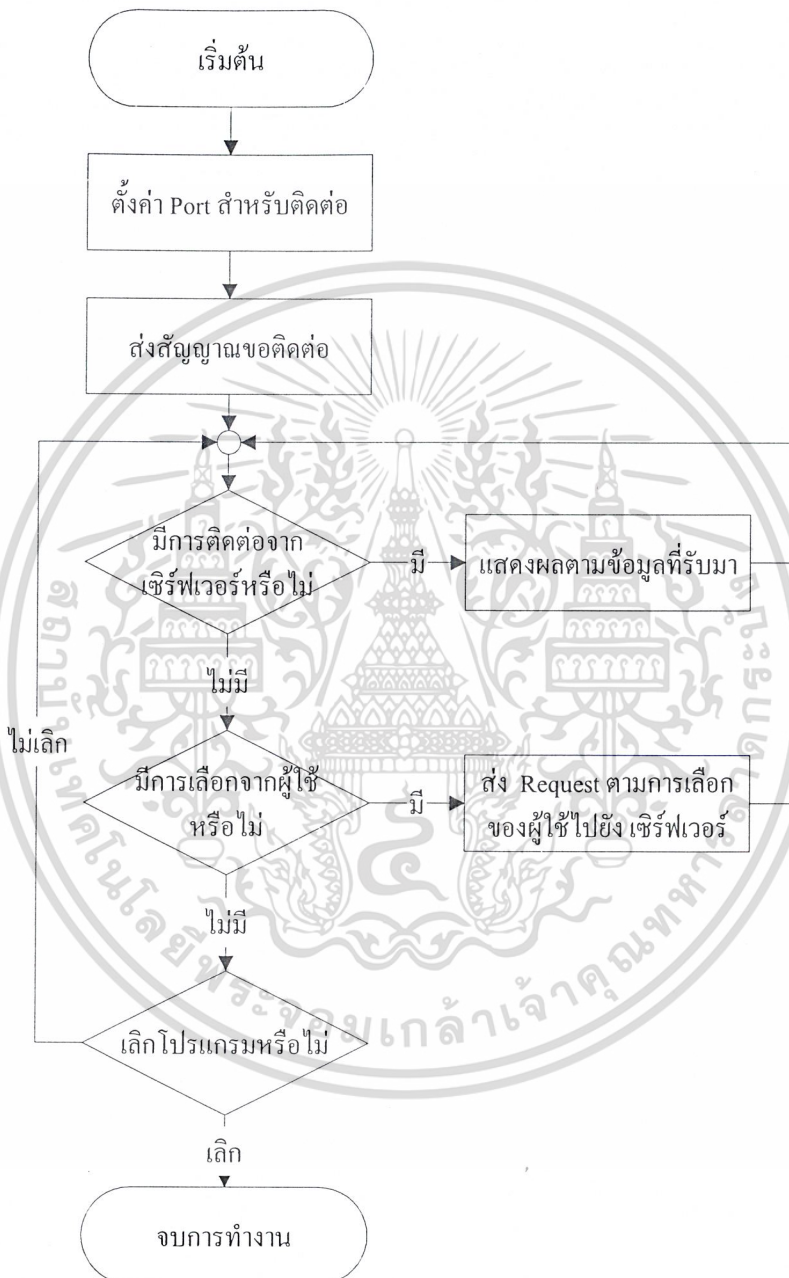


รูปที่ 30 แสดงการทำงานของโปรแกรมในส่วนที่จัดการข้อมูลตาม Request

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การทำงานของโปรแกรม Database on Internet ส่วนไคลเอนต์

การทำงานของโปรแกรมในส่วนไคลเอนต์นั้นจะทำงานตามรูปที่ 31



รูปที่ 31 แสดงการทำงานของโปรแกรมในส่วนไคลเอนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

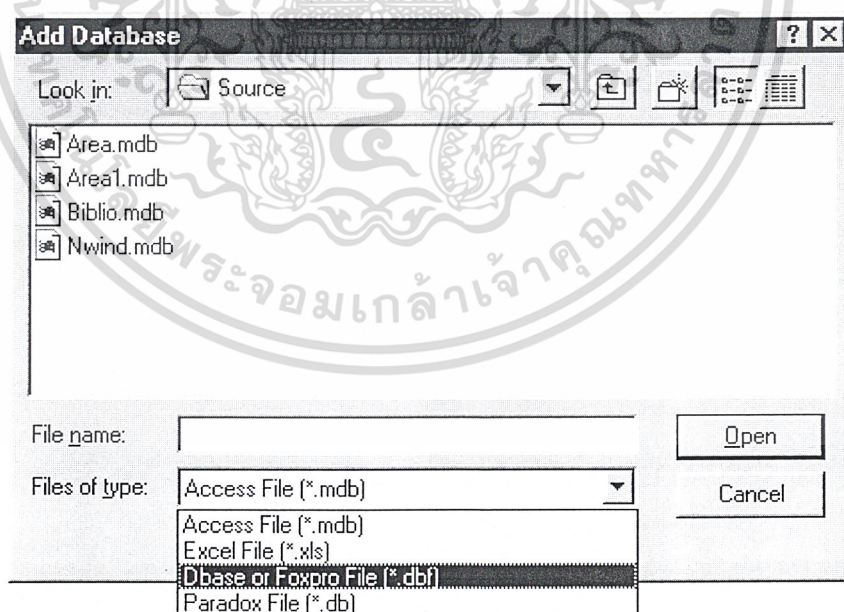
การใช้งานโปรแกรม Database on Internet

เมื่อเริ่มโปรแกรมที่เซิร์ฟเวอร์ต้องเลือกฐานข้อมูลที่ให้ผู้ใช้ดูได้ก่อน โดยการเรียกใช้โปรแกรม Server ที่อยู่บนเครื่องแล้วจะปรากฏดังรูปที่ 28



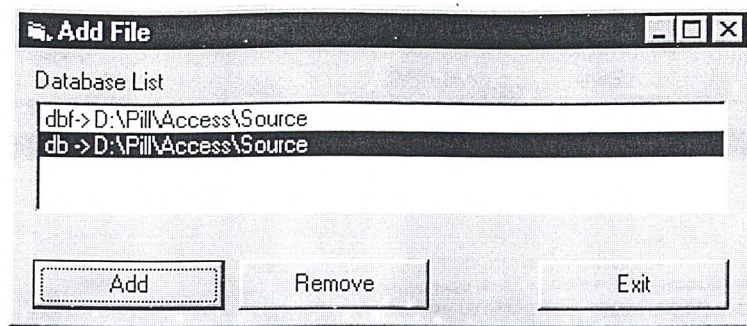
รูปที่ 32 หน้าจอเมื่อเรียกใช้โปรแกรม Server.exe

เมื่อได้หน้าจอดังรูปที่ 32 แล้ว ก็ต้องเลือกเปิดฐานข้อมูลที่ให้ผู้ใช้เรียกดูได้ โดยเลือกปุ่ม Add เพื่อเปิดฐานข้อมูล ก็จะได้หน้าจอดังรูปที่ 33 เมื่อเลือกแล้วก็พร้อมที่จะให้ผู้ใช้เลือกฐานข้อมูลได้



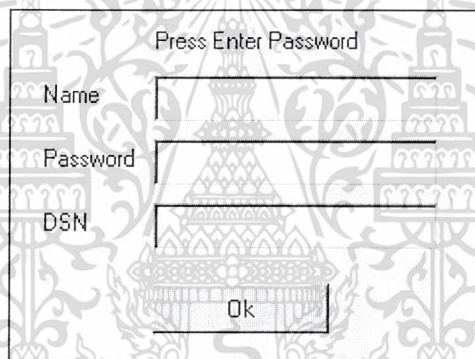
รูปที่ 33 เลือกฐานข้อมูลที่ต้องการ

เมื่อเลือกแล้วเราก็จะได้รายชื่อของฐานข้อมูลที่เปิดไว้ดังรูปที่ 34 หากไม่ต้องการฐานข้อมูลใด เอกสารที่สืบเอกสารที่ส่งมอบให้สำหรับกรณีใช้สืบเลือกการสืบหาเท่านั้น ไม่นอนุญาตให้เข้าไปใช้ประโยชน์ด้านการค้า ก็ให้เลือกปุ่ม Remove เพื่อปิดฐานข้อมูลที่ไม่ต้องการใช้ ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 34 แสดงรายชื่อฐานข้อมูลที่ได้เปิดไว้

ตอนนี้ก็จะพร้อมแล้วที่จะให้ผู้ใช้เลือกข้อมูลผ่านทางอินเทอร์เน็ต เมื่อมีผู้ใช้เรียกบริการข้อมูลขั้นแรกก็ต้องใส่รหัสผ่านก่อนดังรูปที่ 35 โดยรหัสผ่านที่จะเชื่อมกับข้อมูลของออร่าเคิลเพราะฉะนั้นถ้าใส่ผิดก็จะเข้าไปทำงานต่อไม่ได้ ถ้าไม่ต้องการเรียกดูส่วนที่เป็นฐานข้อมูลออร่าเคิลให้ลงชื่อว่า guest ก็จะเข้าไปดูข้อมูลของฐานข้อมูลที่เปิดเซิร์ฟเวอร์เปิดไว้ได้



รูปที่ 35 ใส่รหัสก่อนถึงจะเข้าได้

เมื่อเข้าไปได้ก็จะได้รับรายชื่อของตารางที่สามารถดูได้ ดังรูปที่ 36

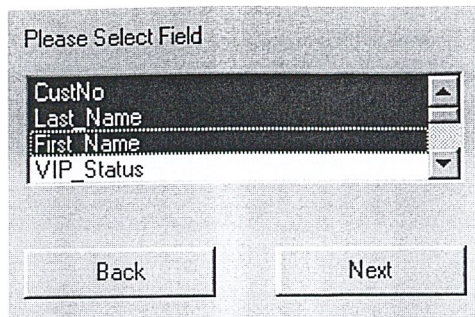
Please Select Table

dbf-> Source-> MASTER
 dbf-> Source-> INDUSTRY
 dbf-> Source-> HOLDINGS
 dbf-> Source-> CLIENTS

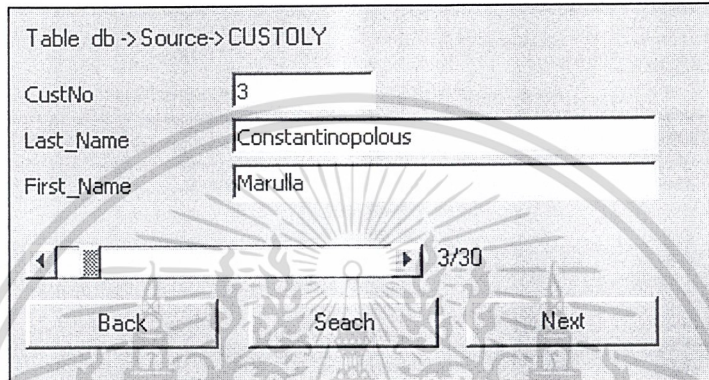
Next

รูปที่ 36 เมื่อเรียกใช้บริการ ก็จะมีรายชื่อของตารางของข้อมูลที่สามารถเรียกดูได้มาให้เลือก

หลังจากที่เลือกตารางแล้ว ก็ต้องเลือกรายการที่ต้องการดูจากตารางนั้น จากรายชื่อดังรูปที่ 37 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าแล้วก็จะ ได้ข้อมูลดังรูปที่ 38 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

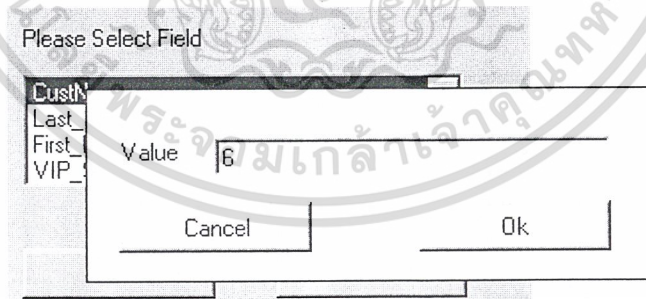


รูปที่ 37 แสดงการเลือกรายการจากตารางที่เลือกในขั้นต้น



รูปที่ 38 ข้อมูลที่ได้จากตาราง

จากจุดนี้ จะสามารถเลื่อนดูข้อมูลโดยใช้ Scrollbar หรือ ใช้ปุ่ม Search เพื่อไปดูยังข้อมูลที่ต้องการก็ได้ ถ้าตารางนี้มีความสัมพันธ์กับตารางอื่น (ในเชิง 1 to Many) แล้วก็จะสามารถเลือกดูรายละเอียดจากตารางอื่นได้โดยสองตารางนี้อาจมาจากคนละฐานข้อมูลก็ได้ (จะพิจารณาจากชื่อรายการที่ตรงกัน) โดยเลือกปุ่ม Next ดังรูปที่ 39 และ 40



รูปที่ 39 แสดงการค้นหาข้อมูล จากค่าของข้อมูลในรายการหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table db -> Source -> CUSTOLY

CustNo

Last_Name

First_Name

VIP_Status

/ 30

Table
RESERVAT

ResNo	EventNo	CustNo	NumTickets	Amt_Paid
15	8	6	7	52.5
61	13	6	4	30
99	14	6	3	22.5

รูปที่ 40 แสดงข้อมูลจากอีกตารางหนึ่งที่มีความสัมพันธ์กับตารางแรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุป และวิจารณ์

โครงการนี้จัดทำขึ้นมาเพื่อศึกษาการทำเว็บแอปพลิเคชันโดยใช้เทคโนโลยี ActiveX ในการพัฒนา โดยเทคโนโลยีนี้ ไม่โครซอฟท์ได้หลักพัฒนามาจากการทำงานของวินโดวส์ที่สามารถจะนำชิ้นงาน (object) จากแอปพลิเคชันหนึ่งไปใช้ในอีกแอปพลิเคชันหนึ่งได้โดยง่ายเหมือนตัดปะกระดาษนั่นเอง การนำเทคโนโลยีนี้มาใช้ก็เปรียบเสมือนการที่เราเขียนโปรแกรมใช้บนวินโดวส์แต่นำมันไปวางไว้บน HTML เพื่อให้เบราว์เซอร์สามารถอ่านได้นั่นเอง

แอปพลิเคชันที่ได้สร้างขึ้นมาจากโครงการนี้คือ Database on Internet โดยอาศัยหลักการของ ActiveX จึงทำให้สามารถตรวจสอบฐานข้อมูลที่อยู่ ณ เครื่องเซิร์ฟเวอร์จากที่ไหนก็ได้ที่สามารถใช้อินเตอร์เน็ต ในการเขียนโปรแกรมนั้นใช้หลักของภาษา Visual Basic ในการเขียนโดยจะแบ่งเป็นส่วนที่เขียนโปรแกรมที่เซิร์ฟเวอร์ เมื่อเขียนเสร็จก็จะ compile ออกมาให้อยู่ในรูปแบบของไฟล์ .exe เพื่อให้สามารถติดตั้งได้ทันทีที่เซิร์ฟเวอร์ อีกส่วนหนึ่งจะเป็นส่วนที่นำไปวางบน HTML ได้แก่ส่วนที่เป็นคอนโทรลนั่นเอง โดยการเขียนโปรแกรมส่วนนี้จะเริ่มจากการเขียนคอนโทรลโดยใช้โปรแกรม VBCCE ช่วยในการเขียน (โปรแกรมนี้เป็นส่วนที่ตัดมาจาก VB5 เพื่อใช้สำหรับเฉพาะการเขียนคอนโทรล) ซึ่งจะใช้ภาษา Visual Basic เช่นกัน เมื่อเขียนเสร็จแล้วจะต้อง compile ให้อยู่ในรูปของไฟล์ .ocx ที่พร้อมสำหรับการนำไปวางใน HTML โดยในการนำไปวางนี้อาจใช้โปรแกรม ActiveX Control Pad เพื่อช่วยในการตกแต่งเพื่อให้สวยงาม และการใช้คอนโทรลใน HTML นั้นจะใช้ภาษา VBScript ช่วยในการเชื่อมการทำงานในแต่ละคอนโทรล ซึ่งภาษานี้ก็มีรากฐานมาจากภาษา Visual Basic เช่นกัน จึงเปรียบเสมือนรู้เพียงภาษาเดียวกันพอ

ข้อดีสำหรับแอปพลิเคชันนี้ก็คือ

1. สามารถเรียกใช้จากที่ไหนก็ได้ที่สามารถต่ออินเทอร์เน็ต
2. การดาวน์โหลดตัวคอนโทรลไปใช้นั้นทำเพียงเฉพาะการติดต่อกครั้งแรกเท่านั้น ไม่ต้องดาวน์โหลดซ้ำในการติดต่อกครั้งใหม่ ซึ่งทำให้เร็วขึ้นสำหรับการติดต่อกครั้งต่อ ๆ ไป
3. สามารถเรียกดูฐานข้อมูลได้หลายชนิด

ข้อเสียของแอปพลิเคชันนี้ก็คือ

1. ผู้ใช้ต้องพิจารณาในด้านความปลอดภัยเองว่าจะรับคอนโทรลนั้นมาใช้หรือไม่
 2. ไม่สามารถใช้พ่วงกับเครื่องเซิร์ฟเวอร์อื่น ๆ ได้ ใช้ได้เฉพาะกับ Internet Information Server ของเอกสารนี้จึงเอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่โครซอฟท์เท่านั้น
- ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แนวทางการพัฒนาต่อ

1. เพิ่มความสามารถให้สามารถแก้ไขฐานข้อมูลได้ด้วยนอกจากการดูเพียงอย่างเดียว
2. ควรตรวจสอบผู้ที่เข้ามาดูฐานข้อมูลได้ว่าเป็นใครมาจากไหน จะให้ดูฐานข้อมูลได้หรือไม่
3. เพิ่มชนิดของฐานข้อมูลที่ต้องการใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

ต้นฉบับโปรแกรมในส่วนเซิร์ฟเวอร์

VERSION 5.00

Object = "{248DD890-BB45-11CF-9ABC-0080C7E7B78D}#1.0#0"; "MSWINSCK.OCX"

Object = "{F9043C88-F6F2-101A-A3C9-08002B2F49FB}#1.1#0"; "COMDLG32.OCX"

Begin VB.Form Serv

Caption = "Add File"

ClientHeight = 2010

ClientLeft = 60

ClientTop = 345

ClientWidth = 5550

LinkTopic = "Form1"

ScaleHeight = 2010

ScaleWidth = 5550

Begin VB.ListBox List11

Height = 870

Left = 1920

TabIndex = 15

Top = 2040

Width = 1695

End

Begin VB.ListBox List10

Height = 870

Left = 3720

TabIndex = 14

Top = 4440

Width = 1695

End

Begin VB.ListBox List9

Height = 870

Left = 1920

TabIndex = 13

Top = 4440

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Width = 1695

End

Begin VB.ListBox List8

Height = 870

Left = 120

TabIndex = 12

Top = 4440

Width = 1695

End

Begin VB.ListBox List7

Height = 870

Left = 3720

TabIndex = 11

Top = 3240

Width = 1695

End

Begin VB.ListBox List6

Height = 870

Left = 1920

TabIndex = 10

Top = 3240

Width = 1695

End

Begin VB.ListBox List5

Height = 870

Left = 120

TabIndex = 9

Top = 5640

Width = 5295

End

Begin VB.ListBox List4

Height = 870

Left = 3720

TabIndex = 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Top = 2040
Width = 1695

End

Begin VB.ListBox List3

Height = 870
Left = 120
TabIndex = 7
Top = 2040
Width = 1695

End

Begin VB.CommandButton Command4

Caption = "Load"
Height = 375
Left = 3240
TabIndex = 6
Top = 1560
Visible = 0 'False
Width = 615

End

Begin VB.ListBox List1

Height = 870
Left = 120
TabIndex = 0
Top = 3240
Width = 1695

End

Begin VB.CommandButton Command3

Caption = "Exit"
Height = 375
Left = 3960
TabIndex = 5
Top = 1560
Width = 1455

End

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Begin VB.CommandButton Command2

Caption = "Remove"
 Enabled = 0 'False
 Height = 375
 Left = 1680
 TabIndex = 3
 Top = 1560
 Width = 1455

End

Begin VB.CommandButton Command1

Caption = "Add"
 Height = 375
 Left = 120
 TabIndex = 2
 Top = 1560
 Width = 1455

End

Begin MSComDlg.CommonDialog Dialog1

Left = 120
 Top = 1440
 _ExtentX = 847
 _ExtentY = 847
 _Version = 327680
 FontSize = 1.73991e-39

End

Begin VB.ListBox List2

Height = 675
 Left = 120
 TabIndex = 1
 Top = 360
 Width = 5295

End

Begin MSWinsockLib.Winsock Winsock2

Left = 4920

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Top      = 600
_ExtentX = 741
_ExtentY = 741
End

Begin MSWinsockLib.Winsock Winsock1
Left     = 4920
Top      = 120
_ExtentX = 741
_ExtentY = 741
End

Begin VB.Label Label1
Caption  = "Database List"
Height  = 255
Left    = 120
TabIndex = 4
Top     = 120
Width   = 2295
End

End

Attribute VB_Name = "Serv"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False

Option Explicit

Private Const Part As String = "D:\Pill\access\source\"

Dim Db(0 To 10) As Database

Dim Tb1, Tb2 As Recordset

Dim SelectField, RecSel, Tb1Name, Tb2Name As String

Dim PValue, Co, dbNum1, dbNum2, RelSel As Integer

Dim OpenOra As Boolean

Private Sub Command1_Click()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 Dim I As Integer
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Dim Bool As Boolean
Dim StripName As String
Dim DbType As String

Dialog1.InitDir = Part

Dialog1.DialogTitle = "Add Database"

Dialog1.Filter = "Access File (*.mdb)|*.mdb|Excel File (*.xls)|*.xls|Dbase or Foxpro File
(*.dbf)|*.dbf|Paradox File (*.db)|*.db"

Dialog1.ShowOpen

Select Case Right(Dialog1.filename, 3)

    Case "mdb": StripName = "mdb->" & Dialog1.filename
    Case "xls": StripName = "xls->" & Dialog1.filename
    Case "dbf": StripName = "dbf->" & StripFileName(Dialog1.filename)
    Case ".db": StripName = "db ->" & StripFileName(Dialog1.filename)

End Select

Bool = True
If List2.ListCount <> 0 Then
    For I = 0 To List2.ListCount - 1
        List2.ListIndex = I
        If List2.Text = StripName Then Bool = False
    Next
End If
If StripName = "" Then Bool = False
If Bool Then List2.AddItem StripName Else Exit Sub

List2.ListIndex = List2.ListIndex + 1
Select Case Left(List2.Text, 3)

    Case "mdb": DbType = ""
    Case "xls": DbType = "Excel 5.0;"
    Case "dbf": DbType = "Dbase 5.0;"
    Case "db ": DbType = "Paradox 5.x;"

End Select

Set Db(Co) = OpenDatabase(Mid(List2.Text, 6), dbDriverNoPrompt, True, DbType)

Co = Co + 1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
List2.ListIndex = List2.ListCount - 1
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
Dim I, J As Integer
```

```
Dim DbType As String
```

```
J = List2.ListIndex
```

```
For I = J To (Co - 2)
```

```
List2.ListIndex = I + 1
```

```
Db(I).Close
```

```
Select Case Left(List2.Text, 3)
```

```
Case "mdb": DbType = ""
```

```
Case "xls": DbType = "Excel 5.0;"
```

```
Case "dbf": DbType = "Dbase 5.0;"
```

```
Case "db ": DbType = "Paradox 5.x;"
```

```
End Select
```

```
Set Db(I) = OpenDatabase(Mid(List2.Text, 6), dbDriverNoPrompt, True, DbType)
```

```
Next
```

```
Db(Co - 1).Close
```

```
Co = Co - 1
```

```
List2.RemoveItem J
```

```
Command2.Enabled = False
```

```
End Sub
```

```
Private Sub Command3_Click()
```

```
Unload Form1
```

```
Unload Serv
```

```
End Sub
```

```
Private Sub Command4_Click()
```

```
Load Form1
```

```
Form1.Show
```

```
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Form_Load()
    Winsock1.LocalPort = "1010"
    Winsock1.Listen
    Co = 0
End Sub
```

```
Private Sub List2_Click()
    Command2.Enabled = True
End Sub
```

```
Private Sub Winsock1_ConnectionRequest(ByVal requestID As Long)
    If Winsock2.State <> sckClosed Then
        Winsock2.Close
        Call AllClear
    Else: Call SetList
    End If
    Winsock2.Accept requestID
    Winsock2.SendData "Yes{"
End Sub
```

```
Private Sub Winsock2_DataArrival(ByVal bytesTotal As Long)
    Dim strData, ChkData, Name, Password, DSN, sConnect As String
    Dim I, J As Integer
    Winsock2.GetData strData, vbString
    While InStr(strData, "{";) <> 0
        I = InStr(strData, "{";)
        ChkData = Left(strData, I - 1)
        strData = Mid(strData, I + 2)
        Select Case Left(ChkData, 1)
            Case "A":
                I = InStr(ChkData, ";")
                Name = Mid(ChkData, 2, I - 2)
```

```
                Name = LCase(Name)
```

```
                J = InStr(I + 1, ChkData, ";")
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Password = Mid(ChkData, I + 2, J - 2 - I)
DSN = Mid(ChkData, J + 2)
OpenOra = False
If (Name = "guest") And (Password = "") Then
Else
    On Error GoTo Err
    sConnect = "ODBC;DSN=" & DSN & ";"
    sConnect = sConnect & "UID=" & Name & ";"
    sConnect = sConnect & "PWD=" & Password & ";"
    Set Db(Co) = OpenDatabase("", 0, 0, sConnect)
    Co = Co
    OpenOra = True
End If
Command1.Enabled = False
Command2.Enabled = False
List2.Enabled = False
Call SendTable
If OpenOra Then Call SendOraTable
Call CheckRelation
GoTo Quit

```

Err:

```

Winsock2.SendData "AERROR{"
Case "B":
    Call OpenTable1(Mid(ChkData, 2))
    Call SendField
Case "C":
    J = CountRelation
    SelectField = CheckType(Mid(ChkData, 2))
    Call SendHeadRec(SelectField, J)
    Tb1.MoveFirst
    PValue = 1
    Call SendRecord(SelectField)

```

Case "D":

```
Call MoveRecord(Mid(ChkData, 2))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Call SendRecord(SelectField)
```

```
Case "E":
```

```
Call OpenTable2(Mid(ChkData, 2))
```

```
Call SendField2
```

```
Case "F":
```

```
RecSel = CheckType2(Mid(ChkData, 2))
```

```
Call SendHeadRec2(RecSel)
```

```
Call SendRecord2(RecSel)
```

```
Case "G":
```

```
I = InStr(ChkData, "};")
```

```
Call Search(Mid(ChkData, 2, I - 2), Mid(ChkData, I + 2))
```

```
Case "H":
```

```
Call SendRelation
```

```
Case "I":
```

```
Call SendRecord2(RecSel)
```

```
End Select
```

```
Quit:
```

```
Wend
```

```
End Sub
```

```
Private Sub SendTable()
```

```
Dim Str1, Str2 As String
```

```
Dim TbTemp As Recordset
```

```
Dim I, J As Integer
```

```
Dim dbTemp
```

```
Winsock2.SendData "AClear{";
```

```
For J = 0 To (List2.ListCount - 1)
```

```
List1.ListIndex = J
```

```
For Each dbTemp In Db(J).TableDefs
```

```
If dbTemp.Attributes = 0 Then
```

```
Str1 = List1.Text & "->" & dbTemp.Name
```

```
Winsock2.SendData "A" & Str1 & "{";
```

```
List3.AddItem Mid(List1.Text, 6)
```

```
List1.AddItem J
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

List4.AddItem dbTemp.Name
Set TbTemp = Db(J).OpenRecordset(dbTemp.Name)
Str2 = ""
For I = 0 To TbTemp.Fields.Count - 1
    Str2 = Str2 & TbTemp.Fields(I).Name & ","
Next
List5.AddItem Str2
TbTemp.Close
End If
Next
Next
End Sub

```

```

Private Sub CheckRelation()
Dim Str1, Str2, Temp As String
Dim I, J, K, L As Integer
For I = 0 To List5.ListCount - 1
    List5.ListIndex = I
    K = InStr(List5.Text, ",")
    Str1 = Left(List5.Text, K - 1)
    For J = 0 To List5.ListCount - 1
        If I <> J Then
            List5.ListIndex = J
            K = 0
            L = InStr(List5.Text, ",")
            While L <> 0
                Str2 = Mid(List5.Text, K + 1, L - 1 - K)
                If Str1 = Str2 Then
                    List4.ListIndex = I
                    List6.AddItem List4.Text
                    List11.ListIndex = I
                    List9.AddItem List11.Text
                    List4.ListIndex = J
                    List7.AddItem List4.Text

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

List11.ListIndex = J
List10.AddItem List11.Text
List8.AddItem Str1
End If
K = L
L = InStr(K + 1, List5.Text, ",")
Wend
End If
Next
Next
End Sub

Private Sub OpenTable1(Str As String)
Dim I, J, K As Integer
Dim StrTemp As String
I = InStr(Str, "->")
J = Left(Str, I - 1)
I = InStr(I + 2, Str, "->")
Str = Mid(Str, I + 2)
List11.ListIndex = J
StrTemp = "Select * from " & Str
Set Tb1 = Db(List11.Text).OpenRecordset(StrTemp, dbOpenSnapshot, dbSQLPassThrough,
dbReadOnly)
Tb1Name = Str
dbNum1 = List11.Text
Tb1.MoveLast
Tb1.MoveFirst

End Sub

```

```
Private Sub OpenTable2(RelaNum As Integer)
```

```
Dim I, J, K As Integer
```

```
Dim StrTemp As String
```

```
K = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

For I = 0 To List6.ListCount - 1
    List6.ListIndex = I
    List9.ListIndex = I
    If (List6.Text = Tb1Name) And (List9.Text = dbNum1) Then
        If RelaNum = K Then
            List7.ListIndex = I
            List10.ListIndex = I
            J = List10.Text
            StrTemp = "Select * from " & List7.Text
            Set Tb2 = Db(J).OpenRecordset(StrTemp, dbOpenSnapshot, dbSQLPassThrough,
dbReadOnly)
            Tb2Name = List7.Text
            Tb2.MoveLast
            Tb2.MoveFirst
            dbNum2 = J
            RelSel = I
            Exit For
        End If
        K = K + 1
    End If
Next
End Sub

Private Sub SendField()
Dim dbTemp
Winsock2.SendData "BClear{;"
For Each dbTemp In Tb1.Fields
    If dbTemp.Type <> 11 Then
        Winsock2.SendData "B" & dbTemp.Name & "{;"
    End If
Next
End Sub

```

```
Dim dbTemp
```

```
Winsock2.SendData "Eclear{;"
```

```
For Each dbTemp In Tb2.Fields
```

```
    If dbTemp.Type <> 11 Then
```

```
        Winsock2.SendData "E" & dbTemp.Name & "{;"
```

```
    End If
```

```
Next
```

```
End Sub
```

```
Private Sub SendHeadRec(ByVal RecSel As String, RelationOfTable As Integer)
```

```
Dim Str, J
```

```
Dim X As Integer
```

```
While InStr(RecSel, ";") <> 0
```

```
X = InStr(RecSel, ";")
```

```
J = Left(RecSel, X - 1)
```

```
RecSel = Mid(RecSel, X + 2)
```

```
X = J
```

```
Str = Str & Tb1.Fields(X).Name & "+-" & Tb1.Fields(X).Type & "+-" & Tb1.Fields(X).Size & ";"
```

```
Wend
```

```
X = RecSel
```

```
Str = Str & Tb1.Fields(X).Name & "+-" & Tb1.Fields(X).Type & "+-" & Tb1.Fields(X).Size & ";"
```

```
Winsock2.SendData "C" & Str & Tb1.RecordCount & "+-" & RelationOfTable & ";"
```

```
End Sub
```

```
Private Sub SendRecord(ByVal RecSel As String)
```

```
Dim Str, J, StrTemp
```

```
Dim X As Integer
```

```
On Error GoTo Quit
```

```
While InStr(RecSel, ";") <> 0
```

```
X = InStr(RecSel, ";")
```

```
J = Left(RecSel, X - 1)
```

```
RecSel = Mid(RecSel, X + 2)
```

```
X = J
```

```
Str = Str & Tb1.Fields(X).Value & ";"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Wend
X = RecSel
Str = Str & Tb1.Fields(X).Value & "+-" & PValue
Winsock2.SendData "D" & Str & "{";"
Exit Sub
Quit:
Winsock2.SendData "DERROR{";"
End Sub

```

```

Private Sub SendHeadRec2(ByVal RecSel As String)
Dim Str, J
Dim X As Integer
While InStr(RecSel, ";") <> 0
X = InStr(RecSel, ";")
J = Left(RecSel, X - 1)
RecSel = Mid(RecSel, X + 2)
X = J
Str = Str & Tb2.Fields(X).Name & "+-" & Tb2.Fields(X).Type & "{";"
Wend
X = RecSel
Str = Str & Tb2.Fields(X).Name & "+-" & Tb2.Fields(X).Type & "{";"
Winsock2.SendData "F" & Tb2Name & ";" & Str & "{";"
End Sub

```

```

Private Sub SendRecord2(ByVal RecSel As String)
Dim Str, J, tempstr
Dim HaveRec As Boolean
Dim I As Integer
List8.ListIndex = RelSel
HaveRec = False
Tb2.MoveFirst
While Not Tb2.EOF
If Tb2.Fields(List8.Text).Value = Tb1.Fields(List8.Text).Value Then

```

```

tempstr = RecSel
While InStr(tempstr, ";") <> 0
    I = InStr(tempstr, ";")
    J = Left(tempstr, I - 1)
    tempstr = Mid(tempstr, I + 2)
    I = J
    Str = Str & Tb2.Fields(I).Value & ";"
Wend
I = tempstr
Str = Str & Tb2.Fields(I).Value
Winsock2.SendData "G" & Str & ";"
Str = ""
HaveRec = True
End If
Tb2.MoveNext
Wend
If HaveRec Then
    Winsock2.SendData "GALL-L{"
Else
    Winsock2.SendData "GALL-0{"
End If
End Sub

```

```
Private Sub MoveRecord(Value As Integer)
```

```

If Tb1.RecordCount > 0 Then
    If Tb1.RecordCount > 99 Then
        If Value - PValue = 1 Then
            Tb1.MoveNext
        ElseIf Value - PValue = -1 Then
            Tb1.MovePrevious
        Else
            If Tb1.RecordCount > 32767 Then

```

```

            Tb1.PercentPosition = (Value / 32767) * 100 + 0.005

```

```

            Value = Int((Tb1.RecordCount * (Tb1.PercentPosition * 0.01)) + 1)

```

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Else
    Tb1.MoveFirst
    Tb1.Move Value - 1
End If
End If
Else
    Tb1.MoveFirst
    Tb1.Move Value - 1
End If
PValue = Value
End If
End Sub

```

```
Private Function CountRelation()
```

```
Dim I, J As Integer
```

```
I = 0
```

```
For J = 0 To List6.ListCount - 1
```

```
List6.ListIndex = J
```

```
List9.ListIndex = J
```

```
If (List6.Text = Tb1Name) And (List9.Text = dbNum1) Then
```

```
I = I + 1
```

```
End If
```

```
Next
```

```
CountRelation = I
```

```
End Function
```

```
Private Sub SendRelation()
```

```
Dim I As Integer
```

```
Winsock2.SendData "HClear{;"
```

```
For I = 0 To List6.ListCount - 1
```

```
List6.ListIndex = I
```

```
List9.ListIndex = I
```

```
If (List6.Text = Tb1Name) And (List9.Text = dbNum1) Then
```

```
List7.ListIndex = I
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Winsock2.SendData "H" & List7.Text & "{";
End If
Next
End Sub

```

```
Private Sub SetList()
```

```
Dim I As Integer
```

```
Dim Temp As String
```

```
If List2.ListCount <> 0 Then
```

```
For I = 0 To List2.ListCount - 1
```

```
List2.ListIndex = I
```

```
Temp = List2.Text
```

```
While InStr(Temp, "\") <> 0
```

```
Temp = Mid(Temp, InStr(Temp, "\") + 1)
```

```
Wend
```

```
List1.AddItem Left(List2.Text, 5) & Temp
```

```
Next
```

```
End If
```

```
End Sub
```

```
Private Function CheckType(OldSel As String)
```

```
Dim dbTemp, RecSel
```

```
Dim Str, J
```

```
Dim X, Y As Integer
```

```
For Y = 0 To Tb1.Fields.Count - 1
```

```
If Tb1.Fields(Y).Type = 11 Then
```

```
RecSel = OldSel
```

```
While InStr(RecSel, ";") <> 0
```

```
X = InStr(RecSel, ";")
```

```
J = Left(RecSel, X - 1)
```

```
RecSel = Mid(RecSel, X + 2)
```

```
X = J
```

```
.If X < Y Then Str = Str & X & ";" Else Str = Str & (X + 1) & ";"
```

```
Wend
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    If RecSel < Y Then OldSel = Str & RecSel Else OldSel = Str & (RecSel + 1)
    End If
Next
    CheckType = OldSel
End Function

```

```

Private Function CheckType2(OldSel As String)

```

```

    Dim dbTemp, RecSel

```

```

    Dim Str, J

```

```

    Dim X, Y As Integer

```

```

    For Y = 0 To Tb2.Fields.Count - 1

```

```

        If Tb2.Fields(Y).Type = 11 Then

```

```

            RecSel = OldSel

```

```

            While InStr(RecSel, ";") <> 0

```

```

                X = InStr(RecSel, ";")

```

```

                J = Left(RecSel, X - 1)

```

```

                RecSel = Mid(RecSel, X + 2)

```

```

                X = J

```

```

                If X < Y Then Str = Str & X & ";" Else Str = Str & (X + 1) & ";"

```

```

            Wend

```

```

            If RecSel < Y Then OldSel = Str & RecSel Else OldSel = Str & (RecSel + 1)

```

```

        End If

```

```

    Next

```

```

    CheckType2 = OldSel

```

```

End Function

```

```

Private Sub Search(FieldValue As String, FieldName As String)

```

```

    Dim X, Y As Integer

```

```

    Dim B As Boolean

```

```

    Y = 0

```

```

    B = False

```

```

    If FieldName = "" Then FieldName = Null

```

```

    While Tbl.Fields(Y).Name <> FieldName

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    Y = Y + 1
Wend
X = PValue
While (Tbl.Fields(Y).Value <> FieldValue) And (Tbl.RecordCount > X)
    X = X + 1
    Tbl.MoveNext
Wend
If (Tbl.Fields(Y).Value = FieldValue) Then
    B = True
Else
    X = 1
    Tbl.MoveFirst
    While (Tbl.Fields(Y).Value <> FieldValue) And (X <= PValue)
        X = X + 1
        Tbl.MoveNext
    Wend
    If (Tbl.Fields(Y).Value = FieldValue) Then B = True
End If
If Not B Then
    Tbl.MoveFirst
    Tbl.Move PValue - 1
    Winsock2.SendData "DCan'tfind{"
Else
    PValue = X
    Call SendRecord(SelectField)
End If
End Sub

```

```
Function StripFileName(rsFileName As String) As String
```

```
On Error Resume Next
```

```
Dim I As Integer
```

```
For I = Len(rsFileName) To 1 Step -1
```

```
    If Mid(rsFileName, I, 1) = "\" Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Exit For
End If
Next

```

```
StripFileName = Mid(rsFileName, 1, I - 1)
```

```
End Function
```

```
Private Sub AllClear()
```

```
List3.Clear
```

```
List4.Clear
```

```
List5.Clear
```

```
List6.Clear
```

```
List7.Clear
```

```
List8.Clear
```

```
List9.Clear
```

```
List10.Clear
```

```
List11.Clear
```

```
End Sub
```

```
Private Sub SendOraTable()
```

```
Dim dbTemp1, dbTemp2 As Recordset
```

```
Dim TbTemp As Recordset
```

```
Dim Str1, Str2 As String
```

```
Dim I, J As Integer
```

```
List1.AddItem "Oracle"
```

```
List1.ListIndex = Co
```

```
Set dbTemp1 = Db(Co).OpenRecordset("Select Table_name from sys.user_tables", dbOpenSnapshot,
dbSQLPassThrough, dbReadOnly)
```

```
dbTemp1.MoveLast
```

```
dbTemp1.MoveFirst
```

```
Set dbTemp2 = Db(Co).OpenRecordset("select Table_name,column_name from sys.user_tab_columns";
dbOpenSnapshot, dbSQLPassThrough, dbReadOnly)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

dbTemp2.MoveLast
For I = 0 To dbTemp1.RecordCount - 1
    List1.ListIndex = Co
    Str1 = List1.Text & "->" & dbTemp1.Fields("Table_name").Value
    Winsock2.SendData "A" & Str1 & "{;"
    List3.AddItem List1.Text
    List11.AddItem (Co)
    List4.AddItem dbTemp1.Fields("Table_name").Value
    Str2 = ""
    dbTemp2.MoveFirst
    While Not dbTemp2.EOF
        If dbTemp1.Fields("table_name").Value = dbTemp2.Fields("table_name").Value Then
            Str2 = Str2 & dbTemp2.Fields("column_name").Value & ","
        End If
        dbTemp2.MoveNext
    Wend
    List5.AddItem Str2
    dbTemp1.MoveNext
Next
dbTemp1.Close
dbTemp2.Close
Co = Co + 1
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

ต้นฉบับโปรแกรมในส่วนไคลเอนต์

VERSION 5.00

Object = "{248DD890-BB45-11CF-9ABC-0080C7E7B78D}#1.0#0"; "MSWINSCK.OCX"

Object = "{5E9E78A0-531B-11CF-91F6-C2863C385E30}#1.0#0"; "MSFLXGRD.OCX"

Begin VB.UserControl Show

BackColor = &H00FFFFFF&

ClientHeight = 7200

ClientLeft = 0

ClientTop = 0

ClientWidth = 5430

ScaleHeight = 7200

ScaleWidth = 5430

Begin VB.PictureBox Picture8

Appearance = 0 'Flat

BackColor = &H00C0FFC0&

ForeColor = &H80000008&

Height = 2655

Left = 0

ScaleHeight = 2625

ScaleWidth = 3585

TabIndex = 36

Top = 0

Width = 3615

Begin VB.TextBox Text4

BeginProperty Font

Name = "MS Sans Serif"

Size = 9.75

Charset = 222

Weight = 400

Underline = 0 'False

Italic = 0 'False

เอกสารนี้เป็น ~~Strikethrough = 0 'False~~ ทรัพย์สินทางปัญญาของสำนักงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

EndProperty
Height      = 360
HideSelection = 0 'False
IMEMode     = 3 'DISABLE
Left       = 1080
TabIndex   = 45
Top        = 1440
Width      = 2175

```

End

Begin VB.TextBox Text3

BeginProperty Font

```

Name      = "MS Sans Serif"
Size      = 9.75
Charset   = 222
Weight    = 400
Underline = 0 'False
Italic    = 0 'False
Strikethrough = 0 'False

```

EndProperty

```

Height      = 360
HideSelection = 0 'False
IMEMode     = 3 'DISABLE
Left       = 1080
PasswordChar = "*"
TabIndex   = 42
Top        = 960
Width      = 2175

```

End

Begin VB.TextBox Text2

BeginProperty Font

```

Name      = "MS Sans Serif"
Size      = 9.75
Charset   = 222
Weight    = 400

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Underline = 0 'False
Italic = 0 'False
Strikethrough = 0 'False
```

```
EndProperty
```

```
Height = 360
Left = 1080
TabIndex = 39
Top = 480
Width = 2175
```

```
End
```

```
Begin VB.CommandButton Command11
```

```
BackColor = &H0080FF80&
Caption = "Cancel"
Height = 375
Left = 240
Style = 1 'Graphical
TabIndex = 38
Top = 2040
Visible = 0 'False
Width = 1215
```

```
End
```

```
Begin VB.CommandButton Command10
```

```
BackColor = &H0080FF80&
Caption = "Ok"
Height = 375
Left = 1080
Style = 1 'Graphical
TabIndex = 37
Top = 2040
Width = 1335
```

```
End
```

```
Begin VB.Label Label10
```

```
BackColor = &H00C0FFC0&
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
Caption = "DSN"
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Height    = 255
Left      = 240
TabIndex  = 44
Top       = 1480
Width     = 735

```

```
End
```

```
Begin VB.Label Label9
```

```

BackColor = &H00C0FFC0&
Caption    = "Password"
Height     = 255
Left       = 240
TabIndex   = 43
Top        = 1000
Width      = 735

```

```
End
```

```
Begin VB.Label Label8
```

```

BackColor = &H00C0FFC0&
Caption    = "Name"
Height     = 255
Left       = 240
TabIndex   = 41
Top        = 520
Width      = 495

```

```
End
```

```
Begin VB.Label Label7
```

```

BackColor = &H00C0FFC0&
Caption    = "Press Enter Password"
Height     = 255
Left       = 1080
TabIndex   = 40
Top        = 120
Width      = 2175

```

```
End
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

Left      = 960
TabIndex  = 30
Top       = 360
Width     = 3015

```

```
End
```

```
Begin VB.Label Label4
```

```

BackColor = &H00C0FFFF&
Caption    = "Value"
Height     = 255
Left       = 240
TabIndex  = 29
Top        = 360
Width     = 495

```

```
End
```

```
End
```

```
Begin VB.PictureBox Picture1
```

```

Appearance = 0 'Flat
BackColor   = &H00C0C0C0&
BorderStyle = 0 'None
ForeColor   = &H800000008&
Height      = 2415
Left        = 0
ScaleHeight = 2415
ScaleWidth  = 3615
TabIndex    = 0
Top         = 0
Width       = 3615

```

```
Begin VB.ListBox List6
```

```

Height     = 1065
Left        = 120
TabIndex    = 2
Top         = 480

```

```
Visible    = 0 'False
```

```
Width      = 3375
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End

Begin VB.ListBox List5

Height = 1065

Left = 120

TabIndex = 27

Top = 480

Width = 3375

End

Begin VB.ListBox List4

Height = 1065

Left = 120

MultiSelect = 1 'Simple

TabIndex = 21

Top = 480

Visible = 0 'False

Width = 3375

End

Begin VB.ListBox List3

Height = 1065

Left = 120

MultiSelect = 1 'Simple

TabIndex = 6

Top = 480

Width = 3375

End

Begin VB.ListBox List2

Height = 1065

Left = 120

TabIndex = 5

Top = 480

Width = 3375

End

Begin VB.CommandButton Command2

Caption = "Next"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Height    = 375
Left      = 2040
TabIndex  = 4
Top       = 1800
Width     = 1455

```

```
End
```

```
Begin VB.CommandButton Command1
```

```

Caption    = "Back"
Height     = 375
Left       = 120
TabIndex   = 3
Top        = 1800
Width      = 1455

```

```
End
```

```
Begin VB.Label Label1
```

```

Caption    = "Please Select File"
Height     = 255
Left       = 120
TabIndex   = 1
Top        = 120
Width      = 2535

```

```
End
```

```
End
```

```
Begin VB.PictureBox Picture7
```

```

Appearance = 0 'Flat
BackColor   = &H00C0C0FF&
ForeColor   = &H80000008&
Height      = 1455
Left        = 600
ScaleHeight = 1425
ScaleWidth  = 2625
TabIndex    = 33

```

```
Top        = 600
```

```
Visible    = 0 'False
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Width      = 2655
Begin VB.CommandButton Command9
    BackColor = &H008080FF&
    Caption   = "Ok"
    Height    = 375
    Left      = 600
    Style     = 1 'Graphical
    TabIndex  = 34
    Top       = 840
    Width     = 1455

```

End

```

Begin VB.Label Label5

```

```

    BackColor = &H00C0C0FF&
    Caption   = "Not Found !"
    Height    = 255
    Left      = 720
    TabIndex  = 35
    Top       = 360
    Width     = 975

```

End

End

```

Begin VB.PictureBox Picture4

```

```

    Appearance = 0 'Flat
    BackColor  = &H00C0C0C0&
    ForeColor  = &H80000008&
    Height     = 3255
    Left       = 0
    ScaleHeight = 3225
    ScaleWidth  = 1785
    TabIndex   = 22
    Top        = 3240
    Visible    = 0 'False

```

```

    Width     = 1815

```

```

Begin MSFlexGridLib.MSFlexGrid Grid1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Height    = 1095
Left      = 120
TabIndex  = 25
Top       = 600
Width     = 3615
_ExtentX = 6376
_ExtentY = 1931
_Version  = 327680
Rows      = 1
Cols      = 0
FixedRows = 0
FixedCols = 0

```

```
End
```

```
Begin VB.CommandButton Command6
```

```

Caption    = "Back"
Height     = 375
Left       = 120
TabIndex   = 23
Top        = 2760
Width      = 1455

```

```
End
```

```
Begin VB.Label Label6
```

```

BackColor  = &H00C0C0C0&
Caption    = "Label"
Height     = 375
Left       = 120
TabIndex   = 24
Top        = 120
Width      = 2655

```

```
End
```

```
End
```

```
Begin MSWinsockLib.Winsock Winsock1
```

```
Left      = 3600
```

```
Top       = 360
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
_ExtentX = 741
```

```
_ExtentY = 741
```

```
End
```

```
Begin VB.PictureBox Picture2
```

```
Appearance = 0 'Flat
```

```
BackColor = &H00C0C0C0&
```

```
ForeColor = &H80000008&
```

```
Height = 3255
```

```
Left = 0
```

```
ScaleHeight = 3225
```

```
ScaleWidth = 5385
```

```
TabIndex = 7
```

```
Top = 0
```

```
Visible = 0 'False
```

```
Width = 5415
```

```
Begin VB.CommandButton Command5
```

```
Caption = "Seach"
```

```
Height = 375
```

```
Left = 1800
```

```
TabIndex = 26
```

```
Top = 2760
```

```
Width = 1455
```

```
End
```

```
Begin VB.CommandButton Command4
```

```
Caption = "Next"
```

```
Height = 375
```

```
Left = 3480
```

```
TabIndex = 20
```

```
Top = 2760
```

```
Width = 1455
```

```
End
```

```
Begin VB.PictureBox Picture3
```

```
Appearance = 0 'Flat
```

```
BackColor = &H00C0C0C0&
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

BorderStyle = 0 'None
ForeColor = &H80000008&
Height = 1815
Left = 120
ScaleHeight = 1815
ScaleWidth = 4815
TabIndex = 14
Top = 480
Width = 4815

```

```
Begin VB.PictureBox picFields
```

```

Appearance = 0 'Flat
BackColor = &H00C0C0C0&
BorderStyle = 0 'None
ForeColor = &H80000008&
Height = 375
Left = 0
ScaleHeight = 372
ScaleMode = 0 'User
ScaleWidth = 4812
TabIndex = 17
TabStop = 0 'False
Top = 0
Visible = 0 'False
Width = 4815

```

```
Begin VB.TextBox txtFieldData
```

```

BackColor = &H00FFFFFF&
DataSource = "Data1"

```

```
BeginProperty Font
```

```

Name = "Tahoma"
Size = 8.25
Charset = 0
Weight = 400
Underline = 0 'False

```

```
Italic = 0 'False
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Strikethrough = 0 'False

EndProperty

ForeColor = &H00000000&

Height = 288

Index = 0

Left = 1560

TabIndex = 18

Top = 0

Visible = 0 'False

Width = 3252

End

Begin VB.Label lblFieldName

BeginProperty Font

Name = "Tahoma"

Size = 8.25

Charset = 0

Weight = 400

Underline = 0 'False

Italic = 0 'False

Strikethrough = 0 'False

EndProperty

ForeColor = &H00000000&

Height = 252

Index = 0

Left = 0

TabIndex = 19

Top = 60

Visible = 0 'False

Width = 1572

End

End

Begin VB.PictureBox Picture5

Appearance = 0 'Flat

BorderStyle = 0 'None

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ForeColor = &H80000008&
Height = 285
Left = 0
ScaleHeight = 298.153
ScaleMode = 0 'User
ScaleWidth = 5313.75
TabIndex = 15
TabStop = 0 'False
Top = 2400
Visible = 0 'False
Width = 5310
Begin VB.Label Label3
    Height = 255
    Left = 3000
    TabIndex = 16
    Top = 0
    Width = 1695
End
End
End
Begin VB.CommandButton Command3
    Caption = "Back"
    Height = 375
    Left = 120
    TabIndex = 13
    Top = 2760
    Width = 1455
End
Begin VB.VScrollBar vsbScrollBar
    Height = 2370
    LargeChange = 3000
    Left = 5040
    SmallChange = 300
    TabIndex = 11

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Top      = 360
Visible  = 0 'False
Width    = 255

```

```
End
```

```
Begin VB.PictureBox picMoveButtons
```

```

Appearance = 0 'Flat
BackColor   = &H00C0C0C0&
BorderStyle = 0 'None
ForeColor   = &H80000008&
Height      = 285
Left        = 120
ScaleHeight = 298.153
ScaleMode   = 0 'User
ScaleWidth  = 4938.485
TabIndex    = 8
TabStop     = 0 'False
Top         = 2400
Visible     = 0 'False
Width       = 4935

```

```
Begin VB.HScrollBar hsc1CurrRow
```

```

Height     = 255
Left       = 0
Max        = 100
Min        = 1
TabIndex   = 9
Top        = 0
Value      = 1
Width      = 3015

```

```
End
```

```
Begin VB.Label lblStatus
```

```

Height     = 255
Left       = 3120
TabIndex   = 10
Top        = 0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Width      = 1695
End
End
Begin VB.Label Label2
BackColor  = &H00C0C0C0&
Caption    = "Label"
Height     = 255
Left       = 120
TabIndex  = 12
Top        = 120
Width      = 2655
End
End
End
Attribute VB_Name = "Show"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = False
Attribute VB_Exposed = True
Option Explicit

Private AllHeight
Private Hostname
Public Event Resize()
Dim FieldCount As Integer
Dim RecordCount As Long
Dim Change, ShowPic4 As Boolean
Dim RelationOfTable As Integer

Private Sub Command1_Click()
Dim I As Integer
Dim Str As String
If List3.Visible = True Then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

List3.Visible = False
List2.Visible = True
List2.ListIndex = -1
Command2.Enabled = False
Command1.Visible = False
ShowPic4 = False
Else
  If List4.Visible = True Then
    UserControl.Height = Picture2.Height
    Call MovePic1L
    Call Page4
  Else
    If List5.Visible = True Then
      List5.Visible = False
      UserControl.Height = Picture2.Height
      Call Page4
      UserControl.Width = Picture2.Width
    Else
      If List6.Visible = True Then
        Picture1.Visible = False
        Call MovePic1L
        Picture2.Enabled = True
      End If
    End If
  End If
End If
End If
End Sub

```

```

Private Sub Command10_Click()
  Screen.MousePointer = vbHourglass
  Picture8.Visible = False
  If Label7.Caption = "Press Enter Password" Then

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น ยกเว้นหากมีเหตุที่เปลี่ยนแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

If Hostname = "" Then Winsock1.RemoteHost = "127.0.0.1"
Winsock1.Connect
Else
Winsock1.SendData "A" & Text2.Text & "};" & Text3.Text & "};" & Text4.Text & "};"
End If
Command1.Enabled = False
Command2.Enabled = False
Change = True
End Sub

```

```

Private Sub Command2_Click()
Dim Str As String
Dim I As Integer
Screen.MousePointer = vbHourglass
If List2.Visible Then
Winsock1.SendData "B" & List2.ListIndex & "->" & Mid(List2.Text, 6) & "};"
Else
If List3.Visible Then
For I = 0 To List3.ListCount - 1
If List3.Selected(I) Then
Str = Str & "};" & I
End If
Next
Str = Mid(Str, 3)
Winsock1.SendData "C" & Str & "};"
Else
If List4.Visible Then
For I = 0 To List4.ListCount - 1
If List4.Selected(I) Then
Str = Str & "};" & I
End If
Next
Str = Mid(Str, 3)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 Winsock1.SendData "F" & Str & "};"
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ข้อมูลและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Picture1.Top = 0
Picture1.Left = 0
Picture1.BorderStyle = 0

Else

If List5.Visible Then

Picture6.Visible = True

Screen.MousePointer = vbDefault

UserControl.Width = 4935

Text1.SetFocus

Picture1.Enabled = False

Else

If List6.Visible Then

Winsock1.SendData "E" & List6.ListIndex & "{"

End If

End If

End If

End If

End If

End Sub

```

```
Private Sub Command3_Click()
```

```
Dim I As Integer
```

```
For I = 1 To FieldCount
```

```
Unload lblFieldName(I)
```

```
Unload txtFieldData(I)
```

```
Next
```

```
Change = False
```

```
hsc1CurrRow.Value = 1
```

```
Change = True
```

```
txtFieldData(0).Text = ""
```

```
picFields.Height = 375
```

```
Picture2.Width = 5310
```

```
vsbScrollBar.Visible = False
```

```

List3.Visible = True
Command2.Enabled = False
For I = 0 To List3.ListCount - 1
    List3.Selected(I) = False
Next
End Sub

```

```

Private Sub Command4_Click()
    Screen.MousePointer = vbHourglass
    Call MovePic1R
    If RelationOfTable > 1 Then
        Winsock1.SendData "H" & "{";"
    Else
        Winsock1.SendData "E0{";"
    End If
End Sub

```

```

Private Sub Command5_Click()
    Dim I As Integer
    Call Page3
    List5.Visible = True
    For I = 0 To List5.ListCount - 1
        List5.Selected(I) = False
    Next
    Command2.Enabled = False
End Sub

```

```

Private Sub Command6_Click()
    Call MovePic1R
    Call Page5
    Picture4.Visible = False
    ShowPic4 = False

```

```

    UserControl.Width = Picture2.Width

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 End Sub
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Command7_Click()
```

```
    Text1.Text = ""
```

```
    Picture6.Visible = False
```

```
    Picture1.Enabled = True
```

```
    UserControl.Width = 3975
```

```
End Sub
```

```
Private Sub Command8_Click()
```

```
    Winsock1.SendData "G" & Text1.Text & ";" & List5.Text & "{"
```

```
    Text1.Text = ""
```

```
    Picture6.Visible = False
```

```
    Picture1.Visible = False
```

```
    Picture1.Enabled = True
```

```
    Call Page4
```

```
    UserControl.Height = Picture2.Height
```

```
    UserControl.Width = Picture2.Width
```

```
    Screen.MousePointer = vbHourglass
```

```
End Sub
```

```
Private Sub Command9_Click()
```

```
    Picture7.Visible = False
```

```
    Picture2.Enabled = True
```

```
End Sub
```

```
Private Sub hsc1CurrRow_Change()
```

```
    If Change Then Winsock1.SendData "D" & hsc1CurrRow.Value & "{"
```

```
    Change = True
```

```
End Sub
```

```
Private Sub List2_Click()
```

```
    Command2.Enabled = True
```

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub List2_DblClick()
    Winsock1.SendData "B" & List2.ListIndex & "->" & Mid(List2.Text, 6) & "{";"
End Sub
```

```
Private Sub List3_Click()
Dim I As Integer
    Command2.Enabled = False
    For I = 0 To List3.ListCount - 1
        If List3.Selected(I) Then Command2.Enabled = True
    Next
End Sub
```

```
Private Sub List4_Click()
Dim I As Integer
    Command2.Enabled = False
    For I = 0 To List4.ListCount - 1
        If List4.Selected(I) Then Command2.Enabled = True
    Next
End Sub
```

```
Private Sub List5_Click()
    Command2.Enabled = True
End Sub
```

```
Private Sub List6_Click()
    Command2.Enabled = True
End Sub
```

```
Private Sub List6_DblClick()
    Winsock1.SendData "E" & List6.ListIndex & "{";"
End Sub
```

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการสงวนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Winsock1.SendData "G" & Text1.Text & "}" & List5.Text & "{"
Text1.Text = ""
Picture6.Visible = False
Screen.MousePointer = vbHourglass
Picture1.Enabled = True
UserControl.Width = 3975

```

```
End If
```

```
End Sub
```

```
Private Sub txtFieldData_KeyPress(Index As Integer, KeyAscii As Integer)
```

```
    KeyAscii = 0
```

```
End Sub
```

```
Private Sub UserControl_Initialize()
```

```
    Call AllPicVisFalse
```

```
End Sub
```

```
Private Sub vsbScrollBar_Change()
```

```
    Dim nTop As Integer
```

```
    picFields.Top = -(vsbScrollBar.Value * 340)
```

```
    
```

```
    
```

```
End Sub
```

```
    
```

```
    
```

```
Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
```

```
    
```

```
    Dim strData, ChkData, SubData As String
```

```
    
```

```
    Dim DataWidth
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรรมใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dim I, J, L As Integer

□

```

Screen.MousePointer = vbDefault
Winsock1.GetData strData, vbString
While InStr(strData, "{;") <> 0
    I = InStr(strData, "{;")
    ChkData = Left(strData, I - 1)
    strData = Mid(strData, I + 2)
    Select Case Left(ChkData, 1)
        Case "A":
            If Mid(ChkData, 2) = "Clear" Then
                Picture1.Visible = True
                UserControl.Width = 3645
                UserControl.Height = 2340
                Call AllListVisFalse
                List2.Clear
                Command1.Visible = False
                Command1.Enabled = True
                Command2.Enabled = False
                Label1.Caption = "Please Select Table"
                List2.Visible = True
                Command2.Visible = True
            Else
                If Mid(ChkData, 2) = "ERROR" Then
                    Label7.Caption = "Wrong Data Enter Again"
                    Text2.Text = ""
                    Text3.Text = ""
                    Text4.Text = ""
                    Picture8.Visible = True
                Else
                    List2.AddItem Mid(ChkData, 2)
                End If
            End If
        End Select
    End While

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆก็ตาม อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
If Mid(ChkData, 2) = "Clear" Then
```

```
List3.Clear
```

```
List5.Clear
```

```
Command1.Visible = True
```

```
Command2.Enabled = False
```

```
Label1.Caption = "Please Select Field"
```

```
Call Page3
```

```
List3.Visible = True
```

```
Else
```

```
List3.AddItem Mid(ChkData, 2)
```

```
List5.AddItem Mid(ChkData, 2)
```

```
End If
```

```
Case "C":
```

```
Call SetDisplay(Mid(ChkData, 2))
```

```
Call Page4
```

```
Case "D":
```

```
If Mid(ChkData, 2) = "Can't find" Then
```

```
Picture2.Enabled = False
```

```
Picture7.Visible = True
```

```
Else
```

```
If Mid(ChkData, 2) = "ERROR" Then
```

```
For I = 0 To txtFieldData.Count - 1
```

```
txtFieldData(I).Text = ""
```

```
Next
```

```
Picture4.Visible = False
```

```
Else
```

```
Call DisplayRecord(Mid(ChkData, 2))
```

```
If ShowPic4 Then
```

```
Winsock1.SendData "I{;"
```

```
Screen.MousePointer = vbHourglass
```

```
If (Grid1.ScrollBars = 3) Or (Grid1.ScrollBars = 2) Then Grid1.Width = Grid1.Width -
```

```

Picture4.Visible = False

ShowPic4 = False

Grid1.Rows = 1

AllHeight = 1

End If

End If

End If

Case "E":

If Mid(ChkData, 2) = "Clear" Then

List4.Clear

Command2.Enabled = False

Call Page5

Else

List4.AddItem Mid(ChkData, 2)

End If

Case "F":

Call SetGrid1(Mid(ChkData, 2))

Grid1.Rows = 1

AllHeight = 1

Case "G":

If Mid(ChkData, 2) = "ALL-L" Then

Call SetGridDisplay(AllHeight)

Grid1.FixedRows = 1

Call Page6

Else

If Mid(ChkData, 2) = "ALL-0" Then

Call Page5

Picture4.Visible = False

ShowPic4 = False

UserControl.Width = Picture2.Width

Picture7.Visible = True

Picture1.Visible = False

Picture2.Enabled = False

Picture1.Top = 0

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งนี้หากมีข้อผิดพลาดหรือต้องการแจ้งถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Picture1.Left = 0

Else

    Call SetGrid2(Mid(ChkData, 2))

    AllHeight = AllHeight + 1

End If

End If

Case "H":

    If Mid(ChkData, 2) = "Clear" Then

        List6.Clear

        Command2.Enabled = False

        Call Page5

        Label1.Caption = "Please Select Table"

        List4.Visible = False

        List6.Visible = True

    Else

        List6.AddItem Mid(ChkData, 2)

    End If

Case "Y":

    Screen.MousePointer = vbHourglass

    Winsock1.SendData "A" & Text2.Text & ";" & Text3.Text & ";" & Text4.Text & ";"

End Select

Wend

End Sub

```

```

Private Sub Page3()

    Label1.Caption = "Please Select Field"

    Call AllListVisFalse

    Call AllPicVisFalse

    Picture1.Visible = True

    Command2.Visible = True

    UserControl.Width = 3645

    UserControl.Height = 2340

```

เอกสาร `RaiseEvent Resize` นี้ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ `End Sub` ใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Private Sub Page4()

Label2.Caption = "Table " & List2.Text

Picture1.Visible = False

Picture2.Visible = True

Command2.Visible = False

Picture2.Enabled = True

End Sub

Private Sub Page5()

Label1.Caption = "Please Select Field"

Call AllListVisFalse

List4.Visible = True

Picture2.Visible = True

Picture2.Enabled = False

Picture1.Visible = True

Command2.Visible = True

UserControl.Height = Picture2.Height

If UserControl.Height < (Picture1.Height + 250) Then

UserControl.Height = Picture1.Height + 250

End If

End Sub

Private Sub Page6()

Picture1.Visible = False

Picture2.Visible = True

Picture2.Enabled = True

Picture4.Visible = True

ShowPic4 = True

Picture2.Height = Picture2.Height

Picture4.Top = Picture2.Height

If ((lblFieldName.Count = 3) Or (lblFieldName.Count = 6) Or (lblFieldName.Count = 9)) And

(AllHeight = 2) Then

UserControl.Height = Picture2.Height + Picture4.Height + 15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น ยกเว้น หมายเหตุและเงื่อนไข และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Else
    UserControl.Height = Picture2.Height + Picture4.Height
End If
End Sub

```

```

Private Sub BigSize()
    Label1.Caption = "Data Table " & List2.Text
    List2.Visible = False
    Grid1.Visible = True
    Command1.Top = 4080
    Command2.Visible = False
    UserControl.Width = 10000
    UserControl.Height = 4590
    RaiseEvent Resize
End Sub

```

```

Public Property Get Width() As Variant
    Width = UserControl.Width
End Property

```

```

Public Property Let Width(ByVal vNewValue As Variant)
    '
End Property

```

```

Function GetFieldWidth(rType As Integer)
    Select Case rType
        Case dbBoolean
            GetFieldWidth = 850
        Case dbByte
            GetFieldWidth = 650
        Case dbInteger
            GetFieldWidth = 900

```

เอกสารนี้เป็นลิขสิทธิ์ของมหาวิทยาลัยพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาติให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ หากมีให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Case dbCurrency
    GetFieldWidth = 1800
Case dbSingle
    GetFieldWidth = 1800
Case dbDouble
    GetFieldWidth = 2200
Case dbDate
    GetFieldWidth = 2000
Case dbText
    GetFieldWidth = 3250
Case dbLongBinary
    GetFieldWidth = 3250
Case dbMemo
    GetFieldWidth = 3250
Case Else
    GetFieldWidth = 3250
End Select
End Function

```

```
Private Sub SetDisplay(Str As String)
```

```
Dim I, J, L As Integer
```

```
Dim nFieldType As Integer
```

```
Dim Temp
```

```
picFields.Visible = True
```

```
J = InStr(Str, "+-")
```

```
L = InStr(J + 1, Str, "+-")
```

```
lblFieldName(0).Caption = Left(Str, J - 1)
```

```
nFieldType = Mid(Str, J + 2, L - J - 2)
```

```
J = InStr(Str, ";")
```

```
L = Mid(Str, L + 2, J - L - 2)
```

```
txtFieldData(0).Width = GetFieldWidth(nFieldType)
```

```
If nFieldType = dbText Then txtFieldData(0).MaxLength = L
```

```

lblFieldName(0).Visible = True
txtFieldData(0).Visible = True
Str = Mid(Str, J + 2)
I = 1
While InStr(Str, ";;")
    J = InStr(Str, "+-")
    L = InStr(J + 1, Str, "+-")
    picFields.Height = picFields.Height + 340
    Load lblFieldName(I)
    lblFieldName(I).Top = lblFieldName(I - 1).Top + 340
    Load txtFieldData(I)
    txtFieldData(I).Top = txtFieldData(I - 1).Top + 340
    lblFieldName(I).Caption = Left(Str, J - 1)
    nFieldType = Mid(Str, J + 2, L - J - 2)
    txtFieldData(I).Width = GetFieldWidth(nFieldType)
    J = InStr(Str, ";;")
    L = Mid(Str, L + 2, J - L - 2)
    If nFieldType = dbText Then txtFieldData(I).MaxLength = L
    lblFieldName(I).Visible = True
    txtFieldData(I).Visible = True
    Str = Mid(Str, J + 2)
    I = I + 1
Wend
FieldCount = I - 1
J = InStr(Str, "+-")
RecordCount = Left(Str, J - 1)
RelationOfTable = Mid(Str, J + 2)
If RelationOfTable < 1 Then
    Command4.Visible = False
Else: Command4.Visible = True
End If
Call SetScrollBar(RecordCount)
picFields.Top = 0
hsc1CurrRow.Visible = True

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าในรูปแบบใด ๆ ทั้งสิ้น ยกเว้นให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

picMoveButtons.Visible = True
If I <= 10 Then
    Picture2.Height = ((I + 1) * 340) + 1500
    Picture3.Height = picFields.Height
Else
    Picture3.Height = 3435
    Picture2.Height = 5200
    Picture2.Width = Picture2.Width + 260
    vsbScrollBar.Visible = True
    vsbScrollBar.SmallChange = 1
    vsbScrollBar.LargeChange = ((I - 10) / 20) + 1
    vsbScrollBar.Min = 0
    vsbScrollBar.Max = I - 10
    vsbScrollBar.Value = 0

```

```
End If
```

```

picMoveButtons.Top = Picture3.Top + Picture3.Height + 250
Command3.Top = picMoveButtons.Top + 400
Command4.Top = picMoveButtons.Top + 400
Command5.Top = picMoveButtons.Top + 400
UserControl.Height = Picture2.Height
UserControl.Width = Picture2.Width

```

```
End Sub
```

```
Private Sub SetScrollBar(mI numRows As Long)
```

```
On Error Resume Next
```

```
If numRows < 2 Then
```

```
hscI CurrRow.Max = 100
```

```
hscI CurrRow.SmallChange = 1 '00
```

```
hscI CurrRow.LargeChange = 100
```

```
ElseIf numRows > 32767 Then
```

```
hscI CurrRow.Max = 32767
```

```
hscI CurrRow.SmallChange = 1
```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น ยกเว้นให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

hscCurrRow.LargeChange = 1000
ElseIf mI numRows > 99 Then
    hscCurrRow.Max = mI numRows
    hscCurrRow.SmallChange = 1
    hscCurrRow.LargeChange = mI numRows \ 20
Else
    hscCurrRow.Max = mI numRows
    hscCurrRow.SmallChange = 1
    hscCurrRow.LargeChange = (mI numRows \ 20) + 1

```

```
End If
```

```
txtFieldData(0).SetFocus
```

```
hscCurrRow.SetFocus
```

```
End Sub
```

```
Private Sub DisplayRecord(Str As String)
```

```
Dim I, J, K As Integer
```

```
Dim Value
```

```
I = InStr(Str, "+-")
```

```
Value = Mid(Str, I + 2)
```

```
UserControl.lblStatus.Caption = Value & "/" & RecordCount
```

```
Change = False
```

```
hscCurrRow.Value = Value
```

```
Change = True
```

```
Str = Left(Str, I - 1)
```

```
J = 0
```

```
While InStr(Str, ";");
```

```
    I = InStr(Str, ";");
```

```
    txtFieldData(J).Text = Left(Str, I - 1)
```

```
    Str = Mid(Str, I + 2)
```

```
    J = J + 1
```

```
Wend
```

เอกสารนี้ให้ txtFieldData(J).Text = Str สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End Sub

Private Sub SetGrid1(ByVal Str As String)

Dim I, J, nFieldType As Integer

Dim AllWidth As Long

Dim Data As String

I = InStr(Str, ";");

Label6.Caption = "Table " & Left(Str, I - 1)

Label6.Width = (I + 6) * 75

Str = Mid(Str, I + 2)

I = InStr(Str, ";");

Grid1.Rows = 1

Grid1.Cols = 0

AllWidth = 0

While I <> 0

Data = Left(Str, I - 1)

Str = Mid(Str, I + 2)

Grid1.Cols = Grid1.Cols + 1

Grid1.Row = 0

Grid1.Col = Grid1.Cols - 1

J = InStr(Data, "+-")

Grid1.Text = Left(Data, J - 1)

nFieldType = Mid(Data, J + 2)

Grid1.ColWidth(Grid1.Cols - 1) = GetFieldWidth(nFieldType)

AllWidth = Grid1.ColWidth(Grid1.Cols - 1) + AllWidth

I = InStr(Str, ";");

Wend

If AllWidth < 10000 Then

Grid1.Width = AllWidth + 80

เอกสารนี้ Grid1.ScrollBars = flexScrollBarNone เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่า Else: Grid1.Width = 10000 ให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Grid1.ScrollBars = flexScrollBarHorizontal
End If

If Grid1.Width > Label6.Width Then
    If Grid1.Width > Command6.Width Then
        Picture4.Width = Grid1.Width + 300
    Else
        Picture4.Width = Command6.Width + 300
    End If
Else
    If Label6.Width > Command6.Width Then
        Picture4.Width = Label6.Width + 300
    Else
        Picture4.Width = Command6.Width + 300
    End If
End If

If UserControl.Width < (Grid1.Width + 300) Then UserControl.Width = Grid1.Width + 300
RaiseEvent Resize
End Sub

Private Sub SetGrid2(ByVal ChkData As String)
Dim SubData As String
Dim L, J As Integer
Grid1.Rows = Grid1.Rows + 1
L = 0
Grid1.Row = Grid1.Rows - 1
While InStr(ChkData, ";") <> 0
    Grid1.Col = L
    J = InStr(ChkData, ";")
    SubData = Left(ChkData, J - 1)
    ChkData = Mid(ChkData, J + 2)
    Grid1.Text = SubData
    L = Grid1.Col + 1
Wend

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Grid1.Col = L
```

```
Grid1.Text = ChkData
```

```
End Sub
```

```
Private Sub SetGridDisplay(ByVal AllHeight As Long)
```

```
    If (AllHeight * 260) < 3000 Then
```

```
        If Grid1.ScrollBars = flexScrollBarNone Then
```

```
            Grid1.Height = (AllHeight * 260)
```

```
        Else
```

```
            Grid1.Height = (AllHeight * 260) + 250
```

```
        End If
```

```
    Else
```

```
        Grid1.Height = 3000
```

```
        If Grid1.ScrollBars = flexScrollBarNone Then
```

```
            Grid1.ScrollBars = flexScrollBarVertical
```

```
        Else
```

```
            If Grid1.ScrollBars = flexScrollBarHorizontal Then
```

```
                Grid1.ScrollBars = flexScrollBarBoth
```

```
            End If
```

```
        End If
```

```
        Grid1.Width = Grid1.Width + 260
```

```
        Picture4.Width = Grid1.Width + 300
```

```
    End If
```

```
    If UserControl.Width < (Grid1.Width + 300) Then UserControl.Width = Grid1.Width + 300
```

```
    Grid1.Visible = True
```

```
    Picture4.Visible = True
```

```
    ShowPic4 = True
```

```
    Picture4.Height = Grid1.Height + 1380
```

```
    Command6.Top = Grid1.Top + Grid1.Height + 250
```

```
End Sub
```

```
Host = Hostname
```

```
End Property
```

```
Public Property Let Host(ByVal vNewValue As Variant)
```

```
    Hostname = vNewValue
```

```
End Property
```

```
Private Sub AllPicVisFalse()
```

```
    Picture1.Visible = False
```

```
    Picture2.Visible = False
```

```
    Picture4.Visible = False
```

```
    Picture5.Visible = False
```

```
    Picture6.Visible = False
```

```
End Sub
```

```
Private Sub AllListVisFalse()
```

```
List2.Visible = False
```

```
List3.Visible = False
```

```
List4.Visible = False
```

```
List5.Visible = False
```

```
List6.Visible = False
```

```
End Sub
```

```
Private Sub MovePic1R()
```

```
    Picture1.Top = 250
```

```
    Picture1.Left = 250
```

```
    Picture1.BorderStyle = 1
```

```
End Sub
```

```
Private Sub MovePic1L()
```

```
    Picture1.Top = 0
```

```
    Picture1.Left = 0
```

```
    Picture1.BorderStyle = 0
```

```
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. Noel Jerke and George Szabo and David Jung and Don Kiely, “Visual Basic 5 Client/Server How To”, Waite Group Press TM
2. Sanders Kaufman Jr. and Jeff Perkins and Dina Fleet, “Teach Yourself ActiveX Programming in 21 day”, Sam Net Publishing
3. ET AL. Eric Smith, “Inside VBScript with ActiveX”, New Riders Publishing
4. Advance Information System Inc., “Oracle Unleash”, SAMS Publishing



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้