



การโปรแกรมระบบวินโดว์ โดยใช้ ไมโครซอฟท์ Visual C++
WINDOWS PROGRAMING USING MICROSOFT VISUAL C++



โดย
นาย วุฒินันท์ ยอดนิล
นาย สมบัติ เทียมตามณี

วัน เดือน ปี... 1 ตุลาคม 2539
เลขทะเบียน..... 038296
เลขเรียกหนังสือ... T.343.16.08210

ปริญญาบัตรนี้เป็นส่วนหนึ่งการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2539

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไปว่ากรณีใดก็ตาม ซึ่งสิ่งนี้คือทั้งห้ามมิให้คัดลอกไปเอง และต้องแจ้งถึงเจ้าของเอกสารฉบับนี้ที่มีภาระนำไปใช้

การโปรแกรมระบบวินโดว์ โดยใช้ ไมโครซอฟท์ Visual C++
WINDOWS PROGRAMING USING MICROSOFT VISUAL C++



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2539

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การโปรแกรมระบบวินโดวส์ โดยใช้ ไมโครซอฟ Visual C++

WINDOWS PROGRAMING USING MICROSOFT VISUAL C++

ผู้จัดทำ

นาย วุฒินันท์ ยอดนิล

37013312

นาย สมบัติ เทียมตามณี

37013315



บุญธีร์ เครือตราชู
.....
(ดร. บุญธีร์ เครือตราชู)

อาจารย์ที่ปรึกษา

กฤตวัน เครือตราชู
.....
(อ. กฤตวัน เครือตราชู)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อ

Adstract

บทที่ 1 บทนำ

- 1.1 โครงข่ายประสาทเทียม 1
- 1.2 วิธีการสร้างเซลล์ของโครงข่ายประสาทเทียม 2
- 1.3 การนำเสนอโครงงาน 2

บทที่ 2 ไมโครซอฟท์วินโดวส์ และ Visual C++

- 2.1 รูปแบบการเขียนโปรแกรมบนวินโดวส์ 4
 - 2.1.1 Win32 กับ Win16 4
 - 2.1.2 การจัดการเมสเสจ 4
 - 2.1.3 Windows Graphics Device Interface (GDI) 5
 - 2.1.4 การโปรแกรมตามประเภทของรีซอร์ส 5
 - 2.1.5 การจัดหน่วยความจำ 6
 - 2.1.6 Dynamics Link Libraries (DLLs) 6
 - 2.1.7 Win32 Application Programming Interface (API) 7
- 2.2 ส่วนประกอบของ Visual C++ 7
 - 2.2.1 กระบวนการในการสร้างโปรแกรม 8
 - 2.2.2 Graphics Editor 9
 - 2.2.3 Compiler C /C++ 10
 - 2.2.4 รีซอสคอมไพล์ 10
 - 2.2.5 ลิงก์เกอร์ 10
 - 2.2.6 ดีบั๊กเกอร์ 11
 - 2.2.7 Appwizard 12
 - 2.2.8 ClassWizard 12
 - 2.2.9 Source Browser 13
 - 2.2.10 Online Help 14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

(ต่อ)	หน้า
2.2.11 Windows Diagnostic Tools	14
2.2.12 Source Code Control	15
2.2.13 Component Gallery	15
2.2.14 Microsoft Foundation Class Library Version 4.0	16
บทที่ 3 การโปรแกรมด้วยไลบรารี MFC บน Windows	
3.1 คลาสไลบรารี MFC เพื่อการพัฒนา Windows	19
3.1.1 ฟังก์ชันสมาชิก เทียบกับ ฟังก์ชัน Windows API	20
3.1.2 ฟังก์ชันสมาชิก	21
3.2 ส่วนสำคัญพื้นฐานของ MFC	21
3.2.1 CWinApp	22
3.2.2 CFrameWnd, CMDIFrameWnd และ CMDIChildWnd	22
3.2.3 CDialog และ CModalDialog	23
3.2.4 การจัดการเมสเสจ	23
3.2.5 โครงร่างเมสเสจ (Message Maps)	24
3.2.6 มากกว่ากับคลาส CWinApp	25
3.3 คลาส Cwnd	26
3.4 โครงร่างเมสเสจต่างๆ	28
3.4.1 ทางเข้าสำหรับเมสเสจค่าคงที่	29
3.4.2 ทางเข้าสำหรับเมสเสจ WM_COMMAND	29
3.4.3 ทางเข้าสำหรับเมสเสจควบคุม	30
3.4.4 ทางเข้าสำหรับเมสเสจ Windows ซึ่งจดทะเบียนไว้	31
3.4.5 สำหรับเมสเสจ Windows มาตรฐาน	32
3.5 ระเบียบแบบแผนของการตั้งชื่อ MFC	32
3.5.1 สัญลักษณ์แบบฮังกาเรียน	33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

(ต่อ)	หน้า
บทที่ 4 เมนู และไดอะล็อกบ็อกซ์บน Windows ด้วยไลบรารี MFC	
4.1 คลาส CMENU	36
4.2 ฟังก์ชันเมนูของคลาส Cwnd	36
4.3 คลาสไดอะล็อกของ MFC	37
4.4 คลาส CModalDialog	40
บทที่ 5 โครงข่ายประสาทเทียมเบื้องต้น	
5.1 ความรู้พื้นฐานทางกายภาพ ของระบบเซลล์ประสาท	42
5.2 หน้าที่แต่ละส่วนของระบบประสาท	42
5.3 คุณสมบัติ และความสามารถที่เป็นประโยชน์ ของโครงข่ายประสาทเทียม	46
5.4 การจัดประเภทของโครงข่ายประสาทเทียม	47
5.4.1 แบ่งตามลักษณะการใช้งาน	47
5.4.2 แบ่งตามลักษณะวิธีการเรียนรู้	47
5.5 แนวคิดเบื้องต้น	47
5.6 แบบจำลองทางคณิตศาสตร์ของหนึ่งหน่วยเซลล์ประสาท	50
บทที่ 6 การเกิดของโครงสร้างเซลล์	
6.1 ความรู้เบื้องต้น (Introduction)	51
6.2 การเกิดของเซลล์แบบไม่มีข้อมูลชี้นำ หรือไม่มีการดูแล	52
6.2.1 ขอบเขตของปัญหา	52
6.2.2 สถาปัตยกรรมของโครงข่าย	53
6.2.3 โครงข่ายแบบไดนามิกส์	55
6.4 การนำเซลล์ออก	60
6.5 การประมาณค่า Voronoi Region	61
6.6 ประสิทธิภาพของการต่อแบบหลายมิติ	62
6.7 วิธีที่มองโครงข่ายที่หลายมิติ	63
6.7.1 วิธีการสร้าง	63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

(ต่อ)	หน้า
บทที่ 7 การสร้าง และการคำนวณ	
7.1 เป้าหมาย	68
7.2 กำหนดคุณสมบัติของโปรแกรม	68
7.3 โฟลว์ชาร์ตแสดงการทำงานของโปรแกรม	70
7.4 ความต้องการของโปรแกรมเบื้องต้น	71
7.5 ขั้นตอนการสร้าง	71
7.6 การเขียนซอร์สโค้ดควบคุมการทำงานต่าง ๆ ของแต่ละฟังก์ชัน	81
บทที่ 8 การทดลอง และผลการทดลอง	82
บทที่ 9 บทวิจารณ์ และบทสรุป	83
ภาคผนวก	
ซอร์สโค้ดทั้งหมดของโปรแกรม	84

สารบัญรูปภาพ

หน้า

รูปที่ 2.1 แสดงขบวนการสร้างแอปพลิเคชัน	9
รูปที่ 2.2 แสดงการใช้ Graphics Editor	10
รูปที่ 2.3 แสดงการตั้งค่าการ Link	11
รูปที่ 2.4 แสดงเมนู ของ Debug	12
รูปที่ 2.5 แสดงการใช้ ClassWizard	13
รูปที่ 2.6 แสดงการใช้ Source Browser	14
รูปที่ 2.7 แสดง Component Gally	16
รูปที่ 3.1 แสดงคลาสพื้นฐานของ MFC	20
รูปที่ 4.1 แสดงคลาสควบคุม และไดอะล็อกบ็อกซ์ของ MFC	39
รูปที่ 5.1 แสดงการสื่อสารข้อมูลในระบบสมองมนุษย์	42
รูปที่ 5.2 แสดงโครงสร้าง และการส่งกระแสของเซลล์ประสาท	43
รูปที่ 5.3 แสดงโครงสร้าง และส่วนประกอบของเซลล์ประสาท	43
รูปที่ 5.4 แสดงลักษณะการเกิดศักย์ไฟฟ้าขณะทำงาน	44
รูปที่ 5.5 แสดงศักย์ไฟฟ้าขณะทำงานที่เกิดจากการรวมกันของซีแนปส์	45
รูปที่ 5.6 แบบจำลองทางคณิตศาสตร์ ของเซลล์ประสาท	50
รูปที่ 6.1 แสดงโครงสร้างเซลล์ที่มีมิติที่ต่างกัน	53
รูปที่ 6.2 แสดงวิธีของ Voronoi tessellation	54
รูปที่ 6.3 แสดงการปรับตัวของโครงข่ายแบบ 2 มิติ	56
รูปที่ 6.4 แสดงการเพิ่มของเซลล์	57
รูปที่ 6.5 แสดงอัลกอริทึมเบื้องต้นของการกำหนดเซลล์	58
รูปที่ 6.6 แสดงการสร้างเซลล์แบบ 2 มิติ	59
รูปที่ 6.7 พัฒนาการของโครงข่ายแบบ 2 มิติ	60
รูปที่ 6.8 แสดงเซลล์ขนาด 400 เซลล์ที่มีการปรับตัว	61
รูปที่ 6.9 แสดงการนำเซลล์ออกจากโครงข่าย	62
รูปที่ 6.10 แสดงหลังจากที่นำเซลล์ออกจากโครงข่าย	63
รูปที่ 6.11 แสดงการใช้วิธีของ kohonen feature map	66

เอกสารนี้เป็นเอกสารทสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

(ต่อ)

หน้า

รูปที่ 6.12 แสดงหลังจากอัลกอริทึมใหม่สร้าง	67
รูปที่ 7.1 แสดง Dialog ที่รับข้อมูลทางหน้าจอ	69
รูปที่ 7.2 แสดง Dialog แสดงข้อมูลเพิ่มเติม	69
รูปที่ 7.3 แสดงไฟล์ชาร์ตของโปรแกรม	70
รูปที่ 7.4 แสดงการเปิด Project workspace	71
รูปที่ 7.5 แสดงการใช้ MFC AppWizard	72
รูปที่ 7.6 แสดงการเลือกประเภท Document	73
รูปที่ 7.7 แสดงการเลือกให้แอปพลิเคชัน Support DataBase	74
รูปที่ 7.8 แสดงการเลือก เมนู ทูลบาร์ และสเตตัสบาร์	75
รูปที่ 7.9 แสดงการเลือกการ Comment	76
รูปที่ 7.10 แสดงการเลือกประเภทของ View	77
รูปที่ 7.11 แสดงผลการเลือกทั้งหมดที่ AppWizard จะสร้างให้	78
รูปที่ 7.12 แสดงการเพิ่ม Dialog	79
รูปที่ 7.13 แสดงการ Edit Dialog	79
รูปที่ 7.14 แสดงการ Edit Icon	80
รูปที่ 7.15 แสดงการใช้ ClassWizard สร้างฟังก์ชันการทำงาน	80
รูปที่ 8.1 แสดงผลการนำเสนอข้อมูล	82

สารบัญตาราง

หน้า

ตารางที่ 3.1 แสดงการกำหนดตัวขึ้นต้นด้วยสัญลักษณ์แบบอังกฤษเรียน	33
ตารางที่ 5.1 แสดงค่าตรรก "และ"	47
ตารางที่ 6.1 แสดงอินพุท	65



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญญากาศ

หน้า

รูปกราฟที่ 5.1 แสดงความสัมพันธ์เชิงเรขาคณิตวิเคราะห์ของตรรก "และ"	48
รูปกราฟที่ 5.2 แสดงเส้นตรงแบ่งพื้นที่แก้ปัญหของตรรก "และ"	48
รูปกราฟที่ 5.3 แสดงเส้นตรงแบ่งพื้นที่แก้ปัญหของตรรก "หรือ"	49



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การโปรแกรมระบบวินโดวส์โดยใช้ไมโครซอฟวิซวล C++

วุฒินันท์ ยอดนิล

สมบัติ เทียมตามณี

ดร. บุญธีร์ เครือตราฐ อาจารย์ที่ปรึกษา

อ. กฤตวัน เครือตราฐ อาจารย์ที่ปรึกษาร่วม

ปีการศึกษา 2539

บทคัดย่อ

ในปฏิญานีพนธ์ฉบับนี้ ได้ทำการศึกษาโปรแกรมไมโครซอฟวิซวล C++ และได้นำข้อดีหรือคุณสมบัติต่าง ๆ ของโปรแกรมแบบ อ็อบเจกโอเรียนเต็ด มาเป็นเครื่องมือในการสร้าง ซอฟแวร์เพ็จเก็จ หลังจากได้ศึกษาแล้วจึง พัฒนาเครื่องมือสำหรับใช้สร้างเซลล์ในระบบโครงข่ายประสาทเทียม โดยแต่เดิมนั้นมีการใช้วิธีการแม็ปปิ้ง ด้วยวิธีการของ Kohonen's feature map ซึ่งการแม็ปดังกล่าวยังไม่สามารถให้รายละเอียดไม่มากพอ หรือการแยกแยะข้อมูลยังไม่ละเอียดปฏิญานีพนธ์ฉบับนี้ ได้นำเสนอข้อมูลจากการที่ได้แม็ปอีกวิธีการหนึ่งซึ่งแสดงให้เห็นว่าสามารถแยกประเภทของข้อมูลที่ดีกว่า มาแนะนำเสนอซึ่งการนำเสนอข้อมูลนั้น โดยอาศัยหลักการทางฟิสิกส์พื้นฐาน มาเป็นเงื่อนไขในการสร้าง วิธีการแม็ปดังกล่าวซึ่งเป็นการนำโครงข่ายประสาทเทียมที่มีมากกว่า 2 มิติมาแสดงให้เห็นอยู่ในรูปของ 2 มิติ

Windows Programming Using Microsoft Visual C++

Wutthinan Yodnin

Sombat Tientamane

Dr. BoonTee Kroutrachu Advisor

Kittawan Kroutrachu Associate Advisor

1996

Abstract

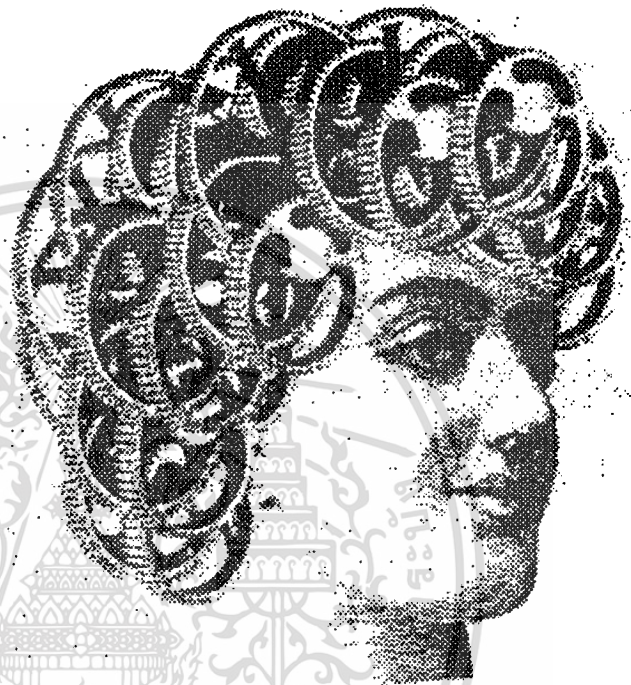
In this thesis ,We have educated about properties and advantages of Microsoft Visual C++ Program that can be the tool for softwate package. Maeaer , We have developed software tool for neural network cell structure . This approach use phisically modal to be the condition for mapping. It can show many dimentions (more than 2 dementions) neural cell structure for 2 dimentsions and can discriminate information type batter than exiting one, Kohonen's feature mapping in tis result has been demenstrated.

บทที่ 1

บทนำ

1.1 โครงข่ายประสาทเทียม

นานมาแล้วที่มนุษย์ได้ประดิษฐ์ และพัฒนาเครื่องจักรขึ้นมาทำงานแทนมนุษย์ และงานที่ให้เครื่องจักรทำส่วนใหญ่เป็นงานที่มีความยุ่งยากซับซ้อน และเป็นงานที่เกิดข้อบ่อน้อย ซึ่งถ้าหากใช้มนุษย์นั้น ก็เกิดความเสียหายได้งานกว่า และต่อ ๆ มาได้มีการพัฒนาสิ่งประดิษฐ์ต่าง ๆ มากมาย หลายอย่าง ปัจจุบันวิศวกร และนักวิทยาศาสตร์แขนงต่าง ๆ ก็ได้พยายามที่จะพัฒนาให้



ดียิ่ง ๆ ขึ้นไป เช่นเดียวกัน เครื่องคอมพิวเตอร์ ก็ได้รับการพัฒนาอย่างไม่หยุดนิ่ง รวมทั้งทำให้เครื่องคอมพิวเตอร์มีความฉลาดยิ่งขึ้น เริ่มจากที่จะให้เครื่องคอมพิวเตอร์มีการเรียนรู้ และสามารถตัดสินใจได้เอง หรือใช้เป็นส่วนช่วยในการตัดสินใจ

ระบบแบบจำลองโครงข่ายประสาท หรือระบบแบบจำลองนิวรอนเน็ตเวก (Artificial Neural Network System : ANNS) เป็นอีกตัวอย่างหนึ่งที่ทำให้เครื่องคอมพิวเตอร์มีศักยภาพเพิ่มมากขึ้น

เมื่อพิจารณาระหว่างมนุษย์ และสัตว์จะเห็นว่าเราสามารถมองจากภาพ และทำการวิเคราะห์ และแยกแยะว่า มนุษย์สามารถแยกแยะเสียงได้ดีกว่า ในที่นี้คอมพิวเตอร์จะถูกสร้างมาเพื่อการนี้โดยเฉพาะ ซึ่งคาดว่าระบบ ANNS เป็นตัวที่จะก่อให้เกิดความเปลี่ยนแปลงในการประมวลผล ข่าวสาร สำหรับใช้แทน หรือสนับสนุนการคำนวณแบบเดิม ระบบ ANNS จะช่วยเสริมให้ ความสามารถของ เครื่องคอมพิวเตอร์ สำหรับการประมวลผลในระบบใหญ่ ๆ ซึ่งระบบนี้สามารถเรียนรู้ กระบวนการทำงาน และการตัดสินใจ สำหรับกระบวนการของระบบนั้นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงข่ายประสาทเทียมนั้นได้ศึกษามาจากเซลล์ประสาท หรือสมองของมนุษย์ การศึกษาเซลล์ประสาทก็เพื่อที่จะนำผลที่ได้มาเป็นพื้นฐานในการจำลองการทำงานด้วยเครื่องคอมพิวเตอร์ หน้าที่หลักของเซลล์ประสาท คือ การคำนวณ และการส่งผลลัพธ์ ต่อไปยังเซลล์อื่น ๆ โดยอาศัยหลัก การทางไฟฟ้า

สำหรับการทำงานของระบบ ANNS จะถอดแบบมาจากการทำงานของระบบสมองมนุษย์ คือ จะมีการส่งผ่านข้อมูล กันระหว่างโหนด และโหนดแต่ละโหนด มีการเชื่อมเข้าหากัน ก่อให้เกิดเป็นโครงข่ายร่างแหอย่างหนาแน่น และมีการทำงานในลักษณะขนาน ซึ่งในหัวข้อถัดไปจะอธิบายการทำงานของระบบเซลล์ ประสาทของมนุษย์

โดยปกติการแก้ปัญหาต่าง ๆ ด้วยโปรแกรมคอมพิวเตอร์นั้นจะเริ่มต้นที่การพิจารณา อินพุต แล้วออกแบบขั้นตอนการที่จะดำเนินการกับอินพุตเหล่านั้นเพื่อจะได้สร้างเอาต์พุตได้ตามที่ ต้องการ วิธีการพัฒนาโปรแกรมคอมพิวเตอร์แบบคลาสสิกนี้มีข้อจำกัด คือ ไม่สามารถใช้ได้กับ ปัญหาซึ่งขั้นตอนการ (Process) มีความซับซ้อนอย่างมากจนเราไม่สามารถสรุปเป็นขั้นตอนที่ ชัดเจนได้ โครงข่ายประสาทเทียมเป็นรูปแบบในการแก้ไขปัญหาลักษณะดังกล่าว โครงข่าย ประสาทเทียมจะสร้างขั้นตอนการ (Process) ขึ้นมาเองจากความรู้ ซึ่งได้มาจากการเรียนรู้จาก ตัวอย่าง (Learning by Examples) ทั้งนี้ตัวอย่างที่นำมาสอนให้กับโครงข่ายประสาทเทียมมี จำนวนมากมายพอ และวิธีการสอนดีพอขั้นตอนการที่ถูกสร้างขึ้นก็จะมีคุณภาพดีพอ เพียงพอ สำหรับการนำไปใช้งานจริงได้

1.2 วิธีการสร้างเซลล์ของโครงข่ายประสาทเทียม

วิธีการสร้างเซลล์ของโครงข่ายมีอยู่หลายวิธี ซึ่งแต่ละวิธีก็มีข้อเสียแตกต่างกันไป แต่ สำหรับการจัดทำในโครงงานนี้เป็นการศึกษาหาวิธีการสร้างเซลล์ของโครงข่าย อันเนื่องจาก Self-Organizing Network แบบเดิมที่มีอยู่ เช่น แบบของ Kohonen feature map นั้นไม่สามารถให้ การแยกแยะข้อมูลไม่ดีเท่ากับวิธีใหม่ แบบ Self-Organizing Network แต่เฉพาะสำหรับโครงข่าย ที่เรียนรู้แบบปรับค่าน้ำหนักโดยที่ไม่มีผู้สอน

1.3 การนำเสนอโครงงาน

การนำเสนองาน หรือการจัดทำโครงงานในครั้งนี้ได้ใช้โปรแกรม ไมโครซอฟท์ วิวอล C++ สำหรับวินโดวส์ เพราะโปรแกรมแบบนี้สร้างมาเป็นแบบโปรแกรมเชิงวัตถุ มีการนำโค้ดกลับมาใช้ใหม่ได้อีก มีระบบการอินเตอร์ต่อผู้ใช้งานแบบกราฟฟิก การสร้างรีซอร์สเป็นแบบ " คุณเห็น เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบไหน แล้วคุณได้อย่างนั้น “ สะดวกในการใช้งานที่ติดต่ออุปกรณ์ที่ใช้งานอยู่ ประกอบกับการทำงานเป็นแบบ 32 บิต ทำให้รวดเร็วเมื่อเรียกใช้งาน คุณสมบัติต่าง ๆ อีกมากมายที่ให้ความสะดวกต่อผู้สร้างโปรแกรม โดยเฉพาะโปรแกรมที่มีขนาดใหญ่ ซึ่งคณะผู้จัดทำได้รวบรวมคุณสมบัติดังกล่าว ในบทที่ 2



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ไมโครซอฟวินโดวส์ และ Visual C++

ในบทนี้เราได้แบ่งออกเป็น 2 ส่วน คือ รูปแบบการเขียนโปรแกรมบนวินโดวส์ ซึ่งส่วนที่เราต้องเข้าใจการทำงานของวินโดวส์ เนื่องจากว่าในการเขียนโปรแกรม ต่าง ๆ ต้องเขียนจะต้องเกี่ยวข้อง และติดต่อกันอยู่เสมอ ส่วนที่สอง เป็นส่วนประกอบของแอปพลิเคชัน ที่เราต้องใช้ทำการพัฒนาโปรแกรม ในส่วนนี้ได้อธิบายหน้าที่หลัก ๆ ของส่วนต่าง ๆ

2.1 รูปแบบการเขียนโปรแกรมบนวินโดวส์

2.1.1 Win32 กับ Win16

Win16 คือ การอ้างถึง 16-bit ที่ใช้สำหรับการอินเตอร์เฟส หรือการเชื่อมต่อระหว่าง แอปพลิเคชัน (API : Application Programming Interface) ที่มีใช้ในไมโครซอฟเวอร์ชัน 3.1 และ วินโดวส์ ฟอร์เวิร์กกรุป (Microsoft Windows For Workgroup)

ส่วน Win32 เป็น 32-bit วินโดวส์แอปพลิเคชัน อินเตอร์เฟส เป็นการอินเตอร์เฟสแบบใหม่ที่มีใช้ ไมโครซอฟวินโดวส์ ' 95 และไมโครซอฟวินโดวส์ เอ็นที

Visual C++ เวอร์ชัน 4.0 หรือสูงกว่า เป็นผลิตภัณฑ์ที่ใช้ Win32 ในการพัฒนาแอปพลิเคชัน

2.1.2 การจัดการเมสเสจ (Message Processing)

การเขียนโปรแกรมประยุกต์บนดอส ด้วยภาษา C สิ่งที่เป็นเพียงสิ่งเดียว คือ ฟังก์ชัน main ระบบปฏิบัติการเรียกฟังก์ชัน main เมื่อมีผู้รับโปรแกรม และจากจุดนั้นเราก็ใช้โครงสร้างการเขียนโปรแกรม ได้ตามที่เราต้องการ ถ้าเราต้องการเขียนโปรแกรมให้รับรู้ การป้อนข้อมูลทางคีย์บอร์ด หรือการใช้บริการต่าง ๆ ที่ระบบปฏิบัติการให้บริการ ก็จะเรียกฟังก์ชัน ที่เหมาะสมกับกับสิ่งที่เราต้องการ เช่น getch หรือ บางทีก็ใช้ 'ไลบรารี' จากวินโดวส์

เมื่อระบบปฏิบัติการเรียกวินโดวส์ เรียกโปรแกรมทำงานจะเรียกฟังก์ชัน Winmain โดยที่ที่ใดที่หนึ่งของโปรแกรมต้องมีฟังก์ชัน Winmain ซึ่งทำงานเฉพาะบางอย่าง แต่งานที่สำคัญที่สุด คือ การสร้าง "วินโดวส์ หลัก " (main Windows) ของแอปพลิเคชัน วินโดวส์หลักจะต้องมีไคด์ของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวเองเพื่อจัดการ กับเมสเสจ ซึ่งวินโดวส์ส่งมาให้ ข้อแตกต่างที่สำคัญระหว่างโปรแกรม ที่เขียนบนวินโดวส์ กับโปรแกรมที่เขียนบนดอส คือ โปรแกรมที่เขียนบนดอส จะเรียกระบบปฏิบัติการเพื่อรับข้อมูลจากผู้ใช้ แต่สำหรับโปรแกรมวินโดวส์ โปรแกรมจัดการรับข้อมูลจากผู้ใช้โดยผ่านทางเมสเสจ ซึ่งมาจากระบบปฏิบัติการ

เมสเสจหลายตัวในวินโดวส์ถูกกำหนดอย่างเข้มงวด และใช้กับทุกแอปพลิเคชันโปรแกรมอย่างเช่น WM_CREATE จะถูกส่งออกไปเมื่อวินโดวส์ถูกสร้าง เมสเสจ WM_LBUTTONDOWN จะถูกส่งเมื่อผู้ใช้กดปุ่มซ้ายของเมาส์ เมสเสจ WM_CHAR ถูกส่งเมื่อผู้ใช้พิมพ์ ตัวอักษร และเมสเสจ WM_CLOSE ถูกส่งเมื่อผู้ใช้ปิดวินโดวส์ เมสเสจอื่น ๆ (เมสเสจคอมมานด์) ถูกส่งไปยังวินโดวส์ ของแอปพลิเคชัน เพื่อตอบสนองกับการเลือกรายการในเมนูของผู้ใช้เมสเสจเหล่านั้น ขึ้นอยู่กับรูปแบบของเมนู นอกจากนี้ผู้ใช้ยังกำหนดเมสเสจอื่น ๆ อีกที่เรียกว่า “ เมสเสจสำหรับผู้ใช้ ” (User Message)

2.1.3 Windows Graphics Device Interface (GDI)

โปรแกรมบนดอสหลายตัวเขียนข้อมูลไปยังหน่วยความจำวิดีโอ และพอร์ท เครื่องพิมพ์โดยตรง ข้อเสียของเทคนิคนี้ก็คือ ความจำเป็นที่ต้องการเตรียมซอฟต์แวร์ ไดรเวอร์ สำหรับการ์ดแสดงผล และเครื่องพิมพ์ทุก ๆ แบบ แต่สำหรับวินโดวส์แล้วได้นำความคิดแบบนามธรรม ที่เรียกว่า Graphics Device Interface (GDI) โดยที่วินโดวส์เป็นผู้เตรียมไดร์ เวอร์สำหรับแสดงผล และสำหรับเครื่องพิมพ์ให้ ดังนั้นโปรแกรมของเราไม่จำเป็นต้องรู้ถึงการ์ดแสดงผล และชนิดของเครื่องพิมพ์ที่ต่ออยู่ แทนที่จะเข้าถึงฮาร์ดแวร์โปรแกรมของเราแค่เรียก ใช้ฟังก์ชัน GDI ที่อ้างถึงโครงสร้างข้อมูลที่เรียกว่า ดีไวร์คอนเท็ก (Device Context) วินโดวส์จับคู่โครงสร้างดีไวร์คอนเท็ก กับอุปกรณ์ทางฟิสิคัล (Physical) และเลือกใช้คำสั่งที่เหมาะสมในการนำข้อมูลเข้า หรือออก การเขียนโปรแกรมโดยใช้ GDI นี้เร็วมากเท่ากับการติดต่อกับหน่วยความจำวิดีโอ และยังช่วยให้แอปพลิเคชันที่ต่างกันแสดงผลร่วมกันได้

2.1.4 การโปรแกรมตามประเภทของรีซอร์ซ (Resource-Based Programming)

ในการเขียนโปรแกรมแบบดาต้าไดรเวน (Data Driven) บนดอสนั้น เราจะต้องสร้างข้อมูลด้วยการกำหนดค่าคงที่ในโปรแกรม หรือเราต้องมีไฟล์ข้อมูลต่างหาก จากโปรแกรมเพื่อใช้ในการอ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ในการเขียนโปรแกรมในวินโดวส์ ข้อมูลของคุณเหล่านั้นจะถูกเก็บไว้ในไฟล์รีซอร์ส (Resource file) ซึ่งมีรูปแบบจำนวนหนึ่ง วินโดวส์รวมไฟล์รีซอร์ส เข้ากับโปรแกรมมิ่งแล้วโดยผ่านขั้นตอนที่เรียกว่า “ไบน์ดิง” (binding) ไฟล์รีซอร์สสามารถเก็บให้อยู่ในรูปแบบ บิตแมป (Bitmaps) ไอคอน (Icon) เมนูดิฟนิชัน (Menu Definition) ไดออล็อกบ็อกซ์ (Dialog Box) และสกรีน นอกจากนี้ยังเก็บในรูปแบบอื่นที่เรากำหนดได้

การใช้เท็กอิดิเตอร์ เพื่อแก้ไขโปรแกรม แต่ในที่นี้เราสามารถใส่เครื่องมือในประเภท “ สิ่งที่คุณเห็น คือสิ่งที่คุณได้ (What you see is what you get : wysiwyg) ” เพื่อแก้ไขรีซอร์สโปรแกรม เช่นถ้าต้องการแก้ไขลิเนน จากอาเรย์ ของไอคอน ที่เราเรียกว่าคอนโทรลพาเลตต์ (Control palette) หรือบางทีเรียกว่า ทูลบ็อกซ์ และได้วาง และปรับขนาดของปุ่ม (buttons) ลิสต์บ็อกซ์ (List Boxes) และอื่น ๆ ด้วยเม้าร์ ดังนั้นเราสามารถที่จะแก้ไขรีซอร์สต่าง ๆ ได้อย่างมีประสิทธิภาพ

2.1.5 การจัดหน่วยความจำ (Memory Management)

เทคนิคการจัดหน่วยความจำ สำหรับวินโดวส์ รวมทั้งคอมไพเลอร์ ของ Visual C++ ได้เตรียมลักษณะการจัดการหน่วยความจำเพิ่มมากขึ้น ผลสุดท้ายขนาดของหน่วยความจำไม่ใช่ปัญหาอีกต่อไปแล้ว ซึ่งเพียงคุณจองหน่วยความจำ ที่คุณต้องการ แล้ววินโดวส์จะจัดการรายละเอียดย่อย ๆ ให้เอง ส่วนต่าง ๆ ของโปรแกรม รวมทั้งรีซอร์สสามารถถูกย้ายจากดิสก์ และสลับเปลี่ยนเป็นหน่วยความจำฟิซิคัลให้โดยอัตโนมัติ

2.1.6 Dynamics Link Libraries (DLLs)

ในสภาพแวดล้อมของดอส อ็อบเจ็กต์โมดูล (Object Module) ของโปรแกรมตัวหนึ่ง ๆ ทั้งหมดจะถูกลิงก์แบบสแตติก (Static) ในระหว่างการสร้างโปรแกรมในวินโดวส์ ได้อนุญาตให้มีการลิงก์แบบไดนามิก (Dynamics) ซึ่งหมายความว่าไลบรารีที่ถูกสร้างแบบพิเศษ นี้สามารถถูกโหลด และลิงก์ในขณะที่รันได้ ซึ่งเป็นการประหยัดหน่วยความจำ และเนื้อที่ดิสก์ การลิงก์แบบไดนามิกช่วยให้โปรแกรมเป็นโมดูลมากยิ่งขึ้น เนื่องจากความสามารถในการคอมไพล์ และทดสอบ DLLs ที่แยกจากกันได้

2.1.7 Win32 Application Programming Interface (API)

การพัฒนาแอปพลิเคชันแบบ 32 บิตบนวินโดวส์ API ของภาษา C ให้โปรโตไทป์ของฟังก์ชัน Win32 นี้แตกต่างจาก Win16 นอกจากนี้ยังมีฟังก์ชันใหม่อีกมากมาย และฟังก์ชันเฉพาะ ในการจัดการกับดิสก์ แอปพลิเคชัน 32 บิต ติดต่อกับไฟล์ โดยผ่านทาง Win32 แทนที่จะผ่านทาง MS-DOS API

การเปลี่ยนแปลงแอปพลิเคชันโปรแกรมแบบ 16 บิตที่เขียนด้วยภาษา C เป็นแอปพลิเคชันแบบ 32 บิต ต้องการเปลี่ยนแปลงเป็นอย่างมาก แต่ถ้าหากเราใช้ Microsoft Foundation Class Library (MFC) ต้องการคอมไพล์ใหม่เท่านั้น เพราะ MFC ถูกออกแบบโดยคำนึงถึง Win32 API อยู่แล้ว

2.2 ส่วนประกอบของ Visual C++ (The Visual C++ Components)

Visual C++ เป็นระบบที่ใช้ในการพัฒนาแอปพลิเคชันบนวินโดวส์ สองระบบที่รวมอยู่ในตัวผลิตภัณฑ์เดียวกัน โดยเริ่มแรกของการพัฒนาซึ่งจะต้องเลือกสภาพแวดล้อมก่อนเสมอ สำหรับ Visual C++ สามารถเลือกสภาพแวดล้อมได้ดังนี้.-

ถ้าเราต้องการพัฒนาแอปพลิเคชันบนวินโดวส์เราสามารถพัฒนาบนสภาพแวดล้อมได้ดังนี้

- ภาษา C ด้วยการ ใช้ Win32 API
- เขียน C++ ที่เป็นคลาสไลบรารีด้วยการ ใช้ Win32
- เลือกใช้ MFC แอปพลิเคชัน Framework โครงการนี้เลือกใช้วิธีนี้ในการพัฒนา
- ใช้แอปพลิเคชันของวินโดวส์ อื่น ๆ ของ Borland 's

ส่วนประกอบของ Visual C++ ประกอบด้วย

- กระบวนการในการสร้างโปรแกรม (The Microsoft Developer Studio and Build Process)
- Graphics Editor
- Compiler C /C++
- รีซอสคอมไพล์
- ลิงก์เกอร์

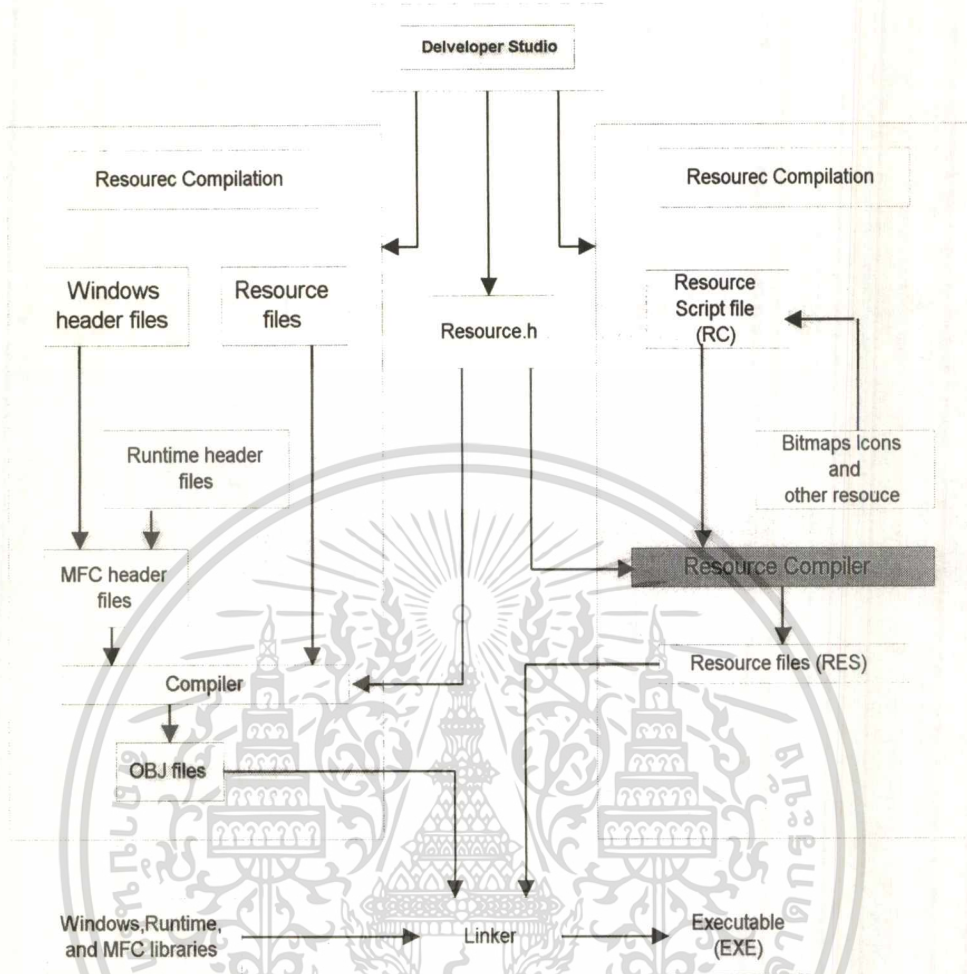
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ดีบั๊กเกอร์
- Appwizard
- ClassWizard
- Source Browser
- Online Help
- Windows Diagnostic Tools
- Source Code Control
- Component Gallery
- Microsoft Foundation Class Library Version 4.0

สำหรับรายละเอียดของส่วนโปรแกรมที่ใช้มีดังนี้.-

2.2.1 กระบวนการในการสร้างโปรแกรม (The Microsoft Developer Studio and Build Process)

Developer Studio คือ วินโดวส์หลักที่รวมสภาพแวดล้อมที่สามารถพัฒนางานได้หลายภาษา เช่น Visual C++ , Microsoft FORTRAN (Windows-based integrated development enviroment : IDE) ซึ่งภายในประกอบด้วย Docking Windows , Configurable toolbars , รันมาโคร และผู้ใช้สามารถเลือกได้อีกมากมาย



รูปที่ 2.1 แสดงขบวนการสร้างแอปพลิเคชัน

2.2.2 Graphics Editor

ลักษณะของการใช้อิติดิตเตอร์ จะเป็นแบบ wysiwyg ตามที่ได้กล่าวมาแล้วนั้น ซึ่งสามารถเพิ่ม OLE โดยทั่วไปแล้วแต่ละโปรเจกมีไฟล์ .RC เพียงตัวเดียว โดยใช้สแตจเมนต์ #Include เพื่อรวบรวมรีซอร์สจากซิปไดเรกทอรี และไม่ควรรแก้ไขไฟล์จากภายนอก



รูปที่ 2.2 แสดงการใช้ Graphics Editor

2.2.3 Compiler C /C++

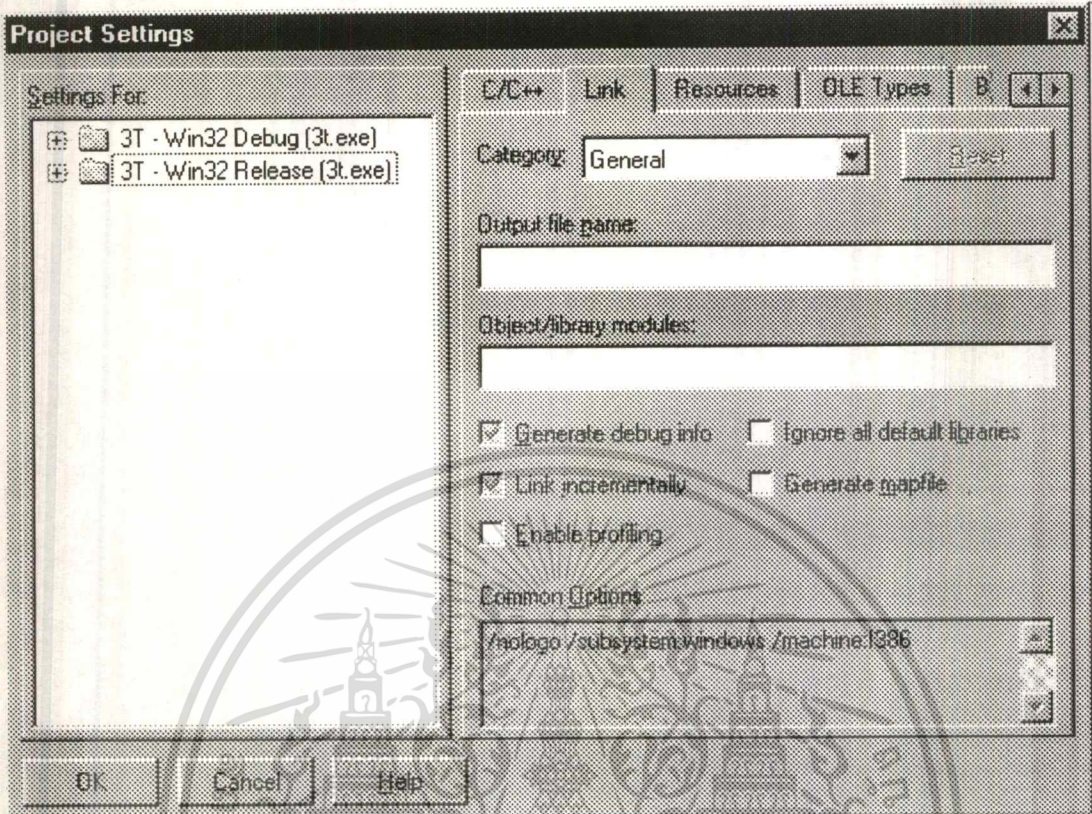
คอมไพเลอร์ใน Visual C++ สามารถจัดการกับซอร์สโค้ด ที่เขียนด้วย C และ C++ ก็ได้ โดยที่การที่จะรู้ว่าเป็นภาษาไหนนั้น ให้ดูที่นามสกุลของไฟล์ ถ้าเป็น C หมายถึง ซอร์สโค้ดที่มาจากการใช้ภาษา C เขียน และถ้าเป็น CPP หรือ CXX ก็เป็น ซอร์สโค้ดที่มาจาก C++ คอมไพเลอร์นี้เป็นไปตาม ANSI เวอร์ชัน 2.1 และจะมีส่วนขยายเพิ่มเติมของ Microsoft เอง Template , Exception และ Runtime type identification (RTTI)

2.2.4 รีซอสคอมไพล์

รีซอร์สคอมไพเลอร์ ของ Visual C++ ทำงานอยู่ในโหมด Copiler หรือโหมด Bind สำหรับคอมไพเลอร์นั้นไฟล์ ASCII Resource (RC) ที่เป็น Graphics Edit จะถูกคอมไพล์เป็นไบนารี RES สำหรับโหมด Bind ไฟล์ RES จะถูก รวมกับไฟล์ execute (EXE) ถ้ามีการแก้ไขไฟล์ RES เราสามารถที่จะรวมไฟล์ (Rebind) เข้าไปในไฟล์ .EXE โดยไม่จำเป็นต้องลิงก์ใหม่

2.2.5 ลิงก์เกอร์

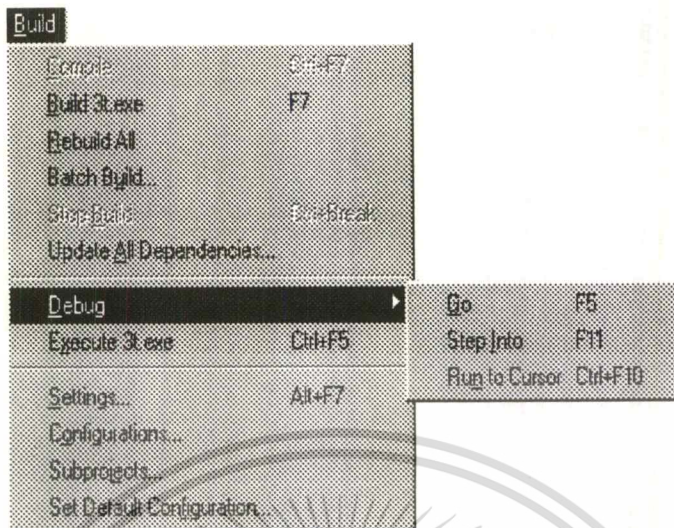
ลิงก์เกอร์เป็นการอ่านไฟล์ .OBJ และ RES ที่ถูกคอมไพล์จากไฟล์ C/ C++ และรีซอร์สคอมไพเลอร์ และมีการเข้าถึง LIB สำหรับ MFC รันไทม์ไลบรารีโค้ด และวินโดวโค้ด ถ้าหากเราต้องการทำการลิงก์เกอร์ แบบ minimize สำหรับช่วงของการ execute เราต้องทำการแก้ไขไฟล์ ที่ MFC จุด header จะเพิ่ม #prama



รูปที่ 2.3 แสดงการตั้งค่าการ Link

2.2.6 ดีบั๊กเกอร์

ดีบั๊กเกอร์ของ Visual C++ เป็นสภาพแวดล้อมในการดีบั๊กบนวินโดวส์ สำหรับ C++ การดีบั๊กเกอร์ จะเป็นการกำหนดพ้อยเตอร์ แล้วเก็บลงดิสก์ โดยประกอบด้วย ปุ่ม toolbars และการควบคุมที่ละเอียด



รูปที่ 2.4 แสดงเมนูของ Debug

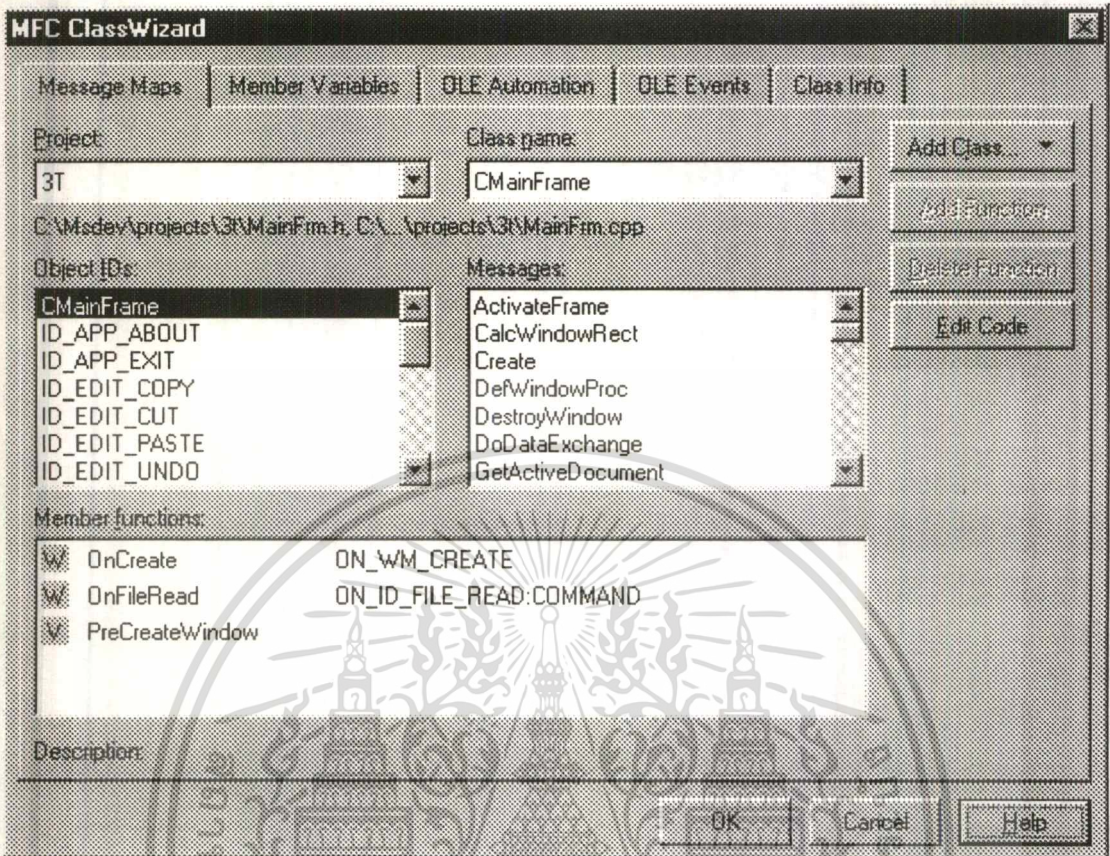
2.2.7 Appwizard

Appwizard เป็นโปรแกรมที่ใช้สำหรับสร้างโค้ดต้นแบบ (Skeleton) สำหรับแอปพลิเคชันบนวินโดวส์ โดยที่เรากำหนดลักษณะ ชื่อคลาส และชื่อรีซอร์ส โค้ดไฟล์ได้โดยผ่านทางได้อล็อกบ็อก และอย่าลืมว่า Appwizard เป็นส่วนหนึ่งของ Application Framework ที่เป็นเบสคลาส จุดประสงค์ก็เพื่อให้เราได้สร้างแอปพลิเคชันได้เร็วยิ่งขึ้น

2.2.8 ClassWizard

คลาส wizard เป็นโปรแกรม (โปรแกรมที่เป็น DLL) ที่ทำงานได้จาก App Studio ของ View menu โดยที่คลาส wizard ทำหน้าที่ที่เป็นงานหนัก และนำเบื่อง่าย ในการรักษาโค้ดของคลาสใน Visula C++ มีความจำเป็นต้องสร้างโค้ดใหม่ หรือฟังก์ชันใหม่เพื่อจัดการกับเมสเสจของวินโดวส์ หรือไม่ก็ เป็นการสร้าง Prototype ส่วนตัวของคลาส wizard ยังสามารถแก้ไขโค้ดในคลาส ซึ่งเราเขียนแล้ว ดังนั้นเราสามารถหลีกเลี่ยงปัญหาที่เกิดจากการสร้าง โค้ดธรรมดา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



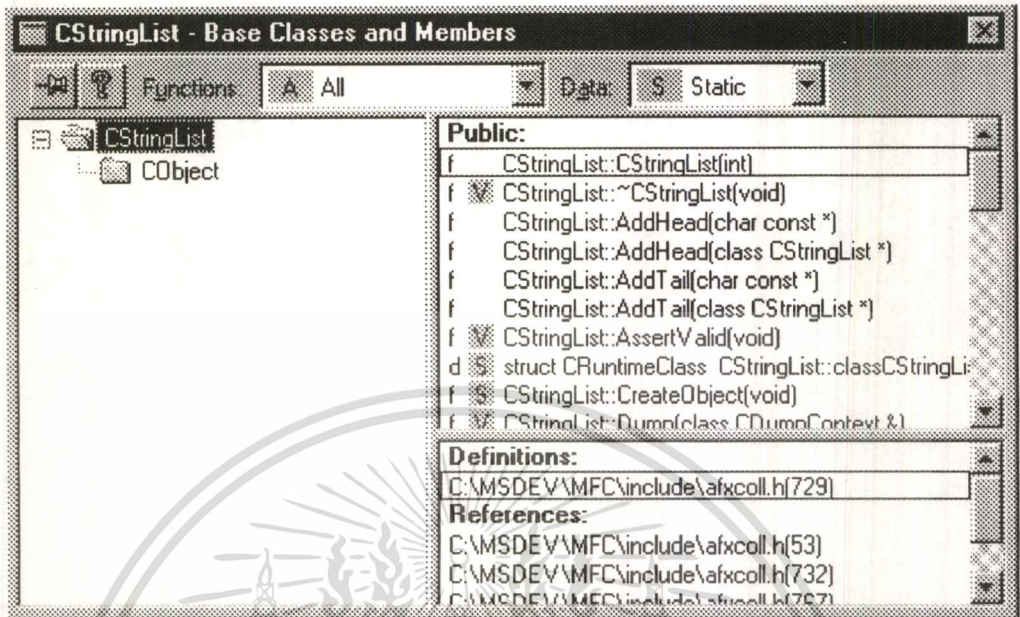
รูปที่ 2.5 แสดง ClassWizard

2.2.9 Source Browser

ถ้าเราเขียนโปรแกรมอย่างที่เราคิด เรายังถึงไฟล์ซอร์สโค้ด ของเรา รวมทั้งคลาส และเมมเบอร์ฟังก์ชัน แต่ถ้ามีการเขียนมาจากคนอื่น เราต้องอาศัย Visual C++ Source Browser ต่อไป จะเรียกสั้น ๆ ว่า Browser ซึ่งจะช่วยให้เราแก้ไขในระดับคลาส และฟังก์ชัน แทนที่จะเป็นระดับไฟล์ ซึ่งคล้ายกับว่าเป็น Inspector โดยที่ Browser มี View mode ในการทำงานดังนี้.-

- ข้อกำหนด และการอ้างอิง (Definition and Reference) เราเลือกฟังก์ชันใด ๆ ตัวแปร ประเภทตัวแปรแมโคร หรือคลาส และนอกจากนั้นยังตรวจสอบดูว่ามันถูกกำหนด แลใช้ในโปรเจกต์แบบใด
- Call Graph / Caller Graph สำหรับฟังก์ชันที่เรียก ซึ่งเราสามารถดูรูป กราฟฟิก แสดงให้เห็นถึงฟังก์ชันที่มันเรียก และฟังก์ชันที่มันถูกเรียก
- Derived Class Graph / Base Class Graph เป็นไดอะแกรม แสดงถึงลำดับของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษายเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 แสดงการใช้ Source Browser

2.2.10 Online Help

ทีมงานที่สร้าง หรือพัฒนาจะต้องการเอกสารในการช่วยสร้างเป็นอย่างมาก เพราะงานสร้างเราถือว่าเป็นงานที่หนัก ดังนั้น Visual C++ Ver 4.0 ได้มี Info View ที่ใช้เทคโนโลยีแบบ MSDN CDs ,MFC Libraries , Runtime Library และ Win32 API เพื่อช่วยให้โปรแกรมเมอร์ทำงานได้อย่างมีประสิทธิภาพมากยิ่งขึ้น

2.2.11 Windows Diagnostic Tools

เครื่องมือสำหรับการวิเคราะห์ใน Visual C++ มีดังนี้-

- SPY++ ทำหน้าที่ให้เห็น tree view ของขบวนการทำงาน threads และวินโดวส์ รวมไปถึงการตรวจสอบเมสเสจ ขณะทีวินโดวส์ทำการรัน
- มี OLE Utilies เพื่อให้ OLE ควบคุมการตรวจสอบการทำงานของโปรแกรม HC31 ช่วยให้คอมไพเลอร์ การจัดการ Library การทำ binary file Viewer และการ Edit ซอร์สโค้ด
- มี DBWIN เป็นการแสดงผลการวิเคราะห์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

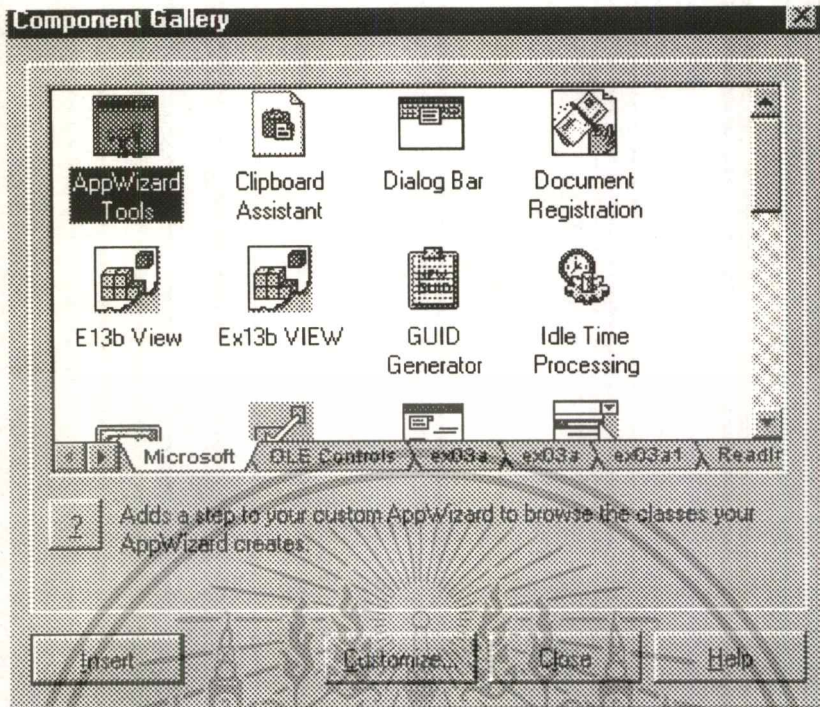
2.2.12 Source Code Control

ซอร์สโค้ดคอนโทรล เป็นส่วนที่ไม่ใครซอฟต์แวร์ได้สร้างเพื่อความถูกต้องของ Source Code โดยให้ชื่อว่า SourceSafe แต่ไม่รวมอยู่ใน Visual C++ และสามารถใช้งานร่วมกับ Developer Studio ได้เป็นอย่างดี

2.2.13 Component Gallery

Component Gallery เป็นส่วนที่เพิ่มเติมขึ้นมาใหม่ เพื่อใช้งานในลักษณะ share software component ที่มีความแตกต่างของโปรเจกต์ โดยที่ Component Gallery มีการจัดการ อยู่ 3 แบบ คือ

- OLE Control เมื่อติดตั้ง OLE Control (OCX) สิ่งต่างๆ เหล่านั้นจะลงไปยัง วินโดวส์ และ OCXs จะปรากฏใน component gallery's windows ขณะที่เราทำงานอยู่ เราสามารถเรียกมาใช้งานได้
- C++ source modules ขณะที่เราเขียนโค้ดขึ้นมาใหม่ เราสามารถที่จะเพิ่ม โค้ดส่วนที่เป็น Component Gallery ได้ เราสามารถคัดลอก หรือเลือกมาจากโครงการอื่น ๆ ได้
- Wizard Modules ที่ component gallery สามารถเก็บ เครื่องมือที่จะเพิ่มเข้ามาใหม่ เป็นการเพิ่มเครื่องมือ นั้นหมายถึง การมีเมมเบอร์ฟังก์ชัน



รูปที่ 2.7 แสดง Component Gallery ที่มีให้ใน Version 4.0

2.2.14 Microsoft Foundation Class Library Version 4.0

MFC Library เป็น C++ Microsoft Windows API และภาษา C++ ปัจจุบันก็เป็นมาตรฐานสำหรับการพัฒนาด้านนี้ไปแล้ว อีกอย่างการอินเตอร์เฟส กับวินโดวส์ การเลือกผลิตภัณฑ์ จากบริษัทเดียวกัน ก็ย่อมที่จะดีกว่าบริษัทอื่นแน่นอน สำหรับโปรเจกต์ซอฟต์แวร์ที่มีขนาดใหญ่ ซึ่งต้องการส่วนประกอบที่เป็นโมดูล (Module) และการนำกลับมาใช้ใหม่ได้อีก (Reusable) เมื่อผู้พัฒนาต้องการความซับซ้อนมากขึ้น เราไม่อาจจะเขียน C ที่มีขนาดใหญ่ได้

การพัฒนาแอปพลิเคชัน บน Application Framework ใช้โครงสร้างที่เป็นมาตรฐานที่ได้รับการพิสูจน์แล้วว่า นักเขียนโปรแกรมส่วนมากจะเลือกใช้โครงสร้าง ที่เป็นคลาสไลบรารี เพราะง่ายต่อการบำรุงรักษา แอปพลิเคชัน บน Application Framework มีขนาดเล็ก และทำงานได้เร็ว เช่น การใช้ 32 บิต Address ทุก ๆ สิ่ง และเป็นแบบอินทีเจอร์ เพราะจะให้มีประสิทธิภาพที่ดีกว่า Win16 แต่ได้เพิ่มเติมความสามารถขึ้นไปอีกมากมาย เช่น

- คลาสที่ใช้งานทั่วไป รวมทั้งคลาสที่รวมเอาลิส อาร์เรย์ และแมป คลาสที่เกี่ยวกับสตริง ที่มากด้วยประสิทธิภาพคลาส Time, time span และ date คลาสที่เกี่ยวกับไฟล์ที่ไม่ขึ้นอยู่กับระบบปฏิบัติการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- รองรับแอปเจ็ทที่มีระบบสำหรับการเก็บข้อมูลลง - นำออก เช่น Save , Save As ,Menu item และอื่น ๆ ที่เกี่ยวกับไฟล์
- ออบเจ็กต์ที่ใช้งานร่วมกับคลาสลำดับ
- รองรับการอินเตอร์เฟซแบบ (Multiple Document Interface: MDI)
- รองรับ OLE (Object Linking and Embedding)
- รองรับสกอลบาร์ ทูลบาร์ สเตตัสบาร์ และสปลิตเตอร์วินโดว์
- การใช้งานคอนโทรล Microsoft Visual Basic
- รองรับ Context-sensitive Help
- รองรับการจัดการข้อมูลโดยอัตโนมัติภายในไดออล็อกเลือก
- สนับสนุนการใช้ DLLs
- รองรับการใช้ ODBC (Open Data Base Connectivity) แก้ไขข้อมูลที่เก็บอยู่ในฐานข้อมูลทั่วไปได้
- มี OLE-based Data Access Object (DAO) สำหรับการใช้ Jet Engine

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การโปรแกรมด้วยไลบรารี MFC บน Windows

ไลบรารี (MFC) คลาสพื้นฐานของไมโครซอฟต์ เป็นแนวทางใหม่ของการเขียนโปรแกรมบนสภาวะแวดล้อมของ Windows และยังมีสิ่งอำนวยความสะดวกมากกว่าเก่ามาก การเขียนโปรแกรมโดยใช้ไลบรารี MFC นั้นเราจะมีขบวนการ (routine) ของ Winmain() โปรแกรมคำสั่งสำหรับเปลี่ยนทิศทางของเมสเสจ (ข้อมูลข่าวสาร) ใช้ MakeProclInstance() และมีระบบป้องกันและรักษาคุณลักษณะของข้อมูล ซึ่งแตกต่างไปจากการเขียนโปรแกรมแบบจินตภาพ โดยลักษณะการเขียนโปรแกรมอีกลักษณะหนึ่งที่อ้างอิงกับระบบ SDK มาตรฐานสำหรับ Windows ส่วนการใช้ไลบรารี MFC นี้สำหรับเขียนโปรแกรมให้กับระบบปฏิบัติการบน Windows ที่ใช้งานกันอยู่ MFC ประสบความสำเร็จมากกว่า เนื่องจาก การเขียนโปรแกรมบน Windows ด้วยวิธีนี้ทำให้การเชื่อมต่อกับระบบ Windows พื้นฐานเป็นไปได้ง่ายดาย เราจะพบว่าการเริ่มต้นเขียนโปรแกรมบน Windows ด้วย MFC ง่ายมาก แม้ว่าเราไม่เคยเขียนโปรแกรมบน Windows มาก่อนก็ตาม เราควรจะปฏิบัติตามตัวอย่างต่าง ๆ ในหนังสือเล่มนี้เพื่อให้เราคุ้นเคย และเข้าใจการเขียนโปรแกรมบน Windows การเขียนโปรแกรมบน Windows เป็นการเรียนรู้วิธีการใช้งานเมสเสจจาก Windows เคอร์เนล (kernel) ถ้าเราสามารถเขียนโปรแกรมบน Windows ด้วย MFC แล้วเราจะสามารถเปลี่ยนไปเขียนโครงการอื่น ๆ ได้อีก โดยการใช้ความรู้พื้นฐานนี้เป็นตัวอ้างอิงสำหรับเรียนรู้ระบบปฏิบัติการของ Windows มากขึ้นไปอีก

ถ้าเรามีประสบการณ์ในการเขียนโปรแกรม Windows บ้างแล้ว การเรียนรู้วิธีใช้ MFC สำหรับเขียนโปรแกรมบน Windows จะเป็นการยุ่งยากสำหรับเราที่จะทำความเข้าใจโครงสร้างของ MFC และอาจถึงกับตกใจในวิธีใหม่นี้ เราจะพบว่าการใช้งานไลบรารี MFC มีความสะดวกสบายมากกว่า และน่าเชื่อถือมากกว่าการเขียนโปรแกรมด้วยภาษาซีแบบเก่า

ใน 3 หัวข้อแรกของบทนี้ เราจะดูในเรื่ององค์ประกอบสำคัญของไลบรารี MFC โดยที่ระบบทั่วไปของคลาสที่ใช้งานใน Windows ของไลบรารีต่าง ๆ จะมีคลาสพื้นฐานหลักอยู่ในไลบรารี MFC และสถิติวิธีการส่งเมสเสจซึ่งเป็นรากฐานของการเขียนโปรแกรมบน Windows ด้วย MFC ถ้าเราชอบที่จะเริ่มต้นด้วยตัวอย่างที่เห็นได้ชัดแล้วให้เรากระโดดหัวข้อเหล่านี้ไปที่ตัวอย่าง HelloApp ซึ่งแสดงตัวอย่างของการเขียนโปรแกรมด้วย MFC บน Windows ส่วนรายละเอียดให้เรากลับมาอ่านพื้นฐานของ MFC ในหัวข้อเหล่านี้อีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ในตอนแรกของบทนี้ เราจะได้ศึกษาเกี่ยวกับเรื่องของคลาส CWinApp และ CWnd
โครงร่างของเมสเสจ

3.1 คลาสไลบรารี MFC เพื่อการพัฒนา Windows

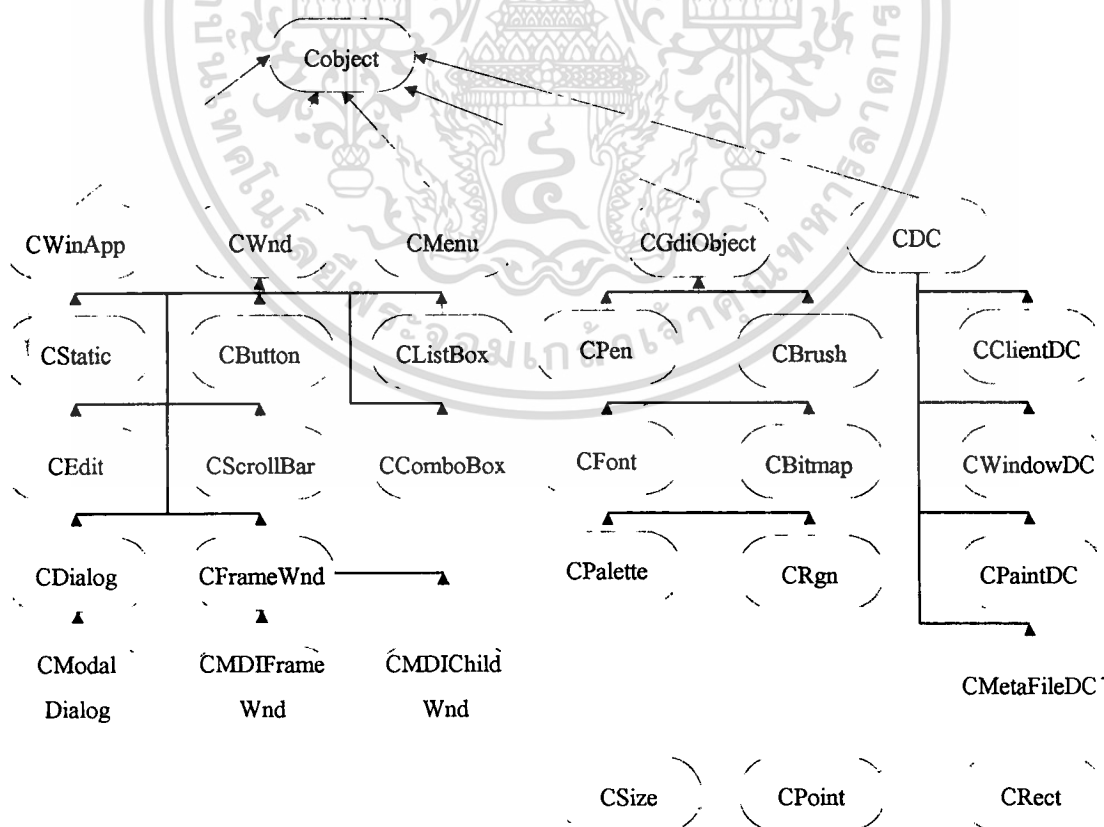
คลาสของไลบรารี MFC ได้พัฒนารูปแบบของการเขียนโปรแกรมเชื่อมต่อกับ Windows โดยคลาสไลบรารีเหล่านี้จะเป็นตัวกลางระหว่างเรากับ Windows แม้ว่าพวกมันจะเปลี่ยนแปลงพื้นที่และเวลาในการทำงานแต่มีประโยชน์มากมายในเรื่องของความน่าเชื่อถือและการเรียกใช้งานซ้ำได้สถาปัตยกรรมของไลบรารี MFC สะท้อนให้เห็นถึงความต้องการของ WINDOWS API ดังแผนผังในรูปที่ 3.1 แสดงชื่อของคลาสต่าง ๆ ที่เกี่ยวข้อง โดยคลาสไลบรารี MFC จะตอบสนององค์ประกอบของ WINDOWS ทุกอย่างคลาสทั้งห้าในรูปที่ 3.1 ซึ่งโยงลงมาจาก Cobject คือ CWinApp, Cwnd, CMenu, CGdiObject และ CDC เป็นคลาสหลักสำหรับการเชื่อมต่อกับส่วนต่างๆ พวกมันทำหน้าที่ควบคุม และดูแลให้ส่วนเชื่อมต่อของ Windows ที่เชื่อถือไม่ค่อยได้ ให้เป็นไปตามกฎเกณฑ์ และพวกมันจะดูแลในรายละเอียดระดับต่ำทั้งหลายด้วย

คลาสทั้งห้านี้ถูกสร้างโดยตรงจาก Cobject ซึ่งเป็นคลาสพื้นฐานในแผนภูมิ (เหมือนต้นไม้กลับหัว) แสดงตระกูลของ MFC บน Windows รายละเอียดภายในของแต่ละตัวคือลักษณะในการเชื่อมต่อกับ Windows แต่ละแบบนั่นเอง Cwnd, CGdiObject และ CDC แยกแขนงออกไปเป็นฟังก์ชันต่างๆ มากมาย เราจะได้เรียนรู้คลาสที่ถูกสร้างขึ้นมาแบบเฉพาะงานแต่ละแบบ โดยคงซึ่งคลาสพื้นฐานเดิมของมัน เราอาจจะหลงลืมฟังก์ชันสมาชิกบางตัวของ CFrameWnd ซึ่งเป็นต้นตระกูล (คลาสพ่อ แม่) ของมัน Cwnd มีฟังก์ชันสมาชิกเพิ่มขึ้นมากกว่า 200 ตัวให้เราพิจารณา ถ้าการสร้างขึ้นมาใหม่คือ a แล้ว CFrameWnd ก็คือ Cwnd ของ a

คลาสทั้งสามในรูปที่ 3.1 ซึ่งไม่ได้โยงลงมา Cobject คือ Csize, Cpoint และ Crect เป็นคลาสของค่าข้อมูลซึ่งปฏิบัติตามโครงสร้างข้อมูลพื้นฐานของ Windows เนื่องจากคลาสเหล่านี้ถูกใช้งานในนิพจน์ และเป็นผลลัพธ์ที่ได้มาจากฟังก์ชันต่างๆ ดังนั้นพวกมันจึงต้องปฏิบัติด้วยพื้นที่และประสิทธิภาพของเวลาที่ที่สุด พวกนี้ไม่สามารถเรียกช้อนภายในคลาสที่ถูกสร้างมาจาก Cobject

3.1.1 ฟังก์ชันสมาชิก เทียบกับ ฟังก์ชัน Windows API

ฟังก์ชันสมาชิกต่างๆ ของคลาสไลบรารี MFC ของ Windows มีชื่อเหมือนกับฟังก์ชันใน Windows API เช่น คลาส CScrollBar บรรจุด้วยฟังก์ชันสมาชิก GetScrollPos(), SetScrollPos(), GetScrollRange() และ SetScrollRange() เราอาจจะคุ้นเคยกับฟังก์ชันเหล่านี้ในส่วนของ Windows API มาตรฐาน เมื่อเราดูที่โปรแกรมคำสั่งมันง่ายที่จะบอกความแตกต่างของพวกมัน เนื่องจากฟังก์ชัน Windows API ส่วนใหญ่ใช้ HANDLE เป็นพารามิเตอร์ตัวแรก ส่วนข้อแตกต่างที่เห็นได้อีกอย่างคือฟังก์ชันสมาชิกเหล่านี้เป็นสมาชิกที่มีเหตุมีผลดีคือ พวกมันถูกเรียกใช้งานได้โดยการใช้ obj.fn() หรือ objptr->fn() (นอกเสียจากว่าพวกมันจะถูกเรียกใช้จากในฟังก์ชันสมาชิกของคลาสเดียวกัน) เราสามารถอ้างอิงถึงกระบวนการทั่วไปของ Windows ได้โดยการใส่ :: (ตัวดำเนินการกำหนดขอบเขต) ให้นำชื่อของฟังก์ชันนั้น เช่น ::SetScrollPos() ถ้าไม่นำหน้าด้วยชื่อคลาสแล้วตัวดำเนินการกำหนดขอบเขต :: จะอ้างถึงขอบเขตหลัก (global scope) ซึ่งอยู่ในฟังก์ชันปกติของ Windows หากเราไม่ได้กำหนดความต้องการตายตัวลงไปแล้ว การวาง :: หน้าชื่อฟังก์ชัน Windows API ทำให้โปรแกรมของเราอ่านง่ายขึ้น



รูปที่ 3.1 แสดงคลาสพื้นฐานของไมโครซอฟต์ (MFC) สำหรับการพัฒนาโปรแกรมบน Windows

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 ฟังก์ชันสมาชิก

เราจะกล่าวถึงฟังก์ชัน scroll ในตัวอย่าง เพราะว่าพวกมันแสดงวิธีการทำซ้ำไว้หลายชนิด คลาสทั้งสองตัวในไลบรารี MFC นี้มีชื่อของฟังก์ชันสมาชิกเหมือนกัน ทั้งคลาส Cwnd และ CScrollBar มีฟังก์ชันสมาชิกของสกรอลล์บาร์อยู่สี่ตัว ถ้าเรามีหน้าต่างหนึ่งซึ่งต้องการใช้สกรอลล์บาร์แบบแนวตั้งและแนวนอนแบบมาตรฐานแล้ว เราควรจะใช้ฟังก์ชัน scrollbar จาก Cwnd แต่ถ้าเราต้องการใช้สกรอลล์บาร์ เพื่อจุดประสงค์พิเศษเช่นตัวเลือกแบบมีสีในไดอะล็อกบ็อกซ์ เราควรจะใช้ฟังก์ชันสมาชิก CscrollBar

3.2 ส่วนสำคัญพื้นฐานของ MFC

การสร้างโปรแกรมประยุกต์ด้วย MFC มักมีผลต่อคลาสทั้งสามแบบคือ คลาสที่เราสร้างเอง คลาสของ MFC และคลาสที่เราสร้างขึ้นมาจากคลาสของ MFC เราจะมาดูรายละเอียดของคลาสใน MFC ส่วนคลาสที่เราสร้างขึ้นเองนั้นเราต้องเข้าใจวิธีการใช้งานแน่นอน แม้ว่าเราสามารถหาทำได้จากการใช้คลาส MFC เป็นคลาสพื้นฐานสำหรับสร้างคลาสอื่นขึ้นมาใหม่ แล้วคลาส MFC ทั้งหมดที่ถูกสร้างขึ้นมาจาก Cwnd ก็ยังสามารถให้บริการเป็นคลาสพื้นฐานกับคลาสอื่นๆ ได้อีก

- WinApp เป็นส่วนสำคัญสำหรับการประยุกต์ใช้ MFC บน Windows โปรแกรมประยุกต์บน Windows ซึ่งถูกพัฒนาด้วยไลบรารี MFC นั้นจะสร้างคลาสต่างๆ ขึ้นมาใช้งานจาก CWinApp
- CFrameWnd จะทำส่วนพื้นฐานต่างๆ เช่นกรอบ หน้าต่างหัวเรื่อง ซึ่งเหมือนกับหน้าต่างหลักของโปรแกรมประยุกต์ส่วนใหญ่บน Windows โปรแกรมประยุกต์ที่ใช้ MFC ส่วนใหญ่จะสร้างคลาสต่างๆ ขึ้นมาจาก CFrameWnd เพื่อจัดการส่วนพื้นฐานของโปรแกรม
- CMDIFrameWnd และ CMDIChildWnd นำมาใช้แทน CFrameWnd ในโปรแกรมประยุกต์สำหรับเชื่อมต่อเอกสารหลายอย่าง (MDI หรือ multiple document interface)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- CDialog และ CModalDialog ใช้ทำอะไรคือบล็อกบ็อกซ์ทั่วไป คลาส (ของโปรแกรมประยุกต์) ที่ถูกสร้างขึ้นมาจากคลาสเหล่านี้จะทำหน้าที่สร้างไดอะล็อกบ็อกซ์ที่ต้องการได้

3.2.1 CWinApp

CWinApp เป็นส่วนสำคัญของคลาสที่ใช้ในโปรแกรมประยุกต์ของ MFC ทั้งหมด โปรแกรม Hello ที่จะอธิบายต่อไปในบทนี้ คลาส CHelloApp ถูกสร้างขึ้นมาจาก CWinApp CWinApp ใช้กำหนดคุณลักษณะภายในที่สำคัญของโปรแกรมประยุกต์บน Windows โดยปกติจะเป็นคุณลักษณะซึ่งถูกจัดการโดย WinMain() ในโปรแกรมประยุกต์ที่เขียนด้วยภาษา C บน Windows คลาสที่ถูกสร้างขึ้นใหม่ในโปรแกรมของเราจะไปซ้อนทับฟังก์ชันสมาชิก InitInstance() ของ CWinApp ซึ่งใช้กำหนดค่าข้อมูลเริ่มต้น โดยมันจะสร้างหน้าต่างหลักให้กับโปรแกรมประยุกต์นั้น และปฏิบัติตามกระบวนการสำหรับการเริ่มต้นระบบโดยปกติ เราต้องจำไว้ด้วยว่าต้องประกาศออบเจกต์หลัก (global) ที่ใช้ในโปรแกรมของเราด้วย เนื่องจากการสร้าง ออบเจกต์นั้นเป็นการกำหนดโครงสร้างเริ่มต้นของโปรแกรมทั้งหมด ในการพัฒนาโปรแกรมที่ใช้ MFC ด้วยการใช้ออบเจกต์ที่ถูกสร้างขึ้นมาจาก CWinApp อย่างเดียว แล้วก็เหมือนกับว่าเราต้องเขียนกระบวนการ (ฟังก์ชัน) main() ในโปรแกรมภาษา C ทั่วไป หรือ WinMain() ในโปรแกรม (ไม่ใช่ MFC) ทั่วไปบน Windows นั่นเอง

3.2.2 CFrameWnd, CMDIFrameWnd และ CMDIChildWnd

CFrameWnd เป็นคลาสโลบวารีของ MFC สำหรับสร้างหน้าต่างที่เชื่อมต่อเอกสารเดี่ยวๆ (single document interface หรือ SDI) หน้าหลักคือการเพิ่มคุณลักษณะในการบริหารหน้าต่าง ตัวเร่ง (accelerator table) และยังมีกระบวนการ Create() ซึ่งสะดวกกว่าใช้ Create() ของคลาสพื้นฐาน Cwnd การประยุกต์ใช้ระบบเชื่อมต่อเอกสารเดี่ยวของ MFC นี้มีคลาสที่สร้างขึ้นมาจาก CFrameWnd เพื่อให้โปรแกรมประยุกต์สามารถตอบสนองกับเมสเสจของ Windows ที่ต้องการได้ เช่น ถ้าโปรแกรมประยุกต์ต้องการตอบสนองเมสเสจ WM_PAINT มันควรมีคลาสที่ถูกสร้างขึ้นมาจาก CFrameWnd ซึ่งบรรจุฟังก์ชันสมาชิก OnPaint() เพื่อจัดการเมสเสจ WM_PAINT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมประยุกต์มากมายใช้ระบบเชื่อมต่อเอกสารเดี่ยว (MDI) นี้ ไม่ใช่ SDI และโปรแกรมแบบ MDI ใช้ CMDIFrameWnd และ CMDIChildWnd แทนการใช้ CFrameWnd เป็นคลาสพื้นฐาน

3.2.3 CDialog และ CModalDialog

Cdialog และ CModalDialog เป็นคลาสไลบรารีของ MFC ซึ่งเป็นคลาสพื้นฐานหลักให้กับคลาสต่างๆ ที่กำหนดในโปรแกรมประยุกต์ ไดอะล็อกบ็อกซ์ส่วนใหญ่จะมีส่วนควบคุมต่าง ๆ นานาต้องการตอบสนองเมสเสจต่างๆ และต้องการทำงานแทนส่วนที่สลับซับซ้อน ไดอะล็อกบ็อกซ์มักจะเกิดจากคลาสที่ถูกสร้างขึ้นมาจากคลาส Cdialog และ CModalDialog คลาสที่สร้างขึ้นใหม่จะบรรจุฟังก์ชันสมาชิกซึ่งตอบสนองเมสเสจของระบบควบคุม สมาชิกข้อมูลซึ่งสัมพันธ์กับระบบควบคุม และอาจจะซ่อนทับ (ทำงานแทน) ฟังก์ชันสมาชิกบางตัวของคลาสพื้นฐานของมัน ไดอะล็อกบ็อกซ์ธรรมดาทั่วไป เช่น ไดอะล็อกบ็อกซ์แสดงที่มาของโปรแกรม (About) ซึ่งบรรจุตัวควบคุมคงที่ (ตัวอักษร) กับปุ่ม OK เท่านั้น สามารถบริหารโดยคลาส CModalDialog

3.2.4 การจัดการเมสเสจ

โปรแกรมบน Windows ต้องการออกแบบให้ตอบสนองเมสเสจจาก Windows kernel (ส่วนที่เป็นแกนกลาง) ผู้อ่านซึ่งยังไม่คุ้นเคยกับหลักการจัดการเมสเสจควรจะอ่านตำราพื้นฐานการเขียนโปรแกรมบน Windows โดยโปรแกรม Windows ซึ่งเขียนด้วยภาษา C นั้นเราควรจะสร้างวงวนของเมสเสจ (message loop) เพื่อรอรับเมสเสจจากเคอเนลแล้วเราจึงเขียนประโยคคำสั่งจัดการกับเมสเสจนั้น ซึ่งรวมไปถึงตัวแปรท้องถิ่นจำนวนมากด้วย สำหรับเมสเสจซึ่งเราไม่อยากรจัดการ เราต้องเรียกใช้ DefWndProc() มาทำกระบวนการมาตรฐานอย่างระมัดระวัง ไลบรารี MFC สามารถจัดการกับเมสเสจได้ดีกว่าระบบดั้งเดิม โปรแกรมประยุกต์ของ MFC จะจัดการกับเมสเสจด้วยฟังก์ชันสมาชิกคลาสแบบหลากหลาย ไม่มีอะไรยากฟังก์ชันจัดการเมสเสจแต่ละตัวจะมีขอบเขตของตัวเอง และถ้าต้องการเรียกใช้คลาสพื้นฐานแล้วเราควรทำแบบธรรมดา ซึ่งปกติจะจัดการให้เองโดยอัตโนมัติ ใช้งานและมีระบบจัดการกับเมสเสจที่ดีกว่าเดิม เมสเสจมาตรฐานของ Windows ส่วนใหญ่ เช่น WM_PAINT และ WM_SIZE ถูกจัดการโดยหนึ่งในฟังก์ชันสมาชิกของระบบจัดการเมสเสจประมาณ 100 ตัวของคลาส Cwnd ฟังก์ชันเหล่านี้สังเกตง่ายเพราะชื่อของมันขึ้นต้นด้วย On เช่น OnPaint() และ OnSize() ฟังก์ชันสมาชิกของระบบจัดการเมสเสจของคลาส Cwnd สามารถเขียนทับได้ตามความจำเป็นในคลาสที่ถูกสร้างขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาใหม่ในโปรแกรมประยุกต์ เมสเสจจากหัวข้อเลือกเมนู และเมสเสจจากตัวควบคุมใดอะลือก บ็อกซ์ส่วนใหญ่ถูกจัดการด้วยฟังก์ชันสมาชิกของระบบจัดการเมสเสจที่เราเขียนขึ้น

3.2.5 โครงร่างเมสเสจ (Message Maps)

เมื่อเราสร้างคลาสหนึ่งจาก CFrameWnd(หรือจากCMDIFrameWnd, CMDIChildWnd, Cdialog หรือ CModalDialog) และเพิ่มฟังก์ชันสมาชิกของระบบจัดการเมสเสจของเราเองเข้าไปอีก แล้วเราต้องเพิ่ม "โครงร่างเมสเสจ" สำหรับคลาสนั้นเข้าไปด้วยตารางแสดงตำแหน่งของฟังก์ชันสมาชิกของระบบจัดการเมสเสจ โครงร่างเมสเสจอยู่ 2 ส่วนคือส่วนที่หนึ่งอยู่ในส่วนประกาศ (declaration) คลาส เราต้องใส่บรรทัดคำสั่งเข้าไปดังนี้

```
DECLARE_MESSAGE_MAP()
```

เพื่อประกาศข้อมูลและสมาชิกฟังก์ชันเพื่ออธิบายโครงร่างเมสเสจแล้ว เราต้องใส่ข้อกำหนดของโครงร่างเมสเสจที่แท้จริงในไฟล์ .cpp หนึ่งของเราซึ่งมักจะใกล้เคียงกับฟังก์ชันสมาชิกของคลาสนั้น โครงร่างเมสเสจมีทางเข้าทางหนึ่งสำหรับเมสเสจแต่ละตัว ซึ่งจัดการโดยคลาสนั้นๆ ตัวอย่างโครงร่างเมสเสจสั้นๆ ซึ่งประยุกต์มา

```
BEGIN_MESSAGE_MAP(CMainWindow, CFrameWnd)
    ON_WM_PAINT()
    ON_COMMAND(IDM_ABOUT, OnAbout)
END_MESSAGE_MAP()
```

โครงร่างนี้เมสเสจแสดงให้เห็นว่าคลาส CMainWindow ถูกสร้างขึ้นมาจากคลาส CFrameWnd ของ MFC และ ฟังก์ชันสมาชิกของมันตอบสนองเมสเสจบน Windows สองตัว บรรทัดแรกในตารางแสดงว่าเมสเสจ WM_PAINT ควรจะเรียกใช้งานฟังก์ชันสมาชิก OnPaint() ของ CMainWindow และบรรทัดที่สองของเมสเสจ IDM_ABOUT ควรถูกจัดการโดยฟังก์ชันสมาชิก OnAbout() ของ CMainWindow

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.6 มากกว่ากับคลาส CWinApp

ส่วนสำคัญสำหรับโปรแกรมประยุกต์ที่ใช้ MFC คือ คลาส CWinApp เป็นที่รองรับข่าวสารบางอย่างที่ส่งมายัง WinMain() เช่น ชื่อโปรแกรมประยุกต์ และค่าสำหรับจัดการชั่วคราวอาร์กิวเมนต์ทั้งสี่ตัวของ WinMain() ถูกเก็บไว้ในสมาชิกแบบ public ทั้ง 4 ของ CWinApp

```
HANDLE m_hInstance
HANDLE m_hPrevInstance
LPSTR m_lpCmdLine
int m_nCmdShow
```

สมาชิกแบบ public ที่สำคัญอีก 2 ตัวคือ CWinApp คือ m_pMainWnd ซึ่งถูกประกาศว่าเป็น CWIND และควรจะถูกตั้งค่าเริ่มต้นในฟังก์ชัน InitInstance() ที่ตำแหน่งแสดงออบเจกต์แสดงหน้าต่างหลักของโปรแกรมประยุกต์ และ m_pszAppName เป็นตัวชี้ข้อมูลไปยังชื่อของโปรแกรมประยุกต์นั้น ถ้าชื่อของโปรแกรมประยุกต์ไม่ได้กำหนดไว้ในตัวผู้สร้างของ CWinApp แล้วชื่อโดยปกติของมันคือชื่อเต็มของไฟล์ .exe ของโปรแกรมประยุกต์นั่นเอง

ในคลาสที่ถูกออกแบบมาเพื่อใช้ในโปรแกรมประยุกต์เดี่ยวๆ เช่น คลาส CHelloWindow ในโปรแกรมประยุกต์ HelloApp ง่ายต่อการอ่านข่าวสารซึ่งเก็บไว้ในออบเจกต์ CWinApp ทุกสิ่งทุกอย่างที่เราต้องการรู้คือ ชื่อออบเจกต์ CWinApp (HelloApp ในตัวอย่างของโปรแกรม HelloApp) และชื่อของสมาชิกที่เราต้องการเข้าถึง แต่คลาสที่ใช้งานหลากหลายต้องการวิธีอื่นเนื่องจากชื่อออบเจกต์ของโปรแกรมประยุกต์จะเปลี่ยนแปลงไปตามโปรแกรมประยุกต์ที่อ้างถึงอยู่ MFC มีชุดของฟังก์ชันหลัก (global) ซึ่งให้เราเข้าถึงออบเจกต์ CWinApp ของโปรแกรมประยุกต์ ฟังก์ชันที่กำหนดให้มากมายสำหรับการเข้าถึงส่วนจัดการและชื่อของโปรแกรมที่เราต้องการ

```
CWinApp* AfxGetApp()
HANDLE AfxGetInstanceHandle()
HANDLE AfxGetResourceHandle()
const char* AfxGetAppName()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชัน `AfxGetApp()` และ `AfxGetAppName()` ให้ผลลัพธ์เป็น `hInstance` ซึ่งถูกส่งผ่านโดย Windows ไปยัง `WinMain()` และเก็บไว้ในตัวแปรสมาชิก `m_hInstance` ของ `CWinApp` แต่เราควรใช้พวกมันตามคำอธิบายต่อไปนี้ สำหรับใช้จัดการงานจริง ณ เวลาปัจจุบันที่เราต้องการ เช่น ในการทำงานบนคลาสแสดงหน้าต่าง (ตัวอย่างเช่น ฟังก์ชัน `::GetClassInfo()` ของ Windows) จะใช้ `AfxGetInstanceHandle()` แต่เมื่อเราเกี่ยวข้องกับข้อมูล (ทรัพยากร) เช่น ในฟังก์ชัน `::LoadAccelerators()` และ `::LoadString()` ของ Windows จะใช้ `AfxGetResourceHandle()` การแยกแยะงานในลักษณะตอนนี้อาจจะทำให้ปลอดภัยในการใช้โปรแกรมในอนาคต

ในโปรแกรมภาษา C บน Windows บางครั้งจำเป็นต้องทำกระบวนการพิเศษสำหรับในช่วงเริ่มแรกของโปรแกรมใน `WinMain()` เป็นเรื่องง่ายเนื่องจาก `hPrevInstance` เป็น `NULL` ในช่วงแรกของโปรแกรมประยุกต์ คุณลักษณะนี้มีอยู่ใน `CWinApp` ด้วยถ้าเราเขียนทับฟังก์ชันสมาชิก `InitApplication()` ฟังก์ชันสมาชิกนี้จะถูกเรียกในช่วงแรกที่โปรแกรมถูกโหลดเท่านั้น ไม่ใช่ส่วนสำเนาที่เกิดขึ้นภายหลังถูกโหลด (การทำงานที่เหมือนกันนี้มีอยู่ในฟังก์ชันสมาชิก `InitInstance()` ด้วยเพื่อทดสอบตัวแปรสมาชิก `m_hInstance`)

3.3 คลาส `Cwnd`

คลาส `Cwnd` เป็นศูนย์กลางของความสะอาดกสบายสำหรับการแสดงหน้าจอบน Windows ของไลบรารี MFC มันเป็นคลาสพื้นฐานสำหรับ `CFrameWnd`, `Cdialog`, `Cstatic`, `Cbutton`, `CListBox`, `CComboBox`, `Cedit` และ `CScrollBar` และเรายังสามารถสร้างคลาสของเราขึ้นมาเองจาก `Cwnd` ได้ ในทำนองเดียวกันเราสามารถสร้างคลาสใหม่ขึ้นมาจากผลผลิตของ `Cwnd` เช่น `CFrameWnd` และ `Cdialog` ได้เช่นกันหนึ่งในสมาชิกข้อมูลแบบ `public` ของ `Cwnd` คือ `m_hWnd`, `HWND` ของ Windows สำหรับจัดการกับหน้าต่างโปรแกรม โดยปกติ เราไม่ต้องการใช้สมาชิกนี้เนื่องจาก `Cwnd` มีฟังก์ชันสมาชิกซึ่งยอมให้เราเข้าถึงสมาชิกของ Windows ธรรมดาทั้งหมดซึ่งต้องการพารามิเตอร์ตัวแรกเป็น `HWND` แต่บางครั้งเราต้องการทำให้เป็นแบบธรรมดาๆ โดยการใช้ `m_hWnd`

ฟังก์ชันสมาชิกของ `Cwnd` เป็นส่วนทดแทนฟังก์ชันปกติของ Windows ฟังก์ชันเหล่านี้มีชื่อเหมือนกันโดยสิ้นเชิงกับฉบับเทียบของ Windows ตัวอย่าง ฟังก์ชันสมาชิก `Cwnd` ที่ชื่อ `ShowWindow()` มีหน้าที่การทำงานเหมือนกับ `::ShowWindows()` ของ Windows ธรรมดา ฟังก์ชันสมาชิกของ `Cwnd` เป็นฟังก์ชันของตัวจัดการเมสเสจ ชื่อของฟังก์ชันตัวจัดการเมสเสจขึ้นต้นด้วย `On` และสิ้นสุดด้วยชื่อต่างๆ ของเมสเสจ `WM_` ของ Windows ที่มันต้องการตอบสนอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างเช่น ฟังก์ชันสมาชิก OnCreate() ถูกกระตุ้นเมื่อหน้าต่างได้รับเมสเสจ WM_CREATE ฟังก์ชันตัวจัดการเมสเสจทุกตัวเป็นแบบ protected เราไม่สามารถเข้าถึงยกเว้นในคลาสที่ถูกสร้างขึ้นมาจาก Cwnd ดังนั้นถ้าเรามีออบเจกต์ของ Cwnd ที่ชื่อ oCwnd แล้วเราสามารถเขียน oCwnd.ShowWindow() ณ ตำแหน่งไหนก็ได้เนื่องจาก ShowWindow() เป็นแบบ public แต่เราไม่สามารถเขียน oCwnd.OnCreate() ยกเว้นใน Cwnd หรือฟังก์ชันสมาชิกที่เป็นผลิตภัณฑ์ของ Cwnd

ฟังก์ชันสมาชิกของตัวจัดการเมสเสจของ Cwnd ถูกออกแบบมาเพื่อเป็นตัวจัดการธรรมดาซึ่งมักจะเรียกใช้งานโดย ::DefWindowProc() โปรแกรมประยุกต์ด้วย MFC ทั้งหมดจะสร้างขึ้นมาจาก CFrameWnd (หรือ CMDIFrameWnd) เพื่อให้พวกมันสามารถเตรียมตัวจัดการเมสเสจของมันเองได้ กฎเกณฑ์ทั่วไปคือ ตัวจัดการเมสเสจของคลาสที่ถูกสร้างขึ้นใหม่ควรจะเรียกตัวจัดการเมสเสจของคลาสพื้นฐานเพื่อเรียกใช้ตัวปฏิบัติการปกติทั่วไป ลักษณะแปลกอย่างหนึ่งของ MFC คือว่า ตัวจัดการเมสเสจ Cwnd ไม่เป็นความจริง (ไม่เสมือน) มีฟังก์ชันตัวจัดการเมสเสจ Cwnd มากมาย และมีผลิตภัณฑ์ของ Cwnd อยู่หลายชนิด การใช้ตารางเสมือนจำนวนมาก ทำให้แต่ละตัวมีทางเข้าเป็นร้อย และทำให้มีการเรียกซ้อนทับกัน 100 ครั้งด้วย แทนที่เราจะใช้โครงร่างของเมสเสจ ซึ่งแสดงรายละเอียดของตัวจัดการเมสเสจของคลาสแต่ละตัวที่ถูกเขียนทับอย่างชัดเจน

ใน Windows ปกติ เมสเสจแต่ละตัวจะมีพารามิเตอร์ 2 ตัวคือ wParam และ lParam พารามิเตอร์ทั้งสองตัวนี้เป็นส่วนที่แย่มากที่สุดในการออกแบบ Windows เนื่องจากพวกมันจัดการข่าวสารทั้งหมดเกี่ยวกับเมสเสจโดยไม่กำหนดชนิดข้อมูล และทำให้การเขียนโปรแกรมประยุกต์ยากขึ้น เนื่องจากทำให้เราติดนิสัยในการใช้แมโคร MAKELONG, HIWORD, LOWORD, LOBYTE และ HIBYTE ฟังก์ชันตัวจัดการเมสเสจ Cwnd มีความสะดวกและปลอดภัยกว่าสำหรับการเชื่อมต่อกับเมสเสจของ Windows เนื่องจากพวกมันเป็นส่วนเชื่อมต่อที่คำนึงถึงชนิดข้อมูลเป็นสำคัญ ตัวอย่างเช่น ฟังก์ชันสมาชิกของตัวจัดการเมสเสจ OnKeyDown() มีพารามิเตอร์คือ nChar, nRepCnt และ nFlags ภายใน OnKeyDown() นี้เราสามารถเข้าถึงส่วนนับการกดคีย์ซ้ำ (ค้าง) ได้โดยการใช้พารามิเตอร์ nRepCnt โดยที่เราไม่จำเป็นต้องใช้ฟังก์ชันของ Windows เพื่อบอกว่ามีอะไรเก็บไว้ใน LOWORD ของ lParam เมื่อเรามีการเขียนทับตัวจัดการเมสเสจ Cwnd ในคลาสที่ถูกสร้างขึ้นใหม่ของเรานั้นเราต้องระมัดระวัง 3 สิ่งเหล่านี้ สิ่งแรกคือ เราต้องมั่นใจว่าฟังก์ชันสมาชิกตัวเมสเสจใหม่ของเรา มีพารามิเตอร์และชนิดของผลลัพธ์เหมือนกับสิ่งที่มันเขียนทับ (แทนที่) อย่างแท้จริง วิธีที่ง่ายที่สุดคือใช้โปรแกรมเอดิเตอร์ (แก้ไข เพิ่มเติม) ตัวเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อักขระของเราลอกส่วนประกาศข้อมูลของตัวจัดการเมสเสจจาก `afxwin.h` มาปะ (วางสอดแทรก) ไว้ในโปรแกรมคำสั่งของเราเองสิ่งที่สองคือ ให้กล่าวถึงตัวจัดการ(handler) ที่ใช้แทนนั้นในโครงร่างเมสเสจของคลาสที่ถูกสร้างขึ้นใหม่ของเรา สิ่งที่สามคือ ให้ตรวจสอบว่าฟังก์ชันสมาชิก `Cwnd` ดั้งเดิมต้องถูกเรียกใช้งานภายในตัวจัดการที่ใช้แทนของเราหรือไม่ ตัวจัดการเมสเสจ `Cwnd` บางตัวไม่ได้ใช้งาน บางตัวจำเป็นและมีการเรียกใช้งานจากภายในตัวจัดการเมสเสจของเราเองด้วย เราสามารถพบข้อมูลเหล่านี้ได้จากคู่มือ `Class Libraries User's Guide` ของ MFC

3.4 โครงร่างเมสเสจต่างๆ

โครงร่างเมสเสจต่างๆ เป็นส่วนสำคัญของการประยุกต์ใช้งาน MFC โดยกำหนดว่าหน้าต่างโปรแกรมจะต้องตอบสนองต่อเมสเสจต่างๆ ของ Windows อย่างไร คลาส `Cwnd` เป็นตัวอย่างพื้นฐานของการจัดการเมสเสจด้วย MFC ตัวอย่างคลาสที่ถูกสร้างขึ้นมาจาก `Cwnd` เช่น `CModalDialog` มีโครงร่างเมสเสจของตัวเองซึ่งรับอาร์กิวเมนต์โครงร่างเมสเสจของตัวเองซึ่งรับอาร์กิวเมนต์โครงร่างเมสเสจ `Cwnd` ส่วนคลาสของไลบรารี MFC ซึ่งถูกสร้างขึ้นมาจาก `Cwnd` คือ `CModalDialog` และ `CMDIFrameWnd` เท่านั้นที่มีโครงร่างเมสเสจเป็นของตัวเอง ส่วน ผลผลิตอื่น ๆ ของ MFC ของ `Cwnd` จะใช้โครงร่างเมสเสจ `Cwnd` เป็นมาตรฐาน หากสร้างคลาสหน้าต่างใหม่จาก `CFrameWnd`, `Cdialog`, `CmodeDialog`, `CMDIFrameWnd` หรือ `CMDIChildWnd` แล้วเราจะต้องรวม(include) โครงร่างเมสเสจเข้ามาด้วยจากโครงร่างเมสเสจของโปรแกรมประยุกต์ Hello ที่ได้กล่าวไปแล้ว จะมีโครงร่างเมสเสจสี่ชนิดข้อมูลอยู่ 2 ตัว

```
BEGIN_MESSAGE_MAP ( CMainWindow, CFrameWnd )
    ON_WM_PAINT ( )
    ON_COMMAND ( IDM_ABOUT, OnAout )
END_MESSAGE_MAP ( )
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1 ทางเข้าสำหรับเมสเสจค่าคงที่

ทางเข้า (entry) แบบแรกคือเมสเสจค่าคงที่ของ Windows ทางเข้าเหล่านี้ง่ายต่อการใช้งาน โดยเราสามารถดูได้จากฟังก์ชันสมาชิก On...() ทุกตัวของคลาส Cwnd ซึ่งจะตรงกับทางเข้าโครงร่างเมสเสจ ON_... ที่กำหนดไว้ใน afxmsg.h การตั้งชื่อนี้จะคล้ายกันเสมอ สำหรับเมสเสจที่กำหนดไว้ใน Windows นั้นฟังก์ชันสมาชิก Cwnd ที่ตรงกันจะถูกตั้งชื่อให้ขึ้นต้นด้วย ON_ โดยไม่มีขีดเส้นใต้ ดังนั้นตัวจัดการของคลาส Cwnd สำหรับเมสเสจ WM_PAINT ก็คือ ฟังก์ชันสมาชิก OnPaint () และทางเข้าโครงร่างเมสเสจ Cwnd คือ แมโคร ON_WM_PAINT () การกำหนดทางเข้า ON_WM_PAINT () ในโครงร่างเมสเสจของคลาสหนึ่ง ๆ นั้นบอกให้ทราบว่าฟังก์ชันสมาชิก OnPaint () สำหรับคลาสนั้น โดยเฉพาะอย่างยิ่งฟังก์ชันสมาชิก OnPaint () ของเรานั้น ต้องมีพารามิเตอร์และค่าผลลัพธ์เหมือนกับฟังก์ชันสมาชิก OnPaint () ของ Cwnd ถ้าฟังก์ชันสมาชิกของเราถูกประกาศไม่ถูกต้องกล่าวเราจะพบกับความผิดพลาด (error) มากมายจากแมโครของทางเข้าเมสเสจที่กำหนดให้

3.4.2 ทางเข้าสำหรับเมสเสจ WM_COMMAND

ทางเข้าชนิดที่สองในโครงร่างเมสเสจของโปรแกรมประยุกต์ Hello คือทางเข้าตัวหนึ่งสำหรับเมสเสจ WM_COMMAND โดยที่ค่า wParam ของมันคือ IDM_ABOUT ส่วนใหญ่เมสเสจ WM_COMMAND ถูกสร้างขึ้นมาจากเมนู และคีย์เร่งความเร็วหรือเร่งด่วน(accelerator) ต่าง ๆ ตัวจัดการเมสเสจต่าง ๆ ต้องไม่มีอาร์กิวเมนต์ และไม่มีผลลัพธ์ เราสามารถใช้ตัวจัดการเมสเสจเพียงตัวเดียว ควบคุมดูแลเมสเสจที่เกี่ยวข้องทั้งหลายได้ส่วนที่เห็นได้ชัดของงานในหน้าที่นี้คือ กำหนดตัวจัดการที่เหมือนกันให้กับเมสเสจต่าง ๆ ตัวอย่างเช่นส่วนของโครงร่างเมสเสจซึ่งกำหนดให้ฟังก์ชันสมาชิก OnYscale() ควรจะถูกเรียกใช้งานสำหรับเมสเสจที่แยกออกจากกัน 4 ตัวคือ

ON_COMMAND (IDM_Gain_1, OnYscale)

ON_COMMAND (IDM_Gain_2, OnYscale)

ON_COMMAND (IDM_Gain_5, OnYscale)

ON_COMMAND (IDM_Gain_10, OnYscale)

ภายในฟังก์ชันสมาชิก OnYscale () นั้นเราต้องกำหนดให้หัวข้อเลือกของเมนู 4 ข้อถูก

เรียกใช้งานจริง ๆ เทคนิค(กลอุบาย)ก็คือใช้ฟังก์ชันสมาชิก GetCurrentMessage() ของ CWnd เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่งผลลัพธ์ซึ่งเป็นตัวชี้ข้อมูลไปยังโครงสร้าง(struct) MSG โดยโครงสร้าง MSG ถูกกำหนดไว้ใน windows.h ดังนี้

```
typedef struct tagMGS
{
    HWNG    hwnd;
    WORD    message;
    WORD    wParam;
    LONG    lParam;
    DWORD   time;
    POINT   pt;
}
```

รหัสคำสั่งใน OnYscale () ซึ่งอ่านค่าตัวเลือก ID จากเมนูที่กำหนดคือ

```
WORD iYscale = GetCurrentMessage ( ) ->wParam;
```

ค่าซึ่งคาดว่าจะได้รับสำหรับ iYscale คือ IDM_Gain_1, IDM_Gain_2, IDM_Gain_5 และ IDM_Gain_10

3.4.3 ทางเข้าสำหรับเมสเสจควบคุม

ทางเข้าชนิดที่สามของทางเข้าโครงสร้างเมสเสจ คือทางเข้าสำหรับเมสเสจควบคุมซึ่งมีอยู่ 4 ชนิดคือ

ON_EN_...	สำหรับการแก้ไขเพิ่มเติม
ON_BN_...	สำหรับกดปุ่ม
ON_LBN_...	สำหรับกรอบรายการ(List box)
ON_CBN_...	สำหรับกรอบแบบ Combo (Combo box)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมสเสจควบคุมที่กำหนดนี้สามารถทำงานสอดคล้องกับเมสเสจควบคุม ซึ่งแสดงอยู่ในหนังสืออ้างอิงของ Windows ซึ่ง แมโครของเมสเสจควบคุมที่กำหนดขึ้นสำหรับใช้ในโครงร่างเมสเสจต่าง ๆ นี้ จะแสดงไว้ใน afxmsg.h แมโครของเมสเสจควบคุมแต่ละตัวต้องการพารามิเตอร์ 2 ตัวคือ ID เดียวกับที่เราใช้ในไฟล์ซอร์ซโค้ดทรัพยากร(resource) ของเรา และชื่อของฟังก์ชันตัวจัดการเมสเสจ นั้น ๆ

ON_BN_CLICKER (IDD_FORTUNE, OnBtnFortune)

ON_LBN_SELCHANGE (IDD_FOOD, OnLbnFortune)

ฟังก์ชันสมาชิกของตัวจัดการเมสเสจที่ใช้ควบคุมนี้ ควรประกาศแบบไม่มีอาร์กิวเมนต์และไม่มีค่าผลลัพธ์

3.4.4 ทางเข้าสำหรับเมสเสจ Windows ซึ่งจดทะเบียนไว้

ทางเข้าโครงร่างเมสเสจชนิดที่สี่และเป็นชนิดสุดท้าย คือทางเข้าสำหรับเมสเสจของวินโดวส์ได้จดทะเบียนไว้แล้ว (registered Windows message) เราสามารถลงทะเบียนเมสเสจของวินโดวส์ใหม่ได้โดยการใช้ฟังก์ชัน ::RegisterWindowsMessage () ตัวจัดการสำหรับเมสเสจของผู้ใช้ควรประกาศในรูปของ ON_REGISTERED_MESSAGE () โดยการให้พารามิเตอร์ตัวแรกเป็นชื่อของตัวแปรแบบ WORD ซึ่งใช้เก็บผลลัพธ์ของฟังก์ชัน :: RegisterWindowsMessage () ซึ่งเป็นตัวแปรไม่ใช่ค่าคงที่ เนื่องจากเราไม่สามารถรู้ล่วงหน้าได้เลยว่า Windows จะกำหนดเมสเสจหมายเลขอะไร ส่วนพารามิเตอร์ตัวที่สองตัวเป็นชื่อของตัวจัดการเมสเสจ

ON_REGISTERED_MESSAGE (nMyMessage, OnMyMessage)

ตัวจัดการเมสเสจสำหรับเมสเสจของWindows ที่จดทะเบียนไว้ต้องมีพารามิเตอร์ 2 ตัวคือแบบ UINT และ LONG ซึ่งสอดคล้องกับค่า wParam และ lParam ซึ่งได้รับจาก Windows เหมือนอย่างเมสเสจ

3.4.5 สำหรับเมสเสจ Windows มาตรฐาน

ในทำนองเดียวกัน เราจะสร้างทางเข้าโครงร่างเมสเสจ เพื่อการจัดการเมสเสจของ Windows แบบมาตรฐานโดยการใช้แมโคร `ON_MESSAGE()` พารามิเตอร์ตัวแรกคือ ID ของเมสเสจนั้น และตัวที่สองเป็นชื่อฟังก์ชันสมาชิกของตัวจัดการซึ่งรับพารามิเตอร์ `UNIT` และ `LONG`

3.5 ระเบียบแบบแผนของการตั้งชื่อ MFC

กฎเกณฑ์ที่สำคัญของการสร้างไลบรารี MFC คือแบบแผนในการตั้งชื่อของมัน แบบแผนของการตั้งชื่อช่วยให้โปรแกรมของเราเข้าใจได้ง่ายขึ้น

กฎเกณฑ์ที่สำคัญของการตั้งชื่อ MFC

- ชื่อคลาสจะขึ้นต้นด้วย `C` และตามหลังด้วยอักษรตัวพิมพ์ใหญ่ ชื่อของคลาสนั้นๆ เช่น `CFrameWnd`, `Cwnd` โดย `C` ย่อมาจาก `Class` (คลาส)
- ชื่อฟังก์ชันสมาชิกจะขึ้นต้นด้วยตัวอักษรตัวพิมพ์ใหญ่ ชื่อฟังก์ชันนั้นๆ เช่น `InitInstance()`
- ชื่อตัวแปรสมาชิกจะขึ้นต้นด้วย `m_` และตามหลังด้วยสัญลักษณ์แบบฮังกาเรียน สำหรับกำหนดชนิดข้อมูล และตามหลังด้วยตัวอักษรตัวพิมพ์ใหญ่ ชื่อตัวแปรนั้น เช่น `m_pMainWnd` ส่วน `m_` ย่อมาจาก `member` (สมาชิก) และสมาชิกข้อมูลแบบสถิต (`static`) จะไม่ขึ้นต้นด้วย `m_`
- ค่าคงที่การแจ้งเตือนของคลาสนั้นๆ มักจะมีชื่อผสมกัน แต่จะขึ้นต้นด้วยอักษรตัวพิมพ์เล็ก เช่น คลาส `CFileException` ในไลบรารี MFC จะกำหนดค่าคงที่ `fileNotFound` และ `badPath`
- นอกจากนี้แล้ว `_` (ขีดเส้นใต้) ใน `m_` นั้น ควรหลีกเลี่ยงการใช้สัญลักษณ์ขีดเส้นใต้ในชื่อแสดงองค์ประกอบโปรแกรมของเรา แต่ให้ใช้ชื่อซึ่งแสดงความหมายขึ้นกันด้วยตัวพิมพ์ใหญ่แต่ละความหมายเช่นให้ใช้ `SetDigit()` ดีกว่า `set_digit()`

การตั้งชื่อเหล่านี้ไม่มีผลต่อการทำงาน แต่ทำให้เห็นได้ชัดเจน และทำความเข้าใจได้ง่ายเราควรมีกฎเกณฑ์เหล่านี้เมื่อมีการคลาสเพิ่มเติมของเราเองในไลบรารี MFC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.1 สัญลักษณ์แบบฮังการี (Hungarian)

สัญลักษณ์แบบฮังการีใช้กำหนดชื่อสมาชิกข้อมูลมากมาย เพื่อให้เข้าใจง่ายกันในหมู่นักเขียนโปรแกรมบน Windows การตั้งชื่อสัญลักษณ์แบบฮังการีจึงช่วยให้ผู้อ่านเข้าใจชนิดข้อมูลนั้นโดยไม่ต้องไปค้นดูในส่วนประกาศดั้งเดิม สัญลักษณ์แบบฮังการีเป็นตัวอักษรตัวเดียวหรือสองตัวซึ่งนำหน้าชื่อของตัวแปรเพื่อแสดงชนิดข้อมูลของมัน ตัวอย่างเช่น `h_` ในชื่อ `m_hInstance` แสดงว่าเป็น HANDLE (ตัวจัดการ) ในตารางที่ 3.1 มีสัญลักษณ์แบบฮังการีส่วนใหญ่ที่นิยมใช้กันโดยทั่วไป

ขึ้นต้นด้วย	ชนิดของข้อมูลภายใน เดิม	ขึ้นต้นด้วย	ชนิดข้อมูล MFC	ขึ้นต้นด้วย	ชนิดข้อมูล Windows
p	ตัวชี้ (pointer)	by	BYTE	h	HANDLE
lp	ตัวชี้ระยะไกล (far pointer)				
r	ตัวอ้างอิง (reference)	n	UNIT		
fn	ฟังก์ชัน (function)	w	WORD		
c	char	l	LONG		
n	int	dw	DWORD		
sz	string	b	BOOL		

1. สตริงที่ปิดท้ายด้วย `NULL(0)`

ตารางที่ 3.1 แสดงการกำหนดตัวขึ้นต้นด้วยสัญลักษณ์แบบฮังการี

สัญลักษณ์แบบฮังการี ใช้กันอย่างกว้างขวางในโปรแกรมภาษา C บน Windows แต่ใช้ไม่ค่อยได้ผลมากนักในชนิดของคลาส C++ เนื่องจาก C++ มีชนิดข้อมูลจำนวนมากมายมหาศาลและ C++ มีชนิดข้อมูลที่เข้มงวดในการตรวจสอบมากกว่า C ซึ่งทำให้ความต้องการสัญลักษณ์แบบฮังการีลดน้อยลง เราอาจยังคงใช้สัญลักษณ์แบบฮังการีบางตัวอยู่ แต่ทว่าในการประกาศออบเจกต์ `Cstring` ต้องขึ้นต้นด้วย `str` เป็นต้น และการใช้ออบเจกต์ C++ ให้มีชื่อเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สั้นเพียงพอต่อความเข้าใจโดยใช้สัญลักษณ์แบบฮังกาเรียนเพื่อถ่ายทอดการจำ เช่นให้ขึ้นต้นด้วย p (ตัวชี้), r (ตัวอ้างอิง) และ fn (ฟังก์ชัน) และขึ้นต้นด้วยชนิดข้อมูลภายในดั้งเดิม และ MFC



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

เมนู และไดอะล็อกบ็อกซ์บน Windows ด้วยไลบรารี MFC

การทำงานทางด้านกราฟฟิก เป็นเส้นหนึ่งของการเชื่อมต่อผู้ใช้ด้านกราฟิกบน Windows (GUI) ซึ่งมีส่วนเชื่อมต่อกับผู้ใช้มีคล้าย ๆ กันเสมอ เช่นโปรแกรมประยุกต์บน Windows เกือบทุกตัวมีเมนูบาร์เป็นฟังก์ชันหลัก ในทำนองเดียวกันโปรแกรมประยุกต์บน Windows ส่วนใหญ่มีไดอะล็อกบ็อกซ์ ซึ่งให้เราสามารถเลือกลักษณะบางอย่างของโปรแกรมได้ การสร้าง และจัดการเมนู กับไดอะล็อกบ็อกซ์ทำให้การเขียนโปรแกรมบน Windows มีลักษณะตรงไปตรงมาและดูดีขึ้น จากบทเรียนที่ผ่านมา ได้กล่าวถึงเครื่องมือหลายอย่างที่จะช่วยให้เราสามารถสร้างรีซอร์สของเมนูกับไดอะล็อกบ็อกซ์ได้ และรูปแบบต่าง ๆ ของไฟล์รีซอร์สบน Windows (ไฟล์ที่นามสกุล .bmp , .ico, .rc เป็นต้น) ค่อนข้างที่จะใช้ง่ายนักเขียนโปรแกรมโดยส่วนใหญ่ลงมติเป็นเอกฉันท์ว่า เมนูและไดอะล็อกบ็อกซ์บน Windows สามารถดำเนินการได้ง่ายกว่า โปรแกรมประยุกต์บน MS-DOS ในคุณลักษณะที่เหมือนกัน

ปัญหาหลักของกระบวนการภาษา C คือ การจัดการเมนู และไดอะล็อกบ็อกซ์มีความสลับซับซ้อนและต้องการโปรแกรมคำสั่งจำนวนมาก หากต้องการให้เมนูมีทางเลือก 50 ข้อ นั้นหมายความว่าเราต้องมีประโยคคำสั่ง switch ของภาษา C 50 case (กรณี) ซ้อนกันอยู่ภายใน และยังมีขนาดใหญ่มากขึ้นอีกถ้าต้องการให้ครอบคลุมเหมือนส่วนจัดการเมสเสจของ Windows อื่น ๆ ด้วย ซึ่งเป็นผลทำให้โปรแกรมมีขนาดใหญ่เทอะทะ และเกิดความผิดพลาดได้ง่ายไลบรารี MFC มีโครงร่างแบบออบเจกต์โอเรียนเท็ด สำหรับจัดการเมนูและไดอะล็อกบ็อกซ์ต่าง ๆ ซึ่งสะดวกสบาย และง่ายกว่าการเขียนด้วยภาษา C และภายในไลบรารีนี้มีฟังก์ชันสมาชิกคลาสซึ่งทำหน้าที่สำหรับการเลือกเมนูแต่ละตัว การเลือกกลุ่มของเมนูควบคุมไดอะล็อกบ็อกซ์ หรือควบคุมกลุ่มของไดอะล็อกบ็อกซ์ สำหรับรองรับการเขียนโปรแกรมโดยทั่วไป

ไลบรารี MFC เข้ากันได้กับเครื่องมือสำหรับการออกแบบโปรแกรมบน Windows ทั้งหลายซึ่งช่วยในการสร้างไฟล์รีซอร์สของเมนู และไดอะล็อกบ็อกซ์ เราสามารถสร้างรีซอร์สของเมนูและไดอะล็อกบ็อกซ์โดยใช้เครื่องมือเหล่านี้ได้ แล้วใช้คลาสต่าง ๆ ของไลบรารี MFC จัดการรีซอร์สเหล่านั้น ในบทนี้จะอธิบายรายละเอียดการทำงานนี้ แต่อย่างไรก็ตามเราจะไม่ลงไปรายละเอียด แนวความคิดของรีซอร์สของโปรแกรมไฟล์รีซอร์สแต่ละตัว และกลไกในการสร้างรี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซอร์สของเมนูหรือไดอะล็อกบ็อกซ์ กฎเกณฑ์พื้นฐานเหล่านี้สามารถหาอ่านได้ใน “the Windows SDK reference manual” และ “Charles Petzold's Programming Windows”

4.1 คลาส CMENU

คลาส CMENU ของไลบรารี MFC เป็นตัวกำหนดลักษณะภายในของกระบวนการต่างๆ บน Windows โดยใช้ตัวจัดการ HMENU Cmenu เหมือนตัวห่อหุ้มบาง ๆ คือฟังก์ชัน Cmenu ส่วนใหญ่ไม่ได้ทำงาน หรือแทบจะไม่ทำงานด้วยตัวพวกมันเองจริง ๆ เพียงแต่ไปเรียกใช้ฟังก์ชันของ Windows อีกต่อหนึ่ง การทำงานแบบเรียกต่อไม่ได้หมายความว่าไม่มีค่าข้อมูลเลย Cmenu มีการตรวจสอบชนิดข้อมูลอย่างเข้มงวดในโปรแกรมคำสั่งของเรา ทำให้เราไม่ต้องใช้พารามิเตอร์ HMENU ซึ่งผ่านไปยังฟังก์ชันเมนูของ Windows ส่วนใหญ่ และยังเป็นโครงร่างการจัดการงานที่สำคัญในคลาส MFC อื่น ๆ ส่วนใหญ่นั้นออบเจกต์ Cmenu ถูกสร้างเป็น 2 แบบ แบบแรกคือตัวผู้สร้างของ C++ ดั้งเดิมซึ่งมีหน้าที่ไม่ค่อยมากนัก ส่วนแบบที่สองเป็นกระบวนการที่มีฟังก์ชันที่สามารถตั้งค่าเริ่มต้นได้สมบูรณ์ในตัว ส่วนช่วยเหลือของตัวผู้สร้างชนิดที่สองทำให้เราสามารถตั้งค่าเริ่มต้นออบเจกต์ Cmenu ที่ซึ่งเรามีตัวจัดการ (handle) HMENU อยู่แล้ว เพื่อโหลดเมนูจากรีซอร์ส หรือสร้างเมนูแบบใหม่ขึ้นมาได้ด้วยตัวทำลายโครงสร้างเมนูมี 2 แบบ เราสามารถออบเจกต์เมนูของเราเองได้โดยง่ายซึ่งจะปล่อยอิสระให้กับรีซอร์สของระบบรวมทั้งเมนูนั้น ๆ ด้วย และจะลบออบเจกต์ของมันเองด้วย แต่เราต้องระวังเกี่ยวกับการลบออบเจกต์ Cmenu ส่วนใหญ่ เนื่องจากเราไม่สามารถลบออบเจกต์ Cmenu ที่เราได้มาจากระบบ และเราไม่สามารถลบออบเจกต์ Cmenu ซึ่งเกี่ยวข้องกับเมนูบาร์ในกรณีที่เราไม่สามารถลบออบเจกต์ Cmenu นั้นเราสามารถเรียก DestroyMenu() เพื่อปล่อยอิสระให้กับรีซอร์สระบบของเมนูนั้น ๆ ได้ ซึ่งเป็นสิ่งจำเป็นอย่างยิ่ง เช่น หลังจากที่เราแทนที่ (สร้าง) เมนูหนึ่ง ๆ ในเมนูบาร์

4.2 ฟังก์ชันเมนูของคลาส Cwnd

คลาส Cwnd ที่เราเห็นตรงกึ่งกลางหน้าต่างของส่วนเชื่อมต่อ Windows มีสิ่งอำนวยความสะดวกสำหรับเมนูหลายอย่าง สิ่งอำนวยความสะดวกเหล่านี้ ทำให้เราสามารถจัดการและเปลี่ยนแปลงเมนูต่าง ๆ ได้ และสามารถทำงานเกี่ยวกับออบเจกต์ Cmenu เพื่อจัดการกับเมนูบาร์ต่าง ๆ ฟังก์ชันสมาชิก Cwnd มูลฐาน 2 ตัวสำหรับเมนูคือ GetMenu() และ GetSystem Menu() เพื่ออ่านค่าตัวชี้ไปยังออบเจกต์เมนูระดับบนสุดของโปรแกรมประยุกต์ และออบเจกต์เมนูของระบบตรงกันข้ามกับ GetMenu () คือ SetMenu () ซึ่งแทนที่เมนูระดับเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

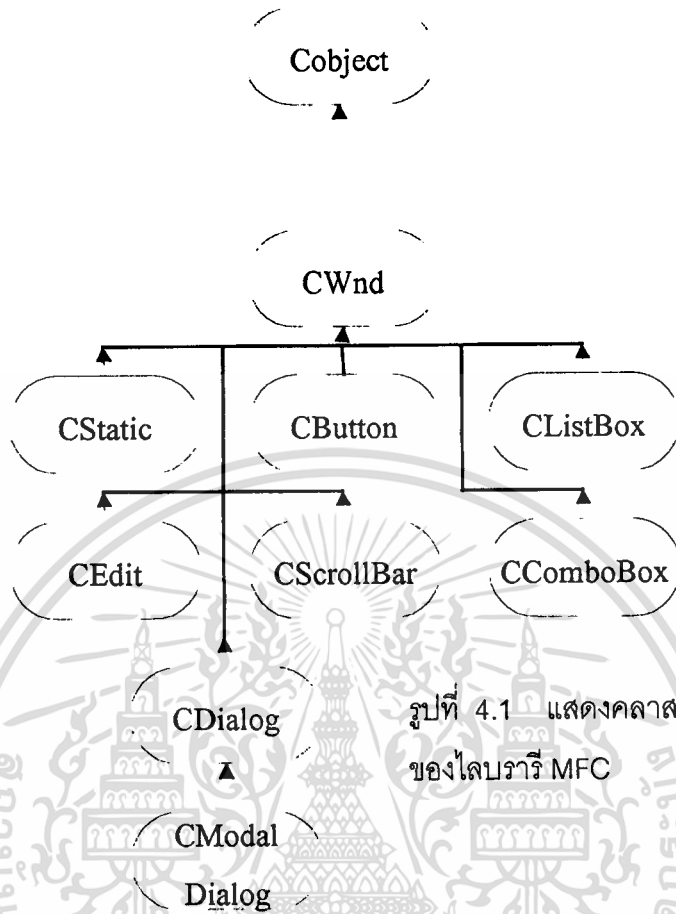
บนสุดของโปรแกรมประยุกต์เมื่อเมนูระดับบนสุดของโปรแกรมถูกเปลี่ยนไป บางที่เราต้องเรียก ฟังก์ชัน DrawMenuBar () ของ Cwnd เพื่อวาดเมนูบาร์ใหม่

ลักษณะอีกสองอย่างของ Cwnd ที่เกี่ยวกับเมนู สิ่งแรกคือ ตัวจัดการเมนูสำหรับเมสเสจที่เกี่ยวข้องกับเมนู เช่น ฟังก์ชัน OnMenuChar (), OnMenuSelect (), OnInitMenu () และ OnInitMenuPopup () ตัวจัดการ OnMenuChar () ใช้ในกรณีที่ผู้ใช้ต้องการเปิดเมนูและกดแป้นตัวอักษรเรียกเมนูเร่งด่วน (ตัวเร่งความเร็ว) โดยที่ตัวอักษรที่ต้องการกดนั้นไม่เข้ากันกับตัวเร่งความเร็วของเมนูตัวอักษรที่กำหนด ก็ยังสามารถใช้ตัวอักษรเร่งด่วนนั้นเรียกเมนูที่ต้องการได้ ฟังก์ชันสมาชิก OnMenuSelect () บอกให้เราทราบว่ามีการเลือกเมนูนั้นแล้ว มันทำให้เราสามารถควบคุมการเลือกเมนูต่าง ๆ ได้ ส่วนฟังก์ชันสมาชิก OnInitMenu () และ OnInitMenuPopup () ทำหน้าที่แจ้งให้โปรแกรมรู้ว่า เมื่อมีการเข้าถึงเมนูครั้งแรกให้โปรแกรมทำการปรับปรุงเมื่อนั้นก่อนที่จะแสดงขึ้นมา หน้าอื่น ๆ ของ Cwnd ในระบบเมนูเกี่ยวข้องกับส่วนเชื่อมต่อเอกสารหลายแบบ (Multiple Document Interface หรือ MDI) คลาส CMDIFrameWnd ถูกสร้างขึ้นมาจาก Cwnd มีฟังก์ชันสมาชิก MDISetMenu ซึ่งทำหน้าที่หลายอย่างในส่วนบริหารงานเมนูอย่างดีของโปรแกรมประยุกต์ของ MDI

4.3 คลาสไดอะล็อกของ MFC

การเขียนโปรแกรมแบบออบเจกต์โอเรียนเต็ลมีข้อดีบางอย่างมากกว่าแบบอื่น การเขียนโปรแกรมแบบออบเจกต์โอเรียนเต็ลเหมาะสำหรับสภาวะแวดล้อมแบบ GUI ตัวอย่างเช่น Windows โดยสภาวะแวดล้อมของมันอยู่ในรูปของออบเจกต์ ซึ่งส่งเสริมและสนับสนุนการใช้งานแบบออบเจกต์โอเรียนเต็ล และความสลับซับซ้อนของสภาวะแวดล้อม GUI ถูกบริหารเป็นลำดับชั้นอย่างดีที่สุด แต่ถ้าเราต้องการเทคนิคการเขียนโปรแกรมแบบออบเจกต์โอเรียนเต็ลแล้วอาจไม่ต้องสนใจ GUI ส่วนการใช้ไดอะล็อกของ GUI นั้นไดอะล็อกบ็อกซ์ใช้คุณสมบัติทางออบเจกต์ได้อย่างเหมาะสมเท่าที่เราได้พบมา ไบบริารี MFC มีคลาสมูลฐาน 2 ตัวสำหรับการควบคุมไดอะล็อกบ็อกซ์ คือ Cdialog สำหรับไดอะล็อกบ็อกซ์ที่ไม่มีรูปแบบ (modeless dialog boxes) และ CModalDialog สำหรับไดอะล็อกบ็อกซ์ที่มีรูปแบบ (modal dialog) คลาส Cdialog ถูกสร้างขึ้นมาจาก Cwnd และ CModalDialog ถูกสร้างขึ้นมาจาก Cdialog สำหรับไดอะล็อกที่มีรูปแบบง่าย ๆ เช่น ไดอะล็อก About เราสามารถใช้ CModalDialog ได้โดยตรง แต่สำหรับไดอะล็อกที่มีรูปแบบส่วนใหญ่และไดอะล็อกบ็อกซ์ที่ไม่มีรูปแบบทุกตัวนั้น เราจะสร้างคลาสของเราเองขึ้นมาจาก CmodalDialog หรือ Cdialog ขึ้นมา ซึ่งนี่คือสิ่งที่เราไม่ควรทำ ไม่ว่าจะวิธีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควบคุมกระบวนการสร้างไดอะล็อกบ็อกซ์ คลาส Cdialog และ CModalDialog ไม่ได้ลดบทบาทของตัวแก้ไขเพิ่มเติม (editor) ไดอะล็อกบ็อกซ์สำหรับการออกแบบไดอะล็อกบ็อกซ์หนึ่ง ๆ คลาสไดอะล็อกทั้งสองตัวจัดหาวิธีการสร้างออบเจกต์ไดอะล็อกบ็อกซ์ วิธีตั้งค่าเริ่มต้นให้ส่วนควบคุมก่อนที่ไดอะล็อกบ็อกซ์จะแสดงเป็นครั้งแรก และจัดหาวิธีเลื่อนประเด็นของส่วนควบคุมจากจุดหนึ่งไปยังอีกจุดหนึ่งโดยปกติคลาสไดอะล็อกบ็อกซ์ที่เราสร้างขึ้นมาจะมีชุดของฟังก์ชันตัวจัดการเมสเสจหนึ่งชุดซึ่งตอบสนองเมสเสจของตัวควบคุมงานต่าง ๆ ทำให้เราสามารถจัดการไดอะล็อกบ็อกซ์ได้ด้วยตัวเอง โดยปกติ คลาสไดอะล็อกบ็อกซ์ที่สร้างขึ้นมีสมาชิกข้อมูลซึ่งเกี่ยวข้องกับตัวควบคุมด้วยในบางครั้ง สมาชิกข้อมูลเป็นแบบสถิต (static) เพื่อให้สามารถเข้าถึงได้แม้ว่าออบเจกต์ไดอะล็อกบ็อกซ์นั้นถูกลบไปแล้วก็ตาม โดยปกติไดอะล็อกบ็อกซ์ทำงานด้วยตัวควบคุม ตัวควบคุมแต่ละตัวเป็นชนิดข้อมูลแบบหนึ่งของหน้าต่างโปรแกรมซึ่งรู้หน้าที่การทำงานต่าง ๆ ของมันเองเป็นอย่างดี ไลบรารี MFC มีคลาสตัวควบคุม 6 ตัว โดยที่แต่ละตัวถูกสร้างขึ้นมาจาก Cwnd เพื่อจัดการเชื่อมต่อไดอะล็อกบ็อกซ์แบบออบเจกต์โอเรียนเท็ด เช่น Cstatic, Cedit, Cbutton, CListBox, CComboBox และ CScrollBar ในรูปที่ 4.1 แสดงคลาสควบคุมและไดอะล็อกของไลบรารี MFC ที่เกี่ยวข้องกับคลาสพื้นฐาน Cwnd



รูปที่ 4.1 แสดงคลาสควบคุม และไดอะล็อก
ของไลบรารี MFC

เราอาจต้องการให้คลาสไดอะล็อกบ็อกซ์พิเศษของเราแต่ละตัวมีออบเจกต์ควบคุมหลายตัวและเราอาจต้องการให้ไดอะล็อกบ็อกซ์ About มีออบเจกต์ Cbutton เพื่อบริหารการใช้ปุ่ม OK แต่อะไรจะเกิดขึ้นจริง ๆ เมื่อ Windows สร้างตัวควบคุมไดอะล็อกบ็อกซ์บนพื้นฐานของแบบไดอะล็อกบ็อกซ์ของเรา แล้วเราสร้างโครงสร้างออบเจกต์แบบแปรผันเพื่อบริหารตัวควบคุมเหล่านั้น ตัวอย่างเช่น ถ้าเราต้องการให้ออบเจกต์ CListBox มีการตอบโต้กับตัวควบคุมของกรอบแสดงรายการ (list box) เราสามารถสร้างมันได้โดยการเรียก GetDlgItem () ของ Cwnd .

ไดอะล็อกบ็อกซ์หนึ่ง ๆ มีประโยชน์หลายอย่าง เช่น สามารถเรียกความสนใจมารวมกันที่จุดเดียวกันได้ และโดยมากมันจะเป็นหน้าต่างมาตรฐานซึ่งมีหน้าต่างควบคุมที่เป็นตัวลูกอีกทีหนึ่ง ดังนั้น Cwnd จึงมีฟังก์ชันสมาชิกเป็นโหล ๆ เพื่อให้เราสามารถตอบโต้กับตัวควบคุมของไดอะล็อกบ็อกซ์ได้ เรามักนำไปใช้งานได้ 2 วิธี ตัวอย่างเช่น ถ้าไดอะล็อกบ็อกซ์หนึ่งมีตัวควบคุมปุ่ม (กด) แล้วเราสามารถใส่ฟังก์ชันสมาชิก CheckDlgButton () ของ Cwnd เพื่อตรวจสอบการกดปุ่มนั้นหรือเราสามารถใส่ GetDlgItem () สร้างออบเจกต์ Cbutton เพื่อบริหารตัวควบคุมการกดปุ่มแล้วฟังก์ชันสมาชิก SetCheck () ของออบเจกต์ Cbutton

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void CDrinkDialog::OnDone();
{
    Cmenu *menu = GetParent()->GetMenu();
    menu->EnableItem(IDM_DRINK,MF_BYCOMMAND|
MF_ENABLED);
    delete this;
}

```

4.4 คลาส CModalDialog

ไดอะล็อกบ็อกซ์ที่มีรูปแบบนิยมใช้กันมากกว่าไดอะล็อกที่ไม่มีรูปแบบ พวกมันให้ผู้ใช้ทำการเลือกได้ทันทีก่อนที่จะทำงานอย่างอื่นต่อไป เราได้พิจารณาไดอะล็อกที่มีรูปแบบง่าย ๆ ไปแล้วตัวหนึ่งคือไดอะล็อก About ในโปรแกรมประยุกต์ต่าง ๆ ได้ใช้ ออบเจกต์ CModalDialog ถูกสร้างแล้วเรียกใช้ฟังก์ชันสมาชิก DoModal () วิธีง่าย ๆ ของไดอะล็อกคือมีเฉพาะปุ่ม OK และ Cancel เท่านั้นเนื่องจากคลาส CModalDialog มีตัวจัดการ OnOk () และ Cancel () สำหรับสองปุ่มเหล่านั้นเรียบร้อยแล้ว โดยตัวจัดการทั้งสองตัวนี้จะถูกใช้งานก็ต่อเมื่อเราใช้ IDควบคุมมาตรฐานสำหรับปุ่มเหล่านี้ คือ IDOK และ IDCANCEL แต่ไดอะล็อกที่รูปแบบส่วนใหญ่มักถูกสร้าง โดยการสร้างคลาสไดอะล็อกเฉพาะจากคลาส CModalDialog ไดอะล็อกเลือกรายการอาหารในโปรแกรม Food ถูกบริหารโดยคลาส CFoodDialog ซึ่งกำหนดไว้ใน dialog.h ไม่เหมือนกับไดอะล็อกบ็อกซ์ที่ไม่มีรูปแบบ ในขณะที่โดยปกติแล้วไดอะล็อกบ็อกซ์ที่มีรูปแบบถูกสร้างขึ้นมาด้วยซีริออริส และส่วนจัดการหน้าต่างหลัก (ส่วนพ่อแม่) ซึ่งยอมให้เราสร้างโครงสร้างไดอะล็อกซึ่งมีการตั้งค่าเริ่มต้นแบบเต็มรูปแบบ ทำให้สามารถแสดง และประมวลผลโดยการเรียกใช้ฟังก์ชันสมาชิก DoModal () นี่เป็นตัวผู้สร้างของคลาส CFoodDialog

```

CFoodDialog (Cwnd* pWndParent = NULL)
: CModalDialog ("FoodDialog", pWndarent)
{ }

```

ตัวผู้สร้าง CFoodDialog ควรจะจัดการตัวชี้ไปยังหน้าต่างพ่อแม่ของมัน และเรียกใช้ตัวผู้สร้างของคลาสนี้พื้นฐาน CModalDialog เพื่อสร้างออบเจกต์ที่กำหนด

คลาสไดอะล็อกบ็อกซ์ที่ถูกสร้างขึ้นมามีหลายตัว มีสมาชิกข้อมูลซึ่งแสดงสถานะบางอย่างของส่วนควบคุม เราสามารถบริหารข้อมูลไดอะล็อกได้หลายวิธี แต่มีวิธีหนึ่งที่ผิดอย่างเห็นได้ชัด เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คือวิธีของภาษา C คือการเก็บข้อมูลข่าวสารแสดงสถานะของไดอะล็อกไว้ในตัวแปรหลัก (global) วิธีซึ่งเก็บข้อมูลข่าวสารแสดงสถานะการควบคุม ทำให้ออบเจ็กต์ไดอะล็อกใช้ได้ตลอด เวลาการทำงานของโปรแกรม เพื่อให้ตัวแปรสมาชิกของออบเจ็กต์ไดอะล็อกใช้งานได้ตลอด อีกวิธีหนึ่งคือทำให้ตัวแปรสมาชิกของไดอะล็อกเป็นแบบสถิต (static) เพื่อให้ตัวแปรสมาชิกเหล่านั้น ใช้งานได้แม้ว่าจะไม่มีออบเจ็กต์ไดอะล็อกนั้น เราสามารถดู ไฟล์ dialog.h ว่าคลาส CFoodDialog ใช้วิธีที่สอง เนื่องจากคลาส CModalDialog ถูกสร้างขึ้นมาจาก Cdialog มัน จึงมีฟังก์ชันสมาชิกแบบเสมือนที่ชื่อ OnInitDialog () สำหรับตั้งค่าเริ่มต้นให้ส่วนควบคุมต่าง ๆ ก่อนที่ไดอะล็อกจะแสดงบนหน้าจอ นี่เป็นฟังก์ชัน OnInitDialog () จากคลาส CFoodDialog



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

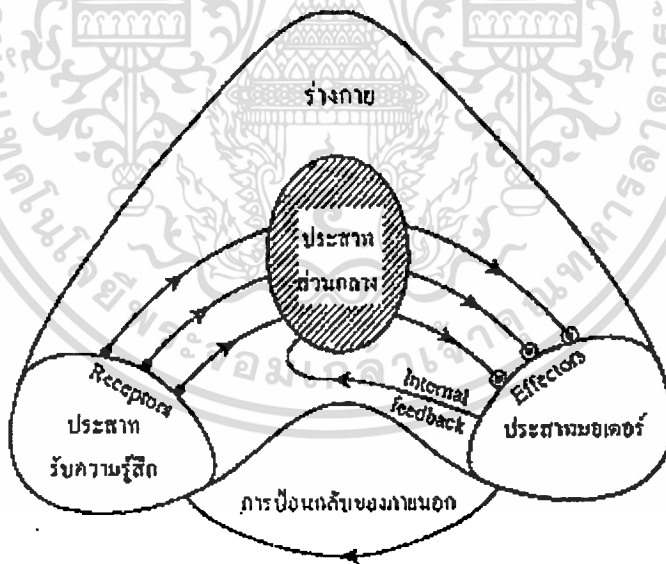
บทที่ 5

โครงข่ายประสาทเทียมเบื้องต้น

5.1 ความรู้พื้นฐานทางกายภาพ ของระบบเซลล์ประสาท

ระบบประสาทมีหน้าที่ควบคุมรักษาความสมดุลภายในร่างกายให้คงที่ ระบบประสาทของมนุษย์ เป็นระบบที่มีโครงสร้างที่ซับซ้อนซึ่งประกอบไปด้วยเซลล์ประสาท (Neural) ประมาณ 10^{11} เซลล์ และเซลล์ประสาทแต่ละเซลล์จะเชื่อมต่อกับเซลล์ประสาทตัวอื่นอีกประมาณ 10^4 เซลล์ ระบบประสาทมีอยู่ 3 ส่วน คือ

- ระบบประสาทส่วนกลาง (Central Nervous System)
- ระบบประสาทรับความรู้สึก (Sensory Systems)
- ระบบประสาทมอเตอร์ (Motor System s)



รูปที่ 5.1 แสดงการสื่อสารข้อมูลในระบบ

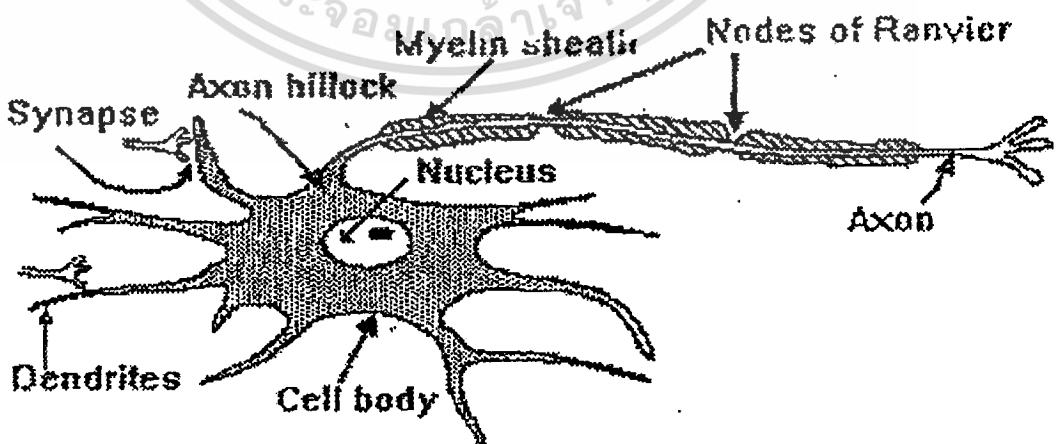
5.2 หน้าที่แต่ละส่วนของระบบประสาท

ระบบประสาทส่วนกลาง ทำหน้าที่ วิเคราะห์ข้อมูลต่าง ๆ ที่รับเข้าจากระบบประสาทรับความรู้สึก ซึ่งทำหน้าที่รับการเปลี่ยนแปลงของสิ่งแวดล้อมรอบตัว และจะส่งข่าวสารไปยังระบบเอกซอร์นเป็นเอกซอร์นที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประสาทส่วนกลาง เมื่อระบบประสาทส่วนกลางประมวลผล แล้วก็จะส่งสัญญาณ ไปควบคุมล้อม
สั่งการระบบประสาทมอเตอร์ของอวัยวะแสดงผล (Effector) ให้ทำงาน ซึ่งจะทำให้เกิดการตอบ
สนองต่อการเปลี่ยนแปลง ทั้งภายใน และภายนอก



รูปที่ 5.2 แสดงโครงสร้าง และการส่งกระแสของเซลล์ประสาท

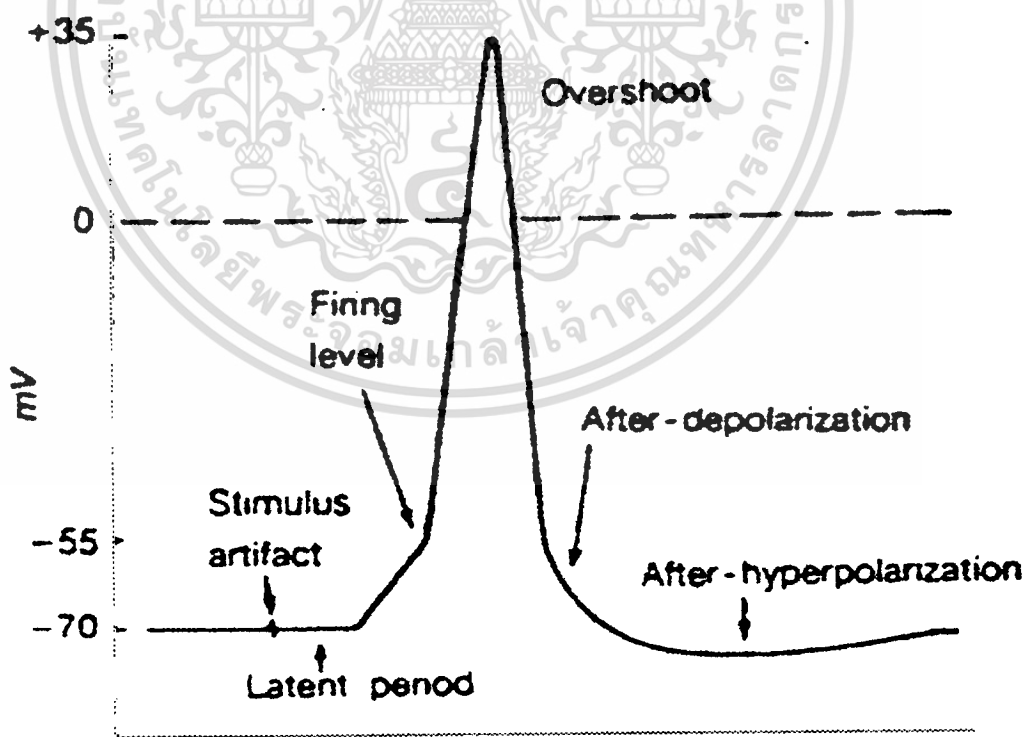


รูปที่ 5.3 แสดงโครงสร้าง และส่วนประกอบของเซลล์ประสาท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซลล์ประสาทแต่ละเซลล์ จะประกอบด้วย ตัวเซลล์ (Cell Body) และแขนงที่แยกออกจากตัวเซลล์ ซึ่งได้แก่ แอกซอน (Axon) และ เดนไดรน์ (Dendrite) แอกซอนมีจุดเริ่มต้นที่บริเวณส่วนที่หนาที่สุดของระบบประสาท ที่เรียกว่า Axon Hillock เซลล์ประสาทแต่ละเซลล์ จะมีแอกซอนเพียงหนึ่งอันปลายของแอกซอน จะแยกออกอีกเป็นหลายเส้น และสิ้นสุดที่ Terminal bottons และที่จุดสิ้นสุดนี้จะมีตัวรับกระแส (Nerve Impluse) ที่เรียกว่า Receptor Zone ซึ่งอยู่กับเดนไดรต์ของเซลล์ ประสาทตัวอื่น กระแสประสาท หรือข้อมูลที่อยู่ในรูปของการเปลี่ยนแปลงทางไฟฟ้าเคมี จะไปเป็นตัวกระตุ้นให้ ตัวเซลล์ส่งสัญญาณไปแอกซอน และเดนไดร จัดทำหน้าที่รับกระแส อื่นเข้าสู่เซลล์

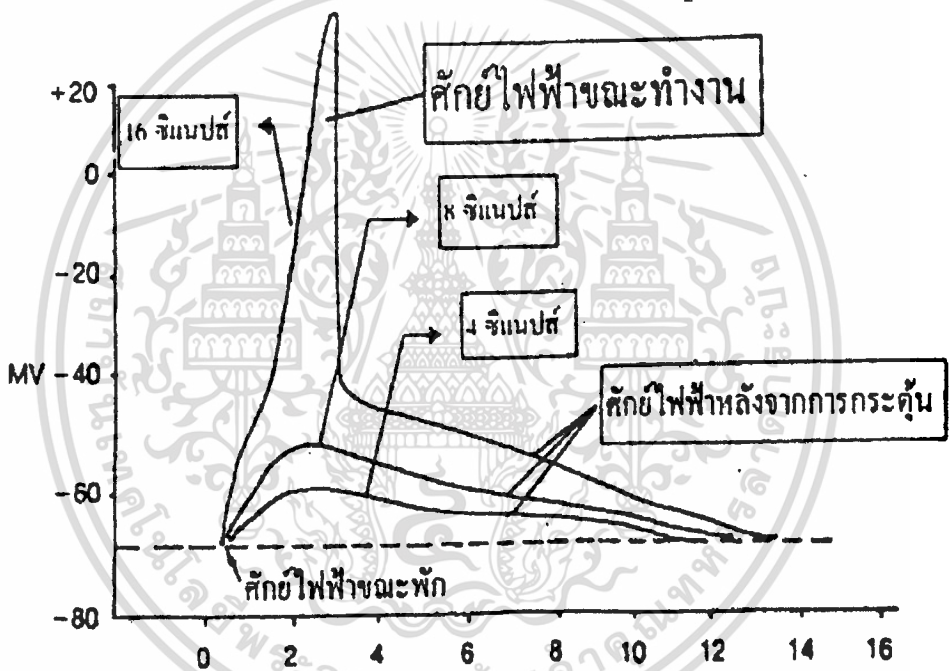
คุณสมบัติเซลล์ประสาทโดยทั่วไป คือ จะมีความไวต่อการถูกกระตุ้น เมื่อเซลล์ประสาทถูกกระตุ้น จำทำให้เกิดการเปลี่ยนแปลง สมดุลย์ของไอออนในบริเวณ reception Zone ซึ่งจะทำให้เกิดความแตกคักย์ และเมื่อความต่างคักย์ ของจุดที่สูงกว่าค่าเทรชโฮลด์ ก็จะทำให้เกิดการส่งสัญญาณกระแสประสาทไปยังเซลล์อื่น ๆ



รูปที่ 5.4 แสดงลักษณะการเกิดคักย์ไฟฟ้าขณะทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าความต่างศักย์ของเยื่อหุ้มเซลล์ประสาทขณะพัก ประมาณ -60 ถึง -70mV มีการกระตุ้นเซลล์ประสาทจะมีการแก้โพลาไรท์ (Depolarization) ของเยื่อหุ้ม และหลังจากการเกิดโพลาไรท์จนความต่างศักย์เพิ่มขึ้นเป็น 15mV . จะทำให้เกิดการแก้โพลาไรท์เพิ่มขึ้นอย่างรวดเร็ว ซึ่งจุดนี้เราเรียกว่า Firing Level เมื่อความต่างศักย์เพิ่มขึ้นอย่างสูงสุด ($\approx 35\text{mV}$) จะทำให้การแก้โพลาไรท์ลดลงอย่างรวดเร็ว ($\approx 70\%$) และจะค่อย ๆ ลดลงกลับเข้าสู่สภาวะปกติแล้วความต่างศักย์ จะลดต่ำกว่าระดับปกติซึ่งเรียกช่วงนี้ว่า After Hyperpolarization ลำดับของการเปลี่ยนแปลงความต่างศักย์ทั้งหมดนี้เราเรียกว่า ศักย์ไฟฟ้าขณะทำงาน (Action Potential)



รูปที่ 5.5 แสดงศักย์ไฟฟ้าขณะทำงานที่เกิดจากการรวมกันของซินแนปส์

กระแสประสาทจะส่งผ่านจากเซลล์หนึ่งไปยังอีกเซลล์หนึ่ง จะต้องผ่านรอยต่อระหว่าง Terminal Bottom และ Receptor Zone ซึ่งเราจะเรียกรอยต่อนี้ว่า ซินแนปส์ (Synapse) การส่งผ่านกระแสประสาทกันระหว่างเซลล์ (Neurotransmission) ซินแนปส์จะปล่อยสารเคมีเพื่อทำการกระตุ้น (Excitatory) สำหรับบางกรณี และบางกรณีก็ปล่อยสารเคมีเพื่อทำการยับยั้ง (Inhibitory) และในขณะใดขณะหนึ่งซินแนปส์จะยอมให้เกิดขึ้นเพียงกรณีเดียวเท่านั้น คือ ถ้าไม่มีการปล่อยสารเคมีเพื่อกระตุ้น ก็จะปล่อยสารเคมีเพื่อยับยั้ง การส่งกระแสประสาท จะต้องอาศัย ซินแนปส์ ประมาณ 10 ถึง หลายร้อยตัวในเวลาเดียวกันโดยขบวนการ Sumation ซึ่งซินแนปส์เพียงตัวเดียวจะไม่สามารถกระตุ้นเซลล์ประสาทให้ถึงสภาวะ Firing Level ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประมาณ 10 ถึง หลายร้อยตัวในเวลาเดียวกันโดยขบวนการ Sumation ซึ่งซิแนปส์เพียงตัวเดียวจะไม่สามารถกระตุ้นเซลล์ประสาทให้ถึงสภาวะ Firing Level ได้

ในสภาวะกระตุ้นจะทำให้เกิดการส่งผ่านสัญญาณกระแสประสาท ไอออนของโซเดียม (a^+), โพแทสเซียม (K^+) และ ครอไรด์ (Cl^-) และจะซึมสู่เยื่อหุ้มเซลล์ได้มากขึ้น ส่วนในสภาวะยับยั้ง โพแทสเซียม และครอไรด์ จะไหลออกมาจากเซลล์ และโซเดียมไม่มีการไหลเข้าเซลล์ ซึ่งเป็นสาเหตุให้ประจุบวกภายในเซลล์ลดลงทำให้ความต่างศักย์ภายในเซลล์เป็นบวกมากขึ้น (ประมาณ -75 mV) ซึ่งจะเป็นผลให้เซลล์ประสาทถูกกระตุ้นยากกว่าปกติ คือ ต้องใช้ความต่างศักย์เพิ่มมากขึ้นเป็น 16 mV ซึ่งจะสามารถกระตุ้นเซลล์ประสาทให้ถึง Firing Level ได้ซึ่งระยะนี้จะเรียกว่าระยะการดื้อ (Refractory Period)

5.3 คุณสมบัติ และความสามารถที่เป็นประโยชน์ ของโครงข่ายประสาทเทียม

โดยปกติการแก้ปัญหาต่าง ๆ ด้วยโปรแกรมคอมพิวเตอร์นั้นจะเริ่มต้นที่การพิจารณาอินพุต แล้วออกแบบขบวนการที่จะดำเนินการกับอินพุตเหล่านั้นเพื่อจะได้สร้างเอาต์พุตได้ตามที่ต้องการ วิธีการพัฒนาโปรแกรมคอมพิวเตอร์แบบคลาสสิกนี้มีข้อจำกัด คือ ไม่สามารถใช้ได้กับปัญหาซึ่งขบวนการ (Process) มีความซับซ้อนอย่างมากจนเราไม่สามารถสรุปเป็นขั้นตอนที่ชัดเจนได้ โครงข่ายประสาทเทียมเป็นรูปแบบในการแก้ไขปัญหาประเภทดังกล่าว โครงข่ายประสาทเทียมจะสร้างขบวนการ (Process) ขึ้นมาเองจากความรู้ ซึ่งได้มาจากการเรียนรู้จากตัวอย่าง (Learning by Examples) ทั้งนี้ตัวอย่างที่นำมาสอนให้กับโครงข่ายประสาทเทียมมีจำนวนมากมายพอบ และวิธีการสอนดีพอ ขบวนการที่ถูกสร้างขึ้นก็จะมีคุณภาพดีเพียงพอสำหรับการนำไปใช้งานจริงได้

คุณสมบัติ และความสามารถที่เป็นประโยชน์ ของโครงข่ายประสาทเทียม

1. มีความยืดหยุ่นสูงจนสามารถจำลองขบวนการของปัญหาใด ๆ
2. มีความสามารถในการจำชุดของอินพุต - เอาต์พุต ที่มีความซับซ้อนมากจนไม่สามารถจำลองแบบในเชิงความน่าจะเป็นได้
3. มีความสามารถในการปรับตัวเข้ากับการเปลี่ยนแปลงของแวดล้อม
4. มีความสามารถในการตอบสนองต่อข้อมูลที่ไม่เคยเห็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ความรู้จะกระจายอยู่ทั้งโครงสร้างของ โครงข่ายประสาทเทียม ทำให้การดำเนินการกับสารสนเทศเชิงอรรถาธิบาย (Contextual Information) ได้อย่างธรรมชาติ

5.4 การจัดประเภทของโครงข่ายประสาทเทียม

ตำราหลายเล่มจัดหมวดหมู่โครงข่ายประสาทเทียมแตกต่างกัน พอสมควร และพอจะมีวิธีการแบ่งอยู่ 2 วิธี คือ

5.4.1 แบ่งตามลักษณะการใช้งาน

- ชนิดระบุกลุ่ม (Classification)
- ชนิดเกี่ยวข้องกัน (Association)
- ชนิดจัดตัวเอง (Selt - Organization)
- ชนิดหาค่าที่เหมาะสม (Optimiation)

5.4.2 แบ่งตามลักษณะวิธีการเรียนรู้

- แบบปรับค่าน้ำหนักโดยมีผู้สอน (Supervised Learnig)
- แบบปรับค่าน้ำหนักโดยไม่มีผู้สอน (Unsupervised Learnig)
- แบบค่าน้ำหนักไม่เปลี่ยน (Fixed Weight)

5.5 แนวคิดเบื้องต้น

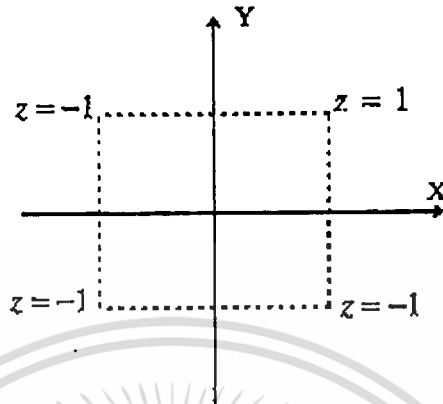
เราลองเริ่มต้นกับปัญหาแบบพื้นฐาน ๆ เช่น การจำลองตรรก “ และ ” (Logic “ AND “) ซึ่งกำหนดได้ดังตารางที่ 5.1 ทั้งที่เราจะใช้ค่า 1 แทนค่าจริง “จริง” และ -1 แทนค่า “เท็จ”

X	Y	Z
1	1	1
1	-1	-1
-1	1	-1
-1	-1	-1

ตารางที่ 5.1 แสดงค่าตรรก “ และ ”

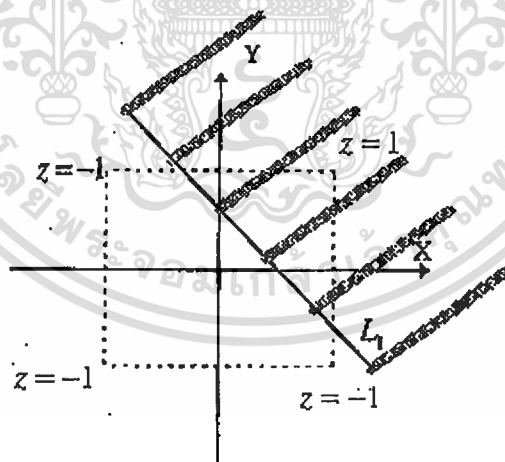
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางเราทดลองค่าความสัมพันธ์เชิงเรขาคณิตวิเคราะห์ได้ด้วยกราฟที่ 5.1



รูปกราฟที่ 5.1 แสดงความสัมพันธ์เชิงเรขาคณิตวิเคราะห์ของตรรก " และ "

จากรูปกราฟในรูปกราฟที่ 5.1 จากต้องแบ่งกลุ่มคำตอบออกเป็น $Z = 1$ และ $Z = -1$ วิธีง่าย ๆ คือใช้เส้นตรงแบ่งพื้นที่ในการออกเป็น 2 ส่วน ดังรูปกราฟที่ 5.2



รูปกราฟที่ 5.2 แสดงเส้นตรงแบ่งพื้นที่แก้ปัญหาของตรรก " และ "

นั่นคือ จุดที่อยู่เหนือเส้นตรง L_1 บนระนาบ $X-Y$ ให้ค่า $Z = 1$ ซึ่งเราอาจจะเขียนสมการทั่วไปของเส้นตรง L_1 ได้ในรูปของ $a_1X + b_1Y + c_1 = 0$

ดังนั้นเราจะได้ว่า $Z = 1$ เมื่อ $a_1X + b_1Y + c_1 \geq 0$

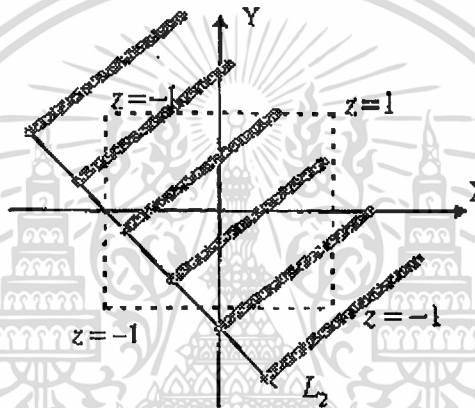
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$Z = -1 \text{ เมื่อ } a_1X + b_1Y + c_1 < 0$$

ถ้าปัญหาเปลี่ยนไป เช่น เป็นการจำลองตรรก “หรือ” (logic “OR”) เราจะได้ออกมาดังรูปกราฟที่ 5.3 ซึ่งในทำนองเดียวกันเราอาจแบ่งพื้นที่ออกเป็นเขต $Z = 1$ และ $Z = -1$ ได้โดยใช้เส้น L_2 ซึ่งจะมีรูปของสมการโดยทั่วไปของ $a_2X + b_2Y + c_2 = 0$

$$\text{ดังนั้นเราจะได้ว่า } Z = 1 \text{ เมื่อ } a_2X + b_2Y + c_2 \geq 0$$

$$Z = -1 \text{ เมื่อ } a_2X + b_2Y + c_2 < 0$$



รูปกราฟที่ 5.3 แสดงเส้นตรงแบ่งพื้นที่แก้ปัญหของตรรก “หรือ”

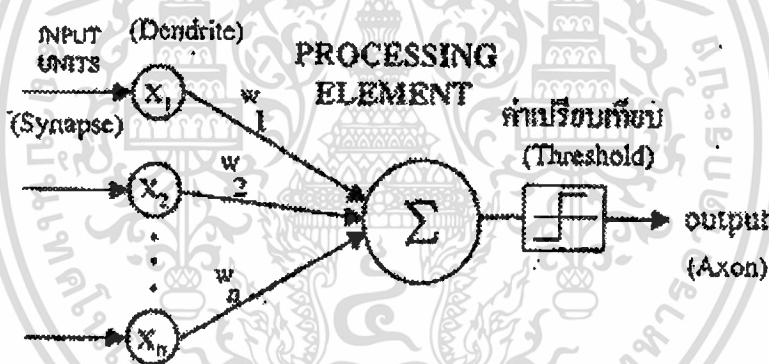
จะเห็นว่าสมการเส้นตรงเราสามารถมีเส้นตรงทั่วไป $L : aX + bY + c = 0$ ให้เป็นสมการที่ 1 และมีระบบในการปรับแต่งค่า a , b และ c ให้เท่ากับ a_1 , b_1 และ c_1 ได้เมื่อเราต้องการจำลองตรรก “และ” และให้เท่ากับ a_2 , b_2 และ c_2 ตามลำดับได้ เมื่อเราต้องจำลองตรรก “หรือ” ก็หมายความว่า เราสามารถสอนเส้นตรง L นั้นให้เรียนรู้ตรรก “และ” หรือตรรก “หรือ” ได้

ขั้นตอนการสอนโครงข่ายประสาทเทียมนั้นจะเป็นการเรียนรู้จากตัวอย่าง ดังนั้นเราต้องการวิธีปรับค่า a, b และ c โดยใช้ตัวอย่างจุด (X, Y, Z) ในการสอน จากหลักการดังกล่าว เราสามารถจำลองของหน่วยประมวลผลระดับพื้นฐานของโครงข่ายประสาทเทียม หรือนิวรอน (Neural)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.6 แบบจำลองทางคณิตศาสตร์ของหนึ่งหน่วยเซลล์ประสาท

โครงสร้างของเซลล์ประสาทแบบจำลอง ประกอบด้วยชั้นอินพุต (Dendrites) มีหน้าที่รับเอาท์พุทที่ได้จาก นิวรอนตัวอื่น ผ่านเข้าไปยังเซลล์ โดยที่มีปริมาณความเข้มข้นของสารเคมีที่บริเวณจุดเชื่อมโยงจะถูกแทนด้วยค่าของเลขทศนิยม ซึ่งเรียกว่าน้ำหนัก (weight) อินพุทของเซลล์ประสาทจะเป็นได้ทั้งค่าบวก และค่าลบ อินพุทที่เป็นบวกจะไปกระตุ้น หรือเร้าให้เกิดเอาท์พุท ส่วนอินพุทจะเป็นลบจะไปห้าม หรือยับยั้งไม่ให้เกิดเอาท์พุท พลังงานที่เป็นอินพุทบวก และลบ จะไปรวมกันถ้ามากเกินกว่าค่าที่กำหนด ค่าหนึ่งซึ่ง เรียกว่า ค่าพิทิด (Threshold Value) ก็จะทำให้เกิดเอาท์พุท



รูปที่ 5.6 แสดงแบบจำลองทางคณิตศาสตร์ประสาทของ McCulloch และ Pitts

แบบจำลองของเซลล์ ประสาทแสดงดังรูปที่ 5.6 อินพุท X_i (สำหรับ $i = 1, 2, 3, n$) จะมีค่าเป็น $\{0, 1\}$ ซึ่งขึ้นอยู่กับสัญญาณอินพุท

บทที่ 6

การเกิดของโครงสร้างเซลล์

(Growing Cell Structure)

การงอกของ โครงสร้างเซลล์ ในโครงข่ายแบบ Self-Organizing ทั้งการเรียนรู้แบบ การสอนแบบชี้นำข้อมูล หรือดูแล (Supervised Learning) และแบบ การสอนแบบไม่มีข้อมูลชี้นำ หรือไม่มีการดูแล (Unsupervised Learning) การแสดงถึง Self-Organizing Neural Network Model แบบใหม่นั้นมี 2 แบบที่แตกต่างกัน คือ

1. การสอนแบบไม่มีข้อมูลชี้นำ หรือไม่มีการดูแล โดยที่เราสามารถใช้ ข้อมูลที่มองเห็น เป็น Clustering และ Vector Quantization เพราะมีประโยชน์กว่าแบบที่เป็นอยู่ รูปแบบของโครงข่ายสามารถที่จะหาสถานะของโครงสร้าง และขนาด วิธีนี้ถูกควบคุม ผ่านทางขบวนการงอกของเซลล์ และการหลุดออกของเซลล์
2. การสอนแบบมีข้อมูลชี้นำ หรือมีการดูแล ที่ผ่านเป็นผลของการรวมกันของ Above-Mentioned Self-Organizing Network ด้วย Radial Basis Function (RBF) วิธีการแบบอาศัยข้อมูลเทียบเคียง เพื่อดูความ ผิดพลาด การปฏิบัติในวิธีการของ RBF ต้องใช้ น้ำหนักแบบคู่ขนาน การแยกประเภทสิ่งต่าง ๆ ก็ใช้ความผิดพลาดที่เกิดขึ้นเป็นตัว กำหนด หรือใช้ RBF เข้าไปทดสอบได้

6.1 ความรู้เบื้องต้น (Introduction)

Self-Organizing Neural Network Model ผู้ที่คิด คือ Willshaw , Von der Malsburg และ Kohonen เป็นวิธีการแมปปีง (Mapping) จากมิติสูงมาเป็นมิติที่ต่ำกว่า (High-Dimensional mapping into Lower-Dimensional) ทั้งหมดเป็นหลักการต่อกันของโครงสร้าง วิธีการแบบนี้สามารถยังคงรักษาสมาชิกที่อยู่ใกล้เคียงไว้ได้ (Neighborhood) ความสัมพันธ์อินพุท และสิทธิในการครอบครองอยู่แสดงได้ถึงขอบเขต จาก ความเข้มข้นสัญญาณที่มีค่าสูง (High Signal Density) ซึ่งส่วนนี้เป็นส่วนที่มากที่สุดของพื้นที่ส่วนใหญ่ ของโครงข่าย

การกระทำแบบนี้เป็นที่น่าสนใจยิ่งนัก เพื่อนำไปประยุกต์กับงานหลาย ๆ ด้านได้ เช่น การจดจำเสียงพูด และการบีบอัดข้อมูล ต่อการรวมกันที่ก่อให้เกิดประโยชน์สูงสุด เหมือนกับการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แม่ปิ้งที่สามารถพบได้ในหลาย ๆ ที่ เช่น ในสมองของมนุษย์ และสัตว์ ที่ชี้ให้เป็นการต่อกันของ โครงสร้างสมอง และเพิ่มหลักของ ระบบประมวลสัญญาณ (Signal Processing System)

สิ่งที่เราพิจารณาก็คือ โครงสร้าง และขนาดของ รูปแบบ Kohonen ที่มีความหมายเกี่ยวกับข้อจำกัดที่มีผลต่อ การแม่ปิ้ง บ่อยครั้งที่เราสมมุติเพื่อให้ข้อมูลเป็นจริง ในเรื่องของความแตกต่างของข้อมูล รูปร่าง จำนวนของส่วนที่ประกอบกัน (Element) ค่าที่เหมาะสม สำหรับใน ส่วนอื่นคือ การให้ความสำคัญต่อข้อมูล ที่หามาได้และยอมให้สถานะของโครงข่ายมีขนาด และรูปร่าง ที่ได้เปรียบ กว่าที่เป็นอยู่

วิถีทางที่จะพิจารณารูปร่าง รวมทั้งขนาดที่มีผลต่อโครงข่าย เป็นปัญหาอย่างมาก เพราะระหว่างที่สถานะนั้นมีการยอมให้มีการเพิ่มได้ นี่คือหลักที่บอกว่ายอมให้เกิดระหว่างที่เกิดขบวนการต่อโครงข่ายได้ แสดงให้เห็นถึงความยืดหยุ่น ปัญหาอยู่ที่ โครงสร้างตัวแปรของตัวเลข ที่บอกถึง Element แสดงการกำหนด จำนวนมิติ ของรูปแบบการต่อรูปแบบของการกำหนดค่า K (K -dimention) สามารถกำหนดได้

ต่อไปเป็นการแสดงถึงโครงสร้างในโครงข่ายแบบไม่มีข้อมูลชี้นำ หรือไม่มีการดูแล และจะขยายไปถึง การสอนแบบมีข้อมูลชี้นำ หรือมีการดูแลต่อไป

6.2 การเกิดของเซลล์แบบไม่มีข้อมูลชี้นำ หรือไม่มีการดูแล

6.2.1 ขอบเขตของปัญหา

ก่อนที่เราจะกำหนด หรือกล่าวรายละเอียดของรูปแบบว่าอะไรนั้น เราได้ประมาณค่าความแม่นยำ กำหนดชนิดของปัญหา ของโครงข่าย เพื่อที่เราจะหาทางแก้ปัญหานั้นได้ สิ่งแรกคือการกำหนดค่า มิติที่เราจะต้องหา โดยระบุสัญญาณอินพุท ที่ไม่ทราบ Probability Distribution $P(\xi)$ และมี $V = R^2$ ที่เป็น อินพุท เวกเตอร์ ของสัญญาณ ที่กำหนดมา เราจะต้องทำการแม่ปิ้งจาก V เป็น Discrete K -Dimentional ให้อยู่ในรูปการต่อแบบโครงข่ายของ A โดยที่วิธีการแม่ปิ้งจะต้องมีคุณสมบัติดังนี้.-

- อินพุทเริ่มต้นที่เหมือนกันต้องแม่ปิ้งเป็น Close Element ของ A
- การต่อแบบ Close Element A ตรวจสอบที่สัญญาณอินพุทที่เหมือนกัน เพื่อที่จะทำให้สัญญาณถูกแม่ปิ้งได้

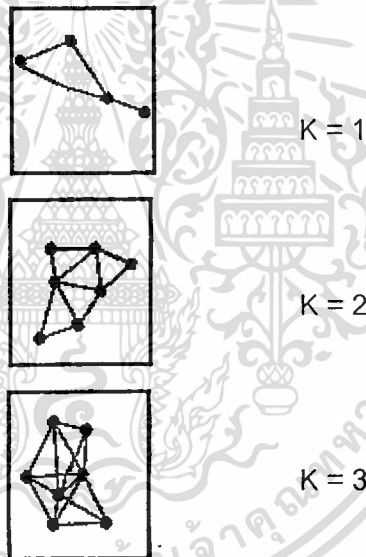
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขอบเขตของ V ซึ่งเราจะประมาณความเข้มของการกระจายของ เวกเตอร์ที่อินพุท คือ มีมากพอที่จะตรงมาก ๆ ในกลุ่ม Element A

ครั้งที่หนึ่ง มี 2 จุด คือ การเฝ้าป้องกันความสัมพันธ์ในทิศทางแบบเดินหน้า และแบบถอยหลัง (Forward and Backward) ถ้า A มีมิติที่น้อยกว่า V มันจะลดการกระทำลงทั้ง ๆ ที่การกระทำที่หาความสัมพันธ์ของข้อมูลที่ยังเป็นอยู่ไม่มีข้อมูล ส่วนจุดที่ 3 คือเราจะได้รับข้อมูลที่เราไม่ทราบค่า คือ $(P(\xi))$ ของสัญญาณอินพุท

6.2.2 สถาปัตยกรรมของโครงข่าย (Network Architecture)

เริ่มจากการต่อกันของโครงข่าย A ที่มีมิติ หรือค่าของ K -Dimemntional ต่าง ๆ ดังนี้.-



รูปที่ 6.1 แสดงโครงสร้างเซลล์ที่มีมิติต่างกัน

ถ้าเรากำหนดค่าของมิติให้มีค่าเท่ากับ 2 เราจะได้ Vertices ของมันมีค่าเป็น Vertices = $K+1$ อันเป็นหลักง่าย ๆ ของเซลล์ หรือนิวรอนและการหาเซลล์ที่อยู่ใกล้เคียง แบบความสัมพันธ์ ได้ดังนี้ $(K+1) K / 2$ ในขณะที่ Self-Organizing กำลังประมวลผลอยู่นั้น มันยอมให้เกิดเซลล์ใหม่ แล้วยังยอมให้เซลล์หลุดออกได้อีกด้วย ทุก ๆ ครั้งที่มันเปลี่ยนแปลง อย่งไรก็ตามมันจะต้องปรับตัวของมันเองตลอดเวลา ในส่วนของค่า K -Dimentional อีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

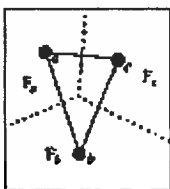
ถ้าเราต้องการให้รูปแบบของเราแสดงได้หลาย ๆ มิติ เราก็สามารถแสดงได้ด้วยค่าน้อย ๆ ได้เป็นลักษณะของการรวมกันของโครงสร้างใหญ่ ๆ โดยที่เราจะพิจารณา จำนวนของ vertices เป็นอย่างแรก ของค่ามิติ ที่มีมิติมากๆ ให้เป็นแบบ ลิเนียร์ เท่านั้นด้วยการกำหนดค่า K อย่างเช่นค่าของ K ที่มีการสร้างก่อให้เกิด Hypercube แล้วมีการสร้างจำนวนของ Vertices (2^K) ดังนั้นจุดที่เราจะนำมาพิจารณาที่มีมิติมาก ๆ ที่มีใน ทุก ๆ High-Dimensional Neural Network ทุก ๆ เซล c มีจำนวน n -dimensional ของ Synaptic Vector w_c ครอบงำโดยที่ c มีตัวเวกเตอร์เป็นตัวแยกตำแหน่งของ C ในอินพุต เวกเตอร์ w_{ij} , $i \in A$ โดยที่การ แมปปีง ทุก ϕw จาก Input Vector space V เป็นโครงข่ายของ A เราสามารถกำหนดโดยการแมปปีง ของ Input signal ที่เข้ามาต่อเซลล์ ด้วยตำแหน่งที่ใกล้ที่สุด หรือที่เรียกว่า เวกเตอร์

$$\phi w : V \rightarrow A, (\xi \in V) \rightarrow (\phi w(\xi)) \in A$$

โดยที่ $\phi w(\xi)$ เราเรียกว่า Best-Matching Unit ซึ่งเป็นการกำหนด ϕw

$$\|w_{\phi w(\xi)} - \xi\| = \min_{r \in A} \|w_r - \xi\|$$

ซึ่งทฤษฎีของ Euclidean Vector ในที่มี V คือส่วนในจำนวนของขอบเขต F_1 ($i \in A$) แต่ละ Location จะมีจุดวงกลมอยู่ใกล้ ๆ เราเรียกว่า synaptic Vector



Voronoi tessellation ที่สร้างมาจาก 2 มิติ เริ่มจากการสร้างสามเหลี่ยม

รูปที่ 6.2 แสดง Voronoi tessellation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ทุก ๆ อินพุตที่เข้ามาจะโปรเจกต์ ให้เป็นรูปตัว v โดยการวาดวงกลมในการอ้าง เวกเตอร์ที่เป็นจุดที่ต้องการ วงกลมจะบอกถึงตัวที่เป็นอยู่ใกล้ที่สุด และนิรอนจะต่อกันโดยลากเส้น Input Space V มีขนาดที่ใหญ่แต่เป็น finite ของ Subarea ของ R^n ผลลัพธ์ของ Finite Input Space ดังนั้นก็คือ Voronoi region ของ Finite แต่ถ้าเป็นทุก ๆ จุดมันจะไม่ถูกเพราะค่าที่ยื่นออกมา

6.3.3 โครงข่ายแบบไดนามิกส์ (Network Dynamics)

เป็นหลักการของการปรับตัวของ Synaptic Vector เช่นโมเดลของเรา คนที่เป็นผู้กำหนดวัตถุประสงค์ คือ Kohonen ประกอบด้วย

- พิจารณาให้เป็น Best-matching Unit สำหรับ Current Input signal
- เพิ่มการ Best-matching Unit และการต่อกับตัวที่มันอยู่ใกล้ที่สุดโมเดลแบบนี้มีการกำหนดกลยุทธ์โดยที่ความสำคัญอยู่ 2 อย่างคือ
 - การปรับตัวเอง เป็นการให้ความถี่ในเวลา ระบุเป็นพารามิเตอร์ ของ ϵ_0 และ ϵ_n สำหรับ Best-Matching Unit เพื่อหาตัวที่อยู่ใกล้ที่สุด
 - สำหรับ Best-Matching Unit เท่านั้น และตัวที่มีการต่อกันโดยตรง เป็นตัวที่ปรับ ทั้งหมดนี้เราได้จำกัด Cooling schedule ออกไปทั้งหมด

N_c = เซตของการต่อแบบตรงของตัวที่อยู่ข้างเคียง C และกำหนดให้มี local Counter Variable T_c เมื่อมีอินพุตเข้ามาเพื่อให้เป็น Best-Matching Unit เพราะว่า เซลจะหมุนรอบ ๆ โดยมีน้ำหนักกำกับอยู่ การลดค่าก็ต้องกระทำเพราะค่าที่แสดงอยู่ต้องเป็นค่าที่มีอยู่จริง

ขั้นตอนการปรับของโมเดลสามารถคำนวณได้ดังนี้.-

- เลือกสัญญาณอินพุต ξ ให้ตรงกับ ค่า Probability distribution $P(\xi)$
- Best-Matching Unit ให้ $S = \phi_w(\xi)$
- เพิ่มค่าที่จะให้ matching สำหรับ S เมื่อมีต่อกันโดยตรง

$$\Delta W_s = \epsilon_b (\xi - w_s)$$

$$\Delta W_s = \epsilon_n (\xi - w_c) \quad (\text{for all } c \in N_s)$$

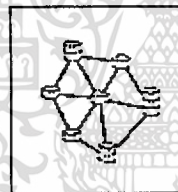
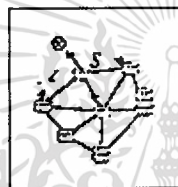
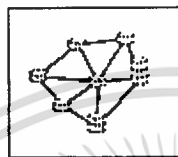
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เพิ่มสัญญาณในการนับ

$$\Delta \tau_c = 1.$$

- ลดค่าสัญญาณการนับ

$$\Delta \tau_c = -\alpha \tau_c \text{ (for } | \epsilon_A)$$



รูปที่ 6.3 การแสดงการปรับตัวเป็นสเต็ป
ของโครงข่ายแบบ 2 มิติ

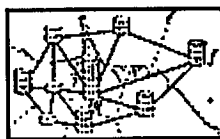
ค่าที่เราเลือกค่า ϵ_b และ ϵ_n ให้มีค่าน้อยเซลล์ที่มากมาจากการสุ่มตำแหน่งตอนเริ่มต้น และจุดที่ต้องการต้องเป็นจุดที่พอใจ ทั้งหมดเป็นการเปลี่ยนทิศทางจุดประสงค์ก็คือโครงสร้างที่ประกอบด้วย Synatic Vector w_c การกระจายให้ตรงต่อ $P(\xi)$ ทำให้ทุก ๆ เซลล์เริ่มการ Best-Matching Unit สำหรับ Current Input Vector ถ้าเราไม่รู้ค่าของ $P(\xi)$ แต่อย่าลืมว่าค่าของ Counter มันจะกำหนดตามนี้ และชื่อของมันจะสัมพันธ์กับความถี่ ของอินพุทที่เข้ามา

Relation Signal Frequency ของเซลล์ C คือ

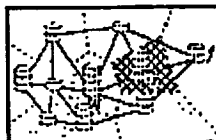
$$h_c = \tau_c / \sum_{i \in a} ij$$

ถ้าค่า h_c มีค่ามาก แสดงว่าตำแหน่งที่ดีที่สุดที่จะเพิ่มเซลล์เข้ามา แต่การเพิ่มนั้น เป็นเหมือนการลดค่าลงดังรูป 6.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก่อนเพิ่ม



หลังเพิ่ม

รูปที่ 6.4 แสดงการเพิ่มเซลล์

ที่ซึ่งมีการเพิ่มกฎเกณฑ์เข้าไป หลังจากที่เรากำหนดค่า λ ให้ค่าคงที่ การปรับตัวที่มีเสถียรของ เซลล์ q ประกอบด้วย

$$h_q \geq h_c \text{ (for all } c \in A \text{)}$$

ตัวที่มีการเชื่อมต่อกับ q โดยตรงจะมีค่าไกลสุดใน Input Space คือ เซลล์ f

$$\|w_f - w_q\| \geq \|w_c - w_q\| \text{ (for all } c \in N_q \text{)}$$

เวลาเราเพิ่มเซลล์ r ระหว่างเซลล์ q และ f แล้วเซลล์ r ก็จะต่อกับเซลล์อื่น ๆ และมีการสร้างโครงข่ายอีกครั้งหนึ่งที่ประกอบด้วย K -Dimensional ดังนั้นค่า Synaptic Vector ของ r จะเพิ่มจาก

$$w_r = 0.5 (w_q - w_f)$$

การเพิ่มของเซลล์ r ก่อให้เกิด Voronoi region F_r ใน Input Space ในเวลาเดียวกันตัวที่อยู่ใกล้ r มันจะลดน้อย

การเปลี่ยนแปลงจะเป็นไปตามค่า τ_c ที่คำนวณการเปลี่ยนแปลงได้จาก Signal Counter คือ

$$\Delta \tau_c = \frac{|F_c^{(new)}| - |F_c^{(old)}|}{|F_c^{(old)}|} \tau_c \text{ (for all } c \in n_r \text{)}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนี้

เมื่อ $|F_c| = N - \text{Dimentional}$ ค่าที่กำหนดครั้งสุดท้ายของเซลล์ใหม่กำหนดได้

$$\tau_r = - \sum_{c \in nr} \Delta \tau_c$$

การกระจายค่า Counter สามารถใช้เหตุผลการอธิบายด้วยเซลล์ใหม่ที่มี input signal เข้ามาเรามี Algorithm ในการสร้างเซลล์ดังนี้.-

Start with a K- Dimensional simplex at random positions in $V = R^n$
While (desired network size not reached)
Perform a constant number λ of adaptation steps.
Insert a new cell and distribute the counter variable according to eqn. 10 - 11.

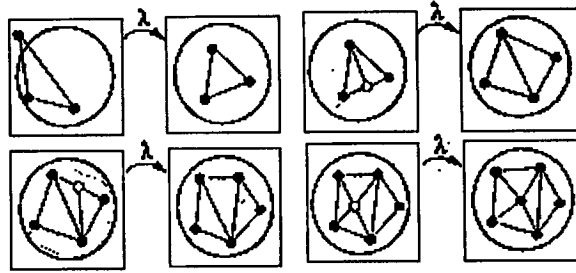
รูปที่ 6.5 อัลกอริทึมเบื้องต้นของการกำหนดโครงสร้างเซลล์

และคุณลักษณะหลักของ Model นั้นมีการปรับหลาย ๆ ครั้ง ตามสัญญาณที่เข้ามา เราสามารถกระทำในลักษณะ Feedback Relation ระหว่าง 2 ชนิดของกิจกรรม

- ทุก ๆ ครั้งที่มีการปรับตัวเป็นสแต็ป ให้เพิ่มสัญญาณนับของ Best - Matching Unit และเพิ่มค่าให้กับเซลล์ใกล้เคียงที่เพิ่มขึ้นมาใหม่
- ถ้ามีการเพิ่มใกล้กับเซลล์ C ให้ลดทั้งขนาด ของ Voronoi field F_c และค่าของ

สัญญาณนับ τ_c และลดค่า Voronoi ที่ทำให้เกิด Best-Matching Unit ดังรูป 6.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.6 แสดงการสร้างเซลล์แบบ 2 มิติ

การกำเนิดของเซลล์ จะมาจากสัญญาณอินพุทที่ป้อนเข้ามาโดยมีเงื่อนไขอีกอย่างหนึ่งคือ ไม่ออกนอก Subarea เริ่มต้นจากการสร้างให้อยู่ในรูปของสามเหลี่ยม สำหรับตำแหน่งของรูปที่จะวาดนั้นใช้วิธีการสุ่มค่าเริ่มต้น เพื่อเป็นการอ้างทางเวกเตอร์ โครงสร้างเดิมเมื่อมีอินพุทเข้าไป โครงข่ายเดิมจะถูกกระทบกระเทือนในรูปที่กระจายออกโดยค่าคงที่ค่าหนึ่ง คือ λ ของสัญญาณอินพุท 0 หมายถึง เซลล์ใหม่ที่เพิ่มเข้าไป และมีการต่อกับเซลล์ข้างเคียง ดังนั้นโครงข่ายเดิมจะต้องปรับโครงข่ายอีกครั้ง ให้เป็นรูปสามเหลี่ยม และจะกระทำทุกครั้งเมื่อมีสัญญาณใหม่เข้ามา

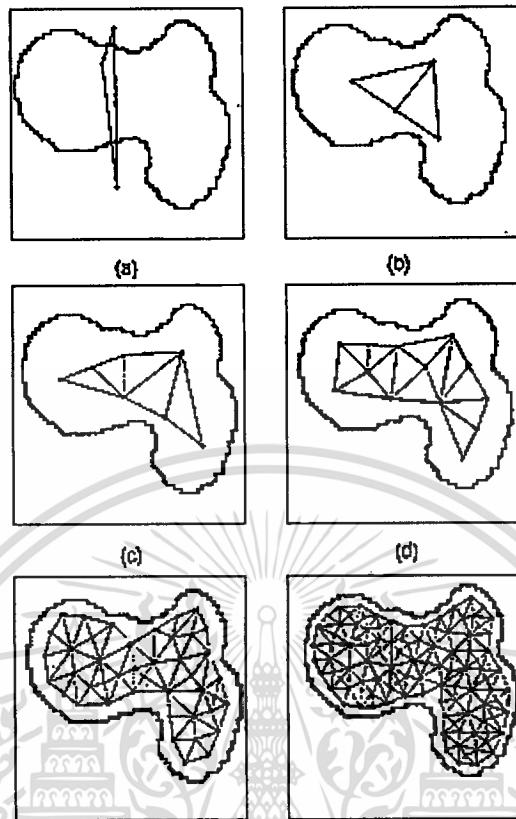
การที่จะทำให้เป็นจริงภายใต้ขอบเขต หรือข้อจำกัดของตัวแปร และโมเดลที่ได้ รับการพิสูจน์แล้วว่า ทุกเซลล์ i ที่เพิ่มเข้ามามันจะก่อให้เกิด Best-Matching Unit สำหรับค่าสัญญาณอินพุทมีค่าตรงต่อ $P(\xi)$ คือค่าประมาณที่เราให้เป็นค่าที่เท่ากันดังนี้.-

$$S = \sum_{c \in A} P_c \log P_c$$

ดังนั้นค่าความเข้มของจุดที่เพิ่มขึ้นมานั้น เราจะได้ค่าเวกเตอร์เป็นค่าความเข้มที่เราไม่ทราบค่าแต่เป็นค่าที่ดีที่สุด ของอินพุทเวกเตอร์ เราจะสามารถประมาณการได้แล้วว่า การดำเนินการของโครงสร้างเซลล์ที่เรากำหนดมานั้นดีกว่า ของ Kohonen's feature map

ส่วนต่อไปเป็นการแสดง การกำเนิดของโครงสร้างของโมเดลที่เป็นพื้นฐานที่สุด การกระทำดังกล่าวจะบอกได้เป็น fractal growth ที่สามารถตรวจดูค่าของความถี่ได้ ความสำคัญอีกอย่างคือขณะที่มีการเปลี่ยนแปลงอยู่นั้น เราสามารถเข้าไปขัดจังหวะ ทุก ๆ เวลา และรูปแบบของโครงสร้างได้แสดงดังรูปที่ 6.7 ต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



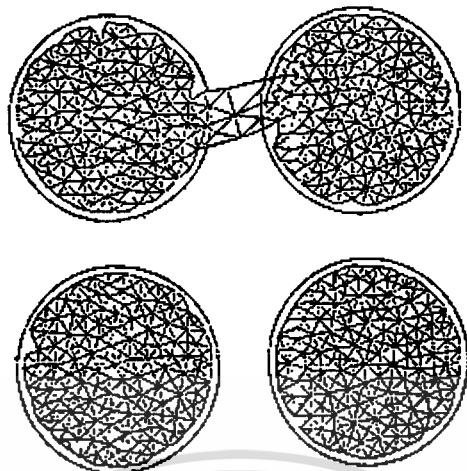
รูปที่ 6.7 แสดงพัฒนาการของโครงข่ายแบบ 2 มิติ เมื่อมีการเพิ่ม

แนนอนจำนวนเซลล์ที่ถูกสร้าง จะมีการเปลี่ยนแปลงตามค่าของเวกเตอร์ที่เกิดขึ้น และแหล่งที่ก่อเกิดใหญ่ที่สุดคือจุดที่มีการเพิ่มเซลล์เข้ามาใหม่นั้นเอง

6.4 การนำเซลล์ออก (Remove of Cells)

ถ้า $P(\xi)$ ประกอบไปด้วยหลายขอบเขตของค่าที่เป็นไปได้ทางด้านบวก และอยู่อย่างนั้น มันก็สามารถนำเซลล์ออกไปได้

$$\bar{P}_c = h_c / |F_c| \quad \text{ค่า } \bar{P}_c \text{ threshold } \eta$$



รูปที่ 6.8 เซลล์ขนาด 400 เซลล์ ที่มีการปรับ และเคลื่อนที่

6.5 การประมาณค่า Voronoi Region (Approximation of the Voronoi Region)

การคำนวณหาค่าของ Voronoi Tessellation ต้องระวังเป็นอย่างมาเมื่อมี $K > 2$ ดังนั้น เราจะแทน Voronoi Region ด้วย F_c ในการพิสูจน์ครั้งแรกโดย n -dimensional hypercube ด้วยการหาค่าที่เท่ากันให้มีความยาว τ_c จาก C ของ $f_c = |F_c|$, $f_c = (\tau_c)^n$

ค่าของ τ_c กำหนดได้จาก

$$\tau_c = 1/\text{card}(N_c) \sum_{i \in N_c} \|W_c - W_i\|$$

ถึงแม้ว่าการประมาณค่าของ probability density นั้นไม่มีความแน่นอนเสมอไป แต่มันก็เป็นค่าที่เชื่อถือได้ค่าของอินพุตที่ได้จะต้องมาทำการ Normalized Value ก่อนเสมอ

$$P^\Lambda = P^\sim \sum_{c \in \mathcal{A}} \tilde{f}_c$$

สิ่งที่เราจะกระทำคือการเลือกค่า K หรือมิติ แบบ hypercube หรือค่าของข้อมูลที่จะทำ ให้ได้ n -dimensional space แล้วให้กระทำครั้งต่อไปอยู่ภายใต้ Subarea R^n ตัวอย่างเช่น ข้อมูลขนาด 100-dimensional เราสามารถมองเห็นเป็นสองมิติได้ นั้นหมายความว่า หนึ่งในเซลล์ของเราที่เกิดขึ้นค่าความยาวอาจจะลดลงประมาณ 5 % จำนวนของ 100-dimensional

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะยุบลงน้อยลงไปกว่า 0.6 % แต่ไม่มีผลต่อการเปลี่ยนสนามของเซลล์ ในจุดนี้ค่าของ 2-dimensional hypercube ควรจะมีค่ามากกว่าค่าที่ประมาณไว้

ดังนั้นเราต้องหาค่าที่ดีที่สุดที่ทำให้ค่าของ มิติต่ำที่สุดที่สามารถแสดงข้อมูล ซึ่งเราได้ประมาณค่าของ Dimensional แบบ Voronoi region ในโมเดลของเรา ซึ่งกฎต่าง ๆ ต่อไปนี้เพื่อนำไปแก้ปัญหาก็จะเกิดขึ้น

- ให้เซตของ $t \leq n$
- ค่าความแตกต่างของ อินพุทเวกเตอร์ ให้ใช้ค่า $t = n$
- ถ้าค่านั้นขึ้นกับค่าของ l ให้ใช้ค่านั้นได้ทันที
- $t \leq$ จำนวนอินพุทเวกเตอร์เสมอ
- ทำการวิเคราะห์ขั้นสุดท้ายของข้อมูล ดังนั้น t ควรจะเป็นเซตของจำนวน

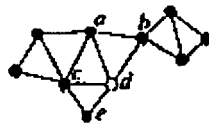
6.6 ประสิทธิภาพของการต่อแบบหลายมิติ (Efficient Manipulation of High-Dimensional Topologies)

การกระทำกับ K-dimensional ของการสร้างเซลล์ที่มีความซับซ้อนมากมาย มากกว่าการทำของ Kohonen feature map และทุกครั้งที่กระทำต่อโครงข่ายจะต้องมีการกระทำอยู่ 2 อย่างต่อไปนี้เป็นที่พื้นฐาน

- การเพิ่ม Neuron
- การลบ Neuron

ทุกครั้งที่เรากระทำต่อโครงข่ายเราต้องใช้ การกระทำ ทั้งสองแบบเสมอ โดยทั่วไปการ แสดงรูปแบบของโครงข่าย

สามารถนำมาแสดงได้แต่ไม่ใช่ กราฟเส้นตรง ดังรูปต่อไปนี้ ซึ่งประกอบด้วย โหนด และเส้นที่ เชื่อมโยงระหว่างคู่ของโหนด โดยโหนดที่จะต่อกันในลักษณะของตัวที่อยู่ใกล้เคียงกัน



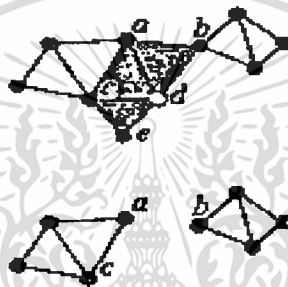
รูปที่ 6.9 แสดงการนำ เซลล์ d ออก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จุดที่เราจะนำมาพิจารณาว่าจะเปลี่ยนระดับ ให้มองจากโหนด และจุดที่เชื่อมต่อการเพิ่ม สามารถคำนวณได้จาก

- การเพิ่ม เช่น เมื่อเราเพิ่มโหนด r ให้แยกโหนดที่มีอยู่ q, f ที่โหนด r ถูกต่ออยู่กับ q, f เมื่อต่อ r เข้าไปแล้วให้ ปรับข้อมูลให้แต่ละ hypercube h ทั้ง q และ f จะถูกแทนด้วย h และ q ตามมาเป็นการทดแทนด้วยโหนด r สุดท้าย q, f จะถูกนำออก
- การลบ การลบเมื่อมีการลบโหนดออกให้ตรวจสอบเส้นที่เชื่อมโยง ซึ่งเส้นที่เชื่อมโยงจะต้องทำหน้าที่เป็นด้านหนึ่งของสามเหลี่ยม ถ้าไม่ใช่ให้ลบออกทันทีดังรูป 6.10



รูปที่ 6.10 แสดงให้เห็นว่าเมื่อนำเซลล์ d ออกแล้วต้องตรวจสอบว่าเส้นที่เชื่อมโยงเป็นส่วนหนึ่งของสามเหลี่ยมตามคุณสมบัติ หรือไม่

6.7 วิธีที่มองโครงข่ายที่หลายมิติ (Network Visualization for High- dimation Input Data)

Kohonen's feature map คือ วิธีการที่สามารถทำการโปรเจ็ก จาก หลาย ๆ มิติมาเป็นมิติที่ระดับต่ำได้ โดยการทำให้เป็นสามเหลี่ยม กริด ทำให้เรามองเห็นข้อมูลที่เป็นไปได้

การกำเนิดของเซลล์ในโครงข่าย แบบ 2- dimentional นับว่าน้อยมาก นั้นหมายถึงโครงข่ายประกอบไปด้วย จำนวนของการถูกต่อเป็นรูปสามเหลี่ยม และสามารถที่จะถูกนำออกมาได้ ในการออกแต่ละครั้งมีได้ตั้งแต่ หนึ่ง หรือมากกว่า ในวิธีการสร้างให้เป็นแบบ 2- dimentional ไม่ปรากฏให้เห็นชัดเจนนัก แต่ถ้าเราฝังมันเข้าไปในโครงข่ายเราก็สามารถเห็นมันได้ วิธีการอื่น ๆ คือ การโปรเจ็ก จากโครงข่ายมายัง อินพุทเวกเตอร์ แล้วนำค่าของอินพุทเวกเตอร์ ควรจัดให้เป็นรูปสามเหลี่ยมเท่านั้น วิธีการ

6.7.1 วิธีการสร้าง

6.7.1.1 กำหนดค่า K

ให้ K -dimensional network ($K \in \{2,3\}$) แล้วเรากำหนดค่าความเป็นไปได้ของ high Vector dimentional ให้มีค่าเพียงพอต่อการการใช้ในโครงข่ายวิธีการแบบนี้ อาศัยหลักฟิสิกส์อย่างง่าย ในการสร้าง K -dimensional ฝังเข้าไปในโครงข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระหว่างที่เกิดกระบวนการ Self-Organization โดยที่เรากำหนดค่า $K = 2$ เพื่อให้มีด้านของสามเหลี่ยม

6.7.1.2 กำหนดคุณสมบัติของเซลล์

- ให้เซลล์แต่ละเซลล์มีรูปเป็น Disc และ Disc ทำจากวัสดุที่ยืดหยุ่นได้
- เส้นผ่านศูนย์กลางของ Disc มีค่าเป็น d เมื่อ Disc1 และ Disc2 มีขอบที่สัมผัสกันพอดีเราจะได้ว่า จากจุดศูนย์กลางของ Disc1 ถึง Disc2 มีค่าเท่ากับ d คือระยะทาง และถ้าระยะทางที่ได้รับมีค่าน้อยกว่า d ทำให้เกิดแรงผลักระหว่าง Disc ทั้งสอง ผลที่ตามมาคือรูปแบบโครงข่ายที่เป็นอยู่จะเปลี่ยนรูปร่างไปจากเดิม
- เซลล์แต่ละเซลล์มีการเชื่อมต่อกัน เส้นที่เชื่อมต่อเรียกว่า Elastic String และถ้า Disc ทั้งสองต่อไม่ถึงกันมันก็จะเกิดแรงดูดซึ่งกัน ละครัน
- ให้ Disc ทั้งหมดมีประจุไฟฟ้าเป็นบวก และผลักรังกันละครัน

จุดเริ่มต้นอยู่ที่ Self-Organization process นั้นมีอยู่ 3 Disc ที่ต่ออยู่และอยู่ในสถานะที่สมดุล ซึ่งก็จะไม่เกิดการ Overlap ซึ่งกันละครัน เมื่อมีเซลล์เข้ามาใหม่ในแต่ละช่วงเวลา เซลล์แรกที่อยู่ก่อนหน้านี้จะถูกเปลี่ยนไป เพราะตัวที่เข้ามาจะมีการอ้างถึง เวกเตอร์ การกระทำแต่ละครั้งมันอาจจะเกิด Overlap ขึ้นมาอีกครั้งก็เป็นได้ หากเกิดขึ้นมาอีก เราต้องคำนวณหาผลรวมของแรงที่กระทำ และเคลื่อนที่ไปยังตำแหน่งที่ควรจะเป็น เป็นการทำให้ลักษณะ แบบไม่เป็นจังหวะ และหลีกเลี่ยงจากการแกว่ง แต่เราไม่สามารถที่จะสร้างโดยให้ โมเดลเป็นที่สมบูรณ์แบบได้ เพราะเนื่องจากว่า Disc นั้นมันไม่ขึ้นกับ มวล และแรง ต่อการเคลื่อนไหว จึงจำเป็นต้องใช้กฎในการตรวจสอบแรงดังนี้-

แรงดูด $f_n * 1.0$

$$f_n = \begin{cases} 0 & \text{if } e < d \\ (e-d)/2 & \text{(otherwise)} \end{cases}$$

แรงผลักร $f_b * 0.2$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$f_b = \begin{cases} 0 & \text{if } 3d < e \\ d/5 & \text{if } 2d < e \leq 3d \\ d/2 & \text{if } d < e \leq 2d \\ d & \text{if } 0 < e \leq d \\ 0 & \text{if } 0 = e \end{cases}$$

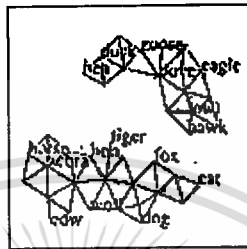
6.7.1.3 อินพุท

	D	G	E	T	H	Z									
Animal	D o H u o O a g F D o C g i r b C v e c s w w l o o l a e o s r o e n k e l k e x g f t r n e a w	D o H u o O a g F D o C g i r b C v e c s w w l o o l a e o s r o e n k e l k e x g f t r n e a w	D o H u o O a g F D o C g i r b C v e c s w w l o o l a e o s r o e n k e l k e x g f t r n e a w	D o H u o O a g F D o C g i r b C v e c s w w l o o l a e o s r o e n k e l k e x g f t r n e a w	D o H u o O a g F D o C g i r b C v e c s w w l o o l a e o s r o e n k e l k e x g f t r n e a w	D o H u o O a g F D o C g i r b C v e c s w w l o o l a e o s r o e n k e l k e x g f t r n e a w	D o H u o O a g F D o C g i r b C v e c s w w l o o l a e o s r o e n k e l k e x g f t r n e a w								
Is															
Small	1	1	1	1	1	1	0	0	0	0	1	0	0	0	0
Medium	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0
Big	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
Has															
Two legs	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
Four legs	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
Hair	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
Hooves	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
Mane	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1
Feather	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
Likes to															
Hunt	0	0	0	0	1	1	1	1	0	1	1	1	1	0	0
Run	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1
Fly	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0
Swim	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0

ตารางที่ 6.1 แสดงอินพุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากนั้นเราก็จะทำการสร้าง Cell Structure ด้วยข้อมูลชนิดเดียวกัน และกระทำในระหว่างการเกิด Self-Organization เป็น 2 - dimentional ที่ฝังไปในโครงข่าย ด้วยวิธีการดังกล่าวดังรูป 6.12 ซึ่งการเปรียบเทียบวิธีการของ Ritt และ Kohonen แล้ววิธีการที่เราสร้างอยู่เป็นวิธีการที่ดีกว่า



รูปที่ 6.12 แสดงให้เห็นการใช้อัลกอริทึมเพื่อสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

การคำนวณ และการสร้าง

การสร้างโปรแกรมนำเสนอการเกิดของเซลล์ในโครงข่ายประสาทเทียม จากหลาย ๆ มิติ มาเป็น 2 มิติ เพื่อให้สามารถมองดูง่าย แต่ยังคงความสัมพันธ์อยู่

7.1 เป้าหมาย

การแสดงผลของการแมปปีง การเกิดของเซลล์ในโครงข่ายประสาทเทียม ที่มีหลาย ๆ มิติ โดยการนำข้อมูลที่ได้รับการหาความสัมพันธ์ แล้วนำมาเสนอข้อมูล วิธีการนำเสนอข้อมูลเป็นการสร้างโปรแกรมจากโปรแกรมไมโครซอฟท์วิชวล C++

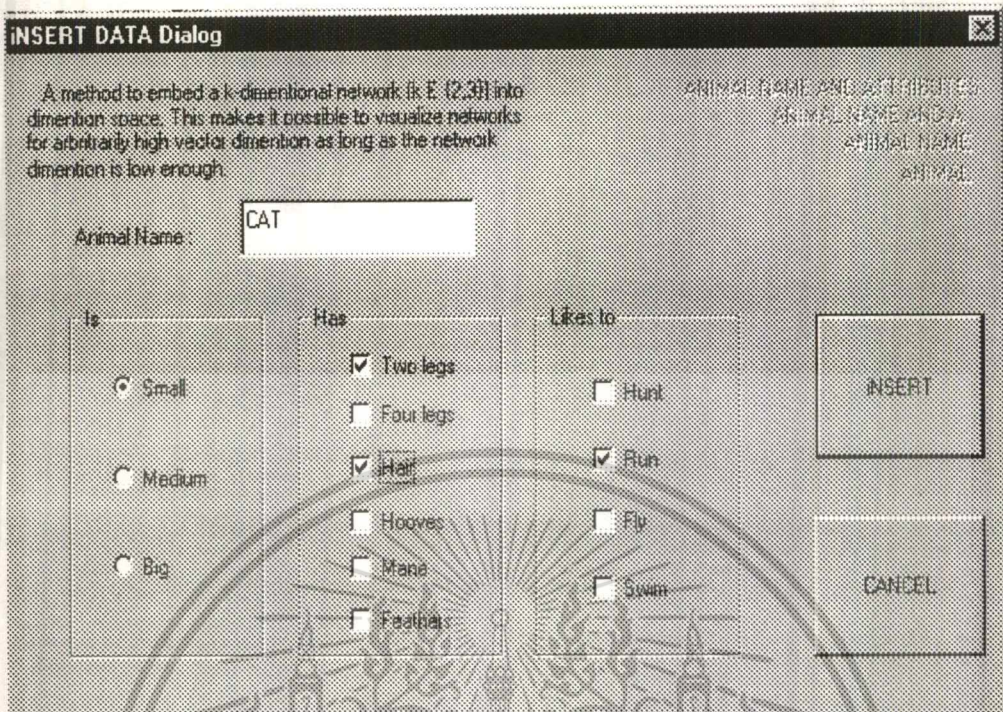
7.2 กำหนดคุณสมบัติของโปรแกรม

1. อินพุทของโปรแกรมมาจาก Text File หรือจาก Dialog

dove	1000010010	1	2	3	4	5	6
hen	1000010000	0	2	4	0	0	0
duck	1000010001	0	1	3	0	0	0
goose	1000010011	0	2	0	0	0	0
owl	1000011010	0	1	1	5	7	8
hawk	1000011011	0	4	6	7	10	13
eagle	1000011110	0	5	13	0	0	0
fox	0110001000	4	5	8	9	10	11
dog	0110000100	4	7	9	0	0	0
wolf	0110101100	7	8	0	0	0	0
cat	0110001001	5	7	11	12	13	0

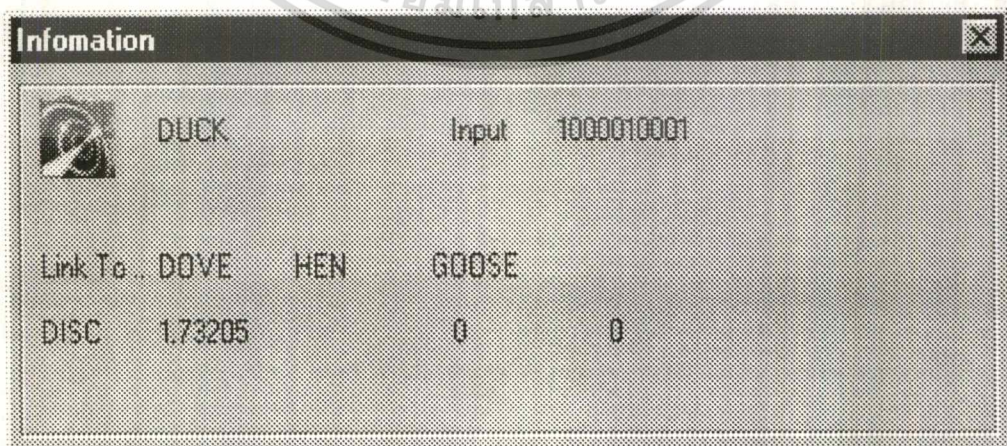
ตัวอย่างของ Text File

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.1 แสดง Dialog ที่รับข้อมูลจากหน้าจอ

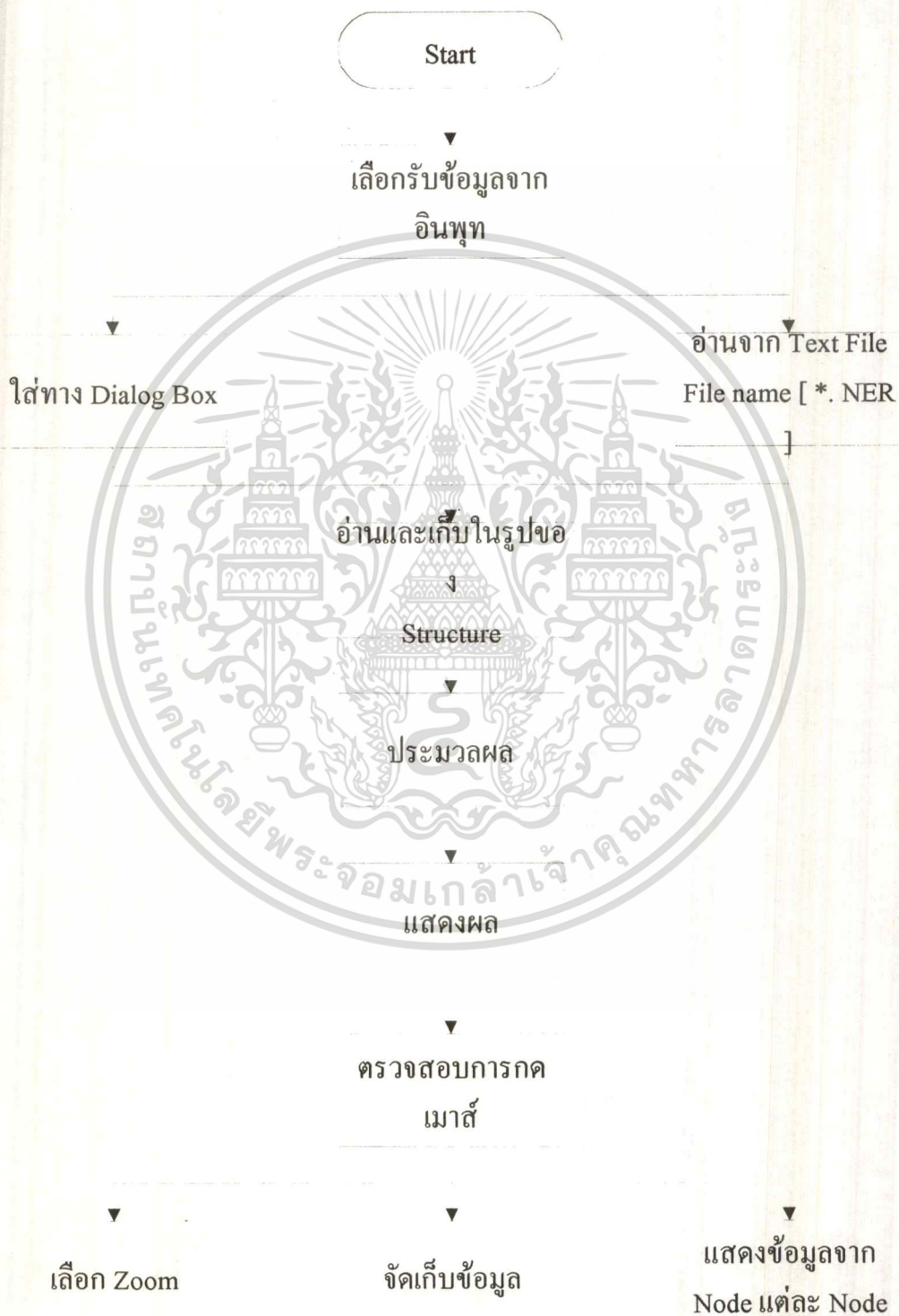
2. ประมวล หรือคำนวณ จากอินพุตที่มีหลายมิติ มาเป็น 2 มิติ ตามวิธีในบทที่ 6
3. นำเสนอข้อมูล และความสัมพันธ์ โดยใช้เส้นเชื่อมโยงเป็นตัวบอกความสัมพันธ์
4. สามารถแสดงรายละเอียดเพิ่มเติมได้ดังนี้



รูปที่ 7.2 แสดงรายละเอียดเพิ่มเติมของแต่ละเซลล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.3 โฟลว์ชาร์ตแสดงการทำงานโดยรวมของโปรแกรม



รูปที่ 7.3 แสดงโฟลว์ชาร์ตของโปรแกรม

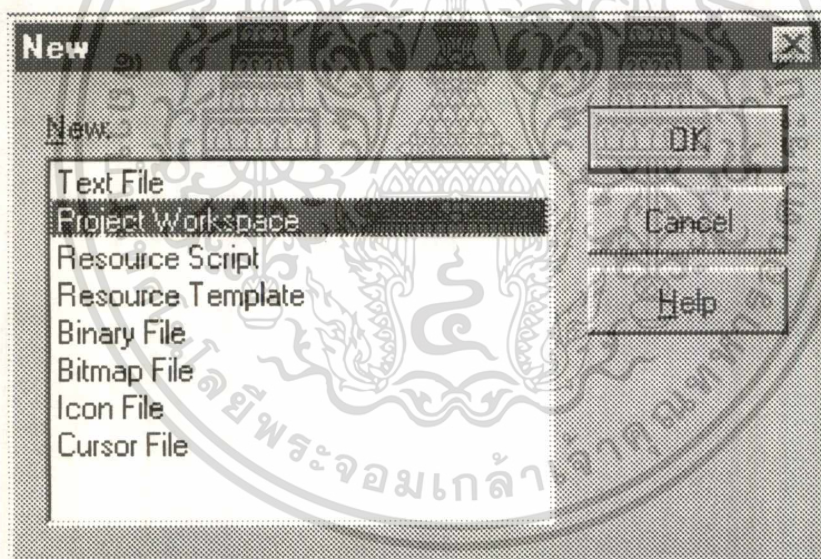
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.4 ความต้องการของโปรแกรมเบื้องต้น

1. ข้อมูลที่ได้รับการแมป์ปิ้งจากหลายมิติ มาเป็น 2 มิติ
2. Software ที่ใช้ คือ ไมโครซอฟท์ วิซวล C++ เวอร์ชันสำหรับ Windows'95 หรือ Windows NT
3. เครื่องคอมพิวเตอร์ตั้งแต่รุ่น DX-486 ขึ้นไป
4. หน่วยความจำแรมอย่างน้อย 8 MB ขึ้นไป ซึ่งควรใช้ขั้นต่ำ 16 MB

7.5 ขั้นตอนการสร้าง

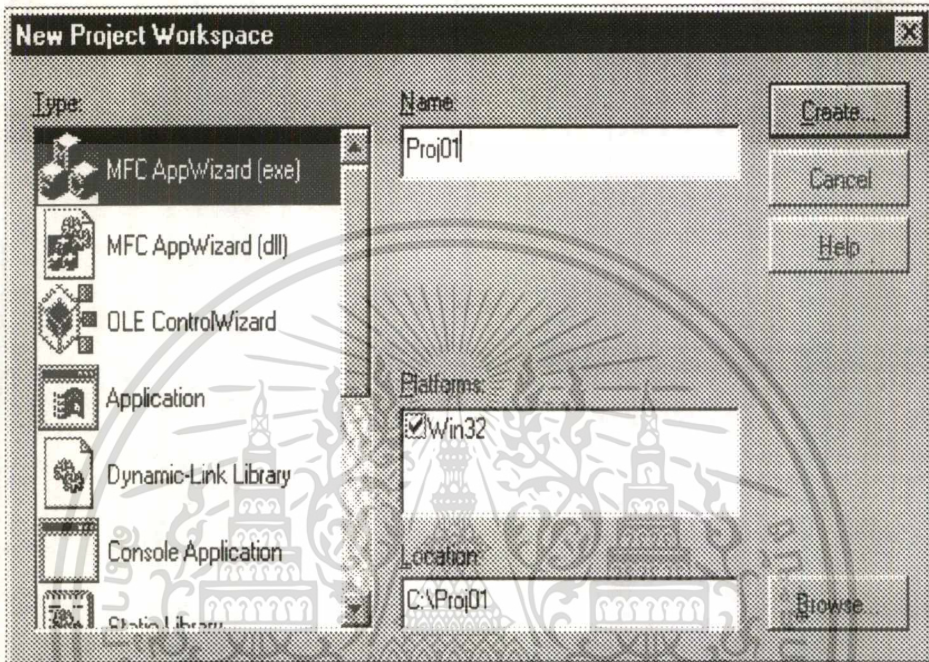
7.5.1 ใช้ Library MFC เลือก -FileNew แล้วจะปรากฏดังรูป



รูปที่ 7.4 แสดงการเปิด Project workspace

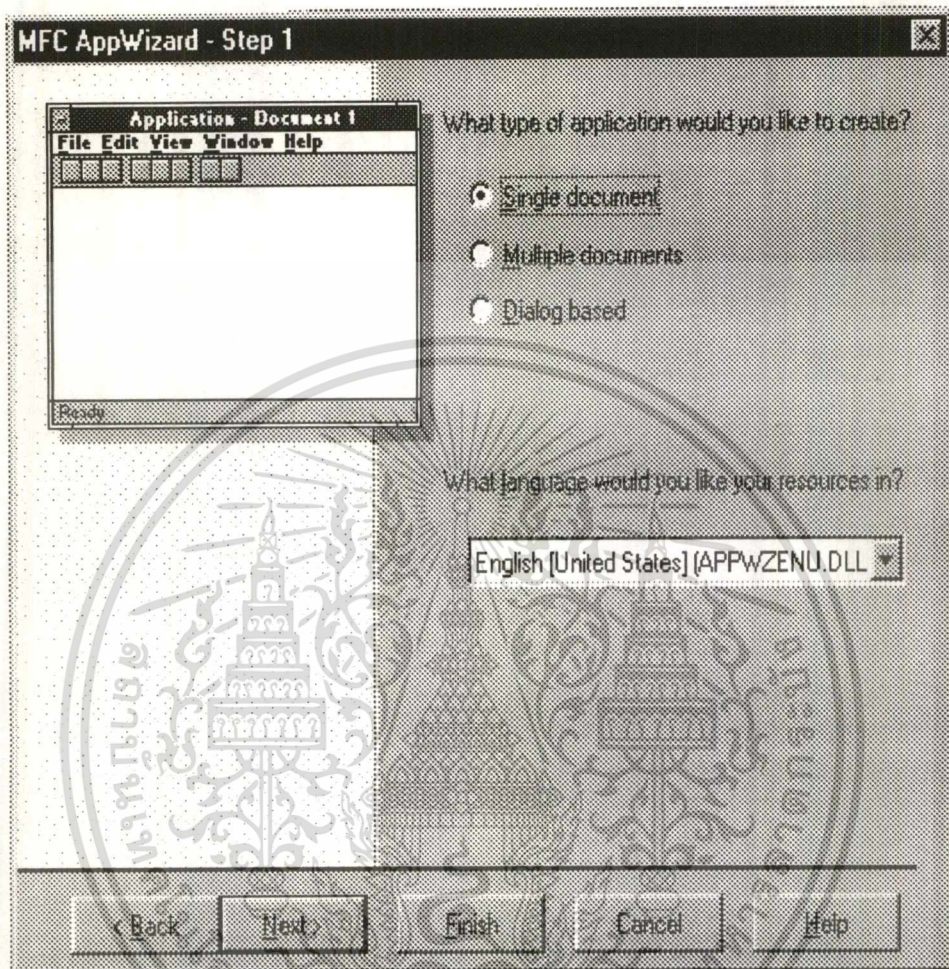
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.5.1 เมื่อเลือก Project workspace แล้วให้ระบุชื่อของโปรแกรมที่จะสร้าง



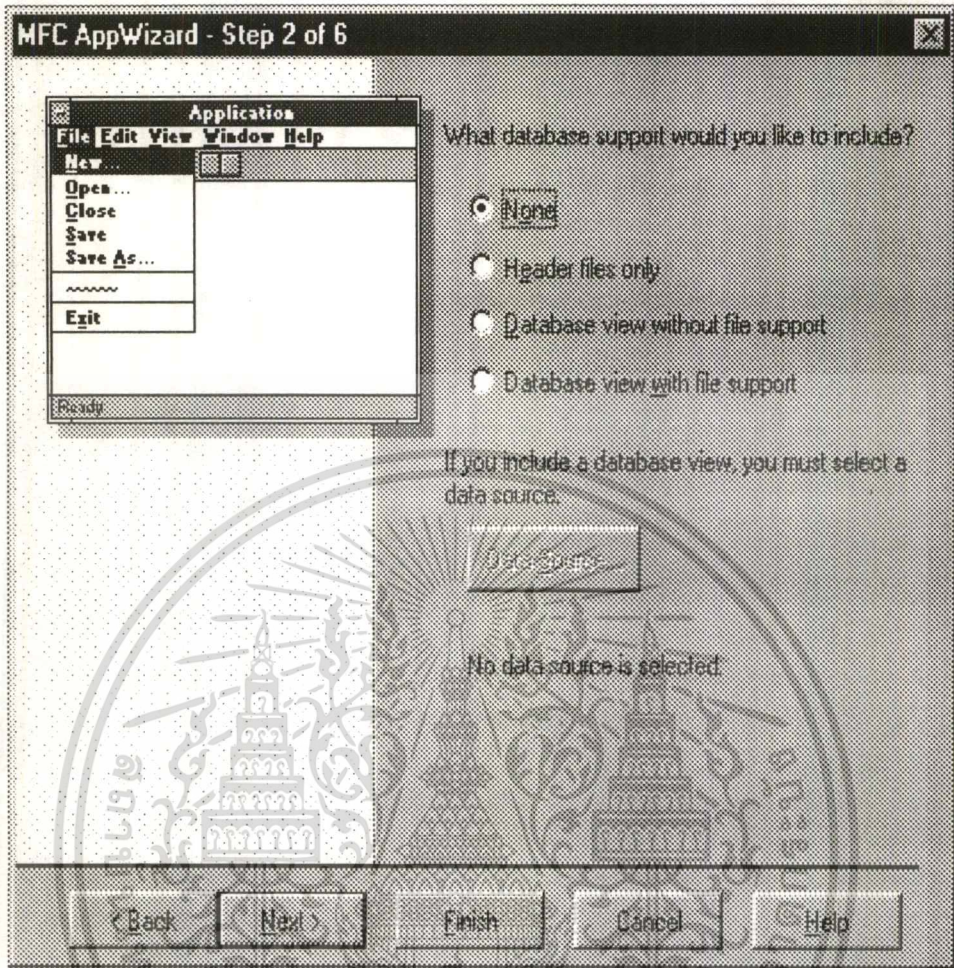
รูปที่ 7.5 แสดงการใช้ MFC AppWizard แล้วระบุชื่อโปรแกรม

7.5.2 ให้ดำเนินการต่าง ๆ ตามค่าจาก Dialog ของ AppWizard



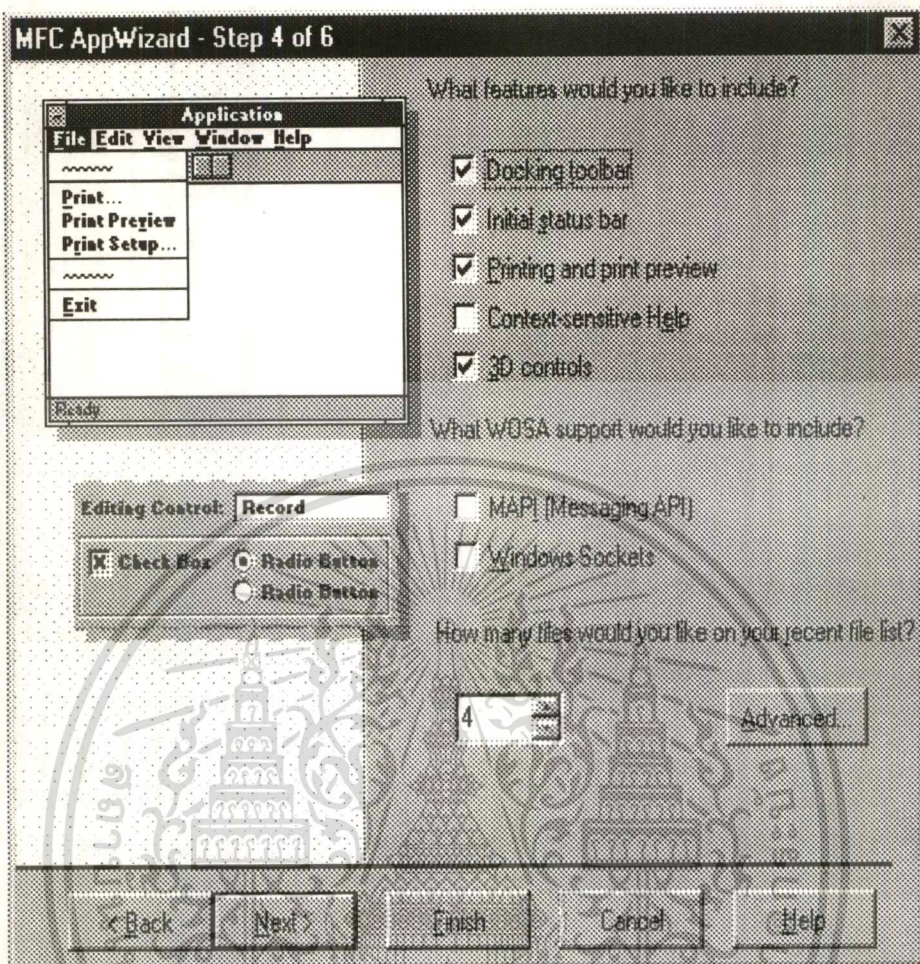
รูปที่ 7.6 การเลือกประเภทของ Document

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



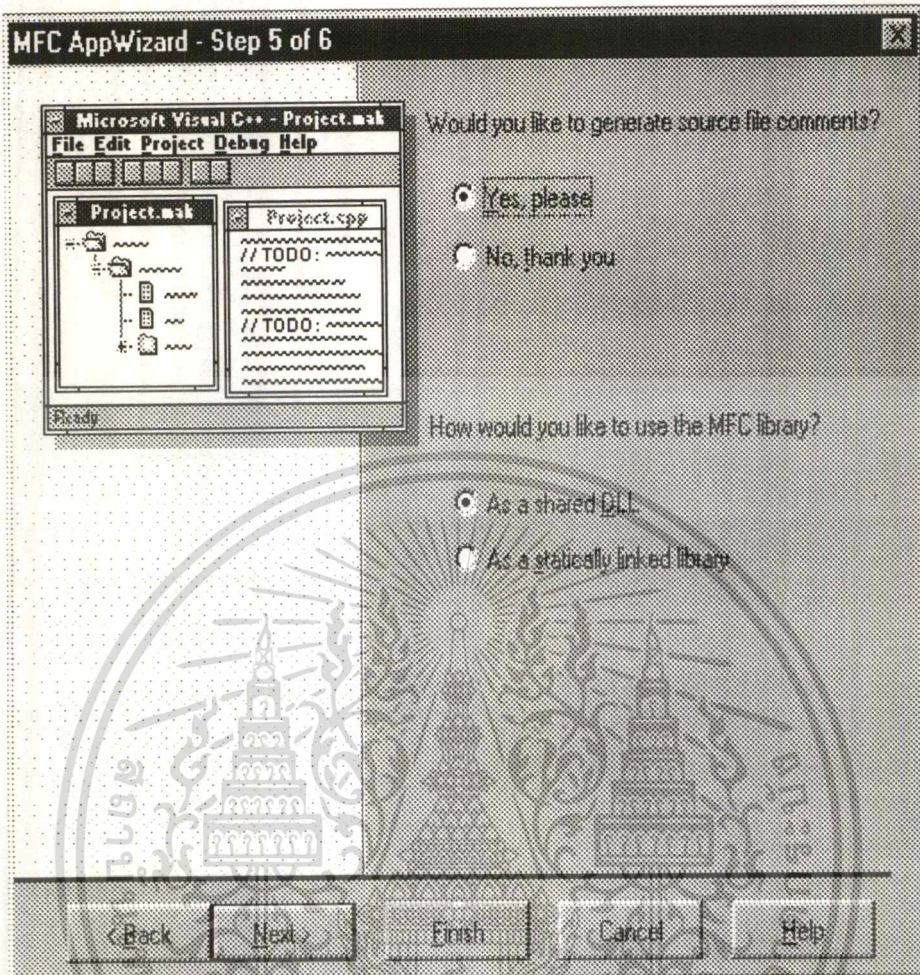
รูปที่ 7.7 การเลือกการ Support Data Base

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



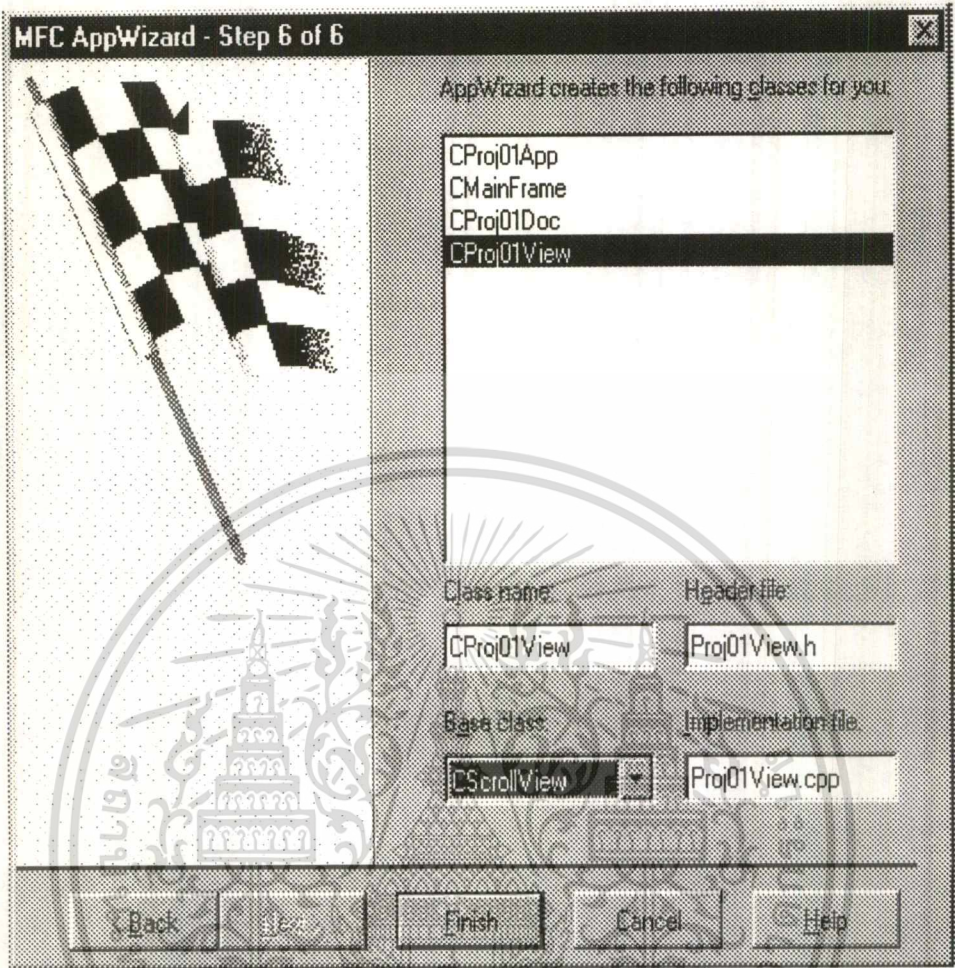
รูปที่ 7.8 การเลือกประเภทของเมนู ทูลบาร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



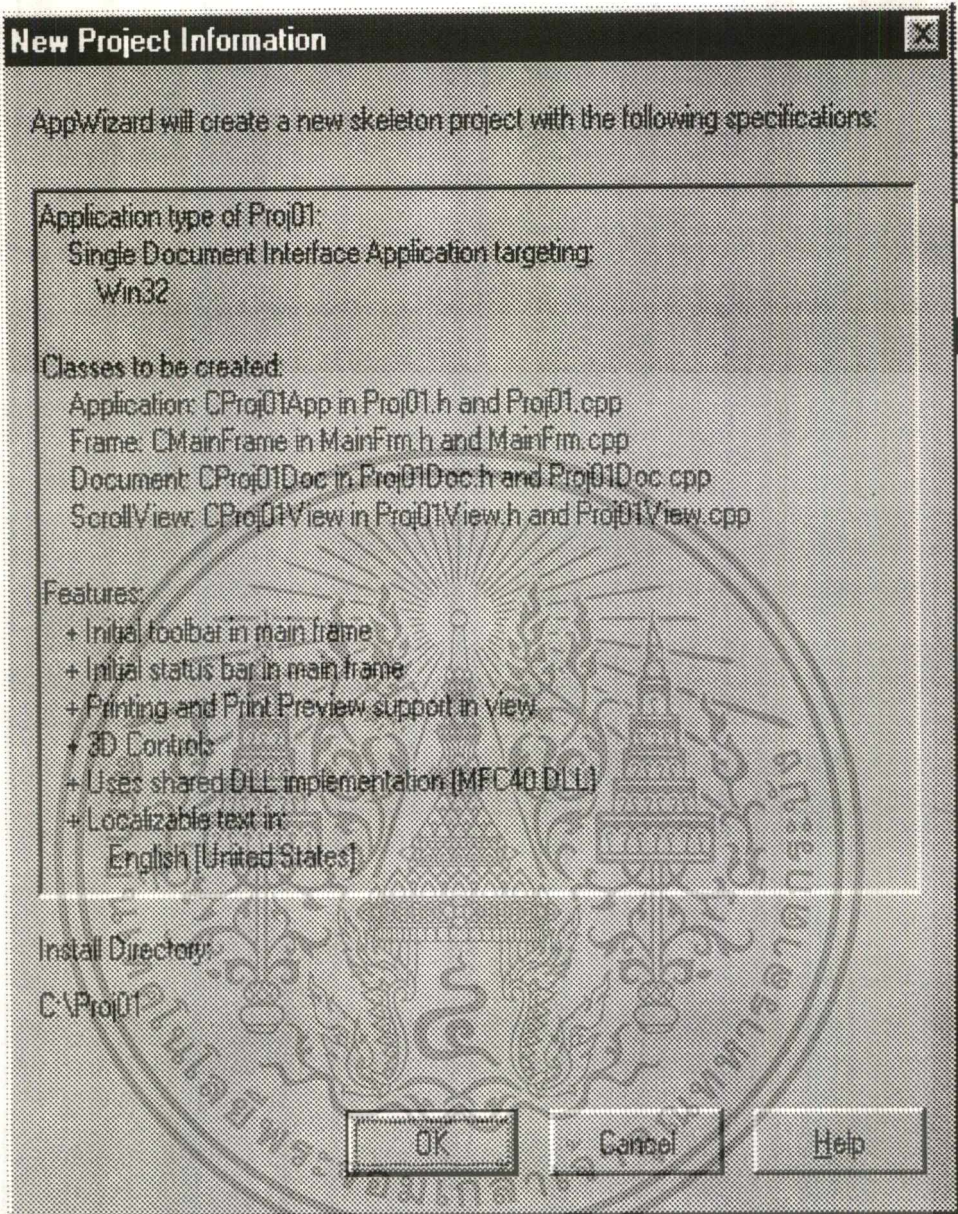
รูปที่ 7.9 การเลือก Comment

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



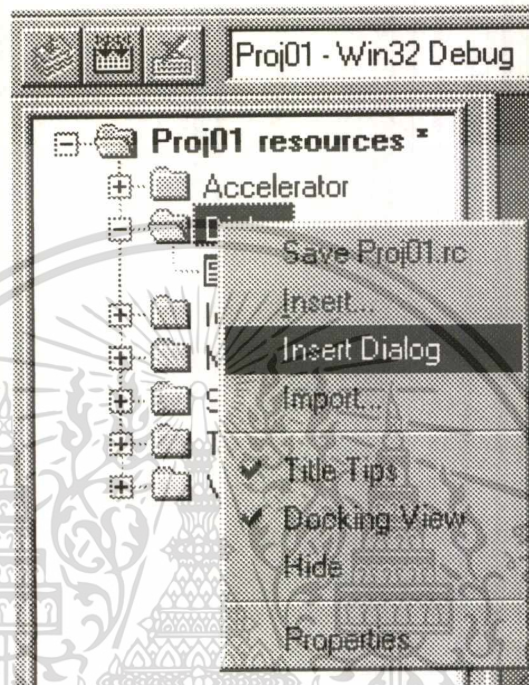
รูปที่ 7.10 การเลือกประเภทของ View

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

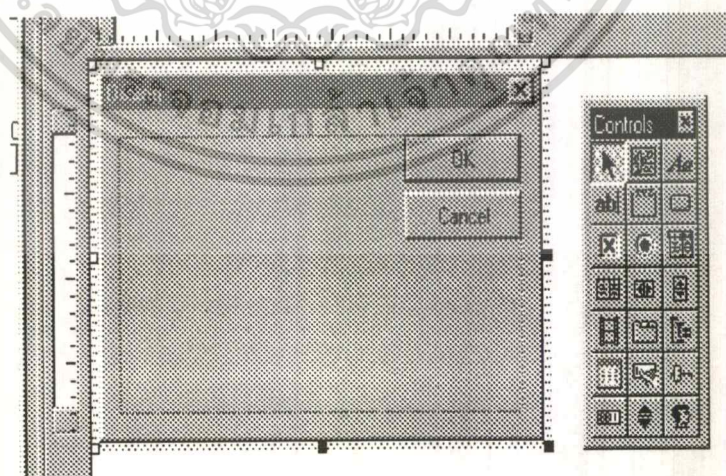


รูปที่ 7.11 AppWizard แสดงผลสุดท้ายที่เลือกทั้งหมด

7.5.3 การสร้าง Dialog เพื่อ Zoom



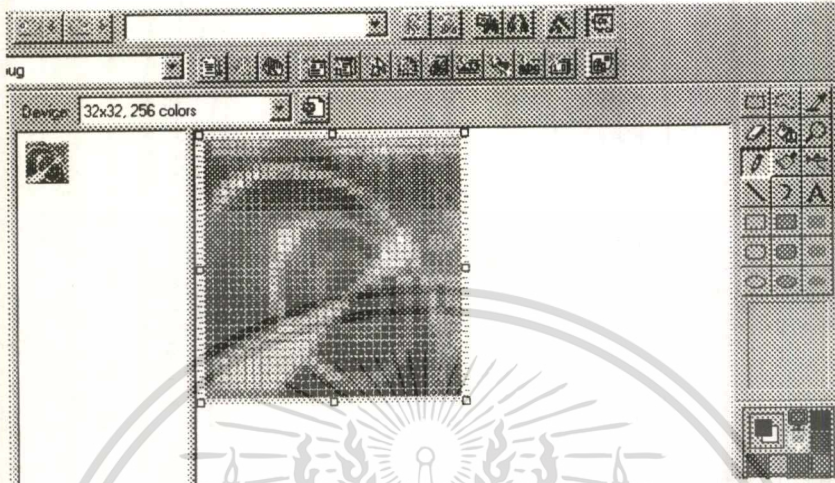
รูปที่ 7.12 แสดงการเพิ่ม Dialog



รูปที่ 7.13 แสดงการ Edit Dialog

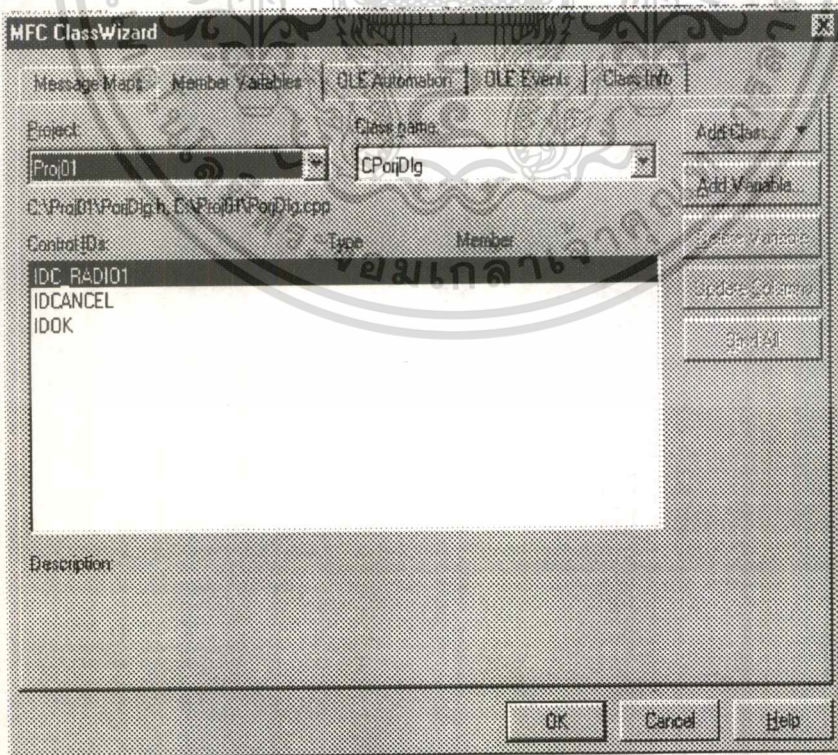
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ .

7.5.4 การสร้าง editor Icon



รูปที่ 7.14 แสดงสร้าง Icon

7.5.5 การ Add Class Wizard หลังจาก สร้าง Dialog , Toolbar



รูปที่ 7.15 แสดงใช้ ClassWizard สร้างฟังก์ชันการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.6 การเขียนชอร์สโค้ดเพื่อควบคุมการทำงานต่าง ๆ ของแต่ละฟังก์ชัน

สำหรับการเขียนชอร์สโค้ดเพื่อควบคุมการทำงานต่าง ๆ ของแต่ละฟังก์ชัน ได้รวบรวมไว้ในภาคผนวกทั้งหมดแล้ว



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8

การทดลอง และผลการทดลอง

8.1 การทดลองกับข้อมูลจากตารางที่ 1

การสร้างได้ใช้กับข้อมูลจากตารางข้อมูล หัวข้อที่ 7.2 ซึ่งข้อมูลดังกล่าวมีค่าอยู่ที่ 0.00 - 4.00 หน่วยเท่านั้นซึ่งผลของโปรแกรมที่ออกมานั้นจะเป็นดังรูปที่ 8.1



รูปที่ 8.1 แสดงผลการนำเสนอข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 9

บทวิจารณ์ และสรุป

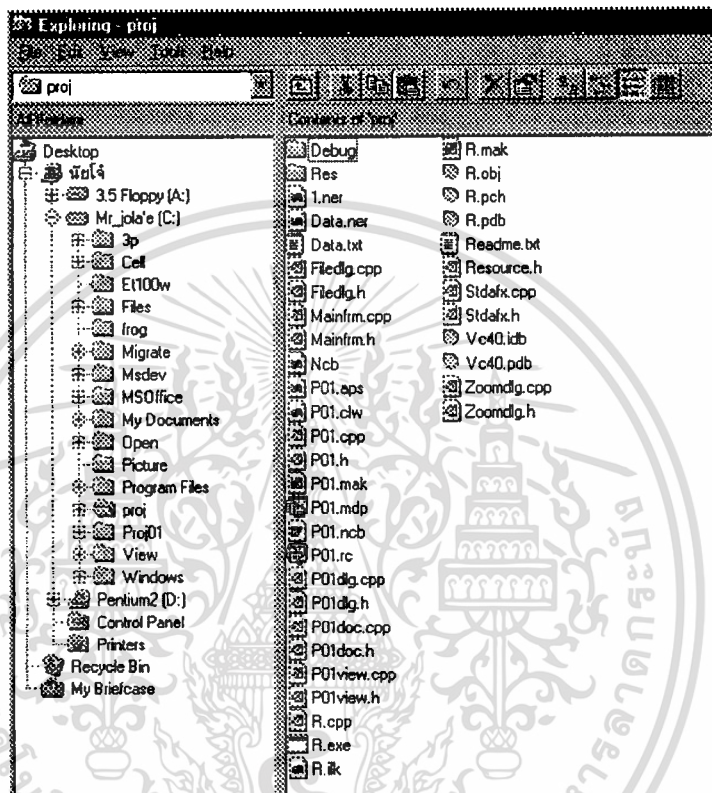
โครงการที่ได้จัดทำครั้งนี้ได้ ผู้ที่จะสร้างต้องมีความรู้ความสามารถทางด้านโครงข่ายประสาทเทียมมาบ้าง เพราะว่าเรื่องนี้ถือเป็นเรื่องยากต่อการศึกษาพอสมควร ทั้งเนื้อหาที่จะศึกษานั้นผู้อ่านต้องมีความสามารถทางคณิตศาสตร์ ฟิสิกส์ ประกอบกับข้อมูลที่เป็นภาษาไทยมีให้อ่านน้อยมาก และต้องใช้เวลาของการศึกษาค้นคว้าพอสมควร

ทางคณะผู้จัดทำได้นั้นไม่ได้เริ่มสร้างตั้งแต่การนำข้อมูลดิบ แต่ได้นำเสนอข้อมูลที่ได้รับ การพิสูจน์แล้วว่าวิธีที่เคยปฏิบัติมานั้นไม่สามารถให้ข้อมูลที่ชัดเจนไปกว่า ข้อมูลที่ใช้วิธีแบบใหม่ แต่วิธีการสร้างนั้นไม่ได้ใช้คณิตศาสตร์ขั้นสูง และได้ทดลองทำบนกระดาษ และอาศัยหลักการ แล้วจึงนำวิธีการคิดสร้างโปรแกรมอีกครั้ง อาจจะมีข้อผิดพลาดอยู่บ้าง



ภาคผนวก

ซอร์สโค้ดประกอบด้วยไฟล์ต่าง ๆ ดังนี้



dove	1000010010	1	2	3	4	5	6
hen	1000010000	0	2	4	0	0	0
duck	1000010001	0	1	3	0	0	0
goose	1000010011	0	2	0	0	0	0
owl	1000011010	0	1	1	5	7	8
hawk	1000011011	0	4	6	7	10	13
eagle	1000011110	0	5	13	0	0	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// FILEDlg.h : header file
//

/////////////////////////////////////////////////////////////////

// CFILEDlg dialog

class CFILEDlg : public CDialog
{
// Construction
public:
    CFILEDlg(CWnd* pParent = NULL); // standard constructor

// Dialog Data
   //{{AFX_DATA(CFILEDlg)
    enum { IDD = IDD_FILE_DIALOG };
    CString m_file;
    //}}AFX_DATA

// Overrides
    // ClassWizard generated virtual function overrides
    //{{AFX_VIRTUAL(CFILEDlg)
protected:
    virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV support
    //}}AFX_VIRTUAL

// Implementation
protected:

    // Generated message map functions
    //{{AFX_MSG(CFILEDlg)
        // NOTE: the ClassWizard will add member functions here
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

};
// FILEDlg.cpp : implementation file
//

#include "stdafx.h"
#include "P01.h"
#include "FILEDlg.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

//////////////////////////////////////
// FILEDlg dialog

CFILEDlg::CFILEDlg(CWnd* pParent /*=NULL*/)
    : CDialog(CFILEDlg::IDD, pParent)
{
   //{{AFX_DATA_INIT(CFILEDlg)
    m_file = _T("");
    //}}AFX_DATA_INIT
}

void CFILEDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    {{{AFX_DATA_MAP(CFILEDlg)
    DDX_Text(pDX, IDC_FILE_EDIT, m_file);
    //}}AFX_DATA_MAP
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

BEGIN_MESSAGE_MAP(CFILEDlg, CDialog)
   //{{AFX_MSG_MAP(CFILEDlg)
        // NOTE: the ClassWizard will add message map macros here
   //}}AFX_MSG_MAP
END_MESSAGE_MAP()

```

```

////////////////////////////////////
// CFILEDlg message handlers

// MainFrm.h : interface of the CMainFrame class
//
////////////////////////////////////

class CMainFrame : public CFrameWnd
{
protected: // create from serialization only
    CMainFrame();
    DECLARE_DYNCREATE(CMainFrame)

```

```
// Attributes
```

```
public:
```

```
// Operations
```

```
public:
```

```
// Overrides
```

```
    // ClassWizard generated virtual function overrides
```

```
    {{{AFX_VIRTUAL(CMainFrame)
```

```
    virtual BOOL PreCreateWindow(CREATESTRUCT& cs);
```

```
    }}}AFX_VIRTUAL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// Implementation
public:
    virtual ~CMainFrame();
#ifdef _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
#endif

protected: // control bar embedded members
    CStatusBar m_wndStatusBar;
    CToolBar m_wndToolBar;

// Generated message map functions
protected:
   //{{AFX_MSG(CMainFrame)
    afx_msg int OnCreate(LPCREATESTRUCT lpCreateStruct);
        // NOTE - the ClassWizard will add and remove member functions here.
        // DO NOT EDIT what you see in these blocks of generated code!
   //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

// MainFrm.cpp : implementation of the CMainFrame class
//

#include "stdafx.h"
#include "P01.h"

#include "MainFrm.h"

```

```
#ifdef _DEBUG
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#define new DEBUG_NEW

#undef THIS_FILE
static char THIS_FILE[] = __FILE__;

#endif

int zoom;

////////////////////////////////////

// CMainFrame

IMPLEMENT_DYNCREATE(CMainFrame, CFrameWnd)

BEGIN_MESSAGE_MAP(CMainFrame, CFrameWnd)
   //{{AFX_MSG_MAP(CMainFrame)
        // NOTE - the ClassWizard will add and remove mapping macros here.
        // DO NOT EDIT what you see in these blocks of generated code !
    ON_WM_CREATE()
    }}AFX_MSG_MAP
END_MESSAGE_MAP()

static UINT indicators[] =
{
    ID_SEPARATOR,           // status line indicator
    ID_INDICATOR_CAPS,
    ID_INDICATOR_NUM,
    ID_INDICATOR_SCRL,
};

////////////////////////////////////

// CMainFrame construction/destruction

CMainFrame::CMainFrame()
{
    // TODO: add member initialization code here

}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CMainFrame::~CMainFrame()
{
}

int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    if (CFrameWnd::OnCreate(lpCreateStruct) == -1)
        return -1;

    if (!m_wndToolBar.Create(this) ||
        !m_wndToolBar.LoadToolBar(IDR_MAINFRAME))
    {
        TRACE0("Failed to create toolbar\n");
        return -1; // fail to create
    }

    if (!m_wndStatusBar.Create(this) ||
        !m_wndStatusBar.SetIndicators(indicators,
        sizeof(indicators)/sizeof(UINT)))
    {
        TRACE0("Failed to create status bar\n");
        return -1; // fail to create
    }

    // TODO: Remove this if you don't want tool tips or a resizable toolbar
    m_wndToolBar.SetBarStyle(m_wndToolBar.GetBarStyle() |
        CBRS_TOOLTIPS | CBRS_FLYBY | CBRS_SIZE_DYNAMIC);

    // TODO: Delete these three lines if you don't want the toolbar to
    // be dockable
    m_wndToolBar.EnableDocking(CBRS_ALIGN_ANY);
    EnableDocking(CBRS_ALIGN_ANY);
    DockControlBar(&m_wndToolBar);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        return 0;
    }

    BOOL CMainFrame::PreCreateWindow(CREATESTRUCT& cs)
    {
        // TODO: Modify the Window class or styles here by modifying
        // the CREATESTRUCT cs

        return CFrameWnd::PreCreateWindow(cs);
    }

    //////////////////////////////////////
    // CMainFrame diagnostics

#ifdef _DEBUG
    void CMainFrame::AssertValid() const
    {
        CFrameWnd::AssertValid();
    }

    void CMainFrame::Dump(CDumpContext& dc) const
    {
        CFrameWnd::Dump(dc);
    }

#endif // _DEBUG

    //////////////////////////////////////
    // CMainFrame message handlers

; CLW file contains information for the MFC ClassWizard

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[General Info]

Version=1

LastClass=CP01View

LastTemplate=CDialog

NewFileInclude1=#include "stdafx.h"

NewFileInclude2=#include "P01.h"

LastPage=0

ClassCount=8

Class1=CP01App

Class2=CP01Doc

Class3=CP01View

Class4=CMainFrame

ResourceCount=7

Resource1=IDD_ABOUTBOX

Resource2=IDR_MAINFRAME

Class5=CAboutDlg

Resource3=IDR_MAINFRAME (English (U.S.))

Resource4=IDD_FILE_DIALOG

Class6=CP01Dlg

Resource5=IDD_ABOUTBOX (English (U.S.))

Class7=CFILEDlg

Resource6=IDD_DIALOG1

Class8=CZoomDlg

Resource7=IDD_INFO_DIALOG

[CLS:CP01App]

Type=0

HeaderFile=P01.h

ImplementationFile=P01.cpp

Filter=N

[CLS:CP01Doc]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Type=0
 HeaderFile=P01Doc.h
 ImplementationFile=P01Doc.cpp
 Filter=N
 LastObject=CP01Doc

[CLS:CP01View]

Type=0
 HeaderFile=P01View.h
 ImplementationFile=P01View.cpp
 Filter=C
 LastObject=ID_VIEW_ZOOM
 BaseClass=CScrollView
 VirtualFilter=VWC

[CLS:CMainFrame]

Type=0
 HeaderFile=MainFrm.h
 ImplementationFile=MainFrm.cpp
 Filter=T

[CLS:CAboutDlg]

Type=0
 HeaderFile=P01.cpp
 ImplementationFile=P01.cpp
 Filter=D
 LastObject=CAboutDlg

[DLG:IDD_ABOUTBOX]

Type=1
 ControlCount=4

Control1=IDC_STATIC,static,1342177283

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Control2=IDC_STATIC,static,1342308352

Control3=IDC_STATIC,static,1342308352

Control4=IDOK,button,1342373889

Class=CAboutDlg

[MNU:IDR_MAINFRAME]

Type=1

Class=CMainFrame

Command3=ID_FILE_SAVE

Command4=ID_FILE_SAVE_AS

Command5=ID_FILE_PRINT

Command6=ID_FILE_PRINT_PREVIEW

Command7=ID_FILE_PRINT_SETUP

Command8=ID_FILE_MRU_FILE1

Command9=ID_APP_EXIT

Command10=ID_EDIT_UNDO

Command11=ID_EDIT_CUT

Command12=ID_EDIT_COPY

Command13=ID_EDIT_PASTE

Command14=ID_VIEW_TOOLBAR

Command15=ID_VIEW_STATUS_BAR

CommandCount=16

Command1=ID_FILE_NEW

Command2=ID_FILE_OPEN

Command16=ID_APP_ABOUT

[ACL:IDR_MAINFRAME]

Type=1

Class=CMainFrame

Command1=ID_FILE_NEW

Command2=ID_FILE_OPEN

Command3=ID_FILE_SAVE

Command4=ID_FILE_PRINT

Command5=ID_EDIT_UNDO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Command6=ID_EDIT_CUT
 Command7=ID_EDIT_COPY
 Command8=ID_EDIT_PASTE
 Command9=ID_EDIT_UNDO
 Command10=ID_EDIT_CUT
 Command11=ID_EDIT_COPY
 Command12=ID_EDIT_PASTE
 CommandCount=14
 Command13=ID_NEXT_PANE
 Command14=ID_PREV_PANE

[DLG:IDD_INFO_DIALOG]

Type=1

Class=CP01Dlg

ControlCount=12

Control1=IDC_NODE_EDIT,edit,1342179464

Control2=IDC_INPUT_STATIC,static,1342308352

Control3=IDC_LINK_STATIC,static,1342308352

Control4=IDC_NODE1_EDIT,edit,1342179464

Control5=IDC_INPUT_EDIT,edit,1342179456

Control6=IDC_NODE2_EDIT,edit,1342179464

Control7=IDC_STATIC,button,1342177287

Control8=IDC_STATIC,static,1342177283

Control9=IDC_NODE3_EDIT,edit,1342179464

Control10=IDC_NODE4_EDIT,edit,1342179464

Control11=IDC_NODE5_EDIT,edit,1342179464

Control12=IDC_NODE6_EDIT,edit,1342179464

[TB:IDR_MAINFRAME (English (U.S.))]

Type=1

Class=CP01View

Command1=ID_FILE_NEW

Command2=ID_FILE_OPEN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Command3=ID_FILE_SAVE
Command4=ID_EDIT_CUT
Command5=ID_EDIT_COPY
Command6=ID_EDIT_PASTE
Command7=ID_FILE_PRINT
Command8=ID_APP_ABOUT
Command9=ID_VIEW_ZOOM
CommandCount=9

```

```
[MNU:IDR_MAINFRAME (English (U.S.))]
```

```
Type=1
```

```

Command1=ID_FILE_NEW
Command2=ID_FILE_OPEN
Command3=ID_FILE_SAVE
Command4=ID_FILE_SAVE_AS
Command5=ID_FILE_PRINT
Command6=ID_FILE_PRINT_PREVIEW
Command7=ID_FILE_PRINT_SETUP
Command8=ID_FILE_MRU_FILE1
Command9=ID_APP_EXIT
Command10=ID_EDIT_UNDO
Command11=ID_EDIT_CUT
Command12=ID_EDIT_COPY
Command13=ID_EDIT_PASTE
Command14=ID_VIEW_TOOLBAR
Command15=ID_VIEW_STATUS_BAR
Command16=ID_VIEW_ZOOM
Command17=ID_APP_ABOUT
CommandCount=17

```

```
[ACL:IDR_MAINFRAME (English (U.S.))]
```

```
Type=1
```

```

Command1=ID_FILE_NEW
Command2=ID_FILE_OPEN

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Command3=ID_FILE_SAVE
Command4=ID_FILE_PRINT
Command5=ID_EDIT_UNDO
Command6=ID_EDIT_CUT
Command7=ID_EDIT_COPY
Command8=ID_EDIT_PASTE
Command9=ID_EDIT_UNDO
Command10=ID_EDIT_CUT
Command11=ID_EDIT_COPY
Command12=ID_EDIT_PASTE
Command13=ID_NEXT_PANE
Command14=ID_PREV_PANE
CommandCount=14

```

```
[DLG:IDD_ABOUTBOX (English (U.S.))]
```

```
Type=1
```

```
ControlCount=6
```

```
Control1=IDC_STATIC,static,1342177283
```

```
Control2=IDOK,button,1342373889
```

```
Control3=IDC_STATIC,static,1342177294
```

```
Control4=IDC_STATIC,static,1342177294
```

```
Control5=IDC_STATIC,static,1342177294
```

```
Control6=IDC_STATIC,static,1342177294
```

```
[CLS:CP01Dlg]
```

```
Type=0
```

```
HeaderFile=P01Dlg.h
```

```
ImplementationFile=P01Dlg.cpp
```

```
BaseClass=CDialog
```

```
Filter=D
```

```
VirtualFilter=dWC
```

```
LastObject=CP01Dlg
```

```
[DLG:IDD_FILE_DIALOG]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Type=1
Class=CFILEDlg
ControlCount=6
Control1=IDOK,button,1342177280
Control2=IDCANCEL,button,1342177280
Control3=IDC_FILE_EDIT,edit,1350631552
Control4=IDC_STATIC,static,1342308352
Control5=IDC_STATIC,button,1342177287
Control6=IDC_STATIC,static,1342177283
```

```
[CLS:CFILEDlg]
```

```
Type=0
HeaderFile=FILEDlg.h
ImplementationFile=FILEDlg.cpp
BaseClass=CDialog
Filter=D
VirtualFilter=dWC
LastObject=CFILEDlg
```

```
[DLG:IDD_DIALOG1]
```

```
Type=1
Class=CZoomDlg
ControlCount=7
Control1=IDOK,button,1342242817
Control2=IDCANCEL,button,1342177280
Control3=IDC_NORMAL_RADIO,button,1342308361
Control4=IDC_100_RADIO,button,1342177289
Control5=IDC_200_RADIO,button,1342177289
Control6=IDC_STATIC,button,1342177287
Control7=IDC_STATIC,static,1342177283
```

```
[CLS:CZoomDlg]
```

```
Type=0
```

```
HeaderFile=ZoomDlg.h
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ImplementationFile=ZoomDlg.cpp
BaseClass=CDialog
Filter=D
LastObject=ID_VIEW_ZOOM
VirtualFilter=dWC

```

```

// P01.h : main header file for the P01 application
//
#ifdef __AFXWIN_H__
    #error include 'stdafx.h' before including this file for PCH
#endif

#include "resource.h" // main symbols

////////////////////////////////////

// CP01App:
// See P01.cpp for the implementation of this class
//

class CP01App : public CWinApp
{
public:
    CP01App();

// Overrides

```

// ClassWizard generated virtual function overrides

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//{{AFX_VIRTUAL(CP01App)
public:
virtual BOOL InitInstance();
//}}AFX_VIRTUAL

```

// Implementation

```

//{{AFX_MSG(CP01App)
afx_msg void OnAppAbout();

// NOTE - the ClassWizard will add and remove member functions here.
// DO NOT EDIT what you see in these blocks of generated code !

//}}AFX_MSG
DECLARE_MESSAGE_MAP()
};

```

```

////////////////////////////////////
// P01.cpp : Defines the class behaviors for the application.

```

```
//
```

```

#include "stdafx.h"
#include "P01.h"

```

```

#include "MainFrm.h"
#include "P01Doc.h"
#include "P01View.h"

```

```

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

```

```

////////////////////////////////////

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// CP01App

BEGIN_MESSAGE_MAP(CP01App, CWinApp)
   //{{AFX_MSG_MAP(CP01App)
    ON_COMMAND(ID_APP_ABOUT, OnAppAbout)

        // NOTE - the ClassWizard will add and remove mapping macros here.
        // DO NOT EDIT what you see in these blocks of generated code!

    /}}AFX_MSG_MAP
    // Standard file based document commands
    ON_COMMAND(ID_FILE_NEW, CWinApp::OnFileNew)
    ON_COMMAND(ID_FILE_OPEN, CWinApp::OnFileOpen)
    // Standard print setup command
    ON_COMMAND(ID_FILE_PRINT_SETUP, CWinApp::OnFilePrintSetup)
END_MESSAGE_MAP()

////////////////////////////////////
// CP01App construction

CP01App::CP01App()
{
    // TODO: add construction code here,
    // Place all significant initialization in InitInstance
}

////////////////////////////////////
// The one and only CP01App object

CP01App theApp;

////////////////////////////////////
// CP01App initialization

BOOL CP01App::InitInstance()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// Standard initialization

// If you are not using these features and wish to reduce the size
// of your final executable, you should remove from the following
// the specific initialization routines you do not need.

#ifdef _AFXDLL
    Enable3dControls(); // Call this when using MFC in a shared
DLL
#else
    Enable3dControlsStatic(); // Call this when linking to MFC statically
#endif

LoadStdProfileSettings(); // Load standard INI file options (including MRU)

// Register the application's document templates. Document templates
// serve as the connection between documents, frame windows and views.
CSingleDocTemplate* pDocTemplate;
pDocTemplate = new CSingleDocTemplate(
    IDR_MAINFRAME,
    RUNTIME_CLASS(CP01Doc),
    RUNTIME_CLASS(CMainFrame), // main SDI frame window
    RUNTIME_CLASS(CP01View));
AddDocTemplate(pDocTemplate);

// Parse command line for standard shell commands, DDE, file open
CCommandLineInfo cmdInfo;
ParseCommandLine(cmdInfo);

// Dispatch commands specified on the command line
if (!ProcessShellCommand(cmdInfo))
    return FALSE;

return TRUE;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

////////////////////////////////////

// CAboutDlg dialog used for App About

class CAboutDlg : public CDialog
{
public:
    CAboutDlg();

// Dialog Data
//{{AFX_DATA(CAboutDlg)
enum { IDD = IDD_ABOUTBOX };
//}}AFX_DATA

// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CAboutDlg)
protected:
virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV support
//}}AFX_VIRTUAL

// Implementation
protected:

//{{AFX_MSG(CAboutDlg)
// No message handlers
//}}AFX_MSG
DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
{
//{{AFX_DATA_INIT(CAboutDlg)
//}}AFX_DATA_INIT
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
   //{{AFX_DATA_MAP(CAboutDlg)
    //}}AFX_DATA_MAP
}

```

```

BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
    {{{AFX_MSG_MAP(CAboutDlg)
        // No message handlers
    }}}AFX_MSG_MAP
END_MESSAGE_MAP()

```

// App command to run the dialog

```

void CP01App::OnAppAbout()
{
    CAboutDlg aboutDlg;
    aboutDlg.DoModal();
}

```

```

////////////////////////////////////

```

// CP01App commands

Microsoft Developer Studio Generated NMAKE File, Format Version 4.20

** DO NOT EDIT **

TARGETTYPE "Win32 (x86) Application" 0x0101

!IF "\$(CFG)" == ""

CFG=P01 - Win32 Debug

!MESSAGE No configuration specified. Defaulting to P01 - Win32 Debug.

!ENDIF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

!IF "$(CFG)" != "P01 - Win32 Release" && "$(CFG)" != "P01 - Win32 Debug"
!MESSAGE Invalid configuration "$(CFG)" specified.
!MESSAGE You can specify a configuration when running NMAKE on this makefile
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "P01.mak" CFG="P01 - Win32 Debug"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "P01 - Win32 Release" (based on "Win32 (x86) Application")
!MESSAGE "P01 - Win32 Debug" (based on "Win32 (x86) Application")
!MESSAGE
!ERROR An invalid configuration is specified.
!ENDIF

!IF "$(OS)" == "Windows_NT"
NULL=
!ELSE
NULL=nul
!ENDIF

#####
#####
# Begin Project
# PROP Target_Last_Scanned "P01 - Win32 Debug"
MTL=mktyplib.exe
CPP=cl.exe
RSC=rc.exe

!IF "$(CFG)" == "P01 - Win32 Release"

# PROP BASE Use_MFC 6
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
# PROP BASE Target_Dir ""
# PROP Use_MFC 5
# PROP Use_Debug_Libraries 0
# PROP Output_Dir "Release"
# PROP Intermediate_Dir "Release"
# PROP Target_Dir ""
OUTDIR=.\Release
INTDIR=.\Release
```

```
ALL : "$(OUTDIR)\P01.exe"
```

```
CLEAN :
```

```
-@erase "$(INTDIR)\FILEDlg.obj"
-@erase "$(INTDIR)\MainFrm.obj"
-@erase "$(INTDIR)\P01.obj"
-@erase "$(INTDIR)\P01.pch"
-@erase "$(INTDIR)\P01.res"
-@erase "$(INTDIR)\P01Dlg.obj"
-@erase "$(INTDIR)\P01Doc.obj"
-@erase "$(INTDIR)\P01View.obj"
-@erase "$(INTDIR)\StdAfx.obj"
-@erase "$(INTDIR)\ZoomDlg.obj"
-@erase "$(OUTDIR)\P01.exe"
```

```
"$(OUTDIR)" :
```

```
if not exist "$(OUTDIR)/$(NULL)" mkdir "$(OUTDIR)"
```

```
# ADD BASE CPP /nologo /MD /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /D
"_AFXDLL" /D "_MBCS" /Yu"stdafx.h" /c
```

```
# ADD CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /D "_MBCS"
/Yu"stdafx.h" /c
```

```
CPP_PROJ=/nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /D
```

```
"_MBCS" /Fp"$(INTDIR)/P01.pch" /Yu"stdafx.h" /Fo"$(INTDIR)/" /c
```

```
CPP_OBJS=.\Release/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

;

```

CPP_SBRS=.\.

# ADD BASE MTL /nologo /D "NDEBUG" /win32
# ADD MTL /nologo /D "NDEBUG" /win32
MTL_PROJ=/nologo /D "NDEBUG" /win32
# ADD BASE RSC /I 0x41e /d "NDEBUG" /d "_AFXDLL"
# ADD RSC /I 0x41e /d "NDEBUG"
RSC_PROJ=/I 0x41e /fo"${INTDIR}/P01.res" /d "NDEBUG"

BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
BSC32_FLAGS=/nologo /o"${OUTDIR}/P01.bsc"
BSC32_SBRS= \

LINK32=link.exe
# ADD BASE LINK32 /nologo /subsystem:windows /machine:I386
# ADD LINK32 /nologo /subsystem:windows /machine:I386
LINK32_FLAGS=/nologo /subsystem:windows /incremental:no\
/pdb:"$(OUTDIR)/P01.pdb" /machine:I386 /out:"$(OUTDIR)/P01.exe"
LINK32_OBJS= \
    "$(INTDIR)FILEDlg.obj" \
    "$(INTDIR)MainFrm.obj" \
    "$(INTDIR)P01.obj" \
    "$(INTDIR)P01.res" \
    "$(INTDIR)P01Dlg.obj" \
    "$(INTDIR)P01Doc.obj" \
    "$(INTDIR)P01View.obj" \
    "$(INTDIR)StdAfx.obj" \
    "$(INTDIR)ZoomDlg.obj"

"$(OUTDIR)P01.exe" : "$(OUTDIR)" $(DEF_FILE) $(LINK32_OBJS)
    $(LINK32) @<<
    $(LINK32_FLAGS) $(LINK32_OBJS)
<<

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
!ELSEIF "$(CFG)" == "P01 - Win32 Debug"
```

```
# PROP BASE Use_MFC 6
```

```
# PROP BASE Use_Debug_Libraries 1
```

```
# PROP BASE Output_Dir "Debug"
```

```
# PROP BASE Intermediate_Dir "Debug"
```

```
# PROP BASE Target_Dir ""
```

```
# PROP Use_MFC 5
```

```
# PROP Use_Debug_Libraries 1
```

```
# PROP Output_Dir "Debug"
```

```
# PROP Intermediate_Dir "Debug"
```

```
# PROP Target_Dir ""
```

```
OUTDIR=.\\Debug
```

```
INTDIR=.\\Debug
```

```
ALL : "$(OUTDIR)\P01.exe"
```

```
CLEAN :
```

```
-@erase "$(INTDIR)\FILEDlg.obj"
```

```
-@erase "$(INTDIR)\MainFrm.obj"
```

```
-@erase "$(INTDIR)\P01.obj"
```

```
-@erase "$(INTDIR)\P01.pch"
```

```
-@erase "$(INTDIR)\P01.res"
```

```
-@erase "$(INTDIR)\P01Dlg.obj"
```

```
-@erase "$(INTDIR)\P01Doc.obj"
```

```
-@erase "$(INTDIR)\P01View.obj"
```

```
-@erase "$(INTDIR)\StdAfx.obj"
```

```
-@erase "$(INTDIR)\vc40.idb"
```

```
-@erase "$(INTDIR)\vc40.pdb"
```

```
-@erase "$(INTDIR)\ZoomDlg.obj"
```

```
-@erase "$(OUTDIR)\P01.exe"
```

```
-@erase "$(OUTDIR)\P01.ilc"
```

```
-@erase "$(OUTDIR)\P01.pdb"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

"\$(OUTDIR)" :

```

if not exist "$(OUTDIR)/$(NULL)" mkdir "$(OUTDIR)"

# ADD BASE CPP /nologo /MDd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D
"_WINDOWS" /D "_AFXDLL" /D "_MBCS" /Yu"stdafx.h" /c
# ADD CPP /nologo /MTd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /D
"_MBCS" /Yu"stdafx.h" /c
CPP_PROJ=/nologo /MTd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" \
/D "_MBCS" /Fp"$(INTDIR)/P01.pch" /Yu"stdafx.h" /Fo"$(INTDIR)/" /Fd"$(INTDIR)/\
/c
CPP_OBJS=.\Debug/
CPP_SBRS=.\
# ADD BASE MTL /nologo /D "_DEBUG" /win32
# ADD MTL /nologo /D "_DEBUG" /win32
MTL_PROJ=/nologo /D "_DEBUG" /win32
# ADD BASE RSC /I 0x41e /d "_DEBUG" /d "_AFXDLL"
# ADD RSC /I 0x41e /d "_DEBUG"
RSC_PROJ=/I 0x41e /fo"$(INTDIR)/P01.res" /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
BSC32_FLAGS=/nologo /o"$(OUTDIR)/P01.bsc"
BSC32_SBRS= \

LINK32=link.exe
# ADD BASE LINK32 /nologo /subsystem:windows /debug /machine:I386
# ADD LINK32 /nologo /subsystem:windows /debug /machine:I386
LINK32_FLAGS=/nologo /subsystem:windows /incremental:yes\
/pdb:"$(OUTDIR)/P01.pdb" /debug /machine:I386 /out:"$(OUTDIR)/P01.exe"
LINK32_OBJS= \
    "$(INTDIR)\FILEDlg.obj" \
    "$(INTDIR)\MainFrm.obj" \
    "$(INTDIR)\P01.obj" \
    "$(INTDIR)\P01.res" \

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
"$(INTDIR)\P01Dlg.obj" \
"$(INTDIR)\P01Doc.obj" \
"$(INTDIR)\P01View.obj" \
"$(INTDIR)\StdAfx.obj" \
"$(INTDIR)\ZoomDlg.obj"
```

```
"$(OUTDIR)\P01.exe" : "$(OUTDIR)" $(DEF_FILE) $(LINK32_OBJS)
```

```
$(LINK32) @<<
```

```
$(LINK32_FLAGS) $(LINK32_OBJS)
```

```
<<
```

```
!ENDIF
```

```
.c{$(CPP_OBJS)}.obj:
```

```
$(CPP) $(CPP_PROJ) $<
```

```
.cpp{$(CPP_OBJS)}.obj:
```

```
$(CPP) $(CPP_PROJ) $<
```

```
.cxx{$(CPP_OBJS)}.obj:
```

```
$(CPP) $(CPP_PROJ) $<
```

```
.c{$(CPP_SBRs)}.sbr:
```

```
$(CPP) $(CPP_PROJ) $<
```

```
.cpp{$(CPP_SBRs)}.sbr:
```

```
$(CPP) $(CPP_PROJ) $<
```

```
.cxx{$(CPP_SBRs)}.sbr:
```

```
$(CPP) $(CPP_PROJ) $<
```

```
#####
```

```
#####
```

```
# Begin Target
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
# Name "P01 - Win32 Release"
```

```
# Name "P01 - Win32 Debug"
```

```
!IF "$(CFG)" == "P01 - Win32 Release"
```

```
!ELSEIF "$(CFG)" == "P01 - Win32 Debug"
```

```
!ENDIF
```

```
#####
```

```
#####
```

```
# Begin Source File
```

```
SOURCE=.\ReadMe.txt
```

```
!IF "$(CFG)" == "P01 - Win32 Release"
```

```
!ELSEIF "$(CFG)" == "P01 - Win32 Debug"
```

```
!ENDIF
```

```
# End Source File
```

```
#####
```

```
#####
```

```
# Begin Source File
```

```
SOURCE=.\P01.cpp
```

```
DEP_CPP_P01_C=\
```

```
    ".\MainFrm.h\"
```

```
    ".\P01.h\"
```

```
    ".\P01Doc.h\"
```

```
    ".\P01View.h\"
```

```
    ".\StdAfx.h\"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

"$(INTDIR)\P01.obj" : $(SOURCE) $(DEP_CPP_P01_C) "$(INTDIR)\
"$(INTDIR)\P01.pch"

# End Source File
#####
#####
# Begin Source File

SOURCE=. \StdAfx.cpp
DEP_CPP_STDAF=\
    ".\StdAfx.h"

!!IF "$(CFG)" == "P01 - Win32 Release"

# ADD CPP /Yc"stdafx.h"

BuildCmnds= \
    $(CPP) /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /D
    "_MBCS" \
    /Fp"$(INTDIR)\P01.pch" /Yc"stdafx.h" /Fo"$(INTDIR)\\"

"$(INTDIR)\StdAfx.obj" : $(SOURCE) $(DEP_CPP_STDAF) "$(INTDIR)\
    $(BuildCmnds)

"$(INTDIR)\P01.pch" : $(SOURCE) $(DEP_CPP_STDAF) "$(INTDIR)\
    $(BuildCmnds)

!ELSEIF "$(CFG)" == "P01 - Win32 Debug"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
# ADD CPP /Yc"stdafx.h"
```

```
BuildCmds= \
```

```
    $(CPP) /nologo /MTd /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D
    "_WINDOWS"
    /D "_MBCS" /Fp"$(INTDIR)/P01.pch" /Yc"stdafx.h" /Fo"$(INTDIR)/" /Fd"$(INTDIR)/\
    /c $(SOURCE) \
```

```
"$(INTDIR)\StdAfx.obj" : $(SOURCE) $(DEP_CPP_STDAF) "$(INTDIR)"
    $(BuildCmds)
```

```
"$(INTDIR)\P01.pch" : $(SOURCE) $(DEP_CPP_STDAF) "$(INTDIR)"
    $(BuildCmds)
```

```
!ENDIF
```

```
# End Source File
```

```
#####
```

```
#####
```

```
# Begin Source File
```

```
SOURCE=.MainFrm.cpp
```

```
DEP_CPP_MAINF=\
```

```
    ".MainFrm.h"
```

```
    ".\P01.h"
```

```
    ".\StdAfx.h"
```

```
"$(INTDIR)\MainFrm.obj" : $(SOURCE) $(DEP_CPP_MAINF) "$(INTDIR)"
```

```
"$(INTDIR)\P01.pch"
```

```
# End Source File
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#####
####
# Begin Source File

SOURCE=.\\P01Doc.cpp
DEP_CPP_P01DO=\\
    ".\\P01.h\\"
    ".\\P01Doc.h\\"
    ".\\StdAfx.h\\"

"$(INTDIR)\\P01Doc.obj" : $(SOURCE) $(DEP_CPP_P01DO) "$(INTDIR)\\
"$(INTDIR)\\P01.pch"

# End Source File
#####
####
# Begin Source File

SOURCE=.\\P01View.cpp
DEP_CPP_P01VI=\\
    ".\\Filedlg.h\\"
    ".\\P01.h\\"
    ".\\P01dlg.h\\"
    ".\\P01Doc.h\\"
    ".\\P01View.h\\"
    ".\\StdAfx.h\\"
    ".\\Zoomdlg.h\\"

"$(INTDIR)\\P01View.obj" : $(SOURCE) $(DEP_CPP_P01VI) "$(INTDIR)\\
"$(INTDIR)\\P01.pch"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
# End Source File
```

```
#####
```

```
#####
```

```
# Begin Source File
```

```
SOURCE=.IP01.rc
```

```
DEP_RSC_P01_R=\
```

```
    ".\res\advisor.bmp"
```

```
    ".\res\Face.ico"
```

```
    ".\res\name.bmp"
```

```
    ".\res\P01.ico"
```

```
    ".\res\P01.rc2"
```

```
    ".\res\P01Doc.ico"
```

```
    ".\res\project2.bmp"
```

```
    ".\res\Raedit.ico"
```

```
    ".\res\SecCast.ico"
```

```
    ".\res\TITLE.bmp"
```

```
    ".\res\Toolbar.bmp"
```

```
"$(INTDIR)\P01.res" : $(SOURCE) $(DEP_RSC_P01_R) "$(INTDIR)"
```

```
    $(RSC) $(RSC_PROJ) $(SOURCE)
```

```
# End Source File
```

```
#####
```

```
#####
```

```
# Begin Source File
```

```
SOURCE=.IP01Dlg.cpp
```

```
DEP_CPP_P01DL=\
```

```
    ".\P01.h"
```

```
    ".\P01dlg.h"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
".\StdAfx.h"
```

```
"$(INTDIR)\P01Dlg.obj" : $(SOURCE) $(DEP_CPP_P01DL) "$(INTDIR)\  
"$(INTDIR)\P01.pch"
```

```
# End Source File
```

```
#####
```

```
#####
```

```
# Begin Source File
```

```
SOURCE=.\FILEDlg.cpp
```

```
DEP_CPP_FILED=\  
".\FileDlg.h"
```

```
"\P01.h"
```

```
"\StdAfx.h"
```

```
"$(INTDIR)\FILEDlg.obj" : $(SOURCE) $(DEP_CPP_FILED) "$(INTDIR)\  
"$(INTDIR)\P01.pch"
```

```
"$(INTDIR)\P01.pch"
```

```
# End Source File
```

```
#####
```

```
#####
```

```
# Begin Source File
```

```
SOURCE=.\ZoomDlg.cpp
```

```
DEP_CPP_ZOOMD=\  
"\P01.h"
```

```
"\StdAfx.h"
```

```
"\Zoomdlg.h"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
"$(INTDIR)\ZoomDlg.obj" : $(SOURCE) $(DEP_CPP_ZOOMD) "$(INTDIR)\P01.pch"
```

```
# End Source File
```

```
# End Target
```

```
# End Project
```

```
#####
```

```
#####
```

```
//Microsoft Developer Studio generated resource script.
```

```
//
```

```
#include "resource.h"
```

```
#define APSTUDIO_READONLY_SYMBOLS
```

```
////////////////////////////////////
```

```
//
```

```
// Generated from the TEXTINCLUDE 2 resource.
```

```
//
```

```
#include "afxres.h"
```

```
////////////////////////////////////
```

```
#undef APSTUDIO_READONLY_SYMBOLS
```

```
////////////////////////////////////
```

```
// Unknown language: 0x1E, 0x1 resources
```

```
#if !defined(AFX_RESOURCE_DLL) || defined(AFX_TARG_THA)
```

```
#ifdef _WIN32
```

```
LANGUAGE 0x1E, 0x1
```

```
#pragma code_page(874)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#endif // _WIN32

#ifdef APSTUDIO_INVOKED
////////////////////////////////////
//
// TEXTINCLUDE
//

1 TEXTINCLUDE DISCARDABLE
BEGIN
    "resource.h\0"
END

2 TEXTINCLUDE DISCARDABLE
BEGIN
    "#include ""afxres.h""\r\n"
    "\0"
END

3 TEXTINCLUDE DISCARDABLE
BEGIN
    "#define _AFX_NO_SPLITTER_RESOURCES\r\n"
    "#define _AFX_NO_OLE_RESOURCES\r\n"
    "#define _AFX_NO_TRACKER_RESOURCES\r\n"
    "#define _AFX_NO_PROPERTY_RESOURCES\r\n"
    "\r\n"
    "#if !defined(AFX_RESOURCE_DLL) || defined(AFX_TARG_ENU)\r\n"
    "#ifdef _WIN32\r\n"
    "LANGUAGE 9, 1\r\n"
    "#pragma code_page(1252)\r\n"
    "#endif\r\n"
    "#include ""res\p01.rc2"" // non-Microsoft Visual C++ edited resources\r\n"
    "#include ""afxres.rc"" // Standard components\r\n"
    "#include ""afxprint.rc"" // printing/print preview resources\r\n"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

"#endif\0"
END

#endif // APSTUDIO_INVOKED

////////////////////////////////////
//
// Icon
//

// Icon with lowest ID value placed first to ensure application icon
// remains consistent on all systems.
IDR_MAINFRAME      ICON    DISCARDABLE    "res\WP01.ico"
IDI_ICON1          ICON    DISCARDABLE    "res\SecCast.ico"
IDI_FACE_ICON      ICON    DISCARDABLE    "res\Face.ico"
IDI_RED_ICON       ICON    DISCARDABLE    "res\Raedit.ico"

////////////////////////////////////
//
// Dialog

IDD_INFO_DIALOG   DIALOG DISCARDABLE  0, 0, 283, 85
STYLE DS_MODALFRAME | DS_CENTER | WS_POPUP | WS_CAPTION | WS_SYSMENU
CAPTION "Infomation"
FONT 8, "MS Sans Serif"
BEGIN
    EDITTEXT      IDC_NODE_EDIT,41,10,65,14,ES_UPPERCASE | ES_AUTOHSCROLL |
                ES_READONLY | NOT WS_BORDER | NOT WS_TABSTOP
    LTEXT         "Input",IDC_INPUT_STATIC,125,10,23,8
    LTEXT         "Link To ...",IDC_LINK_STATIC,8,40,33,8
    EDITTEXT      IDC_NODE1_EDIT,41,40,38,14,ES_UPPERCASE | ES_AUTOHSCROLL |
                ES_READONLY | NOT WS_BORDER | NOT WS_TABSTOP

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

EDITTEXT IDC_INPUT_EDIT,155,10,120,14,ES_AUTOHSCROLL |
ES_READONLY | NOT WS_BORDER | NOT WS_TABSTOP
EDITTEXT IDC_NODE2_EDIT,80,40,38,15,ES_UPPERCASE | ES_AUTOHSCROLL |
ES_READONLY | NOT WS_BORDER | NOT WS_TABSTOP
LTEXT "DISC",IDC_STATIC,8,55,18,8,WS_TABSTOP
EDITTEXT IDC_D_EDIT,41,55,50,14,ES_AUTOHSCROLL | ES_READONLY |
NOT WS_BORDER | NOT WS_TABSTOP
GROUPBOX "",IDC_STATIC,2,0,279,83
ICON IDI_ICON1,IDC_STATIC,8,7,18,20
EDITTEXT IDC_LINK1_EDIT,125,55,40,14,ES_AUTOHSCROLL | ES_READONLY |
NOT WS_BORDER
EDITTEXT IDC_LINK2_EDIT,170,55,40,14,ES_AUTOHSCROLL | ES_READONLY |
NOT WS_BORDER
EDITTEXT IDC_NODE3_EDIT,119,40,38,15,ES_UPPERCASE |
ES_AUTOHSCROLL | ES_READONLY | NOT WS_BORDER | NOT
WS_TABSTOP
EDITTEXT IDC_NODE4_EDIT,158,40,38,15,ES_UPPERCASE |
ES_AUTOHSCROLL | ES_READONLY | NOT WS_BORDER | NOT
WS_TABSTOP
EDITTEXT IDC_NODE5_EDIT,197,40,38,15,ES_UPPERCASE |
ES_AUTOHSCROLL | ES_READONLY | NOT WS_BORDER | NOT
WS_TABSTOP
EDITTEXT IDC_NODE6_EDIT,236,40,38,15,ES_UPPERCASE |
ES_AUTOHSCROLL | ES_READONLY | NOT WS_BORDER | NOT
WS_TABSTOP

```

END

```
IDD_FILE_DIALOG DIALOG DISCARDABLE 0, 0, 186, 80
```

```
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION | WS_SYSMENU
```

```
CAPTION "FILE NAME"
```

```
FONT 8, "MS Sans Serif"
```

```
BEGIN
```

```
PUSHBUTTON "OK",IDOK,62,53,50,14,NOT WS_TABSTOP
```

```
PUSHBUTTON "Cancel",IDCANCEL,126,53,50,14,NOT WS_TABSTOP
```

เอกสารนี้เป็นเอกสารทสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

EDITTEXT      IDC_FILE_EDIT,78,23,90,14,ES_AUTOHSCROLL
LTEXT         "FILE NAME ->",IDC_STATIC,17,26,47,8
GROUPBOX     "",IDC_STATIC,2,0,182,78
ICON         IDI_RED_ICON,IDC_STATIC,19,47,21,18
END

IDD_DIALOG1 DIALOG DISCARDABLE 0, 0, 159, 91
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION
CAPTION "Zoom"
FONT 8, "MS Sans Serif"
BEGIN
    DEFPUSHBUTTON "OK",IDOK,85,32,50,14
    PUSHBUTTON    "Cancel",IDCANCEL,85,59,50,14,NOT WS_TABSTOP
    CONTROL       "Normal",IDC_NORMAL_RADIO,"Button",BS_AUTORADIOBUTTON |
        WS_GROUP,23,35,50,10
    CONTROL       "x 3",IDC_100_RADIO,"Button",BS_AUTORADIOBUTTON,23,53,50,
        10
    CONTROL       "x 5",IDC_200_RADIO,"Button",BS_AUTORADIOBUTTON,23,71,50,
        10
    GROUPBOX     "",IDC_STATIC,2,0,155,89
    ICON         IDI_ICON1,IDC_STATIC,8,9,18,20
END

```

```

////////////////////////////////////

```

```
//
```

```
// DESIGNINFO
```

```
//
```

```
#ifdef APSTUDIO_INVOKED
```

```
GUIDELINES DESIGNINFO DISCARDABLE
```

```
BEGIN
```

```
    IDD_INFO_DIALOG, DIALOG
```

```
    BEGIN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LEFTMARGIN, 2
RIGHTMARGIN, 281
BOTTOMMARGIN, 83
END

IDD_FILE_DIALOG, DIALOG
BEGIN
LEFTMARGIN, 2
RIGHTMARGIN, 184
BOTTOMMARGIN, 78
END

IDD_DIALOG1, DIALOG
BEGIN
LEFTMARGIN, 2
RIGHTMARGIN, 157
BOTTOMMARGIN, 89
END
END
#endif // APSTUDIO_INVOKED

////////////////////////////////////
//
// Bitmap
//

IDB_PRO2_BITMAP BITMAP DISCARDABLE "res\project2.bmp"
IDB_NAME_BITMAP BITMAP DISCARDABLE "res\name.bmp"
IDB_ADVIS_BITMAP BITMAP DISCARDABLE "res\advisor.bmp"
IDB_TITLE_BITMAP BITMAP DISCARDABLE "res\TITLE.bmp"
#endif // Unknown language: 0x1E, 0x1 resources

////////////////////////////////////

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

////////////////////////////////////
// English (U.S.) resources

#ifdef AFX_RESOURCE_DLL || defined(AFX_TARG_ENU)
#ifdef _WIN32
LANGUAGE LANG_ENGLISH, SUBLANG_ENGLISH_US
#pragma code_page(1252)
#endif // _WIN32

////////////////////////////////////
//
// Icon
//
// Icon with lowest ID value placed first to ensure application icon
// remains consistent on all systems.
IDR_P01TYPE ICON DISCARDABLE "res\IP01Doc.ico"

////////////////////////////////////
//
// Dialog
//

IDD_ABOUTBOX DIALOGEX 0, 0, 289, 158
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION | WS_SYSMENU
EXSTYLE WS_EX_CLIENTEDGE | WS_EX_STATICEDGE
CAPTION "About Project II"
FONT 8, "MS Sans Serif"
BEGIN
    ICON        IDI_RED_ICON, IDC_STATIC, 13, 11, 18, 20
    DEFPUSHBUTTON "OK", IDOK, 196, 104, 81, 43, WS_GROUP
    CONTROL     136, IDC_STATIC, "Static", SS_BITMAP, 35, 9, 90, 19
    CONTROL     137, IDC_STATIC, "Static", SS_BITMAP, 83, 61, 147, 35

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CONTROL      141, IDC_STATIC, "Static", SS_BITMAP, 37, 27, 239, 32
CONTROL      140, IDC_STATIC, "Static", SS_BITMAP, 83, 97, 91, 50
END

```

```

/////////////////////////////////////////////////////////////////

```

```
//
```

```
// DESIGNINFO
```

```
//
```

```

#ifdef APSTUDIO_INVOKED
GUIDELINES DESIGNINFO DISCARDABLE
BEGIN
  IDD_ABOUTBOX, DIALOG
  BEGIN
    LEFTMARGIN, 3
    RIGHTMARGIN, 286
    BOTTOMMARGIN, 155
  END
END
#endif // APSTUDIO_INVOKED

```

```

/////////////////////////////////////////////////////////////////

```

```
//
```

```
// Bitmap
```

```
//
```

```
IDR_MAINFRAME      BITMAP MOVEABLE PURE "res\\Toolbar.bmp"
```

```

/////////////////////////////////////////////////////////////////

```

```
//
```

```
// Toolbar
```

```
//
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IDR_MAINFRAME TOOLBAR DISCARDABLE 16, 15

BEGIN

 BUTTON ID_FILE_NEW

 BUTTON ID_FILE_OPEN

 BUTTON ID_FILE_SAVE

 SEPARATOR

 BUTTON ID_EDIT_CUT

 BUTTON ID_EDIT_COPY

 BUTTON ID_EDIT_PASTE

 SEPARATOR

 BUTTON ID_FILE_PRINT

 BUTTON ID_APP_ABOUT

 BUTTON ID_VIEW_ZOOM

END

////////////////////////////////////

//

// Menu

//

IDR_MAINFRAME MENU PRELOAD DISCARDABLE

BEGIN

 POPUP "&File"

 BEGIN

 MENUITEM "&New\tCtrl+N", ID_FILE_NEW

 MENUITEM "&Open...\tCtrl+O", ID_FILE_OPEN

 MENUITEM "&Save\tCtrl+S", ID_FILE_SAVE

 MENUITEM "Save &As...", ID_FILE_SAVE_AS

 MENUITEM SEPARATOR

 MENUITEM "&Print...\tCtrl+P", ID_FILE_PRINT

 MENUITEM "Print Pre&view", ID_FILE_PRINT_PREVIEW

 MENUITEM "P&rint Setup...", ID_FILE_PRINT_SETUP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MENUITEM SEPARATOR
MENUITEM "Recent File",          ID_FILE_MRU_FILE1, GRAYED
MENUITEM SEPARATOR
MENUITEM "E&xit",                ID_APP_EXIT
END
POPUP "&Edit"
BEGIN
    MENUITEM "&Undo\tCtrl+Z",      ID_EDIT_UNDO
    MENUITEM SEPARATOR
    MENUITEM "Cu&t\tCtrl+X",       ID_EDIT_CUT
    MENUITEM "&Copy\tCtrl+C",      ID_EDIT_COPY
    MENUITEM "&Paste\tCtrl+V",     ID_EDIT_PASTE
END
POPUP "&View"
BEGIN
    MENUITEM "&Toolbar",           ID_VIEW_TOOLBAR
    MENUITEM "&Status Bar",       ID_VIEW_STATUS_BAR
    MENUITEM "&Zoom",             ID_VIEW_ZOOM
END
POPUP "&Help"
BEGIN
    MENUITEM "&About P01...",     ID_APP_ABOUT
END
END

```

```

////////////////////////////////////

```

```
//
```

```
// Accelerator
```

```
//
```

```
IDR_MAINFRAME ACCELERATORS PRELOAD MOVEABLE PURE
```

```
BEGIN
```

```
    "N", ID_FILE_NEW, VIRTKEY, CONTROL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

"O",      ID_FILE_OPEN,      VIRTKEY, CONTROL
"S",      ID_FILE_SAVE,      VIRTKEY, CONTROL
"P",      ID_FILE_PRINT,     VIRTKEY, CONTROL
"Z",      ID_EDIT_UNDO,     VIRTKEY, CONTROL
"X",      ID_EDIT_CUT,       VIRTKEY, CONTROL
"C",      ID_EDIT_COPY,     VIRTKEY, CONTROL
"V",      ID_EDIT_PASTE,    VIRTKEY, CONTROL

VK_BACK,  ID_EDIT_UNDO,     VIRTKEY, ALT
VK_DELETE, ID_EDIT_CUT,     VIRTKEY, SHIFT
VK_INSERT, ID_EDIT_COPY,    VIRTKEY, CONTROL
VK_INSERT, ID_EDIT_PASTE,   VIRTKEY, SHIFT
VK_F6,    ID_NEXT_PANE,     VIRTKEY
VK_F6,    ID_PREV_PANE,     VIRTKEY, SHIFT
END

#ifdef _MAC
////////////////////////////////////
//
// Version
//

VS_VERSION_INFO VERSIONINFO
FILEVERSION 1,0,0,1
PRODUCTVERSION 1,0,0,1
FILEFLAGSMASK 0x3fL
#ifdef _DEBUG
FILEFLAGS 0x1L
#else
FILEFLAGS 0x0L
#endif
FILEOS 0x4L
FILETYPE 0x1L
FILESUBTYPE 0x0L

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

BEGIN
    BLOCK "StringFileInfo"
    BEGIN
        BLOCK "040904B0"
        BEGIN
            VALUE "CompanyName", "\0"
            VALUE "FileDescription", "P01 MFC Application\0"
            VALUE "FileVersion", "1, 0, 0, 1\0"
            VALUE "InternalName", "P01\0"
            VALUE "LegalCopyright", "Copyright © 1997\0"
            VALUE "LegalTrademarks", "\0"
            VALUE "OriginalFilename", "P01.EXE\0"
            VALUE "ProductName", "P01 Application\0"
            VALUE "ProductVersion", "1, 0, 0, 1\0"
        END
    END
    BLOCK "VarFileInfo"
    BEGIN
        VALUE "Translation", 0x409, 1200
    END
END

#endif // !_MAC

////////////////////////////////////
//
// String Table
//

STRINGTABLE PRELOAD DISCARDABLE
BEGIN
    IDR_MAINFRAME                "Neural Structure\n Windows Programming using
    MSVC++\nP01\nNER Files (*.ner)\n.NER\nP01.Document\nP01 Document"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

END

STRINGTABLE PRELOAD DISCARDABLE

BEGIN

AFX_IDS_APP_TITLE "P01"

AFX_IDS_IDLEMESSAGE "Ready"

END

STRINGTABLE DISCARDABLE

BEGIN

ID_INDICATOR_EXT "EXT"

ID_INDICATOR_CAPS "CAP"

ID_INDICATOR_NUM "NUM"

ID_INDICATOR_SCRL "SCRL"

ID_INDICATOR_OVR "OVR"

ID_INDICATOR_REC "REC"

END

STRINGTABLE DISCARDABLE

BEGIN

ID_FILE_NEW "Create a new document\nNew"

ID_FILE_OPEN "Open an existing document\nOpen"

ID_FILE_CLOSE "Close the active document\nClose"

ID_FILE_SAVE "Save the active document\nSave"

ID_FILE_SAVE_AS "Save the active document with a new name\nSave As"

ID_FILE_PAGE_SETUP "Change the printing options\nPage Setup"

ID_FILE_PRINT_SETUP "Change the printer and printing options\nPrint Setup"

ID_FILE_PRINT "Print the active document\nPrint"

ID_FILE_PRINT_PREVIEW "Display full pages\nPrint Preview"

END

STRINGTABLE DISCARDABLE

BEGIN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ID_APP_ABOUT          "Display program information, version number and
copyright\nAbout"
ID_APP_EXIT           "Quit the application; prompts to save documents\nExit"
END

```

STRINGTABLE DISCARDABLE

BEGIN

```

ID_FILE_MRU_FILE1    "Open this document"
ID_FILE_MRU_FILE2    "Open this document"
ID_FILE_MRU_FILE3    "Open this document"
ID_FILE_MRU_FILE4    "Open this document"
ID_FILE_MRU_FILE5    "Open this document"
ID_FILE_MRU_FILE6    "Open this document"
ID_FILE_MRU_FILE7    "Open this document"
ID_FILE_MRU_FILE8    "Open this document"
ID_FILE_MRU_FILE9    "Open this document"
ID_FILE_MRU_FILE10   "Open this document"
ID_FILE_MRU_FILE11   "Open this document"
ID_FILE_MRU_FILE12   "Open this document"
ID_FILE_MRU_FILE13   "Open this document"
ID_FILE_MRU_FILE14   "Open this document"
ID_FILE_MRU_FILE15   "Open this document"
ID_FILE_MRU_FILE16   "Open this document"

```

END

STRINGTABLE DISCARDABLE

BEGIN

```

ID_NEXT_PANE         "Switch to the next window pane\nNext Pane"
ID_PREV_PANE         "Switch back to the previous window pane\nPrevious Pane"

```

END

STRINGTABLE DISCARDABLE

BEGIN

```

ID_WINDOW_SPLIT      "Split the active window into panes\nSplit"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

END

STRINGTABLE DISCARDABLE

BEGIN

ID_EDIT_CLEAR "Erase the selection\nErase"
 ID_EDIT_CLEAR_ALL "Erase everything\nErase All"
 ID_EDIT_COPY "Copy the selection and put it on the Clipboard\nCopy"
 ID_EDIT_CUT "Cut the selection and put it on the Clipboard\nCut"
 ID_EDIT_FIND "Find the specified text\nFind"
 ID_EDIT_PASTE "Insert Clipboard contents\nPaste"
 ID_EDIT_REPEAT "Repeat the last action\nRepeat"
 ID_EDIT_REPLACE "Replace specific text with different text\nReplace"
 ID_EDIT_SELECT_ALL "Select the entire document\nSelect All"
 ID_EDIT_UNDO "Undo the last action\nUndo"
 ID_EDIT_REDO "Redo the previously undone action\nRedo"

END

STRINGTABLE DISCARDABLE

BEGIN

ID_VIEW_TOOLBAR "Show or hide the toolbar\nToggle ToolBar"
 ID_VIEW_STATUS_BAR "Show or hide the status bar\nToggle StatusBar"

END

STRINGTABLE DISCARDABLE

BEGIN

AFX_IDS_SCSIZE "Change the window size"
 AFX_IDS_SCMOVE "Change the window position"
 AFX_IDS_SCMINIMIZE "Reduce the window to an icon"
 AFX_IDS_SCMAXIMIZE "Enlarge the window to full size"
 AFX_IDS_SCNEXTWINDOW "Switch to the next document window"
 AFX_IDS_SCPREVWINDOW "Switch to the previous document window"
 AFX_IDS_SCCLOSE "Close the active window and prompts to save the documents"

END

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
STRINGTABLE DISCARDABLE
```

```
BEGIN
```

```
    AFX_IDS_SCRESTORE    "Restore the window to normal size"
```

```
    AFX_IDS_SCTASKLIST  "Activate Task List"
```

```
END
```

```
STRINGTABLE DISCARDABLE
```

```
BEGIN
```

```
    AFX_IDS_PREVIEW_CLOSE "Close print preview mode\nCancel Preview"
```

```
END
```

```
STRINGTABLE DISCARDABLE
```

```
BEGIN
```

```
    ID_VIEW_ZOOM        "Zoom in-out\nZoom"
```

```
END
```

```
#endif // English (U.S.) resources
```

```
////////////////////////////////////
```

```
#ifndef APSTUDIO_INVOKED
```

```
////////////////////////////////////
```

```
//
```

```
// Generated from the TEXTINCLUDE 3 resource.
```

```
//
```

```
#define _AFX_NO_SPLITTER_RESOURCES
```

```
#define _AFX_NO_OLE_RESOURCES
```

```
#define _AFX_NO_TRACKER_RESOURCES
```

```
#define _AFX_NO_PROPERTY_RESOURCES
```

```
#if !defined(AFX_RESOURCE_DLL) || defined(AFX_TARG_ENU)
```

```
#ifdef _WIN32
```

```
    LANGUAGE 9, 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#pragma code_page(1252)

#endif

#include "res\P01.rc2" // non-Microsoft Visual C++ edited resources
#include "afxres.rc"    // Standard components
#include "afxprint.rc"  // printing/print preview resources

#endif

////////////////////////////////////

#endif // not APSTUDIO_INVOKED

// P01Dlg.h : header file
//

////////////////////////////////////

// CP01Dlg dialog

class CP01Dlg : public CDialog
{
// Construction
public:
    CP01Dlg(CWnd* pParent = NULL); // standard constructor

// Dialog Data
//{{AFX_DATA(CP01Dlg)
enum { IDD = IDD_INFO_DIALOG };

float    m_dist;

int      m_input;

CString m_node;
CString m_node1;
CString m_node2;

int      m_link1;
int      m_link2;

CString m_node3;
CString m_node4;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        CString m_node5;
        CString m_node6;
    //}}AFX_DATA

// Overrides
    // ClassWizard generated virtual function overrides
    //{AFX_VIRTUAL(CP01Dlg)
    protected:
    virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV support
    //}}AFX_VIRTUAL

// Implementation
protected:
    // Generated message map functions
    //{AFX_MSG(CP01Dlg)
    // NOTE: the ClassWizard will add member functions here
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};
// P01Dlg.cpp : implementation file
//

#include "stdafx.h"
#include "P01.h"
#include "P01Dlg.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

////////////////////////////////////
// CP01Dlg dialog

CP01Dlg::CP01Dlg(CWnd* pParent /*=NULL*/)
    : CDialog(CP01Dlg::IDD, pParent)
{
   //{{AFX_DATA_INIT(CP01Dlg)
    m_dist = 0.0f;
    m_input = 0;
    m_node = _T("");
    m_node1 = _T("");
    m_node2 = _T("");
    m_link1 = 0;
    m_link2 = 0;
    m_node3 = _T("");
    m_node4 = _T("");
    m_node5 = _T("");
    m_node6 = _T("");
    //}}AFX_DATA_INIT
}

void CP01Dlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
    //{{AFX_DATA_MAP(CP01Dlg)
    DDX_Text(pDX, IDC_D_EDIT, m_dist);
    DDX_Text(pDX, IDC_INPUT_EDIT, m_input);
    DDX_Text(pDX, IDC_NODE_EDIT, m_node);
    DDX_Text(pDX, IDC_NODE1_EDIT, m_node1);
    DDX_Text(pDX, IDC_NODE2_EDIT, m_node2);
    DDX_Text(pDX, IDC_LINK1_EDIT, m_link1);
    DDX_Text(pDX, IDC_LINK2_EDIT, m_link2);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DDX_Text(pDX, IDC_NODE3_EDIT, m_node3);
DDX_Text(pDX, IDC_NODE4_EDIT, m_node4);
DDX_Text(pDX, IDC_NODE5_EDIT, m_node5);
DDX_Text(pDX, IDC_NODE6_EDIT, m_node6);
//{{AFX_DATA_MAP
}

```

```

BEGIN_MESSAGE_MAP(CP01Dlg, CDialog)
    {{{AFX_MSG_MAP(CP01Dlg)
        // NOTE: the ClassWizard will add message map macros here
    }}}AFX_MSG_MAP
END_MESSAGE_MAP()

```

```

////////////////////////////////////

```

```

// CP01Dlg message handlers

```

```

// P01Doc.h : interface of the CP01Doc class

```

```

//

```

```

////////////////////////////////////

```

```

class CP01Doc : public CDocument

```

```

{

```

```

protected: // create from serialization only

```

```

    CP01Doc();

```

```

    DECLARE_DYNCREATE(CP01Doc)

```

```

// Attributes

```

```

public:

```

```

// Operations

```

```

public:

```

```

// Overrides

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CP01Doc)
public:
    virtual BOOL OnNewDocument();
    virtual void Serialize(CArchive& ar);
//}}AFX_VIRTUAL

// Implementation
public:
    virtual ~CP01Doc();
#ifdef _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
#endif
protected:

// Generated message map functions
protected:
   //{{AFX_MSG(CP01Doc)
        // NOTE - the ClassWizard will add and remove member functions here.
        // DO NOT EDIT what you see in these blocks of generated code !
   //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};

////////////////////////////////////

// P01Doc.cpp : implementation of the CP01Doc class
//

#include "stdafx.h"
#include "P01.h"

```

```
#include "P01Doc.h"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

////////////////////////////////////

// CP01Doc

IMPLEMENT_DYNCREATE(CP01Doc, CDocument)

BEGIN_MESSAGE_MAP(CP01Doc, CDocument)
//{{AFX_MSG_MAP(CP01Doc)
// NOTE - the ClassWizard will add and remove mapping macros here.
// DO NOT EDIT what you see in these blocks of generated code!
//}}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////

// CP01Doc construction/destruction

CP01Doc::CP01Doc()
{
    // TODO: add one-time construction code here

}

CP01Doc::~CP01Doc()
{

}

BOOL CP01Doc::OnNewDocument()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (!CDocument::OnNewDocument())
    return FALSE;

// TODO: add reinitialization code here
// (SDI documents will reuse this document)

return TRUE;
}

```

```

////////////////////////////////////
// CP01Doc serialization

```

```

void CP01Doc::Serialize(CArchive& ar)
{
    if (ar.IsStoring())
    {
        // TODO: add storing code here
    }
    else
    {
        // TODO: add loading code here
    }
}

```

```

////////////////////////////////////
// CP01Doc diagnostics

```

```

#ifdef _DEBUG
void CP01Doc::AssertValid() const
{
    CDocument::AssertValid();
}

```

```

void CP01Doc::Dump(CDumpContext& dc) const

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    CDocument::Dump(dc);
}
#endif // _DEBUG

/////////////////////////////////////////////////////////////////

// CP01Doc commands
// P01View.h : interface of the CP01View class
//
/////////////////////////////////////////////////////////////////

class CP01View : public CScrollView
{
protected: // create from serialization only
    CP01View();
    DECLARE_DYNCREATE(CP01View)

// Attributes
public:
    CP01Doc* GetDocument();
    int link[100][6];
    int PosXY[100][2];
    float weight[100][100];
    float d1, d2, R1, R2, R12, a, b, Ry, Ry1, x, x1, x2, y, y1, y2;
    long bin, dec, num, mod, e;
    int flag, PosX, PosY, i, j, count, node1, node2, scale, size, lo, count1,
    zoom, Px, Py;
    struct {
        char name[100];
        long int input;
        long int decimal;
        float dist;
    } animal[100];
};

```

// Operations

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public:

// Overrides
    // ClassWizard generated virtual function overrides
    //{AFX_VIRTUAL(CP01View)
    public:
        virtual void OnDraw(CDC* pDC); // overridden to draw this view
        virtual BOOL PreCreateWindow(CREATESTRUCT& cs);
    protected:
        virtual void OnInitialUpdate(); // called first time after construct
        virtual BOOL OnPreparePrinting(CPrintInfo* pInfo);
        virtual void OnBeginPrinting(CDC* pDC, CPrintInfo* pInfo);
        virtual void OnEndPrinting(CDC* pDC, CPrintInfo* pInfo);
    }AFX_VIRTUAL

// Implementation
public:
    virtual ~CP01View();
#ifdef _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
#endif

protected:

// Generated message map functions
protected:
    //{AFX_MSG(CP01View)
    afx_msg void OnLButtonDown(UINT nFlags, CPoint point);
    afx_msg void OnViewZoom();
    }AFX_MSG
    DECLARE_MESSAGE_MAP()

};

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#ifndef _DEBUG // debug version in P01View.cpp
inline CP01Doc* CP01View::GetDocument()
    { return (CP01Doc*)m_pDocument; }
#endif

/////////////////////////////////////////////////////////////////
// P01View.h : interface of the CP01View class
//
/////////////////////////////////////////////////////////////////

class CP01View : public CScrollView
{
protected: // create from serialization only
    CP01View();
    DECLARE_DYNCREATE(CP01View)

// Attributes
public:
    CP01Doc* GetDocument();
    int link[100][6];
    int PosXY[100][2];
    float weight[100][100];
    float d1, d2, R1, R2, R12, a, b, Ry, Ry1, x, x1, x2, y, y1, y2;
    long bin, dec, num, mod, e;
    int flag, PosX, PosY, i, j, count, node1, node2, scale, size, lo, count1,
    zoom, Px, Py;
    struct {
        char name[100];
        long int input;
        long int decimal;
        float dist;
    } animal[100];

// Operations
public:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// Overrides

    // ClassWizard generated virtual function overrides
    //{AFX_VIRTUAL(CP01View)

public:
    virtual void OnDraw(CDC* pDC); // overridden to draw this view
    virtual BOOL PreCreateWindow(CREATESTRUCT& cs);

protected:
    virtual void OnInitialUpdate(); // called first time after construct
    virtual BOOL OnPreparePrinting(CPrintInfo* pInfo);
    virtual void OnBeginPrinting(CDC* pDC, CPrintInfo* pInfo);
    virtual void OnEndPrinting(CDC* pDC, CPrintInfo* pInfo);
    //}AFX_VIRTUAL

// Implementation
public:
    virtual ~CP01View();
#ifdef _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
#endif

protected:

// Generated message map functions
protected:
    //{AFX_MSG(CP01View)

    afx_msg void OnLButtonDown(UINT nFlags, CPoint point);
    afx_msg void OnViewZoom();
    //}AFX_MSG

    DECLARE_MESSAGE_MAP()

};

```

```
#ifndef _DEBUG // debug version in P01View.cpp
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

inline CP01Doc* CP01View::GetDocument()
    { return (CP01Doc*)m_pDocument; }
#endif

/////////////////////////////////////////////////////////////////

// P01View.cpp : implementation of the CP01View class
//

#include "stdafx.h"
#include "P01.h"
#include "P01Dlg.h"
#include "FILEDlg.h"
#include "ZoomDlg.h"
#include "P01Doc.h"
#include "P01View.h"
#include "stdio.h"
#include "math.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

/////////////////////////////////////////////////////////////////

// CP01View

IMPLEMENT_DYNCREATE(CP01View, CScrollView)

BEGIN_MESSAGE_MAP(CP01View, CScrollView)
   //{{AFX_MSG_MAP(CP01View)
    ON_WM_LBUTTONDOWN()
    ON_COMMAND(ID_VIEW_ZOOM, OnViewZoom)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    //}}AFX_MSG_MAP
    // Standard printing commands
    ON_COMMAND(ID_FILE_PRINT, CScrollView::OnFilePrint)
    ON_COMMAND(ID_FILE_PRINT_DIRECT, CScrollView::OnFilePrint)
    ON_COMMAND(ID_FILE_PRINT_PREVIEW, CScrollView::OnFilePrintPreview)
END_MESSAGE_MAP()

```

```

////////////////////////////////////

```

```

// CP01View construction/destruction

```

```

CP01View::CP01View()

```

```

{
    // TODO: add construction code here

```

```

//    CZoomDlg zdlg; // Add

```

```

}

```

```

CP01View::~CP01View()

```

```

{

```

```

}

```

```

BOOL CP01View::PreCreateWindow(CREATESTRUCT& cs)

```

```

{

```

```

    // TODO: Modify the Window class or styles here by modifying

```

```

    // the CREATESTRUCT cs

```

```

    return CScrollView::PreCreateWindow(cs);

```

```

}

```

```

////////////////////////////////////

```

```

// CP01View drawing

```

```

void CP01View::OnDraw(CDC* pDC)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    CP01Doc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);

    // TODO: add draw code for native data here
    //pDC->TextOut(0,0,animal[0].name);

    RECT rect;
    pDC->SelectStockObject(BLACK_BRUSH);
    for (i=0;i<=10;i++) {
        PosX = PosXY[i][0];
        PosY = PosXY[i][1];
        rect.left = PosX - size;
        rect.top = PosY - size;
        rect.bottom = PosY + size;
        rect.right = PosX + size;
        pDC->Ellipse(&rect);
    }
    for (i=1;i<10;i++) {
        node1 = link[i][0];
        node2 = link[i][1];
        if (i==1) {
            x=PosXY[i-1][0];
            y=PosXY[i-1][1];
            pDC->MoveTo(x,y);

            x=PosXY[i][0];
            y=PosXY[i][1];
            pDC->LineTo(x,y);
        }
        else {
            x=PosXY[i][0];
            y=PosXY[i][1];
            pDC->MoveTo(x,y);
            x1=PosXY[node1][0];
            y1=PosXY[node1][1];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        pDC->LineTo(x1,y1);
        pDC->MoveTo(x,y);
        x1=PosXY[node2][0];
        y1=PosXY[node2][1];
        pDC->LineTo(x1,y1);
    }
}

void CP01View::OnInitialUpdate()
{
    CScrollView::OnInitialUpdate();
    CSize sizeTotal;
    // TODO: calculate the total size of this view
    sizeTotal.cx = sizeTotal.cy = 1000;
    SetScrollSizes(MM_TEXT, sizeTotal);
    //CFILEDlg Dlg;
    //Dlg.DoModal();
    zoom = 0;
    FILE *fp;
    fp=fopen("data.ner","r");
    // fp=fopen(Dlg.m_file,"r");
    i = 0;
    while(fscanf(fp,"%s%d%d%d%d%d%d",animal[i].name,&num,&mod,&dec,&Px,
    &Py,&node1,&node2)!=EOF) {
        animal[i].input = num;
        link[i][0]=mod;
        link[i][1]=dec;
        link[i][2]=Px;
        link[i][3]=Py;
        link[i][4]=node1;
        link[i][5]=node2;
        i++;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

count = i;
for (i= 0;i<count;i++) {
    num = animal[i].input;
    lo = 0;
    dec = 0;
    bin = 1;
    while (num != 0) {
        if (num >= 10) {
            mod = num%10;
            num = (num-mod)/10;
        }
        else {
            mod = num;
            num = 0;
        }
        dec = dec + (mod*bin);
        bin *= 2;
        if(mod==1)
            lo++;
    }
    R1 = lo;
    animal[i].decimal = dec;
    animal[i].dist = sqrt(R1);
}

for (i=0;i<count;i++) {
    d1 = animal[i].decimal;
    for (j=0;j<count;j++) {
        d2 = animal[j].decimal;
        if (d1>d2)
            dec=d1-d2;
        else
            dec=d2-d1;
        bin = 0;
        while (dec>0) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        mod = dec%2;
        dec = (dec-mod)/2;
        bin = bin + mod;
    }
    R1=bin;
    weight[i][j] = sqrt(R1);
}
}
for (i=0;i<100;i++) {
    PosXY[i][0] = 0;
    PosXY[i][1] = 0;
}
size=3;
scale=20;
PosX=300;
PosY=200;
for (i=0;i<=10;i++) {
    if(i==0) {
        PosXY[i][0] = PosX;
        PosXY[i][1] = PosY;
    }
    else {
        if (i==1) {
            d1 = weight[i][0];
            PosXY[i][0] = PosX + (d1 * scale);
            PosXY[i][1] = PosY;
        }
        else {
            node1 = link[i][0];
            node2 = link[i][1];
            flag = 0;
            for (zoom=0;zoom<=5;zoom++) {
                for (e=0;e<=5;e++) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (link[node1][zoom]==link[node2][e])
{
    lo = link[node1][zoom];
    i                                     f
    (((lo==0)&&(zoom==0))||(lo!=0)) {
        if (lo<i) {
            Px = PosXY[lo][0];
            Py = PosXY[lo][1];
            flag = 1;
            e=5;
            zoom=e;
        }
    }
}
/*
if ((link[node1][1]==link[node2][0])||(link[node1][1]==link[node2][1])) {
    lo = link[node1][1];
    if (lo!=i) {
        Px = PosXY[lo][0];
        Py = PosXY[lo][1];
        flag = 1;
    }
}
*/
x1 = PosXY[node1][0];
y1 = PosXY[node1][1];
x2 = PosXY[node2][0];
y2 = PosXY[node2][1];
R1 = weight[i][node1]*scale;
R2 = weight[i][node2]*scale;
R12 = weight[node1][node2];           // distant of 2
nodes.

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

d1 = (R1-(R2+R12))/2;
R1 = R1 - d1;
R2 = R2 + d1+5;
}
else {
    if (R2>(R1+R12)) {
        d2 = (R2-(R1+R12))/2;
        R1 = R1 + d2;
        R2 = R2 - d2;
    }
    else {
        if (R12 > (R1+R2)) {
            d1 = (R12 - (R1 + R2))/2;
            R1 = R1 + d1;
            R2 = R2 + d1;
        }
    }
    a = x1-R1;
    b = x1+R1;
    if (flag==1)
{
    if (((Px<x1)||((Px<x2))&&((Py>y1)||((Py>y2))))||(((Px>x1)||((Px<x2))&&((Py>y1)||((Py>y2)))) {
        for (x=a;x<=b;x+=0.1) {
            y=sqrt(R1*R1-(x-x1)*(x-x1))+y1;
            Ry1=y1*2-y;
            Ry=sqrt((x-x2)*(x-x2)+(Ry1-y2)*(Ry1-y2));
            if ((Ry>=R2-0.5)&&(Ry<=R2+0.5)) {
                PosXY[i][0]=x;
                PosXY[i][1]=Ry1;
                x=b+1;
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

{
    // default preparation
    return DoPreparePrinting(plnfo);
}

void CP01View::OnBeginPrinting(CDC* /*pDC*/, CPrintInfo* /*plnfo*/)
{
    // TODO: add extra initialization before printing
}

void CP01View::OnEndPrinting(CDC* /*pDC*/, CPrintInfo* /*plnfo*/)
{
    // TODO: add cleanup after printing
}

////////////////////////////////////
// CP01View diagnostics

#ifdef _DEBUG
void CP01View::AssertValid() const
{
    CScrollView::AssertValid();
}

void CP01View::Dump(CDumpContext& dc) const
{
    CScrollView::Dump(dc);
}

CP01Doc* CP01View::GetDocument() // non-debug version is inline
{
    ASSERT(m_pDocument->IsKindOf(RUNTIME_CLASS(CP01Doc)));
    return (CP01Doc*)m_pDocument;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#endif // _DEBUG

////////////////////////////////////

// CP01View message handlers

void CP01View::OnLButtonDown(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default

    CP01Dlg          dlg;
    i = point.x;
    j = point.y;
    for (lo=0;lo<count;lo++) {
        PosX= PosXY[lo][0];
        PosY= PosXY[lo][1];
        bin = link[lo][0];
        dec = link[lo][1];
        if (((i<=(PosX+size))&&(i>=(PosX-size)))&&((j<=(PosY+size))&&(j>=(PosY-size)))) {
            dlg.m_node = animal[lo].name;
            dlg.m_input = animal[lo].input;
            dlg.m_dist = animal[lo].dist;
            i=0;
            for(j=0;j<count;j++) {
                if((j==bin)||(j==dec)) {
                    i++;
                    switch(i) {
                        case 1:
                            dlg.m_node1 = animal[j].name;
                            break;
                        case 2:
                            dlg.m_node2 = animal[j].name;
                            break;
                        case 3:
                            dlg.m_node3 = animal[j].name;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปเผยแพร่ภายนอก
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

    }
}
}
}
lo = count;
dlg.DoModal();
}
}
CScrollView::OnLButtonDown(nFlags, point);
}

void CP01View::OnViewZoom()
{
    // TODO: Add your command handler code here
    CZoomDlg dlg;
    dlg.DoModal();
    x = PosXY[0][0];
    y = PosXY[0][1];
    lo = count;
    flag = dlg.m_normal;
    switch (flag) {
        case 0:
            if(size >= 5 ) {
                if (size == 5)
                    scale = 3;
                else
                    scale = 5;

                for (i=0;i<lo;i++) {
                    PosXY[i][0] = (PosXY[i][0]-x)/scale+x;
                    PosXY[i][1] = (PosXY[i][1]-y)/scale+y;
                }
                size = 3;
            }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void CP01View::OnViewZoom()
{
    // TODO: Add your command handler code here
    CZoomDlg      dlg;
    dlg.DoModal();
    x = PosXY[0][0];
    y = PosXY[0][1];
    lo = count;
    flag = dlg.m_normal;
    switch (flag) {
        case 0:
            if(size >= 5 ) {
                if (size == 5)
                    scale = 3;
                else
                    scale = 5;
                for (i=0;i<lo;i++) {
                    PosXY[i][0] = (PosXY[i][0]-x)/scale+x;
                    PosXY[i][1] = (PosXY[i][1]-y)/scale+y;
                }
                size = 3;
            }
            break;

        case 1:
            if (size == 3) {
                scale = 3;
                for (i=0;i<lo;i++) {
                    PosXY[i][0] = (PosXY[i][0]-x)*scale+x;
                    PosXY[i][1] = (PosXY[i][1]-y)*scale+y;
                }
                size = 5;
            }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else {
    if (size == 7) {
        scale = 5;
        for (i=0;i<lo;i++) {
            PosXY[i][0] = (PosXY[i][0]-x)/scale+x;
            PosXY[i][1] = (PosXY[i][1]-y)/scale+y;
        }
        scale = 3;
        for (i=0;i<lo;i++) {
            PosXY[i][0] = (PosXY[i][0]-x)*scale+x;
            PosXY[i][1] = (PosXY[i][1]-y)*scale+y;
        }
    }
    size = 5;
}
break;
case 2:
    if (size == 3) {
        scale = 5;
        for (i=0;i<lo;i++) {
            PosXY[i][0] = (PosXY[i][0]-x)*scale+x;
            PosXY[i][1] = (PosXY[i][1]-y)*scale+y;
        }
        size = 7;
    }
}
else {
    if (size == 5) {
        scale = 3;
        for (i=0;i<lo;i++) {
            PosXY[i][0] = (PosXY[i][0]-x)/scale+x;
            PosXY[i][1] = (PosXY[i][1]-y)/scale+y;
        }
        scale = 5;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        for (i=0;i<lo;i++) {
            PosXY[i][0] = (PosXY[i][0]-x)*scale+x;
            PosXY[i][1] = (PosXY[i][1]-y)*scale+y;
        }
    }
    size =7;
}
break;
}

```

```

Invalidate();
}
#include "stdio.h"
#include "string.h"
char    name[20];
int     n1,i,j,max,len;
char    animal[10][20];

void main()
{

    i = 0;
    j = i;
    FILE    *fp;
    fp = fopen("data.txt","r");
    while (fscanf(fp,"%s%d",name,&n1)!=EOF) {
        printf("ANIMAL : %s\t\t%d\n",name,n1);
        max=strlen(name);
        for(j=0;j<=max;j++)
            animal[i][j]=name[j];
        i++;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    max = i;
    for(i=0;i<max;i++) {
        for(j=0;j<20;j++)
            printf ("%c",animal[i][j]);
        //printf("\n");
    }
}

// ZoomDlg.h : header file
//

/////////////////////////////////////////////////////////////////
// CZoomDlg dialog

class CZoomDlg : public CDialog
{
// Construction
public:
    CZoomDlg(CWnd* pParent = NULL); // standard constructor

// Dialog Data
   //{{AFX_DATA(CZoomDlg)
    enum { IDD = IDD_DIALOG1 };
    int         m_normal;
    //}}AFX_DATA

// Overrides
    // ClassWizard generated virtual function overrides
    //{{AFX_VIRTUAL(CZoomDlg)
protected:
    virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV support
    //}}AFX_VIRTUAL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// Implementation
protected:

    // Generated message map functions
   //{{AFX_MSG(CZoomDlg)
    afx_msg void On100Radio();
    afx_msg void OnNormalRadio();
    afx_msg void On200Radio();
    virtual BOOL OnInitDialog();
   //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};
// stdafx.cpp : source file that includes just the standard includes
// P01.pch will be the pre-compiled header
// stdafx.obj will contain the pre-compiled type information

#include "stdafx.h"

// stdafx.cpp : source file that includes just the standard includes
// P01.pch will be the pre-compiled header
// stdafx.obj will contain the pre-compiled type information

#include "stdafx.h"

//{{NO_DEPENDENCIES}}
// Microsoft Developer Studio generated include file.
// Used by P01.rc
//
#define IDD_ABOUTBOX            100
#define IDR_MAINFRAME           128
#define IDR_P01TYPE             129
#define IDD_INFO_DIALOG        130
#define IDI_ICON1               132
#define IDI_FACE_ICON           133

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#define IDI_RED_ICON      134
#define IDD_FILE_DIALOG  135
#define IDB_PRO2_BITMAP  136
#define IDB_NAME_BITMAP  137
#define IDD_DIALOG1      139
#define IDB_ADVIS_BITMAP  140
#define IDB_TITLE_BITMAP 141
#define IDC_NODE_EDIT    1000
#define IDC_INPUT_STATIC 1001
#define IDC_LINK_STATIC  1002
#define IDC_NODE1_EDIT   1003
#define IDC_INPUT_EDIT   1004
#define IDC_NODE2_EDIT   1005
#define IDC_D_EDIT       1006
#define IDC_LINK1_EDIT   1007
#define IDC_LINK2_EDIT   1008
#define IDC_NODE3_EDIT   1009
#define IDC_NODE4_EDIT   1010
#define IDC_NODE5_EDIT   1011
#define IDC_NODE6_EDIT   1012
#define IDC_FILE_EDIT    1013
#define IDC_NORMAL_RADIO 1021
#define IDC_100_RADIO    1022
#define IDC_200_RADIO    1023
#define ID_VIEW_ZOOM     32771

```

```
// Next default values for new objects
```

```
//
```

```

#ifndef APSTUDIO_INVOKED
#ifndef APSTUDIO_READONLY_SYMBOLS
#define _APS_3D_CONTROLS      1
#define _APS_NEXT_RESOURCE_VALUE 142
#define _APS_NEXT_COMMAND_VALUE 32774
#define _APS_NEXT_CONTROL_VALUE 1026

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define _APS_NEXT_SYMED_VALUE    101
```

```
#endif
```

```
#endif
```

```
=====
```

```
====
```

```
MICROSOFT FOUNDATION CLASS LIBRARY : P01
```

```
=====
```

```
====
```

AppWizard has created this P01 application for you. This application not only demonstrates the basics of using the Microsoft Foundation classes but is also a starting point for writing your application.

This file contains a summary of what you will find in each of the files that make up your P01 application.

P01.h

This is the main header file for the application. It includes other project specific headers (including Resource.h) and declares the CP01App application class.

P01.cpp

This is the main application source file that contains the application class CP01App.

P01.rc

This is a listing of all of the Microsoft Windows resources that the program uses. It includes the icons, bitmaps, and cursors that are stored in the RES subdirectory. This file can be directly edited in Microsoft Developer Studio.

res\P01.ico

This is an icon file, which is used as the application's icon. This

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

icon is included by the main resource file P01.rc.

res\P01.rc2

This file contains resources that are not edited by Microsoft Developer Studio. You should place all resources not editable by the resource editor in this file.

P01.clw

This file contains information used by ClassWizard to edit existing classes or add new classes. ClassWizard also uses this file to store information needed to create and edit message maps and dialog data maps and to create prototype member functions.

////////////////////////////////////

For the main frame window:

MainFrm.h, MainFrm.cpp

These files contain the frame class CMainFrame, which is derived from CFrameWnd and controls all SDI frame features.

res\Toolbar.bmp

This bitmap file is used to create tiled images for the toolbar. The initial toolbar and status bar are constructed in the CMainFrame class. Edit this toolbar bitmap along with the array in MainFrm.cpp to add more toolbar buttons.

////////////////////////////////////

AppWizard creates one document type and one view:

P01Doc.h, P01Doc.cpp - the document

These files contain your CP01Doc class. Edit these files to

add your special document data and to implement file saving and loading

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(via CP01Doc::Serialize).

P01View.h, P01View.cpp - the view of the document

These files contain your CP01View class.

CP01View objects are used to view CP01Doc objects.

//

Other standard files:

StdAfx.h, StdAfx.cpp

These files are used to build a precompiled header (PCH) file named P01.pch and a precompiled types file named StdAfx.obj.

Resource.h

This is the standard header file, which defines new resource IDs.

Microsoft Developer Studio reads and updates this file.

//

Other notes:

AppWizard uses "TODO:" to indicate parts of the source code you should add to or customize.

If your application uses MFC in a shared DLL, and your application is in a language other than the operating system's current language, you will need to copy the corresponding localized resources MFC40XXX.DLL from the Microsoft Visual C++ CD-ROM onto the system or system32 directory, and rename it to be MFCLOC.DLL. ("XXX" stands for the language abbreviation. For example, MFC40DEU.DLL contains resources translated to German.) If you don't do this, some of the UI elements of your application will remain in the language of the operating system.

//

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

ก. เอกสารอ้างอิงที่เป็นวารสารภาษาอังกฤษ

Bernd Fritzsche " Growing Cell Structure - A Self-Organizing Network for Unsupervised and Supervised Learning " , Neural Network Vol. 7, No.9 pp. 1441-1460 ,1994

ข. เอกสารอ้างอิงที่เป็นวารสารภาษาไทย

1. ดร. ธิตติพงษ์ และ จุฬารัตน์ ตันประเสริฐ " โครงข่ายประสาทเทียม ตอนที่ 1 " , วารสารเนคเทค ,ปีที่ 1 ,ฉบับที่ 6 ,2538 , หน้า 19 - 23
2. ดร. ธิตติพงษ์ และ จุฬารัตน์ ตันประเสริฐ " โครงข่ายประสาทเทียม ตอนที่ 2 " , วารสารเนคเทค ,ปีที่ 1 ,ฉบับที่ 7 ,2538 , หน้า 19 - 23

ค. หนังสืออ้างอิงที่เป็นวารสารภาษาไทย และอังกฤษ

1. Dan W. Patterson " Artificial Neural Networks theory and applications " , Printice Hall
2. David J.Kruglinski " Inside Visual C++ " , Microsoft press ,896 page
3. Stephen T. Welstead " Neural Network and Fuzzy Logic Application in C / C ++ " , John wiley&Sons
4. S.Y.Kung " Digital Neural Networks " , PTR Printice Hall
5. ขจร โรจนเมรินทร์ " เทคนิคการเขียนโปรแกรม Turbo C ++ สำหรับวินโดวส์ " , ซีเอ็ดยูเคชั่น 336 หน้า
6. โชคชัย เตชพรรุ่ง " การเขียนโปรแกรม Visual C++ เวอร์ชัน 1.5 บนวินโดวส์ " , ซีเอ็ดยูเคชั่น ,896 หน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กิตติกรรมประกาศ

การจัดทำโครงการครั้งนี้ ต้องขอขอบพระคุณ อาจารย์ ดร. บุญธีร์ เครือตราฐ และอาจารย์ กฤตวัน เครือตราฐ ที่คอยให้คำแนะนำปรึกษาด้วยดีเสมอมา ขอขอบคุณสำนักหอสมุดกลาง ที่บริการได้ดีมาก ทั้งเจ้าหน้าที่ชั้น 1 และ 3 รวมทั้งห้องสมุดสมุดประจำคณะ วิศวกรรมศาสตร์ และพี่ ๆ น้อง ๆ ที่ศึกษาห้องเดียวกันที่มีข้อมูลเกี่ยวกับโครงข่ายประสาทเทียบแล้วให้มาเป็นข้อมูล และสุดท้ายต้องขอขอบพระคุณพ่อ แม่ และพี่ทางบ้านที่ให้กำลังใจ ทั้งเรื่องเรียน และเรื่องทำงาน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้