

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบตรวจสอบผู้เสียภาษีโดยอัตโนมัติ
Automatic Audit System



นายยุทธพงศ์ พงษ์สวัสดิ์ 40013262

นายวิชัย อร่ามแสงสุด 40013268

นายศรุต ชาประเสริฐ 40013271

ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2542

เลขที่.....
เลขทะเบียน..... 37071
วัน, เดือน, ปี 30 ส.ค. 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่อนุญาตให้นำออก หักมัดตราจอง หรือทำซ้ำ หากฝ่าฝืนโดยไม่ได้รับอนุญาตให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบตรวจสอบผู้เสียภาษีโดยอัตโนมัติ

Automatic Audit System

โดย

นายยุทธพงศ์ พงษ์สวัสดิ์ 40013262

นายวิชัย อร่ามแสงสุด 40013268

นายศรุต ชาประเสริฐ 40013271

อาจารย์ที่ปรึกษา

ผศ. บรรจง ปิยะธำรง

ปริญญาโทนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2542

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบตรวจสอบผู้เสียภาษีโดยอัตโนมัติ

Automatic Audit System

ผู้จัดทำ

นายยุทธพงศ์ พงษ์สวัสดิ์ 40013262

นายวิชัย อร่ามแสงสุต 40013268

นายศรุต ชาประเสริฐ 40013271



อาจารย์ที่ปรึกษา

(ผศ. บรรจง ปิยะธำรง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Automatic Audit System

Yutthapong Pongsawad 40013262

Wichai Aramsangsud 40013268

Sarut Chaprasert 40013271

Bunjong Piyathumrong Advisor

Abstract

At present, economic situation in Thailand is seem to be badly. The Government have a nessesity to get a loan from the forieng. Moreover, many tax payers evade their responsibilities. So that the Revenue Department have the duty to find out how to collect more tax and examine the tax payer to be the result of more total tax, that can repay to the foreign. That's why, the tax payer checking system (or programming)is offered to analyst each industry tax through person tax. This thesis is the tendency to check and find the tax payers who suspected to be the evaded. The Revenue Department will send the officers to inspect the matters of fact about them. By this system (or programming) VAT and CIT are suppose to be the consideration and any formulations are also the condition to solve this problem.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ด้วยศึหากไม่ได้รับความช่วยเหลือและร่วมมือจากหลาย ๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้วิทยานิพนธ์นี้เสร็จลงได้ก็คือ อาจารย์บรรจง ปิยะธำรง อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก

และต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจเอาใจใส่เสมอมา ในทุก ๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณมา ณ ที่นี้

นายยุทธพงษ์ พงษ์สวัสดิ์ 40013262

นายวิชัย อร่ามแสงสุค 40013268

นายศรุต ชาประเสริฐ 40013271

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูปภาพ	VI
สารบัญตาราง	VIII
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์ของ โครงการ	2
1.3 ขอบเขตของงานวิจัย	2
1.4 วิธีการดำเนินงาน	3
1.5 การใช้งาน	3
1.6 ประโยชน์ที่คาดว่าจะได้รับ	3
บทที่ 2 การเขียน โปรแกรมกับฐานข้อมูลด้วย ADO Data Control	4
2.1 ADO คืออะไร	4
2.2 ADO Data Control	4
2.3 Method ที่ใช้จัดการข้อมูลกับ ADO Data Control	7
2.4 DataGrid	7
บทที่ 3 คำสั่งที่ใช้ในการเขียนโปรแกรม SQL	10
3.1 คำสั่งที่ใช้ในการกระทำข้อมูลกับตาราง	10
3.2 คำสั่งที่ใช้ในการเรียกดูข้อมูล	10
3.3 คำสั่งที่ใช้ในการเรียงข้อมูลแบบมีเงื่อนไข	11
3.4 คำสั่งที่ใช้ในการ Join ข้อมูล	12
3.5 คำสั่งที่ใช้ในการเรียกข้อมูลโดยใช้โอเปอเรเตอร์ของ SET	12
บทที่ 4 Dynamic HTML	14
4.1 เหตุผลในการเลือกใช้ Dynamic HTML	14
4.2 ข้อดีของ DHTML	14
4.3 ข้อเสียของ DHTML	15
4.4 สร้าง DHTML เพื่อใช้งาน	15
4.5 โครงสร้างของ DHTML Application	18
4.6 Elementของ DHTML	19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7 การแทรก ActiveX Control เข้าไปใน DHTML	21
4.8 Event กับ HTML	23
4.9 การเก็บ Cookie	24
4.10 การอ่านค่า Cookie	24
บทที่ 5 โครงสร้างและส่วนต่าง ๆ ของโปรแกรมระบบตรวจสอบภาษี	25
5.1 Clue	25
5.2 Plan	26
5.3 Assign	26
5.4 Fiscal	27
5.5 Form VAT	29
5.6 Form CIT	30
บทที่ 6 ผลการทดสอบ	31
6.1 ส่วนของ Clue	31
6.2 ส่วนของ Plan	34
6.3 ส่วนของ Assign	36
6.4 ส่วนของ Fiscal	38
6.5 Form VAT	41
6.6 Form CIT	42
บทวิจารณ์ และ สรุป	43
ภาคผนวก ก ภาษี	
ภาคผนวก ข ภาษีเงินได้นิติบุคคล	
ภาคผนวก ค ภาษีมูลค่าเพิ่ม	
ภาคผนวก ง การเขียนโปรแกรม Visual Basic 6.0	

สารบัญรูปภาพ

	หน้าที่
รูปที่ 2.1 การเลือก Micro ADO Data Control 6.0 (OLEDB)	5
รูปที่ 2.2 Icon ที่ใช้แทน ADO Data Control	6
รูปที่ 2.3 Icon ที่ใช้ใน Form	6
รูปที่ 2.4 การเลือก Micro ADO DataGrid Control 6.0 (OLEDB)	8
รูปที่ 2.5 Icon ที่ใช้แทน ADO DataGrid Control	9
รูปที่ 4.1 เลือกสร้าง Application แบบ DHTML	16
รูปที่ 4.2 แสดง Element ต่าง ของ DHTML	16
รูปที่ 4.3 แสดง หน้าต่าง Designer	17
รูปที่ 4.4 หน้าต่าง Code Editor	18
รูปที่ 4.5 แสดง DHTML Document Object Model	18
รูปที่ 4.6 Add Component ActiveX เพิ่มเติม โดยการคลิกเมาส์ขวา	21
รูปที่ 4.7 ภาพ Select ActiveX สำหรับเพิ่มเติมใน DHTML	22
รูปที่ 4.8 ActiveX ถูกเพิ่มเข้ามายัง Toolbox	23
รูปที่ 5.1 แสดงหน้าต่างของ Clue	25
รูปที่ 5.2 แสดงหน้าต่างของ Plan	26
รูปที่ 5.3 แสดง หน้าต่างของ Assign	27
รูปที่ 5.4 แสดง โครงสร้างฐานข้อมูล	28
รูปที่ 5.5 แสดง form VAT	29
รูปที่ 5.6 แสดง Form CIT	30
รูปที่ 6.1 แสดงโปรแกรมในส่วนของ Clue	31
รูปที่ 6.2 แสดงรายละเอียดของรายการที่เลือก	32
รูปที่ 6.3 แสดงรายละเอียดที่เลือกไว้	33
รูปที่ 6.4 แสดงผลลัพธ์ของการตรวจสอบ	33
รูปที่ 6.5 แสดงหน้าต่างของ plan	34
รูปที่ 6.6 แสดงเริ่มแรกเมื่อเข้าโปรแกรม	35
รูปที่ 6.7 แสดงการเลือกแถวเดียว	35
รูปที่ 6.8 แสดงการเลือกหลายแถว	36
รูปที่ 6.9 แสดงการหน้าต่างของ Assign	36
รูปที่ 6.10 แสดงคำตอบที่ได้	37
รูปที่ 6.11 แสดงหน้าจอหลักของ ระบบ	38
รูปที่ 6.12 แสดง Fiscal	40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6.13 แสดง Form VAT

41

รูปที่ 6.14 แสดง Form CIT

42



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้าที่
ตารางที่ 4.1 แสดง Control ของ HTML	19
ตารางที่ 4.2 แสดง Element DHTML	20



บทที่ 1

บทนำ

1.1 ความสำคัญและที่มา

เนื่องจากขณะนี้สถานการณ์ทางด้านเศรษฐกิจของประเทศไทย อยู่ในสถานะที่ไม่สู้ดีเท่าไรนัก ซึ่งจากในช่วงที่ผ่านมาทุกคนคงรู้ว่ารัฐบาลไทยของเรานั้นต้องไปกู้เงินจากต่างชาติมาเป็นจำนวนมาก อีกทั้งยังมีผู้เสียหายที่ยังหลีกเลี่ยงการเสียหายอยู่อีกจำนวนมาก จึงทำให้ทางกรมสรรพากรคิดหาวิธีว่าจะทำอย่างไรดีจึงจะเก็บภาษีให้ได้มากขึ้นกว่าเดิมและจะมีวิธีการหาผู้ที่ยังหลีกเลี่ยงไม่เสียภาษีอย่างไร เพื่อที่จะได้ยอดของการเสียภาษีที่เพิ่มมากขึ้นกว่าเดิมจะได้นำเงินจากการเสียภาษีไปใช้หนี้สินที่กู้จากต่างชาติ จึงได้มีการเสนอให้มีระบบการตรวจสอบของผู้เสียหายขึ้นมา ซึ่งเป็นการวิเคราะห์ผู้เสียหายแต่ละประเภทอุตสาหกรรมจนลงไปถึงรายบุคคล เทคโนโลยีที่นำมาใช้ในการวิเคราะห์และตรวจสอบผู้เสียหายนั้น จะเป็นการเขียน โปรแกรมซึ่งจะอ้างอิงกับ ข้อมูล โดยที่เราจะมีการเก็บข้อมูลของผู้เสียหายทั้งหมด แล้วมาทำการคำนวณ หาว่ามีการหนีภาษีหรือไม่

จะเห็นได้ว่า การทำงานปัจจุบันของกรมสรรพากรนั้น ทำงานช้ามาก เพราะ ผู้เสียหายมีจำนวนมาก แต่จำนวนของ พนักงานตรวจสอบกลับมีน้อย ทำให้ไม่สามารถที่จะตรวจสอบได้หมดทุกคน

โปรแกรมนี้จะช่วยในการตรวจสอบให้สามารถที่จะไปทำการตรวจสอบผู้เสียหายที่น่าสงสัยเลข ซึ่ง ไม่ต้องมาทำการสุ่มเลือก จะผู้เสียหายเป็นจำนวนมาก ทำให้มี ประสิทธิภาพในการทำงานมากกว่า และ พนักงานตรวจสอบสามารถที่จะทำงาน ได้อย่างทั่วถึง

โดยการเก็บข้อมูล จะเก็บในรูปแบบของ SQL Server ซึ่งการที่เรา นำ SQL มาใช้นั้นเพราะว่า SQL สามารถทำงานกับ Microsoft ได้ดี ทั้งนี้ เนื่องจาก การเก็บข้อมูลของ สรรพากรนั้นจะเก็บเป็น Foxpro ดังนั้น การที่ ใช้ SQL นั้น จะทำให้สามารถ เก็บข้อมูลเข้าสู่ ฐานข้อมูล(Data base) ได้ง่าย

เนื่องจาก ปัจจุบันการเขียน โปรแกรม นิยมที่จะเขียน โปรแกรมขึ้น WEB เพราะว่ามีผู้ใช้สามารถเรียกใช้ได้ ทั่วๆที่ไม่จำกัดสถานที่ และเป็นการเป็น กว้างสำหรับ บุคคลทั่วๆ ไป การเขียน โปรแกรม นั้นจะเขียน เพื่อแสดงใน WEB เพื่อสามารถเรียกใช้กัน ได้หมด ภายในหน่วยงานของ กรม สรรพากร จึงจะมีส่วนที่เป็น ผู้ใช้ (Users) จะต้องใส่รหัส (password) เพื่อเข้าสู่ โปรแกรมการเขียนจะเข้าเป็นActive X Control ซึ่งทำเพื่อให้สามารถเรียกใช้ได้ง่าย และสามารถที่จะเชื่อมต่อกับฐานข้อมูล ได้ สะดวก และรวดเร็ว

เนื่องจากการเขียน โปรแกรมจำขึ้น WEB ดังนั้นในการเขียน โปรแกรมจึงต้องมี WEB Server ซึ่งเป็น ตัวที่ใช้ในการจัดการแสดงผลขึ้น WEB โดยจะใช้ ASP ในการติดต่อ และ แสดงรูปแบบของ โปรแกรม

1.2 วัตถุประสงค์ของโครงการ

1.2.1 ศึกษาการทำงานของ ระบบการตรวจสอบภาษี และสูตรในการวิเคราะห์ และ ตรวจสอบ ผู้เสียภาษี

1.2.2 ศึกษาการเขียน โปรแกรม VB และการเชื่อมต่อ VB กับ ฐานข้อมูล SQL ให้สามารถทำงานร่วมกันได้มีประสิทธิภาพ

1.2.3 ศึกษาการเขียน โปรแกรม โดย ศึกษาในส่วนของ Active X Control ใน VB เพื่อที่จะสามารถ นำ Active X Control ที่ได้มาแสดง ผ่าน WEB เพื่อ แสดง ข้อมูลขึ้นสู่หน้าจอ

1.2.4 ศึกษาการเขียน DHTML ได้เพื่อนำมาใช้ใน การดึง Active X Control ขึ้นมาใช้งาน ใน WEB และสามารถเชื่อมต่อ กับ WEB server ได้

1.2.5 เพื่อให้ได้ โปรแกรมที่สามารถทำงานได้จริง และ สามารถ นำไปใช้ในระบบ ตรวจสอบได้อย่างถูกต้อง และ แน่ใจได้ว่า โปรแกรมนี้ สามารถแสดง ใน WEB ได้อย่างถูกต้อง

1.2.6 โปรแกรมที่สร้างขึ้นมา จะต้อง สามารถตรวจสอบ โดย อ่านข้อมูลจาก ฐานข้อมูล ได้ทั้งหมด

1.3 ขอบเขตของงานวิจัย

งานวิจัยนี้ จะสร้าง ระบบที่ใช้ในการวิเคราะห์ และ ตรวจสอบ การเสียภาษี ขึ้น มา โดยในส่วนของ โปรแกรมนั้น จะทำการดึงข้อมูลจาก ฐานข้อมูลเข้ามาตรวจ และ ทำการคำนวณการเสียภาษีว่ามีกรหลีกเลี่ยงภาษีหรือไม่

โดยงานวิจัยนี้ จะมีส่วน ในการ ตรวจสอบอยู่ ทั้งหมด 6 ส่วน มีดังนี้

1.3.1 Fiscal

เป็นส่วนที่ใช้ในการเลือกกลุ่มที่จะตรวจสอบ โดย จะมีเปอร์เซ็นต์ บอกว่า ในส่วนนั้นๆ จำเป็นที่จะต้องไปตรวจสอบ เพราะค่าที่ได้ มาน้อยกว่า เปอร์เซ็นต์ที่ควรจะเป็น

1.3.2 Plan

เป็นส่วนที่ใช้เพื่อ กำหนด กลุ่มที่ จะเข้าไปตรวจสอบ โดยผู้ตรวจสอบจะเป็นผู้เลือก พื้นที่ที่จะเข้าไปตรวจ ว่าต้องการตรวจในพื้นที่ไหน รายได้เท่าไร ประเภทของสินค้า แล้วจะได้ผลลัพธ์เป็น จำนวนผู้เสียภาษีทั้งหมด ในพื้นที่ที่ต้องการ

1.3.3 Assign

เป็นการกำหนดรูปแบบการตรวจสอบ โดยจะสามารถเลือกได้ คล้ายๆ กับ Plan แต่สามารถที่จะเลือกการจัดลำดับการทำงานได้ โดย สามารถจัด จาก มากไปน้อย,น้อยไปมาก หรือ Random ข้อมูลก็ได้ โดยสามารถกำหนดจำนวนที่จะแสดง ได้อีกด้วย

1.3.4 Clue

เป็นส่วนที่ใช้ในการ ตรวจสอบเป็นรายบุคคล โดย จะสามารถที่จะเลือก ภาษีได้ และ จะมี สูตรและ Function ไว้ใช้ สำหรับตรวจสอบในแต่ละกรณี แล้วแต่ว่าผู้ใช้งานที่ต้องการที่จะตรวจสอบในลักษณะไหน มี Function และ สูตร ให้สามารถเลือกใช้ได้

1.3.5 แบบแสดงรายการภาษีมูลค่าเพิ่ม(FORM VAT)

เป็นส่วนที่ใช้ในการแสดงข้อมูลออกหน้าจอ โดยจะแสดงข้อมูลของผู้เสียภาษีมูลค่าเพิ่ม โดยจะทำการใส่ เลขผู้เสียภาษีอากร(TIN) แล้ว โปรแกรมจะไปดึงข้อมูลใน ฐานข้อมูลมาแสดงใน แบบแสดงรายการ โดยจะแสดงให้เห็นรายละเอียด ทั้งหมดของ ผู้เสียภาษีคนนั้น

1.3.6 แบบแสดงรายการภาษีเงินได้นิติบุคคล (FORM CIT)

เป็นส่วนที่ใช้ในการแสดงข้อมูลออกหน้าจอ เหมือนกับ แบบแสดงรายการภาษีมูลค่าเพิ่ม แต่จะแสดงข้อมูลของผู้เสียภาษีเงิน ได้นิติบุคคล โดยจะทำการใส่ เลขผู้เสียภาษีอากร(TIN) แล้ว โปรแกรมจะไปดึงข้อมูลใน ฐานข้อมูลมาแสดงใน แบบแสดงรายการ โดยจะแสดงให้เห็นรายละเอียด ทั้งหมดของ ผู้เสียภาษีคนนั้น

1.4 วิธีการดำเนินงาน

วิทยานิพนธ์ฉบับนี้เป็นแนวทางในการที่จะตรวจสอบและค้นหาผู้เสียภาษีที่อยู่ในขอบข่ายที่น่าสงสัยว่าเป็นผู้หลีกเลี่ยงการเสียภาษี แล้วทางกรมสรรพากรจะทำการจัดส่งเจ้าหน้าที่เข้าไปทำการตรวจสอบข้อเท็จจริงแก่บุคคลนั้น โดยระบบการตรวจสอบของผู้เสียภาษีนี้อาจจะทำการตรวจสอบ ภาษีมูลค่าเพิ่มและภาษีเงิน ได้นิติบุคคล มาเป็นตัวพิจารณาและใช้สูตรกฎเกณฑ์ต่าง ๆ มาเป็นเงื่อนไขในการตรวจสอบด้วย

1.5 การใช้งาน

การเข้าใช้งานนั้นจะเริ่ม โดย หน้าแรกจะเป็นการใส่ รหัส เข้า เร็งแล้วก็ เข้าในส่วนของ Menu จะมี Menu ให้เลือก 6 Menu ดังที่ได้กล่าวไปแล้ว

1.6 ประโยชน์ที่คาดว่าจะได้รับ

สิ่งที่ได้จาก โปรแกรมนี้คือ สามารถนำไปใช้งานจริงได้ และสามารถที่จะทำการตรวจสอบ ภาษีได้ในระดับหนึ่ง

โปรแกรมนี้จะช่วยในการตรวจสอบให้สามารถที่จะไปทำการตรวจสอบผู้เสียภาษีที่น่าสงสัยเลย ซึ่งไม่ต้องมาทำการสุ่มเลือก จะผู้เสียภาษีเป็นจำนวนมาก ทำให้มี ประสิทธิภาพในการทำงานมากกว่า และ พนักงานตรวจสอบ สามารถที่จะทำงาน ได้อย่างทั่วถึง

บทที่ 2

การเขียนโปรแกรมกับฐานข้อมูลด้วย ADO Data Control

Active Data Object (ADO) เป็นเทคโนโลยีใหม่ที่ถูกนำมาใช้ในการติดต่อกับฐานข้อมูล และกำลังได้รับความนิยมรวมทั้งมีแนวโน้มที่จะเป็นเทคโนโลยีที่ถูกนำมาใช้ในอนาคต เนื่องจาก สามารถติดต่อกับข้อมูลได้หลายรูปแบบ ซึ่ง Visual Basic 6.0 ได้นำเอาเทคโนโลยีนี้มาใช้เช่นเดียวกัน โดยผ่านทาง Control ที่มีชื่อว่า "ADO Data" โดยจะเห็นได้จาก Form ที่สร้างขึ้นจาก VB Data Form Wizard ของ Visual Basic 6.0 ส่วนใช้ Control นี้ในการติดต่อกับฐานข้อมูลแทน Data Control ที่ใช้ใน Visual Basic Version ก่อน ๆ ที่ผ่านมา ในส่วนนี้เราจะมาศึกษาถึงการเขียนโปรแกรมกับฐานข้อมูลด้วย ADO Data Control กัน

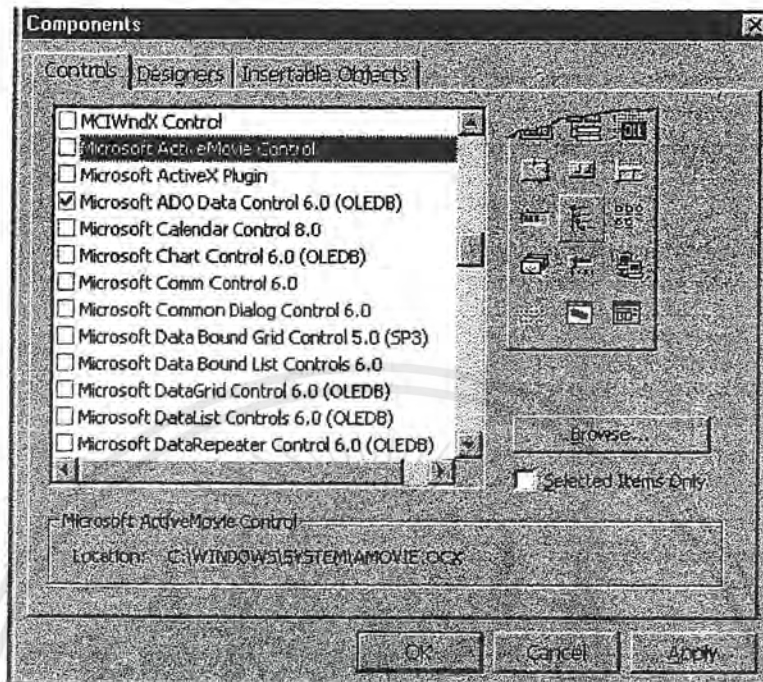
2.1 ADO คืออะไร

ย่อมาจากคำว่า Active Data Object โดยเป็นเทคโนโลยีใหม่ของบริษัท Microsoft ที่ถูกพัฒนาขึ้นเพื่อใช้ติดต่อกับฐานข้อมูลในรูปแบบต่าง ๆ เนื่องจาก ในปัจจุบัน คอมพิวเตอร์มีความสามารถในการใช้งานได้หลากหลาย ซึ่งส่งผลให้ข้อมูลต่าง ๆ ที่ใช้อยู่ในระบบงานคอมพิวเตอร์ มีรูปแบบที่หลากหลายมากขึ้นตามไปด้วย เช่น ข้อมูลที่อยู่ในรูปแบบของ Text ฐานข้อมูลที่มีโครงสร้างข้อมูลในแบบ Relational ข้อมูลที่อยู่ในรูปของ E-mail ข้อมูลที่เป็นรูปภาพ ข้อมูลที่อยู่ในรูปแบบของ Internet เป็นต้น ประกอบกับรูปแบบของข้อมูลที่หลากหลายนี้ ได้ก่อให้เกิดปัญหาในการใช้ข้อมูลร่วมกันระหว่าง Application ต่าง ๆ ที่มีการใช้ข้อมูลที่ต่างรูปแบบกัน ดังนั้นในปัจจุบัน จึงได้เกิดแนวความคิดที่มีชื่อว่า "Universal Data Access" ขึ้น แนวความคิดนี้มีจุดมุ่งหมายที่ต้องการให้เกิดศูนย์กลางของการติดต่อระหว่าง Application ที่มีการใช้ข้อมูลที่ต่างรูปแบบกัน เพื่อให้ Application ต่าง ๆ ที่ใช้ข้อมูลที่ต่างรูปแบบกันนั้น สามารถใช้ข้อมูลร่วมกันได้ โดยไม่ต้องเปลี่ยนแปลงรูปแบบของข้อมูลเดิมของแต่ละ Application แต่อย่างใด ซึ่งคล้ายกับการใช้ Open Database Connectivity (ODBC) แต่ต่างกันที่ ODBC ถูกสร้างขึ้นโดยบริษัทที่มีข้อตกลงในการใช้ข้อมูลร่วมกัน ดังนั้น ODBC จึงไม่สามารถรองรับข้อมูลได้ทุกรูปแบบ

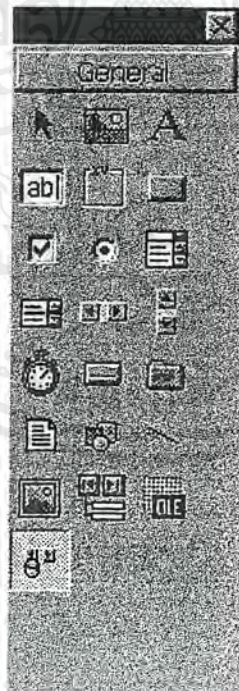
ในการติดต่อกับฐานข้อมูลของ ADO จะกระทำผ่าน OLE DB แทนการใช้ ODBC เนื่องจาก OLE DB เป็นเทคนิคใหม่ที่พัฒนาขึ้นเพื่อรองรับความคิดแบบ Universal Data Access

2.2 ADO Data Control

เนื่องจาก ADO Data Control ไม่ใช่ control มาตรฐานดังนั้นก่อนที่จะนำมาใช้งาน จึงต้องเพิ่ม Control นี้ไว้ใน Toolbox ก่อน โดย การเลือก " Micro ADO Data Control 6.0 (OLEDB)" ในจอภาพ Components ดังรูป



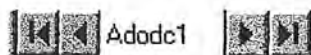
รูปที่ 2.1 การเลือก Micro ADO Data Control 6.0 (OLEDB)



รูปที่ 2.2 Icon ที่ใช้แทน ADO Data Control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับรูป Icon ที่ใช้แทน ADO Data Control ใน Toolbox ลูกศรชี้ จะมีรูปแบบดังรูป และเมื่อนำไปวางไว้บน Form จะเป็น Control ที่มีรูปร่างหน้าตาคล้ายกับ Data Control ดังรูป



รูปที่ 2.3 Icon ที่ใช้ใน Form

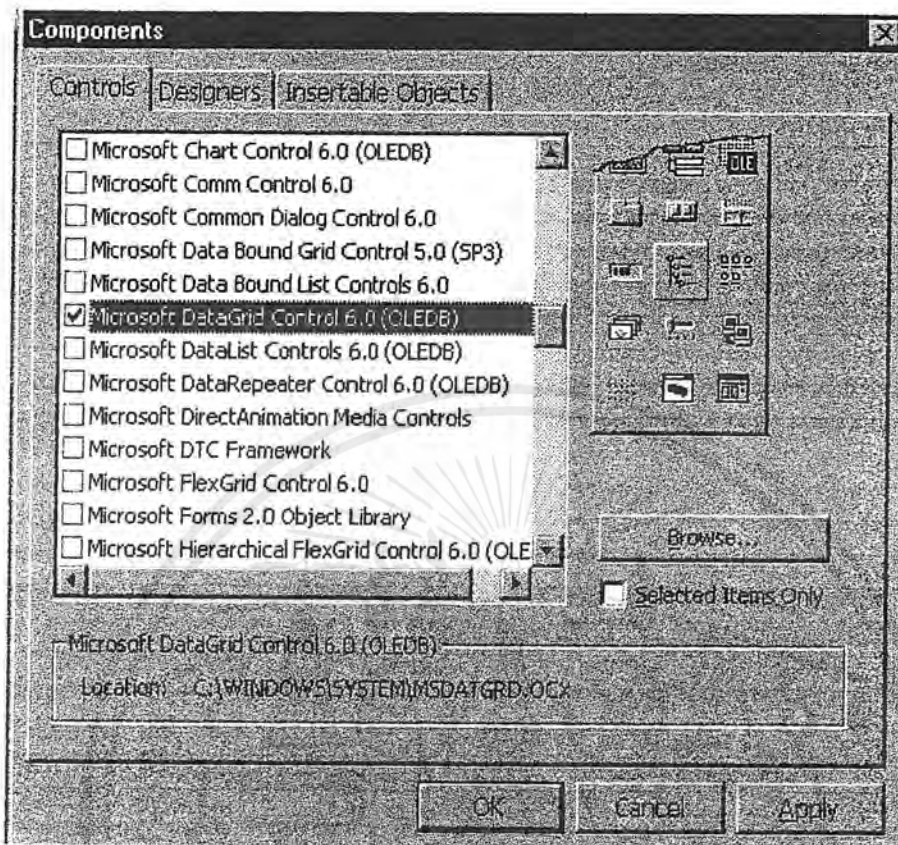
ในการใช้ ADO Data Control เพื่อติดต่อกับฐานข้อมูล จะแตกต่างจากการใช้ Data Control เนื่องจาก จะไม่มีการกำหนดชื่อของ ฐานข้อมูลที่ ADO นั้นอ้างถึงเช่นเดียวกับ Data Control แต่จะใช้การกำหนดค่าใน Property "Connection String " แทน Property " DatabaseName" ของ Data Control เพื่อติดต่อ ไปยัง Active X Data Source เพื่อสร้าง Object ของการติดต่อข้อมูลขึ้นแทน ส่วน Property "RecordSource" ของ ADO Data Control ยังคงใช้กำหนดตัวข้อมูลที่ต้องการติดต่ออยู่เช่นเดิม แต่จะสามารถเลือกรูปแบบได้หลายแบบมากขึ้น รวมทั้งกระทำผ่านจอภาพ Property Page แทน

2.3 Method ที่ใช้จัดการข้อมูลกับ ADO Data Control

Method ต่างๆที่ใช้ในการจัดการข้อมูลไม่ว่าจะเป็น การเพิ่ม แก้ไข และ ลบ ข้อมูลใน Table ของ ADO Data Control ยังคงเป็น Method "Add", "Update" และ "Delete" โดยจะอ้างกับ ADO Data Control แทน

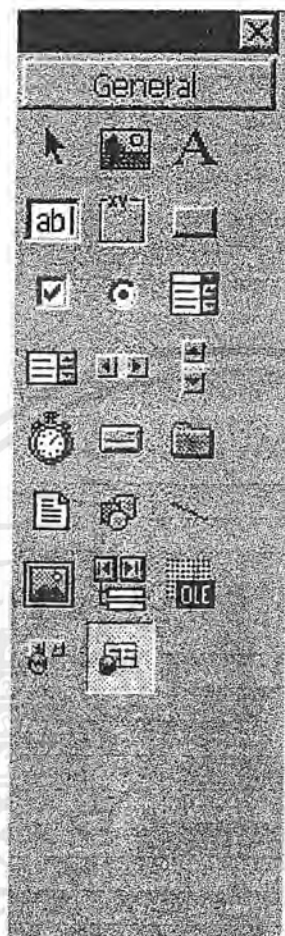
2.4 DataGrid

Datagrid จัดเป็น Bound Control หนึ่งที่เพิ่มเติมเข้ามาใน Visual Basic 6.0 โดยมีลักษณะเช่นเดียวกับ DBGrid แต่ต่างกันที่ DataGrid จะติดต่อ ผ่านทาง ADO Data Control แทน การนำ DataGrid มาใช้งานจะต้องเพิ่ม Control นี้ มาไว้ใน ToolBox ก่อน ด้วยการเลือก "Microsoft DataGrid Control 6.0 (OLEDB)" ในจอภาพ Components ดังรูป



รูปที่ 2.4 การเลือก Micro ADO DataGrid Control 6.0 (OLEDB)

สำหรับ Icon ที่ใช้แทน DataGrid ใน Toolbox (ลูกศรชี้) จะมีลักษณะดังรูป



รูปที่ 2.5 Icon ที่ใช้แทน ADO DataGrid Control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

คำสั่งที่ใช้ในการเขียนโปรแกรม SQL

3.1 คำสั่งที่ใช้ในการกระทำข้อมูลกับตาราง

3.1.1 การป้อนข้อมูลใหม่ จะใช้คำสั่ง

```
INSERT INTO < Table name > ,[( หลักที่1 ,หลักที่2 , ...)]  
VALUES (<value 1 , value 2 ,...>)
```

การป้อนข้อมูลเข้าด้วยคำสั่ง INSERT นี้ อาจไม่จำเป็นที่ต้องระบุชื่อ หลักก็ได้ แต่ค่า ของข้อมูล(value)จะต้องระบุ ค่าโดยเรียงลำดับการใส่ค่า

3.1.2 การคัดลอกข้อมูลจากตารางหนึ่ง ไปอีกตารางหนึ่ง จะใช้คำสั่ง

```
INSERT INTO < Table name > ,[( หลักที่1 ,หลักที่2 , ...)]  
SELECT Statement;
```

คำสั่ง Select จะเป็นตัวกำหนดตารางใหม่ ที่จะนำข้อมูลไปได้

3.1.3 การปรับปรุงข้อมูล จะใช้คำสั่ง

```
UPDATE < Table name >  
SET <หลักที่ 1> [, หลักที่2 ,...] = <expression Lsubquery>  
[WHERE <condition>];
```

3.1.4 การลบข้อมูล ใช้ลบข้อมูลในแถวที่ต้องการ จะใช้คำสั่ง

```
DELETE FROM < Table name >  
[ Where <condition>];
```

3.2 คำสั่งที่ใช้ในการเรียกดูข้อมูล

การเรียกดูข้อมูลนั้น จะใช้คำสั่ง SELECT ดังนี้

3.2.1 เป็นการเรียกดูข้อมูลทั้งหมด

```
SELECT *
```

```
FROM < Table name >;
```

3.2.2 เป็นการเรียกดูข้อมูลบางส่วน

```
SELECT <column1, column2,...>
FROM < Table name >;
```

3.2.3 เป็นการเรียกดูข้อมูลบางส่วนแบบข้อมูลไม่ซ้ำ

```
SELECT [*] DISTINCT <column1, column2,...>
FROM < Table name >;
```

3.2.4 เป็นการเรียกดูข้อมูล โดยการจัดเรียงข้อมูล

```
SELECT [*] DISTINCT <column1, column2,...>
FROM < Table name >
[ order by < column1,... > [DESC]];
```

3.3 คำสั่งที่ใช้ในการเรียงข้อมูลแบบมีเงื่อนไข

3.3.1 การเรียงข้อมูลแบบมีเงื่อนไขทั่วไป

```
SELECT [*]DISTINCT <column1, column2,...>
FROM < Table name >
[ WHERE <CONDITION>;
```

โดยที่ condition จะเป็นเงื่อนไขในการเข้าสู่ข้อมูล

3.3.2 การเรียงข้อมูลแบบมีเงื่อนไขโดยใช้ โอเปอเรเตอร์ทางตรรกะ

การใช้ โอเปอเรเตอร์ทางตรรกะ จะเป็นการใช้ =, <, >, <=, >= มากเป็นตัวเปรียบเทียบ

3.3.3 การเรียงข้อมูลแบบมีเงื่อนไขโดยใช้ โอเปอเรเตอร์ ของ SQL

โอเปอเรเตอร์ ของ SQL ที่ใช้ในการกำหนดเงื่อนไขมีดังนี้

- 1 BETWEEN.... AND เป็น โอเปอเรเตอร์ ที่ กำหนดเป็น ช่วงๆ
- 2 IN เป็น โอเปอเรเตอร์ ที่ใช้ในการจัดกลุ่ม
- 3 LIKE เป็น โอเปอเรเตอร์ ที่ใช้ในการหาข้อมูล โดยเก็บเป็น ตัวอักษรเท่านั้น

4 IS NULL เป็น โอเปอเรเตอร์ ที่บอกว่ามีค่าว่าง หรือ ไม่มีค่า

3.3.4 การเรียงข้อมูลแบบมีเงื่อนไขโดยใช้ โอเปอเรเตอร์บูลีน

โอเปอเรเตอร์ทางบูลีน ได้แก่ AND , OR , NOT มากเป็นตัวเปรียบเทียบ

3.4 คำสั่งที่ใช้ในการเรียงข้อมูลด้วย Function ที่เกี่ยวกับการรวม

3.4.1 Function ที่เกี่ยวกับการรวมมีดังนี้

3.4.1.1 AVG [*]DISTINCT <column name >

3.4.1.2 COUNT [*]DISTINCT <column name >

3.4.1.3 MIN [*]DISTINCT <column name >

3.4.1.4 MAX[*]DISTINCT <column name >

3.4.1.5 SUM [*]DISTINCT <column name >

3.4.2 การใช้ GROUP BY

เป็นการจัดกลุ่มเป็นกลุ่มย่อยๆ หลายๆ กลุ่ม ทำให้สามารถที่จะเข้าถึงข้อมูล ได้ง่ายและ รวดเร็ว ซึ่งการใช้ GROUP BY นั้น ทำเพื่อให้ได้ข้อมูลที่จัดเรียกกันเป็นกลุ่มเป็นก้อน

```
SELECT [*] DISTINCT <column1, column2,...>
```

```
FROM < Table name >
```

```
GROUP BY < column1,... >;
```

3.4.3 การใช้ HAVING

เป็นการจัดกลุ่มเป็นกลุ่ม ซึ่งจะเป็นเงื่อนไขใน GROUP BY อีกที่หนึ่ง HAVING จะต้องใช้คู่กับ GROUP BY เสมอ

```
SELECT [*] DISTINCT <column1, column2,...>
```

```
FROM < Table name >
```

```
GROUP BY < column1,... >
```

```
HAVING <CONDITION>;
```

3.5 คำสั่งที่ใช้ในการ Join ข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นการนำตารางหลายตารางมารวมกัน(JOIN) ทำโดย

```
SELECT [*] DISTINCT <column1, column2,...>
FROM < Table name 1>, < Table name 2>[,< Table name 3>,...];
```

3.6 คำสั่งที่ใช้ในการเรียกข้อมูลโดยใช้โอเปอเรเตอร์ของ SET

3.6.1 UNION โอเปอเรเตอร์ที่ใช้ในการ รวมข้อมูล

```
SELECT [*] DISTINCT <column1, column2,...>
FROM < Table name >;
UNION
SELECT [*] DISTINCT <column1, column2,...>
FROM < Table name >;
```

3.6.2 INTERSECT โอเปอเรเตอร์ที่ใช้ในการดูข้อมูลที่เหมือนกัน

```
SELECT [*] DISTINCT <column1, column2,...>
FROM < Table name >;
INTERSECT
SELECT [*] DISTINCT <column1, column2,...>
FROM < Table name >;
```

3.6.3 MINUS โอเปอเรเตอร์ที่ใช้ในการดูข้อมูลที่ต่างกัน

```
SELECT [*] DISTINCT <column1, column2,...>
FROM < Table name >;
MINUS
SELECT [*] DISTINCT <column1, column2,...>
FROM < Table name >;
```

บทที่ 4

Dynamic HTML

แม้ว่า HTML จะเป็นภาษาที่เราใช้ในการสร้างเว็บเพจเป็นส่วนใหญ่ แต่ตัว HTML เองก็ได้กลายเป็นอุปสรรคสำคัญเมื่อเราต้องการให้เว็บเพจที่สร้างขึ้นมีความสามารถ และฉลาดขึ้น ทำให้ต้องคิดวิธีการใหม่ๆ ไม่ว่าจะเป็นการเติมความสามารถด้านโปรแกรมด้วยภาษาสคริปต์ต่างๆ เช่น VBSCRIPT , JAVA SCRIPT แต่นั่นก็เป็นส่วนหนึ่งสำหรับคนที่คุ้นเคยกับ Visual Basic มาก่อน ทาง Microsoft ก็ได้เตรียมวิธีการสร้างเว็บเพจในแบบที่เรียนรู้ได้ไม่ยาก เพราะเพียงอาศัยความรู้เกี่ยวกับ Visual Basic ไปดัดแปลงเพิ่มขึ้นบางอย่าง

DHTML หรือ Dynamic HTML ก็เป็นส่วนหนึ่งในวิธีการดังกล่าว ที่ Visual Basic เสนอให้คนที่คุ้นเคย Visual Basic สามารถสร้างเว็บเพจแบบไดนามิก ได้โดยอาศัยความรู้เกี่ยวกับ Visual Basic ได้อย่างไม่ยากเย็น

4.1 เหตุผลในการเลือกใช้ Dynamic HTML

4.1.1 HTML ไม่มีความสามารถเพียงพอ หรือไม่สะดวกพอที่เราจะได้ตอบกับผู้ใช้งานได้อย่างมีประสิทธิภาพ

4.1.2 แม้จะมี ภาษา Script ให้เลือกใช้ แต่ก็ยังมีความเกี่ยวข้องกับ HTML ค่อนข้างมาก เราอาจจะไม่มีเวลามากพอที่จะศึกษา HTML แต่ถ้าใช้ความรู้เดิมๆ เกี่ยวกับ Visual Basic มาใช้อาจจะช่วยให้เกิดประโยชน์ได้มากกว่า

4.1.3 เราสามารถใช้ Visual Basic และเครื่องมือเสริมการทำงานต่างๆ ของ Visual Basic มาสร้าง Application Internet

4.1.4 เราอาจต้องการสร้าง Application Intranet ที่มีความน่าเชื่อถือ หรือมีความซับซ้อนมากกว่า ความสวยงามของรูปภาพ หรือลูกเล่นต่างๆ

สำหรับตัวอย่างการนำไปใช้งานนั้นคงจะเน้นไปที่อินเทอร์เน็ตมากกว่าที่จะนำไปใช้ใน อินเทอร์เน็ต เช่น บริษัทอนุญาตให้พนักงานเข้ามาดูรายงานผลการดำเนินงาน นโยบายของบริษัทล่าสุด เป็นต้น

4.2 ข้อดีของ DHTML

4.2.1 การใช้พลังการประมวลผลของไคลเอ็นท์อย่างเต็มที่ เพราะ DHTML จะรันบน บราวเซอร์ของผู้ใช้งาน ทำให้การตอบสนองการทำงานดีกว่า

4.2.2 มีความปลอดภัยสูงกว่า HTML ธรรมดา เพราะเราไม่สามารถแก้ไขโค้ดของ DHTML ได้ เหมือนกับที่เราแก้ไขเว็บเพจที่สร้างจาก HTML ปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 สนับสนุนการจัดการสถานะของผู้ใช้งาน เว็บเพจที่สร้างจาก HTML นั้นเราไม่มีทางรู้ว่าผู้ใช้เคยเข้ามาใช้งานกี่ครั้ง , มีการร้องขอข้อมูลอะไรจาก Server บ้าง ซึ่งเราจะใช้ Cookie มาช่วยในการทำงาน

4.3 ข้อเสียของ DHTML

4.3.1 การดาวน์โหลด DHTML : มาใช้ในครั้งแรก จะได้ไฟล์ที่มีขนาดค่อนข้างใหญ่ (ในหลักร้อย กิโลไบต์) เว้นแต่ว่าเครื่องคอมพิวเตอร์ นั้นติดตั้ง Visual Basic 6.0 หรือมี Application ที่สร้างจาก Visual Basic 6.0 ใช้งานอยู่ก็อาจไม่ต้องโหลดทั้งหมด แต่การ โหลดครั้งต่อๆ มา จะน้อยกว่าเดิมมาก

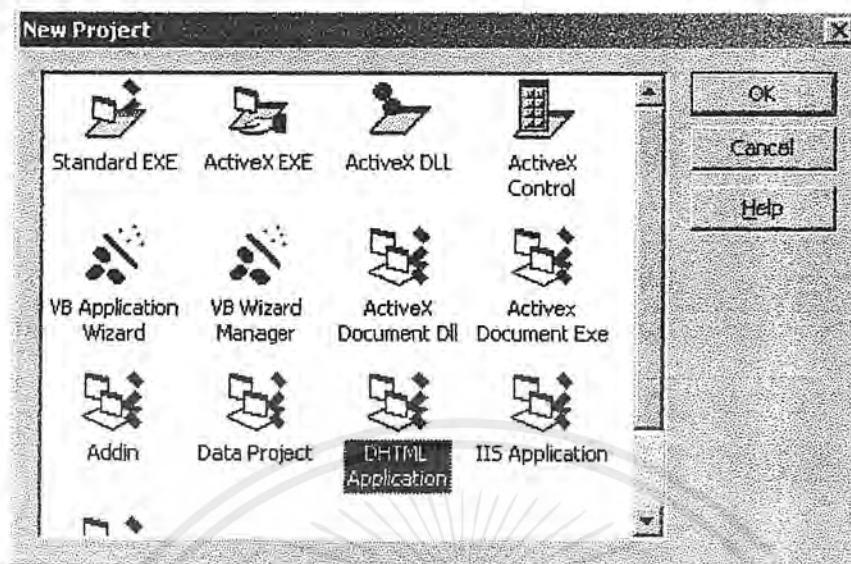
4.3.2 ยังต้องพึ่ง Internet Explorer : การใช้ DHTML ที่สร้างจาก Visual Basic นั้น ณ วันนี้ยังใช้ได้กับ internet Explorer เวอร์ชัน 4.01 ขึ้นไปเท่านั้น และนี่เองเป็นคำตอบที่ว่าทำไม DHTML จึงเหมาะกับ อินเทอร์เน็ตมากกว่า เพราะว่าเราสามารถแนะนำหรือกำหนดให้คนในองค์กรใช้เบราว์เซอร์ที่เราเลือกง่ายกว่าไปบอกคนในอินเทอร์เน็ต

4.3.3 DHTML ยังไม่สนับสนุน CGI ซึ่งเป็นเทคโนโลยีในฝั่งเซิร์ฟเวอร์ที่ใช้กันมานานในโลกของยูนิกซ์ สำหรับข้อมูลเกี่ยวกับ DHTML โดยละเอียดเราสามารถดาวน์โหลด หรือ อัดเททข้อมูลได้จาก เว็บไซต์ของไมโครซอฟต์

4.4 สร้าง DHTML เพื่อใช้งาน

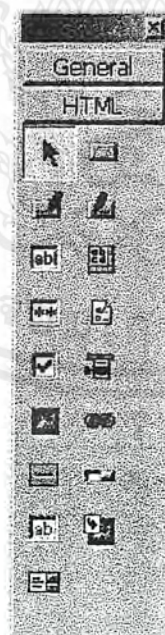
การสร้าง DHTML อย่างง่าย เพื่อเรียนรู้ และเข้าใจขั้นตอนการสร้าง DHTML ซึ่งมีขั้นตอนดังนี้

4.4.1 เลือกสร้าง Application แบบ DHTML โดยเลือกเมนู File > New Project ให้เลือกไอคอน DHTML Application



รูปที่ 4.1 เลือกสร้าง Application แบบ DHTML

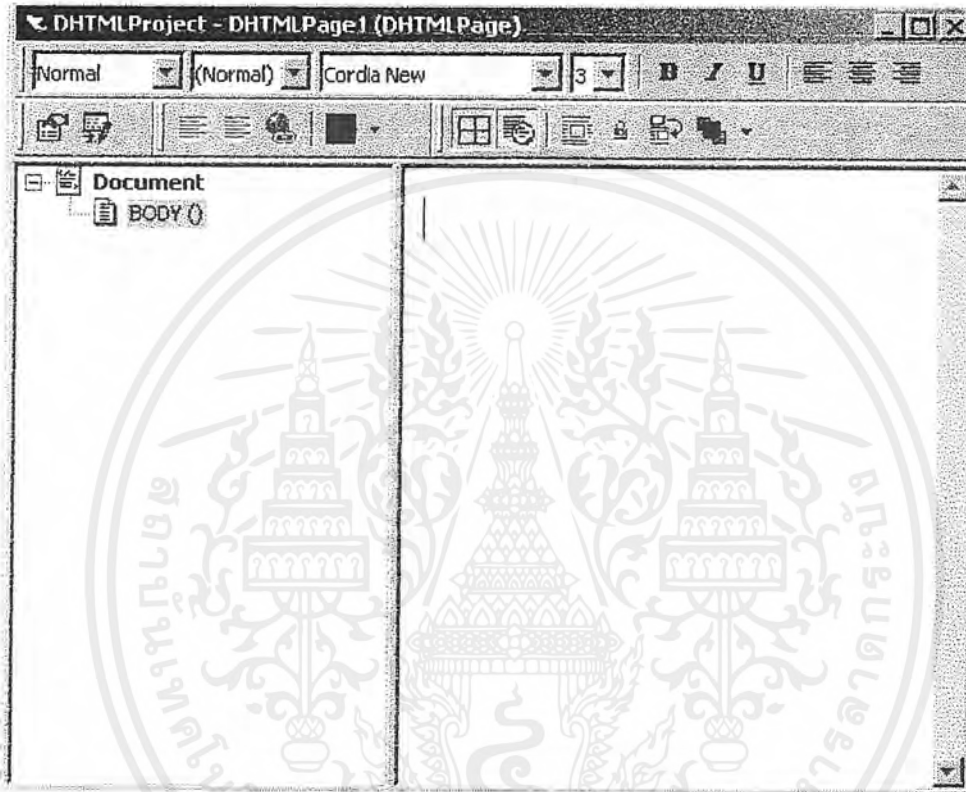
4.4.2 Visual Basic จะเพิ่ม Toolbox HTML ให้เรานอกจาก Toolbox ที่เก็บ ActiveX Control ปกติ และที่ Project Explorer ก็จะมีการเพิ่ม DHTML Designer ให้เรานอกจากโค้ดโมดูล ใน Toolbox HTML นั้นจะมีคอนโทรลต่าง ๆ คล้ายกับ ActiveX Control ซึ่งต่อไปนี้จะเรียกว่า Element ของ DHTML



รูปที่ 4.2 แสดง Element ต่าง ของ DHTML

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

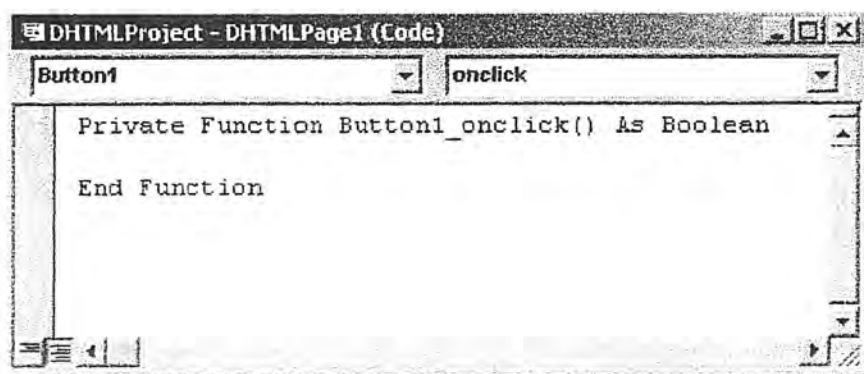
4.4.3 ใน DHTML Designer นั้นจะแบ่งออกเป็น 2 พาเนล พาเนลด้านซ้ายเรียกว่าพาเนล TreeView ซึ่งจะแสดงรายการของ Element ต่างๆ ในลักษณะของ TreeView ส่วนพาเนลด้านขวาจะเรียกว่าพาเนล Detail ซึ่งจะแสดงผลลัพธ์ หรือสิ่งที่เราจะเห็นเมื่อมีการแสดงผลบนบราวเซอร์



รูปที่ 4.3 แสดง หน้าต่าง Designer

4.4.4 เราสามารถใช้เมาส์คลิก Element ที่ต้องการจาก Toolbox ซึ่งมี Element ที่ถูกเลือกจะไปปรากฏในพาเนล Detail พร้อมกับถูกเพิ่มเข้าไปในพาเนล TreeView และเราสามารถปรับขนาดของ Element ต่างๆ ได้จากรูปสี่เหลี่ยมที่อยู่ขอบ และมุมของ Element (เรียกว่า Sizing Handle) ได้ตามต้องการ นอกจากนี้ยังเปลี่ยนตำแหน่งการวางของ Element ได้เหมือนกับที่เราทำงานกับ Visual Basic

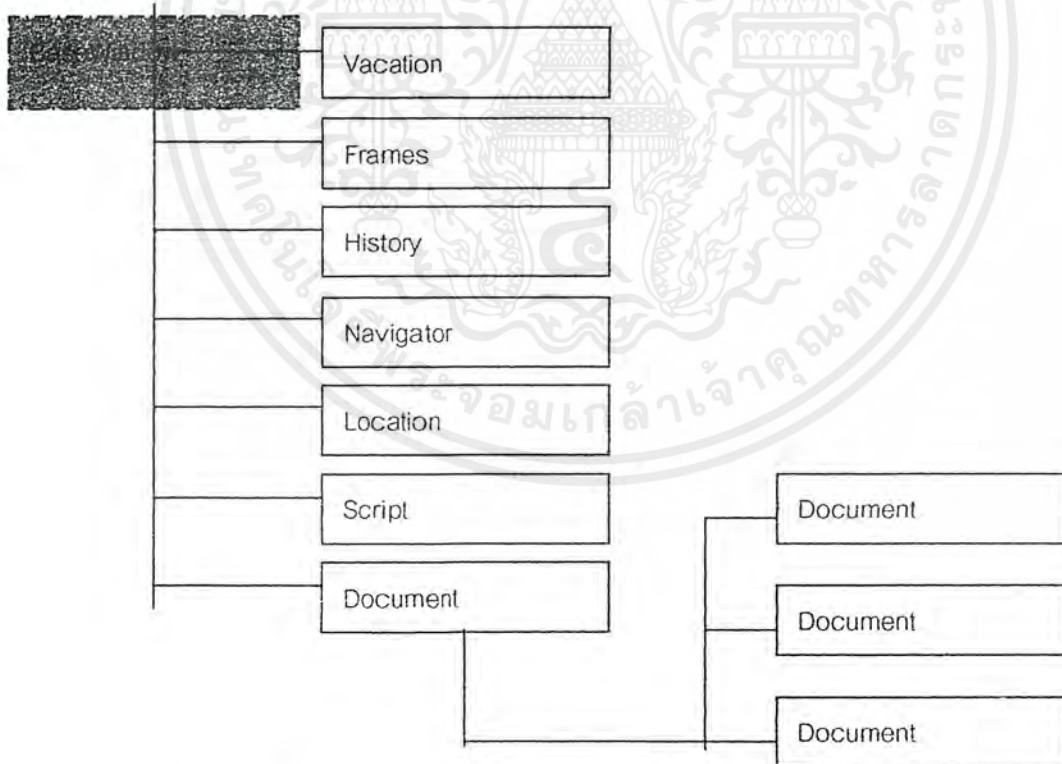
4.4.5 ดับเบิลคลิกที่ Element ที่ต้องการ ก็จะปรากฏได้อิดิเตอร์ เหมือนกับ Visual Basic ทดลองเขียน ได้คเพิ่มเติมเพื่อควบคุมการทำงาน



รูปที่ 4.4 หน้าต่าง Code Editor

4.5 โครงสร้างของ DHTML Application

ในการสร้าง Application ด้วย HTML ธรรมดา เราจะต้องเรียนรู้เกี่ยวกับโครงสร้างการทำงานของเบราว์เซอร์ที่เรียกว่า Document Object Model และสำหรับ DHTML ก็มีโครงสร้างดังกล่าวเช่นเดียวกัน แต่ก็มีข้อแตกต่างกันบ้างเล็กน้อย



รูปที่ 4.5 แสดง DHTML Document Object Model

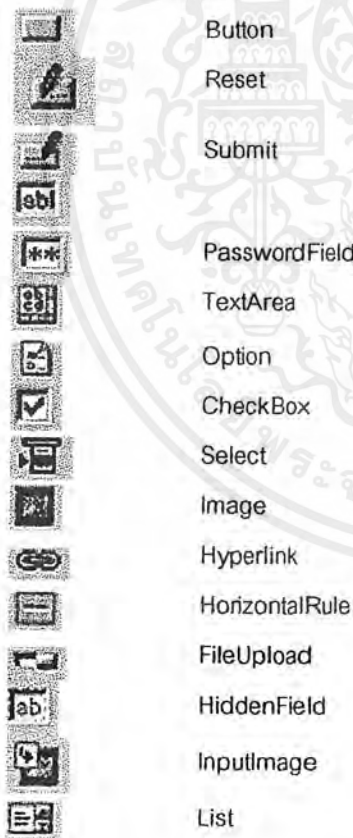
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออบเจกต์ Base Window ยังถือเป็นออบเจกต์ที่เป็นตัวแทนจัดการบราวเซอร์ ออบเจกต์ document ก็เป็นตัวแทนของเว็บเพจ HTML ภายในออบเจกต์ก็จะมีออบเจกต์ DHTML Page และ ภายในออบเจกต์ DHTMLPage จะประกอบด้วย Element ต่างๆ

เราสามารถเปรียบเทียบออบเจกต์ BaseWindow ได้กับแอปพลิเคชันของ Visual Basic ส่วน ออบเจกต์ Document หรือ DHTMLPage ก็เปรียบได้กับฟอร์มของ Visual Basic ส่วน Element ต่าง ก็เป็น เหมือนกับ ActiveX Control ใน Visual Basic แต่ว่าเราก็สามารถแทรก ActiveX Control เข้าไปใน Application DHTML ได้เช่นเดียวกัน

4.6 Element ของ DHTML

สำหรับ Element ของ DHTML ก็จะเหมือนกับ Control ของ HTML ซึ่งแต่ละ Element จะมี Property เหมือนกับ ActiveX Control



ตารางที่ 4.1 แสดง Control ของ HTML

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทุก Element จะมี Property เฉพาะตัวที่ไม่ซ้ำกันเลยคือ ID และจะมี Property Name ซึ่งเป็นชื่อของ Element ซึ่งซ้ำกันได้ (จุดนี้จะต่างจาก Visual Basic) และเราสามารถเปรียบเทียบ Element ได้กับ ActiveX Control ดังนี้

Element	Description
Button	เหมือนกับ Command Button ใน Visual Basic ทำหน้าที่รับคำสั่งจากการกด / คลิก โดยเราแสดงข้อความบนปุ่มโดยกำหนดพรอพเพอร์ตี้ Value
Submit	เป็น Button พิเศษที่ส่งให้ส่งข้อมูลที่ผู้ใช้ป้อนในบราวเซอร์ ส่งไปยังเซิร์ฟเวอร์ โดยเราแสดงข้อความบนปุ่มโดยกำหนดพรอพเพอร์ตี้ Value
Reset	เป็น Button พิเศษที่เคลียร์ค่าต่าง ๆ ที่ผู้ใช้ได้กรอกเข้าไปทั้งหมด โดยเราแสดงข้อความบนปุ่มโดยกำหนดพรอพเพอร์ตี้ Value
TextField	เหมือนกับ TextBox ใน Visual Basic
TextArea	เหมือนกับ TextField แต่ว่าสามารถแสดงผลข้อมูลได้ที่หลายๆ
PasswordField	เป็น TextField พิเศษที่รับรหัสผ่านจากผู้ใช้งาน แสดงเครื่องหมาย ดอกจัน (*)
Option	เหมือนกับ Option ใน Visual Basic การเลือก/ไม่เลือกจะตรวจสอบจากพรอพเพอร์ตี้ checked สำหรับการสร้างกลุ่มของตัวเลือกโดยกำหนดให้พรอพเพอร์ตี้ Name เหมือนกัน (แต่พรอพเพอร์ตี้ ID จะไม่เหมือนกัน)
CheckBox	เหมือนกับ CheckBox ใน Visual Basic เราสามารถกำหนดจำนวนของรายการตัวเลือกจากพรอพเพอร์ตี้ Size อีกทั้งมี Property Page ให้เราใส่รายการตัวเลือกได้
Select	เหมือนกับ ComboBox ใน Visual Basic เราสามารถกำหนดจำนวนของรายการตัวเลือกจาก พรอพเพอร์ตี้ Size อีกทั้งมี Property Page ให้เราใส่รายการตัวเลือกได้
Image	คล้ายกับ Image ใน VB เรากำหนดภาพจากพรอพเพอร์ตี้ SRC นอกจากนี้ยังกำหนดข้อความเหมือน Tooltip Text เมื่อคลิกรูปภาพ ได้จากพรอพเพอร์ตี้ Title
Hyperlink	เป็น element ที่ใช้สร้าง Link เชื่อมไปยังเว็บเพจ หรือเอกสารใดๆ ตามที่เราระบุไว้ในพรอพเพอร์ตี้ HREF
HorizontalRule	เป็นเส้นที่ขีดคั่นในแนวนอน สามารถกำหนดสี และขนาดของเส้นได้
FileUpload	เป็น element ที่ประกอบด้วย textbox ติดกับปุ่ม Browse ใช้เลือกไฟล์ Upload
HiddenField	เป็น element ที่เปรียบได้กับ TextField ที่มองไม่เห็นขณะรัน สามารถจะเข้าถึงได้โดยการเขียนโค้ด มักใช้เก็บข้อมูลคงที่ ที่ไม่ต้องการให้เห็น หรือเปลี่ยนแปลง
List	เหมือนกับ ListBox ใน VB โดยมี Property Page ให้เราใส่ตัวเลือกได้
Input Image	เป็น element ที่แสดงรายการของรูปภาพ

ตารางที่ 4.2 แสดง Element DHTML

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราสามารถกำหนด Property ให้กับ element ต่างๆ ผ่าน ทาง Property Window เหมือนกับ ActiveX Control ทั่วไป รวมทั้งยังกำหนดผ่าน Property Page ได้ด้วย

4.7 การแทรก ActiveX Control เข้าไปใน DHTML

ActiveX Control มาตรฐานส่วนใหญ่ จะมีอีลิเมนต์ ของ DHTML แทนที่แล้ว แต่ถ้าเป็น ActiveX Control อื่นๆ เราสามารถเพิ่มให้กับ DHTML ได้โดยการเพิ่ม ActiveX Control ที่ต้องการเข้าไปใน Toolbox แล้วคลิกมาวางไว้ใน DHTML Designer ตามที่ต้องการ โดยมีขั้นตอนดังนี้

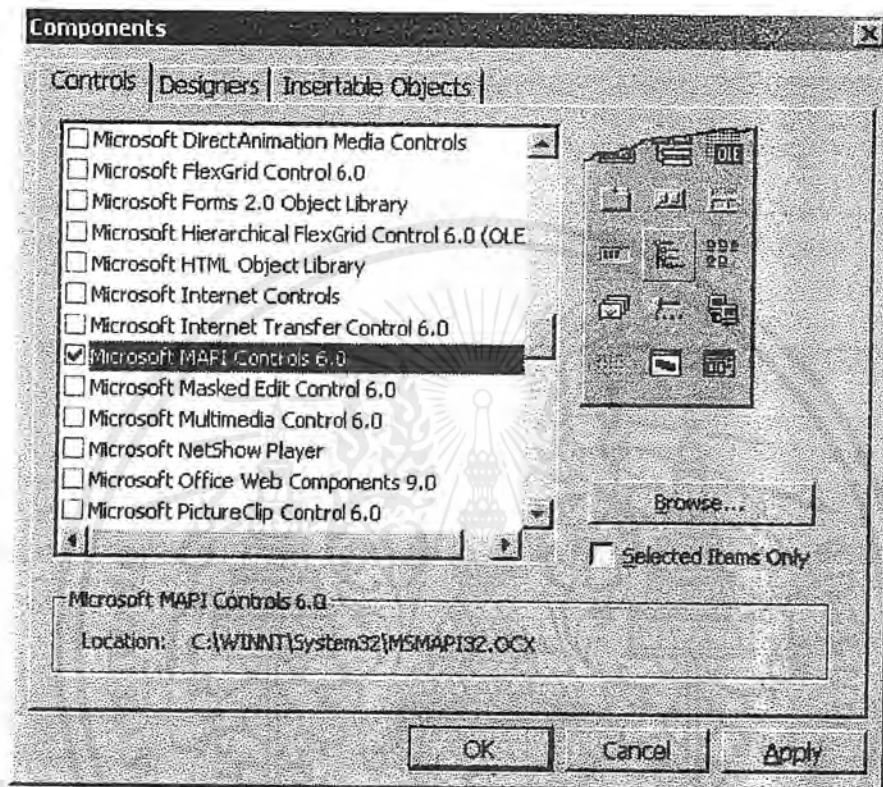
4.7.1. คลิกเมาส์ขวาที่ Object ToolBox



รูปที่ 4.6 Add Component ActiveX เพิ่มเติม โดยการคลิกเมาส์ขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.7.2. คลิกเลือก checkbox Component ของ ActiveX Control ที่เราต้องการใช้งาน



รูปที่ 4.7 ภาพ Select ActiveX สำหรับเพิ่มเติมใน DHTML

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OnUnload เป็นอีเวนต์ที่เกิดเมื่อเราปิดเพจนั้น ช่วงนี้เราจะจัดการคันทรีพายกร หรือคันทรีออบเจกต์ที่ไม่จำเป็นต้องใช้งานให้ระบบ

Terminate เป็นอีเวนต์ที่เกิดก่อนที่เว็บเพจจะถูกเคลียร์ออกไป (เกิดทีหลัง onUnload)

4.9 การเก็บ Cookie

เราจะใช้ Procedure PutProperty ในการเก็บ Cookie ซึ่งมีรูปแบบดังนี้

```
Public Sub PutProperty(objDocument As HTMLDocument, strName As String, vntValue As Variant, Optional Expires As Date)
```

สำหรับความหมายของพารามิเตอร์แต่ละตัวเป็นดังนี้

ObjDocument คือ ออบเจกต์ Document ที่จะเก็บ Cookie

StrName คือ ชื่อของ Cookie

VntValue คือ ค่าของ Cookie

Expire คือ วันที่ที่ Cookie นั้นหมดอายุ โดยเราจะกำหนดหรือไม่ก็ได้

4.10 การอ่านค่า Cookie

เราจะใช้ Procedure GetProperty ในการอ่าน Cookie ซึ่งมีรูปแบบดังนี้

```
Public Sub GetProperty(objDocument As HTMLDocument, strName As String) As Variant
```

สำหรับความหมายของพารามิเตอร์แต่ละตัวเป็นดังนี้

ObjDocument คือ ออบเจกต์ Document ที่จะเก็บ Cookie

StrName คือ ชื่อของ Cookie ที่ต้องการ

บทที่ 5

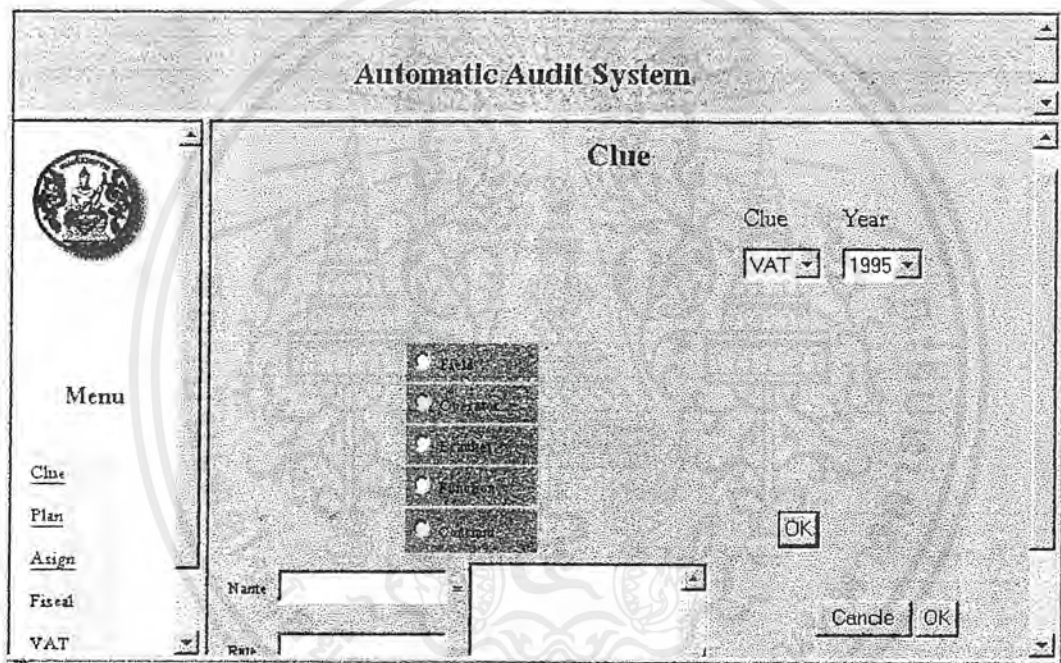
โครงสร้างและส่วนต่าง ๆ ของโปรแกรมระบบตรวจสอบภาษี

การออกแบบโปรแกรมนั้น มีส่วนประกอบของ โปรแกรมอยู่ทั้งหมด 6 ส่วน

5.1 Clue

เป็นส่วนที่ออกแบบมาเพื่อที่จะสามารถที่จะเข้าไปตรวจสอบผู้เสียภาษีอากรเป็นรายๆ โดย สามารถ เข้าได้ในรายละเอียดทั้งหมด และ สามารถรู้ได้ว่า ผู้เสียภาษีนั้นมีการหนีภาษีหรือไม่

รูปร่างของ Clue จะแสดงดังรูปที่ 5.1



รูปที่ 5.1 แสดงหน้าต่างของ Clue

ใน โปรแกรมนี้มีรายละเอียดดังนี้

- 5.1.1 Field เป็นส่วนที่เลือกประเภทของภาษีประกอบด้วย VAT และ CIT
- 5.1.2 Operator เป็นการเลือกใส่เครื่องหมายการคำนวณประกอบด้วย บวก ลบ คูณ และหาร
- 5.1.3 Bracket ในส่วนนี้เป็นการเลือกให้ใส่วงเล็บเปิดและปิด
- 5.1.4 Function เป็นฟังก์ชันที่ใช้ในการคำนวณและเปรียบเทียบปีของภาษีที่ทำการตรวจสอบกับปีก่อนหน้าคือ ฟังก์ชัน N-1
- 5.1.5 Constant เป็นให้ผู้ใช้สามารถกำหนดค่าลงไปได้ตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 Plan

เป็นส่วนที่ออกแบบมาเพื่อที่จะสามารถกำหนดกลุ่มของผู้เสียภาษี เป็นกลุ่มๆ เพื่อที่จะลงไปตรวจสอบได้ ในการแบ่งนั้น คือ เราทำการกำหนด รายการต่างๆ ที่เราต้องการที่จะทราบ โดย สามารถที่จะ AND , OR หรือ END กันได้ โดยการทำให้ทั้ง 3 นี้ ทำเพื่อที่เราจะ ระบุ หรือ กำหนด ข้อมูลตายตัวไปเลย โดยคำตอบที่ได้จะเป็น จำนวนข้อมูลที่อยู่ในฐานข้อมูล โดยนำมาตรวจสอบกับ ส่วนที่เราตั้งไป แล้วเราจะทราบว่า ข้อมูลมีอยู่ที่ Recode รูปออก Plan เป็นดังรูปที่ 5.2

Subject	Oper	attribute	linker
()
()
()
()

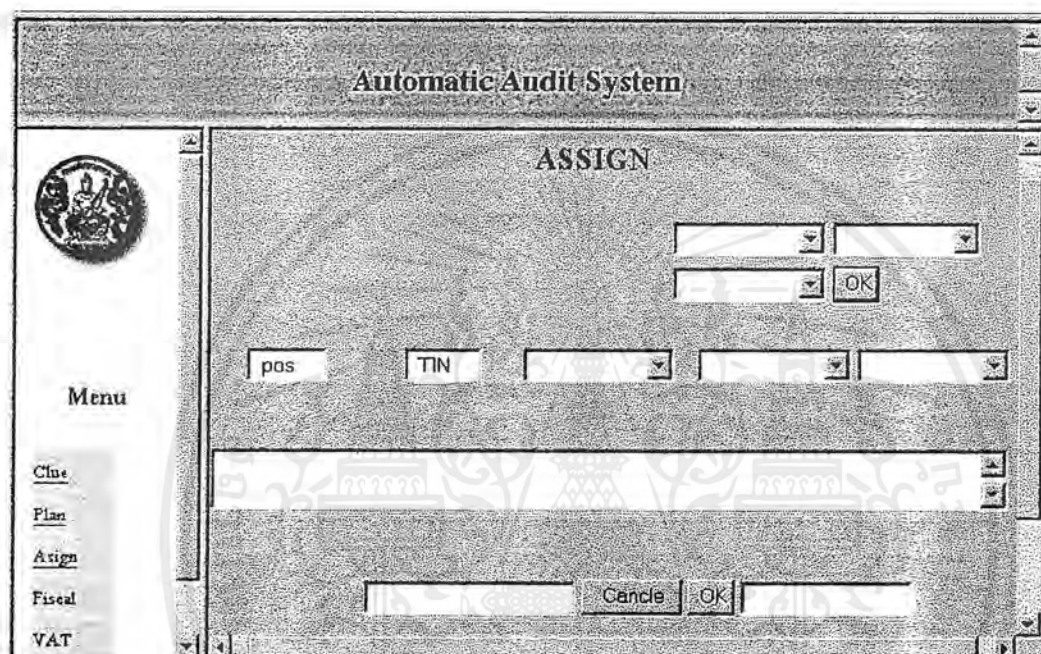
รูปที่ 5.2 แสดงหน้าต่างของ Plan

โดยมีรายละเอียดดังนี้

- 5.2.1 Year คือ ปี ภาษีที่เราต้องการที่จะทราบ
- 5.2.2 Subject คือ ส่วนที่เราต้องการเรียกดูข้อมูล
- 5.2.3 Oper คือ เครื่องหมายที่ใช้เปรียบเทียบ เช่น =, <, >, <=, >= เพื่อใช้ในการเปรียบเทียบ แล้วทำการดึงข้อมูลจาก ฐานข้อมูลมา
- 5.2.4 Attribute คือ ส่วนที่เป็นรายละเอียดภายในของแต่ละ Subject
- 5.2.5 Linker คือ AND, OR และ END ใช้เพื่อ เชื่อมการตรวจสอบหรือ จบการเลือกการตรวจสอบ

5.3 ASSIGN

เป็นส่วนที่คล้ายกับ Plan แต่จะสามารถที่จะเรียกดูรายชื่อ ได้ และสามารถที่จะ จัดลำดับเรียงตาม มาก ไป น้อย , น้อย ไป มาก หรือ Random ได้ ทำมาเพื่อที่จะ ได้สามารถที่จะดูผู้เสียภาษีเป็นพวกๆ เหมือนกับ Plan และสามารถกำหนดการเรียงค่า เช่นเรียงมากไปน้อย โดยเรียงจาก เงิน ได้ของผู้เสียภาษี ทำให้สามารถที่ดู พวก เสียภาษีน้อยไป หรือ มากไป ได้ หน้าต่างของ Assign เป็นดังรูปที่ 5.3



รูปที่ 5.3 แสดง หน้าต่างของ Assign

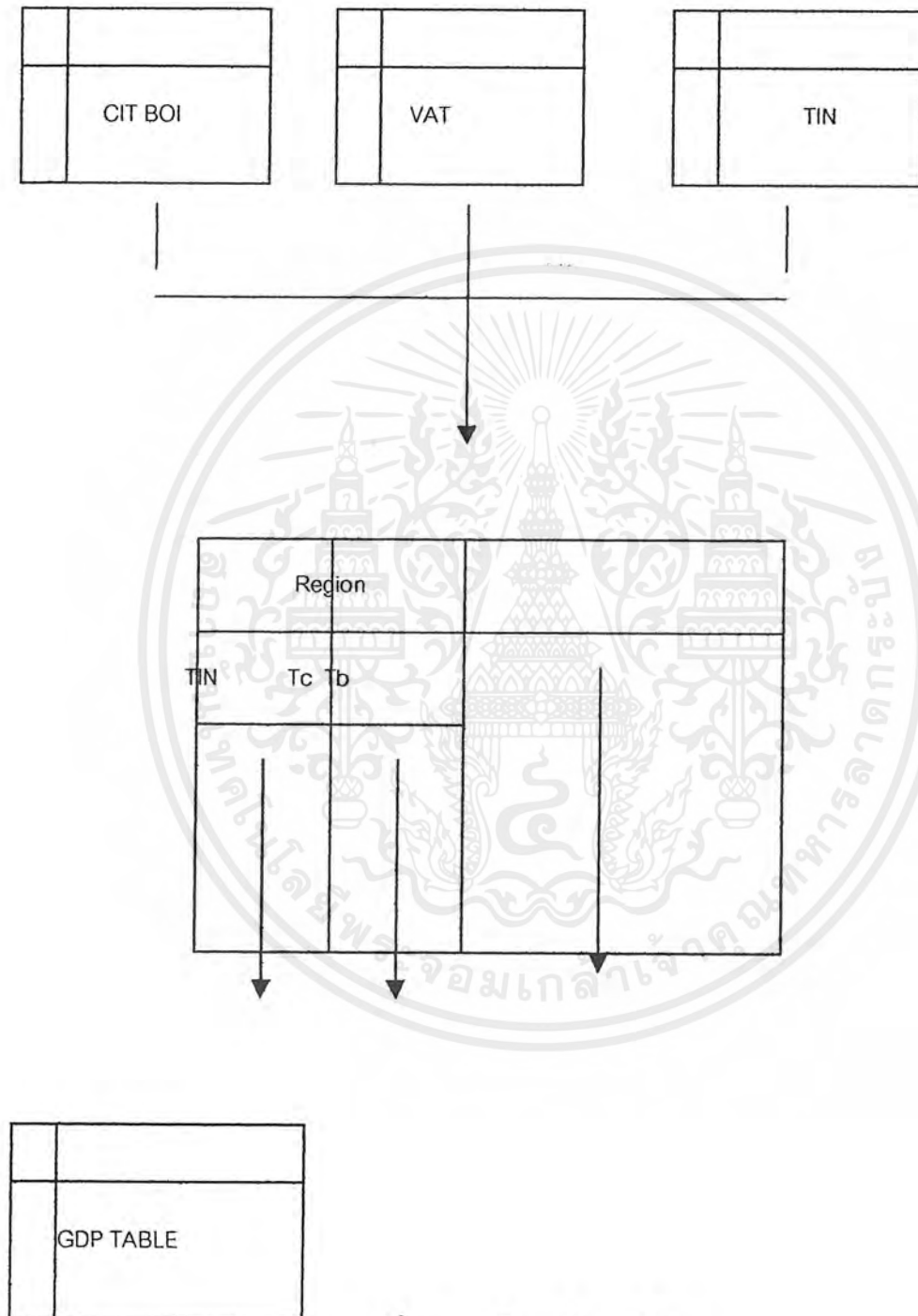
โดยที่ เราจะ Fix Pos และ TIN ไว้ เพื่อจะได้ดูได้ง่าย และสามารถใส่เงื่อนไข ได้ 3 เงื่อนไข แต่ไม่จำเป็นต้องใส่ทั้งหมด

5.4 Fiscal

Fiscal System ถูกออกแบบเพื่อให้ง่ายต่อการใช้งานของ User โดยเฉพาะในส่วนของ MAP ซึ่งจะ ต้องทำการออกแบบ ให้บ่งถึงการ ใช้ที่งานที่บ่งบอกถึงการคลิกเข้าไปดูแต่ละพื้นที่ได้ การออกแบบส่วนใหญ่ เน้นไปที่การใช้งาน DHTML และ ACTIVE X

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างฐานข้อมูล



รูปที่ 5.4 แสดงโครงสร้างฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5 Form VAT

เป็นการดึงข้อมูลจากฐานข้อมูลมาแสดงผลเพื่อใช้ในการดูข้อมูลเป็นส่วนหลักดังรูปที่

Value Added Tax		Nr 30		For official use	
				Document Control nr	From received
A TIN		Branch		B Filing VAT return with branches	
VAT taxpayer is name		Name of business		Join submission at:	Separate declaration
Biz address		Biz address		HQ	HQ
				Branch Nr	Branch Nr
				C	Nra
				on time	out time
				for year	
				Jan	Apr
				Feb	May
				Mar	Jun
				Jul	Oct
				Aug	Nov
				Sep	Dec
Phone		Post code			
For manual receipts		POS receipts			
Book Nr	Amount	Nr	Pay amount		
Signed	Date	(Cashier)	Payment date		
				Instalment	
D		Tax computation			
Output Tax		+ sales			
1 Sales amount II add		- purch			
2 (-) 0% rate sales					
3 (-) Exempted sales					
4 Taxable sales	(1-2-3)				
5 Month Output tax					
Input Tax		+ purch			
6 Purch amount II add		- sales			
7 Input VAT					
Value Added Tax					
8 Tax to be paid					
9 Tax over paid					
10 tax credits					
11 Payable tax					
12 Overpaid tax					
(In letters)					
Late or additional filing					
13 Surcharges					
14 Penalties					
15 Total Tax + surcharges + penalties					
16 Total overpaid taxes after deductions of surcharges + penalties					
(In letters)					
E Refund request			F I/we hereby certify that information declared are correct and in case of ...		
Forwarding credit			Signature		
			Date		

รูปที่ 5.5 แสดง form VAT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.6 Form CIT

เป็นการดึงข้อมูลจากฐานข้อมูลมาแสดงผล เพื่อใช้ในการดูข้อมูล เป็นส่วนหลัก ดังรูปที่ 5.6

Under section 68 and 69 of the Revenue Code			
Name		TIN	pre-examination
Address			Review by
Village	Road	Accounting period	
District	Province	Since	
Telephone	Post code	Until	
For manual procedure		Cash register reception	
Receipt volume	Sena	Year	A
Amount		Pay amount	B
Signed	Receiver	Payment date	C
Date			
The address above is the same as last year		New address	D
Date registered as Juristic Co.	Co. Registration Nr.	Shares held by foreigners	
Province	45	Text30 % of Nr shares	
Country	Nr of branches	DJH	
Text31 Co or Part. stabl under Th law	Text33 Others under foreign law	Text35 Foreign Gov. Business	Text38 Intnl. carrier
Text32 Co or Part. stabl under foreign law	Text34 Securities exchange of Th.	Text36 Joint venture	
		Text37 SIBF business	
Business undertaken (state in order according of importance, sales or income)			
1	2	3	
Text40	Text42	Text43	
In case of being a branch of foreign company			
1 Date commencing operation in Th.			
Head office address			
2 Name of representative			
Office, Number, road, valley and province, district		TIN	Text47
Amphlet Code		Post code	Text51
Text50			
Income Tax to be Paid			
1	Text52 Total CIT paymeng	Text53 Paid in excess (from 2 (6) page 2)	Text54
			Text55
In letters			
Certification of Manager, partner or Director			
I/we have examined the particulars in h-etax return and the attached documents.....			
Signed		Signed	
Title		Title	
		Seal	
Name of authorized auditor	License Nr.	Audit company TIN	

รูปที่ 5.6 แสดง Form CIT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

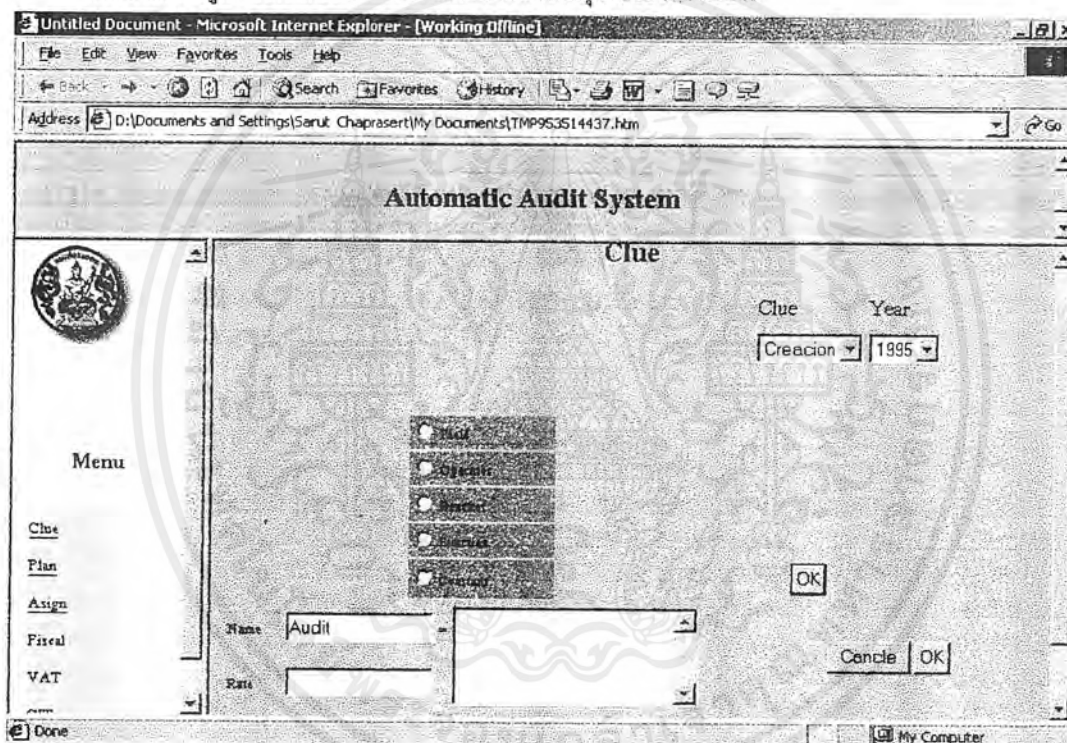
ผลการทดสอบ

ทำการทดสอบในแต่ละส่วนของโปรแกรม โดยแบ่งออกเป็นแต่ละส่วนดังนี้

6.1 ส่วนของ Clue

ในส่วนแรกเป็นการแสดง โปรแกรมในส่วนของ Clue เป็นการตรวจสอบรายชื่อผู้เสียหาย โดยทำการ คีย์ข้อมูลของผู้เสียหายลงไปแล้ว โปรแกรมจะทำการตรวจสอบว่าผู้เสียหายคนนั้นอยู่ในขอบข่ายที่ต้องเข้าไป ทำการตรวจสอบหรือไม่ โดยจะมีข้อความแสดงบอกให้รู้ขั้นตอนมีดังนี้

6.1.1 ใส่ชื่อผู้เสียหายลงไป ในช่อง Name แล้วกดปุ่ม OK ที่ด้านบน



รูปที่ 6.1 แสดงโปรแกรมในส่วนของ Clue

6.1.2 หลังจากที่ใส่ชื่อไปแล้วจะแสดงข้อความที่ในส่วนของ Textbox ทางด้านขวามือ แล้วทำการกดเลือกตามรายด้านบน จะประกอบด้วย Field , Operator , Bracket , Function , Constant ในส่วนนี้จะเป็นรายละเอียดของการเลือกตรวจสอบว่าจะตรวจสอบภาษีประเภทไหน ในการเลือกแต่ละรายการเมื่อเสร็จแล้วต้องกดที่ปุ่ม OK ด้านบน แล้วรายละเอียดต่าง ๆ จึงจะแสดงให้เห็นใน Textbox ด้านล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The screenshot shows a software window titled "Automatic Audit System". Inside, there is a "Clue" dialog box. On the left side of the dialog, there is a "Menu" section with a list of items: Clue, Plan, Assign, Fiscal, VAT, and CIT. The "VAT" item is selected. In the center of the dialog, there is a list of selected items: VAT, ITN, TAXYEA, TAXMON, REGIST, and PAYDAT. Below this list, there are two text boxes: "Name" containing "Audit" and "Rate" which is empty. At the bottom right, there are "OK", "Cancel", and "OK" buttons. The background of the dialog box is a light gray with a faint watermark of a sunburst.

รูปที่ 6.2 แสดงรายละเอียดของรายการที่เลือก

6.1.3 หลังจากทีเลือกใส่รายการต่าง ๆ และเงื่อนไขแล้วให้คลิกที่ปุ่ม OK โปรแกรมจะทำการดึงรายละเอียดของข้อมูลต่างที่ใส่ไว้มาแสดงใน Text box ข้างล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The screenshot shows the 'Automatic Audit System' window. On the left is a 'Menu' sidebar with options: Clue, Plan, Assign, Fiscal, and VAT. The main area has 'Clue' and 'Year' dropdowns set to 'Creacion' and '1995'. A central panel contains radio buttons for 'Field', 'Operator', 'Branch', 'Function', and 'Constant'. The 'Field' option is selected. Below this, there are input fields for 'Name' (containing 'Audit') and 'Rate' (containing '0.07'). A dropdown menu shows the selected field: 'TAXMON*(AYD ATR 5)-Q(1)*0.07'. There are 'OK', 'Cancel', and 'OK' buttons.

รูปที่ 6.3 แสดงรายละเอียดที่เลือกไว้

6.1.4 หลังจากที่ใส่เงื่อนไขต่าง ๆ ลงไปแล้ว ให้ทำการคลิกที่ปุ่ม OK ในส่วนของโปรแกรมจะทำการแสดงข้อความขึ้นมาว่าจากข้อมูลและเงื่อนไขของผู้เสียภาษีทั้งหมดที่ใส่ลงไปนั้น ว่าผู้เสียภาษีบุคคลผู้นั้นอยู่ในขอบข่ายที่ต้องเข้าไปตรวจสอบหรือไม่

This screenshot is identical to the previous one, showing the 'Automatic Audit System' window with the 'Field' option selected. The 'Name' field contains 'Audit' and the 'Rate' field contains '0.07'. The dropdown menu shows the selected field: 'TAXMON*(AYD ATR 5)-Q(1)*0.07'. There are 'OK', 'Cancel', and 'OK' buttons. The text 'ปิดปกติ' (Closed normally) is visible at the bottom center of the main area.

รูปที่ 6.4 แสดงผลลัพธ์ของการตรวจสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 ส่วนของ Plan

6.2.1 ทำการเขียน โปรแกรมของส่วน PLAN โดยมีรูปแบบดังรูป

The screenshot shows a software interface for an 'Automatic Audit System'. The main window is titled 'Automatic Audit System'. On the left side, there is a 'Menu' sidebar with the following items: Clue, Plan, Assign, Fiscal, and VAT. The main area of the window is titled 'PLAN'. At the top of this area, there is a 'Year' label followed by two dropdown menus. Below this is a table with three columns: 'Subject', 'Oper. attribute', and 'linker'. The table contains four rows, each starting with a parenthesis '(' and ending with a parenthesis ')'. At the bottom of the window, there are 'Cancel' and 'OK' buttons.

รูปที่ 6.5 แสดงหน้าต่างของ plan

6.2.2 จะมีส่วนต่างๆดังนี้

6.2.2.1 ส่วนกำหนดปีในการ ตรวจสอบ

6.2.2.2 ส่วนกำหนด รายละเอียด ในการตรวจสอบ

6.2.2.3 ปุ่มยกเลิกค่าที่กำหนดไป

6.2.2.4 ปุ่ม OK จะทำการตรวจสอบข้อมูลใน DATABASE แล้ว นำค่ามาแสดงที่ ช่องว่าง

6.2.2.5 Text box แสดงจำนวนตารางทั้งหมด

6.2.3 จะทำการ Link ข้อมูลกับ DATABASE เพื่อตรวจสอบสิ่งที่กำหนดมานั้น มีที่ Table

6.2.4 สามารถที่จะมีการ AND , OR และ () เพื่อให้สามารถกำหนดรายละเอียดที่ต้องการได้ครบและตามที่ต้องการ

6.2.5 เวลาแสดง Block ที่ใช้ในการกำหนดต่างๆ จะแสดงที่ละบรรทัด แล้วถ้าทำการเลือก AND หรือ OR จะแสดง Block บรรทัดต่อไปขึ้นมา ดังรูป

รูปที่ 6.6 แสดงเริ่มแรกเมื่อเข้าโปรแกรม

6.2.6 ถ้าเลือก END จะ Show ปุ่มต่างๆ เพื่อให้เลือกว่า OK เพื่อหาตารางเลย หรือ เลือก ยกเลิกเพื่อที่จะกำหนดค่าใหม่

6.2.7 ถ้าเลือก OK ก็ จะแสดงค่าของ Recode ใน ฐานข้อมูลออกมา ดังรูป

รูปที่ 6.7 แสดงการเลือกแถวเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Automatic Audit System

PLAN

Year

1995 1996

Subject	Oper.	attribute	linker
(ampcode)	=	11800210) AND
(subcode)	=	20) AND
(typecode)	=	9) END

Cancel ok 5

Menu

Clue

Plan

Assign

Fiscal

VAT

รูปที่ 6.8 แสดงการเลือกหลายแถว

6.3 ส่วนของ Assign

6.3.1 ทำการเขียน โปรแกรมของส่วน Assign โดยมีรูปแบบดังรูป

Automatic Audit System

ASSIGN

pos TIN

Cancel OK

Menu

Clue

Plan

Assign

Fiscal

VAT

รูปที่ 6.9 แสดงการหน้าต่างของ Assign

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3.2 รายละเอียดของส่วนต่างๆ จะมีส่วนต่างๆดังนี้

6.3.2.1 ส่วนกำหนดว่าจะให้จัดลำดับความสำคัญตัว ไหนและ จัดลำดับยังง มากไปน้อย , น้อยไปมาก หรือ Random

6.3.2.2 ส่วนกำหนด รายละเอียด ในการตรวจสอบ โดยจะมีลำดับ และ TIN เป็นตัว fix อยู่ ซึ่งจะไม่สามารถเปลี่ยนได้

6.3.2.3 ปุ่มยกเลิกค่าที่กำหนดไป

6.3.2.4 ปุ่ม OK จะทำการตรวจสอบข้อมูลใน DATABASE แล้ว นำค่ามาแสดงที่ ช่องว่าง

6.3.2.5 Text box ให้ใส่ ชื่อ ผู้เรียง

6.3.2.6 Text box ใส่กลุ่มผู้เรียง

6.3.3 การกำหนดการแสดงผล สามารถกำหนดได้ว่าจะ แสดง ทั้งหมดที่ ตาราง

6.3.4 ส่วน Link จะทำการ Link ข้อมูลกับ DATABASE เพื่อ Show ข้อมูลทั้งหมด ตามจำนวน Table ที่ กำหนด

6.3.5 การเลือกค่า ถ้าเลือก OK จะ Show ข้อมูลใน DATABASE ตามจำนวน Table ที่ กำหนดดังรูป

Automatic Audit System

ASSIGN

Regioncode Random

1 OK

pos TIN Regioncode ampcode

1	1214513213	5	11200200
---	------------	---	----------

Wichai Cancel OK new

Menu

Chue

Plan

Assign

Fiscal

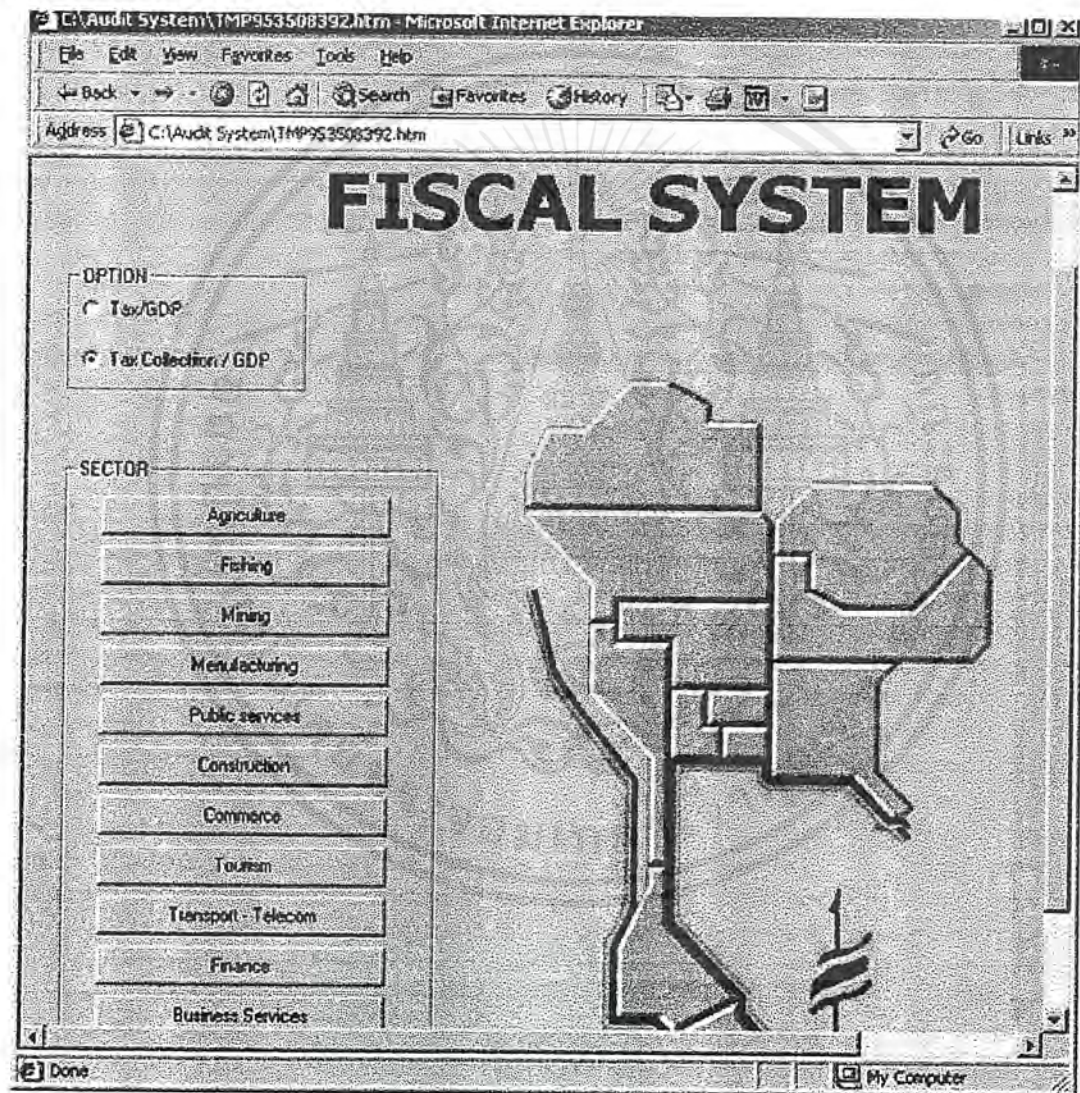
VAT

รูปที่ 6.10 แสดงคำตอบที่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4 ส่วนของ Fiscal

เป็น Module สำคัญส่วนหนึ่งในระบบ Audit เพราะ Fiscal System เป็นระบบที่ทำงาน โคนทำหน้าที่ที่ตรวจสอบ และคัดเลือก line ในการตรวจสอบ หรือเลือก line ในการสอบสวนกลุ่มผู้เสียภาษี เขตใด เขตหนึ่ง , ภาคธุรกิจใด ธุรกิจหนึ่ง โดยอาศัยวิธีการเปรียบเทียบ stat ฐานภาษีของกลุ่มนั้น กลุ่มธุรกิจนั้นๆ เป็นเปอร์เซ็นต์ แล้วคัดเลือก ผู้เสียภาษีในเขตใดเขตหนึ่ง ซึ่งมีเปอร์เซ็นต์ของ ฐานภาษี , ภาษีที่จะต้องชำระ ต่อ ค่า GDP ของแต่ละพื้นที่



รูปที่ 6.11 แสดงหน้าจอหลักของ ระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4.1 การทำงานของระบบ

Interface ของระบบ ถูกออกแบบบน Web Base Technology โดยใช้ Internet Explorer 4.0 หรือรุ่นหลังจากนี้ในการ Browse

จากหน้าจอหลักจะแบ่ง Interface เป็นส่วนใหญ่ สองส่วนคือ

6.4.1.1 ส่วน Input สำหรับให้ User กำหนดค่า Option และเลือก Sector ในการตรวจสอบ

6.4.1.2 Tax Option เลือกคำนวณจาก Tax Base หรือ Tax Collection

6.4.1.3 SECTOR เลือกภาคธุรกิจในการเข้าถึงกลุ่มธุรกิจแต่ละกลุ่ม

6.4.1.4 MAP เลือกพื้นที่

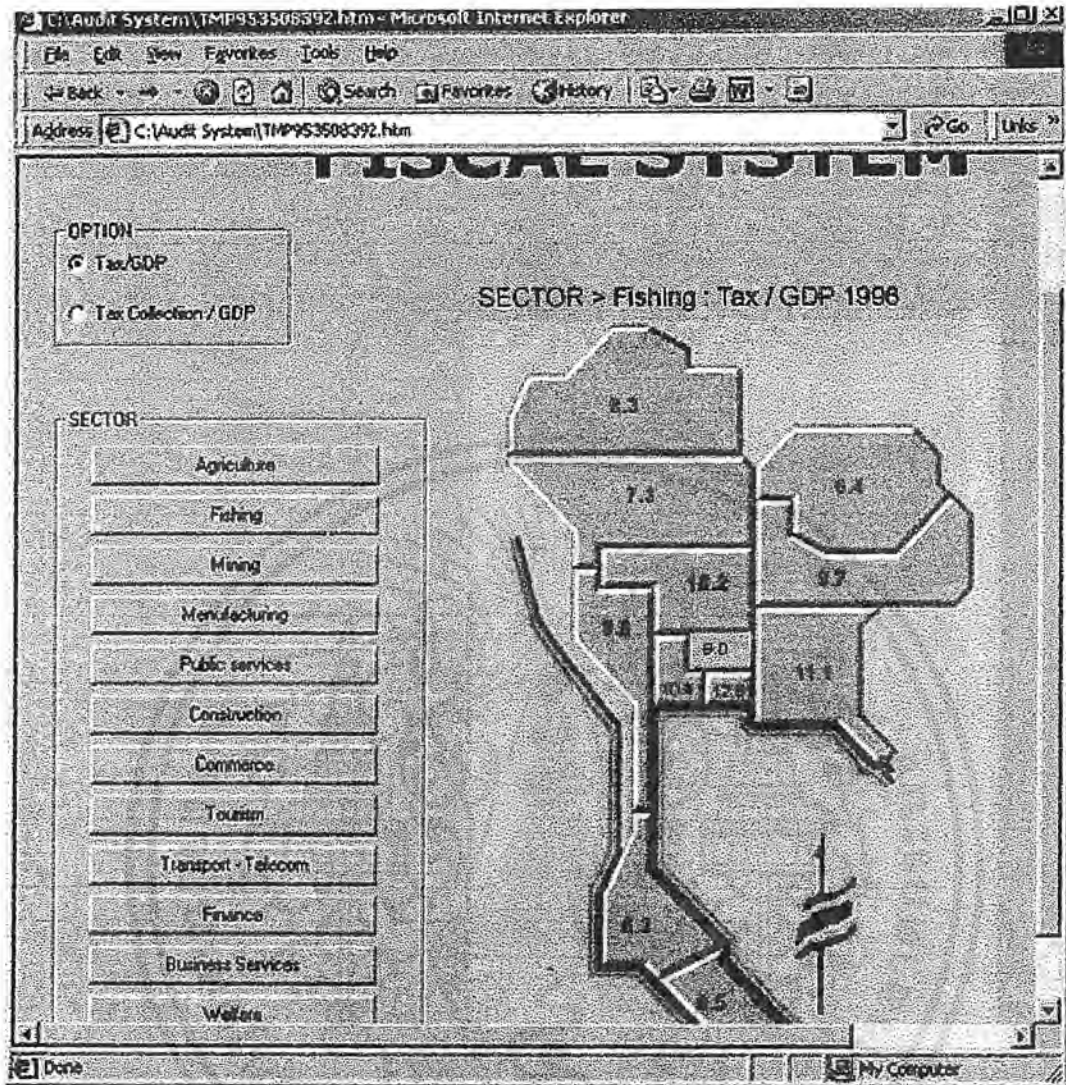
6.4.2 ส่วน Output สำหรับ แสดงค่าผลการคำนวณเพื่อใช้ในการพิจารณาตรวจสอบต่อไป

6.4.2.1 MAP แสดงค่าการคำนวณในแต่ละพื้นที่ (ค่าที่แสดงมีสองสี สีแดง และสีเขียว

6.4.2.2 text แสดง Status ว่าคำนวณจากอะไร

6.4.3 ทดสอบ

ข้อมูลที่น่าทดสอบเป็นข้อมูลที่สร้างขึ้นเอง เปิด Fiscal System Page จากหน้าจอเมนูหลัก เลือก Option ที่ใช้ในการคำนวณ Tax / GDP คลิกปุ่ม Fishing เพื่อกำหนด กลุ่มธุรกิจในการตรวจสอบ จากผลการทดลอง จะเห็นว่าค่าที่แสดงบน Map จะมีสีต่างกัน คือ สีเขียว และ สีแดง ซึ่ง สีเขียว หมายถึง ไม่มีความเสี่ยง สีแดงหมายถึง พื้นที่นั้น(กลุ่มธุรกิจ) ที่เลือก มีความเสี่ยง




รูปที่ 6.12 แสดง Fiscal

6.5 Form VAT

เป็นการดึงค่ามาจากฐานข้อมูลเพียงอย่างเดียว มี form ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

 Value Added Tax		Nr 30		For official use	
				Document Control nr	From received
A TIN		Branch		B Filing VAT return with branches	
VAT taxpayer is name		Join submission at		Separate declaration	
Name of business		HQ		HQ	
Biz address		Branch Nr		Branch Nr	
		C nor		Add N/a	
		on time		cut time	
		for year			
		Jan		Apr	
		Feb		May	
		Mar		Jun	
		Jul		Oct	
		Aug		Nov	
		Sep		Dec	
Phone		Post code			
For manual receipts		PDS receipts			
Book Nr		Nr		Pay amount	
Amount		(Cashier)		Payment date	
Signed				Instalment	
Date					
D					
Tax computation					
Output Tax		+ sales			
1 Sales amount If add		purch			
2 (-) 0% rate sales					
3 (-) Exempted sales					
4 Taxable sales (1-2-3)					
5 Month Output tax					
Input Tax		+ purch			
6 Purch amount If add		sales			
7 Input VAT					
Value Added Tax					
8 Tax to be paid					
9 Tax over paid					
10 tax credits					
11 Payable tax					
12 Overpaid tax					
(In letters)					
Late or additional filing					
13 Surcharges					
14 Penalties					
15 Total Tax+ surcharges+penalties					
16 Total overpaid taxes after deductions of surcharge + penalties					
(In letters)					
E Refund request			F I/we hereby certify that information declared are correct and in case of ...		
Forwarding credit			Signature		
			Date		

รูปที่ 6.13 แสดง Form VAT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.6 Form CIT

เป็นการดึงค่ามาจากฐานข้อมูลเพียงอย่างเดียว มี form ดังรูป

Under section 68 and 69 of the Revenue Code			
TIN		pre-examination	Review by
Name		Accounting period	
Address		Since	
Village	Road	Unit	
District	Province	1: For	
Telephone	Post code	2: Add	3: By pay
For manual procedure		Cash register reception	
Receipt volume	Serial	Year	
Amount		Pay amount	
Signed	Receiver	Payment date	
Date			
The address above is the same as last year		New address	
Date registered as Juristic Co.		Co. Registration Nr.	Shares held by foreigners
Province		45	Text30 % of Nr shares
Country		Nr of branches	DJH
Text31	Co or Part, stabil under Thai law	Text33	Others under foreign law
Text32	Co or Part, stabil under foreign law	Text34	Securities exchange of Tt.
		Text35	Foreign Gov. Business
		Text36	Joint venture
		Text37	SIBF business
		Text38	Inter. comer
Business undertaken (state in order according of importance, sales or income)			
1	Text40	2	Text42
		3	Text43
In case of being a branch of foreign company			
1 Date commencing operation in Th			
Head office address			
2 Name of representative			
Office, Number, road, valley and province, district		TIN	Text47
Amphet Code		Post code	Text51
Text50			
Income Tax to be Paid			
1	Text52	Text53	Text54
	Total CIT payment	Paid in excess (from 2 (6) page 2)	Text55
In letters			
Certification of Manager, partner or Director			
I/we have examined the particulars in tax return and the attached documents.....			
Signed		Signed	
Title		Title	
		Seal	
Name of authorized auditor	Licensee Nr.	Audit company TIN	

รูปที่ 6.14 แสดง Form CIT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทวิจารณ์ และสรุป

จากที่ได้ทำโครงการระบบการตรวจสอบภาษีนี้แล้ว จะเห็น ได้ชัดว่าการทำงานมีการตรวจสอบที่รวดเร็วขึ้นอย่างเห็น ได้ชัด ซึ่งก่อนหน้าการทำงานของกรมสรรพากรนั้น ทำงานช้ามาก เพราะ ผู้เสียภาษีมีจำนวนมาก แต่จำนวนของ พนักงานตรวจสอบกลับมีน้อย ทำให้ไม่สามารถที่จะตรวจสอบ ได้หมดทุกคน

โปรแกรมนี้จะช่วยในการตรวจสอบให้สามารถที่จะไปทำการตรวจสอบผู้เสียภาษีที่น่าสงสัย ได้ดี ซึ่ง ทำให้มี ประสิทธิภาพในการทำงานมากกว่าเดิม และ พนักงานตรวจสอบ สามารถที่จะทำงาน ได้อย่างทั่วถึง

การพัฒนาโปรแกรมนั้นสามารถที่จะพัฒนาให้ใช้กับระบบที่ใหญ่ขึ้นมาได้ และ ใช้กับภาษีอื่นๆ ของ กรมสรรพากรได้เช่นกัน ซึ่งใน โปรแกรมระบบการตรวจสอบผู้เสียภาษี นี้จะทำการ ตรวจสอบ 2 ภาษี คือ ภาษีมูลค่าเพิ่ม และ ภาษีเงิน ได้นิติบุคคล



ภาคผนวก ก

เลขประจำตัวผู้เสียภาษีอากร

หมายเลขที่กรมสรรพากรกำหนดให้ผู้เสียภาษีอากร หรือผู้จ่ายเงินได้แต่ละรายมี และ ใช้เลขประจำตัวนั้น ในการปฏิบัติการ ตามมาตรา 3 เอกาเทศ แห่งประมวลรัษฎากร “มาตรา 3 เอกาเทศ เพื่อประโยชน์ในการจัดเก็บภาษีอากร ตามประมวลรัษฎากร อธิบดีมีอำนาจ กำหนดให้ผู้มี หน้าที่เสียภาษีอากรและ ผู้มีหน้าที่จ่ายเงิน ได้มีและ ใช้เลขประจำตัวในการปฏิบัติการ ตามประมวลรัษฎากร ได้ตามหลักเกณฑ์ และ วิธีการที่ อธิบดีกำหนด ทั้งนี้ โดยอนุมัติรัฐมนตรี การกำหนด ตามวรรคหนึ่งให้ประกาศในราช กิจจานุเบกษา”

บัตรประจำตัวผู้เสียภาษีอากร

บัตรที่แสดงเลขประจำตัวผู้เสียภาษีอากร ชื่อ และที่อยู่ของผู้เสียภาษีอากร ซึ่งกรมสรรพากรออกให้แก่ผู้เสียภาษีอากร หรือผู้มีหน้าที่จ่ายเงิน ได้ สำหรับ ใช้ปฏิบัติการตามประมวลรัษฎากร

หมายเลขประจำตัวผู้เสียภาษีอากร

หมายเลขประจำตัวผู้เสียภาษีอากร ประกอบด้วยหมายเลข 10 หลัก แบ่งออกเป็น 4 ระบบ คือ

1. ระบบบุคคลธรรมดา
2. ระบบคณะบุคคล
3. ระบบนิติบุคคล
4. ระบบผู้จ่ายเงินได้

ผู้เสียภาษีอากร

ผู้มีหน้าที่เสียภาษีอากรตามประมวลรัษฎากร ผู้มีหน้าที่จ่ายเงินได้ และให้หมายรวมถึง

1. ผู้แทนผู้เสียภาษีอากรที่มีอำนาจตามกฎหมาย
2. ผู้แทนโดยชอบธรรม ผู้อนุบาล ผู้พิทักษ์ หรือ ผู้จัดการกิจการอันก่อให้เกิดเงินได้พึงประเมินแล้วแต่กรณี
3. ผู้รับมอบอำนาจ

บุคคลธรรมดา

ผู้เสียเงิน ได้ซึ่งมีหน้าที่ต้องเสียภาษีเงินได้ บุคคลธรรมดาตามประมวลรัษฎากร

1. ผู้เยาว์
2. ผู้ที่ศาลสั่งให้เป็นคนไร้ความสามารถหรือ เสมือนไร้ความสามารถ
3. ผู้ถึงแก่ความตายระหว่างปีภาษี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. กองมรดกที่ยังไม่แบ่ง

5. ผู้อยู่ในต่างประเทศ เว้นแต่ คนต่างด้าวที่เดินทางเข้ามาในประเทศไทย ครั้งละ ไม่เกิน 14 วัน และรวมกัน ไม่เกิน 90 วัน ในปีภาษีหนึ่งๆ

คณะบุคคล

คณะบุคคลที่มีใช้นิติบุคคล ห้างหุ้นส่วนสามัญ ซึ่งมีหน้าที่ต้องเสียภาษีเงินได้บุคคลธรรมดาตามประมวลรัษฎากร

นิติบุคคล

บริษัทหรือห้างหุ้นส่วนนิติบุคคลตามมาตรา 39 แห่งประมวลรัษฎากร

ผู้มีหน้าที่จ่ายเงินได้

ผู้จ่ายเงินได้ซึ่งมีหน้าที่หักภาษี ณ ที่จ่ายตาม ประมวลรัษฎากร และ ไม่มีหน้าที่ต้องเสียภาษีเงินได้ ตามประมวลรัษฎากร

เลขประจำตัวสำหรับผู้เสียภาษีเงินได้นิติบุคคล

1. ขึ้นแบบคำขอ ลป. 10 ภายใน 60 นับแต่วันที่ได้รับจดทะเบียน เป็นนิติบุคคลหรือวันที่เริ่มประกอบกิจการในประเทศไทย แล้วแต่กรณี

1.1 ในกรุงเทพมหานคร ให้ยื่นต่อสรรพากรพื้นที่ ณ สำนักงานสรรพากรพื้นที่ ที่สำนักงานใหญ่ตั้งอยู่

1.2 ในจังหวัดอื่น ให้ยื่น ณ ท้องที่ที่สำนักงานใหญ่ตั้งอยู่ ดังนี้

1.2.1 สรรพากรจังหวัด หรือสรรพากรจังหวัด (สาขา) ณ สำนักงานสรรพากรจังหวัด หรือสำนักงานสรรพากรจังหวัด (สาขา)

1.2.2 สรรพากรอำเภอ หรือ สรรพากรอำเภอ (สาขา) ณ สำนักงานสรรพากรอำเภอ หรือสำนักงานสรรพากรอำเภอ (สาขา)

2. ให้แจ้งเลขประจำตัวผู้เสียภาษีอากรและแสดงบัตรประจำตัวผู้เสียภาษีอากรของตน ในการปฏิบัติการเกี่ยวกับภาษีเงินได้นิติบุคคลตามประมวลรัษฎากรเว้นแต่ อธิบดีกรมสรรพากรจะสั่งเป็นอย่างอื่นในการขึ้นแบบ แสดงรายการนำส่งภาษีเงินได้หัก ณ ที่จ่ายทุกกรณี ให้ผู้จ่ายเงินได้ แจ้งเลขประจำตัวของผู้ถูกหักภาษีเงินได้ ณ ที่จ่าย ไว้ในแบบแสดงรายการนำส่งภาษีเงินได้หัก ณ ที่จ่ายด้วย

3. ถ้ามีเลขประจำตัวผู้เสียภาษีอากรมากกว่า 1 หมายเลข ให้แจ้ง ต่อ เจ้าหน้าที่ ณ สถานที่ตามข้อ

1. แล้วแต่กรณี ภายใน 60 วัน นับแต่วันที่มิได้เลขฯ เกินกว่า 1 หมายเลข เพื่อให้กำหนดให้มีเพียงหมายเลขเอกสารนี้เป็นเอกสารที่ส่งมอบไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คือ “ ธนาคารพาณิชย์ที่ได้รับอนุญาตให้ประกอบกิจการวิเทศธนกิจ ให้ยื่นคำร้องขอมีเลขประจำตัวและ บัตรประจำตัวผู้เสียภาษีอากรตามแบบอธิบดีกำหนด สำหรับการเสียภาษีเงินได้จากการประกอบ ธุรกิจวิเทศธนกิจ แยกต่างหาก ธนาคารพาณิชย์ที่ตั้งขึ้นตามกฎหมายไทย หรือที่ตั้งขึ้นตามกฎหมายของ ต่างประเทศและกระทำการกิจการในประเทศไทย ตามมาตรา 60 วรรคหนึ่ง แห่งประมวลรัษฎากร ที่ได้รับ อนุญาตให้ประกอบกิจการวิเทศธนกิจ มีหน้าที่ยื่นคำร้องขอมีเลขประจำตัวและ บัตรประจำตัวผู้เสียภาษี อากรตามแบบที่อธิบดีกำหนด สำหรับการ เสียภาษีเงิน ได้จากการประกอบธุรกิจวิเทศธนกิจในอัตราร้อยละ 10 แยกต่างหาก จากเลขประจำตัวผู้เสียภาษีอากรสำหรับการเสียภาษีในอัตราร้อยละ 30 อีกหนึ่ง หมายถึง เลข ทั้งนี้ ไม่ว่าจะ มีสถานประกอบการที่แห่งก็ตาม”



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

ผู้มีหน้าที่เสียภาษีเงินได้นิติบุคคล ภ.ง.ด. 50

ผู้มีหน้าที่เสียภาษีเงินได้นิติบุคคล ได้แก่ บริษัทหรือห้างหุ้นส่วนนิติบุคคล ที่จดทะเบียนตามประมวลกฎหมายแพ่งและพาณิชย์ และหมายความรวมถึงนิติบุคคลอื่นๆ ที่ไม่ได้จดทะเบียนตามประมวลกฎหมายแพ่งและพาณิชย์ ด้วย ดังนี้

บริษัทหรือห้างหุ้นส่วนนิติบุคคลที่มีหน้าที่เสียภาษีเงินได้นิติบุคคล มีดังนี้

1. บริษัทหรือห้างหุ้นส่วนนิติบุคคลที่ตั้งขึ้นตามกฎหมายไทย ได้แก่

- 1.1 บริษัท จำกัด
- 1.2 บริษัทมหาชน จำกัด
- 1.3 ห้างหุ้นส่วนจำกัด
- 1.4 ห้างหุ้นส่วนสามัญจดทะเบียน

2. บริษัทหรือห้างหุ้นส่วนนิติบุคคลที่ตั้งขึ้นตามกฎหมายต่างประเทศ ซึ่งมีหน้าที่เสียภาษีเงินได้นิติบุคคลในประเทศไทย ก็ต่อเมื่อเข้าเงื่อนไขข้อใดข้อหนึ่ง ดังต่อไปนี้

- 2.1 บริษัทหรือห้างหุ้นส่วนนิติบุคคลต่างประเทสนั้น เข้ามากระทำการในประเทศไทย (มาตรา 66 วรรคแรก แห่งประมวลรัษฎากร)
- 2.2 บริษัทหรือห้างหุ้นส่วนนิติบุคคลต่างประเทสนั้น กระทำการในที่อื่นๆ รวมทั้งในประเทศไทย (มาตรา 66 วรรคสอง แห่งประมวลรัษฎากร)
- 2.3 บริษัทหรือห้างหุ้นส่วนนิติบุคคลต่างประเทสนั้น กระทำการอื่นๆรวมทั้งในประเทศไทยและกิจการที่กระทำนั้นเป็นกิจการขนส่งระหว่างประเทศ (มาตรา 67 แห่งประมวลรัษฎากร)
- 2.4 บริษัทหรือห้างหุ้นส่วนนิติบุคคลต่างประเทสนั้น มิได้ประกอบกิจการในประเทศไทย แต่ได้รับเงินได้พึงประเมินมาตรา 40 (2) (3) (4) (5) หรือ (6) ที่จ่ายจากหรือในประเทศไทย (มาตรา 70)
- 2.5 บริษัทหรือห้างหุ้นส่วนนิติบุคคลต่างประเทสนั้น มิได้เข้ามาทำกิจการในประเทศไทยโดยตรง หากแต่มีลูกจ้างหรือผู้ทำการแทนหรือผู้มาทำการติดต่อ ในการประกอบกิจการในประเทศไทย ซึ่งเป็นเหตุให้ได้รับเงินได้หรือผลกำไรในประเทศไทย (มาตรา 76 ทวิ)

3. กิจการซึ่งดำเนินการเป็นทางค้า หรือหากำไร โดย

- 3.1 รัฐบาลต่างประเทศ
- 3.2 องค์การของรัฐบาลต่างประเทศ
- 3.3 นิติบุคคลอื่นที่ตั้งขึ้นตามกฎหมายของต่างประเทศ

4. กิจการร่วมค้า (Joint Venture) ได้แก่ กิจการที่ดำเนินการร่วมกันเป็นทางค้าหรือหากำไรระหว่างบุคคลดังต่อไปนี้คือ

- 4.1 บริษัทกับบริษัท
- 4.2 บริษัทกับห้างหุ้นส่วนนิติบุคคล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4.3 ห้างหุ้นส่วนนิติบุคคลกับห้างหุ้นส่วนนิติบุคคล
- 4.4 บริษัทและหรือห้างหุ้นส่วนนิติบุคคลกับบุคคลธรรมดา
- 4.5 บริษัทและหรือห้างหุ้นส่วนนิติบุคคลกับคณะบุคคลที่มีหุ้นนิติบุคคล
- 4.6 บริษัทและหรือห้างหุ้นส่วนนิติบุคคลกับห้างหุ้นส่วนสามัญ
- 4.7 บริษัทและหรือห้างหุ้นส่วนนิติบุคคลกับนิติบุคคลอื่น

5. มูลนิธิหรือสมาคมที่ประกอบกิจการซึ่งมีรายได้แต่ไม่รวมถึงมูลนิธิหรือสมาคมที่รัฐมนตรีประกาศกำหนดให้เป็นองค์การหรือสถานสาธารณกุศล

6. นิติบุคคลที่อธิบดีกำหนด โดยอนุมัติรัฐมนตรีและประกาศในราชกิจจานุเบกษาให้เป็นบริษัทหรือห้างหุ้นส่วนนิติบุคคลตามประมวลรัษฎากร

นิติบุคคลที่ไม่ต้องเสียภาษีเงินได้

นิติบุคคลอื่นๆ นอกจากที่กล่าวในข้างต้น และเฉพาะที่ตั้งขึ้นตามกฎหมายไทย เช่น กระทรวง ทบวง กรม องค์การของรัฐบาลหรือสหกรณ์ ไม่มีหน้าที่ต้องเสียภาษีเงินได้นิติบุคคลแต่อย่างใด

อย่างไรก็ตาม ยังมีนิติบุคคลอีกบางประเภทที่เข้าลักษณะต้องเสียภาษีเงินได้นิติบุคคลตามประมวลรัษฎากรแต่ได้รับการยกเว้นตามบทบัญญัติของกฎหมายต่างๆ ได้แก่

1. บริษัทหรือห้างหุ้นส่วนนิติบุคคลตามข้อผูกพันที่ประเทศไทยมีอยู่ตามสัญญาว่าด้วยความร่วมมือทางเศรษฐกิจหรือทางเทคนิคระหว่างรัฐบาลไทยกับรัฐบาลต่างประเทศ
2. บริษัทจำกัดที่ได้รับการยกเว้นภาษีเงินได้ตามกฎหมายว่าด้วยการส่งเสริมการลงทุน
3. บริษัทจำกัดและนิติบุคคลที่มีสภาพเช่นเดียวกับบริษัทจำกัดซึ่งตั้งขึ้นตามกฎหมายไทยหรือกฎหมายต่างประเทศได้รับการยกเว้นภาษีเงินได้นิติบุคคลตามพระราชบัญญัติภาษีเงินได้ปิโตรเลียม
4. บริษัทหรือห้างหุ้นส่วนนิติบุคคลที่อยู่ในประเทศที่มีอนุสัญญาว่าด้วยการเว้นการเก็บภาษีซ้อนกับประเทศไทย ตามเงื่อนไขที่กำหนดในอนุสัญญา
5. มูลนิธิหรือสมาคมที่รัฐมนตรีว่าการกระทรวงการคลังประกาศกำหนดให้เป็นองค์การ หรือสถานสาธารณกุศล

ฐานภาษีของภาษีเงินได้นิติบุคคล

ภาษีเงินได้นิติบุคคล คำนวณจากเงินได้ที่ใช้เป็นหลักฐานในการคำนวณภาษีคูณด้วยอัตราภาษีที่กำหนด ดังนั้น เงินได้ที่ต้องเสียภาษีเงินได้นิติบุคคลหรือฐานภาษีเงินได้นิติบุคคลนั้น โดยทั่วไปได้แก่กำไรสุทธิที่คำนวณตามเงื่อนไขที่กำหนด แต่เพื่อความเป็นธรรมและอุดช่องว่างในการจัดเก็บภาษีเงินได้ จึงได้มีการบัญญัติจัดเก็บภาษีเงินได้นิติบุคคล จากเงินได้หรือฐานภาษี ที่แตกต่างกัน ดังนี้

1. กำไรสุทธิ
2. ยอดขายรับก่อนหักรายจ่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เงินได้ที่จ่ายจากหรือในประเทศไทย

4. การจำหน่ายเงินกำไรออกไปจากประเทศไทยวิธีการเสียภาษีเงินได้นิติบุคคล

การเสียภาษีเงินได้นิติบุคคลของบริษัทหรือห้างหุ้นส่วนนิติบุคคลตามประมวลรัษฎากร มีดังนี้

1. การเสียภาษีโดยยื่นรายการประเมินตนเอง

โดยทั่วไปผู้มีหน้าที่เสียภาษีเงินได้นิติบุคคลจะต้องคำนวณภาษี และยื่นและแสดงรายการตามกำหนดเวลาที่กฎหมายบัญญัติไว้ ซึ่งแตกต่างกันไปตามประเภทของนิติบุคคลและวิธีการคำนวณเสียภาษี

2. การเสียภาษีโดยถูกหักไว้ ณ ที่จ่ายประมวลรัษฎากรได้บัญญัติให้ผู้จ่ายเงินได้ให้แก่บริษัทหรือห้างหุ้นส่วนนิติบุคคลในบางกรณีต้องหักภาษี ณ ที่จ่ายและนำส่งรัฐบาล โดยภาษีที่หักไว้ให้นั้นให้ถือเสมือนว่าภาษีในการคำนวณภาษีเงินได้นิติบุคคลเมื่อถึงกำหนดเวลาขึ้นแบบแสดงรายการตามปกติ

3. การเสียภาษีโดยการตรวจสอบของเจ้าพนักงานประเมิน

ในกรณีที่บริษัทหรือห้างหุ้นส่วนนิติบุคคลได้คำนวณและชำระภาษีเงินได้นิติบุคคลไว้ไม่ถูกต้องครบถ้วน เจ้าพนักงานประเมินมีอำนาจเรียกตรวจสอบและประเมินเงินภาษี รวมทั้งเบี้ยปรับและเงินเพิ่มให้บริษัทเสียภาษีเงินได้นิติบุคคลให้ถูกต้อง

การประเมินของเจ้าพนักงานประเมิน สามารถประเมินได้จากการตรวจสอบ และประเมินโดยไม่ได้ทำการตรวจสอบนอกจากนี้ในกรณีจำเป็นพนักงานอาจประเมินภาษีก่อนถึงกำหนดเวลาขึ้นแบบแสดงรายการก็ได้

รอบระยะเวลาบัญชี

ผู้มีหน้าที่เสียภาษีจากกำไรสุทธิ ต้องคำนวณกำไรสุทธิ จากรายได้จากกิจการ หรือเนื่องจากการที่กระทำในรอบระยะเวลาบัญชี หักด้วยรายจ่ายตามเงื่อนไขที่ระบุไว้ในมาตรา 65 ทวิ และมาตรา 65 ตรี ซึ่งประมวลรัษฎากร ได้กำหนดระยะเวลาบัญชีหนึ่งๆ ไว้ดังนี้

1. รอบระยะเวลาบัญชีโดยทั่วไป

ตามบทบัญญัติมาตรา 65 แห่งประมวลรัษฎากร รอบระยะเวลาบัญชีสำหรับการคำนวณภาษีเงินได้นิติบุคคล จะต้องเท่ากับ 12 เดือน โดยจะเริ่มต้นและสิ้นสุดลงเมื่อใดก็ได้

2. รอบระยะเวลาบัญชีซึ่งน้อยกว่า 12 เดือน

กรณีที่กฎหมายขอมให้รอบระยะเวลาบัญชีน้อยกว่า 12 เดือนได้ มีเฉพาะกรณีดังต่อไปนี้คือ

2.1 บริษัทหรือห้างหุ้นส่วนนิติบุคคลเริ่มตั้งใหม่ จะถือวันเริ่มตั้งถึงวันหนึ่งวันใดเป็นรอบระยะเวลาบัญชีแรกก็ได้ แต่รอบระยะเวลาบัญชีต่อไปต้องเท่ากับ 12 เดือน

2.2 บริษัทหรือห้างหุ้นส่วนนิติบุคคลอาจยื่นคำร้องขอเปลี่ยนวันสุดท้ายของรอบระยะเวลาบัญชีก็ได้ตามที่อธิบดีกรมสรรพากรจะเห็นสมควรและตั้งอนุญาตซึ่งรอบระยะเวลาบัญชีแรกที่ได้รับอนุญาตให้เปลี่ยนจะน้อยกว่า 12 เดือน

2.3 บริษัทหรือห้างหุ้นส่วนนิติบุคคลที่เลิกกัน ให้ถือเอาวันที่เจ้าพนักงานจดทะเบียนเลิกเป็นวันสุดท้ายของรอบระยะเวลาบัญชี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 บริษัทหรือห้างหุ้นส่วนนิติบุคคลควบเข้ากัน ให้ถือว่าบริษัทหรือห้างหุ้นส่วนนิติบุคคลนั้น เลิกกัน รอบระยะเวลาบัญชีที่ควบเข้ากันจึงเป็นไปตาม (ค) ซึ่งอาจน้อยกว่า 12 เดือน

ในกรณีที่บริษัทเลิกกิจการและยังชำระบัญชีไม่เสร็จ หากมีกำไรสุทธิเกิดขึ้น จะต้องนำมาเสีย ภาษีเงินได้นิติบุคคลเพราะประมวลกฎหมายแพ่งและพาณิชย์ถือว่าบริษัทยังมีสภาพเป็นนิติบุคคลอยู่ตราบ เท่าที่ยังชำระบัญชีไม่เสร็จสิ้น

3. รอบระยะเวลาบัญชีมากกว่า 12 เดือน

รอบระยะเวลาบัญชีอาจขยายออกไปมากกว่า 12 เดือนก็ได้ ในกรณีที่บริษัทหรือห้างหุ้นส่วนนิติ บุคคลเลิกกิจการ หากผู้ชำระบัญชี และผู้จัดการ ไม่สามารถยื่นรายการและเสียภาษีได้ภายใน 150 วันนับแต่ วันสุดท้ายของรอบระยะเวลาบัญชีแล้ว ถ้าได้ยื่นคำร้องต่ออธิบดีภายใน 30 วันนับแต่วันที่เข้าพนักงานรับ จดทะเบียนเลิก อธิบดีกรมสรรพากรอาจพิจารณาอนุมัติให้ขยายรอบระยะเวลาบัญชีออกไปได้ ซึ่งรอบ ระยะเวลาบัญชีรอบนี้อาจเกิน 12 เดือนก็ได้

กำไรสุทธิเพื่อเสียภาษีเงินได้นิติบุคคล

ตามประมวลรัษฎากร ได้กำหนด ให้บริษัทหรือห้างหุ้นส่วนนิติบุคคลเสียภาษีเงิน ได้นิติบุคคลจาก กำไรสุทธิโดยคำนวณจากรายได้ซึ่ง ได้จากกิจการหรือเนื่องจากการที่กระทำในรอบระยะเวลาบัญชีหัก ด้วยรายจ่ายตามเงื่อนไขที่ระบุไว้ในมาตรา 65 ทวิ และมาตรา 65 ตรี

การคำนวณรายได้และรายจ่ายดังกล่าว ให้ใช้เกณฑ์สิทธิ โดยให้นำรายได้ที่เกิดขึ้นในรอบระยะเวลา บัญชีใด แม้ว่าจะยัง ไม่ได้รับชำระในรอบระยะเวลาบัญชีนั้นมารวมคำนวณเป็นรายได้ในรอบระยะเวลา บัญชีนั้นและให้นำรายจ่ายทั้งสิ้นที่เกี่ยวกับราย ได้นั้นแม้จะยังมี ได้จ่ายในรอบระยะเวลาบัญชีนั้น มา รวมคำนวณเป็นรายจ่ายของรอบระยะเวลาบัญชีนั้น

ในกรณีจำเป็น ผู้มีเงินได้จะ ขออนุมัติต่ออธิบดี เพื่อเปลี่ยนแปลงเกณฑ์สิทธิ และวิธีการทางบัญชี เพื่อคำนวณรายได้และรายจ่ายตามเกณฑ์อื่นก็ได้และเมื่อ ได้รับอนุมัติจากอธิบดีแล้วให้ถือปฏิบัติตั้งแต่รอบ ระยะเวลาบัญชีที่อธิบดีกำหนดเป็นต้นไป

อัตราภาษี และการคำนวณภาษี

1. อัตราภาษี

1.1 ภาษีจากกำไรสุทธิของบริษัทหรือห้างหุ้นส่วนนิติบุคคลร้อยละ 30

1.2 ภาษีจากกำไรสุทธิเฉพาะกรณีที่ ได้จากการประกอบกิจการวิเทศนกิจตามประกาศกระทรวง การคลัง เรื่องการประกอบกิจการวิเทศนกิจของธนาคารพาณิชย์ ลงวันที่ 16 กันยายน 2535 ร้อยละ 10

2. การคำนวณภาษีเงินได้นิติบุคคลจากกำไรสุทธิ

บริษัทหรือห้างหุ้นส่วนนิติบุคคลที่มีหน้าที่เสียภาษีเงิน ได้นิติบุคคลจากกำไรสุทธิและต้อง คำนวณภาษีเงิน ได้ นิติบุคคล และยื่นแบบแสดงรายการและชำระภาษีปีละ 2 ครั้ง ดังนี้

2.1 การคำนวณเงิน ได้นิติบุคคลครั้งรอบระยะเวลาบัญชีนั้น ได้มีบัญญัติไว้ในมาตรา 67 ทวิ แห่ง เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประมวลรัษฎากรดังนี้

2.1.1 ในกรณีบริษัทหรือห้างหุ้นส่วนนิติบุคคล นอกจากที่กล่าวใน (2) ให้จัดทำประมาณการกำไรสุทธิ หรือขาดทุนสุทธิ ซึ่งได้จากกิจการหรือเนื่องจากกิจการที่ได้กระทำหรือจะได้กระทำในรอบระยะเวลาบัญชีนั้น แล้วให้คำนวณและชำระภาษีเงินได้นิติบุคคลจากจำนวนกึ่งหนึ่งของประมาณการกำไรสุทธิในรอบระยะเวลาบัญชีนั้น

2.1.2 ในกรณีบริษัทจดทะเบียนธนาคารพาณิชย์ตามกฎหมายว่าด้วยการธนาคารพาณิชย์ หรือ บริษัทเงินทุน บริษัทหลักทรัพย์ หรือ บริษัทเครดิตฟองซิเอร์ หรือ บริษัทหรือห้างหุ้นส่วนนิติบุคคล ตามหลักเกณฑ์ วิธีการ และเงื่อนไขที่อธิบดีกำหนดให้คำนวณและชำระภาษีจากกำไรสุทธิ ของรอบระยะเวลาหกเดือนนับแต่วันแรกของรอบระยะเวลาบัญชีตามเงื่อนไขที่ระบุไว้ในมาตรา 65 ทวิ และ 65 ครี

ภาษีเงินได้นิติบุคคลครั้งรอบระยะเวลาบัญชีนี้ให้ถือเป็นเครดิตในการเสียภาษีเงินได้นิติบุคคล เมื่อสิ้นรอบระยะเวลาบัญชีคือ เอาไปหักออกจากภาษีที่ต้องเสียจากกำไรสุทธิของทั้งรอบระยะเวลาบัญชี และในกรณีที่ภาษีที่เสียไว้ครั้งรอบระยะเวลาบัญชีสูงกว่าภาษีที่จะต้องเสียทั้งรอบระยะเวลาบัญชี บริษัทหรือห้างหุ้นส่วนนิติบุคคลก็มีสิทธิขอคืนภาษีที่ชำระไว้เกินได้

กรณีที่บริษัทหรือห้างหุ้นส่วนนิติบุคคลมีรอบระยะเวลาบัญชีแรกหรือรอบระยะเวลาบัญชีสุดท้าย น้อยกว่า 12 เดือน ไม่ต้องยื่นแบบแสดงรายการและเสียภาษีเงินได้นิติบุคคลครั้งรอบระยะเวลาบัญชี

2.2 การคำนวณเงินได้นิติบุคคลจากกำไรสุทธิ เมื่อสิ้นรอบระยะเวลาบัญชี

การคำนวณกำไรสุทธิของบริษัทหรือห้างหุ้นส่วนนิติบุคคลให้คำนวณกำไรสุทธิตามเงื่อนไขที่บัญญัติไว้ในประมวลรัษฎากร โดยนำกำไรสุทธิตั้งกล่าวคูณด้วยอัตราภาษีเงินได้นิติบุคคล จะได้ภาษีเงินได้นิติบุคคลที่ต้องชำระ ถ้าคำนวณกำไรสุทธิออกมาแล้วปรากฏว่า ไม่มีกำไรสุทธิ หรือขาดทุนสุทธิ บริษัทไม่ต้องเสียภาษีเงินได้นิติบุคคล

ถ้าการจัดทำบัญชีของบริษัทได้จัดทำขึ้นตามหลักบัญชีโดยไม่ได้ปฏิบัติตามเงื่อนไขในประมวลรัษฎากรเมื่อจะคำนวณภาษีบริษัทจะต้องปรับปรุงกำไรสุทธิตั้งกล่าวให้เป็นไปตามเงื่อนไขที่บัญญัติไว้ในประมวลรัษฎากรแล้วจึงคำนวณภาษีเงินได้นิติบุคคล

การยื่นแบบแสดงรายการและชำระภาษี

บริษัทหรือห้างหุ้นส่วนนิติบุคคลที่เสียภาษีเงินได้จากกำไรสุทธิจะต้องยื่นแบบแสดงรายการและชำระภาษีดังนี้

1. การเสียภาษีเงินได้นิติบุคคลครั้งรอบจะต้องยื่นแบบแสดงรายการพร้อมชำระภาษี(ถ้ามี) ตามแบบ ภ.ง.ด.51 ภายใน 2 เดือนนับจากวันสุดท้ายของทุกเดือนแรกของรอบระยะเวลาบัญชี

2. การเสียภาษีเงินได้จากกำไรสุทธิเมื่อสิ้นรอบระยะเวลาบัญชีจะต้องยื่นแบบแสดงรายการพร้อมชำระภาษี(ถ้ามี) ตามแบบ ภ.ง.ด.50 ภายใน 150 วันนับแต่วันสุดท้ายของรอบระยะเวลาบัญชี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถานที่ยื่นแบบแสดงรายการภาษี

ในเขตกรุงเทพมหานคร ให้ยื่น ณ

1. สำนักงานเขตท้องที่ที่สำนักงานใหญ่ตั้งอยู่

2. ธนาคารพาณิชย์ไทย และ สาขา ในเขตกรุงเทพมหานคร ในเขตจังหวัดอื่น ให้ยื่น ณ

2.1 ที่ว่าการอำเภอหรือกิ่งอำเภอท้องที่ที่สำนักงานใหญ่ตั้งอยู่ ในกรณีสำนักงาน
สรรพากรอำเภอมิได้ตั้งอยู่ ณ ที่ว่าการอำเภอให้ยื่น ณ สำนักงานสรรพากรอำเภอ หรือ

2.2 สำนักงานสาขาของธนาคารพาณิชย์ไทยในเขตอำเภอหรือกิ่งอำเภอท้องที่ที่สำนัก
งานใหญ่ตั้งอยู่



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

ผู้มีหน้าที่จดทะเบียนภาษีมูลค่าเพิ่ม

ผู้ประกอบการที่ขายสินค้าหรือให้บริการในทางธุรกิจหรือวิชาชีพเป็นปกติธุระ ไม่ว่าจะการขายสินค้าหรือให้บริการดังกล่าวนั้น จะได้รับประโยชน์หรือผลตอบแทนหรือไม่ และไม่ว่าจะประกอบกิจการในรูปของบุคคลธรรมดา คณะบุคคล นิติบุคคล ส่วนราชการหรือองค์การของรัฐบาล ถ้าประกอบกิจการมีรายรับ เกินกว่า 600,000 บาทต่อปี ผู้ประกอบการนั้นมีหน้าที่ต้องจดทะเบียนภาษีมูลค่าเพิ่ม แต่หากการประกอบกิจการนั้นเป็นกิจการที่ได้รับยกเว้นภาษีมูลค่าเพิ่ม ผู้ประกอบการ ไม่มีหน้าที่ต้องจดทะเบียนภาษีมูลค่าเพิ่ม

ผู้ที่ได้รับการยกเว้นไม่ต้องจดทะเบียนภาษีมูลค่าเพิ่ม

ผู้ประกอบการดังต่อไปนี้ ได้รับยกเว้น ไม่ต้องจดทะเบียนภาษีมูลค่าเพิ่ม

1. ผู้ประกอบการที่อยู่นอกราชอาณาจักร และเข้ามาประกอบกิจการขายสินค้า หรือให้บริการในราชอาณาจักรเป็นครั้งคราว
2. ผู้ประกอบการที่ให้บริการจากต่างประเทศ และได้มีการใช้ บริการนั้น ในราชอาณาจักร

ผู้ประกอบการที่ได้รับการยกเว้นภาษีมูลค่าเพิ่มแต่สามารถขอจดทะเบียนภาษีมูลค่าเพิ่มได้

ตามที่ได้กล่าวข้างต้นแล้วว่า ผู้ประกอบการที่ประกอบกิจการที่ได้รับยกเว้น ภาษีมูลค่าเพิ่ม เช่น ผู้ประกอบกิจการที่มีรายรับไม่เกิน 600,000.- บาทต่อปี ผู้ประกอบการที่ขายพืชผลทางการเกษตรที่ยังไม่มีการแปรรูป หรือการสีข้าว เป็นต้น ไม่มีหน้าที่ต้องจดทะเบียนภาษีมูลค่าเพิ่ม แต่ผู้ประกอบการที่ประกอบกิจการ ได้รับยกเว้นภาษีมูลค่าเพิ่มบางประเภท สามารถขอจดทะเบียนภาษีมูลค่าเพิ่มได้ การประกอบกิจการที่ได้รับยกเว้นภาษีมูลค่าเพิ่ม แต่สามารถขอจดทะเบียนภาษีมูลค่าเพิ่มได้ ได้แก่

1. ผู้ประกอบการ ขายพืชผลทางการเกษตร สัตว์ ไม่ว่าจะมีชีวิตหรือไม่มีชีวิต นุ้ช ปลาแปน อาหารสัตว์ ยาหรือเคมีภัณฑ์ที่ใช้สำหรับพืชหรือสัตว์ หนังสือพิมพ์ นิตยสาร หรือตำราเรียน
2. ผู้ประกอบกิจการขายสินค้าหรือให้บริการ ซึ่งไม่ได้รับยกเว้นภาษี และมีรายรับไม่เกิน 600,000.- บาทต่อปี
3. การให้บริการขนส่ง ในราชอาณาจักร โดยอากาศยาน
4. การส่งออกของผู้ประกอบการในเขตอุตสาหกรรมส่งออกตามกฎหมายว่าด้วย

นิคมอุตสาหกรรมแห่งประเทศไทย

กำหนดเวลาจดทะเบียน

1. กรณีผู้ประกอบการ จะเริ่มประกอบกิจการ กำหนดเวลาจดทะเบียนภายในกำหนด 6 เดือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก่อนวันเริ่มประกอบกิจการ แต่มีสิทธิยื่นคำขอจดทะเบียนภาษีมูลค่าเพิ่มก่อน วันเริ่มประกอบกิจการ ได้ เมื่อมีแผนงานที่สามารถพิสูจน์ได้ว่า ได้เตรียมการเพื่อประกอบกิจการขายสินค้าหรือให้บริการที่ต้องเสียภาษีมูลค่าเพิ่ม และมีการดำเนินการ เพื่อเตรียมประกอบกิจการ อันเป็นเหตุให้ต้องมีการซื้อสินค้า หรือรับบริการที่ต้องเสียภาษีมูลค่าเพิ่ม เช่น ก่อสร้าง โรงงาน ซื้อเครื่องจักร โดยสิทธิยื่นคำขอจดทะเบียน ได้ภายใน เวลาที่เหมาะสมและจำเป็นต่อการก่อสร้าง โรงงานหรืออาคารสำนักงาน หรือติดตั้งเครื่องจักร

2. กรณีประกอบการอยู่แล้วและมีรายรับเกิน 600,000 บาทต่อปี กำหนดเวลาจดทะเบียน ภายใน 30 วันนับแต่วันที่มีรายรับเกิน 600,000 บาทต่อปี

3. กรณีผู้ประกอบการที่ได้รับการยกเว้น แต่มีสิทธิของจดทะเบียนภาษีมูลค่าเพิ่ม และขอใช้สิทธิจดทะเบียนฯ ไม่มีกำหนดเวลาหากแต่จะต้องแจ้งต่ออธิบดีกรมสรรพากรเพื่อขอจดทะเบียนภาษีมูลค่าเพิ่ม และยื่นคำขอจดทะเบียนภาษีมูลค่าเพิ่มภายใน 30 วัน นับแต่วันแจ้งต่ออธิบดีกรมสรรพากร

สถานที่จดทะเบียนภาษีมูลค่าเพิ่ม

การจดทะเบียนภาษีมูลค่าเพิ่มของผู้ประกอบการ ให้ยื่นคำขอจดทะเบียนภาษีมูลค่าเพิ่มตามแบบ ภ.พ.01 ณ สถานที่ดังต่อไปนี้

1. กรณีสถานประกอบการตั้งอยู่ในเขตกรุงเทพมหานคร ให้ยื่นคำขอจดทะเบียนที่

1.1 สำนักงานสรรพากรเขต สำนักงานสรรพากรเขต (สาขา) หรือ สำนักงานสรรพากรที่สถานประกอบการตั้งอยู่

1.2 สำนักงานภาษีสรรพากรพื้นที่ที่สถานประกอบการตั้งอยู่

2. กรณีสถานประกอบการตั้งอยู่ในเขตจังหวัดอื่น ให้ยื่นคำขอจดทะเบียนที่สำนักงานสรรพากรอำเภอ สำนักงานอำเภอ (สาขา) หรือ สำนักงานสรรพากรกิ่งอำเภอ ที่สถานประกอบการตั้งอยู่

ใบทะเบียนภาษีมูลค่าเพิ่ม

เมื่อเจ้าพนักงานได้รับคำขอจดทะเบียนภาษีมูลค่าเพิ่ม ตามแบบ ภ.พ.01 พร้อมเอกสารที่เกี่ยวข้องครบถ้วนแล้ว จะมีการออกใบทะเบียนภาษี มูลค่าเพิ่ม (แบบ ภ.พ.20 หรือ แบบ ภ.พ.21) ให้ ซึ่งจะมีผลให้ผู้ประกอบการเป็น ผู้ประกอบการจดทะเบียน ตามกฎหมายตั้งแต่วันที่ระบุไว้ในทะเบียนภาษีมูลค่าเพิ่มเป็นต้นไป

1. แบบ ภ.พ.20

ผู้ประกอบการจดทะเบียน ที่มีหน้าที่เสียภาษีมูลค่าเพิ่ม โดยคำนวณภาษีที่ต้องเสียจากภาษีขายหักด้วยภาษีซื้อ จะได้รับใบทะเบียนภาษีมูลค่าเพิ่มตามแบบ ภ.พ.20 ฉบับลายน้ำ

2. แบบ ภ.พ.21

ผู้ประกอบการจดทะเบียนที่คำนวณภาษีที่ต้องเสียจากรายรับในแต่ละเดือน คูณด้วยอัตราภาษีร้อยละ 1.5 จะได้รับใบทะเบียนภาษีมูลค่าเพิ่มตามแบบ ภ.พ.21 ซึ่งมีพื้นสีเขียว ในกรณีการออกใบทะเบียนภาษีมูลค่าเพิ่มให้แก่ผู้ประกอบการจดทะเบียนแต่ละประเภท เจ้าพนักงานจะออกให้ตามจำนวนสถาน

ประกอบการที่มีอยู่จริงตามที่ได้แจ้งไว้ในแบบ ก.พ.01 ผู้ประกอบการจดทะเบียนจะต้องนำใบทะเบียนดังกล่าว ไปแสดงไว้ ณ สถานประกอบการแต่ละแห่งในสถานที่ที่เห็นได้ง่ายและเปิดเผย หากใบทะเบียนชำรุด สูญหาย ต้องรีบขอใบแทนใหม่ก่อนายทะเบียนภายใน 15 วัน นับแต่วันที่ชำรุดหรือสูญหายโดยใช้แบบ ก.พ.04 เป็นแบบขอรับใบแทนใบทะเบียนภาษีมูลค่าเพิ่ม ในกรณีสูญหายต้องแนบสำเนาบันทึกประจำวัน (ใบแจ้งความ) ของเจ้าพนักงานไปด้วย

หน้าที่ของผู้ประกอบการจดทะเบียนภาษีมูลค่าเพิ่ม

1. ผู้ประกอบการที่จดทะเบียน ประเภท คำนำณภาษีจากภาษีขาย หักด้วยภาษีซื้อ มีหน้าที่

1.1 เรียกเก็บภาษีมูลค่าเพิ่มจากผู้ซื้อสินค้าหรือผู้รับบริการ และออกไปกำกับภาษีเพื่อเป็นหลักฐาน ในการเรียกเก็บภาษีมูลค่าเพิ่ม

1.2 จัดทำรายงานตามที่กฎหมายกำหนดซึ่งได้แก่

1.2.1 รายงานภาษีซื้อ

1.2.2 รายงานภาษีขาย

1.2.3 รายงานสินค้าและวัตถุดิบหรือ รายละเอียดสินค้าคงเหลือ ณ วันที่ 31

ธันวาคม พ.ศ

1.3 ขึ้นแบบแสดงรายการเพื่อเสียภาษี

2. ผู้ประกอบการที่จดทะเบียน ประเภท คำนำณภาษีจากรายรับคูณด้วยอัตราภาษีร้อยละ 1.5 มีหน้าที่

2.1 จัดทำรายงานตามที่กฎหมายกำหนด ซึ่งได้แก่

2.2.1 รายงานมูลค่าของฐานภาษี

2.2.2 รายงานสินค้าและวัตถุดิบหรือ รายละเอียดสินค้าคงเหลือ ณ วันที่ 31

ธันวาคม พ.ศ

2.2 ขึ้นแบบแสดงรายการเพื่อเสียภาษี

การเปลี่ยนแปลงการจดทะเบียนภาษีมูลค่าเพิ่ม

เมื่อมีการเปลี่ยนแปลงเกี่ยวกับการประกอบกิจการ เช่น มีการ โอนกิจการรับ โอนกิจการ ย้ายสถานประกอบการ เปลี่ยนแปลงรายการที่จดทะเบียนภาษีมูลค่าเพิ่ม เปิดสถานประกอบการเพิ่มเติม ปิดสถานประกอบการบางแห่ง หยุดประกอบกิจการชั่วคราว ควบกิจการหรือเลิกประกอบกิจการ ผู้ประกอบการต้องแจ้งการเปลี่ยนแปลงรายการต่าง ๆ โดยกรอกแบบแจ้งการเปลี่ยนแปลงทะเบียนภาษีมูลค่าเพิ่ม (ก.พ.09) จำนวน 3 ฉบับ พร้อมเอกสารประกอบการพิจารณา ขึ้น ณ สถานที่ที่ได้จดทะเบียนภาษีมูลค่าเพิ่มไว้ เอกสารประกอบการพิจารณาที่ผู้ประกอบการต้อง แนบไปพร้อมกับ แบบ ก.พ.09 ได้แก่ ใบทะเบียนภาษีมูลค่าเพิ่มที่แจ้งการเปลี่ยนแปลง สำเนาทะเบียนบ้าน หนังสือแสดงการเปลี่ยนชื่อ-นามสกุล หนังสือรับรองของ นายทะเบียนหุ้นส่วนบริษัท บัญชีและเอกสารประกอบการลงบัญชีหรือ หนังสือมอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อำนาจ (กรณีมิได้มาดำเนินการเอง) พร้อมบัตรประจำตัวประชาชนของผู้มอบและผู้รับมอบอำนาจ และ
อื่นๆ แล้วแต่กรณี เช่น โบบมระบัตร สัญญาเช่า หนังสือยินยอมให้ใช้สถานประกอบการ เป็นต้น

กิจการที่ได้รับการยกเว้นภาษีมูลค่าเพิ่ม

การประกอบกิจการต่าง ๆ ดังต่อไปนี้ ให้ได้รับยกเว้นไม่ต้องเสียภาษีมูลค่าเพิ่ม

การขายสินค้า

1. การขายสินค้าหรือ ให้บริการของผู้ประกอบการที่มีรายรับไม่เกิน 600,000 บาทต่อปี
2. การขายพืชผลทางการเกษตร ไม่ว่าจะเป็นส่วนใด ๆ ของพืช รวมถึงวัตถุดิบที่ได้จากพืช ที่ยังมีสภาพเดิมและไม่ทำเป็นอุตสาหกรรม แต่ไม่รวมถึง ไม้ซุง ฝืน หรือผลิตภัณฑ์ที่ได้จากการเลื่อยไม้
3. การขายสัตว์ไม่ว่าจะมีชีวิตหรือ ไม่มีชีวิต ไม่ว่าจะเป็นเนื้อ ส่วนต่างๆ ของสัตว์ ไข่ นม และ วัตถุดิบที่ได้จากสัตว์ ที่อยู่ในสภาพสด หรือรักษาสภาพไว้เพื่อมิให้เสียเป็นการชั่วคราว แต่ไม่รวมถึงผลิตภัณฑ์อาหารบรรจุกระป๋อง กากขนหรือหีบห่อที่ทำเป็นอุตสาหกรรม
4. การขายปุ๋ย
5. การขายปลาป่น อาหารสัตว์
6. การขายชาหรือเคมีภัณฑ์สำหรับพืชหรือสัตว์
7. การขายหนังสือพิมพ์ นิตยสาร หรือตำราเรียน
8. การขายสินค้าของกระทรวง ทบวง กรม ซึ่งส่งรายรับทั้งสิ้นให้แก่รัฐโดยไม่หักรายจ่าย
9. การขายสินค้าเพื่อประโยชน์แก่การศาสนาหรือการสาธารณสุขในประเทศไทย ซึ่งไม่นำผล

กำไรไปจ่ายในทางอื่น

10. การขายหรือเฉพาะที่ผลิต โดยองค์การของรัฐบาล และผู้ขายที่มีชื่อผู้ประกอบการอุตสาหกรรมยาสูบที่ผลิตสินค้าดังกล่าว

11. การขายสลากกินแบ่งของรัฐบาล สลากออมสินของรัฐบาลและสลากบำรุงกาชาดไทย
12. การขายแสดมปีปรมณีย์ แสดมปีอากรหรือแสดมปีอื่นของรัฐบาลเฉพาะที่ยัง ไม่ได้ใช้ในราคาไม่เกินมูลค่าที่ตราไว้

13. การบริจาคสินค้าให้แก่

13.1 สถานพยาบาล และสถานศึกษาของทางราชการ

13.2 องค์การหรือสถานสาธารณกุศล หรือแก่สถานพยาบาลและสถานศึกษาอื่นนอก
จากใน (ก) ตามที่รัฐมนตรีประกาศกำหนดในราชกิจจานุเบกษา

การให้บริการ

1. การให้บริการการศึกษาของสถานศึกษาของทางราชการและเอกชน
2. การให้บริการที่เป็นงานศิลปะและวัฒนธรรมในสาขานาฏศิลป์และสาขาดุริยางค์ศิลป์และคีต

ศิลป์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การให้บริการการประกอบโรคศิลป์ การสอบบัญชี การว่าความ หรือการประกอบวิชาชีพ อื่นๆตามที่อธิบดีกำหนด โดยอนุมติรัฐมนตรี ทั้งนี้เฉพาะวิชาชีพอิสระที่มีกฎหมายควบคุมการ ประกอบวิชาชีพอิสระนั้น

4. การให้บริการรักษาพยาบาลของสถานพยาบาลตามกฎหมายว่าด้วยสถานพยาบาล

5. การให้บริการวิจัย หรือการให้บริการทางวิชาการในสาขาวิทยาศาสตร์และสังคมศาสตร์ที่มีใช้ ทางธุรกิจ

6. การให้บริการห้องสมุด พิพิธภัณฑ์ สวนสัตว์

7. การให้บริการตามสัญญาจ้างแรงงาน

8. การให้บริการการแข่งขันกีฬาสมัครเล่น

9. การให้บริการของนักแสดงสาธารณะ

10. การให้บริการขนส่งในราชอาณาจักร

11. การให้บริการขนส่งระหว่างประเทศเฉพาะทางบก

12. การให้บริการเช่าอสังหาริมทรัพย์

13. การให้บริการของราชการส่วนท้องถิ่น ที่มีใช้เป็นการพาณิชย์หรือหารายได้

14. การให้บริการของกระทรวง ทบวง กรม ซึ่งส่งรายรับทั้งสิ้นให้แก่รัฐโดยไม่มีกำไรจ่าย

15. การให้บริการเพื่อประโยชน์แก่การศาสนาหรือการสาธารณกุศลในประเทศไทย ซึ่งไม่นำผล กำไร ไปจ่ายในทางอื่น

16. การให้บริการสีข้าว

17. การขนส่งระหว่างประเทศโดยผู้ประกอบการที่เป็นนิติบุคคลที่ตั้งขึ้นตามกฎหมายต่างประเทศ ซึ่งตามกฎหมายของประเทศนั้นยกเว้นภาษีทางอ้อมให้แก่ผู้ประกอบการที่เป็นนิติบุคคลที่ตั้งขึ้น ตามกฎหมายไทย ตามหลักกติกายกเว้นภาษีปฏิบัติ

18. การให้บริการสื่อสารทางวิทยุเกี่ยวกับการบินระหว่างประเทศ ทั้งนี้ เฉพาะส่วนบริษัทหรือ ห้างหุ้นส่วนนิติบุคคลซึ่งได้รับมอบหมายจากรัฐบาลให้จัดบริการดังกล่าว

19. การให้บริการการศึกษาของสถานศึกษาที่รัฐมนตรีกำหนด

การนำเข้า

1. การนำเข้าพืชผลทางการเกษตร รวมถึงวัตถุดิบได้จากพืช การนำเข้าสัตว์ไม่ว่าจะมีชีวิตหรือไม่ ซึ่งรวมถึงวัตถุดิบได้จากสัตว์ด้วยเช่นกัน การนำเข้าปุ๋ย ปลาป่น อาหารสัตว์ ยา หรือเคมีภัณฑ์ สำหรับพืชหรือสัตว์ หนังสือพิมพ์ นิตยสารหรือตำราเรียน

2. การนำเข้าสินค้าจากต่างประเทศที่นำเข้าไปในเขตอุตสาหกรรมส่งออก เฉพาะสินค้าที่ได้รับการ ยกเว้นอากรขาเข้าตามกฎหมายว่าด้วยการนิคมอุตสาหกรรมแห่งประเทศไทย

3. การนำเข้าสินค้าที่จำแนกประเภทไว้ในภาคว่าด้วยการส่งออกที่ได้รับยกเว้นอากรตามกฎหมาย ว่าด้วยพิกัดอัตราศุลกากร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. สินค้าซึ่งนำเข้ามาและอยู่ในอารักขาของศุลกากรแล้วได้ส่งกลับออกไปต่างประเทศ โดยได้คืน อารักขาเข้าตามกฎหมายว่าด้วยศุลกากร

การส่งออก

การส่งออกสินค้าหรือบริการของผู้ประกอบการจดทะเบียนซึ่งมีมูลค่าของฐานภาษีตั้งแต่ 600,000 บาท แต่ไม่เกิน 1,200,000 บาท และจดทะเบียนภาษีมูลค่าเพิ่มเสียภาษีจากขอตรารับก่อนหัก รายจ่าย

การยื่นแบบแสดงรายการภาษีมูลค่าเพิ่ม

1. ผู้มีหน้าที่ยื่นแบบ

1.1 ผู้มีหน้าที่ยื่นแบบแสดงรายการภาษีมูลค่าเพิ่ม

1.1.1 ผู้ประกอบการจดทะเบียนที่เสียภาษีมูลค่าเพิ่มในอัตราร้อยละ 10 หรือผู้ประกอบการจดทะเบียนภาษีมูลค่าเพิ่มในอัตราร้อยละ 0 โดยคำนวณภาษีมูลค่าเพิ่มจากภาษีขายหักด้วย ภาษีซื้อในแต่ละเดือนภาษี

1.1.2 ผู้ประกอบการจดทะเบียนที่เสียภาษีมูลค่าเพิ่มในอัตราร้อยละ 1.5 โดยคำนวณภาษีมูลค่าเพิ่มจากขอขายสินค้าหรือบริการก่อนหักรายจ่ายใด ๆ ในแต่ละเดือนภาษี ทั้งนี้ ไม่ว่าผู้ประกอบการดังกล่าวจะประกอบการ ในรูปของบุคคลธรรมดา คณะบุคคล ห้างหุ้นส่วนสามัญ กองมรดก บริษัทหรือห้างหุ้นส่วนนิติบุคคล องค์การของรัฐบาล หรือนิติบุคคลในรูปแบบใดก็ตาม

1.2 ผู้มีหน้าที่ยื่นแบบนำส่งภาษีมูลค่าเพิ่ม

1.2.1 ผู้จ่ายเงินเมื่อจ่ายเงินค่าซื้อสินค้าหรือค่าบริการ ให้แก่

1.2.1.1 ผู้ประกอบการที่อยู่นอกราชอาณาจักรและได้เข้ามาประกอบกิจการขายสินค้าหรือให้บริการในราชอาณาจักร เป็นการชั่วคราว และไม่ได้จดทะเบียนภาษีมูลค่าเพิ่มเป็นการชั่วคราว

1.2.1.2 ผู้ประกอบการที่ได้ให้บริการในต่างประเทศและได้มีการใช้บริการนั้นในราชอาณาจักร หรือ

1.2.1.3 ผู้ประกอบการอื่นตามที่กำหนดโดยพระราชกฤษฎีกา

1.2.2 ผู้รับโอนสินค้าหรือผู้รับโอนสิทธิในบริการ ที่ได้เสียภาษีมูลค่าเพิ่มในอัตราร้อยละ 0 ได้แก่ การรับโอนสินค้าหรือรับโอนสิทธิในบริการที่ได้มีการขายหรือการให้บริการกับการสหประชาชาติ ทบวงการชำนัญพิเศษของสหประชาชาติ สถานเอกอัครราชทูต สถานทูต สถานกงสุลใหญ่ สถานกงสุล ทั้งนี้ เฉพาะการขายสินค้าหรือการให้บริการดังกล่าวที่เป็นไปตามหลักเกณฑ์วิธีการและเงื่อนไขที่อธิบดีกำหนด

1.2.3 ผู้ทอดตลาด ซึ่งขายทอดตลาดทรัพย์สินของผู้ประกอบการจดทะเบียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. แบบแสดงรายการที่ใช้

2.1 แบบ ภ.พ.30

ใช้สำหรับผู้ประกอบการจดทะเบียนที่เสียภาษีมูลค่าเพิ่มในอัตราร้อยละ 10 หรือผู้ประกอบการจดทะเบียนที่เสียภาษีมูลค่าเพิ่มในอัตราร้อยละ 0 โดยคำนวณภาษีมูลค่าเพิ่มจากภาษีขายหักด้วยภาษีซื้อในแต่ละเดือนภาษี กรณีเดือนภาษีใดมีจำนวนภาษีซื้อมากกว่าภาษีขาย ผู้ประกอบการจะได้รับคืนภาษี และสามารถ ใช้แบบ ภ.พ.30 นี้เป็น ค่าขอคืนภาษีมูลค่าเพิ่มได้ในตัว

2.2 แบบ ภ.พ.31

ใช้สำหรับผู้ประกอบการจดทะเบียนที่เสียภาษีมูลค่าเพิ่มในอัตราร้อยละ 1.5 โดยคำนวณภาษีมูลค่าเพิ่มจากยอดขายสินค้าหรือบริการก่อนหักรายจ่ายใด ๆ ในแต่ละเดือนภาษี

2.3 แบบใบขนสินค้าขาเข้า

ใช้สำหรับผู้มีหน้าที่เสียภาษีมูลค่าเพิ่มที่เป็นผู้นำเข้า และหากผู้ประกอบการจดทะเบียนเป็นผู้นำเข้าสินค้าก็ต้องยื่นแบบ ใบขนสินค้าขาเข้า และชำระภาษี เช่นเดียวกับผู้นำเข้าอื่น ๆ อีกทั้งต้องยื่นแบบแสดงรายการภาษีของแต่ละเดือนภาษีด้วยแบบ ภ.พ.30 ตามปกติ

2.4 แบบ ภ.พ.36

ใช้สำหรับผู้มีหน้าที่นำส่ง ภาษีมูลค่าเพิ่ม

กำหนดเวลา สถานที่ยื่นแบบและการชำระภาษี

1 กำหนดเวลายื่นแบบ

ผู้ประกอบการจดทะเบียนต้องยื่นแบบ ภ.พ.30 หรือ ภ.พ.31 พร้อมชำระภาษีมูลค่าเพิ่ม(ถ้ามี) เป็นรายเดือนทุกเดือนภาษี ไม่ว่าจะมีการขายสินค้าหรือให้บริการ ในเดือนภาษีนั้นหรือ ไม่ก็ตาม โดยยื่นแบบภายในวันที่ 15 ของเดือนถัดไป

ในกรณีผู้ประกอบการมีสถานประกอบการหลายแห่งให้แยกยื่นแบบเป็นรายสถานประกอบการ เว้นแต่จะได้รับอนุมัติจากอธิบดีกรมสรรพากร ให้ยื่นแบบรวมกัน ณ สถานประกอบการแห่งใดแห่งหนึ่ง ก็ให้ยื่นแบบ ณ สถานประกอบการที่ได้รับอนุมัตินั้น

ผู้นำส่งภาษีมูลค่าเพิ่ม จะต้องยื่นแบบ ภ.พ.36 พร้อมนำส่งภาษีมูลค่าเพิ่มภายในเวลาที่กฎหมายกำหนด ภายใน 7 วันนับแต่วันสิ้นเดือนของเดือนที่จ่ายเงินให้ผู้ประกอบการ

กรณีจ่ายเงินค่าซื้อสินค้าหรือบริการ ให้แก่ผู้ประกอบการที่เข้ามาประกอบกิจการ ในราชอาณาจักร เป็นการชั่วคราวหรือ ให้แก่ผู้ประกอบการที่ได้ให้บริการ ในต่างประเทศ หรือ ให้แก่ผู้ประกอบการอื่นตามที่กำหนดโดยพระราชกฤษฎีกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีขายทอดตลาดทรัพย์สินของผู้ประกอบการจดทะเบียนหรือส่วนราชการขายทรัพย์สินของผู้ประกอบการจดทะเบียนที่ถูกยึดมาตามกฎหมาย โดยวิธีอื่นนอกจากการขายทอดตลาดภายใน 30 วัน นับแต่วันที่ได้รับโอน

กรณีรับโอนสินค้าหรือโอนสิทธิในบริการที่ได้เสียภาษีในอัตราร้อยละ 0

2. สถานที่ยื่นแบบ

2.1 ในกรุงเทพมหานคร ให้ยื่นแบบ ณ

2.1.1 สำนักงานสรรพากรเขตในท้องที่ที่สถานประกอบการตั้งอยู่

2.1.2 สถานที่อื่นซึ่งอธิบดีกรมสรรพากรกำหนดให้เป็นสถานที่ยื่นแบบ และชำระภาษี (ปัจจุบันยังไม่มีกำหนด)

2.1.3 กองรายได้ กรมสรรพสามิต สำหรับผู้ประกอบการที่ขายสินค้าหรือให้บริการที่ต้องเสียทั้งภาษีสรรพสามิตและ ภาษีมูลค่าเพิ่ม

2.2 ในจังหวัดอื่น ให้ยื่นแบบ ณ

2.2.1 สำนักงานสรรพากรอำเภอหรือกิ่งอำเภอ ในท้องที่ที่สถานประกอบการตั้งอยู่

2.2.2 สถานที่อื่นซึ่งอธิบดีกรมสรรพากรกำหนดให้เป็นสถานที่ยื่นแบบและชำระภาษี (ปัจจุบันยังไม่มีกำหนด)

2.2.3 สำนักงานสรรพสามิตอำเภอหรือกิ่งอำเภอ หรือสำนักงานสรรพสามิตจังหวัดท้องที่ที่โรงงานอุตสาหกรรมหรือสถานบริการตั้งอยู่ สำหรับผู้ประกอบการที่ขายสินค้าหรือให้บริการที่ต้องเสียทั้งภาษีสรรพสามิตและภาษีมูลค่าเพิ่ม

3 การชำระภาษี

3.1 ชำระเป็นเงินสด หรือ

3.2 ชำระด้วยเช็คขีดคร่อม ตั้งจ่ายแก่ กรมสรรพากร โดยขีดจ่าคำว่า ผู้ถือ และหรือ ตามคำสั่ง

ใบกำกับภาษี

เอกสารหลักฐานสำคัญ ซึ่งผู้ประกอบการจดทะเบียนในระบบภาษีมูลค่าเพิ่มจะต้องจัดทำ และออกให้กับผู้ซื้อสินค้าหรือผู้รับบริการทุกครั้งที่มีการขายสินค้าหรือให้บริการ เพื่อแสดงมูลค่าของสินค้าหรือบริการและจำนวนภาษีมูลค่าเพิ่ม ที่ผู้ประกอบการจดทะเบียนเรียกเก็บ หรือพึงเรียกเก็บจากผู้ซื้อสินค้าหรือรับบริการในแต่ละครั้ง

กรณีขายสินค้า ผู้ประกอบการจดทะเบียนมีหน้าที่ต้องออกใบกำกับภาษีพร้อมทั้งส่งมอบใบกำกับภาษีให้กับผู้ซื้อในทันทีที่มีการส่งมอบสินค้าให้กับผู้ซื้อ

กรณีให้บริการ ผู้ประกอบการจดทะเบียนมีหน้าที่ต้องออกใบกำกับภาษี เมื่อได้รับชำระราคาค่าบริการ ผู้ประกอบการจดทะเบียนต้องจัดทำใบกำกับภาษีอย่างน้อย 2 ฉบับ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ต้นฉบับ ผู้ประกอบการต้องส่งมอบให้กับ ผู้ซื้อ

2. สำเนา ผู้ประกอบการต้องเก็บรักษาไว้เพื่อเป็นหลักฐานประกอบการลงรายงานภาษีขาย สำหรับผู้ประกอบการจดทะเบียนที่เป็นผู้ซื้อสินค้าหรือรับบริการ ต้องเรียกใบกำกับภาษีจากผู้ประกอบการจดทะเบียนอื่นซึ่งเป็นผู้ขายสินค้าหรือให้บริการเพื่อใช้เป็นหลักฐาน ในการคำนวณภาษีมูลค่าเพิ่มที่ต้องชำระ สำหรับผู้ประกอบการจดทะเบียนที่เป็นผู้นำเข้าจะได้รับใบเสร็จ รับเงินของกรมศุลกากรเป็นหลักฐานว่าได้ชำระภาษีมูลค่าเพิ่มและถือเป็นใบกำกับภาษีซึ่งใช้เป็นหลักฐาน ในการคำนวณภาษีมูลค่าเพิ่มได้เช่นเดียวกัน

ผู้ออกใบกำกับภาษี

ผู้มีหน้าที่ออกใบกำกับภาษี ได้แก่

1. ผู้ประกอบการจดทะเบียนที่เสียภาษีมูลค่าเพิ่มในอัตราร้อยละ 0 หรือร้อยละ 10 จากภาษีขาย หักด้วยภาษีซื้อ

2. ผู้ทอดตลาดที่มีชื่อส่วนราชการ ซึ่งขายทรัพย์สินของผู้ประกอบการจดทะเบียน มีหน้าที่ออกใบกำกับภาษีในนามของผู้ประกอบการจดทะเบียนเจ้าของทรัพย์สิน สำหรับกิจการที่ได้รับยกเว้นภาษีมูลค่าเพิ่ม กิจการที่จดทะเบียนเสียภาษีมูลค่าเพิ่มในอัตราร้อยละ 1.5 และกิจการที่เสียภาษีธุรกิจเฉพาะ รวมทั้งผู้ประกอบการที่มีได้จดทะเบียนในระบบภาษีมูลค่าเพิ่ม ไม่มีสิทธิออกใบกำกับภาษีให้กับผู้ซื้อแต่อย่างใด

ใบกำกับภาษีมีกี่ประเภท

ใบกำกับภาษี มีหลายประเภท ได้แก่

1. ใบกำกับภาษีแบบเต็มรูป
2. ใบกำกับภาษีอย่างย่อ
3. ใบเพิ่มหนี้
4. ใบลดหนี้
5. ใบเสร็จรับเงินค่าภาษีมูลค่าเพิ่มที่ส่วนราชการเป็นผู้ออก

ผู้ประกอบการจดทะเบียนจะมีสิทธิออกใบกำกับภาษีแบบใด ขึ้นอยู่กับลักษณะของการประกอบกิจการ ดังนี้

1. ใบกำกับภาษีแบบเต็มรูป

ผู้ประกอบการจดทะเบียน โดยทั่วไป ต้องออกใบกำกับภาษีแบบเต็มรูป ซึ่งจะต้องมีรายการครบถ้วนตามที่กฎหมายกำหนด ดังนี้

คำว่า ใบกำกับภาษี

เลขประจำตัวผู้เสียภาษีอากรของผู้ขาย

ชื่อ ที่อยู่ ของผู้ขาย

ชื่อ ที่อยู่ ของผู้ซื้อ

เลขที่ เล่มที่ (ถ้ามีเล่มที่) ของ ใบกำกับภาษี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วัน เดือน ปี ที่ออกใบกำกับภาษี

ชื่อ ชนิด ประเภท ปริมาณ และราคาของสินค้าหรือบริการ

จำนวนภาษีมูลค่าเพิ่ม โดยให้แยกออกจากราคาของสินค้าหรือบริการ ให้ชัดเจน

การจัดทำใบกำกับภาษีแบบเต็มรูปในกรณีต่าง ๆ

1. ใบกำกับภาษีแบบเต็มรูปทั่วไป ใบกำกับภาษีแบบเต็มรูปโดยทั่วไปต้องมีข้อความเหมือนกันทั้งต้นฉบับและสำเนาผู้ประกอบการจดทะเบียนต้องส่งมอบต้นฉบับใบกำกับภาษี (เอกสารฉบับแรก) ให้กับ ผู้ซื้อส่วนสำเนาผู้ประกอบการจดทะเบียนต้องเก็บรักษาไว้เพื่อเป็นหลักฐานประกอบการลงบัญชีและรายงานภาษีขาย กรณีที่มีการขายสินค้าเป็นเงินสด และผู้ประกอบการจดทะเบียนได้ออก ใบกำกับภาษี / ใบส่งของ อยู่ในฉบับเดียวกัน โดยส่งมอบสำเนาใบกำกับภาษี / ใบส่งของให้กับผู้ซื้อพร้อมกับการส่งมอบสินค้า ส่วนต้นฉบับของใบกำกับภาษี/ใบส่งของ จะส่งมอบให้กับผู้ซื้อต่อเมื่อ ได้รับชำระราคาสินค้า กรณีนี้ถือว่ามิได้ปฏิบัติให้ถูกต้องตามกฎหมาย ซึ่งกำหนดให้ผู้ประกอบการจดทะเบียนต้องส่งมอบต้นฉบับของใบกำกับภาษี (เอกสารฉบับแรก) ให้กับผู้ซื้อเมื่อมีการส่งมอบสินค้า สำหรับสำเนาใบกำกับภาษี / ใบส่งของที่ผู้ซื้อ ได้รับ ผู้ซื้อจะนำไปใช้เป็นหลักฐาน ในการขอหักภาษีซื้อไม่ได้ ในทางปฏิบัติ หากผู้ประกอบการต้องการเก็บต้นฉบับใบส่งของไว้เพื่อเป็นหลักฐาน ในการฟ้องคดีกรณีผู้ซื้อไม่ชำระราคาสินค้า ผู้ประกอบการควรออกใบกำกับภาษีแยกต่างหากจากใบส่งของและส่งมอบต้นฉบับใบกำกับภาษี พร้อมทั้งสำเนาใบส่งของให้กับผู้ซื้อเมื่อมีการส่งมอบสินค้า โดยเก็บต้นฉบับใบส่งของไว้เพื่อเป็นหลักฐาน ในการชำระหนี้ หรือเพื่อความสะดวก ผู้ประกอบการอาจออกใบกำกับภาษีและใบส่งของอยู่ใน เอกสารชุดเดียวกัน โดยถือปฏิบัติตามข้อกำหนดเกี่ยวกับการออกใบกำกับภาษีแบบเป็นชุด ก็ได้

2. ใบกำกับภาษีแบบเต็มรูปเป็นชุดออกรวมกับเอกสารทางการค้าอื่น ในกรณีผู้ประกอบการจดทะเบียนประสงค์จะออกใบกำกับภาษีร่วมกับเอกสารทางการค้าอื่น เช่น ใบเสร็จรับเงิน ใบส่งของ ใบแจ้งหนี้ ซึ่งมีจำนวนหลายฉบับอยู่ในชุดเดียวกัน และใบกำกับภาษีมิใช่เอกสารฉบับแรกของเอกสารดังกล่าว ให้ปฏิบัติดังต่อไปนี้

2.1 กรณีผู้ประกอบการจดทะเบียน ได้ส่งมอบสินค้าก่อน ได้รับชำระราคาสินค้าหรือได้ออกใบกำกับภาษีก่อน ได้รับชำระราคาค่าบริการ

2.1.1 ในเอกสารฉบับแรกของเอกสารชุดดังกล่าว หรือในเอกสารฉบับอื่นที่มี ใ้ฉบับแรก แต่เป็นต้นฉบับของเอกสารฉบับอื่นใดก็ตาม ซึ่งมีใ้ฉบับที่เป็น ใบกำกับภาษี จะต้องมีข้อความว่า ไม่ใช่ใบกำกับภาษี ไว้ในเอกสารฉบับนั้นด้วย

2.1.2 ในเอกสารทุกฉบับของเอกสารชุดดังกล่าว จะต้องมีข้อความว่า เอกสารออกเป็นชุด ไว้ด้วย

2.1.3 ในเอกสารฉบับที่ถือว่าเป็นสำเนาของเอกสารฉบับอื่นจะต้องมีข้อความว่าสำเนา ไว้ทุกฉบับ

2.2 กรณีผู้ประกอบการจดทะเบียน ได้รับชำระราคาสินค้าก่อนส่งมอบสินค้า หรือได้รับชำระราคาค่าบริการก่อนการให้บริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 ในเอกสารฉบับแรกของเอกสารชุดดังกล่าว หรือในเอกสารฉบับอื่นที่มี
ไขฉบับแรก แต่เป็นต้นฉบับของเอกสารฉบับอื่นใดก็ตาม ซึ่งมิใช่ ฉบับที่เป็นใบกำกับภาษี จะต้องมีย่อ
ความว่า ไม่ใช่ใบกำกับภาษี ไว้ในเอกสารฉบับนั้นด้วย

2.2.2 ในเอกสารทุกฉบับของเอกสารชุดดังกล่าว จะต้องมีย่อความว่า เอกสาร
ออกเป็นชุด ไว้ด้วย

2.2.3 ในเอกสารฉบับที่ถือว่าเป็นสำเนาของเอกสารฉบับอื่นจะต้องมีย่อความ
ว่า สำเนา ไว้ทุกฉบับ ย่อความว่า ไม่ใช่ใบกำกับภาษี และเอกสารออกเป็นชุด และ สำเนา ตามที่กล่าวข้าง
ต้น จะต้องตีพิมพ์ขึ้นหรือจัดทำขึ้นด้วยระบบคอมพิวเตอร์ ในกรณีจัดทำใบกำกับภาษีขึ้นด้วยระบบ
คอมพิวเตอร์ทั้งฉบับ จะประทับด้วยตราขาง เขียนด้วยหมึก พิมพ์ดีด หรือกระทำให้ปรากฏขึ้นด้วยวิธีการ
อื่นในลักษณะทำนองเดียวกันไม่ได้

3 กรณีผู้ประกอบการจดทะเบียนมีสถานประกอบการหลายแห่ง และ สถานประกอบการที่มีมิใช่
สำนักงานใหญ่ได้นำใบกำกับภาษีของสถานประกอบการที่เป็นสำนักงาน ใหญ่ไปส่งมอบให้แก่ผู้ซื้อสินค้า
หรือผู้รับบริการทุกครั้งที่มีการขายสินค้าหรือการ ให้บริการ จะต้องมีย่อความว่าสาขาที่ออกใบกำกับภาษี
คือ ไว้ในใบกำกับภาษีดังกล่าว โดยข้อความดังกล่าวจะตีพิมพ์ จัดทำขึ้นด้วยระบบคอมพิวเตอร์ ประทับ
ด้วยตราขาง เขียนด้วยหมึก พิมพ์ดีด หรือทำให้ปรากฏขึ้นด้วย วิธีการอื่นใดในลักษณะทำนองเดียวกันก็ได้

4 กรณีผู้ประกอบการจดทะเบียนซึ่งประกอบกิจการสถานบริการน้ำมัน ได้ขายน้ำมันเชื้อเพลิง
หรือได้ขายสินค้าหรือบริการที่เกี่ยวข้อง โดยตรงกับรถยนต์ ที่ ต้องจดทะเบียนตามกฎหมายว่าด้วยรถยนต์
จะต้องระบุเลขทะเบียนรถยนต์ไว้ใน ใบกำกับภาษี โดยข้อความดังกล่าวจะตีพิมพ์ จัดทำขึ้นด้วยระบบ
คอมพิวเตอร์ ประทับด้วยตราขาง เขียนด้วยหมึก พิมพ์ดีด หรือทำให้ปรากฏขึ้นด้วยวิธีการอื่นใดใน
ลักษณะทำนองเดียวกันก็ได้

5 กรณีของการฝากขายสินค้าตามสัญญาการ ตั้งตัวแทนเพื่อขายสินค้า ซึ่งตัวการและตัวแทนเป็น
ผู้ประกอบการจดทะเบียนที่เสียภาษีมูลค่าเพิ่มในอัตราร้อยละ 10 หรือร้อยละ 0 และทำสัญญาตั้งตัวแทน
เพื่อขายสินค้า โดยตัวการ ได้ส่งมอบสัญญาให้แก่เจ้าพนักงานสรรพากรภายใน 15 วัน นับแต่วันทำสัญญา
ตั้งตัวแทน เมื่อตัวการ (ผู้ฝากขาย) ได้ส่งมอบสินค้าให้ ตัวแทน (ผู้รับฝาก) แล้วเมื่อตัวแทน ได้ขายสินค้าให้
กับผู้ซื้อ ตัวแทนต้องออกใบกำกับภาษีแบบเต็มรูปในนามของตัวการ โดยตัวแทนต้องระบุชื่อ ที่อยู่และเลข
ประจำตัวผู้เสียภาษีอากรของตัวแทนด้วย

การยกเลิกใบกำกับภาษีฉบับเดิม แล้วออกฉบับใหม่

ผู้ประกอบการมีหน้าที่ตรวจสอบว่า ใบกำกับภาษีที่ตนได้รับมามีราชการครบถ้วนถูกต้องตามที่
กล่าวแล้วข้างต้นหรือไม่ หากพบว่าใบกำกับภาษีดังกล่าวมีราชได้ไม่ครบถ้วนถูกต้อง ผู้ประกอบการจด
ทะเบียนต้องขอให้ผู้ออกใบกำกับภาษียกเลิกและออกฉบับใหม่ที่ถูกต้องแทนก่อนที่จะนำไปคำนวณหัก
เป็นภาษีซื้อ การยกเลิกใบกำกับภาษีฉบับเดิม แล้วออกฉบับใหม่ ให้กระทำได้ดังนี้

1 เมื่อผู้ประกอบการจดทะเบียนได้รับการร้องขอจาก ผู้ประกอบการจดทะเบียนที่ซื้อสินค้าหรือ
รับบริการ ให้ยกเลิกใบกำกับภาษีฉบับเดิมและออกฉบับใหม่แทนเนื่องจากราชการ ใบกำกับภาษีไม่เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไปตามที่กฎหมายกำหนด หรือราชการ ใบกำกับภาษีไม่ถูกต้อง ให้ผู้ประกอบการจดทะเบียนที่เป็นผู้ออกใบกำกับภาษีเรียกคืนใบกำกับภาษีฉบับเดิม ทำการประทับตรา หรือขีดฆ่ายกเลิก แล้วให้เก็บรวมไว้กับสำเนาใบกำกับภาษีฉบับเดิม แล้วจึงออกใบกำกับภาษีฉบับใหม่ให้แก่ผู้ประกอบการที่ร้องขอ โดยลงวันเดือนปี ในใบกำกับภาษีฉบับใหม่ ให้ตรงกับวันเดือนปี ในใบกำกับภาษีฉบับเดิม

2 ให้ผู้ประกอบการจดทะเบียน ที่ออกใบกำกับภาษีฉบับใหม่หมายเหตุไว้ในใบกำกับภาษีฉบับใหม่ด้วยว่าเป็นการยกเลิกและออกใบกำกับภาษีฉบับใหม่แทนฉบับเดิม เลขที่..... เล่มที่.....

3 ผู้ประกอบการที่ขอให้ยกเลิกใบกำกับภาษีฉบับเดิมและออกฉบับใหม่ ต้องถ่ายเอกสารใบกำกับภาษีฉบับเดิม (ฉบับที่ขอยกเลิก) ติดเรื่องไว้ และส่งมอบใบกำกับภาษีฉบับที่ขอยกเลิกคืนให้ผู้ออกใบกำกับภาษีนำไปรวมกับสำเนาไว้เป็นหลักฐาน

4 ให้ผู้ประกอบการจดทะเบียนที่ออกใบกำกับภาษีฉบับใหม่หมายเหตุ การยกเลิกใบกำกับภาษีไว้ในรายงานภาษีขายประจำเดือนที่ออกใบกำกับภาษีฉบับใหม่แทนด้วย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สิ่งที่ควรรู้ก่อนเริ่มเล่น Visual Basic 6.0

ก่อนเริ่มเล่น Visual Basic 6.0

ในอดีตการเขียนโปรแกรมหรือการพัฒนาแอปพลิเคชันขึ้นมาสักตัว โปรแกรมเมอร์จะต้องมีความรู้ ความเข้าใจในศัพทภาษา ที่จะนำมาใช้ในการพัฒนา เป็นอย่างมาก ไม่ว่าจะพัฒนาด้วยภาษาใดก็ตาม เช่น ภาษา C, C++, ปาสคาล เป็นต้น และจะต้องเขียนโค้ดที่มีความสัมพันธ์กันตั้งแต่ บรรทัดแรกจนถึง บรรทัดสุดท้าย หรือที่เรียกว่า การเขียน โค้ดแบบ command line อีกทั้งยังต้องออกแบบรูปร่างหน้าตาของ แอปพลิเคชัน ให้ตรงกับความต้องการของผู้ใช้อีกด้วย ซึ่ง ถ้าถูกใจผู้ใช้ก็ดีไป แต่ถ้าต้องมีการแก้ไขแล้วละก็ ทุกสิ่งทุกอย่างที่คุณ ได้ทำมาก็ต้องแทบหรือทำใหม่หมด ทำให้โปรแกรมเมอร์เสียเวลา ในการพัฒนาแอปพลิเคชันเป็นอย่างยิ่ง

ต่อมาไมโครซอฟท์ได้นำเสนอรูปแบบในการเขียนแอปพลิเคชันชนิดใหม่ด้วยการออก Visual Basic 1.0 แม้ว่า เวอร์ชันแรกนี้ จะถูกโปรแกรมเมอร์ในยุคนั้น มองว่าเป็นเวอร์ชันทดลอง แต่มันก็ได้สร้างความแปลก ใหม่ในการเขียน โปรแกรมเป็นอย่างยิ่ง ความยุ่งยากซับซ้อน ถูกซ่อนไว้เบื้องหลัง มีแต่ความสะดวกสบายไว้เบื้องหน้า ที่เตรียมไว้ให้โปรแกรมเมอร์ เนื่องจาก concept ในการเขียน โปรแกรมแทบจะเปลี่ยนไปอย่างสิ้นเชิง จนกระทั่งในยุคปัจจุบัน Visual Basic ได้พัฒนามาถึงเวอร์ชัน 6.0 แล้ว ความสามารถของตัว ภาษา VB เองก็มีมากขึ้นเพราะ ความก้าวหน้าของเทคโนโลยีในยุคปัจจุบัน ทำให้ไมโครซอฟท์เพิ่มเติม ฟีเจอร์ต่างๆเข้าไปมากมาย จนกระทั่ง VB แทบจะเป็นเครื่องมือ ในการพัฒนาแอปพลิเคชันที่ไร้เทียมทาน เพราะเทคโนโลยีใหม่ๆ ล้วนแล้วแต่มาจาก ไมโครซอฟท์แทบทั้งสิ้น ดังนั้นจึงไม่เป็นที่แปลกใจเลยว่า ไมโครซอฟท์ ได้เพิ่มความสามารถในทุกๆเวอร์ชันของ VB เสมอ เช่น สามารถสร้าง แอปพลิเคชันชนิด DHTML ซึ่งใช้ run บน web ได้, รวมถึงการผนวกเทคโนโลยี ActiveX เข้ากับตัว คอนโทรลของ VB ทำให้สามารถเชื่อมโยงเข้ากับเครื่องมือ ที่สนับสนุนเทคโนโลยีนี้ได้อีกด้วย แต่ VB ยังคง รักษาเอกลักษณ์อย่างหนึ่งไว้ได้เป็นอย่างดีนั่นคือ สามารถที่จะพัฒนาแอปพลิเคชันได้ ในระยะเวลาอันสั้น รวมถึงความง่าย ต่อการเรียนรู้ในศัพทภาษา และทำความเข้าใจได้อย่างรวดเร็ว เมื่อเทียบกับการเริ่มต้นเรียนรู้ด้วยภาษาอื่นๆ คุณสามารถที่จะสร้างแอปพลิเคชันออกมา ได้อย่างทันอกทันใจ เพราะ VB ได้ลดขั้นตอนที่แสนจะยุ่งยากเก็บไว้ เบื้องหลัง แล้วนำเสนอผลงานของคุณ ได้อย่างรวดเร็ว VB ได้จัดเตรียม เครื่องมือต่างๆที่เรียกว่า คอนโทรล (controls) ไว้คอยอำนวยความสะดวกให้แก่ โปรแกรมเมอร์มากมาย คุณจะต้องศึกษา และทำความเข้าใจกับตัวคอนโทรลให้มากที่สุด ซึ่งตัวคอนโทรลเหล่านี้เองที่อยู่เบื้องหลังทำให้ VB ประสบความสำเร็จเป็นอย่างยิ่ง ในยุคปัจจุบัน เพราะเนื่องจากมัน ได้ลดขั้นตอนต่างๆ ในการพัฒนา ไปได้มากที่สุด

แนวทางการเขียนโปรแกรมด้วยภาษา VB จะนำไปในลักษณะการนำคอนโทรลชนิดต่างๆ เช่น TextBox, Label, ComboBox เป็นต้น นำมาวางเพื่อออกแบบหน้าต่างแอปพลิเคชันที่เรียกว่า กราฟฟิกยูสเซอร์อินเตอร์เฟซ (Graphic User Interface-GUI) คุณสามารถที่จะออกแบบ อินเตอร์เฟซได้อย่างอิสระให้ตรงกับจุดประสงค์และ การนำไปใช้งานของคุณก่อน แล้วจึงเริ่มเขียน โค้ด เพื่อตอบสนองการกระทำของผู้ใช้ (ใน VB เรียกว่า เหตุการณ์ event) ซึ่งถือเป็นหลักการเขียน โปรแกรมที่เรียกว่า การเขียนโปรแกรมเพื่อตอบสนอง เหตุการณ์ที่เกิดขึ้น (Event-Driven Programming)

สิ่งต่างๆ ที่คุณนำไปใช้ร่วมกันเพื่อเป็นแอปพลิเคชันหนึ่งๆ เช่น แถบเมนู, dialog box, toolbars, TextBox, ปุ่ม OK ฯลฯ เป็นต้น จะถูกมองเปรียบเสมือนเป็นวัตถุชิ้นหนึ่งที่เรียกว่า อ็อบเจกต์ (object model) ทุกสิ่งทุกอย่าง ในแอปพลิเคชัน VB จะ มองเป็นอ็อบเจกต์ ที่คุณสามารถควบคุมพฤติกรรม, แก้ไข และกระทำโดยตรงต่ออ็อบเจกต์นั้นได้ ด้วยการเขียน โค้ด หรือสามารถเปลี่ยนแปลง คุณสมบัติ (properties) ประจำตัวของอ็อบเจกต์นั้น ได้โดยตรง ตัวคอนโทรลก็ถูกมองเป็นอ็อบเจกต์ เช่นกัน ในทุกๆ อ็อบเจกต์จะมีคุณสมบัติ (properties) และ เมธอด (methods) ประจำตัว ในแต่ละอ็อบเจกต์ อาจจะมีคุณสมบัติและเมธอดที่เหมือนหรือต่างกันได้ขึ้นอยู่กับชนิดของอ็อบเจกต์ คุณจะพัฒนาแอปพลิเคชัน โค้ดดีหรือไม่ ส่วนหนึ่งจะมาจากที่คุณสามารถใช้งานคอนโทรล, แก้ไขคุณสมบัติและ เมธอดได้ตรงตามความต้องการของคุณ และเต็มประสิทธิภาพของคอนโทรลนั้นๆ

ในการพัฒนาแอปพลิเคชันด้วย VB การเขียน โค้ดจะถูกแบ่งออกเป็นส่วนๆ ที่เรียกว่า โพรซีเจอร์ (procedure) แต่ละ โพรซีเจอร์จะประกอบไปด้วย โค้ดที่คุณพิมพ์เข้าไปแล้วทำให้คอนโทรลหรืออ็อบเจกต์นั้นๆ ตอบสนองการกระทำ ของผู้ใช้ได้โดยสมบูรณ์ ในตัวมันเอง ซึ่งเรียกว่า การเขียน โปรแกรมเชิงวัตถุ (Object Oriented Programming-OOP) แต่ด้วยภาษา VB ยังไม่ถึงว่าเป็น OOP อย่างแท้จริง การเขียนโปรแกรมเชิงวัตถุ มีข้อดีก็คือ โค้ดจะถูกแบ่งออกเป็นส่วนๆ ทำให้ง่ายต่อการตรวจสอบ และดีกัจับข้อผิดพลาด (debug) ซึ่งการแก้ไขดังกล่าวนี้ ไม่ได้ไปกระทบกับโค้ด ส่วนอื่นๆ ในตัวแอปพลิเคชันนั้นเลย ทำให้โปรแกรมเมอร์สามารถพัฒนาแอปพลิเคชันออกมา ได้อย่างสมบูรณ์แบบมากที่สุด โดยที่ไม่ต้องเสียเวลามากมายดังเช่นในอดีต

Hardware ที่เหมาะสมกับการเล่น Visual Basic 6.0

เพื่อที่จะสามารถรัน Visual Basic ได้อย่างไม่มีปัญหา คุณต้องแน่ใจก่อนว่า hardware และ software ของคุณพร้อมแล้ว ซึ่งความต้องการของระบบมีดังนี้

1. Microsoft Windows 95 หรือมากกว่า, หรือ Microsoft Windows NT Workstation 4.0 (Service Pack 3 recommended) หรือมากกว่า
2. 486DX/66 MHz หรือสูงกว่า (แนะนำให้ใช้ Pentium หรือสูงกว่า), หรือชิป Alpha processor ที่สามารถรัน Microsoft Windows NT Workstation ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

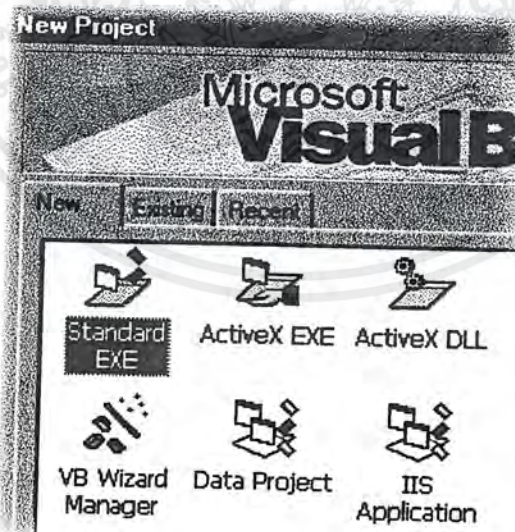
3. ไครพี CD-ROM
4. การ์ด VGA หรือสูงกว่า สนับสนุนการแสดงผลระบบ Windows
5. แรม 16 MB สำหรับ Windows 95, 32 MB สำหรับ Windows NT Workstation.
6. เม้าส์และอื่นๆที่ระบบ Windows รองรับ

หมายเหตุจากผู้เขียน นี่เป็นสเปกขั้นต่ำของการเล่น Visual Basic 6.0 ที่ไมโครซอฟท์ระบุไว้ ถ้าคุณรัน Visual Basic 6.0 ภายใต้ Windows 98 คุณควรจะใช้ เครื่องขนาด Celeron 300A กับแรมขนาด 64 MB ขึ้นไป จึงจะสามารถเล่น Visual Basic 6.0 โดยที่คุณไม่รำคาญ :-) โดยเฉพาะแรม ยิ่งมากยิ่งดี

เริ่มต้นกับ Visual Basic 6.0

สิ่งแรกที่คุณจะได้พบเมื่อคุณเข้าสู่โปรแกรม VB 6.0 ก็คือ VB จะแสดงไดอะล็อกบ็อกซ์เพื่อให้คุณเลือกพัฒนาชนิดของแอปพลิเคชัน ใน VB 6.0 ชนิดของแอปพลิเคชันมีมากมายหลายชนิด เช่น ชนิด DHTML เป็น แอปพลิเคชันที่ใช้รันบน web คุณสามารถสร้างเว็บเพจแบบ DynamicHTML ได้ตรงตามมาตรฐานขององค์การ w3 consortium คุณ สามารถตรวจสอบได้จาก www.w3.org เป็นต้น

ในเบื้องต้นผู้เขียนขอแนะนำให้เลือกพัฒนาแอปพลิเคชันชนิด Standard.EXE ก่อน เพราะเป็นแอปพลิเคชัน ที่ใช้ทั่วไป เมื่อคุณคอมไพล์ (compile) โปรเจกต์แล้ว จะได้แอปพลิเคชันที่มีนามสกุล .exe เมื่อคุณเลือก ชนิดของแอปพลิเคชันแล้ว จะเข้าสู่สภาพแวดล้อม (environment) ของ VB 6.0 ซึ่งไมโครซอฟท์ เรียกว่า Integrated Development Environment – VBIDE

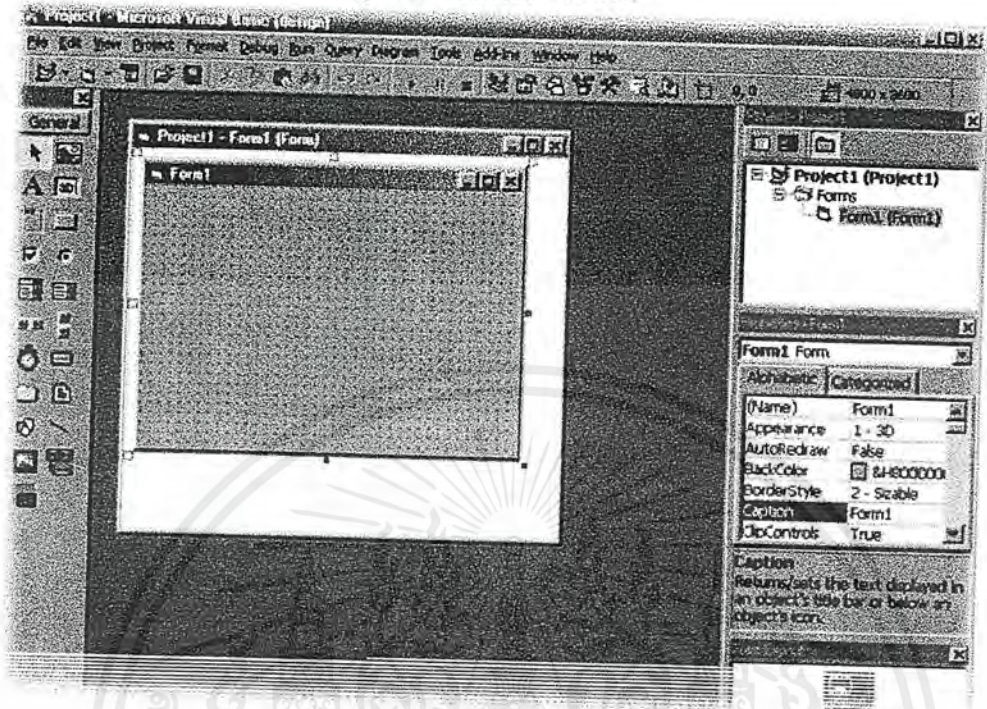


รูปที่ 1-1 แสดงไดอะล็อกบ็อกซ์ชนิดของแอปพลิเคชันสำหรับให้โปรแกรมเมอร์เลือกพัฒนา

ทำความเข้าใจกับส่วนต่างๆของ VBIDE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VBIDE คือ กลุ่มของเมนู, ชุดบาร์, หน้าต่าง properties, หน้าต่าง project explorer ฯลฯ ที่ประกอบกันขึ้นมาเป็นสภาพแวดล้อมของโปรแกรม VB 6.0 ดังนั้น ต่อไปถ้าผู้เขียนจะกล่าวถึงสภาพแวดล้อมของ VB 6.0 ผู้เขียนจะใช้คำว่า " VBIDE "



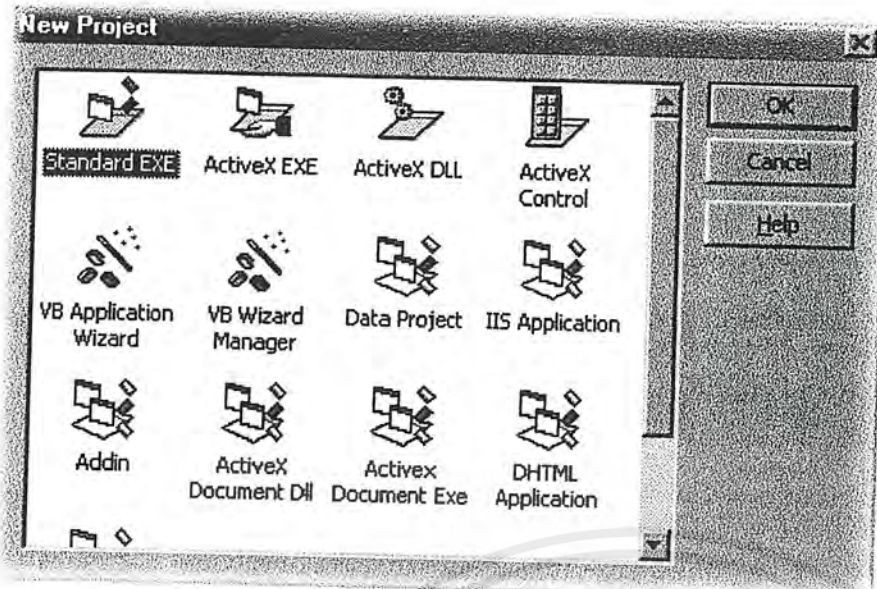
รูปที่ 1-2 แสดงสภาพแวดล้อมของ VB 6.0

ในแต่ละส่วนของ VBIDE จะมีหน้าที่แตกต่างกันไป ซึ่ง ในระหว่างการพัฒนาแอปพลิเคชัน คุณจะต้องใช้ส่วนต่างๆ เหล่านี้ ในการพัฒนาแอปพลิเคชัน เช่น แถบเมนูบาร์ (Menu Bars) จะมีคำสั่งต่างๆ ที่ครอบคลุมการทำงานทั้งหมดในการพัฒนา, แถบชุดบาร์ (Tool Bars) จะประกอบไปด้วย ปุ่มต่างๆ ที่แทนคำสั่งในเมนูที่ใช้งานบ่อยๆ เช่น การเปิดโปรเจกต์, เซฟโปรเจกต์ เป็นต้น หัวข้อนี้จะเป็นการอธิบายการใช้งาน VBIDE เบื้องต้นที่คุณควร ทราบและสำหรับใน ส่วนที่ผู้เขียน ไม่ได้อธิบายในหัวข้อนี้ ผู้เขียนจะอธิบายเมื่อเมื่อมีความสัมพันธ์กับ VBIDE ส่วนอื่นๆ ซึ่งจะช่วยให้คุณ สามารถทำความเข้าใจได้อย่างต่อเนื่องและรวดเร็ว ส่วนเมนูบาร์ผู้เขียนจะไม่อธิบายโดยตรง เพราะเนื่องจากว่า คำสั่งในเมนูบาร์ที่มีการใช้งานบ่อยๆ เกือบทั้งหมด ได้ถูกรวบรวมไว้ในชุดบาร์แล้ว ดังนั้นเมื่อมีการกล่าวถึงการ ใช้งานชุดบาร์ใดผู้เขียนจะอ้างถึงคำสั่งในเมนูบาร์ด้วย

หน้าต่าง New Project (New Project Window)







หน้าต่าง New Project จะปรากฏขึ้นมาเมื่อคุณเริ่มต้นรัน VB หรือเลือกเมนู File/New Project ได้อะลือกบ็อกซ์นี้ จะแสดงชนิดของแอปพลิเคชัน ที่คุณต้องการพัฒนา ไอคอนแต่ละตัวจะแทนประเภทของแอปพลิเคชัน ซึ่งสื่อด้วยชื่อ หรือนามสกุลที่ปรากฏประจำตัว ไอคอนนั้นๆ โดยปกติแล้ว แอปพลิเคชันชนิด Standard.EXE จะเป็นแอปพลิเคชันพื้นฐาน ที่คุณควรจะเริ่มค้นพัฒนา เพราะเป็นแอปพลิเคชันที่ใช้งานโดยทั่วไป ไฟล์ที่ได้จากการ คอมไพล์แล้วจะมีนามสกุล .exe

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้










รูปที่ 1-3 แสดงไอคอนบ็อกซ์ New Project

สำหรับรายละเอียดแอปพลิเคชันชนิดอื่นๆ มีดังนี้

 Standard EXE	ใช้พัฒนาแอปพลิเคชันทั่วไป
 ActiveX EXE	ใช้พัฒนาแอปพลิเคชันที่สามารถใช้งาน และเชื่อมโยงกับแอปพลิเคชันอื่นๆที่สนับสนุนเทคโนโลยี ActiveX
 ActiveX DLL	เป็นแอปพลิเคชันชนิดเดียวกับ ActiveX.EXE แต่จะเก็บอยู่ในไฟล์ไลบรารีที่ .dll ซึ่งไม่สามารถรันได้ด้วยตัวมันเอง จะต้องถูกเรียกใช้งานจากแอปพลิเคชันอื่นๆ
 ActiveX Control	ใช้สร้างคอนโทรลขึ้นมาใช้งานเอง
 VB Application Wizard	เป็นเครื่องมือที่อำนวยความสะดวกให้สามารถสร้างแอปพลิเคชันได้อย่างรวดเร็ว โดยมันจะสร้างองค์ประกอบเบื้องต้น หลักๆ ของแอปพลิเคชัน จากขั้นตอนที่คุณได้เลือกไว้ แอปพลิเคชันที่ได้จะมีความสมบูรณ์ในระดับหนึ่งเท่านั้น คุณต้องปรับปรุง, เพิ่มเติมและแก้ไขเพื่อทำให้เป็นมาตรฐานยิ่งขึ้น ถือได้ว่าเป็นเครื่องมือที่ช่วยให้โปรแกรมเมอร์ เริ่มต้นโปรเจกต์ได้เป็นอย่างดี ผู้เริ่มต้นอาจใช้เครื่องมือตัวนี้เพื่อศึกษาแนวทางในการพัฒนาแอปพลิเคชันก็ได้
 VB Wizard Manager	ใช้สำหรับสร้างโปรแกรมการจัดการต่างๆ เช่นการติดต่อกับฐานข้อมูล เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

 Data Project	เป็นชนิด โปรเจกต์ที่เป็นแบบฟอร์ม ติดต่อกับฐานข้อมูล โดยผ่านทางคอนโทรล ADO Data Control
 IIS Application	แอปพลิเคชันชนิด Web server
 Addin	ใช้สำหรับเพิ่มเติม utility เข้าไปใน VBIDE เพื่อเพิ่มประสิทธิภาพ ซึ่งอาจจะมาจากผู้ผลิตอิสระรายอื่นๆ
 ActiveX Document Dll	ใช้สร้างแอปพลิเคชันชนิดที่รันบน Internet จะเก็บอยู่ในรูปไฟล์ .dll ไม่สามารถรันได้ด้วยตัวเอง ต้องให้แอปพลิเคชันอื่นๆ ที่สนับสนุนเทคโนโลยี ActiveX เรียกใช้งาน
 ActiveX Document Exe	ใช้สร้างแอปพลิเคชันชนิดที่รันบน Internet เช่นกัน แต่จะเก็บอยู่ในรูปไฟล์ .exe สามารถรันได้ด้วยตัวเอง แต่ server จะต้องสนับสนุนเทคโนโลยี ActiveX เช่นกัน
 DHTML Application	ใช้พัฒนาแอปพลิเคชันรูปแบบของ เอกสาร Dynamic HTML ซึ่งจะเป็มาตรฐานใหม่ของการแสดงผลบน web
 VB Enterprise Edition Co...	ใช้สำหรับ โหลด VBIDE ในรูปแบบที่ใช้พัฒนาแอปพลิเคชันในระดับ Enterprise ซึ่ง VB จะเพิ่มเติมคอนโทรล ActiveX อีกจำนวนหนึ่งขึ้นมาโดยอัตโนมัติ เหมาะสำหรับเครื่องที่มีประสิทธิภาพสูง และใช้กับโปรแกรมเมอร์มืออาชีพ ที่มีความเชี่ยวชาญมากพอสมควร

สำหรับใน ไอเดอล็อกบ็อกซ์ New Project จะมีแท็บ (Tab) อยู่อีก 2 แท็บมีรายละเอียดดังนี้

- Existing** ใช้สำหรับเปิด โปรเจกต์ที่คุณมีอยู่แล้ว
- Recent** จะแสดงรายชื่อ โปรเจกต์ที่ถูกพัฒนาขึ้นมาครั้งล่าสุด โดยจะจัดเรียงลำดับไปเรื่อยๆ จะแสดงรายละเอียด โค้ด, โฟลเดอร์และชื่อโปรเจกต์ ที่คุณจัดเก็บไว้

ทูลบาร์ (Tool Bars)

เป็นที่รวบรวมคำสั่งที่ใช้งาน ในการพัฒนาแอปพลิเคชันบ่อที่สุด ซึ่งจะมีคำสั่งเหมือนกันกับเมนูบาร์ คุณสามารถที่จะปรับแต่งทูลบาร์นี้ได้ โดยการเลือกเมนู View/Toolbars (หรือให้คุณคลิกขวาวบริเวณใดก็ได้บนทูลบาร์จะมี Pop-up เมนูปรากฏขึ้นมา ซึ่งจะมีคำสั่งให้เลือกเช่นกัน) สำหรับชื่อของปุ่มต่างๆ ที่อยู่บนทูลบาร์ คุณไม่จำเป็นต้องไปจดจำว่า มันแทนคำสั่งอะไร เพียงแค่ให้คุณเลื่อนเมาส์ไปวางบนปุ่มนั้น 2-3 วินาที จะมีทูลทิป (ToolTip) ขึ้นมา เพื่อบอกคำสั่งที่ปุ่มนั้นทดแทนอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทูลบาร์สามารถแบ่งออกได้เป็น 4 กลุ่มใหญ่ๆ คือ

1.ทูลบาร์ Standard ถือได้ว่าเป็นทูลบาร์ปกติ (default)ที่คุณต้องใช้งานทุกครั้งและบ่อยที่สุด เนื่องจากว่ามันประกอบไปด้วย คำสั่งที่เกี่ยวข้องกับการใช้งานทั่วไป เช่น การเปิดโปรเจกต์,เซฟโปรเจกต์ เป็นต้น เป็นหัวใจหลักของทูลบาร์เลยก็ว่าได้ ซึ่งรวบรวมคำสั่งมาจากเมนู File,Project,Debug,Run,Tool เป็นต้น



รูปที่ 1.5.1 รายการทูลบาร์ Standard

2.ทูลบาร์ Edit จะใช้ทูลบาร์นี้เมื่อคุณเริ่มเขียนโค้ดใน code editor คำสั่งหลักของทูลบาร์กลุ่มนี้ก็คือ Cut,Paste ซึ่งก็คือ คำสั่งในเมนู Edit นั่นเอง



รูปที่ 1.5.2 รายการทูลบาร์ Edit

3.ทูลบาร์ Debug จะประกอบไปด้วยคำสั่งที่ใช้ในการตรวจสอบโค้ดของคุณ เช่น Run,Stop,Pause เป็นต้น เป็นกลุ่มคำสั่งที่คุณต้องใช้บ่อยเช่นกัน เพราะจะเป็นการทดสอบโค้ดของคุณว่าทำงานได้ตามความต้องการของคุณหรือไม่ ในบางครั้งอาจต้องใช้ควบคู่ไปกับหน้าต่าง Immediate



รูปที่ 1.5.3 รายการทูลบาร์ Debug

4.ทูลบาร์ Form Editor คุณจะใช้กลุ่มคำสั่งนี้เมื่อคุณต้องการปรับขนาด,ย้าย,เปลี่ยนตำแหน่งคอนโทรลต่างๆ ที่อยู่บนฟอร์ม เป็นคำสั่งที่เหมือนกับเมนู Format



รูปที่ 1.5.4 รายการทูลบาร์ Form Editor

ทูลบ็อกซ์ (Tool Box)

แถบทูลบ็อกซ์เป็นที่รวบรวมคอนโทรล (controls) ต่างๆที่ใช้ในการพัฒนาแอปพลิเคชัน คุณสามารถเลือกใช้คอนโทรลชนิดต่างๆ ได้จากที่นี่ คอนโทรลเป็นสิ่งที่อยู่เบื้องหลังความสำเร็จของ Visual Basic ทุกเวอร์ชัน เพราะมันได้ซ่อนความยุ่งยากด้านเทคนิคไว้ภายใต้การเลือกใช้งานที่แสนง่าย ตัวคอนโทรลใน VB สามารถแยกออกเป็นกลุ่มใหญ่ๆ ได้ 2 กลุ่มคือ

1.คอนโทรลภายใน (Intrinsic controls) เป็นชุดคอนโทรลมาตรฐานของ VB ทุกๆ ครั้งที่คุณรัน VB คอนโทรลชุดนี้ จะถูกโหลดขึ้นมาเสมอ คุณสามารถเลือกใช้งาน คอนโทรลกลุ่มนี้ได้ทันที โดยที่ไม่ต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

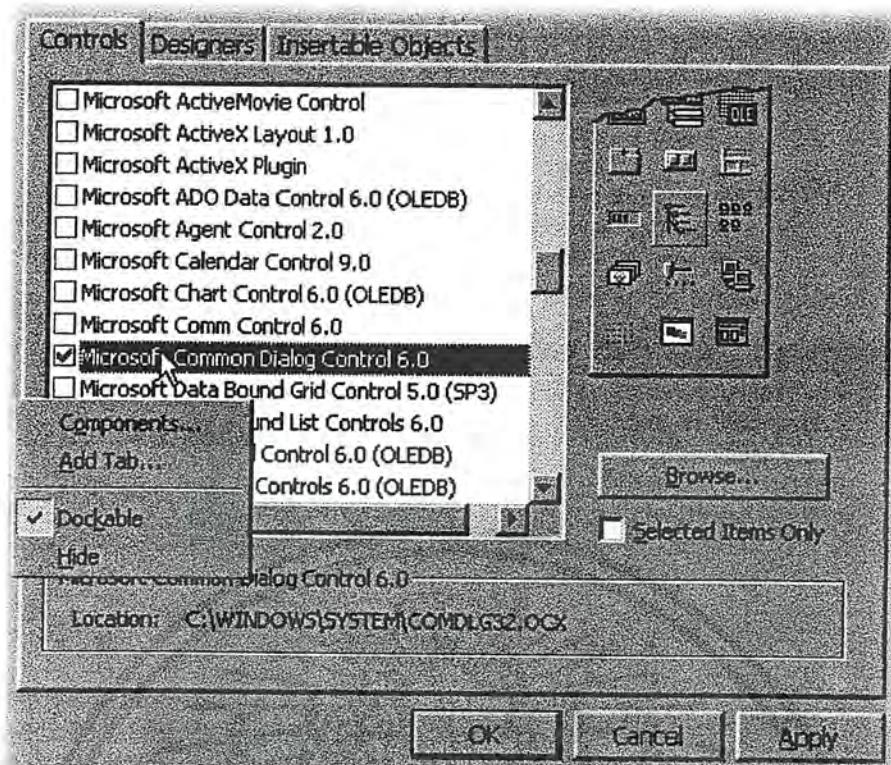
ใช้ไฟล์เพิ่มเติม และคุณไม่สามารถที่จะถอด (remove) คอนโทรลชุดนี้ออกจาก VBIDE ได้ เป็นชุดคอนโทรลที่ใช้งาน โดยทั่วไปในทุกๆ แอปพลิเคชัน คุณจะต้องใช้งานคอนโทรลกลุ่มนี้มากที่สุด

2.คอนโทรล ActiveX (ActiveX controls) เป็นชุดคอนโทรลที่ไม่โครซอฟท์ผนวกเอาเทคโนโลยี ActiveX เข้าไปด้วย ถือเป็นคอนโทรลเพิ่มเติม ถ้าคุณต้องการใช้งานคอนโทรลกลุ่มนี้ คุณต้องใช้ไฟล์ .ocx ประจำแต่ละคอนโทรล เข้ามาประกอบในโปรเจกต์ด้วย การเพิ่มคอนโทรลกลุ่มนี้เข้ามาในทูลบ็อกซ์ โดยการเลือกเมนู Project/Components... (หรือคลิกขวาวบริเวณแถบทูลบ็อกซ์ก็ได้ เลือกคำสั่ง Components... ก็ได้) การใช้งานเบื้องต้น ผู้เขียนขอแนะนำให้คุณยังไม่ต้องเพิ่มเติมคอนโทรลในชุดนี้



รูปที่ 1.6 แสดงแถบเครื่องมือ control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

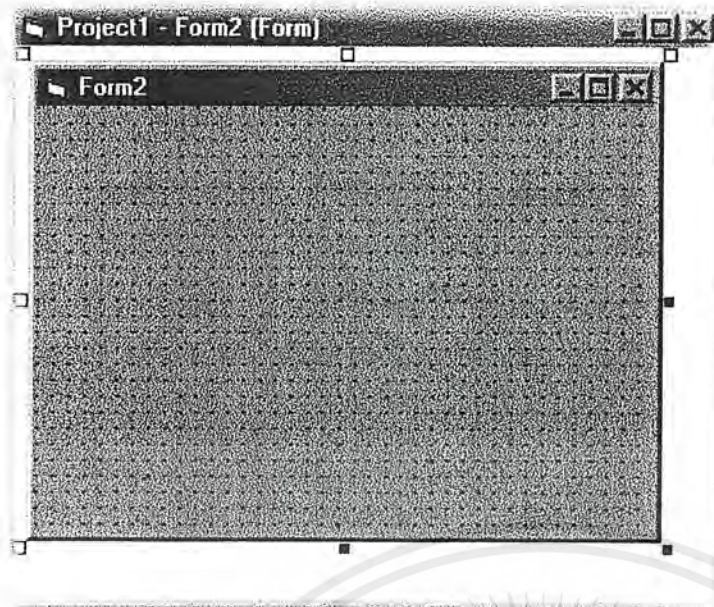


รูปที่ 1.7 แสดงการเพิ่มเติมคอนโทรล ActiveX เข้ามาในทูลบ็อกซ์

สำหรับการเริ่มต้นนี้ผู้เขียนจะกล่าวถึงคอนโทรลภายในก่อน มีรายละเอียดเบื้องต้นดังนี้

ฟอร์ม (Form1)

ฟอร์มถือได้ว่าเป็นอ็อบเจกต์ตัวแรกที่คุณได้ใช้งาน คุณต้องใช้ร่วมกับคอนโทรลต่างๆประกอบกันขึ้นมาเป็นแอปพลิเคชัน ฟอร์มเป็นตัวบรรจุ (container) ของคอนโทรล และทุกครั้งที่คุณรัน VB ฟอร์มต่างๆจะถูกโหลดขึ้นมาเสมอ ซึ่งจะมีความสัมพันธ์กับหน้าต่าง Project Explorer ที่จะกล่าวถึงต่อไปด้วย ใน 1 โปรเจกต์จะประกอบไปด้วยอย่างน้อยที่สุด 1 ฟอร์ม (มีโปรเจกต์บางชนิดจะไม่มีฟอร์ม เช่น พวกนามสกุล .dll เป็นต้น มันจะทำงานร่วมกับระบบ ซึ่งผู้เขียนจะอธิบายในหัวข้อต่อไป) สำหรับรายละเอียดการใช้งานฟอร์ม ผู้เขียนจะอธิบายอีกครั้งในบทที่ 5

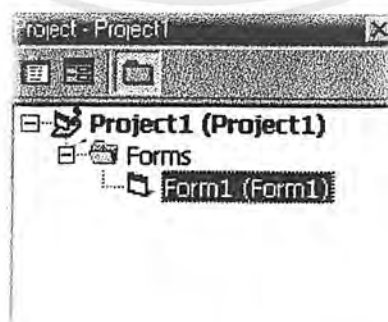


รูปที่ 1.8 แสดงฟอร์มว่างๆใน VBIDE

หน้าต่าง Project Explorer (Project Explorer Window)

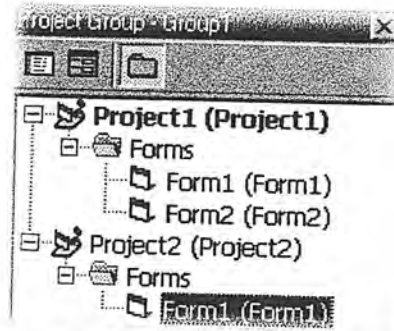
หน้าต่าง Project Explorer ช่วยให้คุณสามารถบริหารและจัดการหลายๆ โปรเจกต์ได้ในเวลาเดียวกันอย่างมีประสิทธิภาพ เพราะเนื่องจาก VB 6.0 สนับสนุนการพัฒนาแบบ multiple project ซึ่งสามารถเซฟโปรเจกต์เป็นกลุ่มงานได้ (นามสกุล .vbproj) โดยที่ VB จะจัดกลุ่มโปรเจกต์ต่างๆ ให้คุณเสมอ ซึ่งถ้ามีโปรเจกต์เดียวก็จะมีนามสกุล .vbproj

หน้าต่าง Project Explorer จะแสดงรายการองค์ประกอบของแต่ละ โปรเจกต์ แบบโครงสร้างต้นไม้ (tree-view) ตัว โปรเจกต์จะถือว่าเป็นตัวแทนแอปพลิเคชันทั้งหมด ซึ่งจะอยู่ส่วนบนสุด ถัดมา จะแสดงองค์ประกอบต่างๆ ของโปรเจกต์นั้นๆ ว่า ประกอบด้วยอะไรบ้าง เช่น มีฟอร์มกี่ฟอร์ม, มีโมดูล เป็นต้น ถ้ามี 2 โปรเจกต์ขึ้นไป ก็จะแสดงแยกออกเป็นส่วนตัวหากอีก โปรเจกต์ ถ้าคุณต้องการใช้งานส่วนใดของโปรเจกต์ไหน คุณสามารถคลิกเลือกได้โดยตรง ซึ่งคุณต้องพิจารณาด้วยว่า รายการใดอยู่ภายใน โครงสร้างของโปรเจกต์ไหน ก็จะเป็นองค์ประกอบของโปรเจกต์นั้นๆ ดังรูป



รูปที่ 1.9 แสดง Project Explorer แบบโปรเจกต์เดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.10 แสดง Project Explorer แบบหลายโปรเจกต์

ในโปรเจกต์หนึ่งจะประกอบไปด้วยอย่างน้อยที่สุด 1 ฟอร์ม (ยกเว้นกรณีที่คุณสร้างแอปพลิเคชันชนิด *.dll ไม่จำเป็นจะต้องมีฟอร์ม) ถ้าคุณต้องการเพิ่มฟอร์มเข้าไปในโปรเจกต์ใด ให้ทำดังนี้



Note

- เลื่อนเมาส์ไปบริเวณหน้าต่าง Project Explorer
- คลิกขวาเพื่อแสดง pop-up เมนู
- เลือกคำสั่ง Add/Form

อาจเลือกจากเมนู Project แล้วเลือกรายการที่ต้องการเพิ่มก็ได้ ซึ่งจะเห็นได้ว่าคุณสามารถเพิ่มรายการอื่นๆ เข้ามาในโปรเจกต์ได้อีกหลายชนิด ซึ่งผู้เขียนจะกล่าวถึงในหัวข้อต่อไป

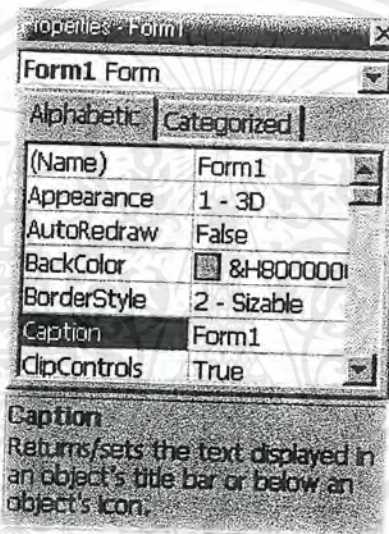
สำหรับรายการชนิดของอ็อบเจกต์ที่ปรากฏอยู่ในหน้าต่าง Project Explorer นี้มีหลายชนิด นี่เป็นบางส่วนของผู้เขียนคิดว่าเป็นสิ่งจำเป็น ในการเริ่มต้นสร้างแอปพลิเคชันหนึ่งๆ ให้มีความสมบูรณ์ในระดับหนึ่งได้

Project(n)	แสดงแอปพลิเคชันที่คุณพัฒนาอยู่ อาจมีโปรเจกต์เดียวหรือหลายโปรเจกต์ก็ได้ โดยปกติจะมีนามสกุล .vbg ถ้ามีหลายโปรเจกต์จะมีนามสกุล .vbg
Form(n)	เป็นรายการฟอร์มที่มีอยู่ในโปรเจกต์นั้นๆ ใน 1 โปรเจกต์อาจมีมากกว่า 1 ฟอร์มก็ได้ จะมีนามสกุล .frm
Modules	เป็นที่เก็บชุดคำสั่งหรือ rutin ที่คุณเขียนขึ้นมาและคุณสามารถเซฟเก็บไว้ใช้ในภายหลังได้อีกด้วย โปรแกรมเมอร์มักจะเก็บคำสั่งหรือ โปรซีเคอร์ที่ใช้บ่อยๆ ไว้เป็น โมดูล ซึ่งจะมีนามสกุล .bas
Class modules	เป็น โมดูลชนิดพิเศษที่มีลักษณะเป็นอ็อบเจกต์ที่คุณสามารถสร้างขึ้นมาได้ จะมีนามสกุล .cls
User controls	คือ คอนโทรล ActiveX ที่คุณสร้างขึ้นมา มีนามสกุล .ctl

ยังมีชนิดของอ็อบเจ็กต์อื่นๆอีกที่สามารถเป็นส่วนหนึ่งของโปรเจ็กต์ได้ ซึ่งคุณจะศึกษาในบทต่อไป รายการอ็อบเจ็กต์ที่กล่าวมาแล้ว ถือได้ว่าเป็นองค์ประกอบหลักที่จะทำให้แอปพลิเคชันของคุณมีความเป็นมาตรฐานและมีประสิทธิภาพ คั่งเช่นแอปพลิเคชัน ที่ได้รับความนิยมทั่วไป ในท้องตลาด

หน้าต่างคุณสมบัติ (Properties Window)

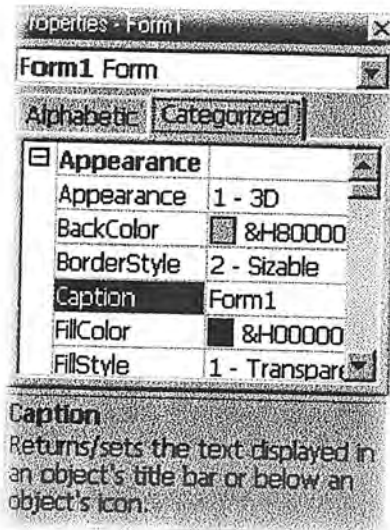
หน้าต่างคุณสมบัตินี้ จะมีความสัมพันธ์กับฟอร์ม, คอนโทรล หรืออ็อบเจ็กต์ใดๆ ที่มีสถานะเอกทิฟ (active) หรือได้รับความสนใจ (focus) อยู่ โดยที่ VB จะเปลี่ยนแปลงหน้าต่างคุณสมบัติให้ตรงกับอ็อบเจ็กต์ โดยอัตโนมัติ ซึ่งในตอนนี้คุณสามารถอ่านค่า, ปรับแต่งค่าเริ่มต้น (initial) ของคอนโทรล หรืออ็อบเจ็กต์ใดๆ ได้จากหน้าต่างนี้ ถือได้ว่าเป็นหัวใจหลักอันหนึ่งของการ programming ด้วย VB ที่ช่วยอำนวยความสะดวกให้คุณ ในการเขียนโค้ดเป็นอย่างยิ่ง



รูปที่ 1-11 แสดงหน้าต่างคุณสมบัติแบบ Alphabetic

โดยปกติแล้ว ไม่ว่าจะใช้อ็อบเจ็กต์ หรือคอนโทรลใดๆ VB จะตั้งค่าปกติ (default) ให้คุณโดยอัตโนมัติ ซึ่งจะมีทั้งส่วนที่เหมาะสมอยู่แล้ว และส่วนที่ VB ตั้งค่ากลางๆ เอาไว้ จะมีทั้งที่คุณต้องปรับแต่งและไม่ต้องยุ่งเกี่ยว คุณสมบัติบางตัวเท่านั้น ที่คุณควรปรับแต่งค่าเริ่มต้น ให้ตรงกับความต้องการของคุณ ในหน้าต่างคุณสมบัตินี้ อีกส่วนหนึ่งจะใช้วิธีการเขียนโค้ด เพื่อปรับแต่งค่าคุณสมบัติ ซึ่งเป็นเทคนิคที่ไม่ใครชอบพท์แนะนำให้ใช้ ในหน้าต่างคุณสมบัติ จะประกอบไปด้วยแท็บ 2 แท็บคือ

1. แท็บ Alphabetic เป็นแท็บที่แสดงรายการคุณสมบัติ เรียงตามตัวอักษร ในภาษาอังกฤษ
2. แท็บ Categorized เป็นแท็บที่แสดงรายการคุณสมบัติ โดยการจัดกลุ่มของคุณสมบัติที่มีหน้าที่คล้ายกัน หรือมีความสัมพันธ์กัน

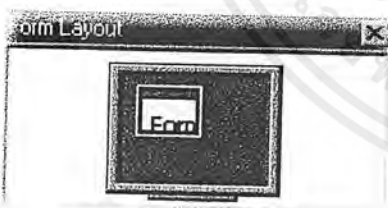


รูปที่ 1-12 แสดงหน้าต่างคุณสมบัติแบบ Categorized

คุณสามารถเลือกรายการคอนโทรลได้จากหน้าต่างคุณสมบัติเช่นกัน โดยคลิกที่ drop-down list box มันจะแสดงรายการคอนโทรลที่คุณใช้งานอยู่ และคุณยังสามารถทราบความหมายของคุณสมบัติที่คุณเลือกโดยคำนิยามสั้นๆ ที่อยู่ด้านล่าง

หน้าต่าง Form Layout

หน้าต่าง Form Layout ใช้สำหรับกำหนดตำแหน่งของฟอร์ม ที่จะปรากฏบนจอภาพในขณะที่รัน คุณสามารถกำหนดตำแหน่งของฟอร์ม โดยการเคลื่อนย้ายฟอร์มจำลอง ที่อยู่ในจอภาพจำลอง ด้วยการ drag เมาส์ ไปยังตำแหน่งที่คุณต้องการ (เมาส์เปลี่ยนเป็นรูปร่าง ลูกศร 4 ทาง) มีข้อสังเกตคือ เมื่อคุณเคลื่อนย้ายฟอร์มจำลองแล้ว ฟอร์มจริงไม่ได้เคลื่อนย้ายไปยังตำแหน่งดังกล่าวแต่อย่างใด เพราะว่ามันจะมีผลในขณะที่รันเท่านั้น



รูปที่ 1-13 แสดงหน้าต่าง Form Layout

หน้าต่างแสดงผลทันที (Immediate Window)

เป็นหน้าต่างที่ให้ประโยชน์ ในกรณีที่คุณต้องการทราบผล การประมวลผลโดยทันที เช่น การทดสอบโปรแกรมเมอร์ต่างๆ เป็นต้น เมื่อคุณสั่งรัน โปรแกรม หน้าต่างนี้จะปรากฏขึ้น โดยอัตโนมัติ คุณสามารถแสดงหน้าต่างนี้ โดยการเลือกที่เมนู View/Immediate Window เช่นกัน

□□□□□□!



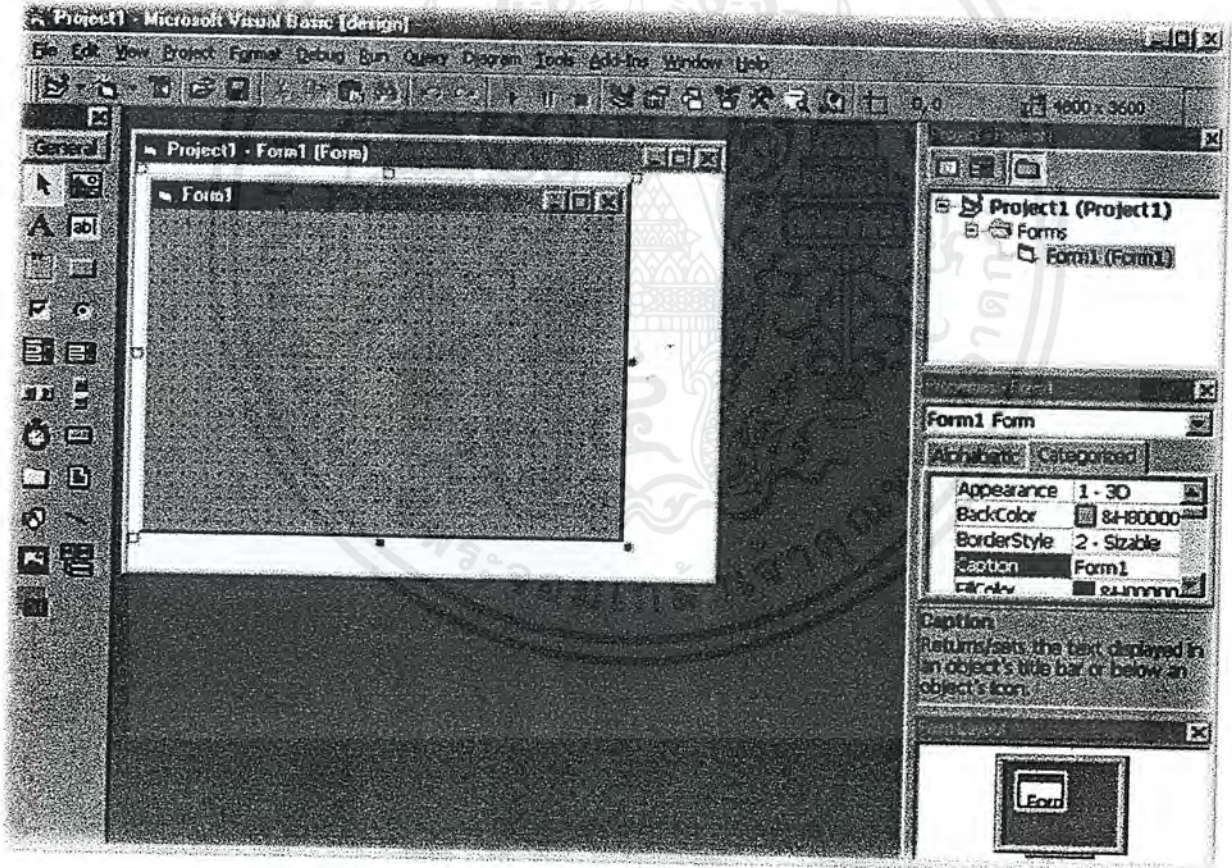
รูปที่ 1-14 แสดงหน้าต่าง Immediate

สภาพแวดล้อมโดยรวมของ VBIDE

สภาพแวดล้อมของ VBIDE สามารถแบ่งได้ 2 ประเภทดังนี้

1. โหมด MDI (Multiple Document Interface) จะแสดงหน้าต่างในรูปแบบเป็นหนึ่งเดียว ซึ่งเป็นสภาพแวดล้อมปกติของ VB ดังรูป

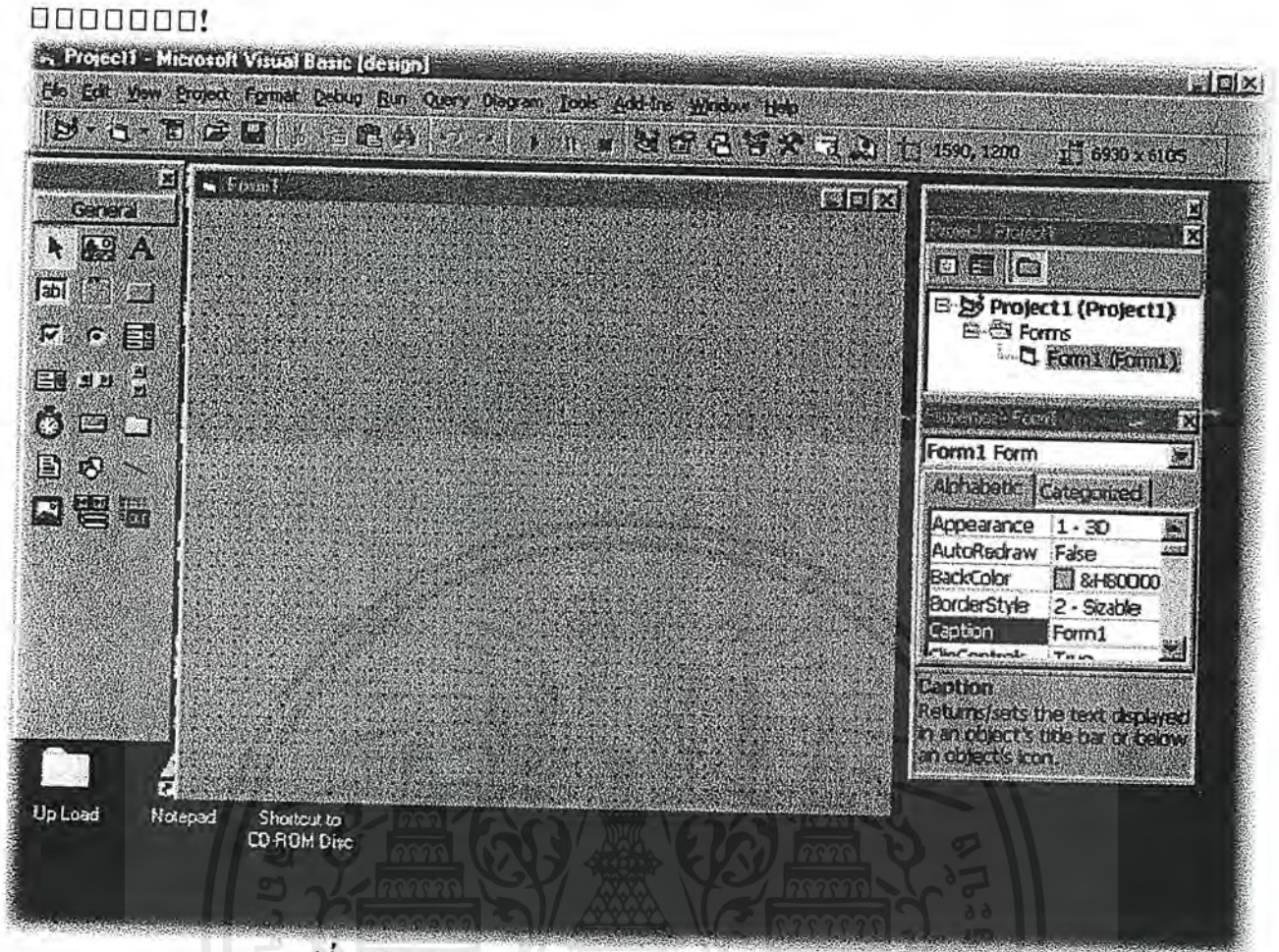
□□□□□□!



รูปที่ 1-15 แสดงสภาพแวดล้อมของ VBIDE แบบ MDI

2. โหมด SDI (Single Document Interface) จะแสดงหน้าต่างที่มีลักษณะเป็นอิสระต่อกัน แต่ยังคงมีความสัมพันธ์กันเหมือน โหมด MDI แต่จะใช้พัฒนาแอปพลิเคชันอีกชนิดหนึ่ง ดังรูป

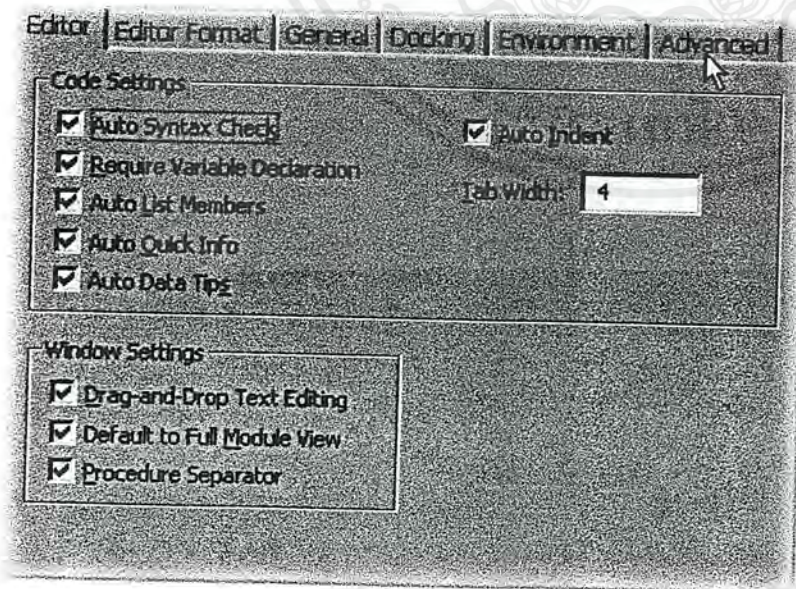
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1-16 แสดงสภาพแวดล้อมของ VBIDE แบบ SDI

การเปลี่ยนสภาพแวดล้อมจาก MDI เป็น SDI

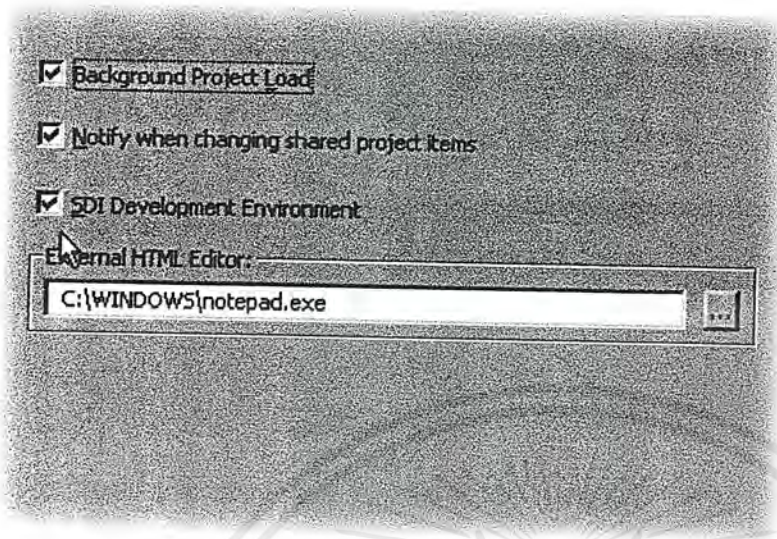
1. เลือกเมนู Tools/Options จะปรากฏไดอะล็อกบ็อกซ์ Options ดังรูป



รูปที่ 1-17 แสดงไดอะล็อกบ็อกซ์ Options

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.คลิกที่แท็บ Advanced เลือก SDI Development Environment แล้วคลิก VBIDE จะเปลี่ยนเป็นแบบ SDI ในการรัน VB ครั้งต่อไป



รูปที่ 1-18 แสดงแท็บ Advanced

การปรับแต่งสภาพแวดล้อมให้ตรงกับความต้องการ

จากหัวข้อที่ผ่านมา ผู้เขียน ได้อธิบายการใช้งานส่วนต่างๆ ของ VBIDE ซึ่งในการใช้งานจริงๆ แล้ว คุณไม่จำเป็นต้องแสดงผลทุกอย่าง หน้าต่าง พร้อมๆ กัน เพราะจะทำให้พื้นที่ในการทำงานของคุณมีน้อย เมื่อคุณพัฒนาแอปพลิเคชันไปได้ระยะหนึ่ง คุณจะพบว่า สภาพแวดล้อมแบบใด เหมาะสมกับโปรเจกต์ใด และควรจะแสดงหน้าต่างอะไรบ้าง ในเวลาหนึ่งๆ ผู้เขียนจะพยายามปิดหน้าต่าง ที่ไม่ได้ใช้งานเสมอ ผู้เขียนถือว่า ถ้าคุณจัดโต๊ะทำงานของคุณ ให้เป็นระเบียบ เรียบร้อย เก็บสิ่งที่ยังไม่ได้ใช้งานไว้ในลิ้นชัก เอาสิ่งที่ต้องการใช้งานวางไว้บน โต๊ะ มันจะส่งผลให้การพัฒนาแอปพลิเคชันของคุณ สะดวก และรวดเร็ว ในระดับหนึ่งอีกด้วย

ในบทต่อไป จะเริ่มเข้าสู่การเขียนแอปพลิเคชันอย่างแท้จริง ต่อไปนี้จะเป็นแนวทางการนำเสนอของผู้เขียน ที่จะใช้เป็นรูปแบบต่างๆ ซึ่งมีรายละเอียด ดังนี้

1. ถ้าเป็นส่วนของ โคลด์ หรือตัวอย่าง source code ของ โปรแกรม ผู้เขียนจะใช้ฟอนต์ที่มีลักษณะแตกต่างจากฟอนต์ปกติ และใช้การเว้นวรรค คำตัดคำ ให้ตรงกับไวยากรณ์ที่ VB ซอมรับ เช่น

```
Private Sub Command1_Click ()  
    Text1.Text = "hello world"  
End Sub
```

2. ถ้าเป็นรูปแบบไวยากรณ์ จะใช้ตัวเข้มเพื่อแทนรูปแบบ (format), คำสั่ง (statements), ฟังก์ชัน (functions), คำสงวน (keyword) หรืออื่นๆ ให้ตรงกับรูปแบบการนำเสนอของ msdn Library เพื่อให้คุณมีความคุ้นเคย และจะใช้ตัวเอน เพื่อแสดงส่วนของตัวแปร, พารามิเตอร์, อาร์กิวเมนต์ หรืออื่นๆ ซึ่งคุณสามารถเปลี่ยนแปลงได้ เช่น

```
Form1.Enabled [= boolean]
```

3. ส่วนตัววงเล็บ [] คุณไม่ต้องพิมพ์ลงไป และตัวแปรที่อยู่ในวงเล็บหมายถึง คุณจะใส่ หรือไม่ใส่ก็ได้ ขึ้นอยู่กับการใช้งานของคุณ เช่น

```
{Dim|Private} varname As datatypes
```

4. ส่วนตัววงเล็บ { } คุณไม่ต้องพิมพ์ลงไปเช่นกัน และตัวแปรที่อยู่ในวงเล็บดังกล่าว จะถูกค้นหาคีย์เครื่องหมาย | หมายถึง คุณจะต้องใส่ ตัวแปรตัวใดตัวหนึ่งในวงเล็บ { } ดังกล่าว เช่น

```
Private varname As datatypes หรือ
```

```
Dim varname As datatypes
```

5. สำหรับค่าของตัวแปร ถ้ามีคำว่า default หมายถึง เป็นค่าที่ VB ตั้งไว้ให้คุณแล้ว คุณจะเปลี่ยนหรือไม่ก็ได้, ถ้ามีคำว่า Optional หมายถึง เป็นข้อกำหนดเพิ่มเติมของการใช้งานตัวแปรนั้นๆ คุณจะใส่หรือไม่ใส่ก็ได้

6. ในส่วนของ คุณสมบัติ (properties), เมธอด (methods), และค่าคงที่ (constants) รวมทั้งที่ผู้เขียนจะระบุเฉพาะเจาะจง ผู้เขียนจะไม่อธิบายทั้งหมด เพราะว่าคุณสามารถค้นหาเพิ่มเติมได้จาก msdn Library จุดประสงค์ของผู้เขียนก็คือ จะเน้นการนำไปใช้งานให้มากที่สุด โดยแสดงตัวอย่างประกอบ เพื่อให้ผู้อ่านเห็นภาพ ได้อย่างชัดเจนมากที่สุด แต่ยังไม่ถึง concept ที่เป็นพื้นฐานในการใช้งาน และส่วนที่ผู้เขียนละไว้ เนื่องจากว่า คุณสามารถทำความเข้าใจได้ไม่ยาก ซึ่งจะระบุหัวข้อใน msdn Library ที่มีความเกี่ยวข้องกับหัวข้อนั้นๆ เสมอ เพื่อให้คุณผู้อ่านรับข้อมูลต่างๆ ครบถ้วนอย่างมีระบบ

7. สำหรับในส่วนของเนื้อหาใดที่มีความสัมพันธ์กัน ในลักษณะข้ามบท เนื้อหาที่อยู่ในบทหลัง จะมีการระบุคำใช้งานมากกว่า ซึ่งผู้เขียนจะระบุทุกครั้ง ถึงหัวข้อที่เกี่ยวข้อง และเนื้อหาในทุกๆ บท คุณไม่จำเป็นต้องอ่านเรียงบทก็ได้ คุณสามารถค้นหาหัวข้อที่คุณต้องการ ได้จากสารบัญ

8. เนื้อหาที่เกิดขึ้นในบทต่างๆ เป็นเพียงสิ่งที่ผู้เขียน ได้มาจากการศึกษา ค้นคว้า มิได้เป็นการคัดลอก หรือแปลมาจากส่วนใด โดยใช้ msdn Library ของ VB 6.0+SP2 เป็นแกนหลักในเนื้อหา และส่วนที่ผู้เขียนจะระบุอ้างอิงเพิ่มเติม

9. สำหรับชื่อคอนโทรล หรืออ็อบเจกต์ต่างๆ ผู้เขียนจะใช้ชื่อ คลาส (class) ของคอนโทรล หรืออ็อบเจกต์นั้นๆ ในการอ้างอิง ซึ่งจะทำให้ผู้อ่าน มีความคุ้นเคยกับคอนโทรล หรืออ็อบเจกต์ของ VB ในลักษณะคลาสมากขึ้นอีกด้วย

10. คำนิยาม หมายถึง คำหรือข้อความต่างๆ ที่ผู้เขียนจะใช้สำหรับกล่าวอ้างอิงในภายหลัง ซึ่งจะมีความหมายดังนี้

10.1 แอปพลิเคชัน (application) หมายถึง คำที่ผู้เขียนใช้แทน โปรแกรมต่างๆ ไปในท้องตลาด ซึ่งอาจรวมถึง โปรแกรมที่รันภายใต้ระบบปฏิบัติการ Dos หรือ Windows ก็ได้

10.2 อาร์กิวเมนต์ (argument) หมายถึง ตัวแปร, ค่าคงที่ หรือพารามิเตอร์ ซึ่งใช้สำหรับส่งค่าเข้าไปในฟังก์ชัน หรือส่งค่าเข้าไปในโพรซีเจอร์

10.3 คอมไพล์ (compile) หมายถึง การแปลง โปรเจกต์ ให้เป็นแอปพลิเคชันที่สมบูรณ์ ซึ่งโดยปกติแล้วจะมีนามสกุล .exe

10.4 ดีไซน์ไทม์ (design time) หมายถึง ช่วงเวลาที่โปรแกรมเมอร์ออกแบบอินเทอร์เฟซ หรือเขียนโค้ด (coding) อาจใช้คำว่า ขณะออกแบบ ก็ได้

10.5 คำสงวน (keyword) หมายถึง คำหรือข้อความ ที่ Visual Basic นำไปใช้เป็นชุดคำสั่ง, คอนโทรล, อ็อบเจกต์, ฟังก์ชันหรืออื่นๆ ซึ่งคำหรือข้อความเหล่านี้ คุณไม่สามารถนำไปใช้ตั้งชื่อ หรืออื่นใดที่ไม่ใช่หน้าที่ หรือไวยากรณ์ของมัน

10.6 โปรเจกต์ (project) หมายถึง คำที่ใช้เรียกแทน โปรแกรม หรือ source code ที่ยังไม่ได้ออก compile

10.7 รันไทม์ (run time) หมายถึง ช่วงเวลาที่โปรแกรมเมอร์กำลังทดสอบโค้ด หรือกำลังรันโปรเจกต์อยู่ เป็นช่วงเวลาที่ไม่สามารถแก้ไขโค้ด โดยการกดปุ่ม F5 หรือกดปุ่มรันบนทูลบาร์

การใช้งาน Visual Basic

ในบทที่ 1 ผู้เขียนได้อธิบายเกี่ยวกับการใช้งาน VBIDE ขั้นพื้นฐาน ซึ่งมีความสำคัญเป็นอย่างยิ่งสำหรับผู้ที่ไม่เคยใช้งานตัว VB มาก่อนเลย สำหรับในหัวข้อนี้ จะอธิบายเกี่ยวกับรูปแบบ และแนวทางในการพัฒนาแอปพลิเคชันด้วย VB เป็นที่ทราบกันว่า concept หลักของการพัฒนาแอปพลิเคชันด้วย VB ก็คือการสร้างองค์ประกอบต่างๆของแอปพลิเคชันด้วยคอนโทรล โดยมีรูปแบบที่สื่อด้วยภาพ หรือที่เรียกกันติดปากว่า การออกแบบอินเตอร์เฟซ ต่อมาก็คือการเขียนโค้ดเพื่อรองรับเหตุการณ์ต่างๆ ที่อาจเกิดขึ้นได้ ซึ่งผู้เขียนจะแบ่งออกเป็น 5 ขั้นตอนใหญ่ๆ คือ

- 1.เลือกพัฒนาชนิดของแอปพลิเคชัน
- 2.สร้างยูสเซอร์อินเตอร์เฟซ (หรือส่วนติดต่อกับผู้ใช้ อาจเรียกสั้นๆว่า อินเตอร์เฟซ)
- 3.เขียนโค้ดเพื่อรองรับเหตุการณ์ต่างๆ ที่อาจเกิดขึ้นกับแต่ละคอนโทรลหรืออ็อบเจกต์
- 4.การทดสอบ ตรวจสอบ และดีบักข้อผิดพลาด
- 5.compile โปรแกรมให้เป็นแอปพลิเคชันที่สมบูรณ์ (เช่น *.exe หรือ *.dll เป็นต้น)

ทั้ง 5 ขั้นตอนถือได้ว่าเป็นหัวใจหลักในการสร้างแอปพลิเคชันด้วย VB สิ่งที่คุณสามารถสังเกตเห็น โค้ดที่เขียนที่สุดก็คือ ผลงานของคุณจะแสดงออกมา ได้รวดเร็วมาก โดยใช้งานคอนโทรลไม่กี่ตัว หัวข้อนี้จะเป็นแนวทางที่อธิบายรูปแบบการสร้างอินเตอร์เฟซ เพราะไม่ว่าคุณ จะพัฒนาแอปพลิเคชันชนิดใดๆ แอปพลิเคชันของคุณก็จะต้องมีมาตรฐานตามข้อกำหนดของแอปพลิเคชัน ที่รันภายใต้ Windows 95 ที่ไม่โครซอฟท์กำหนดไว้ เช่น สนับสนุนการทำงานแบบ 32 บิต, สนับสนุนการประมวลผลแบบ multitasking เป็นต้น ซึ่ง VB สนับสนุนอยู่แล้ว แต่รูปแบบอินเตอร์เฟซต่างหาก ที่คุณจะต้องสร้างสรรค์ออกมา ให้มีมาตรฐานเดียวกับแอปพลิเคชัน ที่ได้รับความนิยมในด้านนั้นๆ เช่น ถ้าคุณต้องการพัฒนาแอปพลิเคชันด้านฐานข้อมูล คุณจะต้องเชื่อม โยง เข้ากับชนิดของฐานข้อมูลต่างๆ ที่มีอยู่ไม่ว่าจะเป็น Access, Foxpro หรือ Oracle เป็นต้น สำหรับแนวทางที่ง่ายและรวดเร็วที่สุด ที่จะทำให้คุณสามารถ ออกแบบอินเตอร์เฟซที่เป็นมาตรฐานได้ก็คือ คุณสามารถศึกษารูปแบบและแนวทาง จากแอปพลิเคชันอื่นๆ ที่ได้รับความนิยมศึกษาข้อดี ข้อเสีย ของแอปพลิเคชันนั้นๆ มันจะช่วยช้่นระยะเวลาไปได้มากทีเดียว

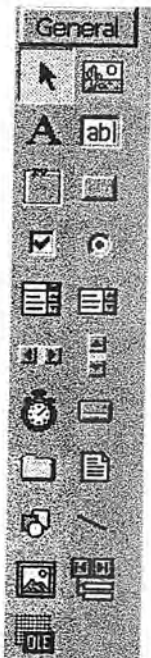
สำหรับในหัวข้อนี้ ผู้เขียนจะกล่าวถึงขั้นตอนการสร้างอินเตอร์เฟซ และการเขียนโค้ดจาก 5 ขั้นตอนดังกล่าวเท่านั้น ส่วนหัวข้อที่เหลือ ผู้เขียนจะกล่าวถึงในบทต่อไป

การใช้งานคอนโทรลในการสร้างอินเตอร์เฟซ

จุดเริ่มต้นของการพัฒนาแอปพลิเคชันด้วย VB ก็คือการนำคอนโทรลชนิดต่างๆ ที่ VB จัดเตรียมไว้ นำมาวางคอนโทรล แอปพลิเคชันของคุณจะมีหน้าตา เป็นอย่างไร ก็อยู่ในขั้นตอนนี้ ถ้าคุณเริ่มต้นพัฒนาแอปพลิเคชันได้ดี โดยการออกแบบอินเตอร์เฟซที่ใช้งานง่าย เป็นมิตรกับผู้ใช้ มันจะส่งผลให้ระยะเวลา ในการพัฒนาแอปพลิเคชันหนึ่งลดลงไปได้มากทีเดียว เพราะสิ่งที่เหลืออยู่คือ การเขียนโค้ดเพื่อทำให้

โปรเจกต์ของคุณทำงานให้สมบูรณ์มากที่สุด ผลงานของคุณก็จะออกมาเหมาะสมกับเวลาที่คุณต้องเสียไป :-)

คอนโทรลแต่ละชนิดจะมีหน้าที่ หรือจุดประสงค์ในการนำไปใช้งานต่างกัน เช่น คอนโทรล CommandButton ใช้สร้างปุ่มกดเพื่อตอบรับ, คอนโทรล OptionButton ใช้สำหรับให้ผู้ใช้มีทางเลือก, คอนโทรล TextBox ใช้รับและแสดงข้อมูลที่ผู้ใช้สามารถแก้ไขได้, คอนโทรล Label ใช้แสดงข้อมูล ที่ผู้ใช้ไม่สามารถแก้ไขได้ เป็นต้น



สำหรับชื่อของคอนโทรลแต่ละตัว คุณสามารถเลื่อนเมาส์ไปบริเวณบนตัวคอนโทรลนั้นๆ แล้วจะมี Tooltip แสดงชื่อของคอนโทรลนั้นๆ ให้ผู้ใช้ทราบ คุณจะได้พบกับ Tooltip ในทุกๆ แอปพลิเคชันที่รันอยู่ภายใต้ Windows 98/95 และ NT ซึ่งถือได้ว่าเป็นมาตรฐานไปแล้ว และคุณก็สามารถทำให้แอปพลิเคชันของคุณมี Tooltip ได้ด้วย VB เช่นกัน

รูปที่ 2.1 แสดงกลุ่มคอนโทรลมาตรฐานของ VB

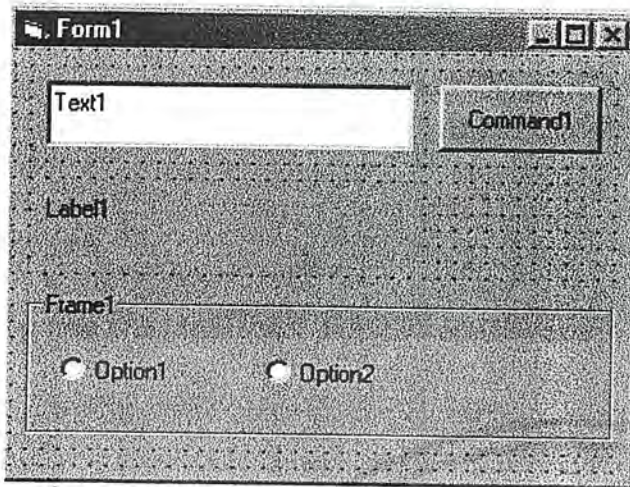
รูปแบบการปรากฏของคอนโทรลแต่ละชนิดบนฟอร์ม

คอนโทรลแต่ละชนิดมีรูปแบบการปรากฏตัว มีความแตกต่างกัน คอนโทรลบางตัวเมื่อคุณนำมาใช้งานอาจจะไม่มีการปรากฏตัวขึ้นมา หรืออาจจะปรากฏขึ้นมาเลยก็ได้ ซึ่งจะขึ้นอยู่กับหน้าที่ของแต่ละคอนโทรล สำหรับคอนโทรลที่คุณจะต้องใช้งานในทุกๆ แอปพลิเคชันจะมี เช่น CommandButton, TextBox, CheckBox, OptionButton, ScrollBar เป็นต้น คอนโทรลพวกนี้ถือได้ว่าเป็น คอนโทรลคลาสสิก หมายถึง เป็นคอนโทรลที่คุณพบเห็น ได้ทั่วไป จะเห็นได้ว่าถ้าคุณสามารถนำไปใช้งานได้ดั่งเดิมแล้ว แอปพลิเคชันของคุณ ก็จะมีประสิทธิภาพเป็นอย่างยิ่ง

รูปแบบการปรากฏตัวของคอนโทรลสามารถแยกออกได้เป็น 2 แบบ คือ

1. ปรากฏตัวทันทีเมื่อคุณนำมาใช้งาน หมายถึง คุณสามารถเห็น ได้ในขณะที่คุณออกแบบ ที่เรียกว่า design time ซึ่งคอนโทรลส่วนใหญ่จะมีลักษณะเช่นนี้
2. ไม่ปรากฏตัวเมื่อคุณนำมาใช้งาน หมายถึง คอนโทรลประเภทที่คุณไม่สามารถมองเห็นการ

ทำงานของตัวมัน ได้ในขณะที่ออกแบบ มันจะทำงานอยู่เบื้องหลัง แอปพลิเคชันของคุณ ซึ่งจะเป็นเวลาในช่วงของการรันแอปพลิเคชัน ที่เรียกว่า run time เช่น คอนโทรล Timer, CommonDialog เป็นต้น



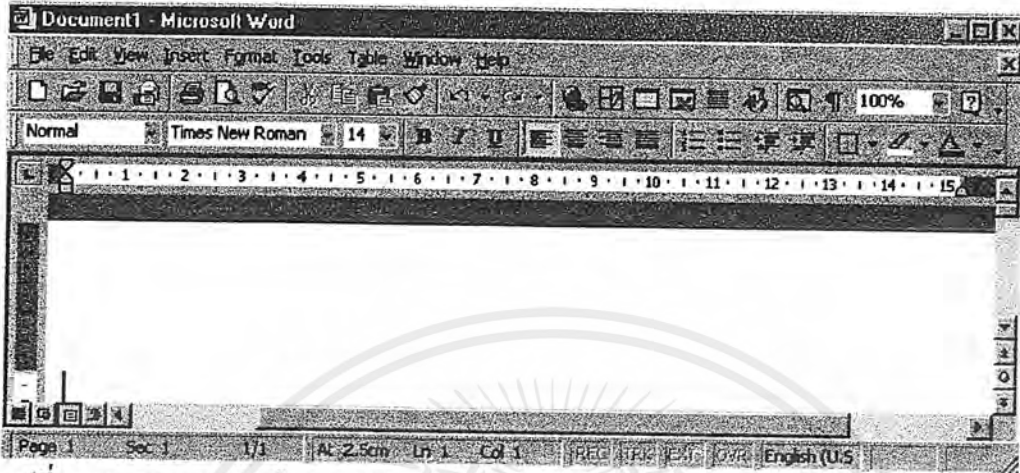
รูปที่ 2.2 แสดงรูปแบบของคอนโทรลหลายๆชนิดบนฟอร์ม

แนวทางในการออกแบบอินเตอร์เฟซแรกของคุณ

สำหรับการออกแบบอินเตอร์เฟซของคุณ VB อนุญาตให้คุณสามารถที่จะวาดคอนโทรลได้อย่างอิสระ ไม่ถูกเกณฑ์ตายตัว ณ ตำแหน่งใดก็ได้ ที่อยู่ในบริเวณฟอร์ม คุณสามารถสร้างสรรค์ให้แอปพลิเคชันของคุณ มีหน้าตาเป็นอย่างไรก็ได้ แต่ขอให้คุณยึดถือไว้ว่า อินเตอร์เฟซที่ออกมา จะต้องเป็นมิตรกับผู้ใช้ มีรูปแบบเดียวกับแอปพลิเคชันทั่วไป ไม่ทำให้ผู้ใช้สับสนกับแอปพลิเคชันของคุณ ยกตัวอย่างเช่น ถ้าคุณออกแบบอินเตอร์เฟซให้ผู้ใช้ ต้องคอยคำถามมากมาย กว่าจะได้เซฟงาน, มีตัวเลือกมากจนกระทั่ง ผู้ใช้ไม่รู้ว่า จะเลือกอะไรดี, จัควาง Lay out ไม่เป็นระเบียบ ไม่เป็นกลุ่ม, ทำให้ผู้ใช้หลงทางในการใช้งาน สิ่งเหล่านี้ถ้าเป็นคุณพื้บเจอเอง เชื่อ ได้ว่าคุณจะเกิดความรู้สึก ไม่อยากใช้งานแอปพลิเคชันนั้นอีกเลย แอปพลิเคชันทั่วไป คุณจะเห็น ได้อย่างหนึ่งว่า จะพยายามเน้นที่การใช้งานง่าย รวดเร็ว และมีประสิทธิภาพ ซึ่งเป็นการอำนวยความสะดวกให้ผู้ใช้เป็นอันดับแรก แต่การะต่างๆ จะอยู่ที่โปรแกรมเมอร์ต่างหาก :-)

สิ่งที่จะสร้างแนวความคิดให้คุณออกแบบอินเตอร์เฟซได้อย่างรวดเร็วก็คือการที่ตัวคุณเองจะต้องมีประสบการณ์ ในการใช้งานแอปพลิเคชันในด้านต่างๆ ให้มากที่สุด เท่าที่จะเป็นไปได้ แล้วตัวคุณเองจะได้ concept ในการสร้างแอปพลิเคชัน ได้อย่างไม่ยากเย็น ผู้เขียนมักจะใช้แอปพลิเคชันประเภท freeware หรือ shareware ต่างๆ เป็นแนวทางในการออกแบบอินเตอร์เฟซ ในบางครั้งคุณอาจพบว่า แอปพลิเคชันประเภทนี้ยังดีกว่าแอปพลิเคชันที่มีราคาสูงด้วยซ้ำ ข้อสังเกตอีกอย่างหนึ่งคือ ในแอปพลิเคชันทั่วไป เช่น Word, Excel รวมถึงผลิตภัณฑ์ที่ไม่ใช่ของ ไมโครซอฟท์ จะมีส่วนอินเตอร์เฟซที่มีลักษณะ คล้ายคลึงกันมาก ทำให้ผู้ใช้มีความคุ้นเคย สามารถเรียนรู้ ทำความเข้าใจได้ไม่ยากเย็นนัก อย่างน้อยที่สุด ทำให้ผู้ใช้งานมีความรู้สึก ว่า ไม่ต้องเริ่มต้นเรียนรู้ใหม่ มีความสนใจอยากที่จะเรียนรู้ ใช้งานแอปพลิเคชันนั้นๆ มากขึ้น ซึ่งเป็นสิ่งที่โปรแกรมเมอร์ จะต้องใส่ใจออกมากับอินเตอร์เฟซให้ได้

ข้อได้เปรียบที่เห็นได้ชัดเจนมากที่สุดในการพัฒนาแอปพลิเคชันด้วย VB นั่นก็คือ คุณสามารถใช้ระยะเวลาสั้นๆ ในการออกแบบอินเทอร์เฟซ ได้อย่างรวดเร็ว ทำให้คุณสามารถเปลี่ยนแปลง แก้ไขอินเทอร์เฟซ ให้ตรงกับความต้องการของผู้ใช้งาน ได้อย่างง่ายดาย ซึ่งเป็นการ programming ในสมัยใหม่ ซึ่งจะช่วยย่นระยะเวลาในการพัฒนาแอปพลิเคชันได้เป็นอย่างดี

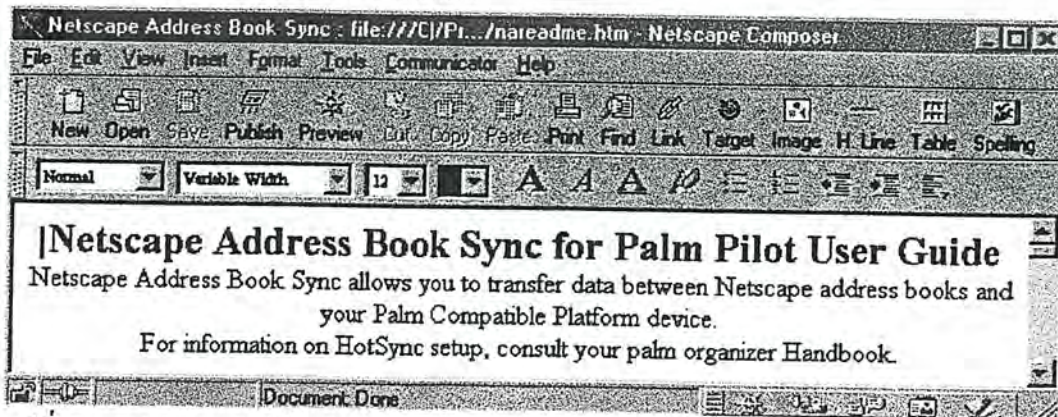


รูปที่ 2.3 แสดงสภาพแวดล้อมของ Word 2000



รูปที่ 2.4 แสดงสภาพแวดล้อมของ Excel 2000

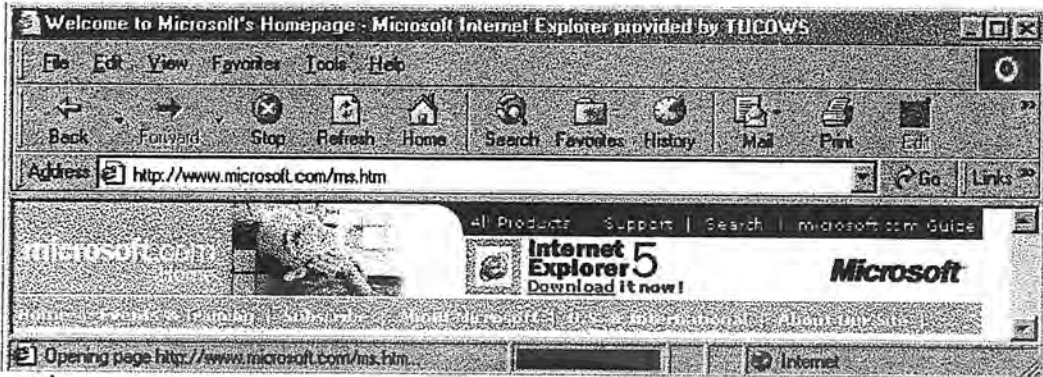
สำหรับแอปพลิเคชันที่ไม่ใช่ผลิตภัณฑ์ของ ไมโครซอฟท์ ก็ยังคงมีอินเทอร์เฟซที่ใกล้เคียงกัน จะแตกต่างกันในแง่ของหน้าที่การใช้งานเท่านั้น



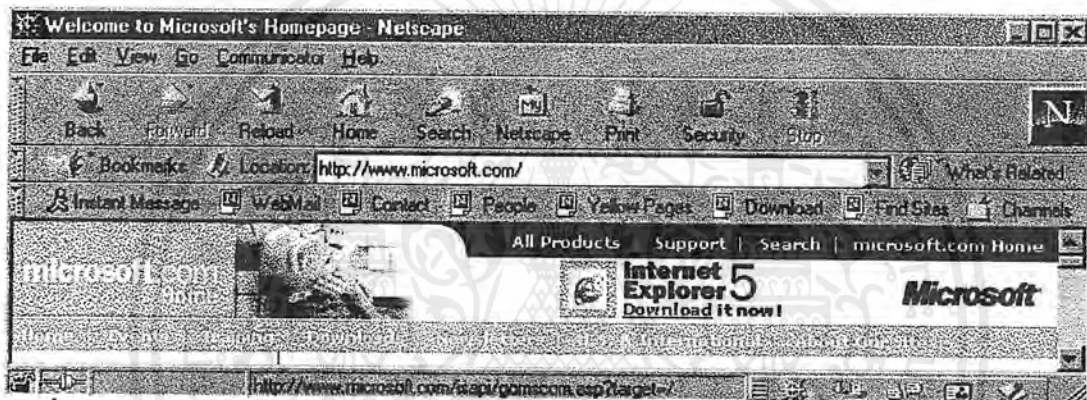
รูปที่ 2.5 แสดงสภาพแวดล้อมของ Netscape Composer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือแม้แต่ browser ขอคนชมก็ยังคงมีอินเตอร์เฟสที่คล้ายคลึงกัน ทำให้คุณคุ้นเคยกับสภาพแวดล้อมเป็นอย่างดี



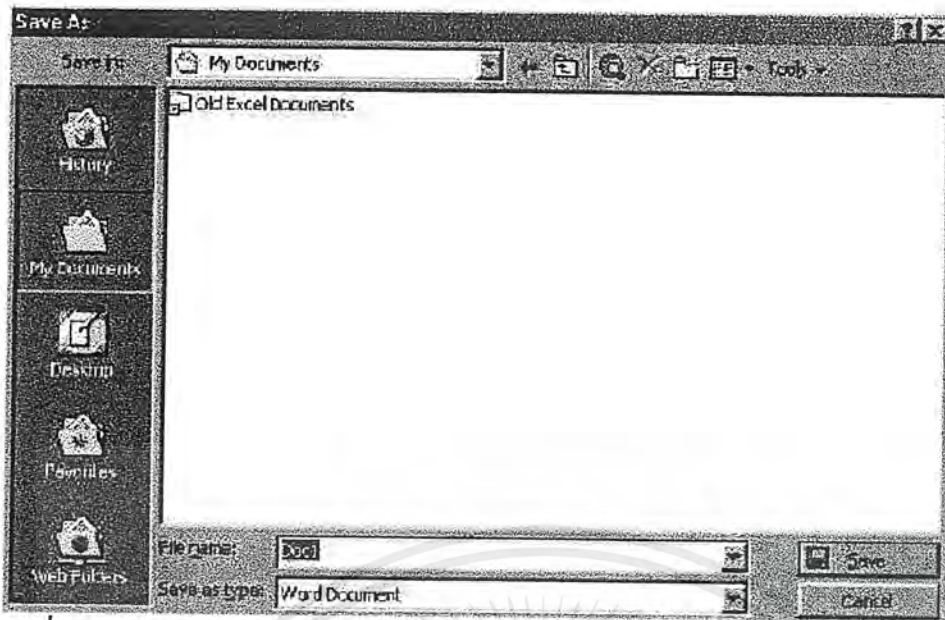
รูปที่ 2.6 แสดงสภาพแวดล้อม Internet Explorer 5.0



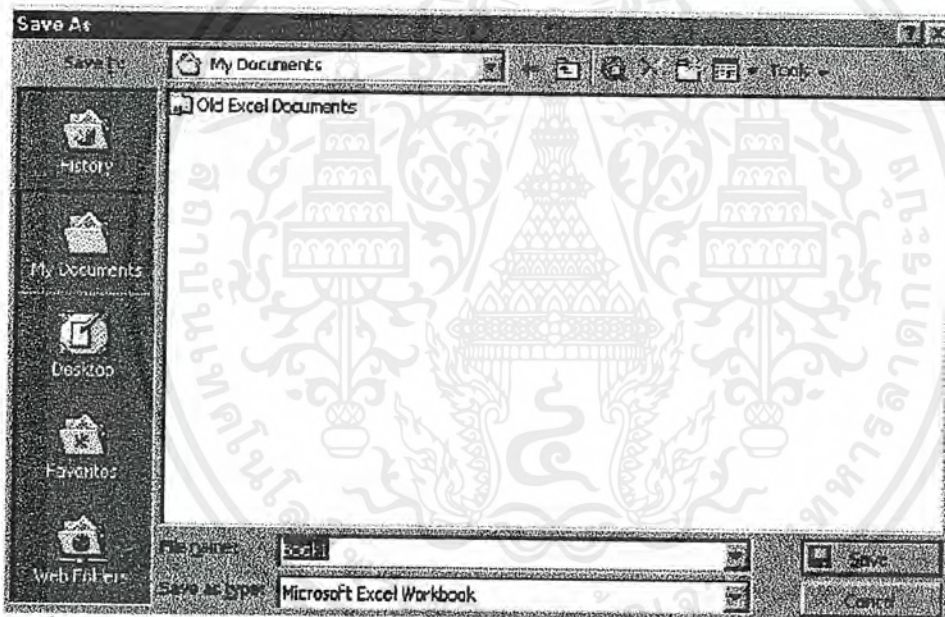
รูปที่ 2.7 แสดงสภาพแวดล้อม Netscape Communicator 4.51

องค์ประกอบหลักๆ ของอินเตอร์เฟส จะประกอบไปด้วย เมนู, ToolBar, dialog box ชนิดต่างๆ เช่น เปิดไฟล์, เซฟไฟล์, พิมพ์งาน เป็นต้น สิ่งเหล่านี้ VB เตรียมไว้ให้คุณแล้ว ที่เหลือก็ขึ้นอยู่กับคุณว่าจะนำไปใช้งานให้เกิดประสิทธิภาพมากเพียงใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 แสดงไดอะล็อกบ็อกซ์เซฟไฟล์ของ Word 2000

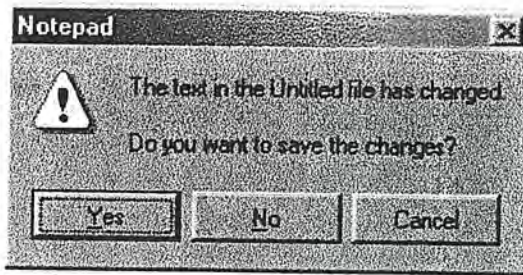


รูปที่ 2.9 แสดงไดอะล็อกบ็อกซ์เซฟไฟล์ของ Excel 2000

จะเห็นว่ามีความคล้ายคลึงกัน เนื่องจากใช้งานกลุ่มฟังก์ชัน Windows API ชุดเดียวกัน มีผลทำให้ผู้ใช้มีความรู้สึกได้ว่า ไม่ต้องเริ่มต้นเรียนรู้ใหม่ทั้งหมด และมีความรู้สึกต้องการใช้แอปพลิเคชันนั้นๆ ค่อนข้าง

ส่วนประกอบของอินเทอร์เฟซ เกิดจากการใช้คอนโทรลชนิดต่างๆ เข้าด้วยกัน เช่น คอนโทรล CommandButton, Frame, CheckBox เป็นต้น แต่คุณไม่จำเป็นต้องออกแบบอินเทอร์เฟซทุกอย่างในแอปพลิเคชันหนึ่งๆ VB ได้เตรียมกลุ่มฟังก์ชัน ที่ให้คุณเขียน โค้ดเพียงบรรทัดเดียว ก็สามารถเรียกใช้งานได้แล้ว เช่น message box ที่เตือนให้ผู้ใช้เซฟงาน ก่อนออกจากโปรแกรม, ไดอะล็อกบ็อกซ์เซฟไฟล์, เปิดไฟล์ เป็นต้น ซึ่งผู้เขียนจะอธิบายในบทต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 แสดง message box ที่เตือนให้ผู้ใช้เซฟงาน ใน VB คุณเขียนโค้ดเพียงบรรทัดเดียวเท่านั้น
การนำคอนโทรลมาใช้งาน

สำหรับวิธีการนำคอนโทรลมาใช้งาน ควบคุมอินเทอร์เน็ตเฟสบนฟอร์ม มี 2 วิธี คือ

1.คลิกที่ตัวคอนโทรลนั้นๆ บน ToolBox แล้วนำไปวางบนฟอร์ม

คุณสามารถกำหนดความกว้างและความยาวของคอนโทรลได้อย่างอิสระรวมถึงกำหนดตำแหน่งได้เช่นกัน VB จะแสดง Tooltip แสดงขนาดของคอนโทรล ให้อัตโนมัติ ส่วนตำแหน่งของคอนโทรลได้จาก บนToolBar โดยแสดงแบบพิกัด co-ordinate วัดจากมุมซ้ายบนของฟอร์มที่บรรจุอยู่ ถ้าคอนโทรลนั้นอยู่ในคอนโทรล Frame ก็จะวัดจากมุมซ้ายบนของคอนโทรล Frame แทน ดังรูป



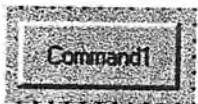
รูปที่ 2.11 แสดง Tooltip บอกขนาดของคอนโทรล



รูปที่ 2-12 แสดงขนาดของคอนโทรลบน Tool Bar

2.ดับเบิลคลิกที่ตัวคอนโทรลนั้นเลย

แล้ว VB จะนำคอนโทรลไปวางบนฟอร์มให้คุณ โดยอัตโนมัติ ซึ่ง VB จะตั้งค่า default ไว้ให้คุณทั้งตำแหน่ง และขนาดของคอนโทรล แล้วคุณค่อยแก้ไขในภายหลัง สำหรับคอนโทรล CommandButton คุณอาจใช้ขนาด default ที่ VB ตั้งมา ไปใช้งานเลยก็ได้ เพราะมีขนาดเหมาะสมอยู่แล้ว



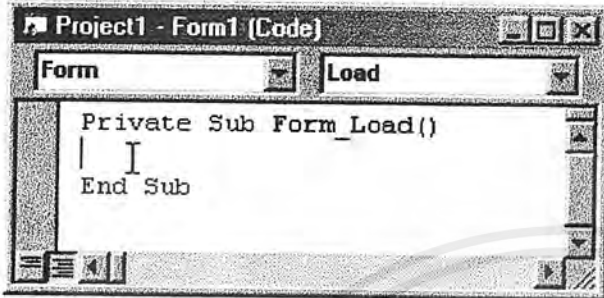
รูปที่ 2.13 แสดงคอนโทรล CommandButton แบบ default

สำหรับการปรับขนาดของคอนโทรล ให้คุณคลิกที่คอนโทรลนั้นๆ เพื่อให้อยู่ในสถานะแอกทีฟ (active) แล้วจะมีปุ่มสี่เหลี่ยมทึบ 8 ปุ่มอยู่บนตัวคอนโทรล ให้คุณเลื่อนเมาส์ไปบริเวณปุ่มทึบ (เคอร์เซอร์เปลี่ยนเป็นลูกศร 2 หัว) คุณก็สามารถปรับขนาดได้ตามต้องการ สำหรับการเลื่อนตำแหน่งของคอนโทรล คุณสามารถ drag เมาส์ บนคอนโทรลนั้นๆ แล้วย้ายไปยังตำแหน่งที่คุณต้องการ ได้โดยตรง ซึ่งก็จะเหมือนกับ การใช้งานแอปพลิเคชันทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พื้นฐานการเขียนโค้ด

สำหรับหัวข้อนี้จะเป็นขั้นตอนที่ต่อจากการออกแบบอินเตอร์เฟซ หลังจากที่คุณได้ออกแบบอินเตอร์เฟซเรียบร้อยแล้ว คุณจะต้องเริ่มเขียนโค้ด สำหรับคอนโทรลแต่ละตัว เพราะเมื่อคุณออกแบบอินเตอร์เฟซแล้ว มันยังไม่สามารถทำอะไรได้เลย ใน VB จะมี editor ไว้สำหรับให้คุณเขียนโค้ดโดยเฉพาะ เรียบไว้ให้คุณแล้ว ดังรูป



รูปที่ 2.14 แสดง editor ของ VB

มี 2 วิธีที่คุณสามารถเรียก editor ขึ้นมาใช้งานคือ

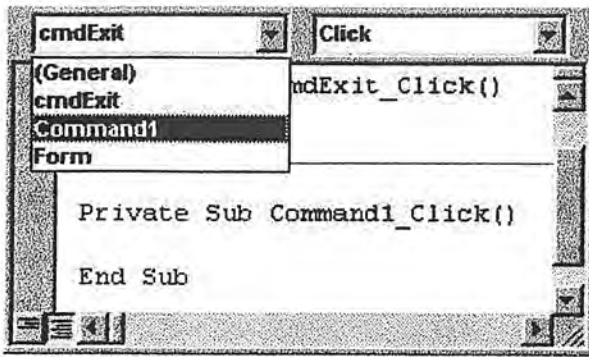
- 1.ดับเบิลคลิกที่ตัวคอนโทรลนั้นๆ
- 2.คลิกที่คอนโทรลนั้น ให้อยู่ในสภาพแอกทีฟ (active) หรือ ได้รับความสนใจ (focus) แล้วกด F7

การใช้งาน editor

เครื่องมือตัวนี้ถือได้ว่าเป็นเครื่องมือที่มีความสำคัญที่สุดอีกตัวหนึ่งในบรรดาเครื่องมือที่ VB มี เพราะใช้สำหรับเขียนโค้ด ให้แอปพลิเคชันของคุณทำงานได้ ซึ่งผู้เขียนจะใช้คำว่า *editor* จะหมายถึง editor ของ VB เครื่องมือตัวนี้คุณต้องใช้งานมากที่สุด ในขบวนการพัฒนาแอปพลิเคชันด้วย VB การศึกษาสภาพแวดล้อมของ editor จึงมีความสำคัญเป็นอย่างยิ่ง ไม่ยิ่งหย่อนไปกว่าส่วนใดๆ ใน VBIDE การทำความเข้าใจ รูปแบบการใช้งาน และความหมายของส่วนต่างๆของ editor จะช่วยลดระยะเวลาในการเขียนโค้ดได้อีกระดับหนึ่ง จากรูปที่ 2.15 สามารถแยกส่วนต่างๆ ของ editor ออกได้เป็น 3 ส่วนดังนี้

1.ส่วน object list box

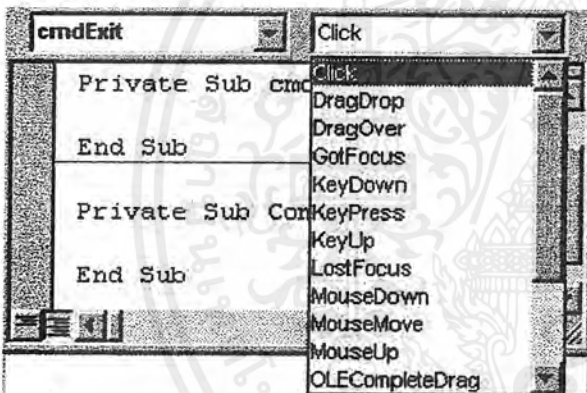
มีหน้าที่แสดงชื่อคอนโทรลหรืออ็อบเจกต์ที่คุณนำมาใช้งาน เมื่อคุณเพิ่มคอนโทรลเข้ามาในฟอร์ม รายชื่อของคอนโทรล จะถูกเพิ่มเข้ามาโดยอัตโนมัติ ถ้าคุณมีการเปลี่ยนแปลงชื่อของคอนโทรล (คุณสมบัตินาม Name) ชื่อที่ปรากฏอยู่ใน object list box ก็จะเปลี่ยนไปตามชื่อที่คุณตั้งไว้เช่นกัน



รูปที่ 2.15 แสดงส่วน object list box และคอนโทรล CommandButton ชื่อปกติและ CommandButton ที่เปลี่ยนชื่อแล้ว

2. ส่วน event list box

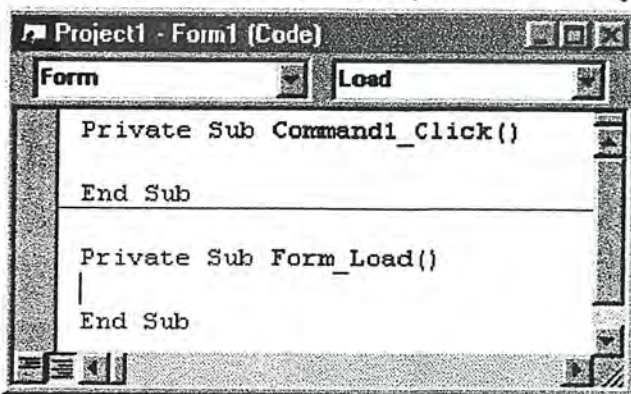
มีหน้าที่แสดงเหตุการณ์ (event) ของคอนโทรลที่ถูกเลือกใน object list box สลับสนุนอยู่ มันจะเปลี่ยนแปลงโดยอัตโนมัติ เมื่อคุณเปลี่ยนชนิดของคอนโทรลใน object list box ทั้ง 2 ส่วนนี้จะมีความสัมพันธ์กันตลอดเวลา



รูปที่ 2.16 แสดงส่วน event list box และรายการ events ที่คอนโทรล CommandButton สลับสนุน

3. ส่วนการเขียนโค้ด

เมื่อคุณเลือกคอนโทรลใน object list box และเลือกเหตุการณ์ใน event list box แล้ว VB จะสร้างโพรซีเจอร์ (procedure) วางๆขึ้นมาให้คุณ โดยอัตโนมัติ ดังรูป



รูปที่ 2.17 แสดงโพรซีเจอร์เหตุการณ์ Click ของคอนโทรล CommandButton

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาดูความหมายในแต่ละคำของโพรซีเจอร์ สำหรับพื้นที่ในการเขียนโค้ดจะอยู่ระหว่าง 2 บรรทัด ดังกล่าว (ที่เคอร์เซอร์กระพริบอยู่)

Private เป็นคำสงวนที่กำหนดขอบเขตของโพรซีเจอร์

Sub เป็นคำสงวนที่บอกชนิดของโพรซีเจอร์ กรณีนี้เป็นแบบซับรูทีน (ในการใช้งานจริงจะมีรูปแบบของโพรซีเจอร์หลายชนิด เช่น โพรซีเจอร์ทั่วไป ฟังก์ชันที่คุณสร้างขึ้นมาเอง ซึ่งผู้เขียน จะกล่าวถึงในบทต่อไป)

Command1 หมายถึงชื่อของอ็อบเจกต์ หรือคอนโทรล ถ้าคุณมีการเปลี่ยนชื่อ (คุณสมบัติ Name) ส่วนนี้จะเปลี่ยนไปเป็นชื่อเดียวกับชื่อที่คุณตั้งไว้

_ เครื่องหมายอันเดอร์สกออร์ ใช้แบ่งชื่อของคอนโทรลและเหตุการณ์ออกจากกัน

Click เหตุการณ์ประจำโพรซีเจอร์ หมายความว่า โพรซีเจอร์นี้จะทำงานเมื่อมีการคลิกปุ่มที่ชื่อว่า **Command1** เท่านั้น

() ในวงเล็บ อาจจะมีรายการตัวแปรที่คุณจำเป็นต้องใช้ในโพรซีเจอร์นี้ ซึ่งเรียกว่า อาร์กิวเมนต์ (arguments) ซึ่งในการใช้งานจริง คุณจะได้พบอีกเช่นกัน :-)

End Sub จบโพรซีเจอร์

โพรซีเจอร์ประจำคอนโทรล

เมื่อคุณนำคอนโทรลชนิดต่างๆ มาใช้งานแล้ว VB จะสร้างโพรซีเจอร์ว่างๆพร้อมเหตุการณ์ประจำคอนโทรลนั้นๆ ขึ้นมาเสมอ โดยที่ VB จะคำนึงถึงหน้าที่ของแต่ละคอนโทรลเป็นหลัก แล้วสร้างโพรซีเจอร์เหตุการณ์ขึ้นมาโดยอัตโนมัติ เช่น คอนโทรล **CommandButton** มีหน้าที่เป็นปุ่มให้ผู้ใช้คลิกตอบรับ (เหตุการณ์ **Click**) VB ก็จะสร้างโพรซีเจอร์ **Private Sub Command1_Click()**, คอนโทรล **TextBox** มีหน้าที่รับหรือแสดงข้อมูลที่ผู้ใช้สามารถเปลี่ยนแปลงได้ VB จะสร้างโพรซีเจอร์ **Private Sub Text1_Change()**, ฟอร์ม เป็นตัวบรรจุคอนโทรลอื่นๆ และจะถูกโหลดเข้ามาในหน่วยความจำก่อนเสมอ VB จะสร้างโพรซีเจอร์ **Private Sub Form_Load()** เป็นต้น โพรซีเจอร์ต่างๆ ที่ VB สร้างขึ้นมา คุณไม่จำเป็นต้องใช้งานทั้งหมด หรือถ้าจะใช้ทั้งหมดก็ได้ คุณสามารถปล่อยให้มันเป็นโพรซีเจอร์ว่างๆได้ โดยไม่ปัญหาในการประมวลผลแต่อย่างใด

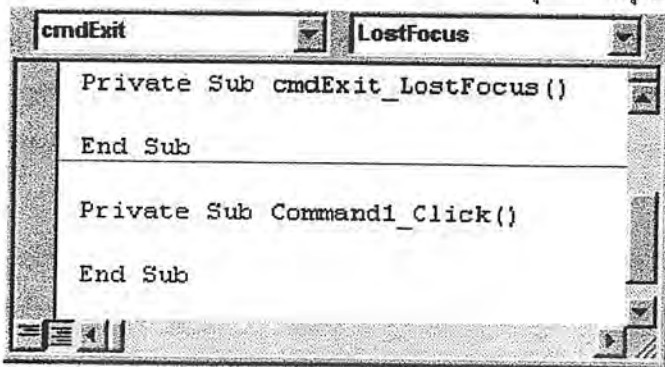
แม้ว่า VB จะสร้างโพรซีเจอร์ประจำคอนโทรลให้คุณโดยอัตโนมัติอยู่แล้ว แต่ในความเป็นจริงโพรซีเจอร์เหล่านี้ ยังไม่สามารถรองรับการทำงาน ของแอปพลิเคชันต่างๆ ได้ทั้งหมด คุณจะต้องเพิ่มเหตุการณ์ต่างๆ ขึ้นมาเพื่อรองรับการใช้งานของผู้ใช้ ดังนั้น สิ่งที่คุณต้องศึกษาต่อไปก็คือ จะมีเหตุการณ์อะไรบ้างที่อาจเกิดขึ้นกับคอนโทรล คอนโทรลแต่ละชนิดสนับสนุนเหตุการณ์อะไรบ้าง เหตุการณ์ที่เกิดขึ้นในแต่ละคอนโทรล มีความสัมพันธ์กันอย่างไร ซึ่งผู้เขียนจะอธิบายแยกแต่ละคอนโทรลในบทต่อไป สำหรับวิธีการเพิ่มโพรซีเจอร์เหตุการณ์ของแต่ละคอนโทรลมีดังนี้

1. ให้คุณเลือกชนิดของคอนโทรล หรืออ็อบเจกต์ใน object list box

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

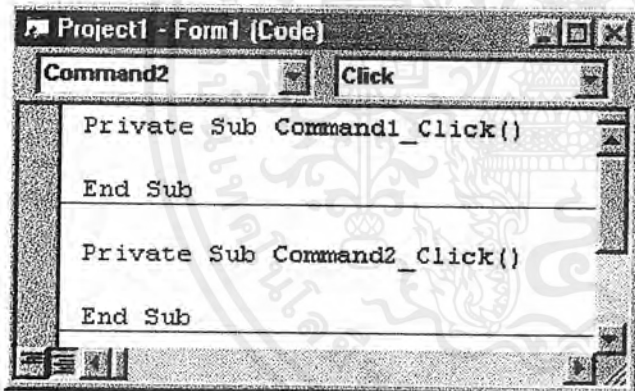
2.เลือกเหตุการณ์ใน event list box

3.VB จะสร้างโปรซีเจอร์ใหม่ที่ตรงกับเหตุการณ์ที่คุณเลือกไว้ กับคอนโทรลนั้นๆ



รูปที่ 2.18 แสดงโปรซีเจอร์เหตุการณ์ LostFocus ของคอนโทรล CommandButton ที่ถูกเพิ่มเข้ามา

และถ้าในกรณีที่คุณเลือกคอนโทรลเหมือนกัน เหตุการณ์เหมือนกัน ในฟอร์มเดียวกัน VB จะสร้างโปรซีเจอร์เหตุการณ์ให้กับคอนโทรลทั้ง 2 ตัว เพราะ VB ถือว่าเป็นคนละอ็อบเจ็กต์กัน เช่น ถ้าคุณใช้งานคอนโทรล CommandButton 2 ตัว ซึ่งมีเหตุการณ์คลิกเหมือนกัน จึงมีเหตุการณ์คลิก 2 โปรซีเจอร์สำหรับคอนโทรล Command1 และ Command2 ผู้ใช้คลิกปุ่ม Command1 อาจจะออกจากโปรแกรม ผู้ใช้คลิก Command2 อาจจะแสดงข้อความ Hello World ! ก็ได้



รูปที่ 2.19 แสดงโปรซีเจอร์เหตุการณ์คลิกของคอนโทรล CommandButton 2 ตัว

สำหรับในส่วน object list box จะมีรายการพิเศษที่เขียนว่า (General) และใน event list box จะเขียนว่า (declarations) จะเป็นพื้นที่พิเศษที่ใช้สำหรับประกาศ (declare) ตัวแปรต่างๆ, กลุ่มฟังก์ชัน Windows API เป็นต้น รายละเอียดผู้เขียนจะถึงอีกครั้งในบทที่ 3

โค้ดชุดแรกของคุณ

หลังจากที่ VB ได้สร้างโปรซีเจอร์ต่างๆ ขึ้นมาประจำคอนโทรลแล้ว คุณจะต้องเริ่มเขียนโค้ดเพื่อที่จะทำให้แต่ละโปรซีเจอร์ทำงานได้ ซึ่งในแต่ละโปรซีเจอร์จะรองรับเหตุการณ์เดิขวเท่านั้น คุณจะต้องเขียนโค้ดให้แต่ละโปรซีเจอร์ทำงานครบ และเสร็จสิ้นภายในโปรซีเจอร์มันเอง เช่น เมื่อผู้ใช้คลิกปุ่ม OK แล้วจะเกิดอะไรขึ้นบ้าง, เมื่อฟอร์มถูกโหลดเข้ามาจะมีอะไรบ้าง, เมื่อผู้ใช้ดับเบิลคลิกที่ฟอร์ม จะเกิดอะไร เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้น คุณก็ต้องเขียน โค้ดรองรับเหตุการณ์เหล่านี้ทั้งหมด เป็นต้น

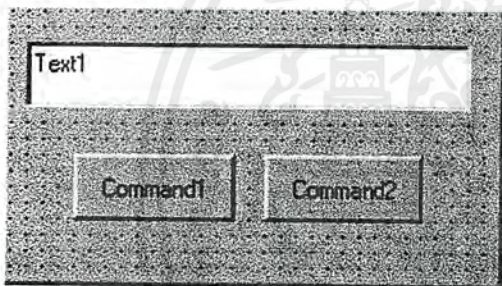
องค์ประกอบหลักของการเขียนโค้ดตอบรับ คือ คุณจะต้องเชื่อมโยงโพธิ์เคอร์ต่างๆ ให้สามารถทำงานสอดคล้องกัน ไปตามการทำงานของแอปพลิเคชันนั้นๆ ให้ได้ ผู้เขียนจะยกตัวอย่างโพธิ์เคอร์ที่คุณต้องใช้งาน ในทุกๆ แอปพลิเคชัน ก็คือ เมื่อผู้ใช้คลิกปุ่ม Command1 แล้วให้จบการทำงานของโปรแกรม

```
Private Sub Command1_Click()
```

```
End
```

```
End Sub
```

คำสั่ง End เป็นการบอกให้รู้ว่า เมื่อผู้ใช้คลิกปุ่ม Command1 แล้วให้จบการทำงาน ต่อไปผู้เขียนจะยกตัวอย่างที่แสดงให้เห็น การเชื่อมโยง 2 โพธิ์เคอร์เข้าด้วยกัน ให้วาคอนเตอร์เฟสคังรูป และเขียนโค้ดต่อไปนี้



รูปที่ 2.20 แสดงอินเตอร์เฟสเพื่อสั่งให้คอนโทรล TextBox แสดงข้อความ hello world

```
Private Sub Command1_Click ()
```


```
Text1.Text = "hello world"
```

```
End Sub
```

```
Private Sub Command2_Click ()
```

```
Text2.Text = " "
```

```
End Sub
```

ให้ทดลองรันโปรแกรมโดยการคลิกปุ่ม  บน ToolBar หรือกดปุ่ม F5 แล้วลองคลิกที่ปุ่ม Command1 และ Command2 คูผลที่เกิดขึ้น เมื่อคุณคลิกที่ปุ่ม Command1 จะมีข้อความ hello world ปรากฏในคอนโทรล TextBox และเมื่อคลิกที่ปุ่ม Command2 ข้อความดังกล่าวจะหายไป

ทำความเข้าใจกับโค้ดชุดแรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาลองดูความหมายของ โค้ดที่คุณเขียนเข้าไป มีรายละเอียดดังนี้

```
Private Sub Command1_Click ()
```

หมายถึง โพรซีเจอร์นี้ทำงานเมื่อผู้ใช้คลิกที่ปุ่ม Command1 เท่านั้น

```
Text1.Text = "hello world"
```

หมายถึง สั่งให้คอนโทรล TextBox ที่ชื่อ Text1 แสดงข้อความ hello world ผ่านทางคุณสมบัติ Text เพราะว่าคุณสมบัติ Text นี้มีหน้าที่แสดงข้อความ ส่วนจุด . ให้เรียกชื่อคอนโทรลกับคุณสมบัติ (properties) หรือเรียกชื่อคอนโทรลกับเมธอด (methods) กรณีนี้ Text เป็นคุณสมบัติของคอนโทรล TextBox

```
End Sub
```

หมายถึง จบโพรซีเจอร์

```
Private Sub Command2_Click ()
```

หมายถึง โพรซีเจอร์นี้ทำงานเมื่อผู้ใช้คลิกที่ปุ่ม Command2 เท่านั้น

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

หมายถึง สั่งให้คอนโทรล TextBox ที่ชื่อ Text1 ไม่ต้องแสดงข้อความ เครื่องหมาย "" แทนความว่างเปล่า

```
End Sub
```

จบโพรซีเจอร์

หมายเหตุในการเขียนโค้ด (Comment)

ในการเขียนโปรแกรม ไม่ว่าจะเป็นภาษาใดก็ตาม คุณควรที่จะต้องเขียนหมายเหตุไว้ เพื่อบอกความหมายของโค้ดบรรทัดต่อไป อาจเขียนไว้เพื่อบอกว่าตัวแปรนี้ใช้แทนอะไร ข้อความบอกจุดประสงค์ของโค้ดบรรทัดนั้นๆ เพื่อเป็นประโยชน์ ในกรณีที่คุณต้องย้อนกลับมา ศึกษาโค้ดนี้อีกครั้ง รูปแบบการเขียนหมายเหตุก็คือ ควรจะเป็นข้อความสั้นๆ กระชับ แต่ได้ใจความ ที่บอกความหมายของโค้ดบรรทัดนั้น หรือบอกว่าตัวแปรนั้นแทนอะไร คุณไม่ควรใช้คำอธิบาย เพราะตัวหมายเหตุเอง จะทำให้คุณงงมากกว่า โค้ดเสียอีก ในกรณีที่คุณต้องเขียน โค้ดมากๆ สำหรับการเขียน โค้ดที่มีการใช้งานตัวแปร ผู้เขียนจะกล่าวอีกครั้งในบทที่ 3

วิธีการเขียนหมายเหตุใน VB คือ จะใช้เครื่องหมาย ' (apostrophe) หน้าบรรทัดที่คุณต้องการทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ แล้ว editor จะเปลี่ยนบรรทัดนั้น เป็นสีเขียว และจะไม่สนใจบรรทัดดังกล่าว ในการประมวลผล
ทันที คุณอาจใช้หมายเหตุ เพื่อทดสอบที่ละโพธิ์เซอร์ก็ได้ เช่น กรณีที่โพธิ์เซอร์เกิด error บ่อยมาก :-
คุณก็หมายเหตุ โพธิ์เซอร์ ที่ error ไว้ก่อนก็ได้ คุณอาจใช้คำสั่ง Rem มาจากคำว่า Remark แทนเครื่องหมาย
' ก็ได้ โปรแกรมเมอร์ส่วนใหญ่ จะใช้หมายเหตุ เพื่อบอกความหมายของโค้ดเป็นช่วงๆ เสมอ แต่ผู้
เขียน จะหมายเหตุเกือบทุกบรรทัด เพราะผู้เขียนยังไม่ใช้โปรแกรมเมอร์ :-)

คุณสมบัติและเมธอดของคอนโทรลคืออะไร

สำหรับคอนโทรลและอ็อบเจกต์ทุกชนิดใน VB จะมีคุณสมบัติและเมธอดประจำตัวของมันเอง
ความหมายของทั้ง 2 อย่างมีดังนี้

1. คุณสมบัติ (properties)

หมายถึง รูปร่าง ลักษณะ ความกว้าง ความยาว สี ฯลฯ ของตัวคอนโทรลหรืออ็อบเจกต์ ขอให้
คุณมองคอนโทรลหรืออ็อบเจกต์ เป็นเสมือนวัตถุชิ้นหนึ่ง ที่มีความยาว ความกว้าง สีของตัวเอง คุณ
สามารถแก้ไขได้เหมือนสิ่งเช่น คุณกำลังปรับแต่งวัตถุชิ้นหนึ่งอยู่

สำหรับในแต่ละคอนโทรล หรืออ็อบเจกต์ อาจจะมีคุณสมบัติที่เหมือนกันหรือต่างกันได้ ขึ้นอยู่
กันหน้าที่ของแต่ละคอนโทรล คอนโทรลหรืออ็อบเจกต์หนึ่งๆ จะมีคุณสมบัติมากมาย หลายอย่าง ซึ่งคุณ
สามารถปรับแต่งคุณสมบัติ ให้ตรงกับความต้องการของคุณ มากเพียงใด แอปพลิเคชันของคุณก็จะมี ประ
สิทธิภาพมากขึ้นเท่านั้น ซึ่งถือได้ว่าเป็นจุดเริ่มต้น ได้ดี ในการพัฒนาแอปพลิเคชัน คุณสามารถปรับแต่ง
คุณสมบัติได้จากหน้าต่าง Properties หรือปรับแต่งด้วยการเขียนโค้ดก็ได้ จะมีคุณสมบัติบางตัว ที่
ไม่ใคร่ชอบพ้แนะนำ ให้ปรับแต่งด้วยการเขียนโค้ด และบางตัวปรับแต่งด้วยการแก้ไขในหน้าต่าง
Properties ซึ่งผู้เขียนจะอธิบายอย่างละเอียดอีกครั้ง ในการใช้งานแต่ละคอนโทรล และในทางปฏิบัติ คุณ
ไม่จำเป็นต้องปรับแต่งทุกๆ คุณสมบัติ เพราะ VB ได้ตั้งค่า default ไว้ให้คุณแล้ว ซึ่งก็สามารถใช้งานได้ใ
ระดับหนึ่ง

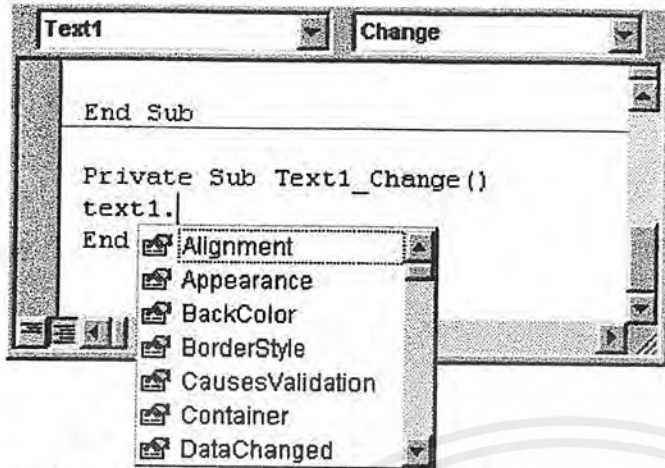
2. เมธอด (methods) หมายถึง พฤติกรรมของคอนโทรลหรืออ็อบเจกต์ อาจกล่าวได้ว่าเป็นการ
ควบคุมการทำงานของคอนโทรล หรืออ็อบเจกต์นั่นเอง จะใช้จุดเป็นตัวคั่นระหว่าง ชื่อคอนโทรลกับเม
ธอด ซึ่งจะเห็นได้ว่า คุณสมบัติและเมธอดมีความใกล้เคียงกันมาก เนื่องจากจะใช้จุด เป็นตัวแยกระหว่างชื่อ
คอนโทรลกับคุณสมบัติ หรือชื่อคอนโทรลกับเมธอดเหมือนกัน จะมีความแตกต่างกัน ในแง่ของการควบ
คุมคอนโทรล หรืออ็อบเจกต์ ในการใช้งานซึ่งคุณจะได้ศึกษาในหัวข้อต่อไป

ความสามารถพิเศษของ editor

จากตัวอย่างข้างต้นจะพบว่า ขณะที่คุณพิมพ์โค้ดให้กับแต่ละโพธิ์เซอร์ เมื่อคุณพิมพ์ ตัว editor
จะแสดง Tooltip ที่เป็นรายการคุณสมบัติ หรือรายการเมธอด ที่คอนโทรลนั้นสนับสนุนอยู่ขึ้นมาทันที

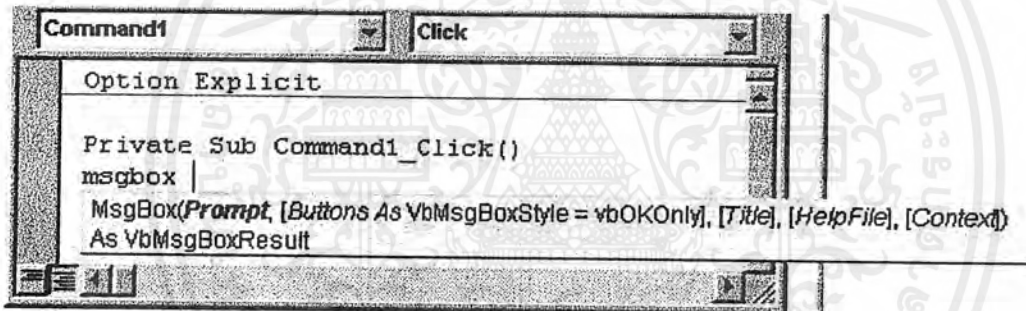
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มันช่วยให้คุณไม่ต้องไปจดจำว่า คอนโทรลนี้มีคุณสมบัติอะไรบ้าง มีเมธอดอะไร รวมถึงป้องกันไม่ให้คุณพิมพ์ผิดอีกด้วย ดังรูป



รูปที่ 2.21 แสดง Tooltip ของ editor ขณะเขียนโค้ด

และถ้าในการเขียนโค้ดที่ต้องการมีการเรียกใช้งานฟังก์ชันมาตรฐานต่างๆ Tooltip ก็จะแสดงรูปแบบไวยากรณ์อีกด้วย



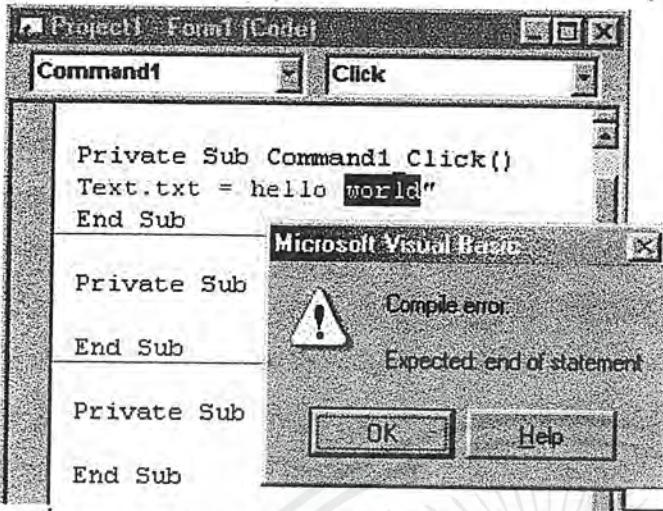
รูปที่ 2.22 แสดงไวยากรณ์ประจำฟังก์ชันนั้นๆ

ความสามารถทั้ง 2 อย่างช่วยให้โปรแกรมเมอร์ช่นระยะเวลาในการเขียนโค้ดเป็นอย่างมาก เพราะจะทำให้ข้อผิดพลาดที่เกิดจากการใช้งานผิดไวยากรณ์ หรือพิมพ์ผิด หดหายไปได้อย่างสิ้นเชิง และที่สำคัญคุณสามารถพัฒนาแอปพลิเคชันได้เต็มที่ โดยที่ไม่ต้องเสียเวลาเกี่ยวกับเรื่องที่คุณต้องท่องจำคุณสมบัติต่างๆ นานา รูปแบบการใช้งานที่มากมายเหลือเกิน ถ้าพึ่งแต่ความรู้ เทคโนโลยีใหม่ที่เกิดขึ้นมา ก็แทบเรี้นรู้กันไม่หมดแล้ว :- (ซึ่งผู้เขียนเห็นว่า คุณสามารถแสดง ความสามารถ ได้สุดฝีมือ โดยที่ยังอยู่ในขอบเขตที่ VB จะทำได้ เพราะอย่างไร VB ก็ยังไม่สามารถเทียบเท่า Visual C++ อยู่ดี แต่คุณไม่ต้องห่วง ผู้เขียนเคยเห็นแอปพลิเคชันที่พัฒนาด้วย VB ที่มีการเขียนโค้ดแบบประยุกต์ โดยโปรแกรมเมอร์ต่างประเทศ ถ้าคุณทำได้ คุณก็สามารถ พัฒนาแอปพลิเคชันที่มีประสิทธิภาพ ไม่ยิ่งหย่อนไปกว่าแอปพลิเคชันที่พัฒนาด้วย Visual C++ หรือภาษาอื่นๆ เลย :-)

สิ่งที่น่าสนใจอีกอย่างหนึ่งก็คือ มันสามารถตรวจสอบไวยากรณ์ (syntax) ตามโครงสร้างของภาษา VB ได้อีกด้วย ซึ่งเรียกว่า Auto Syntax Check ขณะที่คุณพิมพ์โค้ดเข้าไป เมื่อคุณกด Enter จบบรรทัด VB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะทำการตรวจสอบไวยากรณ์ทันที ถ้ามีข้อผิดพลาด ในการใช้งานไวยากรณ์เกิดขึ้น VB จะแสดงข้อความช่วยเหลือที่เกี่ยวข้องกับความผิดพลาดนั้นๆ ขึ้นมาทันที ทำให้คุณแก้ไขความผิดพลาดนั้น ได้อย่างถูกต้อง




รูปที่ 2.23 แสดงข้อความช่วยเหลือ เมื่อคุณพิมพ์ผิดไวยากรณ์

จากรูป เมื่อคุณกดปุ่ม help VB จะแสดงข้อความที่เกี่ยวข้องกับความผิดพลาดที่เกิดขึ้นทันที (ซึ่งผู้เขียนใช้บริการนี้บ่อยมาก)

สำหรับรูปแบบการแสดง ข้อผิดพลาดที่พิมพ์ลงไปในแต่ละโปรแกรม VB จะแยกส่วนของข้อผิดพลาดออกเป็น 3 กลุ่ม ตัวอย่างของฟอนต์ คือ

1. ตัวแปร, ชื่อคอนโทรล, เหตุการณ์ จะมีสีดำ
2. คำสั่ง, ฟังก์ชัน จะมีสีน้ำเงิน
3. หมายเหตุ จะมีสีเขียว

 **Note** เมื่อคุณดู source code คุณสามารถใช้กลุ่มสีของฟอนต์ช่วยในการทำความเข้าใจโค้ดด้วยก็ได้

สิ่งหนึ่งที่จะทำให้เกิดข้อผิดพลาด ได้ก็คือ ตัวแปรที่คุณประกาศใช้งานเป็นรูปแบบหนึ่ง เวลาคุณพิมพ์อีกแบบหนึ่ง VB จะเปลี่ยนให้คุณ โดยอัตโนมัติ เช่น เมื่อคุณประกาศตัวแปร tName As String แต่เวลาที่คุณนำไปใช้ คุณพิมพ์ผิดเป็น tname VB จะทำการเปลี่ยนตัวแปรที่คุณพิมพ์ไม่เหมือนกับต้นแบบให้เหมือนกัน โดยอัตโนมัติ และในกรณีที่คุณมีการ ใช้คำสั่ง (statement) (หรือคำสั่งวงประเภทที่ VB ให้คุณใช้ ซึ่ง คำสวมนั้นจะมีผลในการทำงานของโค้ด ในโปรแกรมด้วย) VB จะเปลี่ยนอักษรตัวแรก ให้เป็นตัวใหญ่เสมอ เช่น เมื่อคุณพิมพ์ end เข้าไป VB จะเปลี่ยนเป็น End หรือเมื่อคุณพิมพ์ text1.txt = "hello world" เข้าไป VB จะเปลี่ยนเป็น Text1.Text = "hello world" ให้คุณ

ทั้งหมดที่ผู้เขียนกล่าวมา จะเห็นได้ว่าไม่ว่าใครจะพยายามทำให้ editor มีบทบาทช่วย

โปรแกรมเมอร์ในการพัฒนาแอปพลิเคชันมากที่สุด ยังมีความสามารถอื่นๆ อีกของ editor ที่ผู้เขียนไม่ได้กล่าวไว้ เทคนิคที่คุณใช้ใน Word 97 บางอย่าง คุณสามารถนำมาใช้งานกับ editor ได้เช่นกัน ทดลองดูครับ

คุณสมบัติประจำตัวของคอนโทรล

จากหัวข้อที่ผ่านมา ทำให้คุณทราบว่า คอนโทรลแต่ละชนิดล้วนแล้วแต่มีทั้งคุณสมบัติ และเมธอดทั้งสิ้น ยกเว้นการจกจำ (และไม่ต้องจำ) แต่ในทุกๆคอนโทรล จะมีคุณสมบัติอยู่ตัวหนึ่งที่ใช้สำหรับเก็บค่าแสดงค่า หรืออ่านค่าของตัวเอง เช่น คอนโทรล TextBox มีหน้าที่สำหรับ รับหรือแสดงข้อมูลที่ผู้ใช้สามารถแก้ไขได้ ทั้ง 2 กรณีตัวคอนโทรล TextBox จะเก็บข้อมูลไว้ที่คุณสมบัติ Text จึงถือได้ว่า คุณสมบัติ Text เป็นคุณสมบัติประจำตัวของคอนโทรล TextBox, คอนโทรล CommandButton มีหน้าที่เป็นปุ่มกรับคำสั่งของผู้ใช้ เช่น ปุ่ม OK , Cancel เป็นต้น คุณสมบัตินี้จะแสดงชื่อปุ่มคือ คุณสมบัติ Caption, คอนโทรล Label มีหน้าที่แสดงข้อมูลให้ผู้ใช้ทราบ โดยที่ผู้ใช้ไม่สามารถแก้ไขได้ ก็จะมีคุณสมบัตินี้ Caption เป็นตัวเก็บค่า เป็นต้น ซึ่งคุณสมบัตินี้ที่กล่าวมาคุณจะต้องเข้าไปใช้งานอย่างแน่นอน ตารางต่อไปนี้ เป็นคุณสมบัติประจำ ของแต่ละคอนโทรล

คอนโทรล	คุณสมบัติประจำตัว
CheckBox	Value
ComboBox	Text
CommandButton	Caption
CommonDialog	Action
Data	Caption
DBCombo(Data-Bound ComboBox)	Text
DBGrid(Data-Bound Grid)	Text
DBList(Data-Bound ListBox)	Text
DirListBox	Path
DriveListBox	Drive
FileListBox	FileName
Frame	Caption
HscrollBar	Value
Image	Picture
Label	Caption

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Line	Visible
ListBox	Text
OptionButton	Value
PictureBox	Picture
Shape	Shape
TextBox	Text
Timer	Enabled
VscrollBar	Value

ในการเขียนโค้ดกับคุณสมบัติประจำตัวของคอนโทรล คุณไม่จำเป็นต้องระบุคุณสมบัติก็ได้ โดยที่ไม่คิดไวยากรณ์แต่อย่างใด สำหรับในการใช้งานจริงๆแล้ว โปรแกรมเมอร์ทุกๆไปจะไม่นิยมใช้เพราะว่ามันจะทำให้คุณสับสนเป็นอย่างยิ่ง เนื่องจากเฉพาะโค้ดของแอพลิเคชันหนึ่ง จะประกอบไปด้วยจำนวนโพรซีเจอร์มากมาย ซึ่งในแต่ละโพรซีเจอร์ยังมีจำนวนบรรทัดของโค้ดอีก หลายสิบ หลายร้อย บรรทัด ทำให้ไม่นิยมใช้ เช่น

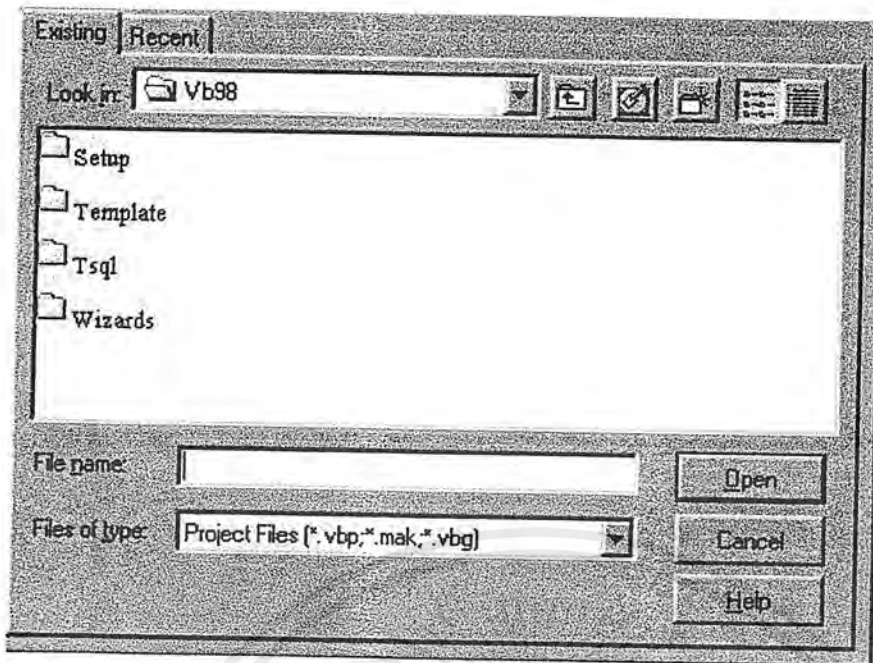
รูปแบบที่ 1 Text1.Text = "hello world"

รูปแบบที่ 2 Text1 = "hello world"

รูปแบบที่ 1 เป็นการเขียนโค้ดแบบปกติ ส่วนรูปแบบที่ 2 เป็นการเขียนโค้ดกับคุณสมบัติประจำตัวคอนโทรล ดังนั้นจึงไม่ต้องระบุคุณสมบัติ Text ก็ได้ ซึ่งไม่คิดไวยากรณ์แต่อย่างใด ที่ผู้เขียนกล่าวถึงเนื่องจากว่าเป็นการแสดงความชัดเจนที่ไม่โครซอฟท์เตรียมไว้ให้คุณเท่านั้น

การเปิดโปรเจกต์ที่มีอยู่แล้ว

1. ให้คุณเลือกเมนู File/Open Project หรือคลิกที่ปุ่ม  ที่ทูลบาร์ แล้ว VB จะแสดงไดอะล็อกบ็อกซ์ Open Project ดังรูป

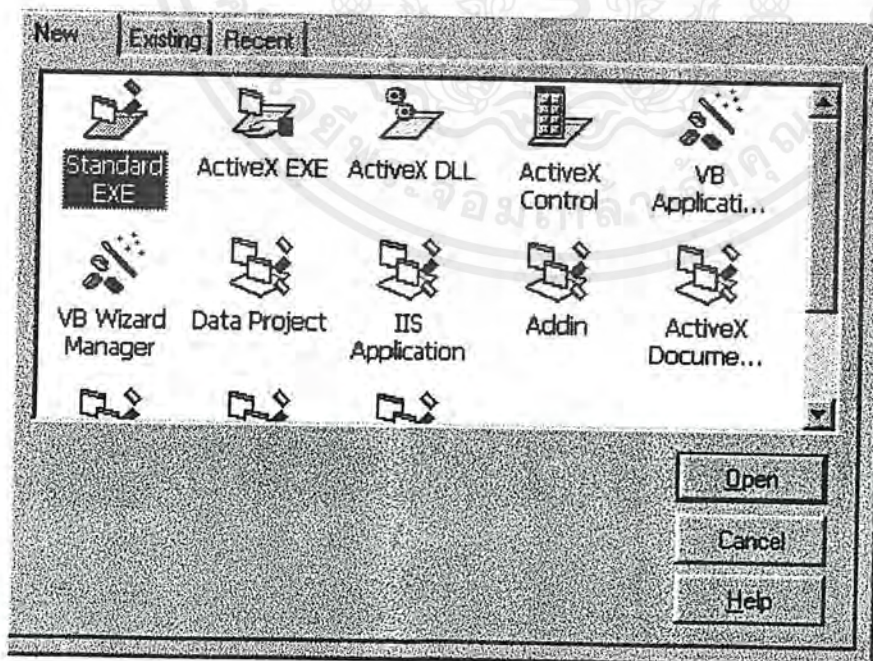


รูปที่ 2-24 แสดงไดอะล็อกบ็อกซ์ *Open Project*

2.แท็บ Existing หมายถึง คุณต้องการเปิด โปรเจกต์ที่มีอยู่ โดยที่ไม่สนใจว่าจะเป็น โปรเจกต์ที่คุณสร้างมาเมื่อใด ส่วนแท็บ Recent หมายถึง คุณต้องการเปิด โปรเจกต์ที่คุณพัฒนาครั้งหลังสุด ซึ่งจะแสดงไครฟ์ และโฟลเดอร์ที่เก็บโปรเจกต์นั้นๆ อยู่

การเพิ่มโปรเจกต์ เข้ามาในสภาพแวดล้อม

1.ให้คุณเลือกเมนู Project/Add Project แล้ว VB จะแสดง ไดอะล็อกบ็อกซ์ Add Project ดังรูป



รูปที่ 2-25 แสดงไดอะล็อกบ็อกซ์ *Add Project*

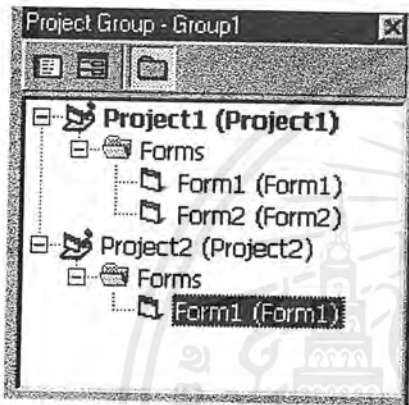
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.คุณสามารถเลือกชนิดของโปรเจกต์ที่คุณต้องการ ได้จากไอคอนบ็อกซ์ดังกล่าว แท็บ Existing หมายถึง คุณต้องการเพิ่มโปรเจกต์ที่มีอยู่ โดยที่ไม่สนใจว่าจะเป็นโปรเจกต์ที่คุณสร้างมาตอนไหน ส่วนแท็บ Recent หมายถึง คุณต้องการเพิ่มโปรเจกต์ที่คุณพัฒนาครั้งล่าสุด ซึ่งจะแสดง ใครที่ และ โพลเดอร์ที่เก็บโปรเจกต์นั้นๆ อยู่

การถอดโปรเจกต์ออกจากสภาพแวดล้อม

จะใช้สำหรับกรณีที่คุณมีการเพิ่มโปรเจกต์เข้ามา แล้วคุณต้องการถอดโปรเจกต์นั้นๆ ออกจากสภาพแวดล้อม คุณต้องทำดังนี้


1.ให้คุณเลือก โปรเจกต์ที่คุณต้องการถอด จากหน้าต่าง Project Explorer ดังรูป

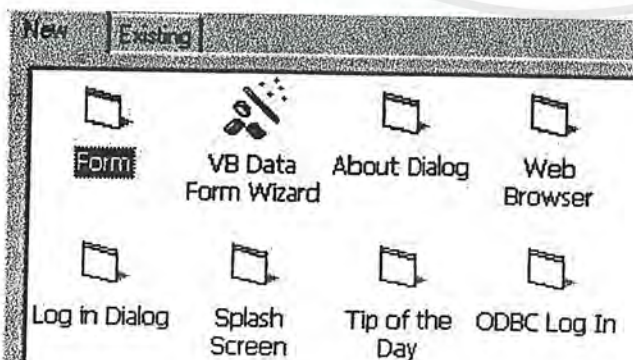


รูปที่ 2-26 แสดงการเลือกโปรเจกต์ที่คุณต้องการถอดออกจากสภาพแวดล้อม

2.เลือกคำสั่ง File/Remove Project











การเพิ่ม Form หรือโมดูลอื่นๆ เข้ามาในสภาพแวดล้อม

1.ให้คุณเลือกเมนู Project จากนั้น ให้คุณเลือกชนิดของ โมดูล ที่คุณต้องการเพิ่มเข้ามาในสภาพแวดล้อม และถ้าในกรณีที่คุณต้องการเพิ่ม Form เข้ามาในโปรเจกต์ คุณสามารถคลิกที่ปุ่ม  บนทูลบาร์เลขก็ได้



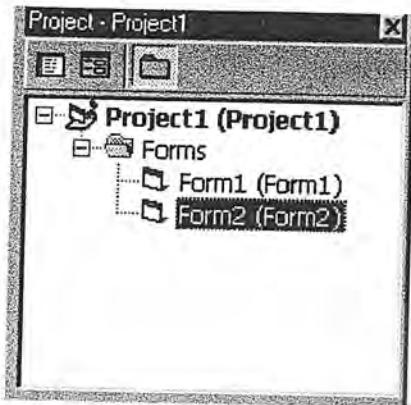
รูปที่ 2-27 แสดงการเพิ่ม Form เข้ามาในโปรเจกต์

แท็บ Existing หมายถึง คุณต้องการเพิ่มฟอร์มที่มีอยู่แล้ว ในกรณีนี้เป็นการเพิ่มฟอร์มว่างๆ เข้ามา ในสภาพแวดล้อม สำหรับไอคอนต่างๆ มีความหมายดังนี้

 Form	เป็นฟอร์มว่างๆ แบบปกติ
 VB Data Form Wizard	เป็นการใช้ความสามารถของ VB Data Form Wizard เพื่อสร้างแบบฟอร์มรายงาน ที่ติดต่อกับฐานข้อมูล โดยผ่านคอนโทรล ADO Data Control
 About Dialog	เป็นแบบฟอร์ม About สำเร็จรูป คุณอาจจะต้องปรับแต่งเพื่อให้ได้ฟอร์ม About ตามที่คุณต้องการ
 Web Browser	เป็นฟอร์มที่มีลักษณะเหมือนกับ browser คล้ายๆ กับ Internet Explorer หรือ Netscape Communicator
 Dialog	ไอคอนล็อกบ็อกซ์สำเร็จรูป ถ้าคุณเบื่อกับ ไอคอนล็อกบ็อกซ์แบบเก่าๆ นี่เป็น ไอคอนล็อกบ็อกซ์ที่คุณออกแบบเอง
 Log in Dialog	ถ้าคุณสร้างแอปพลิเคชันที่ต้องการตรวจสอบ หรือ จำกัดสิทธิการใช้งานผู้ใช้ ในแต่ละระดับ แบบฟอร์ม Log in สำเร็จรูปช่วยคุณได้ ซึ่งจะต้องมีการใส่ชื่อผู้ใช้ และ password ด้วย
 Splash Screen	เป็นจอภาพเริ่มต้นก่อนเข้าแอปพลิเคชัน ทุกๆ แอปพลิเคชันต้องมี
 Tip of the Day	คุณคงเคยเห็น ไอคอนล็อกบ็อกซ์ที่แสดง Tip ตอนเข้าสู่แอปพลิเคชัน นี่เป็น ไอคอนล็อกบ็อกซ์ที่แสดง Tip สำเร็จรูป ซึ่งคุณต้องเพิ่มเทคนิคต่างๆ เข้าไปเอง
 ODBC Log In	เป็นแบบฟอร์มที่ใช้สำหรับติดต่อกับฐานข้อมูล โดยผ่านไคลเอนต์ ODBC
 Options Dialog	เป็นแบบฟอร์มที่มีลักษณะเป็นตัวเลือกเพื่อกำหนดหัวข้อค่านต่างๆ ตามความต้องการของผู้ใช้ ซึ่งคุณจะพบเห็นได้ในแอปพลิเคชันต่างๆ ไป


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

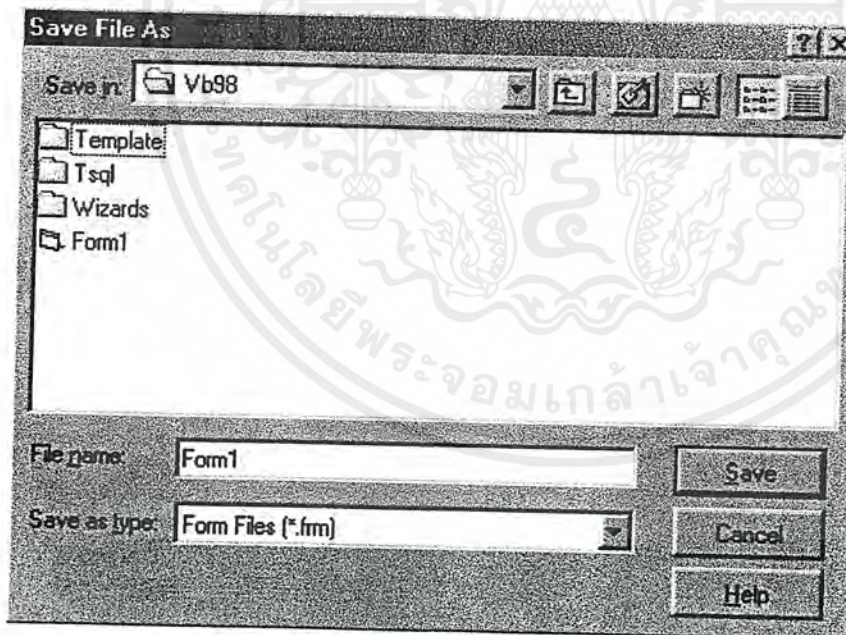
2.ให้คุณสังเกตที่หน้าต่าง Project Explorer จะมีรายการแสดงเพิ่มขึ้นมา ซึ่งอยู่ภายใต้ Project เดียวกัน จากรูปจะเห็นว่า Project1 ประกอบด้วย Form 2 ฟอর্ম



รูปที่ 2-28 แสดง Form ที่ถูกเพิ่มเข้ามาในโปรเจกต์

การเซฟโปรเจกต์

1.ให้คุณเลือกเมนู File/Save Project Group As... หรือคลิกที่ปุ่ม  บน Tool Bar แล้ว VB จะแสดงไดอะล็อกบ็อกซ์ เซฟโปรเจกต์ ดังรูป ซึ่ง VB จะแสดงรายการส่วนประกอบต่างๆ ที่อยู่ใน โปรเจกต์ นั้นๆ ให้คุณทราบด้วย






รูปที่ 2-29 แสดงไดอะล็อกบ็อกซ์ เซฟโปรเจกต์


2.คลิกปุ่ม 


ในกรณีที่เลือกคำสั่ง Save Form As... หมายถึง เป็นการเซฟเฉพาะฟอর্মที่คุณ ใช้งานอยู่เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดสอบ (run) โปรเจกต์

1. ให้คุณเลือกเมนู Run/Start หรือกดปุ่ม F5 ที่คีย์บอร์ด หรือกดปุ่ม  ที่ทุลบาร์
2. ที่ฟอร์มจะมีการเปลี่ยนรูปแบบ ไม่มีเส้นกริด และจะปรากฏหน้าต่าง Immediate ด้านล่างของสภาพแวดล้อม
3. ถ้าคุณมีการสั่งให้พิมพ์โดยผ่านทางคำสั่ง Debug.Print ที่หน้าต่าง Immediate จะแสดงผลจากการประมวลผลด้วย
4. คุณจะสังเกตเห็นว่าที่บนทุลบาร์ จะมีปุ่ม  และ  มีสถานะ Active อยู่ โดยมีความหมายดังนี้

4.1 ปุ่ม  หมายถึง ให้หยุดการรัน โปรเจกต์

4.2 ปุ่ม  หมายถึง ให้หยุดรัน โปรเจกต์ชั่วคราวเพื่อ แก้ไขโค้ด



โครงสร้างในการเขียนโปรแกรมด้วย Visual Basic 6.0

การประกาศตัวแปร

1. แบบ Implicit

2. แบบ Explicit

กฎการตั้งชื่อตัวแปรและค่าคงที่

1. การตั้งชื่อคอนโทรลและอ็อบเจ็กต์ตามคำแนะนำของไมโครซอฟท์

ชนิดของข้อมูล

การใช้งานตัวแปรแบบใช้สัญลักษณ์พิเศษกำกับขอบเขตของตัวแปร

1. ตัวแปรแบบ Local

2. ตัวแปรแบบ Public

การใช้งานตัวแปรร่วมระดับโปรซีเคอร์

การตั้งชื่อตัวแปรแบบบอกรชนิดและขอบเขต

ตัวแปรอาร์เรย์

1. แบบสแตติก

2. แบบไดนามิก

การสร้างตัวแปรอาร์เรย์มากกว่า 1 มิติ

การใช้งานคอนโทรลอาร์เรย์

การสร้างชนิดของข้อมูลขึ้นใช้เอง

การประกาศค่าคงที่

ตัวดำเนินการใน Visual Basic 6.0

1. ด้านคณิตศาสตร์

2. ด้านตรรกะ

3. ด้านการเปรียบเทียบ

4. ด้านเชื่อมข้อความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในบทที่ 1 และบทที่ 2 ผู้เขียน ได้อธิบายแนวทางเบื้องต้นสำหรับการพัฒนาแอปพลิเคชันด้วย VB หวังว่าจะเป็นประโยชน์เกี่ยวกับชนิดของข้อมูล, การใช้งานตัวแปร และค่าคงที่ VB สนับสนุน ซึ่งถือได้ว่าเป็นความสำคัญเป็นอย่างยิ่ง เพราะจะส่งผลถึงประสิทธิภาพ การทำงานของแอปพลิเคชันของคุณด้วย ไม่ว่าจะเป็ความเร็วในการประมวลผลขนาดของแอปพลิเคชัน ประสิทธิภาพในการทำงาน รวมถึงแอปพลิเคชันที่ได้เหมาะสมกับเวลาที่ผู้ใช้ไปหรือไม่

ไม่ว่าคุณ จะพัฒนาแอปพลิเคชันด้วยภาษาใดก็ตาม สิ่งที่คุณต้องเรียนรู้เป็นอันดับต้นๆ ก็คือ ตัวแปร ค่าคงที่ และชนิดของข้อมูล ที่ภาษานั้นๆ สนับสนุน หรือรองรับอยู่ ซึ่งจะช่วยให้คุณเห็นข้อจำกัดต่างๆ จุดเด่น จุดด้อย ในแต่ละส่วน ทำให้คุณสามารถเลือกใช้งานได้อย่างถูกต้อง และมีประสิทธิภาพ เพราะว่า ตัวแปร และค่าคงที่ถือได้ว่าเป็นตัวแทนของข้อมูล ที่คุณจะไปเก็บข้อมูล ประมวลผล และแสดงข้อมูลที่ เกิดจากการประมวลผล ซึ่งจะดีเพียงใด ก็ขึ้นอยู่กับคุณ ดังนั้นคุณควรศึกษารูปแบบเบื้องต้น ในการใช้งาน ให้เข้าใจเสียก่อน คุณจึงสามารถนำไปใช้ได้

การประกาศตัวแปร (Declarations)

ก่อนที่คุณจะใช้งานตัวแปร หรือค่าคงที่ทุกครั้ง คุณควรที่จะประกาศตัวแปร (declare) ก่อน เพื่อบอกให้ VB รู้ว่า ตัวแปรชื่อนี้ คุณต้องการใช้งาน และตัวแปรดังกล่าว ใช้แทนข้อมูลชนิดใด เหตุที่ผู้เขียนใช้คำว่า ควรที่จะ เนื่องจากว่า VB จะอนุญาตให้คุณใช้งานตัวแปร ได้ โดยที่คุณ ไม่จำเป็นต้องประกาศตัวแปร แต่ตัวแปรที่คุณ ได้มา มันจะกินทรัพยากรระบบ มากเกินความจำเป็น รวมถึงประมวลผล ได้ช้าอีกด้วย เพราะจะเป็นตัวแปรที่สามารถแทนข้อมูล ได้ทุกชนิด ซึ่ง VB เรียกว่า ตัวแปรชนิด Variant คุณอาจคิดว่า ตัวแปรชนิด Variant ก็คืออยู่แล้ว ไม่ต้องยุ่งยาก ในการกำหนดรายละเอียดต่างๆ ให้วุ่นวาย แต่โปรแกรมเมอร์จะไม่นิยมใช้ และหลีกเลี่ยงที่จะใช้งานตัวแปรชนิด Variant อีกด้วย จะใช้ในกรณีที่ทำเป็นเท่านั้น เนื่องจากว่ามีผลเสียมากกว่าผลดีที่คุณจะได้รับ เหตุผลที่สำคัญอีกประการหนึ่งก็คือ ตัวแปรต่างๆ ที่คุณนำมาใช้งาน ใน โปรเจกต์ของคุณ ตัวคุณเองข้อมรู้ดีว่า จะใช้ตัวแปรใดบ้าง และจะให้ตัวแปรตัวใด แทนข้อมูลชนิดไหน ซึ่งคุณเองเป็นผู้กำหนดทั้งหมด ดังนั้นจึงไม่มีเหตุผลใด ที่คุณจะใช้ตัวแปรชนิด Variant จากเหตุผลข้างต้น คุณควรที่จะประกาศตัวแปรทุกครั้ง ก่อนที่จะนำไปใช้งาน ใน VB มีรูปแบบการประกาศดังนี้

Dim varname As datatype

ความหมายของแต่ละส่วน มีดังนี้

1. Dim คือ คำสั่ง (statements) ที่บอกให้ VB รู้ว่า คุณต้องการประกาศตัวแปร
2. varname คือ ชื่อของตัวแปร คุณสามารถตั้งชื่อตัวแปร ได้อย่างอิสระ แต่ต้อง ไม่ผิดกฎการตั้งชื่อของ VB คุณควรที่จะตั้งชื่อตัวแปร ให้สื่อกับข้อมูลที่ตัวมันเก็บอยู่ จะทำให้คุณอ่าน ได้ค ได้ง่ายขึ้น

3. As คือ คำสงวน (keywords) ที่บอกให้ VB รู้ว่า คุณต้องการกำหนดให้ตัวแปรชื่อดังกล่าวแทนข้อมูลชนิดใด

4. datatypes คือ ชนิดของข้อมูลที่ VB สนับสนุนอยู่ ซึ่งมีอยู่หลายชนิด เช่น ตัวเลขจำนวนเต็ม (integer), ตัวอักษร (string) เป็นต้น

การประกาศตัวแปรใน VB สามารถแยกได้ 2 แบบคือ

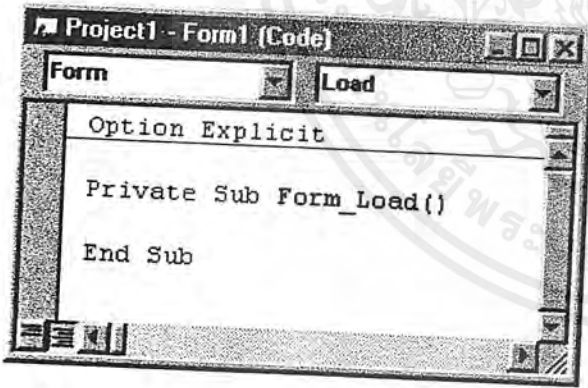
1.การประกาศตัวแปรแบบ Implicit Declaration

หมายถึง VB ยอมให้คุณใช้งานตัวแปรได้ โดยไม่ต้องมีการประกาศตัวแปร ชนิดของข้อมูลที่ได้จะเป็นแบบ Variant ซึ่งไม่ควรนำมาใช้งานด้วยเหตุผลข้างต้น แต่ถ้าคุณต้องการใช้งานตัวแปรชนิดอื่นๆ โดยการประกาศแบบนี้ คุณสามารถใช้สัญลักษณ์พิเศษที่ VB กำหนดไว้ เพื่อเปลี่ยนชนิดของข้อมูล เป็นชนิดอื่นๆ ได้ ให้ดูหัวข้อ ชนิดของข้อมูลเพิ่มเติม

2.การประกาศตัวแปรแบบ Explicit Declaration

หมายถึง คุณจะต้องการมีการประกาศตัวแปรก่อนการใช้งานทุกครั้ง แล้วจึงสามารถนำตัวแปรนั้นๆ ไปใช้งานได้

Note การประกาศตัวแปรแบบ Explicit ให้คุณใส่คำสั่ง **Option Explicit** ไว้ในส่วนบนสุด ก่อนการสร้างโปรซีจเจอร์ จะเป็นการบังคับให้คุณ ต้องประกาศตัวแปรก่อน ที่จะนำไปใช้งานทุกครั้ง ถ้าคุณ ไม่ต้องการพิมพ์ข้อความดังกล่าว ทุกครั้ง ให้คุณเลือกเมนู Tools/Option เลือกหัวข้อ **Require Variable Declaration** VB จะใส่ข้อความดังกล่าว ใ้คุณโดยอัตโนมัติทุกครั้ง ดังรูป



รูปที่ 3.1 แสดงข้อความ Option Explicit

กฎการตั้งชื่อตัวแปรและค่าคงที่

1. ให้ขึ้นต้นด้วยพยางค์เท่านั้น
2. ความยาวของชื่อที่คั้งสูงสุดไม่เกิน 255 ตัวอักษร ในทางปฏิบัติ การตั้งชื่อที่ยาวจนเกินไป จะไม่เกิดประโยชน์ใดๆ กลับก่อให้เกิดผลเสียคือ คุณต้องเสียเวลาพิมพ์โดยเปล่าประโยชน์
3. ชื่อที่คุณตั้ง จะต้อง ไม่ซ้ำกับคำสงวน (keywords), คำสั่ง (statements), ฟังก์ชัน (functions) หรือ


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อื่นๆ ที่ VB กำหนดไว้

4. ห้ามตั้งชื่อซ้ำกันในโปรซีเจอร์เดียวกัน หรือในขอบเขตเดียวกัน

5. ห้ามใช้เครื่องหมายทางคณิตศาสตร์, ตัวดำเนินการ (Operators) หรือ เครื่องหมายพิเศษ เช่น @, # มาตั้งชื่อ

6. ห้ามมีช่องว่างในชื่อของตัวแปร ถ้าต้องการเว้นว่าง ให้ใช้เครื่องหมาย _ (underscore) เท่านั้น เช่น end_salary เป็นต้น

 **Note** ถ้าคุณตั้งชื่อผิดกฎ VB จะมีข้อความเตือนให้คุณทราบ

การตั้งชื่อคอนโทรลและอ็อบเจกต์ตามคำแนะนำของไมโครซอฟท์

เนื่องจาก VB มีคอนโทรลต่างๆ มากมาย โปรแกรมเมอร์มักจะใช้คำนำหน้า (prefixes) ในการตั้งชื่อคอนโทรล ต่อไปนี้เป็นคำแนะนำของไมโครซอฟท์ ที่คุณควรใช้ตั้งชื่อคอนโทรล และอ็อบเจกต์ ดังตารางต่อไปนี้

คอนโทรล	คำนำหน้า (Prefixes)	ตัวอย่าง
CheckBox	chk	chkStudents
ComboBox	cbo	cboSalary
CommandButton	cmd	cmdExit
CommonDialog	dlg	dlgSave
Data Control	dat	datStudents
DBComboBox	dbc	dbcList
DBGrid	dbg	dbgWork1
DBListBox	dbl	dblDisplay
DirListBox	dir	dirDestination
DriveListBox	drv	drvMain
FileListBox	fil	filInput
Frame	fra	fraDisplay
HscrollBar	hsb	hsbColor
Image	img	imgMain
Label	lbl	lblName
Line	lin	linTop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ListBox	lst	lstMouth
OptionButton	opt	optSex
Ole Container	ole	oleWord
PictureBox	pic	picPeople
Shape	shp	shpCircle
TextBox	txt	txtInput
Timer	tmr	tmrCount
VscrollBar	vsb	vsbHeight

ชนิดของข้อมูล

VB มีชนิดของข้อมูลที่เป็นพื้นฐาน ให้คุณเลือกใช้ได้เหมาะสมตามที่คุณต้องการมากมาย ไม่ว่าจะเป็นจำนวนเต็ม (Integer), เศษส่วน ที่คุณยังสามารถ เลือกความละเอียดจำนวนตัวเลข ได้อีกด้วย เพื่อให้เหมาะสมกับข้อมูลที่คุณต้องการเก็บ, ข้อความ (string), ตัวเลขทางการเงิน (Currency), ค่าทางตรรก (boolean) เป็นต้น ข้อมูลแต่ละชนิดที่กล่าวมา จะใช้พื้นที่ในการเก็บไม่เท่ากัน รวมถึงความเร็วในการประมวลผลก็แตกต่างกันด้วย เช่น ถ้าเป็นตัวเลข ให้คุณพยายามใช้ข้อมูลชนิด Integer ให้มากที่สุด เพราะจะใช้ทรัพยากรน้อย และประมวลผลได้เร็ว แต่ก็ต้องขึ้นอยู่กับข้อมูลที่จะเก็บด้วย เนื่องจากข้อมูลชนิด Integer สามารถเก็บค่าที่อยู่ระหว่าง -32768 ถึง 32767 เท่านั้น ให้คุณเลือกใช้ชนิดของข้อมูลที่ใช้ทรัพยากรระบบให้น้อยที่สุด เท่าที่จะเป็นไปได้ก่อน โดยที่ยังสามารถรองรับความต้องการของคุณ ได้

ตารางต่อไปนี้ เป็นชนิดของข้อมูลเบื้องต้นที่ VB สนับสนุน สิ่งที่คุณควรสนใจก็คือ ชนิดของข้อมูลที่สามารถเก็บได้, ขอบเขตของข้อมูล, ใช้น้อยความจำระบบเท่าใด, และสัญลักษณ์พิเศษที่ใช้ในกรณีประกาศแบบ Implicit

ชนิดของข้อมูล	สัญลักษณ์พิเศษ	ใช้น้อยความจำ	รายละเอียด
Boolean	ไม่มี	2 Bytes	เก็บค่าทางตรรก มีได้ 2 ค่า คือ true (จริง) , false (เท็จ) โดยที่ VB กำหนดไว้ว่า 0 มีค่าเท่ากับ false และตัวเลขจำนวนเต็มใดที่ไม่เท่ากับ 0 มีค่าเท่ากับ true
Byte	ไม่มี	1 Byte	เก็บค่าเลขจำนวนเต็มตั้งแต่ 0-255 ซึ่งเป็นรหัสแอสกี ASCII
Currency	@	8 Bytes	ใช้เก็บตัวเลขจำนวนจริง มีค่าระหว่าง -922,337,203,685,477.5808 ถึง 922,337,203,685,477.5807 ใช้สำหรับเก็บตัวเลขทางการเงิน โดยเฉพาะ เพราะมีความละเอียดสูง มีทศนิยม 4 ตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Variant	ไม่มี	ขึ้นอยู่กับชนิดข้อมูลที่เก็บ	สามารถเก็บข้อมูลได้ทุกชนิด
Integer	%	2 Bytes	เก็บค่าเลขจำนวนเต็มที่มีค่าระหว่าง -32768 ถึง 32767
Long	&	4 Bytes	ใช้เก็บเลขจำนวนเต็มที่มีค่าระหว่าง -2,147,483,648 ถึง 2,147,483,647
Single	!	4 Bytes	ใช้เก็บตัวเลขจำนวนจริง แยกเป็น 2 กรณี คือ ค่าบวกอยู่ระหว่าง 1.401298E-45 ถึง 3.402823E38 และค่าลบอยู่ระหว่าง -3.402823E38 ถึง -1.401298E-45
Double	#	8 Bytes	ใช้เก็บตัวเลขจำนวนจริง แยกเป็น 2 กรณี คือ ค่าบวกอยู่ระหว่าง 4.94065645841247E-324 ถึง 1.79769313486232E308 ค่าลบอยู่ระหว่าง -1.79769313486232E308 ถึง -4.94065645841247E-324
String	\$	1 ตัว/1 ไบต์ (ascii)	ใช้เก็บตัวอักษร ข้อความ และตัวเลข
Date	ไม่มี	8 Bytes	ใช้สำหรับเก็บวันเดือนปี
Decimal	ไม่มี	12 Bytes	กรณีเลขจำนวนเต็ม มีค่าตั้งแต่ -79,228,162,514,264,337,593,543,950,335 ถึง +79,228,162,514,264,337,593,543,950,335 กรณีเลขทศนิยม มีค่าตั้งแต่ -7.9228162514264337593543950335 ถึง +7.9228162514264337593543950335 ค่าที่น้อยที่สุดที่ไม่เท่ากับศูนย์ที่สามารถเก็บได้คือ 0.000000000000000000000001
Object	ไม่มี	?	ใช้สำหรับแทนอ็อบเจกต์ หรือคอนโทรลที่ VB สนับสนุน

ตารางดังกล่าว เป็นเพียงชนิดของข้อมูลเบื้องต้นเท่านั้น ยังมีชนิดของข้อมูลอื่นๆ อีก ที่ยังไม่สามารถกล่าวไว้ ผู้เขียนจะอธิบายอีกครั้ง ในกรณีที่มีการใช้ข้อมูลชนิดอื่นๆ รูปแบบการประกาศตัวแปรมีดังนี้

1. Dim x As Integer หมายถึง ให้ตัวแปร x เก็บข้อมูลที่เป็นตัวเลขจำนวนเต็มเท่านั้น (มีค่าอยู่ระหว่าง -32768 ถึง 32767)

2. Dim y As String หมายถึง ให้ตัวแปร y เก็บข้อมูลที่เป็นข้อความ หรือตัวอักษรเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้คณาสังเกต VB จะใช้ฟอนต์สีน้ำเงินแทนค่าสวางน หรือค่าสั่ง, สีดำแทนตัวแปร และสีเขียวแทนหมายเหตุ รายละเอียดของข้อมูล แต่ละชนิด มีดังนี้

Boolean

เป็นข้อมูลชนิดบูลีน (Boolean) ใช้หน่วยความจำ 2 bytes สามารถมีค่า True (จริง) หรือ False (เท็จ) เท่านั้น ถ้าคุณกำหนดเป็นตัวเลข โดยที่ 0 หมายถึง False และตัวเลขจำนวนเต็มใดๆ ที่ไม่เท่ากับ 0 หมายถึง True เช่น

Dim a As Boolean กำหนดให้ตัวแปร a เป็นตัวแปรชนิดบูลีน

Byte

เป็นข้อมูลชนิดเลขจำนวนเต็ม มีค่าระหว่าง 0-255 โดยใช้หน่วยความจำ 1 byte มักใช้เก็บรหัสไค้ดแอสกี ที่มีจำนวน 256 ตัวอักษร เช่น

Dim asc As Byte กำหนดให้ตัวแปร asc เป็นตัวแปรชนิด Byte

Currency

เป็นข้อมูลชนิดที่มีค่าตั้งแต่ -922,337,203,685,477.5808 ถึง 922,337,203,685,477.5807 ใช้หน่วยความจำ 8 bytes ใช้สำหรับเก็บตัวเลขทางการเงิน โดยเฉพาะ เพราะมีความละเอียดสูง มีทศนิยม 4 ตำแหน่ง เช่น

Dim Deb As currency

Date

เป็นข้อมูลสำหรับใช้เก็บวันที่ ซึ่งสามารถเก็บได้ตั้งแต่วันที่ 1 January 100 ถึงวันที่ 31 December 9999 และเก็บเวลาได้ตั้งแต่ 0:00:00 ถึง 23:59:59 ในการใช้ข้อมูลชนิดนี้ คุณต้องใช้เครื่องหมาย #...# ล้อมรอบค่าของตัวแปรด้วย ใช้หน่วยความจำ 8 bytes เช่น

Dim dte As Date

dte=#12/31/1999 11:59:59 PM#

Decimal

เป็นข้อมูลชนิดเลขจำนวนจริง ใช้หน่วยความจำ 12 bytes ซึ่งจะแยกออกเป็น 3 กรณีคือ

1. กรณีเลขจำนวนเต็ม มีค่าตั้งแต่ -79,228,162,514,264,337,593,543,950,335 ถึง +79,228,162,514,264,337,593,543,950,335

2. กรณีเลขทศนิยม มีค่าตั้งแต่ -7.9228162514264337593543950335 ถึง +7.9228162514264337593543950335

3. ค่าที่น้อยที่สุดที่ไม่เท่ากับศูนย์ ที่สามารถเก็บได้คือ 0.00000000000000000000000001

ใน Visual Basic 6.0 คุณยังไม่สามารถประกาศ ให้เป็นข้อมูลชนิดนี้ได้โดยตรง คุณต้องทำ 2 ขั้นตอน ดังนี้



1. ประกาศตัวแปร ให้เป็นข้อมูลชนิด Variant ก่อน

2. ใช้ฟังก์ชัน CDec () เพื่อแปลงชนิดของข้อมูลจาก Variant เป็น Decimal ดังนี้

Dim a As Variant

a = CDec("123456789123456789")

Double

ใช้เก็บตัวเลขจำนวนจริง แยกเป็น 2 กรณี คือ ค่าบวกอยู่ระหว่าง 4.94065645841247E-324 ถึง 1.79769313486232E308 ค่าลบอยู่ระหว่าง -1.79769313486232E308 ถึง -4.94065645841247E-324 ใช้หน่วยความจำ 8 bytes เช่น

Dim aBs As Double

Integer

ใช้เก็บเลขจำนวนเต็มที่มีค่าตั้งแต่ -32768 ถึง 32767 ใช้หน่วยความจำ 2 byte เช่น

Dim count As Integer

Long

ใช้เก็บเลขจำนวนเต็มที่มีค่าระหว่าง -2,147,483,648 ถึง 2,147,483,647 ใช้หน่วยความจำ 4 byte เช่น

Dim num As Long

Single

ใช้เก็บตัวเลขจำนวนจริง แยกเป็น 2 กรณี คือ ค่าบวกอยู่ระหว่าง 1.401298E-45 ถึง 3.402823E38 และค่าลบอยู่ระหว่าง -3.402823E38 ถึง -1.401298E-45 ใช้หน่วยความจำ 4 byte เช่น

Dim Intr As Single

String

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นข้อมูลชนิดข้อความ ซึ่งอาจเป็น ได้ทั้งตัวอักษร หรือตัวเลขก็ได้ ขนาดหน่วยความจำที่ใช้จะขึ้นอยู่กับรูปแบบการจัดเก็บ คือ ถ้าตัวอักษรที่เก็บอยู่ในรูปแบบ ASCII จะใช้พื้นที่ 1 ตัวต่อ 1 byte แต่ถ้าเก็บในรูปแบบ Unicode อักษรแต่ละตัวจะใช้พื้นที่ 1 ตัวต่อ 2 bytes ข้อมูลชนิด String สามารถแบ่งออกได้ 2 ประเภท คือ

1. **Variable-length string** หมายถึง เป็นตัวแปร String ที่คุณ ไม่ได้กำหนดขนาดของตัวอักษรไว้ คุณสามารถใส่ตัวอักษร ได้ 65526 ตัวอักษร (ascii) และ 32763 ตัวอักษร (unicode)

2. **Fixed-length string** หมายถึง เป็นตัวแปร String ที่คุณจำกัดขนาดตัวอักษรไว้ คุณจะต้องใช้เลขจำนวนเต็มเท่านั้น ในการจำกัดขนาดตัวอักษร เช่น Dim a As String * 99 หมายถึง ตัวแปร a สามารถเก็บตัวอักษร รวมช่องว่าง ได้ไม่เกิน 99 ตัวอักษร ในการใช้งานตัวแปร String แบบจำกัดขนาดนี้ ให้คุณระวังในเรื่อง ขนาดของข้อความที่จัดเก็บ ถ้าข้อความมีจำนวนตัวอักษร เกินกว่าที่คุณกำหนดไว้ข้อความส่วนที่เกินจะถูกตัดออกไป

การใช้งานตัวแปรแบบใช้สัญลักษณ์พิเศษกำกับ

คุณยังสามารถใช้งานตัวแปร ได้ โดยการ ใช้สัญลักษณ์พิเศษต่อท้ายตัวแปรนั้นๆ ให้คุณดูตารางต่อไปนี้

ชนิดของข้อมูล	สัญลักษณ์พิเศษ
Integer	%
Long	&
Single	!
Double	#
Currency	@
String	\$

การประกาศใช้งานตัวแปรแบบปกติ ถ้าคุณต้องการตัวแปรแบบ Integer คุณจะต้องประกาศ ดังนี้

Dim x As Integer หมายถึง ให้ตัวแปร x เก็บข้อมูลชนิดเลขจำนวนเต็ม

แต่คุณสามารถใช้สัญลักษณ์พิเศษต่อท้ายตัวแปรเพื่อ กำหนดให้ตัวแปรแทนชนิดของข้อมูลที่คุณต้องการ ได้ ซึ่งถ้าคุณ ใช้วิธีนี้ คุณต้องใช้ตลอดทั้ง โปรแกรม เช่น

Dim x% หมายถึง ให้ตัวแปร x เก็บข้อมูลชนิดเลขจำนวนเต็ม Integer เช่นกัน

Dim x& หมายถึง ให้ตัวแปร x เก็บข้อมูลชนิดเลขจำนวนเต็ม Long


Dim x! หมายถึง ให้ตัวแปร x เก็บข้อมูลชนิดเลขทศนิยมแบบ Single (ความละเอียดต่ำ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dim x# หมายถึง ให้ตัวแปร x เก็บข้อมูลชนิดเลขทศนิยมแบบ Double (ความละเอียดสูง)

Dim x@ หมายถึง ให้ตัวแปร x เก็บข้อมูลชนิดเลขทางการเงิน Currency

Dim x\$ หมายถึง ให้ตัวแปร x เก็บข้อมูลชนิดข้อความ String

 **Note** การประกาศตัวแปรแบบปกติ หรือประกาศแบบใช้สัญลักษณ์พิเศษต่อท้าย มีผลเหมือนกันทุกประการ คุณสามารถเลือกใช้ ได้อย่างอิสระ

ขอบเขตของตัวแปร (Scope of Variable)

ตัวแปรที่คุณต้องการ ใช้งานในแอปพลิเคชันหนึ่งๆ จะมีมากมายหลายชนิด ถึงหนึ่งที่คุณควรทราบก็คือ ตัวแปรแต่ละตัวมีขอบเขต การใช้งานเป็นอย่างไร หัวข้อนี้จะเป็นการอธิบาย ข้อยกเว้นและขอบเขตของตัวแปร ใน VB สามารถแบ่งขอบเขตตัวแปรได้ 2 ประเภท คือ

1. ตัวแปรแบบ Local

2. ตัวแปรแบบ Public

ตัวแปรแบบ Local

หมายถึง ตัวแปรที่คุณประกาศขึ้นมา และสามารถเรียกใช้งานได้ ในเฉพาะ โพรซีเจอร์ ที่ประกาศเท่านั้น มักใช้ประกาศตัวแปรที่คุณต้องการ ใช้ชั่วคราว หรือต้องการ ใช้แค่ใน โพรซีเจอรันๆ เช่น

```
Private Sub Command1_Click()
```

```
Dim x As Integer
```

```
Dim y As Long
```

```
x=x+y
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
x=x*y
```

```
End Sub
```

จากตัวอย่างข้างต้น ตัวแปร x แทนชนิดข้อมูล Integer และ y แทนข้อมูล Long คุณสามารถเรียกใช้งานตัวแปรทั้ง 2 ได้เฉพาะใน โพรซีเจอร์ Command1_Click() เท่านั้น กรณีที่คุณกำหนดให้ VB ไล่ประโยค Option Explicit ถ้าคุณเรียกใช้งานตัวแปรดังกล่าว ใน โพรซีเจอร์ Command2_Click() VB จะแสดงข้อความ Variable not defined ซึ่งหมายถึง ตัวแปร x และ y ใน โพรซีเจอร์ Command2_Click() ไม่ได้มีการประกาศตัวแปร จึงไม่สามารถนำตัวแปร x และ y มาใช้ได้ เพราะตัวแปร x และ y ที่คุณประกาศไว้ มีขอบเขตการใช้งานในระดับ Local หรือแค่ใน โพรซีเจอร์ที่ประกาศไว้เท่านั้น แต่ถ้าในกรณีที่ คุณไม่ได้กำหนดให้ VB ไล่ข้อความ Option Explicit ตัวแปร x และ y คุณสามารถเรียกใช้งานได้ ใน โพรซีเจอร์ Command2_Click() แต่ตัวแปรทั้ง 2 จะเป็นข้อมูลชนิด Variant ซึ่งคุณ ไม่มีความจำเป็นต้องใช้ และตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แปรชนิด Local นี้จะถูกยกเลิก หรือถูกทำลาย ซึ่งหมายถึง คืนหน่วยความจำให้กับระบบ เมื่อจบ โพรซีเจอร์ (End Sub) ทันที

แต่ถ้าคุณต้องการ ใช้งานตัวแปร ใน โพรซีเจอร์ Command2_Click () โดยใช้ชื่อ x และ y เหมือนกัน VB จะถือว่าเป็นตัวแปรคนละตัว และจะจองหน่วยความจำให้แต่ละตัว ไม่เกี่ยวข้องกันแต่อย่างใด เช่น

```
Private Sub Command1_Click()
```

```
Dim x As Integer
```

```
Dim y As Long
```

```
x=x+y
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
Dim x As Long
```

```
Dim y As Single
```

```
x=x*y
```

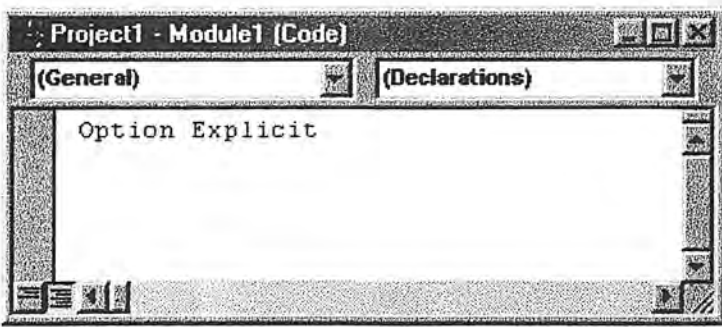
```
End Sub
```

จากตัวอย่างข้างต้น คุณยังสามารถประกาศให้ตัวแปร x และ y ให้แทนชนิดข้อมูลต่างกัน ได้อีกด้วย เหตุผลก็คือ ตัวแปรทั้ง 4 ตัว เป็นแบบ Local นั่นเอง

ตัวแปรแบบ Public

หมายถึง ตัวแปรที่คุณประกาศใช้งานแล้ว คุณสามารถเรียกใช้งาน ได้ทั้ง โปรเจกต์ โพรซีเจอร์ใด ฟอรัมใดก็ได้ อาจเรียกได้อีกอย่างหนึ่งว่า ตัวแปรร่วมแบบ Public หรือตัวแปรระดับ โมดูล มักจะใช้ใน กรณีที่ต้องเก็บค่าตัวแปร ที่ต้องใช้ทุกฟอรัม หรือทุกโพรซีเจอร์ เช่น กรณีที่คุณเขียน โปรแกรม ชื่อ-ขายสินค้า ไม่ว่าจะเป็นการซื้อ หรือขาย ภาษีก็เท่ากับ 7 % เสมอ (สมมติ) คุณไม่ต้องประกาศตัวแปรภาษี ทุกโพรซีเจอร์ หรือถ้าเป็น โปรแกรมเงินเดือน จะต้องมีการหักเงินประกันสังคม 3% ของพนักงานทุกคน คุณก็ไม่ ต้องประกาศตัวแปรเงินประกันสังคม ทั้ง 2 กรณี คุณสามารถเลือกใช้ตัวแปรแบบ Public ได้ มีวิธีดังนี้

1. ให้คุณเลือกเมนู Project/Add Module VB จะเพิ่มไฟล์ โมดูล .bas (Module) เข้ามาใน โปรเจกต์ ให้คุณสังเกตที่หน้าต่าง Project Explorer จะมีรายการ Module1 เพิ่มเข้ามาใน โปรเจกต์ ดังรูป



รูปที่ 3-2 หน้าต่างโมดูล (Module)

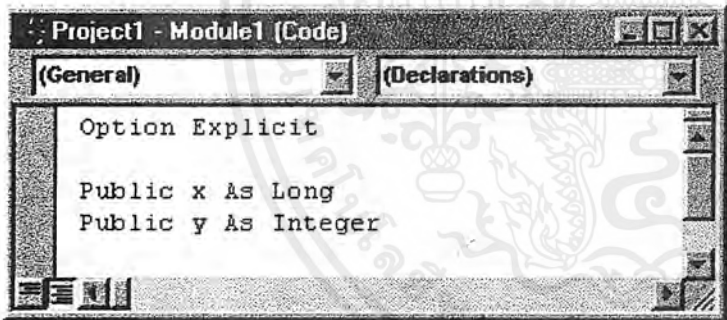
ไฟล์โมดูล (Module) เป็นพื้นที่พิเศษที่ใช้สำหรับประกาศตัวแปร, ฟังก์ชันที่คุณเขียนขึ้นมา, General Procedure, หรือฟังก์ชัน Windows API ฯลฯ เท่านั้น คุณไม่สามารถเขียนโค้ดในไฟล์โมดูลนี้ได้ คุณสามารถสังเกตได้โดยดูที่ช่อง Object list box และช่อง Event list box ไม่มีรายการอ็อบเจกต์ หรือรายการเหตุการณ์แต่อย่างใด



Note

ในบางครั้ง คุณอาจการประกาศค่าคงที่ เพื่อแทนตัวแปรแบบ Public ก็ได้

2. คุณต้องประกาศตัวแปรโดยใช้คำสั่ง Public แทนคำสั่ง Dim ดังรูป ให้คุณระวังในจุดนี้ไว้ เพราะเนื่องจากว่า คุณอาจมีความเคยชินในการใช้คำสั่ง Dim เสมอในการประกาศตัวแปร เพราะถ้าคุณใช้คำสั่ง Dim ในการประกาศตัวแปร ที่ไฟล์โมดูล คุณไม่สามารถใช้ตัวแปรแบบ Public ได้ และจะเกิดข้อความ Variable not defined เข้าใจ :-)



รูปที่ 3-3 แสดงการประกาศตัวแปรแบบ Public ในไฟล์โมดูล

เมื่อคุณประกาศตัวแปรแบบ Public แล้ว ในโปรซีเจอร์ Command1_Click () และ Command2_Click () คุณสามารถใช้งานตัวแปรทั้ง 2 ได้ โดยไม่ต้องประกาศตัวแปรอีกแต่อย่างใด

```
Private Sub Command1_Click()
```

```
    x=x+y
```

```
End Sub
```

```
Private Sub Command2_Click()
```

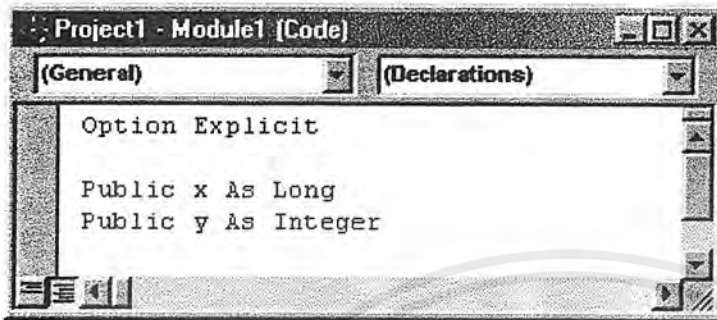
```
    x=x*y
```

```
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าในกรณีที่คุณมีการประกาศตัวแปรแบบ Local และตัวแปรแบบ Public โดยใช้ชื่อเดียวกัน VB จะถือว่าเป็นตัวแปรคนละตัว คนละแบบ และจะจองหน่วยความจำ ให้กับตัวแปรทั้ง 2 ไม่เกี่ยวข้องกันแต่อย่างใด และจะให้ความสำคัญกับตัวแปรแบบ Local ก่อนตัวแปรแบบ Public เสมอ เช่น

ให้คุณทดสอบโดยการประกาศตัวแปร ในไฟล์คลด ดังรูป



รูปที่ 3-4 ตัวอย่างทดสอบการประกาศตัวแปรแบบ Public และแบบ Local

จากนั้นในแต่ละ โพรซีเจอร์ ให้คุณประกาศตัวแปร โดยใช้ชื่อเหมือนกัน แต่ต่างชนิดข้อมูลกัน ดัง
นี้

```
Private Sub Command1_Click()
    x = x * y
End Sub

Private Sub Command2_Click()
    Dim x As String
    x = x + y
End Sub
```

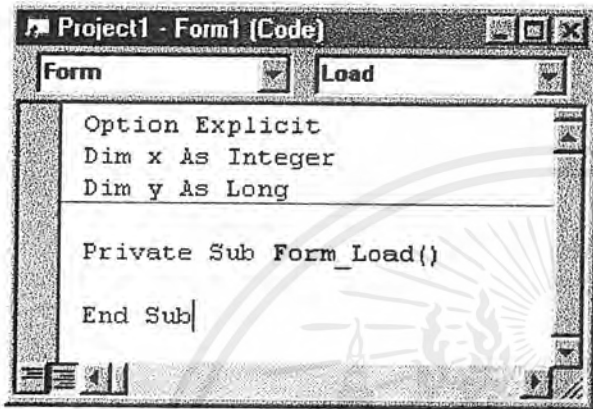
ให้คุณทดลองรัน แล้วคุณลองคลิกปุ่ม Command2 ปรากฏว่าจะเกิดข้อความ Run-time '13' Type mismatch ซึ่งหมายถึง ข้อมูลผิดประเภท เนื่องจากว่า ตัวแปร x (แบบ Public) ที่คุณประกาศไว้ใน โมดูล เป็นข้อมูลตัวเลขจำนวนเต็ม Long แต่ตัวแปร x (แบบ Local) ที่คุณประกาศไว้ใน โพรซีเจอร์ Command2_Click () เป็นชนิดข้อความ String เป็น ไปไม่ได้ที่ ตัวเลข+ตัวเลข จะเท่ากับข้อความ ถึงแม้ว่าจะเป็นข้อความที่เป็นตัวเลข แต่เป็นตัวเลขที่ไม่สามารถนำไปคำนวณได้ ให้คุณระวังในจุดนี้ให้ดี และขอให้คุณใช้ตัวแปรแบบ Public เท่าที่จำเป็นเท่านั้น

การใช้งานตัวแปรร่วมระดับโพรซีเจอร์

หมายถึง เป็นตัวแปรที่คุณประกาศแล้ว สามารถใช้งานได้ทุกๆ โพรซีเจอร์ แต่ไม่สามารถเรียกใช้ ในระดับ โปรเจกต์แบบ Public ได้ ซึ่งถ้าเรียงลำดับขอบเขต ความสามารถในการเรียกใช้งานแล้ว ตัวแปร

แบบ Public จะมีขอบเขตกว้างที่สุดก็คือ คุณสามารถเรียกใช้งาน ที่จุดใดก็ได้ใน โปรเจกต์ รองลงมาก็คือ ตัวแปรร่วมระดับ โพรซีเจอร์ ซึ่งคุณสามารถเรียกใช้งาน ได้ในทุกๆ โพรซีเจอร์ที่อยู่ในฟอร์มที่ประกาศเท่านั้น และท้ายสุดคือตัวแปรแบบ Local ก็คือ คุณสามารถเรียกใช้งานตัวแปรแค่ ใน โพรซีเจอร์ ที่ประกาศไว้เท่านั้น วิธีการประกาศตัวแปรร่วมระดับ โพรซีเจอร์ ก็คือ

1. ให้คุณดับเบิลคลิก ที่ฟอร์มเพื่อเปิดหน้าต่าง editor แล้วเลือก General ใน Object list box ประกาศตัวแปร โดยการ ใช้คำสั่ง Dim ดังรูป



รูปที่ 3-5 ประกาศตัวแปรระดับโพรซีเจอร์ ใน Form1

เนื่องจากว่า ในแอปพลิเคชันหนึ่งๆ อาจจะมี 2 ฟอร์มขึ้นไป คุณอาจใช้ตัวแปรร่วมระดับโพรซีเจอร์ เพื่อจำกัดขอบเขตการเรียกใช้งานก็ได้ โดยตัวแปรร่วมระดับโพรซีเจอร์ จะมีขอบเขตเรียกใช้งานแค่ในฟอร์มที่ประกาศไว้เท่านั้น ยกตัวอย่างเช่น ถ้าคุณเขียน โปรแกรมซื้อ-ขาย และ โปรแกรมคิดเงินเดือน โดยกำหนดให้ โปรแกรมซื้อ-ขาย อยู่ใน Form1 และ โปรแกรมคิดเงินเดือนให้อยู่ใน Form2 ซึ่งทั้ง 2 โปรแกรม ยังคงอยู่ใน โปรเจกต์เดียวกัน เป็น โปรเจกต์สารพัดประโยชน์จริงๆ :-) คุณอาจใช้ตัวแปรร่วมระดับโพรซีเจอร์ เพื่อประกาศอัตราภาษี 7% ซึ่งใช้ร่วมกันทุกโพรซีเจอร์ เฉพาะใน Form1 เท่านั้น ส่วนในโปรแกรมเงินเดือน ไม่มีการใช้ภาษี 7% แต่อย่างไร ในทางกลับกัน คุณต้องการประกาศอัตราเงินประกันสังคม 3% ซึ่งใช้ร่วมกันทุกโพรซีเจอร์เฉพาะใน Form2 เท่านั้น โปรแกรมซื้อ-ขาย ไม่ต้องการ อัตราเงินประกันสังคมแต่อย่างใด เพื่อให้คุณเห็นภาพ ได้ชัดเจนมากขึ้น ให้คุณทดลองทำดังนี้

ใน Form1 ประกอบไปด้วย 3 โพรซีเจอร์ ให้คุณใส่คอนโทรล CommandButton 3 ตัว และเขียนโค้ดดังนี้

```
Private Sub Command1_Click()
    a = a + b
End Sub
```

```
Private Sub Command2_Click()
```

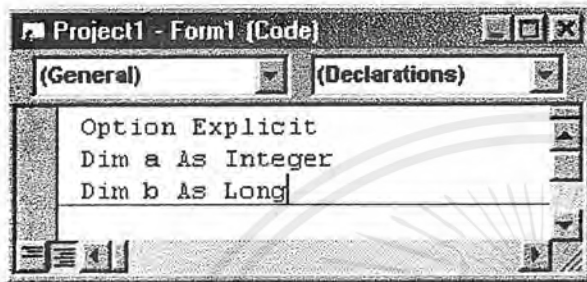
```

a = a * b
End Sub

Private Sub Command3_Click()
    Form2.Show 'Display Form2
End Sub

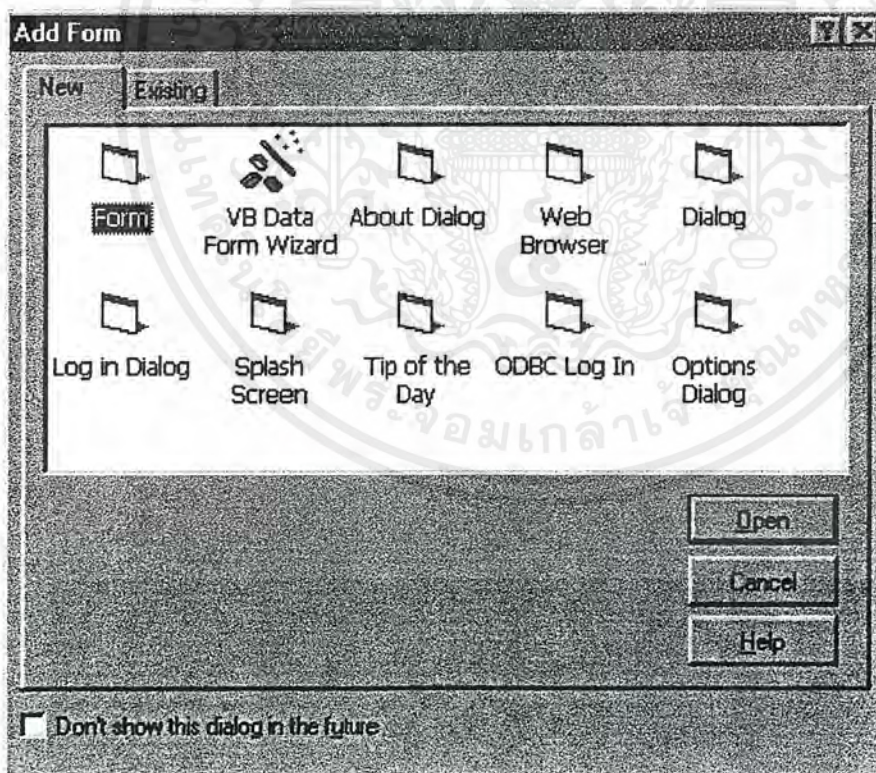
```

ในส่วนของ Form1 ให้คุณประกาศตัวแปรระดับโปรซีเจอร์ ดังนี้



รูปที่ 3-6 ประกาศตัวแปร *a* และ *b* ซึ่งเป็นตัวแปรร่วมระดับโปรซีเจอร์ ใน *Form1*

2. ให้คุณเพิ่ม Form2 เข้ามา โดยเลือกเมนู Project/Add Form เลือกไอคอน Form ดังรูป



รูปที่ 3-7 แสดงไอคอน *Form* จากคำสั่ง *Add Form*

จากนั้นใส่คอนโทรล CommandButton 2 ตัว ที่ Form2 และเขียนโค้ด ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Command1_Click()
```

```
    a = a + b
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
    a = a * b
```

```
End Sub
```

ให้คุณรันโปรแกรมที่ทดลองคลิกที่ปุ่ม Command1 และ Command2 ของ Form1 ปรากฏว่าไม่มีอะไรเกิดขึ้น เพราะเป็นการคำนวณธรรมดา ให้คุณคลิกที่ปุ่ม Command3 แล้ว Form2 จะปรากฏขึ้นมา ให้คุณคลิกที่ปุ่ม Command1 หรือ Command2 ของ Form2 ปรากฏว่า จะมีข้อความ Variable not defined ปรากฏขึ้นมา

จากการทดลองดังกล่าว สรุปได้ว่า ตัวแปร a และ b ซึ่งถูกประกาศให้เป็นตัวแปรร่วมระดับโปรซีเจอร์ มีขอบเขตการใช้งานแค่ใน Form1 เท่านั้น คุณสามารถเรียกใช้งานได้ในโปรซีเจอร์ Command1_Click () และ Command2_Click () ที่อยู่ใน Form1 เท่านั้น แต่คุณไม่สามารถใช้งานตัวแปร a และ b ใน Form2 ได้เลย เพราะตัวแปร a และ b มีข้อจำกัดการใช้งาน แค่ในฟอร์มที่ประกาศเท่านั้น ให้คุณลองเปลี่ยนจากการประกาศใน Form1 เป็นประกาศใน โมดูล ปรากฏว่า จะไม่มีปัญหาในการใช้งานตัวแปร a และ b แต่อย่างใด

การตั้งชื่อตัวแปรแบบบอกชนิดและขอบเขตของตัวแปร

ในการเขียนแอปพลิเคชันต่างๆ ไป คุณจะต้องมีการเรียกใช้งานตัวแปรต่างๆ มากมาย ซึ่งคุณไม่สามารถที่จะจดจำได้อย่างถูกต้องทั้งหมด มีแนวทางการตั้งชื่อที่โปรแกรมเมอร์มืออาชีพ เพื่อบอกชนิด หรือขอบเขตของตัวแปร เมื่อเวลาที่ย้อนกลับมาดู source code สามารถที่จะศึกษาได้ source code ดังกล่าวได้ไม่ยากนัก ซึ่งจะอำนวยความสะดวกเป็นอย่างยิ่ง เช่น ถ้าคุณมีการประกาศตัวแปรเพื่อ ใช้สำหรับเก็บค่าเงินเดือน คุณอาจตั้งชื่อดังนี้ก็ได้

```
Dim iSalary As Integer
```

ตัว i สื่อให้รู้ว่า ตัวแปร iSalary เป็นข้อมูลชนิด Integer

```
Dim strUser As String
```

ตัว str สื่อให้รู้ว่า ตัวแปร strUser เป็นข้อมูลชนิด String หรือถ้าคุณระบุขอบเขตของตัวแปรด้วยก็ได้ เช่น

```
Dim iLocSalary As Integer
```

ตัว i สื่อให้รู้ว่า เป็นตัวแปรชนิด Integer โดยมีขอบเขตระดับ Local

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


Dim strModName As String

ตัว str ถือให้รู้ว่า เป็นตัวแปรชนิด String โดยมีขอบเขตการใช้งานระดับ Module

Dim lngLocEmployee As Long

ตัว lng ถือให้รู้ว่า เป็นตัวแปรชนิด Long โดยมีขอบเขตการใช้งานระดับ Local เป็นต้น

การตั้งชื่อแบบนี้ เหมาะกับแอปพลิเคชันที่ต้องมีการใช้งานตัวแปรจำนวนมากๆ และคุณต้องการให้



Note source code มีระเบียบ มีความเป็นมืออาชีพยิ่งขึ้น และสามารถย้อนกลับมาศึกษา source code นี้ได้
อย่างไม่ยากเย็น คุณจะพบ source code ที่มีการตั้งชื่อตัวแปรแบบนี้ เมื่อมีการเรียกใช้งานกลุ่มฟังก์ชัน
Windows API ร่วมด้วย หรือเป็นตัวอย่างแอปพลิเคชันที่มีขนาดใหญ่หลายๆ

ตัวแปรอาร์เรย์ (Array)

ตัวแปรอาร์เรย์ เป็นกลุ่มของตัวแปรที่คูณประกาศขึ้นมา โดยใช้ชื่อของตัวแปรแต่ละตัวเหมือนกัน ข้อแตกต่างของตัวแปรแต่ละตัว จะใช้ค่า *Index* ในการอ้างอิง ตัวแปรชนิดนี้มีประโยชน์ในแง่ของการเก็บข้อมูลที่คล้ายๆ กัน เป็นชุดๆ โดยที่คุณ ไม่ต้องห่วงเรื่องการตั้งชื่อตัวแปรแต่ละตัว เพราะมีชื่อเหมือนกัน ต่างกันที่ค่า *Index* จะทำให้การเรียกใช้งานตัวแปรเหล่านั้นง่าย และสะดวกกว่า ซึ่งตัวแปรที่ประกาศเป็นแบบอาร์เรย์นี้ แต่ละตัวจะเรียกว่า สมาชิกตัวที่... มีรูปแบบการประกาศดังนี้

Dim intCnt () As Integer

จะเห็นได้ว่า มีความคล้ายกับการประกาศตัวแปรแบบปกติ ที่เพิ่มขึ้นมาก็คือ เครื่องหมายวงเล็บต่อท้ายชื่อตัวแปร ซึ่งหมายถึง คุณต้องการใช้งาน ตัวแปรแบบอาร์เรย์ ซึ่งสมาชิกแต่ละตัวในตัวแปรอาร์เรย์ จะต้องเป็นข้อมูลชนิดเลขจำนวนเต็ม Integer เท่านั้น ตัวแปรอาร์เรย์มี 2 ชนิด คือ

- 1.ตัวแปรอาร์เรย์แบบสแตติก (Static Arrays)
- 2.ตัวแปรอาร์เรย์แบบไดนามิก (Dynamic Arrays)

ตัวแปรอาร์เรย์แบบสแตติก (Static Arrays)

เป็นอาร์เรย์ที่มีจำนวนสมาชิกที่แน่นอน ซึ่งคุณต้องระบุจำนวนสมาชิก โดยใช้เลขจำนวนเต็ม เข้าไปในวงเล็บด้วย ขอให้คุณจำลองหน่วยความจำในเครื่องว่า เป็นห้องๆ ติดกัน แต่ละห้องสามารถเก็บข้อมูลได้ 1 ตัวอักษร VB จะจองจำนวนห้องเพื่อเก็บข้อมูล ให้เท่ากับจำนวนตัวแปรอาร์เรย์ที่คุณระบุไว้ เช่น

Dim x(5) As Integer หมายถึง ตัวแปร x เป็นตัวแปรอาร์เรย์ชนิดสแตติก ที่ใช้เก็บเลขจำนวนเต็ม Integer โดยที่มีสมาชิกทั้งสิ้น 5 ตัว การใช้งานตัวแปรแต่ละตัว โดยการใช้ชื่อ x(0) เป็นตัวที่ 1 หรือเรียกอีกอย่างหนึ่งว่า ขอบเขตล่าง ไปจนถึง x(4) เป็นตัวที่ 5 หรือเรียกอีกอย่างหนึ่งว่า *ขอบเขตบน* ซึ่งจะมีความสำคัญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นอย่างยิ่ง ในการใช้งานตัวแปรแบบอาร์เรย์ โดยปกติแล้วสมาชิกตัวแรกของตัวแปรแบบอาร์เรย์ จะมีค่าที่ 0 เสมอ

`Dim x(1 To 10) As Long` หมายถึง ประกาศตัวแปรอาร์เรย์ ที่มีสมาชิก 10 ตัว แต่ละตัวแทนข้อมูลชนิด Long โดยที่มีขอบเขตล่างเท่ากับ 1 สมาชิกตัวแรกคือ $x(1)$ ไปจนถึง $x(10)$ โดยที่มีขอบเขตบนเท่ากับ 10 เป็นต้น

ค่าที่อยู่ในวงเล็บจะเรียกว่า ค่า Index ทำให้ตัวแปร x แต่ละตัวมีความแตกต่างกันนั่นเอง และในการใช้งาน ขอให้ผู้ใช้ตัวแปรอาร์เรย์ชนิดนี้ ให้น้อยที่สุดเท่าที่จะเป็นไปได้ เนื่องจากว่า เมื่อคุณประกาศตัวแปรอาร์เรย์แบบสแตติกแล้ว VB จะจองหน่วยความจำเท่ากับจำนวนสมาชิกทันที ถึงแม้ว่า คุณจะไม่มีการใช้งานตัวแปรก็ตาม หรือใช้งานตัวแปร ไม่ครบทุกตัว คุณจะสูญเสียหน่วยความจำในส่วนนี้ไป เช่นกรณีข้างต้น สมมติว่า คุณ ใช้งานเพียง 3 ตัวแปร $x(0)$, $x(1)$ และ $x(2)$ คุณต้องเสียหน่วยความจำไป 5 ส่วนซึ่งไม่มีความจำเป็นแต่อย่างใด ขอให้ผู้ใช้อาร์เรย์ชนิดไดนามิกจะเหมาะสมกว่า

ตัวแปรอาร์เรย์แบบไดนามิก (Dynamic Arrays)

เป็นตัวแปรอาร์เรย์ที่ใช้สำหรับกลุ่มของตัวแปร ที่คุณไม่ทราบจำนวนที่แน่นอน หรือคุณทราบแต่ต้องการใช้อาร์เรย์แบบไดนามิก เพื่อประหยัดทรัพยากรระบบ โดยที่ตัวแปรแบบอาร์เรย์ จะปรับขนาดจำนวนสมาชิกให้เท่ากับ จำนวนตัวแปรที่คุณต้องการ ใช้ในขณะนั้น จะเห็นได้ว่า มีการใช้ทรัพยากรระบบเมื่อต้องการเท่านั้น ซึ่งเป็นข้อดีเป็นอย่างยิ่ง มีรูปแบบการประกาศดังนี้

```
Dim intCnt( ) As Integer
```

ข้อแตกต่างของตัวแปรอาร์เรย์แบบไดนามิก คือ คุณไม่ต้องระบุจำนวนสมาชิกในวงเล็บ จะเห็นได้ว่ามีความยืดหยุ่นมากกว่า ตัวแปรอาร์เรย์แบบสแตติก แต่ถ้าในขณะรัน คุณต้องการกำหนดจำนวนสมาชิกที่แน่นอน คุณต้องใช้คำสั่งดังนี้

```
ReDim [Preserve] varname(subscripts) As type
```

คำสั่ง `ReDim` หมายถึง คุณต้องการกำหนดจำนวนสมาชิกในตัวแปรอาร์เรย์ใหม่

คำสั่ง `Preserve` หมายถึง คุณต้องการเก็บข้อมูลเดิม ที่ตัวแปรอาร์เรย์ดังกล่าวเก็บไว้

ตัวแปร `varname` หมายถึง ตัวแปรอาร์เรย์ที่คุณต้องการกำหนดจำนวนสมาชิก

ตัวแปร `subscripts` หมายถึง ขนาดจำนวนสมาชิกที่คุณต้องการ จะต้องเป็นเลขจำนวนเต็ม

ตัวแปร `type` หมายถึง ชนิดของข้อมูลที่คุณต้องการ ให้ตัวแปรอาร์เรย์ทดแทน เช่น

สมมติว่าคุณประกาศตัวแปรอาร์เรย์ `Dim intCnt() As Integer` ต่อมาคุณต้องการระบุขนาดจำนวนสมาชิกของตัวแปรดังกล่าว ให้คุณทำดังนี้

`ReDim intCnt(10) As Integer`

แต่ถ้าคุณต้องการรักษาค่าที่ตัวแปรอาร์เรย์ `intCnt()` เก็บไว้ด้วย คุณต้องเพิ่มเติมดังนี้

`ReDim Preserve intCnt(10) As Integer`

จากกรณีทีกล่าวมา เป็นการสร้างตัวแปรอาร์เรย์แบบ 1 มิติ คุณสามารถเพิ่มเติมให้ตัวแปรอาร์เรย์มีประสิทธิภาพมากยิ่งขึ้น ด้วยการ ใช้ ตัวแปรอาร์เรย์แบบหลายมิติ

การสร้างตัวแปรอาร์เรย์มากกว่า 1 มิติ

มีว่าตัวแปรแบบอาร์เรย์ จะสามารถเก็บค่าตัวแปรเป็นชุดได้แล้วก็ตาม คุณยังสามารถขยายขีดความสามารถของตัวแปรอาร์เรย์ได้อีก โดยที่ขอให้คุณจำลองตัวแปรอาร์เรย์ให้มีลักษณะช่องสี่เหลี่ยมติดๆ กัน ถ้าคุณต้องการใช้งานห้องใด ที่ห้องนั้นๆ จะมีเลขที่ห้องอยู่ ดังรูป

x(0)	x(1)	x(2)	x(3)	x(n)
------	------	------	------	------

เมื่อคุณต้องการใช้งานตัวแปรอาร์เรย์ตัวใด คุณจะ ใช้ความแตกต่างของค่า subscript เพื่ออ้างถึงตัวแปรนั้นๆ กรณีนี้เป็นอาร์เรย์ชนิด 1 มิติ แต่ถ้าคุณต้องการเพิ่มแถวเข้าไปอีก จะเกิดอะไรขึ้น ดังรูป

x(0) (0)	x(0) (1)	x(0) (2)	x(0) (3)	x(0) (n)
x(1) (0)	x(1) (1)	x(1) (2)	x(1) (3)	x(1) (n)

ตัวอย่างข้างต้น คุณยังสามารถเพิ่มเติมมิติ (Dimensions) เข้าไปได้อีก โดยที่ค่า subscript ไม่มีทางซ้ำกันได้ ดังนั้น จึงเป็นแนวความคิดที่สร้างตัวแปรอาร์เรย์ที่มีมิติที่ 2 ขึ้นมา ซึ่งคุณต้องทำดังนี้

`Dim x(1 To 5,1 To 4) As Long`

ข้อสังเกต จะใช้เครื่องหมาย , เพื่อเพิ่มมิติขึ้นมา จากกรณีนี้ สามารถจำลองตัวแปรอาร์เรย์ 2 มิติได้ดังนี้

x(1,1)	x(1,2)	x(1,3)	x(1,4)
x(2,1)	x(2,2)	x(2,3)	x(2,4)
x(3,1)	x(3,2)	x(3,3)	x(3,4)
x(4,1)	x(4,2)	x(4,3)	x(4,4)
x(5,1)	x(5,2)	x(5,3)	x(5,4)

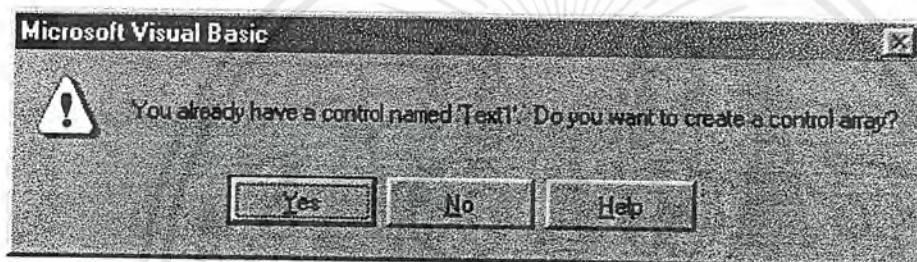
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณจะได้ตัวแปรทั้งสิ้น 20 ตัว ที่มีความแตกต่างกัน ซึ่งคุณอาจจะใช้แบบ Dim x(19) As Long ก็ได้ ไม่ผิดแต่อย่างใด นั่นคือ คุณจะได้ x(0) ถึง x(19)

การใช้งานคอนโทรลอาร์เรย์

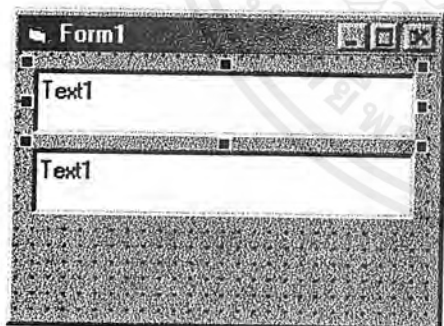
ในการใช้งานตัวแปร เราสามารถใช้งานในลักษณะแบบปกติ และใช้งานแบบอาร์เรย์ ซึ่งหมายถึงการใช้งานตัวแปรชุด ที่มีการตั้งชื่อเหมือนกัน ต่างกันตรงที่ตัวแปรแต่ละตัวมีลำดับ (ค่า index) ต่างกัน ซึ่งแนวความคิดนี้เอง สามารถนำมาใช้กับคอนโทรลได้เช่นกัน นั่นคือ เราสามารถใช้งานคอนโทรลประเภทเดียวกัน โดยที่ตั้งชื่อเหมือนกัน ต่างกันตรงที่คอนโทรลแต่ละตัว จะมีค่า index ที่ทำให้คอนโทรลแต่ละตัวไม่เหมือนกันนั่นเอง ซึ่งก็คือ การใช้งานคอนโทรลแบบอาร์เรย์ (Control Array)

ให้คุณทดลองโดยการวางคอนโทรล TextBox 1 ตัวบนฟอร์ม จากนั้นให้คุณคลิกที่ตัวคอนโทรล TextBox เลือกคำสั่ง Copy แล้วเลือกคำสั่ง Paste ซึ่งคุณจะได้พบกับไอคอนบล็อกซ์ ดังรูป



รูปที่ 3-8 แสดงไอคอนบล็อกซ์ยืนยันการใช้งานคอนโทรลอาร์เรย์

ไอคอนบล็อกซ์ดังกล่าว เป็นการยืนยันว่า คุณต้องการใช้งานคอนโทรล TextBox ในรูปแบบอาร์เรย์ ซึ่งเมื่อคุณตอบ Yes แล้ว คอนโทรล TextBox จะถูกใช้งานในลักษณะคอนโทรลอาร์เรย์



รูปที่ 3-9 แสดงการใช้งานคอนโทรล TextBox แบบคอนโทรลอาร์เรย์

จากรูปที่ 3-9 คุณจะพบว่า ชื่อของคอนโทรล TextBox ทั้ง 2 ตัวเหมือนกัน ให้คุณสังเกตได้ที่คุณสมบัติ Name ในหน้าต่างคุณสมบัติ รวมถึงชื่อของคอนโทรล TextBox ที่ปรากฏอยู่ในช่อง list box ในหน้าต่างคุณสมบัติด้วย ซึ่งจะเห็นว่า มีตัวเลขอยู่ในวงเล็บ Text1(0) สำหรับคอนโทรล TextBox ตัวแรก และ Text1(1) สำหรับคอนโทรล TextBox ตัวที่ 2 ดังรูป



รูปที่ 3-10 แสดงข้อแตกต่างของคอนโทรล TextBox แต่ละตัว เมื่อใช้งานในลักษณะของคอนโทรลอาร์เรย์

เมื่อคุณต้องการอ้างอิงถึงคอนโทรล TextBox แต่ละตัว จึงจำเป็นที่จะต้องมีการใส่ค่า index ประจำตัวของคอนโทรล TextBox แต่ละตัวเข้าไปด้วย เช่น ถ้าคุณต้องการเพิ่มข้อความเข้าไปที่คอนโทรล TextBox ตัวที่ 1 ให้คุณเขียนโค้ดดังนี้

```
Text1(0).Text = "Your Text"
```

สำหรับการใช้งานคอนโทรลอาร์เรย์นี้ ถ้าคุณมีการแก้ไขคุณสมบัติต่างๆ ของคอนโทรลต้นแบบแล้ว คุณยังสามารถ copy คุณสมบัติต่างๆ ที่คุณได้แก้ไขไว้กับคอนโทรลต้นแบบนั้นๆ นำมาใช้กับคอนโทรลอาร์เรย์ตัวต่อๆ ไปได้อีกด้วย ซึ่งจะช่วยให้คุณสามารถแก้ไขคุณสมบัติต่างๆ กันได้อย่างรวดเร็วอีกด้วย

การสร้างชนิดของตัวแปรขึ้นใช้เอง (User-defined data type)

นอกจากชนิดของข้อมูลที่ VB สนับสนุนแล้ว คุณยังสามารถนำชนิดของข้อมูลพื้นฐานดังกล่าวทั้งหมด มาสร้างเป็นชนิดข้อมูล ตามที่คุณต้องการ ได้อีกด้วย โดยมีรูปแบบดังนี้

	คุณต้องใส่ ซึ่งเป็นชื่อของชนิดข้อมูลที่กำหนดขึ้นมาใหม่ โดยสร้างจากชนิดข้อมูลพื้นฐานที่ VB สนับสนุน
<i>Elementname</i>	ชื่อของตัวแปร ซึ่งเป็นส่วนหนึ่งของชนิดข้อมูลที่สร้างขึ้นมา
<i>Subscripts</i>	คุณยังสามารถสร้างตัวแปรแบบอาร์เรย์ ที่เป็นส่วนหนึ่งของชนิดข้อมูลที่กำหนดมาได้อีกด้วย
<i>Type</i>	ชนิดของข้อมูลพื้นฐานแต่ละตัว ที่เป็นองค์ประกอบของข้อมูลที่สร้างขึ้นมา เช่น Byte, Boolean, Integer, Long, Currency, Single, Double, Date, String (ไม่จำกัดความยาว), String * length (จำกัดความยาว), Object, Variant รวมถึงอาร์เรย์ด้วย

ตัวอย่างเช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Type Student

FirstName As String

LastName As String

ID As Integer

End Type

เป็นการสร้างชนิดข้อมูลขึ้นมาใหม่ โดยให้ชื่อว่า Student เมื่อคุณต้องการใช้งานตัวแปรชนิดนี้ คุณต้องประกาศดังนี้

Dim Input As Student

ส่งผลให้ตัวแปร Input มีโครงสร้างเหมือนกับชนิดข้อมูล Student และเมื่อคุณต้องการอ้างถึงตัวแปรต่างๆ ที่อยู่ใน โครงสร้างนี้ ให้คุณทำดังนี้

```
Input.FirstName = "Suphachai"
```

```
Input.LastName = "Somphanit"
```

```
Input.ID = 12345678
```

คุณจะต้องระบุข้อมูลให้ตรงกับชนิดของข้อมูลแต่ละตัว ใน โครงสร้างนั้นๆ ด้วย ซึ่งชนิดของข้อมูลต่างๆ ที่ประกอบขึ้นมา จะมีขอบเขต และความสามารถ เหมือนกับการ ใช้งานชนิดของข้อมูลตามปกติ การประกาศค่าคงที่ (Constant)

ในการประกาศค่าคงที่ คุณต้องใช้คำสั่ง Const เพื่อสร้างค่าคงที่ โดยปกติแล้ว จะใช้ค่าคงที่เมื่อคุณทราบจำนวนที่แน่นอน และค่าดังกล่าว คุณ ไม่ต้องการ ให้มีการเปลี่ยนแปลง ถ้ามีการเปลี่ยนแปลงค่าคงที่จะเกิดข้อผิดพลาด มีรูปแบบการประกาศดังนี้

```
[Public | Private] Const constname [As type] = expression
```

ความหมายของแต่ละส่วนมีดังนี้

คำสั่ง	รายละเอียด
Public	Optional หมายถึง ให้ค่าคงที่สามารถเรียกใช้งานได้ ทุกฟอร์ม ทุกโทรมซีเคอร์ คุณต้องประกาศในไฟล์โมดูลเท่านั้น ซึ่งจะมีความหมายเช่นเดียวกับ ตัวแปรระดับ Public หรือระดับโมดูลนั่นเอง
Private	Optional หมายถึง ให้ค่าคงที่สามารถใช้งานได้เฉพาะในฟอร์ม หรือ โมดูลที่ประกาศเท่านั้น ซึ่งเป็นค่าปกติที่ VB ตั้งไว้อยู่แล้ว
Const	คำสั่งนี้ หมายถึงเป็นการเริ่มต้นสร้างค่าคงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<i>Constname</i>	ชื่อค่าคงที่ คุณสามารถตั้งชื่อได้อย่างอิสระ โดยที่ไม่ผิดกฎการตั้งชื่อของ VB
<i>Type</i>	Optional เป็นการกำหนดชนิดของค่าคงที่ ซึ่งอาจเป็น Byte, Boolean, Integer, Long, Currency, Single, Double, Date, String, หรือ Variant คุณสามารถกำหนดชนิดของค่าคงที่ โดยใช้คำสั่งนี้ร่วมด้วยก็ได้
<i>Expression</i>	ค่าคงที่ ที่คุณต้องการกำหนดให้กับ constname

เช่น Const Vat = 0.07

Public Const MyName = "Suphachai"

Private Const cntyear As Integer = 12

Const MyStr = "Hello World", mpga As Double = 4.00

ตัวดำเนินการใน VB 6.0 (Operators)

ตัวดำเนินการ คือ การนำข้อมูลอย่างน้อยที่สุด 2 ตัวมากระทำอย่างใดอย่างหนึ่ง เช่น บวกกัน ลบกัน เชื่อมต่อกัน เปรียบเทียบกัน ทดสอบค่ากัน เป็นต้น ซึ่งสามารถแบ่งได้ 4 ประเภท ดังนี้

ตัวดำเนินการด้านคณิตศาสตร์ (Arithmetic Operators)

เป็นตัวดำเนินการที่ใช้สำหรับ บวก ลบ คูณ หาร ตัวเลข มีรายละเอียดดังนี้

ชื่อตัว ดำเนินการ	ลักษณะ ตัว ดำเนินการ	ตัวอย่าง	ผลลัพธ์
การบวก	+	15+20	35
การลบ	-	20-15	5
การคูณ	*	5*4	20
การหาร	/	9/3	3
การหาร จำนวนเต็ม	\	10\3	3
การ Modulo	Mod	10 Mod 3	1
การยก กำลัง	^	2^2	4
การเปลี่ยน	-	5	-5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่อง			
หมาย			

รายละเอียดของตัวดำเนินการทางคณิตศาสตร์

การบวก, ลบ, คูณ, หาร เป็นไปตามกฎทางคณิตศาสตร์ ที่น่าสนใจคือ

1. การหารจำนวนเต็ม หมายถึง จะเอาแค่เฉพาะเลขที่เป็นจำนวนเต็มเท่านั้น ส่วนเศษ จะปัดทิ้ง เช่น $10 \div 6 = 1.6666$ แต่จะได้ผลลัพธ์เท่ากับ 1 แม้ว่า .6666 จะเกินครึ่งก็ตาม ไม่มีการปัดเศษแต่อย่างใด

2. การ Mod หมายถึง จะเอาแต่เศษที่เหลืออยู่จากการ Mod เช่น $10 \text{ Mod } 3 = 1$ เนื่องจาก 10 หาร 3 ไม่ลงตัว เหลือเศษ 1 จึงได้ผลลัพธ์ 1, $20 \text{ Mod } 6 = 2$ (เหลือเศษ 2 นั่นเอง)

ตัวดำเนินการทางด้านตรรกะ (Logical Operator)

เป็นตัวดำเนินการที่ให้ผลลัพธ์แค่ 2 ค่าเท่านั้น คือ True และ False มักจะใช้ในการตรวจสอบเงื่อนไข หรือสร้างเงื่อนไขขึ้นมาเพื่อทดสอบกรณีต่างๆ เสมอในการเขียนแอปพลิเคชัน มีรายละเอียดดังนี้

A	B	And	Or	Xor	Eqv	Imp	Not A
True	True	True	True	False	True	True	False
True	False	False	True	True	False	False	False
False	True	False	True	True	False	True	True
False	False	False	False	False	True	True	True

รายละเอียดของตัวดำเนินการทางตรรกะ

1. ตัวดำเนินการ And จะมีการนิยามที่เป็นจริงก็คือ ทั้ง 2 พจน์ต้องเป็นจริง
2. ตัวดำเนินการ Or จะมีการนิยามที่เป็นเท็จก็คือ ทั้ง 2 พจน์ต้องเป็นเท็จ
3. ตัวดำเนินการ Xor ถ้า 2 พจน์มีค่าต่างกัน จะได้ผลลัพธ์เป็นจริง
4. ตัวดำเนินการ Eqv ถ้า 2 พจน์มีค่าเหมือนกัน จะได้ผลลัพธ์เป็นจริง
5. ตัวดำเนินการ Imp จะมีการนิยามที่มีค่าเป็นเท็จคือ พจน์หน้าเป็นจริง พจน์หลังเป็นเท็จ
6. ตัวดำเนินการ Not เป็นการสลับค่าของพจน์ เช่น ถ้า A เป็นจริง จะได้ผลลัพธ์เป็นเท็จ

ตัวดำเนินการทางด้านเปรียบเทียบ (Comparison Operators)

หมายถึง การนำพจน์ตั้งแต่ 2 พจน์ขึ้นไป มาเปรียบเทียบกัน เพื่อทดสอบเงื่อนไข หรือสร้างเงื่อนไข มักจะใช้คู่กับตัวดำเนินการทางด้านตรรกะ เพื่อตรวจสอบเงื่อนไขต่างๆ เสมอ มีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัว ดำเนินการ	ชื่อตัวดำเนินการ
<	น้อยกว่า
>	มากกว่า
<=	น้อยกว่าหรือเท่ากับ
>=	มากกว่าหรือเท่ากับ
=	เท่ากับ
<>	ไม่เท่ากับ

ตัวดำเนินการทางด้านการเชื่อมข้อความ (Concatenation Operators)

ถ้าโดยตามหน้าที่แล้ว จะเป็นการเชื่อมข้อความ 2 ข้อความเข้าด้วยกัน แต่ยังมีกรณีขเว้นที่จะเป็นการบวกกันของพจน์ 2 พจน์ เข้าด้วยกัน ซึ่งขึ้นอยู่กับชนิดของพจน์ที่จะมากระทำกัน สามารถแยกออกเป็นกรณี ได้ดังนี้

ตัวดำเนินการ	กรณี	ตัวอย่าง	ผลลัพธ์
+	String + String	"Visual"+"Basic 6.0"	"Visual Basic 6.0"
&	String & String	"Visual" & "Basic 6.0"	"Visual Basic 6.0"
+	String(numeric)+numeric	"20"+6	26
&	String(numeric)&numeric	"20"+6	206

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Using the Standard Statements and Functions in Visual Basic

การใช้งานคำสั่งและฟังก์ชันมาตรฐานของ Visual Basic

กลุ่มคำสั่ง Statements

- กลุ่มคำสั่งที่สร้างเงื่อนไขในการตัดสินใจ
- กลุ่มคำสั่งในการวนรูป หรือทำซ้ำ

กลุ่มฟังก์ชัน Functions

- ฟังก์ชันที่เกี่ยวข้องกับข้อมูลชนิด String
- กลุ่มคำสั่งและฟังก์ชันที่เกี่ยวข้องกับอาร์เรย์
- กลุ่มฟังก์ชันที่เกี่ยวข้องกับตัวแปร Variant
- กลุ่มฟังก์ชันที่เกี่ยวข้องกับการแปลงชนิดของข้อมูล
- กลุ่มฟังก์ชันสำหรับสร้างกรอบโต้ตอบผู้ใช้

กลุ่มฟังก์ชันและคำสั่งอื่นๆ ที่น่าสนใจ

- ฟังก์ชัน Format () และ Format\$ ()
- การใช้งานคำสั่ง With...End With

การสร้างโปรซีเจอร์ต่างๆ ไป

- การสร้างซับรูทีน
- การสร้างฟังก์ชัน

ตัวอย่างประจำบทที่ 4

- ฟังก์ชันสำหรับหาค่า Max และหาค่า Min
- ฟังก์ชันสำหรับหาค่ายกกำลัง
- ฟังก์ชันสำหรับหาค่า factorial

คำสั่ง (statements) และฟังก์ชัน (functions) เป็นสิ่งหนึ่งที่คุณต้องทำความเข้าใจมากพอสมควร เนื่องจากถ้าคุณทราบความหมาย รูปแบบการใช้งาน และทราบหน้าที่มากเท่าใด คุณจะสามารปรับแต่งโค้ดของคุณ ให้ประมวลผลได้รวดเร็วมากขึ้น ไม่ว่าจะเป็ฟังก์ชัน หรือคำสั่งใดๆ ก็ตาม จะมีทั้งที่มีหน้าที่

คล้ายๆ กัน และเหมือนกัน ซึ่งล้วนแล้วแต่มีจุดดี จุดด้อย เหมาะสมกับสภาพการใช้งานแตกต่างกันไป ดังนั้นคุณควรเลือกใช้คำสั่ง หรือฟังก์ชัน ให้ตรงกับความต้องการของคุณมากที่สุด และเป็นทางเลือกที่ดีที่สุดเช่นกัน และคุณไม่ต้องกังวลว่า ใน VB มีฟังก์ชัน หรือคำสั่งอยู่มากมาย จะจำได้ไม่หมด แต่ด้วยความสามารถของ editor จะมี Tool Tip ที่เป็นไวยากรณ์ประจำคำสั่ง หรือฟังก์ชันนั้นๆ ขึ้นมาโดยอัตโนมัติ เมื่อคุณเลือกใช้คำสั่ง หรือฟังก์ชันใดๆ ก็ตาม ที่เหลือเพียงแต่ คุณต้องทำความเข้าใจกับโครงสร้างของมันนั่นเอง

ข้อแตกต่างของคำสั่งและฟังก์ชันก็คือ ถ้าคุณใช้คำสั่ง จะหมายถึง สั่งให้ VB ปฏิบัติตามหน้าที่ประจำตัวของมัน แล้วไม่มีการคืนผลลัพธ์ หรือคืนค่าออกมาจากคำสั่งนั้นๆ เช่น คำสั่งในการวนลูป Do While-Loop มีหน้าที่สำหรับสั่งให้ VB ประมวลผลซ้ำข้อความที่อยู่ระหว่างคำสั่ง Do While-Loop ซึ่งจะไม่มีการคืนค่า จากคำสั่ง Do While-Loop แต่อย่างใด แต่อาจจะคืนค่าที่ได้จากการประมวลผลก็ได้ สำหรับฟังก์ชัน หมายถึง สั่งให้ VB ประมวลผลไปตามหน้าที่ของแต่ละฟังก์ชัน เช่นเดียวกับคำสั่ง แต่มันจะคืนค่าออกมา 1 ค่า ซึ่งโปรแกรมเมอร์จะเอาค่าดังกล่าวนี้ ไปเป็นเงื่อนไข ในการประมวลผลต่อไป สรุปได้ว่า เมื่อคุณมีการใช้งานฟังก์ชัน คุณจะต้องสร้างตัวแปร 1 ตัวต่อ 1 ฟังก์ชัน เพื่อรับค่าจากฟังก์ชัน และตัวแปรดังกล่าว จะต้องเป็นข้อมูลชนิดเดียวกันกับ ค่าที่ฟังก์ชันนั้นๆ ส่งคืนมาด้วย

คำสั่งที่น่าสนใจใน VB (Statements)

คำสั่งใน VB มีมากมายหลายชนิด ซึ่งล้วนแล้วแต่มีความสำคัญ และมีหน้าที่ต่างๆ กันไป ซึ่งคุณสามารถนำไปใช้เพื่อให้เกิดผล ได้อย่างมากมาย หัวข้อนี้ เป็นการอธิบายคำสั่ง (statements) ที่น่าสนใจซึ่งคุณควรทราบ ผู้เขียนจะแบ่งออกเป็นกลุ่มๆ ตามหน้าที่การใช้งานดังนี้

คำสั่งที่เกี่ยวข้องกับการสร้างเงื่อนไข (Condition Statements)

เป็นกลุ่มคำสั่งที่คุณต้องใช้มากที่สุด ไม่ว่าคุณจะเขียน โปรแกรมด้วยภาษาอะไรก็ตาม และก็คือได้ว่าเป็นกลุ่มคำสั่งที่มีความสำคัญ มากที่สุดเช่นกัน ซึ่งประกอบไปด้วยกลุ่มคำสั่ง 2 ชุดดังนี้

- กลุ่มคำสั่งที่สร้างเงื่อนไขในการตัดสินใจ (Decision Structures)
- กลุ่มคำสั่งในการวนลูป หรือทำซ้ำ (Loop Structures)

กลุ่มคำสั่งที่สร้างเงื่อนไขในการตัดสินใจ (Decision Structures)

- คำสั่ง If-Then

เป็นการสร้างเงื่อนไข ตรวจสอบเงื่อนไข หรือเป็นทางเลือก เพื่อตัดสินใจแบบ ธรรมดาและง่ายที่สุด นั่นคือ ถ้าเงื่อนไขเป็นจริง ก็จะทำตามเงื่อนไขที่คุณ ได้สร้างไว้ มีรูปแบบการใช้งานดังนี้

If condition Then statements

- ตัวแปร condition หมายถึง เงื่อนไขที่คุณกำหนดขึ้นมา
- ตัวแปร statements หมายถึง เมื่อเงื่อนไขเป็นจริงแล้ว จะให้ทำอะไร เช่น

If cnt < 5 Then a = b + cnt

จากตัวอย่าง ถ้าตัวแปร cnt น้อยกว่า 5 ให้เอาตัวแปร b+cnt แล้วเก็บไว้ในตัวแปร a จะเห็นได้ว่า เฉพาะกรณีตัวแปร cnt น้อยกว่า 5 เท่านั้น จึงจะมีการบวกกัน แต่ถ้าตัวแปร cnt มากกว่า 5 ก็จะไม่ทำอะไร แต่คุณยังสามารถสร้างเงื่อนไข เพิ่มขึ้น เมื่อไว้ในกรณีที่ เงื่อนไขไม่เป็นจริง โดยมีรูปแบบการใช้งานดังนี้

If condition Then

statements

Else

elsestatements

End If

- คำสั่ง Else หมายถึง กรณีที่เงื่อนไขดังกล่าวเป็นเท็จ
- คำสั่ง End If หมายถึง จบเงื่อนไข
- ตัวแปร elsestatements หมายถึง ชุดคำสั่งที่จะทำ เมื่อเงื่อนไขเป็นเท็จ เช่น

If cnt < 5 Then

a = b + cnt

Else

a = b - cnt

End If

จากตัวอย่าง เห็นได้ว่า มีการเพิ่มกรณีที่ตัวแปร cnt ≥ 5 ให้เอาตัวแปร b-cnt แล้วให้เก็บไว้ในตัวแปร a คุณยังสามารถเพิ่มกรณีตรวจสอบ เงื่อนไขมากขึ้นอีกก็ได้ เมื่อคุณมีการสร้างแอปพลิเคชันขึ้นมา บางครั้งคุณอาจต้องสร้างเงื่อนไขตรวจสอบมากมาย และแยกเป็นหลายๆ กรณี ซึ่งเป็นรูปแบบที่คุณจะได้พบแน่นอน มีรูปแบบการใช้งานดังนี้

If condition Then

statements

ElseIf condition2 Then

elseifstatements

Else

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

elsestatements :

End If

- คำสั่ง ElseIf หมายถึง คำสั่งสำหรับเพิ่มกรณีตรวจสอบขึ้นมา เพื่อสร้างเงื่อนไขขึ้นมาอีก 1 กรณี
- ตัวแปร condition2 หมายถึง เป็นเงื่อนไขที่ 2 ที่คุณกำหนดขึ้นมา
- ตัวแปร elseifstatements หมายถึง ชุดคำสั่งสำหรับกรณีเงื่อนไขที่ 2 เช่น

If cnt < 5 Then

a = b + cnt

ElseIf cnt < 10 Then

a = b * cnt

Else

a = b - cnt

End If

สมมติว่าตัวแปร cnt=6 ซึ่งจะทำให้กรณีแรก (cnt<5) เป็นเท็จทันที จึงมาที่เงื่อนไขที่ 2 ปรากฏว่า เป็นจริง จึงนำตัวแปร b*cnt แล้วเก็บไว้ในตัวแปร a ให้คุณลองเปลี่ยนเป็น cnt=10 ทำให้กรณีที่ 1 และ 2 เป็นเท็จทันที ก็จะมาทำคำสั่งที่อยู่ในเงื่อนไข else ทันที คุณสามารถเพิ่มกรณีแบบนี้ ได้ไม่จำกัด กล่าวคือ ใส่ชุดของ ElseIf condition2 Then elseifstatements มากเท่าใดก็ได้ ตามที่คุณต้องการ แต่ถ้ามีการตรวจสอบเงื่อนไขมากๆ หลายกรณี โปรแกรมเมอร์ไม่นิยมใช้ชุดคำสั่งนี้ จะใช้ชุดคำสั่ง Select Case เพราะมีการตรวจสอบ ที่ทำให้ source code มีระเบียบ เรียบร้อยมากกว่า

- คำสั่ง Select-Case

คำสั่งนี้เหมาะสำหรับ กรณีที่มีเงื่อนไขมากๆ ยิ่งมากยิ่งขึ้น เพราะ ถ้าคุณใช้เงื่อนไขแบบ If-Then-ElseIf จะทำให้คุณสับสนเป็นอย่างยิ่ง เพราะถ้ากรณีที่คุณ ต้องการตรวจสอบเงื่อนไขมากๆ แล้ว คุณจะไม่สามารถจัดกลุ่ม หรือจัดระเบียบให้ source code เรียบร้อย ได้เลย แต่ด้วยคำสั่ง Select Case มันจะช่วยลดขั้นตอน และความสับสนทั้งหมดที่เกิดขึ้นมาได้ มีรูปแบบการใช้งานดังนี้

Select Case testexpression

Case expressionlist1

statements1

Case [Is] expressionlist2

statements2

[Case [Is] expressionlist-n]

[statements-n]

Case Else

elsestatements

End Select

- คำสั่ง Select Case หมายถึง เริ่มกรณีตรวจสอบแบบ Select Case
- ตัวแปร testexpression หมายถึง ตัวแปรที่จะนำมาตรวจสอบ
- ตัวแปร expressionlist1 หมายถึงค่าของตัวแปรในกรณีที่ 1
- ตัวแปร statement1 หมายถึง ชุดคำสั่งที่ต้องกระทำ เมื่อค่าของตัวแปรตรงกับกรณีที่ 1
- ตัวแปร expressionlist2 หมายถึงค่าของตัวแปรในกรณีที่ 2
- ตัวแปร statement2 หมายถึง ชุดคำสั่งที่ต้องกระทำ เมื่อค่าของตัวแปรตรงกับกรณีที่ 2
- คำสงวน Is จะใช้ในกรณีที่คุณใส่เฉพาะเงื่อนไข ซึ่ง VB จะใส่ให้คุณโดยอัตโนมัติ
- คำสั่ง Case Else หมายถึง เมื่อค่าของตัวแปร ไม่เท่ากับกรณีที่ผ่านมาทั้งหมด ให้มาที่กรณีนี้
- ตัวแปร elsestatements หมายถึง ชุดคำสั่งสำหรับกรณีที่ค่าของตัวแปรไม่ตรงกับกรณีใดเลย
- คำสั่ง End Select หมายถึง จบการตรวจสอบแบบ Select Case เช่น

```
Private Sub Command1_Click()
```

```
Dim cnt As Integer
```

```
cnt = 6
```

```
Select Case cnt
```

```
Case Is <= 5
```

```
Debug.Print "Cnt Less than or equal 5"
```

```
Case Is <= 7
```

```
Debug.Print "Cnt Less than or equal 7"
```

```
Case Else
```

```
Debug.Print "Cnt Very big"
```

```
End Select
```

```
End Sub
```

จากตัวอย่างข้างต้น ให้คุณทดลองทดสอบเปลี่ยนค่าของตัวแปร cnt ไปเรื่อยๆ สังเกตผลที่หน้าต่าง Immediate

กลุ่มคำสั่งที่ใช้สำหรับสั่งให้ทำซ้ำ (Loop Statements)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งในกลุ่มนี้เป็นอีกกลุ่มหนึ่งที่คุณจะต้องใช้งานบ่อยครั้งที่สุด ไม่ว่าคุณจะพัฒนาแอปพลิเคชันด้วยภาษาอะไรก็ตาม เพราะมีหน้าที่สำหรับสั่งให้ VB ประมวลผลซ้ำกลุ่มคำสั่งเดิม วนไปเรื่อยๆ จนกว่าจะตรงกับเงื่อนไขที่คุณต้องการ และยังมีหน้าที่สำหรับลดขั้นตอนในการเขียน โค้ดที่มีลักษณะเหมือนๆ กันได้อีกด้วย

1. For-Next
2. Do While-Loop
3. Do-Loop While
4. Do Until-Loop
5. Do-Loop Until

การวนรูปแต่ละชนิด มีรายละเอียดดังนี้

- คำสั่ง For-Next

คำสั่งนี้มีหน้าที่สำหรับสั่งให้ VB วนรูป เหมาะสมกับการทำงาน ที่คุณทราบจำนวนรอบ หรือทราบขอบเขตการวนที่แน่นอน มีรูปแบบการใช้งานดังนี้

```
For counter = start To end [Step step]
```

```
statements
```

```
[Exit For]
```

```
[statements]
```

```
Next [counter]
```

- ตัวแปร counter หมายถึง ตัวแปรที่คุณกำหนดขึ้นมาเพื่อ เป็นตัวนับรอบที่ใช้ในการวนรูป
- ตัวแปร start หมายถึง ค่าเริ่มต้นของตัวนับ เป็นเลขจำนวนจริง คุณควรที่จะใช้เลขจำนวนเต็มเท่านั้น เพื่อง่ายต่อการตรวจสอบค่าตัวนับ
- คำสงวน To หมายถึง ขอบเขตการนับ
- ตัวแปร end หมายถึง ค่าสิ้นสุดของตัวนับ ไม่จำเป็นต้องเป็นตัวเลข ในบางครั้งคุณอาจสร้างเป็นเงื่อนไขที่ไม่ใช่ตัวเลขก็ได้ แต่ในการใช้งานโดยทั่วๆ ไป จะเป็นตัวเลข และควรจะเป็นเลขจำนวนเต็มด้วย
- คำสงวน Step หมายถึง เป็นการกำหนดค่าที่จะเพิ่มขึ้นของตัวนับในแต่ละรอบ ถ้าคุณ ไม่กำหนด จะเพิ่มรอบละ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ตัวแปร step หมายถึง ค่าของตัวนับที่จะเพิ่มขึ้นในแต่ละรอบ จะต้องเป็นเลขจำนวนจริงเท่านั้น ในการใช้งานตามปกติ คุณควรใช้เลขจำนวนเต็ม เพื่อง่ายต่อการตรวจสอบจำนวนรอบในภายหลัง
- ตัวแปร statements หมายถึง ชุดคำสั่งที่คุณต้องการวนลูป
- คำสั่ง Exit For หมายถึง คำสั่งที่บังคับให้ออกจากลูปทันที จะใช้ในบางกรณีเท่านั้น เช่นการค้นจับข้อผิดพลาด

สาเหตุที่ผู้เขียนแนะนำให้กำหนดค่าตัวนับ หรือตัวสิ้นสุด เป็นเลขจำนวนเต็ม เพราะเนื่องจากว่า ถ้าคุณเคยทดสอบ การประมวลผลเครื่องคอมพิวเตอร์ ด้วยโปรแกรม Benchmark, Wintune 99, หรือ FPU WinMark ฯลฯ ตามมาตรฐาน iComp Index 1.0, 2.0 หรือ 3.0 คุณจะพบว่า ไม่ว่าคุณจะใช้ CPU ยี่ห้อใดก็ตาม การประมวลผลแบบ ทศนิยมจะช้ากว่า การประมวลผลแบบเลขจำนวนเต็มอยู่พอสมควร ซึ่งจะส่งผลเป็นอย่างยิ่ง ในกรณีที่คุณมีการตั้งวนรอบหลายหมื่น หรือแสนรอบ รวมถึงการสั่งให้เข้าถึงข้อมูลในแต่ละ record ในฐานข้อมูลที่มี records จำนวนมากๆ แอปพลิเคชันจะประมวลผลช้ามาก คุณควรให้หน้าหนักในเรื่องรายละเอียดปลีกย่อยนี้พอสมควรครับ :-) ด้วยเหตุนี้เอง คุณควรที่จะใช้ข้อมูลชนิดที่เป็นเลขจำนวนเต็ม ไม่ว่าจะเป็น Integer หรือ Long ให้มากที่สุดเท่าที่จะเป็นไปได้ และขอให้ใช้ชนิดข้อมูลที่เป็นพื้นฐาน ให้มากที่สุด ซึ่งยังเป็นการประหยัดทรัพยากรระบบอีกทางหนึ่งด้วย

สำหรับตัวอย่างการใช้งานคำสั่ง For-Next มีดังนี้

```
Private Sub Form_Activate()
Dim i As Integer
For i = 1 To 10
Print i
Next i
End Sub
```

ตัวอย่างดังกล่าว เป็นการพิมพ์เลขตั้งแต่ 1-10 ลงบนฟอร์ม ให้คุณลองเปลี่ยน step จาก 1 เป็น 2 แล้วสังเกตผล

- คำสั่ง Do While-Loop

เป็นคำสั่งทำซ้ำ ที่มีจำนวนรอบขึ้นอยู่กับเงื่อนไข

```
Do While condition
statements
[Exit Do]
Loop
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ตัวแปร condition หมายถึง เงื่อนไขที่คุณกำหนดขึ้นมา
- ตัวแปร statements หมายถึง ชุดคำสั่งที่คุณต้องการทำซ้ำ
- คำสั่ง Exit Do หมายถึง ออกจาก Do While-Loop ทันที
- คำสั่ง Loop หมายถึง ขอบเขตสิ้นสุดการวนลูป

รูปแบบ Do While นี้ จะวนลูปก็ต่อเมื่อเงื่อนไข (ตัวแปร condition) ยังเป็นจริงอยู่ ถ้าเงื่อนไขดังกล่าวเป็นเท็จเมื่อใด ก็จะออกจากลูปทันที มีข้อที่น่าสังเกตคือ การใช้รูปแบบนี้ จะมีโอกาสที่ไม่ต้องวนลูปเลยแม้แต่รอบเดียว นั่นคือ ถ้าเงื่อนไขที่เข้ามาเป็นเท็จ ก็ไม่ต้องวนลูปทันที เช่น

```
Private Sub Form_Activate()
```

```
Dim i As Integer
```

```
Do While i <= 10
```

```
Print i
```

```
i = i + 1
```

```
Loop
```

```
End Sub
```

ตัวอย่างด้านบน จะมีผลเหมือนกับลูป For-Next เพียงแต่เปลี่ยนลักษณะวนลูป และเงื่อนไข ลองเปลี่ยนเงื่อนไข ดังนี้

```
Private Sub Form_Activate()
```

```
Dim i As Integer
```

```
i = 11
```

```
Do While i <= 10
```

```
Print i
```

```
i = i + 1
```

```
Loop
```

```
End Sub
```

จะเห็นได้ว่า บนฟอร์มไม่มีการพิมพ์เลข แต่อย่างใด เพราะตัวแปร i มีค่าเท่ากับ 11 ส่งผลให้เงื่อนไขเป็นเท็จทันที

- คำสั่ง Do-Loop While

คำสั่งนี้เป็นการสั่งวนลูป มีจำนวนรอบขึ้นอยู่กับเงื่อนไข มีรูปแบบการใช้งานดังนี้

Do

statements

[Exit Do]

Loop While condition

- ตัวแปร condition หมายถึง เงื่อนไขที่คุณกำหนดขึ้นมา
- ตัวแปร statements หมายถึง ชุดคำสั่งที่คุณต้องการทำซ้ำ
- คำสั่ง Exit Do หมายถึง ออกจาก Do- Loop While ทันที
- คำสั่ง Loop หมายถึง ขอบเขตสิ้นสุดการวนลูป

จะเห็นได้ว่า มีลักษณะคล้ายกับแบบที่ 1 เพียงแต่ย้ายเงื่อนไขมาไว้ด้านล่าง ซึ่งหมายความว่า ลูปแบบนี้จะวนอย่างน้อยที่สุด 1 รอบ เพื่อตรวจสอบเงื่อนไข (ตัวแปร condition) ในรอบแรกที่เข้ามาก่อนว่าเป็นจริงหรือเท็จ ถ้าจริงก็จะวนลูปต่อไปตามปกติ แต่ถ้าเป็นเท็จ จะออกจากลูปทันที เช่น

```
Private Sub Form_Activate()
```

```
Dim i As Integer
```

```
Do
```

```
Print i
```

```
i = i + 1
```

```
Loop While i <= 10
```

```
End Sub
```

เช่นกัน ตัวอย่างนี้ เป็นการพิมพ์เลขตั้งแต่ 1 ถึง 10 บนฟอร์ม ให้คุณลองเปลี่ยนเงื่อนไขดังนี้

```
Private Sub Form_Activate()
```

```
Dim i As Integer
```

```
i = 11
```

```
Do
```

```
Print i
```

```
i = i + 1
```

```
Loop While i <= 10
```

```
End Sub
```

ข้อแตกต่างของลูป Do While - Loop กับ Do-Loop While นั่นคือ ถ้าเป็น Do While-Loop กรณีที่เงื่อนไขเป็นเท็จ บนฟอร์ม จะไม่ปรากฏตัวเลขอะไรเลย เพราะเนื่องจากว่า ไม่ได้มีการวนลูปแต่อย่างใด เพราะ

เงื่อนไขเป็นเท็จ (ค่า $i=11$) แต่รูปแบบ Do-Loop While จะปรากฏค่าของตัวเงื่อนไขออกมาก่อน ถ้าจริง ก็
จะวนลูปตามปกติ แต่กรณีนี้เป็นเท็จ จึงแสดงค่าออกมา 1 ค่า ซึ่งเกิดมาจากการวนรอบแรกนั่นเอง

- คำสั่ง Do Until-Loop

คำสั่งนี้มีหน้าที่สั่งให้วนลูปเช่นกัน มีจำนวนรอบขึ้นอยู่กับเงื่อนไข (condition) ถ้าเงื่อนไขเป็นเท็จ จึง
จะวนลูป แต่ถ้าเป็นจริงจะไม่มีกรวนแต่อย่างใด ซึ่งจะตรงกันข้ามกับลูปชนิด Do While-Loop มีรูปแบบ
การใช้งานดังนี้

Do Until condition

statements

[Exit Do]

statements

Loop

- ตัวแปร condition หมายถึง เงื่อนไขที่คุณกำหนดขึ้นมา
- ตัวแปร statements หมายถึง ชุดคำสั่งที่คุณต้องการทำซ้ำ
- คำสั่ง Exit Do หมายถึง ออกจาก Do Until- Loop ทันที
- คำสั่ง Loop หมายถึง ขอบเขตสิ้นสุดการวนลูป เช่น

```
Private Sub Command1_Click()
```

```
Dim x As Integer
```

```
Do Until x >= 10
```

```
Print x
```

```
x = x + 1
```

```
Loop
```

```
End Sub
```

ตัวอย่างนี้เป็นการพิมพ์เลข 1 ถึง 10 ลงบนฟอร์ม ทุกๆ ครั้งที่มีการคลิกที่ปุ่ม Command1

- คำสั่ง Do-Loop Until

คำสั่งการวนลูปแบบนี้มีลักษณะคล้ายกับรูปแบบ Do-Loop While เพียงแต่ เงื่อนไขที่เข้ามาจะต้องเป็น
เท็จ จึงจะวนลูป ถ้าเงื่อนไขเป็นจริง จะออกจากลูปทันที การใช้รูปแบบนี้จะมีกรวนอย่างน้อยที่สุด 1 รอบ
เช่นกัน เนื่องจากการวนรอบแรก ที่จะต้องมีการตรวจสอบเงื่อนไขนั่นเอง มีรูปแบบการใช้งานดังนี้

Do

statements

[Exit Do]

[statements]

Loop Until condition

- ตัวแปร condition หมายถึง เงื่อนไขที่คุณกำหนดขึ้นมา
- ตัวแปร statements หมายถึง ชุดคำสั่งที่คุณต้องการทำซ้ำ
- คำสั่ง Exit Do หมายถึง ออกจาก Do Loop-Until ทันที
- คำสั่ง Loop หมายถึง ขอบเขตสิ้นสุดการวนลูป เช่น

```
Private Sub Command1_Click()
```

```
Dim x As Integer
```

```
x = 11
```

```
Do
```

```
Print x
```

```
x = x + 1
```

```
Loop Until x >= 10
```

```
End Sub
```

จากตัวอย่างนี้ จะเห็นได้ว่า ค่า $x=11$ ซึ่งถึงแม้ว่า จะทำให้เงื่อนไขเป็นจริง (ออกจากลูป) ก็ตาม แต่ฟอร์มก็ยังคงพิมพ์ค่า x ออกมา 1 ค่า เนื่องจาก การวนรอบแรกนั่นเอง

ข้อควรระวัง ไม่ว่าคุณจะใช้รูปแบบใดก็ตาม ในกรณีที่คุณประกาศตัวนับ เป็นเลขจำนวนเต็มชนิด Integer ขอให้คุณระวังเกี่ยวกับจำนวนรอบ ในการวนให้ดี เพราะเนื่องจากว่า ขอบเขตของข้อมูลชนิด Integer มีค่าระหว่าง -32768 ถึง 32767 ซึ่งถ้าคุณใช้แค่ค่าบวก จะทำให้ตัวแปรดังกล่าว สามารถวนได้เพียง 32768 รอบเท่านั้น ซึ่งจะมีผลในตอนรัน ถ้ามีโอกาสที่ทำให้เกิดการวนมากกว่า 32768 รอบ ก็จะเกิด error ได้ ซึ่งเป็นเหตุการณ์ที่ผู้เขียนเคยพบ โดยบังเอิญ

ฟังก์ชันที่น่าสนใจใน VB (Functions)

ฟังก์ชัน ใน VB มีมากมาย หลายหน้าที่ ซึ่งครอบคลุมการทำงานเกือบทั้งหมด เป็นหัวข้อที่คุณต้องทำความเข้าใจเป็นอย่างดี เพราะว่า ในการเขียนโค้ด ไม่ว่าคุณจะใช้ฟังก์ชันใดๆ ก็ตาม คุณจะต้องทราบหน้าที่ และข้อจำกัดของฟังก์ชันนั้นๆ ก่อน หรือแม้กระทั่งฟังก์ชันที่มีหน้าที่เหมือนกัน อาจจะเหมาะสมกับการใช้งานต่างกันก็ได้ ซึ่งผู้เขียนคิดว่า นี่เป็นรายละเอียดที่คุณควรทราบ และน่าจะเป็นประโยชน์อย่างยิ่งในการเลือกใช้งาน มีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชันที่เกี่ยวข้องกับข้อมูลชนิด String

สำหรับฟังก์ชันในกลุ่มนี้ เป็นฟังก์ชันที่นิยมใช้กันมากอีกกลุ่มหนึ่ง ซึ่งจะเป็นฟังก์ชันที่จัดการด้านต่างๆ ที่เกี่ยวข้องกับข้อมูลชนิด String ทั้งสิ้น โดยมีรายละเอียดดังนี้

- ฟังก์ชัน Len () และฟังก์ชัน LenB ()

หน้าที่ ใช้สำหรับหาขนาดของตัวแปร String
รูปแบบการใช้งาน

Len(string | varname)

ตัวแปร string หรือ varname หมายถึง ข้อความที่คุณต้องการหาขนาด

ค่าที่ส่งกลับ เป็นตัวเลขจำนวนจริง Long ซึ่งสามารถแยกได้ 2 กรณี คือ

- ฟังก์ชัน Len () จะคืนค่าเป็นจำนวนตัวอักษรของตัวแปร String
- ฟังก์ชัน LenB () จะคืนค่าเป็นจำนวนหน่วยความจำที่ตัวแปร String นั้น ใช้อยู่ มีหน่วยเป็น ไบต์ (bytes) และถ้าตัวแปร String มีช่องว่างอยู่ ช่องว่าง 1 ช่อง จะถือเป็น 1 ตัวอักษรด้วย

ตัวอย่างการใช้งาน

```
Private Sub Form_Load()
```

```
Dim name As String 'ตัวแปรสำหรับเก็บข้อความ
```

```
Dim rtn As Long 'ตัวแปรสำหรับเก็บค่าที่ส่งคืนมาจากฟังก์ชัน Len ()
```

```
name = "Suphachai Somphanit" 'กำหนดข้อความให้กับตัวแปร
```

```
rtn = Len(name)
```

```
Label1.Caption = rtn 'แสดงค่าที่ได้ โดยใช้คอนโทรล Label มีค่าเท่ากับ 19
```

```
End Sub
```



Note จากตัวอย่างเป็นการหาขนาดโดยผ่านตัวแปร name คุณอย่าได้เครื่องหมาย " " ที่ตัวแปร name เด็ดขาด เพราะจะเป็นการหาขนาดตัวแปร name

คั่งตัวอย่างต่อไปนี้

```
Private Sub Form_Load()
```

```
Dim name As String
```

```
Dim rtn As Long
```

```
name = "Suphachai Somphanit"
```

```
rtn = Len("name")
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Label1.Caption = rtn :

End Sub

ตัวอย่างนี้จะคืนค่าเท่ากับ 4 จะเห็นได้ว่า ค่าที่ได้ผิดทันที เพราะเป็นการหาขนาดของตัวแปร name ซึ่งมี 4 ตัวอักษร แต่ถ้าคุณต้องการหาขนาด โดยตรง คุณต้องใส่เครื่องหมาย " " ด้วย เช่น

Private Sub Form_Load()

Dim rtn As Long

rtn = Len("Suphachai Somphanit")

Label1.Caption = rtn

End Sub

ตัวอย่างนี้จะคืนค่าเท่ากับ 19 ซึ่งเป็นค่าที่ถูกต้อง

- ฟังก์ชัน StrConv

หน้าที่ ใช้สำหรับแปลงข้อมูลชนิด String ให้อยู่ในรูปแบบที่เราสามารถกำหนดได้ รูปแบบการใช้งาน

StrConv(string,conversion)

ตัวแปร string หมายถึง ข้อความ string ที่คุณต้องการแปลง

ตัวแปร conversion หมายถึง ตัวแปรที่กำหนดรูปแบบการแปลง string ซึ่งคุณสามารถกำหนดได้ดังนี้

ค่าคงที่	ค่าตัวเลข	รายละเอียด
vbUpperCase	1	แปลงตัวอักษร ให้เป็นตัวพิมพ์ใหญ่ทั้งหมด
vbLowerCase	2	แปลงตัวอักษร ให้เป็นตัวพิมพ์เล็กทั้งหมด
vbProperCase	3	แปลงเฉพาะอักษรตัวแรก ของทุกๆ คำทั้งหมด ให้เป็นตัวพิมพ์ใหญ่
vbWide*	4*	แปลงตัวอักษรชนิด 1 ไบต์ ให้เป็น 2 ไบต์
vbNarrow*	8*	แปลงตัวอักษรชนิด 2 ไบต์ ให้เป็น 1 ไบต์
vbKatakana**	16**	แปลงอักษรแบบ Hirakana ให้เป็นอักษรแบบ Katakana
vbHiragana**	32**	แปลงอักษรแบบ Katakana ให้เป็นอักษรแบบ Hiragana
vbUnicode	64	แปลงรูปแบบการจัดเก็บตัวอักษร ให้จัดเก็บอยู่ในรูปแบบ Unicode
vbFromUnicode	128	แปลงรูปแบบการจัดเก็บจากแบบ Unicode ให้เป็นรูปแบบจัดเก็บเดียวกับระบบที่ใช้อยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

*ใช้กับอักษรของภาษาอาหรับเท่านั้น

**ใช้กับอักษรของภาษาญี่ปุ่นเท่านั้น

ค่าที่ส่งกลับ เป็นข้อมูลชนิด String

ตัวอย่างการใช้งาน

Private Sub Form_Activate()

```
Dim FirstName As String 'ตัวแปรเก็บชื่อ
```

```
Dim LastName As String 'ตัวแปรเก็บนามสกุล
```

```
Dim rtn As String 'ตัวแปรรับค่าส่งกลับ
```

```
Dim rtn2 As String 'ตัวแปรรับค่าส่งกลับ
```

```
FirstName = "sophachai" 'กำหนดค่าให้กับตัวแปร FristName
```

```
LastName = "somphanit" 'กำหนดค่าให้กับตัวแปร LastName
```

```
rtn = StrConv(FirstName, 1) 'แปลงให้เป็นตัวใหญ่ทั้งหมด
```

```
rtn2 = StrConv(LastName, 3) 'แปลงเฉพาะอักษรตัวแรก
```

```
Print rtn
```

```
Print rtn2
```

```
End Sub
```

จากตัวอย่างจะเห็นได้ว่า คำว่า sophachai จะถูกแปลงเป็นตัวพิมพ์ใหญ่ทั้งหมด ส่วนคำว่า somphanit ตัว s จะถูกแปลงเป็นตัวใหญ่ เพียงตัวเดียวเท่านั้น

กลุ่มคำสั่งและฟังก์ชันที่เกี่ยวกับอาร์เรย์

สำหรับคำสั่งและฟังก์ชันในกลุ่มนี้ คุณจะ ได้ใช้งานบ่อยมากที่สุดอีกกลุ่มหนึ่ง ประกอบด้วยคำสั่งและฟังก์ชัน ดังนี้

- คำสั่ง Erase
- ฟังก์ชัน LBound และ UBound
- คำสั่ง Option Base
- คำสั่ง ReDim

สำหรับรายละเอียดของแต่ละคำสั่งและฟังก์ชันมีดังนี้

- คำสั่ง Erase

หน้าที่ คำสั่ง Erase จะใช้สำหรับ กำหนดค่าตัวแปรอาร์เรย์ชนิดสแควคิก ให้กลายเป็นค่าเริ่มต้นใหม่ รวมถึงคืนหน่วยความจำให้กับระบบ เมื่อคุณใช้งานตัวแปรอาร์เรย์แบบไดนามิก

Erase arraylist

ตัวแปร arraylist หมายถึง ชื่อของตัวแปรอาร์เรย์ ที่คุณต้องการกำหนดค่าใหม่ หรือต้องการลบตัวแปรอาร์เรย์ เพื่อคืนหน่วยความจำให้กับระบบ

- ฟังก์ชัน LBound () และ UBound ()

หน้าที่และค่าที่ส่งกลับ ฟังก์ชัน LBound มีหน้าที่รายงานค่าของตัวแปรอาร์เรย์ ซึ่งเป็นค่าขอบเขตล่าง ส่วนฟังก์ชัน UBound มีหน้าที่รายงานค่าตัวแปรอาร์เรย์ ซึ่งเป็นค่าขอบเขตบน

LBound(arrayname[,dimension])

UBound(arrayname[,dimension])

ตัวแปร arrayname หมายถึง ชื่อของตัวแปรอาร์เรย์ ที่คุณต้องการค้นหาขอบเขตบน หรือขอบเขตล่าง
ตัวแปร dimension หมายถึง มิติของอาร์เรย์ ใช้ในกรณีที่ตัวแปรอาร์เรย์มีตั้งแต่ 2 มิติขึ้นไป ถ้าคุณไม่กำหนด ค่า default คือ แบบ 1 มิติ

ตัวอย่างการใช้งาน

Dim x(1 To 50, 5 To 10, -10 To 20)

ฟังก์ชัน	ค่าที่ส่งกลับมา
LBound(x, 1)	1
UBound(x, 1)	50
LBound(x, 2)	5
UBound(x, 2)	10
LBound(x, 3)	-10
UBound(x, 3)	20

- คำสั่ง Option Base

หน้าที่ ใช้สำหรับกำหนดค่าขอบเขตล่าง ของตัวแปรอาร์เรย์

รูปแบบการใช้งาน

Option Base {0 | 1}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อแนะนำในการใช้งาน. โดยปกติแล้ว ตัวแปรอาร์เรย์ลำดับแรก จะมีค่าเป็น 0 แต่ด้วยคำสั่งนี้ คุณสามารถกำหนดให้ตัวแปรอาร์เรย์ เริ่มต้นที่ 1 ได้ ในการใช้งานคำสั่งนี้ คุณต้องประกาศคำสั่ง Option Base นี้ในระดับ Module เท่านั้น และจะต้องประกาศก่อนที่จะมีการใช้งาน ตัวแปรอาร์เรย์ใดๆ เสมอ โดยที่ใน 1 โปรแกรมก็แค่ คุณสามารถกำหนดได้เพียงครั้งเดียวเท่านั้น

- คำสั่ง ReDim

หน้าที่ ใช้สำหรับเปลี่ยนแปลงจำนวนสมาชิกของตัวแปรอาร์เรย์ ในขณะที่รันรูปแบบการใช้งาน

```
ReDim [Preserve] varname(subscripts) [As type] [, varname(subscripts) [As type]] . . .
```

ตัวแปร Preserve หมายถึง เมื่อเปลี่ยนแปลงขนาดของตัวแปรอาร์เรย์แล้ว VB ยังคงเก็บรักษาค่าของตัวแปรดังกล่าวไว้คงเดิม

ตัวแปร varname หมายถึง ชื่อของตัวแปรอาร์เรย์

ตัวแปร subscripts หมายถึง เลขจำนวนเต็ม ซึ่งใช้สำหรับกำหนดประเภทของตัวแปรอาร์เรย์ว่า จะเป็นแบบสแควร์ หรือแบบไดนามิก

ตัวแปร As type หมายถึง ชนิดข้อมูลของตัวแปรอาร์เรย์แต่ละตัว เช่น Integer, Long, Single เป็นต้น

ข้อควรระวัง เนื่องจากว่าคำสั่ง ReDim มีความสามารถเพียง เปลี่ยนแปลงจำนวนสมาชิกในตัวแปรอาร์เรย์เท่านั้น ไม่สามารถเปลี่ยนแปลงชนิดข้อมูล ของตัวแปรอาร์เรย์แต่ละตัวได้

ตัวอย่างการใช้งาน

```
Option Explicit
```

```
Dim intMember() As Integer
```

```
Private Sub Form_Load()
```

```
    ReDim intMember(9) As Integer
```

```
    ...
```

```
End Sub
```

ตัวอย่างนี้เป็นการเปลี่ยนแปลง ตัวแปรอาร์เรย์ intMember () แบบไดนามิก ให้กลายเป็นตัวแปรอาร์เรย์แบบสแควร์ ที่มีจำนวนสมาชิก 10 ตัว แต่ถ้าคุณมีการเปลี่ยนแปลงชนิดของข้อมูล ในตัวแปรอาร์เรย์ ก็จะทำให้เกิด error ทันที ดังตัวอย่างโค้ดด้านล่าง

```
Option Explicit
```

```
Dim intMember() As Integer
```

```
Private Sub Form_Load()
```

```
    ReDim intMember(9) As Long
```

End Sub

กลุ่มฟังก์ชันที่เกี่ยวข้องกับตัวแปร Variant

เนื่องจากตัวแปร Variant สามารถเก็บข้อมูลได้ทุกชนิด ฟังก์ชันในกลุ่มนี้ จะมีประโยชน์ ในกรณีที่คุณไม่สามารถหลีกเลี่ยง การใช้งานตัวแปรชนิด Variant ได้ ซึ่งประกอบด้วยกลุ่มของฟังก์ชัน ดังนี้

- ฟังก์ชัน Istype ()
- ฟังก์ชัน TypeName ()
- ฟังก์ชัน Vartype ()

สำหรับรายละเอียดของแต่ละฟังก์ชัน มีดังนี้

- ฟังก์ชัน Istype ()

หน้าที่ ใช้สำหรับตรวจสอบชนิดของข้อมูล ในตัวแปร Variant ว่าเป็นข้อมูลชนิดใด รูปแบบการใช้งาน

IsArray(varname)

IsDate(expression)

IsEmpty(expression)

IsError(expression)

IsMissing(argname)

IsNull(expression)

IsNumeric(expression)

IsObject(identifier)

ตัวแปร varname, expression, argname และ identifier หมายถึง ตัวแปร Variant ที่คุณต้องการตรวจสอบ

ค่าที่ส่งกลับ เป็นค่า Boolean ซึ่งมีความหมายดังนี้

ค่าที่ได้	ความหมาย
True	หมายถึง ตัวแปร Variant เก็บข้อมูลชนิดเดียวกับฟังก์ชัน ที่คุณ ใช้ตรวจสอบ
False	หมายถึง ตัวแปร Variant ไม่ได้เก็บข้อมูลชนิดเดียวกับฟังก์ชัน ที่คุณ ใช้ตรวจสอบ

ข้อแนะนำในการใช้งาน เนื่องจากว่า ตัวแปร Variant สามารถเก็บข้อมูลได้ทุกชนิด ฟังก์ชันนี้จึงมีประโยชน์ ในกรณีที่คุณต้องการตรวจสอบว่า ตัวแปร Variant ดังกล่าว เก็บข้อมูลชนิดใดอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ฟังก์ชัน TypeName ()

หน้าที่ ใช้สำหรับรายงานข้อมูลของตัวแปร Variant
รูปแบบการใช้งาน

TypeName(varname)

ตัวแปร varname หมายถึง ตัวแปร Variant ที่คุณต้องการตรวจสอบ

ค่าที่ส่งกลับ สตรีง ซึ่งเป็นข้อมูลที่เกี่ยวข้องกับตัวแปร Variant ซึ่งมีรายละเอียดดังนี้

ค่าที่ส่งกลับมา	ความหมาย
ชื่อของอ็อบเจกต์ ใดๆ	ตัวแปรนั้นๆ เก็บอ็อบเจกต์ดังกล่าวอยู่
Byte	ข้อมูลชนิด Byte
Integer	ข้อมูลชนิด Integer
Long	ข้อมูลชนิด Long
Single	ข้อมูลชนิด Single
Double	ข้อมูลชนิด Double
Currency	ข้อมูลชนิด Currency
Decimal	ข้อมูลชนิด Decimal
Date	ข้อมูลชนิด Date
String	ข้อมูลชนิด String
Boolean	ข้อมูลชนิด Boolean
Error	ข้อมูลชนิด error
Empty	เป็นข้อมูลที่ยังไม่เคยกำหนดค่าเริ่มต้นมาก่อน
Null	ตัวแปรนั้นๆ ไม่ได้จัดเก็บข้อมูลใดๆ
Object	ข้อมูลชนิดอ็อบเจกต์
Unknown	ไม่ทราบชนิดของข้อมูล
Nothing	เป็นตัวแปรอ็อบเจกต์ ซึ่งไม่ได้อ้างอิงถึงอ็อบเจกต์ ใดๆ

ข้อแนะนำในการใช้งาน เนื่องจากว่า หน้าที่ของฟังก์ชันนี้ จะมีลักษณะคล้ายกับฟังก์ชัน Istype () ต่างกัน
ตรงที่ ฟังก์ชัน TypeName () คุณจะใช้เมื่อ ต้องการทราบเพียงอย่างเดียวว่า ตัวแปร Variant เก็บข้อมูล
ชนิดใดอยู่ เป็นการตรวจสอบเบื้องต้นเท่านั้น ส่วนฟังก์ชัน Istype () จะใช้เมื่อ คุณต้องการทราบว่า ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แปร Variant เก็บข้อมูลชนิดใดเช่นกัน แต่ตัวแปรดังกล่าว คุณต้องการตรวจสอบให้แน่ชัด หรือต้องการแปลงข้อมูลจากชนิด Variant ไปเป็นชนิดอื่นๆ จะเห็นได้ว่า การตรวจสอบตัวแปร Variant ด้วยฟังก์ชัน IType () จะมีความละเอียด และเน้นการตรวจสอบที่ชัดเจนมากกว่า

ถ้าตัวแปรที่นำมาตรวจสอบ เป็นตัวแปรอาร์เรย์ ฟังก์ชัน TypeName () จะคืนค่าเป็น ชนิดของข้อมูลนั้นๆ เหมือนกับตารางด้านบน และจะคืนค่า () ต่อท้าย ซึ่งมีความหมายว่า เป็นตัวแปรอาร์เรย์ชนิดนั้นๆ เช่น intStd () As Integer เมื่อคุณนำมาตรวจสอบด้วยฟังก์ชัน TypeName (intStd) ค่าที่คุณได้จากฟังก์ชันนี้คือ Integer () เป็นต้น

- ฟังก์ชัน VarType ()

หน้าที่ ใช้สำหรับตรวจสอบตัวแปรชนิด Variant ว่า ตัวแปรดังกล่าวเก็บข้อมูลชนิดใด เนื่องจากว่า ตัวแปรชนิด Variant สามารถทดแทนข้อมูลได้ทุกชนิด เราสามารถใช้ฟังก์ชันนี้ เพื่อตรวจสอบชนิดของข้อมูลที่ถูกจัดเก็บอยู่ในตัวแปร Variant ได้

VarType (expression)

ตัวแปร expression หมายถึง ตัวแปร Variant ที่คุณต้องการตรวจสอบค่าที่ส่งกลับ เป็นเลขจำนวนเต็ม Integer ซึ่งมีความหมายดังนี้

ค่าคงที่	ค่าตัวเลข	รายละเอียด
vbEmpty	0	Empty หมายถึง เป็นตัวแปร Variant ที่คุณยังไม่ได้กำหนดค่าอะไรให้กับตัวแปรดังกล่าวเลย
vbNull	1	Null หมายถึง ไม่มีข้อมูล
vbInteger	2	ข้อมูลชนิด Integer
vbLong	3	ข้อมูลชนิด Long
vbSingle	4	ข้อมูลชนิด Single
vbDouble	5	ข้อมูลชนิด Double
vbCurrency	6	ข้อมูลชนิด Currency
vbDate	7	ข้อมูลชนิด Date
vbString	8	ข้อมูลชนิด String
vbObject	9	ข้อมูลชนิด Object

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

vbError	10	ข้อมูลชนิด Error
vbBoolean	11	ข้อมูลชนิด Boolean
vbVariant	12	ข้อมูลชนิด Variant (ใช้กับอาร์เรย์ของ variants เท่านั้น)
vbDataObject	13	ข้อมูลชนิด data access object
vbDecimal	14	ข้อมูลชนิด Decimal
vbByte	17	ข้อมูลชนิด Byte
vbUserDefinedType	36	ตัวแปรดังกล่าว เก็บข้อมูลชนิดที่ผู้ใช้กำหนดขึ้นเอง (user-defined types)
vbArray	8192	ข้อมูลชนิด Array

ตัวอย่างการใช้งาน

```
Private Sub Form_Activate()
```

```
Dim A As Variant 'กำหนดให้ตัวแปร A เป็นข้อมูลชนิด Variant
```

```
Print "A is: "; VarType(A) 'แสดงค่า A ที่ฟอร์ม
```

```
A = 3 'กำหนดค่า A
```

```
Print "A is: "; VarType(A) 'แสดงค่า A ที่ฟอร์ม
```

```
A = 32769 'กำหนดค่า A
```

```
Print "A is: "; VarType(A) 'แสดงค่า A ที่ฟอร์ม
```

```
A = True
```

```
Print "A is: "; VarType(A)
```

```
A = 1234.9999
```

```
Print "A is: "; VarType(A)
```

```
A = #7/25/1999 10:00:00 PM#
```

```
Print "A is: "; VarType(A)
```

```
A = "Suphachai Somphanit"
```

```
Print "A is: "; VarType(A)
```

```
End Sub
```

จากตัวอย่าง ผู้เขียนสมมติให้ตัวแปร A เป็นข้อมูลชนิด Variant แล้วจึงค่อยกำหนดให้ตัวแปร A เก็บค่าต่างๆ กัน จากนั้นจึงใช้ฟังก์ชัน VarType () เพื่อตรวจสอบชนิดของตัวแปร A แล้วแสดงออกมาที่ฟอร์มให้ผู้อ่านสังเกตค่าที่ผู้เขียนกำหนดขึ้นมา ประกอบกับผลที่เกิดขึ้นด้วย

กลุ่มฟังก์ชันที่เกี่ยวข้องกับการแปลงชนิดของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สิ่งหนึ่งที่คุณจะพบได้เสมอ ในการเขียนโปรแกรม ก็คือ การแปลงชนิดของข้อมูลจากชนิดหนึ่ง ไปสู่อีกชนิดหนึ่ง บ่อยครั้งข้อมูลที่ input เข้ามา มักจะไม่ตรงกับความต้องการของคุณ ฟังก์ชันกลุ่มนี้ เป็นอีกกลุ่มหนึ่งที่คุณต้องใช้งาน และทำความเข้าใจเป็นอย่างยิ่ง เพราะว่า ถ้าคุณแปลงข้อมูลที่เข้ามา ผิด, แยกต่าง หรือ ไม่ตรงกับที่คุณต้องการ ก็จะส่งผลให้ output หรือข้อมูลที่ออกมา คลาดเคลื่อนจากที่ควรจะเป็น หรืออาจถึงทำงานผิดพลาดเลยก็ได้ ขอให้คุณทำความเข้าใจฟังก์ชันต่างๆ ที่อยู่ในกลุ่มนี้ให้ดี :-)

- ฟังก์ชัน Asc (), Chr () และ Chr\$ ()

หน้าที่ ฟังก์ชัน Asc () ใช้สำหรับแปลงตัวอักษรตัวแรกของข้อมูล ให้เป็นรหัสแอสกี ซึ่งมีค่าตั้งแต่ 0-255 ส่วนฟังก์ชัน Chr () ใช้สำหรับแปลงข้อมูลจากรหัสแอสกีกลับเป็นตัวอักษร ทั้ง 2 ฟังก์ชัน มีหน้าที่ตรงข้ามกันนั่นเอง

รูปแบบการใช้งาน

Asc (string)

Chr (charcode)

Chr\$ (charcode)

ตัวแปร string และ charcode หมายถึง ตัวแปรที่คุณต้องการแปลงค่า

ข้อแตกต่างของฟังก์ชัน Chr () และ Chr\$ () คือ ค่าที่ฟังก์ชัน Chr\$ () ส่งกลับมา จะเป็นข้อมูลชนิด String ส่วนฟังก์ชัน Chr () ค่าที่ส่งกลับมา จะเป็นชนิด Variant

- ฟังก์ชัน CBool ()

หน้าที่ ใช้สำหรับแปลงข้อมูล ให้เป็นชนิดของข้อมูล Boolean

รูปแบบการใช้งาน

CBool(expression)

ตัวแปร expression หมายถึง ตัวแปร หรือข้อมูลที่คุณต้องการแปลง

ค่าที่ส่งกลับ โดยหน้าที่แล้วจะเป็นข้อมูลชนิดตัวแปร Boolean แต่ก็ขึ้นอยู่กับข้อมูลที่เข้ามาด้วย ดังตารางต่อไปนี้

ชนิดของตัวแปร expression	ค่าที่ได้
ตัวเลขใดๆ (Numeric)	<ul style="list-style-type: none"> • กรณีที่เป็นเลข 0 ซึ่งรวมถึงเลขทศนิยมที่ขึ้นต้นด้วย 0.x ด้วย เช่น 0.1, 0.01 เป็นต้น จะถูกแปลงค่าเป็น False • ส่วนตัวเลขอื่นๆ ทุกตัว จะถูกแปลงเป็นค่า True

ข้อความ (String)	<p>ถ้าเป็นข้อความที่เป็นตัวอักษรจะ error แต่ถ้าเป็นข้อความที่เป็นตัวเลข จะแยกเป็นกรณีดังนี้</p> <ul style="list-style-type: none"> • กรณีเลข 0 แต่ไม่รวมที่เป็นเลขทศนิยมที่ขึ้นต้นด้วย 0.x จะถูกแปลงค่าเป็น False • กรณีตัวเลขอื่นๆ ทุกตัวรวมทั้งเลขทศนิยมที่ขึ้นต้นด้วย 0.x จะถูกแปลงค่าเป็น True ซึ่งจะแตกต่างจากกรณีที่เป็น Numeric
Null	Error
Empty	False

- ฟังก์ชัน CByte ()

หน้าที่ ใช้สำหรับแปลงข้อมูล ให้เป็นชนิดของข้อมูล Byte
รูปแบบการใช้งาน

CByte (expression)

ตัวแปร expression หมายถึง ตัวแปร หรือข้อมูลที่คุณต้องการแปลง

ค่าที่ส่งกลับ โดยหน้าที่แล้วจะเป็นข้อมูลชนิดตัวเลข Byte แต่ก็จะขึ้นอยู่กับข้อมูลที่เข้ามาด้วย ดังตารางต่อไปนี้

ชนิดของตัวแปร expression	ค่าที่ได้
ตัวเลขใดๆ (Numeric)	<p>แยกได้ 2 กรณี คือ</p> <ul style="list-style-type: none"> • ตัวเลขจำนวนเต็ม ที่อยู่ระหว่าง 0-255 เท่านั้น จึงจะสามารถแปลงค่าได้ • ถ้าเป็นตัวเลขทศนิยม จะใช้หลักการทางคณิตศาสตร์ เพื่อปัดเศษทศนิยม ขึ้นหรือลงก่อน แล้วจึงแปลงค่า
ค่า Boolean	<p>แยกได้ 2 กรณี คือ</p> <ul style="list-style-type: none"> • True จะแปลงเป็นค่า 255 • False จะแปลงเป็นค่า 0
ข้อความ (String)	ถ้าเป็นข้อความที่เป็นตัวอักษรจะ error แต่ถ้าเป็นข้อความที่เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	ตัวเลข จะต้องอยู่ระหว่าง 0-255 เท่านั้น
Null	Error
Empty	0

- ฟังก์ชัน CDate ()

หน้าที่ ใช้สำหรับแปลงข้อมูล ให้เป็นชนิดของข้อมูล Date
รูปแบบการใช้งาน

CDate (expression)

ตัวแปร expression หมายถึง ตัวแปร หรือข้อมูลที่ผู้ต้องการแปลง

ค่าที่ส่งกลับ โดยหน้าที่แล้วจะเป็นข้อมูลชนิด Date แต่ก็จะขึ้นอยู่กับข้อมูลที่เข้ามาด้วย ดังตารางต่อไปนี้

ชนิดของตัวแปร expression	ค่าที่ได้
ตัวเลขใดๆ (Numeric)	<p>แยกเป็นกรณีได้ดังนี้</p> <ul style="list-style-type: none"> เลข -65535 จะแปลงค่าเป็น 7/26/1720 เลข -255 จะแปลงค่าเป็น 4/19/1899 เลข -1 จะแปลงค่าเป็น 12/29/1899 **เลข 0 จะแปลงค่าเป็น 12:00:00 AM เลข 1 จะแปลงค่าเป็น 12/31/1899 เลข 2 จะแปลงค่าเป็น 1/1/1900 เลข 3 จะแปลงค่าเป็น 1/2/1900 เลข 255 จะแปลงค่าเป็น 9/11/1900 เลข 65535 จะแปลงค่าเป็น 6/4/2079 <p>**สำหรับเลข 0 ผู้เขียนคิดว่า น่าจะเป็น bug ของ VB 6.0 เนื่องจาก ว่า ควรจะแปลงเป็น 12/30/1899</p>
ค่า Boolean	<ul style="list-style-type: none"> ค่า True จะแปลงค่าเป็น 12/29/1899

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	<ul style="list-style-type: none"> ค่า False จะแปลงค่าเป็น 12:00:00 AM
ข้อความ (String)	<p>ถ้าเป็นข้อความธรรมดาจะ error จะต้องเป็นข้อความที่มีรูปแบบหรือลักษณะ วันและเวลา เช่น</p> <pre>Private Sub Form_Load() Dim a As String Dim b As Date a = "1/1/1999 12:00:00 PM" b = CDate(a) Debug.Print b End Sub</pre>
Null	Error
Empty	12:00:00 AM
สามารถมีค่าได้ตั้งแต่วันที่ 1 January 100 ถึงวันที่ 31 December 9999 และเก็บเวลาได้ตั้งแต่ 0:00:00 ถึง 23:59:59	

- ฟังก์ชัน CDbl (), CInt (), CLng () และ CSng ()

หน้าที่ ใช้สำหรับแปลงข้อมูล ให้เป็นข้อมูลชนิดเดียวกับฟังก์ชันนั้นๆ โดยมีขอบเขตของข้อมูลเท่ากับข้อมูลชนิดนั้นๆ คือ

- ฟังก์ชัน CDbl () ใช้แปลงข้อมูลให้เป็นชนิด Double มีขอบเขตดังนี้ ค่าบวกอยู่ระหว่าง 4.94065645841247E-324 ถึง 1.79769313486232E308 ค่าลบอยู่ระหว่าง -1.79769313486232E308 ถึง -4.94065645841247E-324
- ฟังก์ชัน CLng () ใช้แปลงข้อมูลให้เป็นชนิด Long มีขอบเขตตั้งแต่ -2,147,483,648 ถึง 2,147,483,647
- ฟังก์ชัน CSng () ใช้แปลงข้อมูลให้เป็นชนิด Single มีขอบเขตดังนี้ ค่าบวกอยู่ระหว่าง 1.401298E-45 ถึง 3.402823E38 และค่าลบอยู่ระหว่าง -3.402823E38 ถึง -1.401298E-45

รูปแบบการใช้งาน

CDbl (expression)

CInt (expression)

CLng (expression)

CSng (expression)

ตัวแปร expression หมายถึง ตัวแปร หรือข้อมูลที่คุณต้องการแปลง

ค่าที่ส่งกลับ จะขึ้นอยู่กับตัวแปรที่เข้ามา ดังตารางต่อไปนี้

ชนิดของตัวแปร expression	ค่าที่ได้
ตัวเลขใดๆ (Numeric)	จะแปลงเป็นค่าที่เท่ากับค่าเดิม แต่เปลี่ยนจากชนิดของข้อมูลเดิม เป็นข้อมูลชนิดเดียวกับฟังก์ชันที่ใช้
ค่า Boolean	แยกได้ 2 กรณี คือ <ul style="list-style-type: none">• True จะแปลงเป็นค่า -1• False จะแปลงเป็นค่า 0
ข้อความ (String)	ถ้าเป็นข้อความที่เป็นตัวอักษรจะ error แต่ถ้าเป็นข้อความที่เป็นตัวเลข จะแปลงเป็นค่า ที่เท่ากับค่าเดิม แต่เปลี่ยนจากชนิดของข้อมูล String เป็นข้อมูลชนิดเดียวกับฟังก์ชันที่ใช้
Null	Error
Empty	0

- ฟังก์ชัน CStr ()

หน้าที่ ใช้สำหรับแปลงข้อมูล ให้เป็นข้อมูลชนิด String
รูปแบบการใช้งาน

CStr (expression)

ตัวแปร expression หมายถึง ตัวแปร หรือข้อมูลที่คุณต้องการแปลง

ค่าที่ส่งกลับ โดยหน้าที่แล้วจะเป็นข้อมูลชนิด String แต่ก็จะขึ้นอยู่กับข้อมูลที่เข้ามาด้วย ดังตารางต่อไปนี้

ชนิดของตัวแปร expression	ค่าที่ได้
ตัวเลขใดๆ (Numeric)	จะแปลงเป็นค่าที่เท่ากับค่าเดิม แต่เปลี่ยนจากชนิดของข้อมูลเดิม เป็นข้อมูลชนิด String
ค่า Boolean	จะคงค่าเดิมไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Null	Error
Empty	จะแปลงเป็น String ว่าง " "

- ฟังก์ชัน CVar ()

เนื่องจาก ข้อมูลชนิด Variant สามารถเก็บข้อมูลได้ทุกชนิดอยู่แล้ว การแปลงข้อมูลใดๆ ให้เป็นชนิด Variant จึงสามารถแปลงข้อมูล ได้อย่างถูกต้อง ครบถ้วนทุกประการ เพียงแค่เปลี่ยนชนิดของข้อมูลจากเดิม เป็นชนิด Variant เท่านั้น ซึ่งผู้เขียนเห็นว่า ไม่มีประโยชน์อะไร ที่จะใช้ทรัพยากรระบบมากกว่า โดยที่ยังคงเก็บค่าเดิมไว้ แต่จะมีประโยชน์ในแง่ที่ เพิ่มความยืดหยุ่นของข้อมูลที่เข้ามา ให้มีความหลากหลายยิ่งขึ้นเท่านั้น และใช้ในกรณีที่มีการประมวลผล, คำสั่งใดๆ, ฟังก์ชันใดๆ และอ็อบเจกต์ใดๆ ต้องการแค่ข้อมูลชนิด Variant เท่านั้น

- ฟังก์ชัน Str () และฟังก์ชัน Str\$ ()

หน้าที่ ใช้สำหรับแปลงข้อมูลจาก ตัวเลขไปเป็นสตริงตัวเลข กล่าวคือ เป็นการเปลี่ยนสถานะจากชนิดของข้อมูลตัวเลข เช่น Integer, Long, Single etc. ไปเป็นสถานะข้อมูลชนิด String นั่นเอง รูปแบบการใช้งาน

Str (numeric)

ตัวแปร numeric หมายถึง ตัวเลขใดๆ ที่คุณต้องการแปลง

ข้อแตกต่างของฟังก์ชัน Str () และฟังก์ชัน Str\$ () คือ ค่าที่ฟังก์ชัน Str () ส่งกลับจะเป็นข้อมูลชนิด Variant ส่วนฟังก์ชัน Str\$ () จะส่งค่ากลับเป็นข้อมูลชนิด String

- ฟังก์ชัน Val ()

หน้าที่ แปลงตัวแปร หรือข้อมูล ให้เป็นข้อมูลชนิดตัวเลขจำนวนจริง Double รูปแบบการใช้งาน

Val (string)

ตัวแปร string หมายถึง ตัวแปร หรือข้อมูลที่คุณต้องการแปลงค่า

ค่าที่ส่งกลับ โดยหน้าที่แล้วจะเป็นข้อมูลชนิดตัวเลขจำนวนจริง Double แต่ที่ขึ้นอยู่กับข้อมูลที่เข้ามาด้วย ** ผู้เขียนคิดว่า น่าจะเป็น bug ของ VB 6.0 กรณีแรก ควรที่จะแปลงอักษรตัวแรกของข้อความที่เข้ามา ให้เป็นเลขจำนวนเต็มบวก Double ที่มีค่าเท่ากับ รหัส ASCII ของตัวอักษรนั้นๆ ไม่ใช่แปลงเป็น 0 ทั้งหมด ส่วนกรณีที่ 2 ค่า Boolean ควรที่จะแปลงค่า False เป็น 0 และแปลงค่า True ให้เป็น -1

ข้อแนะนำในการใช้งาน ฟังก์ชันนี้มักจะใช้ควบคู่กับคอนโทรล TextBox เสมอ เนื่องจากคอนโทรล TextBox เป็นคอนโทรลที่ผู้ใช้สามารถ Input ข้อมูลต่างๆ เข้ามาได้ สมมติว่า ผู้ใช้ป้อนตัวเลขเข้ามา ตัวเลขดังกล่าวจะถูกเก็บไว้ที่คุณสมบัติ Text ซึ่งเป็นตัวเลขที่มีสถานะเป็น String ทำให้ไม่สามารถนำไปประมวลทางคณิตศาสตร์ได้อย่างเต็มรูปแบบ ฟังก์ชันนี้จะทำหน้าที่ แปลงข้อมูลตัวเลข (ที่อยู่ในฐานะ String) ให้กลายเป็นตัวเลข (ที่อยู่ในฐานะตัวเลข Double)

ตัวอย่างในการใช้งาน

Dim Test As String

Test = Val("1234") 'คืนค่า 1234

Test = Val(" 1 2 3 4") 'คืนค่า 1234

Test = Val("12 and 34") 'คืนค่า 12

ผู้ใช้ป้อนค่า 12345 เข้ามาในคอนโทรล TextBox

Test = Val(Text1.Text) 'คืนค่า 12345

กลุ่มฟังก์ชันสำหรับสร้างกรอบโต้ตอบผู้ใช้

- ฟังก์ชัน MsgBox () และคำสั่ง MsgBox

หน้าที่ใช้สำหรับสร้างกรอบโต้ตอบกับผู้ใช้ แบบเมสเสจบ็อกซ์ หมายถึง กรอบโต้ตอบที่แสดงข้อมูลเพียงอย่างเดียว แล้วให้ผู้ใช้คลิกเลือก เพื่อตอบรับหรือไม่ เช่น เมสเสจบ็อกซ์ที่เตือนให้ผู้ใช้เซฟงานใน Word 2000 หรืออื่นๆ เป็นต้น

รูปแบบการใช้งาน

MsgBox(prompt[, buttons] [, title] [, helpfile, context])

MsgBox prompt[, buttons] [, title] [, helpfile, context]

ตัวแปร prompt ข้อความที่คุณต้องการแสดงในเมสเสจบ็อกซ์ ในกรณีที่คุณต้องการแสดงข้อความมากกว่า 1 บรรทัดในเมสเสจบ็อกซ์ ให้คุณใช้อักขระ Chr(13) & Chr(10) คั่นข้อความ เพื่อให้ข้อความต่อไปขึ้นบรรทัดใหม่

ตัวแปร buttons หมายถึง เลขจำนวนเต็ม Integer ใช้สำหรับกำหนดปุ่ม, ไอคอน และรูปแบบ ที่คุณต้องการแสดงในเมสเสจบ็อกซ์ โดยแยกออกเป็น 5 กลุ่มใหญ่ๆ และจะใช้เครื่องหมาย + เพื่อแยกกลุ่มต่างๆ ออกจากกัน ซึ่งสามารถมีค่าได้ดังนี้

กลุ่มที่ใช้สำหรับกำหนดปุ่ม ที่จะแสดงในเมสเสจบ็อกซ์		
ค่าคงที่	ค่าตัวเลข	รายละเอียด
vbOKOnly	0	แสดงปุ่ม OK ปุ่มเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

vbOKCancel	1	แสดงปุ่ม OK และปุ่ม Cancel
vbAbortRetryIgnore	2	แสดงปุ่ม Abort, Retry, และ Ignore
vbYesNoCancel	3	แสดงปุ่ม Yes, No, และ Cancel
vbYesNo	4	แสดงปุ่ม Yes และ No
vbRetryCancel	5	แสดงปุ่ม Retry และ Cancel

กลุ่มที่ใช้สำหรับกำหนดไอคอน ที่จะแสดงในเมสเสจบ็อกซ์		
ค่าคงที่	ค่าตัวเลข	รายละเอียด
vbCritical	16	แสดงไอคอน Critical Message
vbQuestion	32	แสดงไอคอน Warning Query
vbExclamation	48	แสดง ไอคอน Warning Message
vbInformation	64	แสดง ไอคอน Information Message

กลุ่มที่ใช้สำหรับกำหนดปุ่ม Default		
ค่าคงที่	ค่าตัวเลข	รายละเอียด
vbDefaultButton1	0	กำหนดให้ปุ่มแรก เป็นปุ่ม Default
vbDefaultButton2	256	กำหนดให้ปุ่มที่ 2 เป็นปุ่ม Default
vbDefaultButton3	512	กำหนดให้ปุ่มที่ 3 เป็นปุ่ม Default
vbDefaultButton4	768	กำหนดให้ปุ่มที่ 4 เป็นปุ่ม Default

กลุ่มที่ใช้กำหนดรูปแบบของเมสเสจบ็อกซ์		
ค่าคงที่	ค่าตัวเลข	รายละเอียด
vbApplicationModal	0	กำหนดให้แสดงเมสเสจบ็อกซ์แบบ Application modal หมายถึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		ถึง ผู้ใช้ไม่จำเป็นต้องตอบสนองเมสเสจบ็อกซ์นี้ ก็ สามารถสลับ ไปทำอย่างอื่นก่อนก็ได้
vbSystemModal	4096	กำหนดให้แสดงเมสเสจบ็อกซ์แบบ System modal หมายถึง ผู้ใช้จะต้อง คลิกเลือกปุ่มใดปุ่มหนึ่ง เพื่อตอบสนองในเมส เสจบ็อกซ์ก่อนเท่านั้น จึงจะสามารถกระทำอย่างอื่นต่อไปได้

กลุ่มที่ใช้สำหรับกำหนดลักษณะพิเศษของเมสเสจบ็อกซ์		
ค่าคงที่	ค่าตัวเลข	รายละเอียด
vbMsgBoxHelpButton	16384	เพิ่มปุ่ม Help เข้าไปในเมสเสจบ็อกซ์
VbMsgBoxSetForeground	65536	กำหนดให้เมสเสจบ็อกซ์ แสดงส่วนบนตลอด เวลา
vbMsgBoxRight	524288	ข้อความชิดขวาตลอดเวลา ซึ่งจะใช้กับภาษา Hebrew และ Arabic
vbMsgBoxRtlReading	1048576	กำหนดให้ข้อความที่ปรากฏออกมา มีลักษณะ อ่านจากขวาไปซ้าย ซึ่งจะใช้ในภาษา Hebrew และ Arabic

ข้อแตกต่างของฟังก์ชัน MsgBox () และคำสั่ง MsgBox คือ ถ้าคุณใช้งานในลักษณะของคำสั่ง MsgBox หมายถึง จะไม่มีการส่งค่าใดๆ กลับมา แต่ถ้าคุณใช้งานในลักษณะฟังก์ชัน MsgBox () ก็จะมีการส่งค่ากลับมา 1 ค่า ซึ่งก็คือ ค่าประจำปุ่มที่ผู้ใช้คลิกเลือกในเมสเสจบ็อกซ์นั่นเอง คุณจะต้องนำค่านี้ ไปเป็นเงื่อนไขในการประมวลผลต่อไป สำหรับค่าที่ส่งกลับมา มีความหมายดังนี้

ค่าคงที่	ค่าตัวเลข	รายละเอียด
vbOK	1	หมายถึง ผู้ใช้คลิกปุ่ม OK
vbCancel	2	หมายถึง ผู้ใช้คลิกปุ่ม Cancel
vbAbort	3	หมายถึง ผู้ใช้คลิกปุ่ม Abort
vbRetry	4	หมายถึง ผู้ใช้คลิกปุ่ม Retry
vbIgnore	5	หมายถึง ผู้ใช้คลิกปุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		Ignore
vbYes	6	หมายถึง ผู้ใช้คลิกปุ่ม Yes
vbNo	7	หมายถึง ผู้ใช้คลิกปุ่ม No

ตัวแปร title หมายถึง ข้อความที่คุณต้องการแสดงบนแถบไตเติลบาร์ของเมสเสจบ็อกซ์ สำหรับตัวแปร helpfile และ context หมายถึง การอ้างอิงระบบไฟล์ช่วยเหลือ ซึ่งมีนามสกุล *.hlp โดยคุณต้องกำหนดหมายเลข context ในตัวแปร helpfile และรายละเอียดประจำหัวข้อในตัวแปร context

- ฟังก์ชัน InputBox () และคำสั่ง InputBox

หน้าที่ ใช้สำหรับสร้างกรอบโต้ตอบผู้ใช้ แบบอินพุตบ็อกซ์ หมายถึง คุณสามารถแสดงข้อความเพื่อให้ผู้ใช้ตอบรับหรือไม่ เช่นเดียวกับแบบเมสเสจบ็อกซ์ แต่ผู้ใช้สามารถป้อนข้อความเข้าไปในอินพุตบ็อกซ์ได้อีกด้วย

รูปแบบการใช้งาน

InputBox(prompt[, title] [, default] [, xpos] [, ypos] [, helpfile, context])

ตัวแปร prompt หมายถึง ข้อความที่คุณต้องการแสดงในอินพุตบ็อกซ์ ในกรณีที่คุณต้องการแสดงข้อความมากกว่า 1 บรรทัดในอินพุตบ็อกซ์ ให้คุณใช้อักขระ Chr(13) & Chr(10) ต่อท้ายข้อความ เพื่อให้ข้อความต่อไป ขึ้นบรรทัดใหม่

ตัวแปร title หมายถึง ข้อความที่คุณต้องการแสดงบนแถบไตเติลบาร์ของอินพุตบ็อกซ์

ตัวแปร default หมายถึง ข้อความที่คุณกำหนดให้แสดงขึ้นมา ในช่องกรอกข้อความ โดยอัตโนมัติ เมื่ออินพุตบ็อกซ์ปรากฏขึ้นมา

ตัวแปร xpos และ ypos หมายถึง ตำแหน่งของอินพุตบ็อกซ์เมื่อปรากฏขึ้นมาบนจอภาพ โดยวัดแบบคู่ลำดับ co-ordinate มีหน่วยเป็น twip (หรือขึ้นอยู่กับคุณสมบัติ ScaleMode ของฟอร์ม) ซึ่งวัดจากมุมซ้ายบนของอินพุตบ็อกซ์ กับมุมซ้ายบนของจอภาพ ถ้าคุณไม่กำหนด อินพุตบ็อกซ์จะปรากฏตรงตำแหน่งกึ่งกลางจอภาพ

สำหรับตัวแปร helpfile และ context หมายถึง การอ้างอิงระบบไฟล์ช่วยเหลือ ซึ่งมีนามสกุล *.hlp โดยคุณต้องกำหนดหมายเลข context ในตัวแปร helpfile และรายละเอียดประจำหัวข้อในตัวแปร context

กลุ่มฟังก์ชันและคำสั่งอื่นๆ ที่น่าสนใจ

กลุ่มฟังก์ชันและคำสั่งที่ถูกจัดอยู่ในกลุ่มนี้ ผู้เขียนไม่สามารถจัดลงกลุ่มอื่นๆ ได้ จึงรวบรวมมาอยู่ในส่วนนี้

- ฟังก์ชัน Format () และ Format\$ ()

หน้าที่ ใช้สำหรับจัดรูปแบบตัวเลข หรือข้อมูล โดยเฉพาะตัวเลข หรือข้อมูลที่เป็นตัวเลข, วันและเวลา รูปแบบการใช้งาน

Format(expression[, format[, firstdayofweek[, firstweekofyear]])

ตัวเลข expression หมายถึง ตัวเลขหรือข้อมูลที่คุณต้องการจัดรูปแบบ

ตัวเลข format หมายถึง ตัวเลขที่กำหนดรูปแบบให้กับข้อมูลที่ต้องการจัด ซึ่งสามารถกำหนดค่าได้ดังนี้

ตัวอักษร	ความหมาย
0	จัดรูปแบบโดยการเติมเลข 0 เพิ่มลงไปข้อมูล (โดยที่ค่าไม่เปลี่ยนแปลง) ให้เท่ากับจำนวนเลข 0 ที่คุณกำหนดไว้
#	จัดรูปแบบโดยการบังคับให้ข้อมูล หรือตัวเลข มีจำนวนหลักเท่ากับเครื่องหมาย # ถ้าข้อมูลมีจำนวนหลักน้อยกว่า เครื่องหมาย # ก็จะเว้นช่องว่างไว้ เพื่อให้ข้อมูลมีจำนวนหลักเท่ากับเครื่องหมาย #
%	จัดรูปแบบโดยการคำนวณข้อมูลเป็นเปอร์เซ็นต์ก่อน แล้วจึงแสดงข้อมูลออกมามีเครื่องหมาย % ต่อท้ายข้อมูลดังกล่าวด้วย
'	ใช้จัดรูปแบบข้อมูลที่เป็นตัวเลข โดยแสดงเครื่องหมาย , แบ่งตำแหน่งตัวเลขครั้งละ 3 ตำแหน่ง
:	ใช้จัดรูปแบบข้อมูลที่เป็นเวลา เป็นตัวคั่นระหว่างหน่วยเวลาดังนี้ ชั่วโมง:นาที:วินาที
/	ใช้จัดข้อมูลที่เป็นวันที่ เป็นตัวคั่นระหว่างหน่วยวันดังนี้ วันที่/เดือน/ปี

ตัวเลข firstdayofweek หมายถึง กำหนดให้วันใดวันหนึ่งในสัปดาห์ เป็นวันเริ่มต้น หรือเป็นวันแรก ซึ่งมีค่าได้ดังนี้

ค่าคงที่	ค่าตัวเลข	รายละเอียด
vbUseSystem	0	ใช้ค่าที่กำหนดใน NLS API
VbSunday	1	(default) วันอาทิตย์
vbMonday	2	วันจันทร์
vbTuesday	3	วันอังคาร
vbWednesday	4	วันพุธ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

vbThursday	5	วันพฤหัสบดี
vbFriday	6	วันศุกร์
vbSaturday	7	วันเสาร์

ตัวแปร firstweekofyear หมายถึง กำหนดให้สัปดาห์ใดสัปดาห์หนึ่ง เป็นสัปดาห์แรกของปี มีค่าได้ดังนี้

ค่าคงที่	ค่าตัวเลข	รายละเอียด
vbUseSystem	0	ใช้ค่าที่กำหนดใน NLS API
vbFirstJan1	1	(default) โดยใช้สัปดาห์แรกของเดือนมกราคม (วันที่ 1 มกราคม)
vbFirstFourDays	2	กำหนดให้ใช้สัปดาห์แรก ที่มีจำนวนวันใน 1 สัปดาห์ อย่างน้อยที่สุด 4 วัน
vbFirstFullWeek	3	กำหนดให้ใช้สัปดาห์แรก ที่มีจำนวนวันใน 1 สัปดาห์ ครบทั้ง 7 วัน

ข้อแตกต่างของฟังก์ชัน Format () และฟังก์ชัน Format\$ () คือ ฟังก์ชัน Format () ค่าที่คืนกลับมาจะเป็นข้อมูลชนิด Variant ส่วนฟังก์ชัน Format\$ () ค่าที่คืนกลับมา เป็นข้อมูลชนิด String

- คำสั่ง With...End With

หน้าที่ ใช้สำหรับแก้ไขคุณสมบัติหรือเมธอด ของอ็อบเจกต์หรือคอนโทรลหลายๆ อย่างพร้อมกัน รูปแบบการใช้งาน

With object

[statements]

End With

ตัวแปร object หมายถึง อ็อบเจกต์หรือคอนโทรลใดๆ ที่คุณต้องการแก้ไข

ตัวแปร statements หมายถึง คุณสมบัติหรือเมธอดของอ็อบเจกต์ ที่คุณต้องการแก้ไข เช่น

ตัวอย่างการใช้งาน

สมมติว่าคุณต้องการแก้ไขคุณสมบัติของคอนโทรล TextBox ถ้าโดยปกติแล้ว คุณต้องเขียน โค้ดดังนี้

```
Private Sub Form_Load()
```

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

```
Text1.Alignment = 2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Text1.BorderStyle = 1
```

```
End Sub
```

แต่ถ้าคุณ ใช้คำสั่ง With...End With ช่วย ก็จะทำให้โค้ดของคุณเรียบร้อยยิ่งขึ้น ดังนี้

```
Private Sub Form_Load()
```

```
With Text1
```

```
.Text = "Your Text"
```

```
.Alignment = 2
```

```
.BorderStyle = 1
```

```
End With
```

```
End Sub
```

การสร้างโปรซีเจอร์ทั่วไป (General Procedure)

ถึงแม้ว่า VB จะมีชุดคำสั่ง และฟังก์ชันมาตรฐานอยู่มากมาย แต่ก็ยังไม่สามารถครอบคลุมการทำงานได้ทั้งหมด คุณสามารถสร้างชุดคำสั่ง หรือฟังก์ชัน ขึ้นมาใช้งานเองได้ ซึ่งแยกออกได้ 2 แบบใหญ่ๆ คือ

- **ซับรูทีน (Sub routine)** มีลักษณะเหมือนกับคำสั่งมาตรฐานใน VB กล่าวคือ เป็นชุดคำสั่งที่ไม่มี การส่งค่ากลับ
- **ฟังก์ชัน (Function)** มีลักษณะเช่นเดียวกับฟังก์ชันมาตรฐานใน VB ซึ่งจะส่งค่ากลับออกมา 1 ค่าเสมอ

การสร้างซับรูทีน (Sub routine)

จะใช้คำสั่ง Sub เพื่อต้องการสร้างโปรซีเจอร์ทั่วไป แบบซับรูทีน ซึ่งมีโครงสร้าง ดังนี้

```
[Private | Public] [Static] Sub name [(arglist)]
```

```
[statements]
```

```
[Exit Sub]
```

```
[statements]
```

```
End Sub
```

คำสั่ง **Private** หมายถึง กำหนดให้ซับรูทีนที่สร้างขึ้นมา มีขอบเขตในการเรียกใช้งานเฉพาะฟอร์ม หรือ โมดูลที่ประกาศเท่านั้น

คำสั่ง **Public** หมายถึง กำหนดให้ซับรูทีนที่สร้างขึ้นมา สามารถถูกเรียกใช้งานได้ทุกส่วน

คำสั่ง **Static** หมายถึง กำหนดให้ซับรูทีนเก็บรักษาค่าที่ได้จากการประมวลผลไว้ เมื่อมีการเรียกใช้งาน ซับรูทีนครั้งต่อๆ ไป ค่าดังกล่าวจะมีค่าเท่าเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง Sub หมายถึง ต้องการสร้าง โพรซีเจอร์ต่างๆ ไป แบบซับรูทีน

ตัวแปร name หมายถึง ชื่อของซับรูทีน

ตัวแปร arglist หมายถึง อาร์กิวเมนต์ที่ต้องใช้ในซับรูทีน ซึ่งจะมีหรือไม่มีก็ได้

ตัวแปร statements หมายถึง ชุดคำสั่งต่างๆ ในซับรูทีน

คำสั่ง Exit Sub หมายถึง บังคับให้ออกจากซับรูทีนนี้ทันที

คำสั่ง End Sub หมายถึง จบโพรซีเจอร์

ตัวอย่างต่อไปนี้ เป็นซับรูทีนที่นิยมใช้งานมากที่สุดอีกตัวหนึ่ง นั่นคือ ซับรูทีนที่ทำหน้าที่เคลื่อนย้าย
ฟอร์ม ไปอยู่กึ่งกลางจอภาพ เมื่อฟอร์มดังกล่าว ถูกโหลดขึ้นมา ซึ่งผู้เขียนเห็นว่า คุณผู้อ่านสามารถทำ
ความเข้าใจกับซับรูทีนนี้ได้ไม่ยาก ดังนี้

Option Explicit

Sub FormCenter(frm As Form)

ซับรูทีนนี้ต้องการอาร์กิวเมนต์ 1 ตัว คือ ฟอร์มที่คุณต้องการจัดตำแหน่งกึ่งกลาง ผ่านทางอาร์กิวเมนต์ frm

Dim x As Integer 'สำหรับเก็บตำแหน่งฟอร์มตามแกน x

Dim y As Integer 'สำหรับเก็บตำแหน่งฟอร์มตามแกน y

x = (Screen.Width - frm.Width) \ 2 'ความกว้างของจอภาพ-ความกว้างของฟอร์มก่อน แล้วหารด้วย 2

แบบผลลัพธ์จำนวนเต็ม เก็บไว้ในตัวแปร x

y = (Screen.Height - frm.Height) \ 2 'ความสูงของจอภาพ-ความสูงของฟอร์มก่อน แล้วหารด้วย 2

แบบผลลัพธ์จำนวนเต็ม เก็บไว้ในตัวแปร y

frm.Move x, y 'ย้ายฟอร์มไปยังตำแหน่ง x และ y

End Sub

Private Sub Form_Load()

เรียกใช้งานซับรูทีน FormCenter โดยใช้คำสั่ง Call ส่วนคำสั่ง Me หมายถึง แทนฟอร์มปัจจุบัน

Call FormCenter(Me)

End Sub

จากตัวอย่างซับรูทีน ถ้าคุณมีการเปลี่ยนแปลงชื่อ (คุณสมบัตินาม Name) ของฟอร์ม ให้คุณนำชื่อดังกล่าว
ใส่แทนคำสั่ง Me เพื่อตั้งให้ฟอร์มนั้นๆ อยู่กึ่งกลางจอภาพ

การสร้างฟังก์ชัน (Function)

จะใช้คำสั่ง Function เพื่อสร้างฟังก์ชันขึ้นมาใช้งานเอง โดยมีโครงสร้าง ดังนี้

[Public | Private] [Static] Function name [(arglist)] As type

[statements]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[Exit Function]

[statements]

End Function

คำสั่ง Private หมายถึง กำหนดให้ฟังก์ชันที่สร้างขึ้นมา มีขอบเขตในการเรียกใช้งานเฉพาะฟอร์ม หรือ โมดูลที่ประกาศเท่านั้น

คำสั่ง Public หมายถึง กำหนดให้ฟังก์ชันที่สร้างขึ้นมา สามารถถูกเรียกใช้งาน ได้ทุกส่วน

คำสั่ง Static หมายถึง กำหนดให้ฟังก์ชันเก็บรักษาค่าที่ได้จากการประมวลผลไว้ เมื่อมีการเรียกใช้งาน ฟังก์ชันครั้งต่อไป ค่าดังกล่าวจะมีค่าเท่าเดิม

คำสั่ง Sub หมายถึง ต้องการสร้าง โปรซีเจอร์ทั่วไป แบบฟังก์ชัน

ตัวแปร name หมายถึง ชื่อของฟังก์ชัน

ตัวแปร arglist หมายถึง อาร์กิวเมนต์ที่ต้องใช้ในฟังก์ชัน ซึ่งจะมีหรือไม่มีก็ได้

ตัวแปร As type หมายถึง ฟังก์ชันนี้ จะส่งข้อมูลชนิดใดกลับมา เช่น Integer, Long, Single เป็นต้น

ตัวแปร statements หมายถึง ชุดคำสั่งต่างๆ ในฟังก์ชัน

คำสั่ง Exit Function หมายถึง บังคับให้ออกจากฟังก์ชันนี้ทันที

คำสั่ง End Function หมายถึง จบ โปรซีเจอร์

ต่อไปนี้เป็นตัวอย่างฟังก์ชันที่ชื่อว่า Add มีหน้าที่สำหรับบวกตัวเลข ต้องการอาร์กิวเมนต์ 2 ตัว โดยคืนค่ากลับเป็นข้อมูลชนิด Integer

Option Explicit

Function Add(x As Integer, y As Integer) As Integer

Dim z As Integer 'สำหรับรับค่าที่ได้จากการบวก

z = x + y 'นำอาร์กิวเมนต์ที่เข้ามา 2 ตัวบวกกัน แล้วเก็บไว้ในตัวแปร z

Add = z 'ย้ายหอดค่าที่ได้ ไปยังฟังก์ชัน Add

End Function 'จบฟังก์ชันแล้วส่งค่ากลับ (ชนิดข้อมูลแบบ Integer) ไปยัง โปรซีเจอร์ ที่เรียกใช้งานฟังก์ชันนี้

Private Sub Form_Activate()

Print Add(10, 20) 'แสดงผลลัพธ์ที่ออกที่ฟอร์ม โดยเรียกใช้งานฟังก์ชัน Add และส่งอาร์กิวเมนต์เข้าไป 2 ตัว คือ 10 และ 20

End Sub

ขอให้คุณระวังในเรื่องของอาร์กิวเมนต์ที่ต้องส่งเข้าไปเป็นพิเศษ คุณต้องส่งอาร์กิวเมนต์ที่มีชนิดของข้อมูล เหมือนกับฟังก์ชันกำหนดไว้เท่านั้น ในกรณีนี้ผู้เขียนกำหนดไว้ว่า ฟังก์ชัน Add ต้องการอาร์กิวเมนต์ ที่มีชนิดของข้อมูลเป็น Integer คุณก็ต้องส่งเลขจำนวนเต็ม Integer เข้ามาเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง โพรเจกต์ประจำบทที่ 4

ฟังก์ชันสำหรับการหาค่า Max และ Min

แนวความคิด ตัวอย่าง โพรเจกต์นี้ เป็นการสร้างฟังก์ชันสำหรับหาค่าสูงสุด และหาค่าต่ำสุด โดยการรับอาร์กิวเมนต์เข้ามา 2 ตัว ซึ่งผู้เขียนกำหนดให้เป็นข้อมูลชนิด variant ทั้ง 2 ตัว คุณผู้อ่านสามารถเปลี่ยนแปลงให้เป็นข้อมูลชนิดอื่นๆ ได้

```
Function Max(ByVal Val1 As Variant, ByVal Val2 As Variant) As Variant
```

```
If Val1 > Val2 Then 'ถ้าค่าของตัวแรกมากกว่าตัวที่ 2 แล้ว
```

```
Max = Val1 'ให้เอาค่าของตัวแรกเป็นค่าสูงสุด แล้วเก็บไว้ที่ตัวแปร Max
```

```
Else
```

```
Max = Val2 'ถ้าไม่ใช่ให้เอาค่าของตัวที่ 2 เป็นค่าสูงสุด แล้วเก็บไว้ที่ตัวแปร Max
```

```
End If
```

```
End Function
```

```
Function Min(ByVal Val1 As Variant, ByVal Val2 As Variant) As Variant
```

```
If Val1 < Val2 Then 'ถ้าค่าของตัวแรกน้อยกว่าตัวที่ 2 แล้ว
```

```
Min = Val1 'ให้เอาค่าของตัวแรกเป็นค่าต่ำสุด แล้วเก็บไว้ที่ตัวแปร Min
```

```
Else
```

```
Min = Val2 'ถ้าไม่ใช่ให้เอาค่าของตัวที่ 2 เป็นค่าต่ำสุด แล้วเก็บไว้ที่ตัวแปร Min
```

```
End If
```

```
End Function
```

ฟังก์ชันสำหรับหาค่าเลขยกกำลัง

แนวความคิด เป็นฟังก์ชันที่ใช้คำนวณหาค่าเลขยกกำลัง ต้องการอาร์กิวเมนต์ 2 ตัว คือ เลขฐาน (base) และ เลขยกกำลัง (exp)

```
Function powern(ByVal base As Double, ByVal exp As Integer) As Double
```

```
Dim rtnVal As Double 'ตัวแปรรับค่าส่งกลับ
```

ตรวจสอบเลขฐานเท่ากับศูนย์หรือไม่

```
If base = 0 Then
```

```
MsgBox "Base can not be zero" 'ถ้าใช่แสดงข้อความ Base can not be zero แล้วจบการทำงาน
```

```
End
```

```
End If
```

'ตรวจสอบเลขยกกำลังว่าเป็นค่าติดลบหรือไม่

```
If exp < 0 Then
```

'ถ้าให้แสดงข้อความ Exponent must greater than or equal to zero แล้วจบการทำงาน

```
MsgBox "Exponent must greater than or equal to zero"
```

```
End
```

```
End If
```

'ตรวจสอบว่า เลขยกกำลังมีค่าเป็น 0 หรือไม่

```
If exp = 0 Then 'ถ้าใช่' ส่งค่า 1 กลับ
```

```
    rtnVal = 1
```

```
Else 'ถ้าไม่ใช่'
```

```
    rtnVal = 1 'กำหนดค่าส่งกลับ เริ่มต้นที่ 1
```

```
Do While exp > 0 'วนลูปจนกว่าเลขยกกำลังมีค่าไม่ติดลบ
```

```
    rtnVal = base * rtnVal 'เอาเลขฐานที่ส่งเข้ามาคูณกับค่าเริ่มต้นแล้วเก็บไว้ที่ตัวแปร rtnVal
```

```
    exp = exp - 1 'ลดค่าเลขยกกำลัง
```

```
Loop
```

```
End If
```

```
powerm = rtnVal 'ส่งค่าที่คำนวณได้
```

```
End Function
```

ฟังก์ชันสำหรับคำนวณค่า factorial

แนวความคิด เป็นฟังก์ชันที่ใช้คำนวณค่า factorial โดยมีขอบเขตในการใช้งานคือ สามารถหาค่าได้สูงสุด 12! เท่านั้น

```
Function factorial(ByVal n As Integer) As Double
```

```
Dim rtnVal As Long 'ตัวแปรรับค่าส่งกลับ
```

'ตรวจสอบค่า n ว่าติดลบหรือไม่

```
If n < 0 Then 'ถ้าใช่' แสดงข้อความ n must greater than or equal to zero แล้วออกจากฟังก์ชัน
```

```
MsgBox "n must greater than or equal to zero"
```

```
End
```

```
End If
```

'ตรวจสอบค่า n=0 หรือไม่

```
If n = 0 Then 'ถ้าใช่' คืนค่า 1
```

```
    rtnVal = 1
```

Else 'ถ้าไม่ใช่'

rtnVal = 1 'กำหนดค่าส่งกลับเท่ากับ 1

Do While n > 0 'ให้วนลูปจนกว่า n มากกว่า 0

rtnVal = rtnVal * n 'เอาค่า n คูณกับ rtnVal แล้วเก็บไว้ที่ตัวแปร rtnVal

n = n - 1 'ลดค่า n

Loop

End If

factorial = rtnVal 'ส่งค่ากลับ

End Function



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Link your Apps to the web

การเชื่อมโยงและพัฒนาแอปพลิเคชันสู่อินเทอร์เน็ต

การสร้างเมนูเชื่อมโยงกับอินเทอร์เน็ต

ในปัจจุบัน ผลิตภัณฑ์ของบริษัทซอฟต์แวร์โดยทั่วไป ไม่ว่าจะเป็นของไมโครซอฟท์ หรือของบริษัทอื่นๆ เรามักจะได้พบกับเมนูอยู่หัวข้อหนึ่ง เมื่อผู้ใช้คลิกเลือกเมนูดังกล่าวแล้ว ก็จะเชื่อมโยงไปยังเว็บไซต์ของผู้ผลิตซอฟต์แวร์นั้นๆ ตัวอย่างที่คุ้นเคย และชัดเจนที่สุดคือ ใน Visual Basic 6.0 ซึ่งมีเมนู Help/Microsoft on the Web หรือ ใน Word 2000 ซึ่งมีเมนู Help/Office on the Web ปรากฏอยู่ เราจะมาลองทำเมนูอย่างนี้ดูบ้าง โดยการใช้ Visual Basic จะทำให้แอปพลิเคชันของคุณ มีความเป็นมืออาชีพมากยิ่งขึ้น

หัวใจหลักของการทำงานอยู่ที่ การเรียกใช้คำสั่ง Start.exe ของ Windows 9x โดยปกติจะอยู่ในโฟลเดอร์ C:\windows\command ซึ่งมีหน้าที่สำหรับ เปิดแอปพลิเคชันต่างๆ ที่อยู่ในระบบ ให้ตรงกับชนิดของไฟล์ที่ผู้ใช้เรียก ด้วยคุณสมบัตินี้เอง เราจึงใช้ความสามารถของคำสั่ง Start.exe นี้ ในการเรียกใช้ browser ผ่านทาง Visual Basic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. การใช้งานโปรแกรม

งานวิจัยนี้จะสร้าง ระบบที่ใช้ในการวิเคราะห์ และ ตรวจสอบ การเสียภาษี ขึ้น มา โดยในส่วนของโปรแกรมนี้ จะทำการดึงข้อมูลจาก ฐานข้อมูลเข้ามาตรวจสอบ และ ทำการคำนวณการเสียภาษีว่ามีกรหลีกเลี่ยง ภาษีหรือไม่

โดยงานวิจัยนี้ จะมีส่วนในการ ตรวจสอบอยู่ ทั้งหมด 6 ส่วน มีดังนี้

1.1 Fiscal

เป็นส่วนที่ใช้ในการเลือกกลุ่มที่จะตรวจสอบ โดย จะมีเปอร์เซ็นต์ บอกว่า ในส่วนนั้นๆ จำเป็นที่จะต้องไปตรวจสอบ เพราะค่าที่ได้ มาน้อยกว่า เปอร์เซ็นต์ที่ควรจะเป็น

1.2 Plan

เป็นส่วนที่ใช้เพื่อ กำหนด กลุ่มที่ จะเข้าไปตรวจสอบ โดยผู้ตรวจสอบจะเป็นผู้เลือก พื้นที่ที่จะเข้าไป ตรวจสอบ ว่าต้องการตรวจในพื้นที่ไหน รายได้เท่าไร ประเภทของสินค้า แล้วจะได้ผลลัพธ์เป็น จำนวนผู้เสียภาษี ทั้งหมด ในพื้นที่ที่ต้องการ

1.3 Assign

เป็นการกำหนดรูปแบบการตรวจสอบ โดยจะสามารถเลือกได้ คล้ายๆ กับ Plan แต่สามารถที่จะเลือกการจัดลำดับการทำงานได้ โดย สามารถจัด จาก มากไปน้อย, น้อยไปมาก หรือ Random ข้อมูลก็ได้ โดยสามารถกำหนดจำนวนที่จะแสดงได้อีกด้วย

1.4 Clue

เป็นส่วนที่ใช้ในการ ตรวจสอบรายคน โดย จะสามารถที่จะเลือก ภาษีได้ และ จะมี สูตรและ Function ไว้ใช้ สำหรับตรวจสอบในแต่ละกรณี แล้วแต่ที่ผู้ใช้ต้องการที่จะตรวจสอบในลักษณะไหน มี Function และ สูตร ให้สามารถเลือกใช้ได้

1.5 แบบแสดงรายการภาษีมูลค่าเพิ่ม(FORM VAT)

เป็นส่วนที่ใช้ในการแสดงข้อมูลออกหน้าจอ โดยจะแสดงข้อมูลของผู้เสียภาษีมูลค่าเพิ่ม โดยจะทำการใส่ เลขผู้เสียภาษีอากร(TIN) แล้ว โปรแกรมจะไปดึงข้อมูลใน ฐานข้อมูลมาแสดงใน แบบแสดงรายการ โดยจะแสดงให้เห็นรายละเอียด ทั้งหมดของ ผู้เสียภาษีนั่น

1.6 แบบแสดงรายการภาษีเงินได้นิติบุคคล (FORM CIT)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นส่วนที่ใช้ในการแสดงข้อมูลออกหน้าจอ เหมือนกับ แบบแสดงรายการภาษีมูลค่าเพิ่ม แต่จะแสดงข้อมูลของผู้เสียภาษีเงินได้นิติบุคคล โดยจะทำการใส่ เลขผู้เสียภาษีอากร(TIN) แล้ว โปรแกรมจะไปดึงข้อมูลใน ฐานข้อมูลมาแสดงใน แบบแสดงรายการโดยจะแสดงให้เห็นรายละเอียด ทั้งหมดของ ผู้เสียภาษีคน นั้น

2. วิธีการดำเนินงาน

วิทยานิพนธ์ฉบับนี้เป็นแนวทางในการที่จะตรวจสอบและค้นหาผู้ที่เสียภาษีที่อยู่ในขอบข่ายที่น่าสงสัยว่าเป็นผู้ที่หลีกเลี่ยงการเสียภาษี แล้วทางกรมสรรพากรจะทำการจัดส่งเจ้าหน้าที่เข้าไปทำการตรวจสอบข้อเท็จจริงแก่บุคคลนั้น โดยระบบการตรวจสอบของผู้เสียภาษีนี้อาจทำการตรวจสอบ ภาษีมูลค่าเพิ่มและ ภาษีเงินได้นิติบุคคล มาเป็นตัวพิจารณาและใช้สูตรกฎเกณฑ์ต่าง ๆ มาเป็นเงื่อนไขในการตรวจสอบด้วย

3. การใช้งาน

การเข้าใช้งานนั้นจะเริ่มโดย หน้าแรกจะเป็นการใส่ รหัส เข้า เสร็จแล้วก็ เข้าในส่วนของ Menu จะมี Menu ให้เลือก 6 Menu ดังนี้

3.1 ส่วนของ Clue

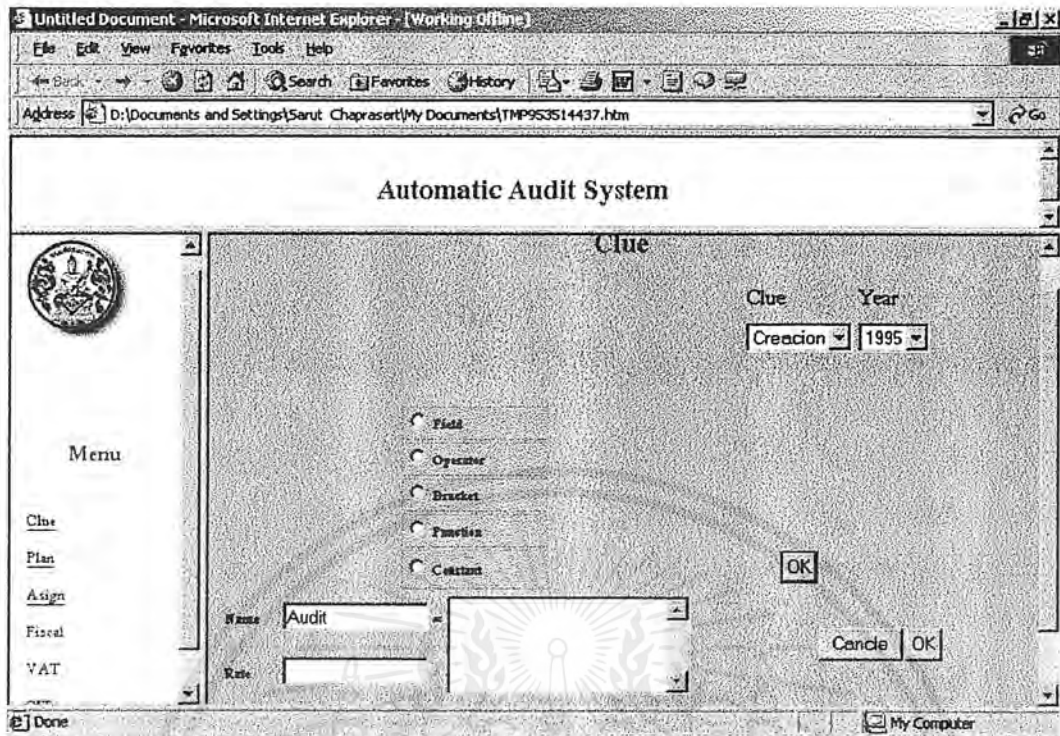
ในส่วนแรกเป็นการแสดง โปรแกรมในส่วนของ Clue เป็นการตรวจสอบรายชื่อผู้เสียภาษี โดยทำการ คีย์ข้อมูลของผู้เสียภาษีลงไปแล้ว โปรแกรมจะทำการตรวจสอบว่าผู้เสียภาษีคนนั้นอยู่ในขอบข่ายที่ต้องเข้าไป ทำการตรวจสอบหรือไม่ โดยจะมีข้อความแสดงบอกให้รู้ชั้นตอนมีดังนี้

3.1.1 ใส่ชื่อผู้เสียภาษีลงไป ในช่อง Name แล้วกดปุ่ม OK ที่ด้านบน

3.1.2 หลังจากที่ได้ชื่อ ไปแล้วจะแสดงข้อความที่ในส่วนของ Textbox ทางด้านขวามือ แล้วทำการกด เลือกตามรายด้านบน จะประกอบด้วย Field , Operator , Bracket , Function , Constant ในส่วนนี้จะเป็นรายละเอียดของการเลือกตรวจสอบว่าจะตรวจสอบภาษีประเภทไหน ในการเลือกแต่ละรายการเมื่อเสร็จแล้วต้อง กดที่ปุ่ม OK ด้านบน แล้วรายละเอียดต่าง ๆ จึงจะแสดงให้เห็น

3.1.3 หลังจากที่ได้เลือกใส่รายการต่าง ๆ และเงื่อนไขแล้วให้คลิกที่ปุ่ม OK โปรแกรมจะทำการดึงรายละเอียดของข้อมูลค่าที่ใส่ไว้มาแสดงใน Text box

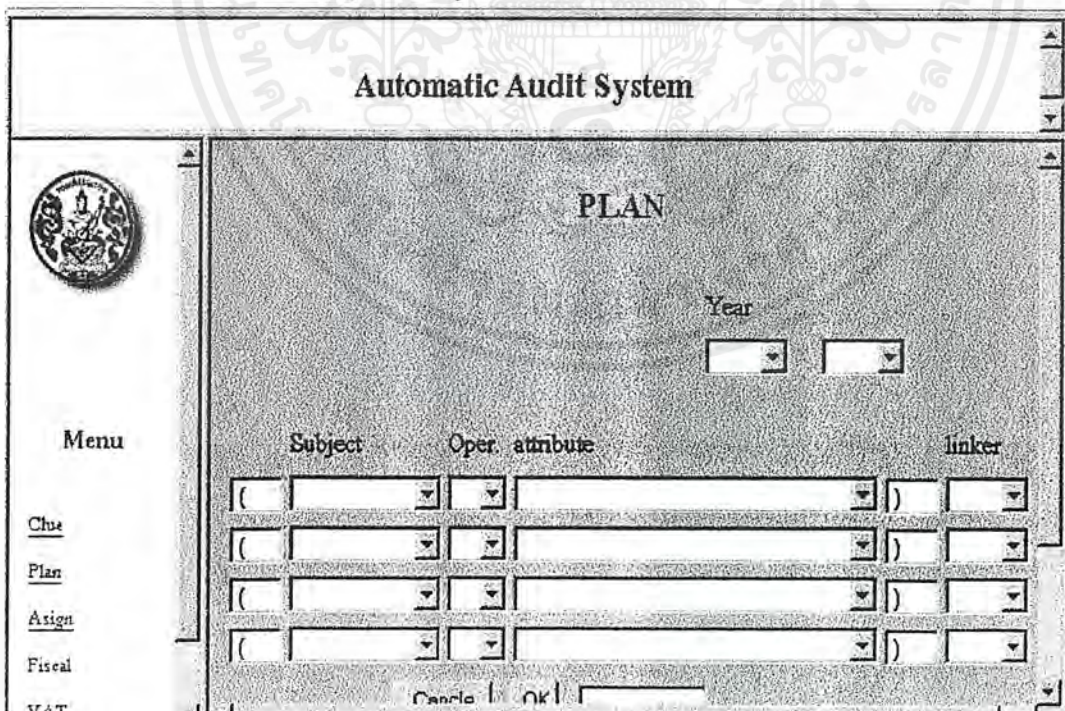
3.1.4 หลังจากที่ได้ใส่เงื่อนไขต่าง ๆ ลงไปแล้ว ให้ทำการคลิกที่ปุ่ม OK ในส่วนของโปรแกรมจะทำการ แสดงข้อความขึ้นมาว่าจากข้อมูลและเงื่อนไขของผู้เสียภาษีทั้งหมดที่ได้ลงไปนั้น ว่าผู้เสียภาษีบุคคลนี้อยู่ใน ขอบข่ายที่ต้องเข้าไปตรวจสอบหรือไม่ มีรูปแบบดังรูปที่ 3.1



รูปที่ 3.1 แสดงโปรแกรมในส่วนของ Clue

3.2 ส่วนของ Plan

3.2.1 ส่วนของ PLAN มีรูปแบบดังรูป



รูปที่ 3.2 แสดงหน้าต่างของ plan

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 จะมีส่วนต่างๆดังนี้

3.2.2.1 ส่วนกำหนดปีในการ ตรวจสอบ

3.2.2.2 ส่วนกำหนด รายละเอียด ในการตรวจสอบ

3.2.2.3 ปุ่มยกเลิกค่าที่กำหนดไป

3.2.2.4 ปุ่ม OK จะทำการตรวจสอบข้อมูลใน DATABASE แล้ว นำค่ามาแสดงที่ ช่องว่าง

3.2.2.5 Text box แสดงจำนวนตารางทั้งหมด

3.2.3 จะทำการ Link ข้อมูลกับ DATABASE เพื่อตรวจสอบสิ่งที่กำหนดมานั้น มีที่ Table

3.2.4 สามารถที่จะมีการ AND , OR และ () เพื่อให้สามารถกำหนดรายละเอียดที่ต้องการได้ครบและตามที่ต้องการ

3.2.5 เวลาแสดง Block ที่ใช้ในการกำหนดต่างๆ จะแสดงที่ละบรรทัด แล้วถ้าทำการเลือก AND หรือ OR จะแสดง Block บรรทัดต่อไปขึ้นมา

3.2.6 ถ้าเลือก END จะ Show ปุ่มต่างๆ เพื่อให้เลือกว่า OK เพื่อหาตารางเลข หรือ เลือก ยกเลิกเพื่อที่จะกำหนดค่าใหม่

3.2.7 ถ้าเลือก OK ก็จะแสดงค่าของ Recode ใน ฐานข้อมูลออกมา

3.3 ส่วนของ Assign

3.3.1 ส่วนของ Assign มีรูปแบบดังรูป

รูปที่ 3.3 แสดงการหน้าต่างของ Assign

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 รายละเอียดของส่วนต่างๆ จะมีส่วนต่างๆดังนี้

3.3.2.1 ส่วนกำหนดว่าจะให้จัดลำดับความสำคัญตัวไหนและ จัดลำดับยังไง มากไปน้อย , น้อยไปมาก หรือ Random

3.3.2.2 ส่วนกำหนด รายละเอียด ในการตรวจสอบ โดยจะมีลำดับ และ TIN เป็นตัว fix อยู่ ซึ่งจะไม่สามารถเปลี่ยนได้

3.3.2.3 ปุ่มยกเลิกค่าที่กำหนดไป

3.3.2.4 ปุ่ม OK จะทำการตรวจสอบข้อมูลใน DATABASE แล้ว นำค่ามาแสดงที่ ช่องว่าง

3.3.2.5 Text box ให้ใส่ ชื่อ ผู้เรียง

3.3.2.6 Text box ใส่กลุ่มผู้เรียง

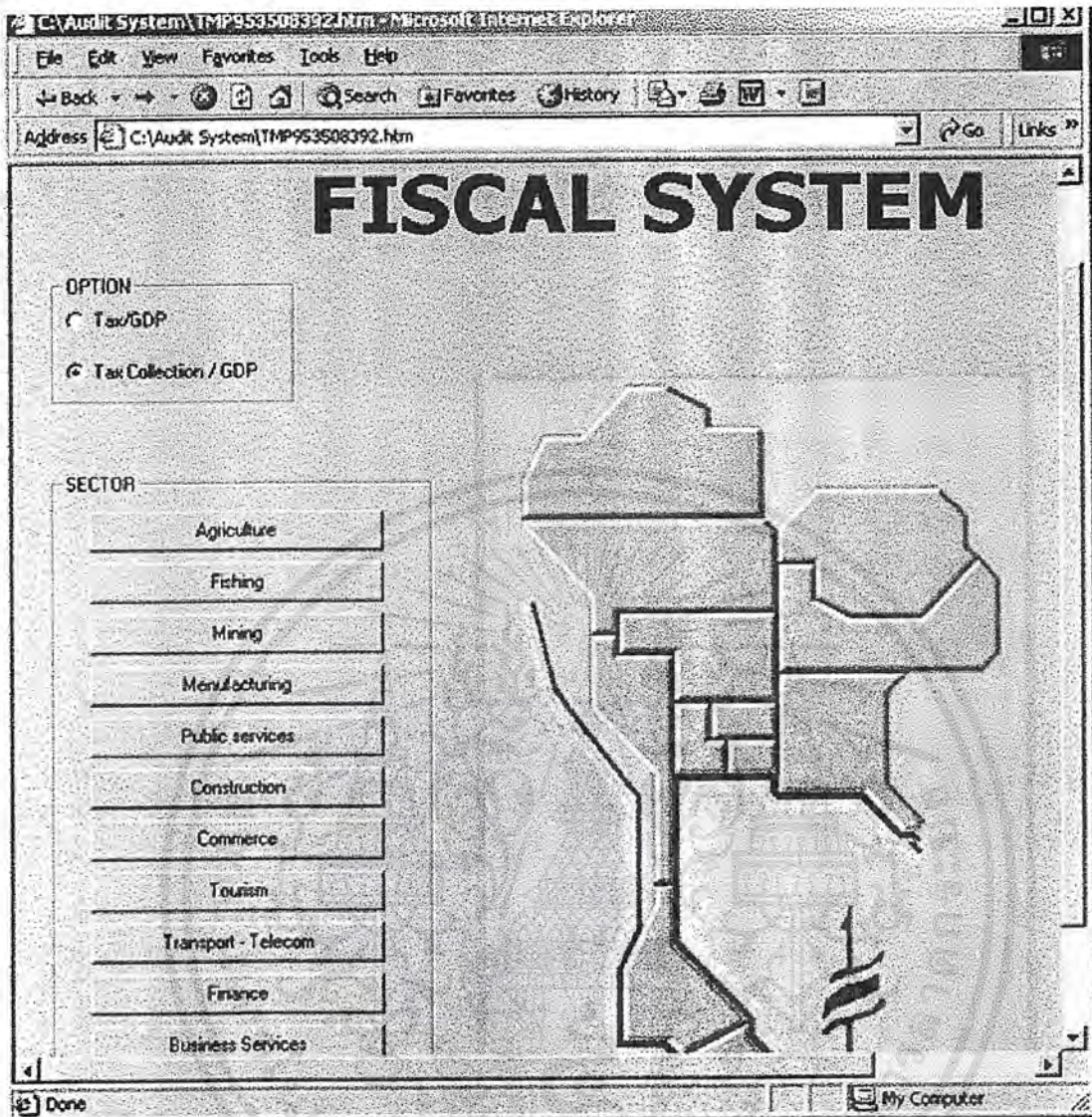
3.3.3 การกำหนดการแสดงผล สามารถกำหนดได้ว่าจะ แสดง ทั้งหมดก็ ตาราง

3.3.4 ส่วน Link จะทำการ Link ข้อมูลกับ DATABASE เพื่อ Show ข้อมูลทั้งหมด ตามจำนวน Table ที่ กำหนด

3.3.5 การเลือกค่า ถ้าเลือก OK จะ Show ข้อมูลใน DATABASE ตามจำนวน Table ที่ กำหนด

3.4 ส่วนของ Fiscal

เป็น Module สำคัญส่วนหนึ่งในระบบ Audit เพราะ Fiscal System เป็นระบบที่ทำงาน โคนทำหน้าที่ ตรวจสอบ และคัดเลือก line ในการตรวจสอบ หรือเลือก line ในการสอบสวนกลุ่มผู้เสียภาษี เขตใด เขตหนึ่ง , ภาคธุรกิจใด ธุรกิจหนึ่ง โดยอาศัยวิธีการเปรียบเทียบ stat ฐานภาษีของกลุ่มนั้น กลุ่มธุรกิจนั้นๆ เป็นเปอร์เซ็นต์ แล้วคัดเลือก ผู้เสียภาษีในเขตใดเขตหนึ่ง ซึ่งมีเปอร์เซ็นต์ของ ฐานภาษี , ภาษีที่ จะต้องชำระ ต่อ ค่า GDP ของแต่ละพื้นที่



รูปที่ 3.4 แสดงหน้าจอหลักของ ระบบ

3.4.1 การทำงานของระบบ

Interface ของระบบ ถูกออกแบบบน Web Base Technology โดยใช้ Internet Explorer 4.0 หรือรุ่น
หลังจากนี้ในการ Browse

จากหน้าจอหลักจะแบ่ง Interface เป็นส่วนใหญ่ สองส่วนคือ

- 3.4.1.1 ส่วน Input สำหรับให้ User กำหนดค่า Option และเลือก Sector ในการตรวจสอบ
- 3.4.1.2 Tax Option เลือกคำนวณจาก Tax Base หรือ Tax Collection
- 3.4.1.3 SECTOR เลือกภาคธุรกิจในการเข้าถึงกลุ่มธุรกิจแต่ละกลุ่ม
- 3.4.1.4 MAP เลือกพื้นที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2 ส่วน Output สำหรับ แสดงค่าผลการคำนวณเพื่อใช้ในการพิจารณาตรวจสอบต่อไป

3.4.2.1 MAP แสดงค่าการคำนวณในแต่ละพื้นที่ (ค่าที่แสดงมีสองสี สีแดง และสีเขียว

3.4.2.2 text แสดง Status ว่าคำนวณจากอะไร


ข้อมูลที่น่าทดสอบเป็นข้อมูลที่สร้างขึ้นมาเอง เปิด Fiscal System Page จากหน้าจอเมนูหลัก เลือก Option ที่ใช้ในการคำนวณ Tax / GDP คลิกปุ่ม Fishing เพื่อกำหนด กลุ่มธุรกิจในการตรวจสอบ จากผลการทดลอง จะเห็นว่าค่าที่แสดงบน Map จะมีสีต่างกัน คือ สีเขียว และ สีแดง ซึ่ง สีเขียว หมายถึง ไม่มีความเสี่ยง สีแดงหมายถึง พื้นที่นั้น(กลุ่มธุรกิจ) ที่เลือก มีความเสี่ยง

3.5 Form VAT

เป็นการดึงค่ามาจาก ฐานข้อมูลเพียงอย่างเดียว มี form ดังรูป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

 Value Added Tax		Nr 30		For official use	
				Document Control nr	From received
A TIN		Branch		B Filing VAT return with branches	
VAT taxpayer is name		Join submission at		Separate declaration	
Name of business		HQ		HQ	
Biz address		Branch Nr.		Branch Nr.	
		C nor on time		Add out time	
		Nra		for year	
		Jan		Apr	
		Feb		May	
		Mar		Jun	
		Jul		Oct	
		Aug		Nov	
		Sep		Dec	
Phone		Post code			
For manual receipts		POS receipts			
Book Nr		Nr		Pay amount	
Amount		(Cashier)		Payment date	
Signed				Instalment	
Date					
D Tax computation					
Output Tax		+ sales			
1 Sales amount II add		- purch			
2 (-) 0% rate sales					
3 (-) Exempted sales					
4 Taxable sales	(1-2-3)				
5 Month Output tax					
Input Tax		+ purch			
6 Purch. amount II add		- sales			
7 Input VAT					
Value Added Tax					
8 Tax to be paid					
9 Tax over paid					
10 tax credits					
11 Payable tax					
12 Overpaid tax					
(In letters)					
Late or additional filing					
13 Surcharges					
14 Penalties					
15 Total Tax+ surcharges+penalties					
16 Total overpaid taxes after deductions of surcharge + penalties					
(In letters)					
E Refund request			F I/we hereby certify that information declared are correct and in case of ...		
Forwarding credit			Signature		
			Date		

รูปที่ 3.5 แสดง Form VAT

6.4 Form CIT

เป็นการดึงค่ามาจากฐานข้อมูลเพียงอย่างเดียว มี form ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Under section 68 and 69 of the Revenue Code			
TIN		pre-examination	Review by
Name		Accounting period	
Address		Since	
Village	Road	Unit	
District	Province	1. Thai 2. Abroad 3. Overseas	
Telephone		Post code	
For manual procedure		Cash register reception	
Receipt volume	Serial	Year	A
Amount		Pay amount	B
Signed	Receiver	Payment date	C
Date			
The address above is the same as last year		New address	D
Date registered as Juristic Co.	Co. Registration Nr.	Shares held by foreigners	
Province	45	Text30 % of Nr shares	
Country	Nr of branches	Q/H	
Text31 Co or Part. establ under Thai law	Text33 Others under foreign law	Text35 Foreign Gov. Business	Text38 Inter. carrier
Text32 Co or Part. establ under foreign law	Text34 Securities exchange of Th.	Text36 Joint venture	
		Text37 SIBF business	
Business undertaken (state in order according of importance, sales or income)			
1	2	3	
Text40	Text42	Text43	
In case of being a branch of foreign company			
1 Date commencing operation in Th.			
Head office address			
2 Name of representative		TIN	Text47
Office, number, road, valley and province, district		Post code	Text51
Amphur Code		Text50	
Income Tax to be Paid			
1	Text52 Total CIT payment	Text53 Paid in excess (from 2 (B) page 2)	Text54
In letters		Text55	
Certification of Manager, partner or Director			
I/We have examined the particulars in his/our return and the attached documents.....			
Signed		Signed	
Title		Title	
Name of authorized auditor		License No.	Audit company TIN

รูปที่ 3.6 แสดง Form CIT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้