โปรแกรมเมอร์มือใหม่ หัดเขียนโปรแกรม

Microsoft Visual Basic 6.0 Enterprise Edition

แสงเทียนแห่งปัญญา

สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย

โปรแกรมเมอร์มือใหม่ หัดเขียนโปรแกรม

Microsoft Visual Basic 6.0 Enterprise Edition

ธีระศักดิ์ สุโชตินันท์
ประยุทธ อินแบน

สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย
2549

380.-

ธีระศักดิ์ สุโชตินันท์

ล้วงลึกโปรแกรมเมอร์มือใหม่ หัดเขียนโปรแกรม Microsoft Visual Basic 6.0 Enterprise edition / ธีระศักดิ์ สุโชตินันท์ , ประยุทธ อินแบน

1. การเขียนโปรแกรมคอมพิวเตอร์.
2. ไมโครซอฟต์วิชวลเบสิก.
3. ประยุทธ อินแบน.

005.1

ISBN 978-974-99-9581-5

สพจ. 1037

สรรคุณค่าวิชาการ สู่สังคม
http://www.ChulaPress.com

จัดจำหน่ายโดย ศูนย์หนังสือแห่งจุฬาลงกรณ์มหาวิทยาลัย
ถนนพญาไท เขตปทุมวัน กรุงเทพมหานคร 10330

ศาลาพระเกี้ยว โทร. 0-2218-7000-3 โทรสาร 0-2255-4441

สยามสแควร์ โทร. 0-2218-9869 โทรสาร 0-2254-9495

ม.บูรพา จ.ชลบุรี โทร. 0-3839-4855-9 โทรสาร 0-3839-3239

ม.นเรศวร จ.พิษณุโลก โทร. 0-5526-0162-4 โทรสาร 0-5526-0165

ม.เทคโนโลยีสุรนารี จ.นครราชสีมา โทร. 0-4421-6131-4 โทรสาร 0-4421-6135

CALL CENTER โทร. 0-2255-4433 http://www.chulabook.com
ร้านค้าติดต่อทีมขายส่ง สยามสแควร์ ชั้น 14 โทร. 0-2218-9889-90 โทรสาร 0-2254-9495

พิมพ์ที่ ห้างหุ้นส่วนจำกัด สามลดา โทร. 0-2895-1500, 0-7100-6500
ออกแบบปก ธีระศักดิ์ สุโชตินันท์

คำนำ

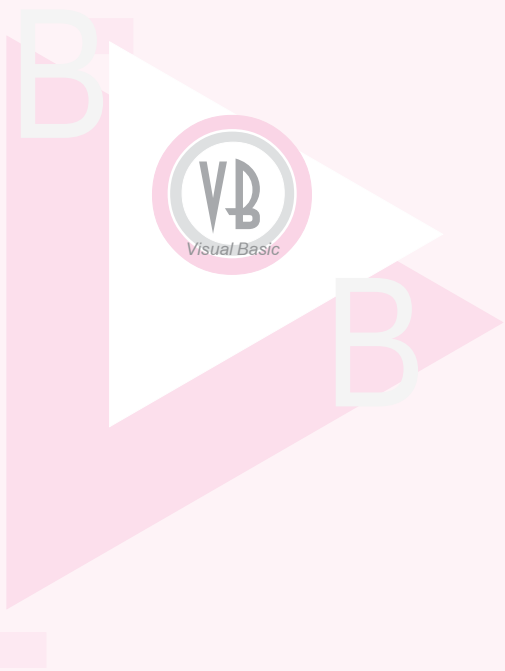
เมื่อท่านหยิบหนังสือเล่มนี้ขึ้นมาอ่านแล้ว ท่านอาจจะพบว่าทำไมเนื้อหาถึงมีมากมายขนาดนี้ จะอ่านซักกี่วันถึงจะหมด อ่านแล้วจะใช้งานได้จริงหรือไม่ก็ยังไม่รู้ ซึ่งไม่เป็นที่น่าแปลกใจเลย เพราะผู้เขียนเองก็เคยเป็นดังเช่นท่าน ตอนที่เรียนอยู่ในมหาวิทยาลัย การที่จะเขียนโปรแกรมเพื่อนำมาใช้งานเองสักโปรแกรม ทำไมมันช่างแสนยากเย็นเช่นนี้ และจะใช้ภาษาอะไรเป็นตัวเขียนโปรแกรมดีล่ะ แต่ที่กล่าวมาก็ยังไม่เท่าไร ปัญหากับอยู่ตรงที่ว่า จะหาหนังสือที่อ่านแล้วทำให้เราทำงานได้ และรู้จริง ไม่เน้นด้านทฤษฎี หรืออธิบายแต่ทางหลักวิชาการเพียงอย่างเดียว หนังสือที่ท่านหยิบอ่านเล่มนี้สามารถหลุดพ้นจากพันธนาการที่กล่าวมาข้างต้น เพราะเน้นทุกด้านให้กับทุกท่านที่คิดจะเป็นโปรแกรมเมอร์ หรือเริ่มต้นการเขียนโปรแกรมใช้งานเอง โดยจะอธิบายด้วยถ้อยคำ และภาษาที่เข้าใจง่าย โดยในหนังสือเล่มนี้ ผู้เขียนจะแบ่งเนื้อหาออกเป็น 3 ส่วนด้วยกัน ดังนี้

ส่วนที่ 1 ***"เปิดประตู โปรแกรมวิชวลเบสิก"*** เป็นเรื่องของทฤษฎี ซึ่งอธิบายรูปแบบ และวิธีการต่าง ๆ ที่ใช้ในโปรแกรมวิชวลเบสิกขั้นพื้นฐาน แบบนำไปใช้ในงานจริงอย่างไม่มีปัญหา

ส่วนที่ 2 ***"ทำไปลองไป มือใหม่หัดเขียนโปรแกรม"*** เป็นการทดลองเขียนโปรแกรมทีละขั้นตอน เพื่อนำไปใช้งานจริง ซึ่งเน้นทั้งทฤษฎี และปฏิบัติควบคู่กันไปในขณะเขียนโปรแกรม

ส่วนที่ 3 ***"กลเม็ดเคล็ดลับกับโค้ดเด็ด"*** เป็นการเน้นทำ WorkShop และศึกษาโค้ดการใช้งาน ให้กับผู้เขียนโปรแกรม เพื่อนำไปประยุกต์ใช้กับงาน

ธีระศักดิ์ สุโชตินันท์
ประยุทธ อินแบน


Visual Basic

สารบัญ

PART 1
"เปิดประตู โปรแกรมวิชวลเบสิก"

PART 2

"ทำไปลองไป มือใหม่หัดเขียนโปรแกรม"

PART 3

"กลเม็ดเคล็ดลับกับโค้ดเด็ด"

“ขอขอบคุณที่ท่านเลือกและให้ความไว้วางใจ ในผลงานของ
สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย”

ประวัติศาสตร์...

Augusta Ada Byron
โปรแกรมเมอร์คนแรกของโลก

ออกุสต้า

นับได้ว่าเป็นโปรแกรมเมอร์คนแรกของโลก ซึ่งเป็นชาวอังกฤษ ในปี ค.ศ 1842 (พ.ศ 2385) ได้เขียนขั้นตอนคำสั่งในการสั่งงานเครื่องวิเคราะห์แบบเบจ (Analytical Engine) และยังพบหลักการทำงานแบบวนซ้ำ (Loop) อีกด้วย

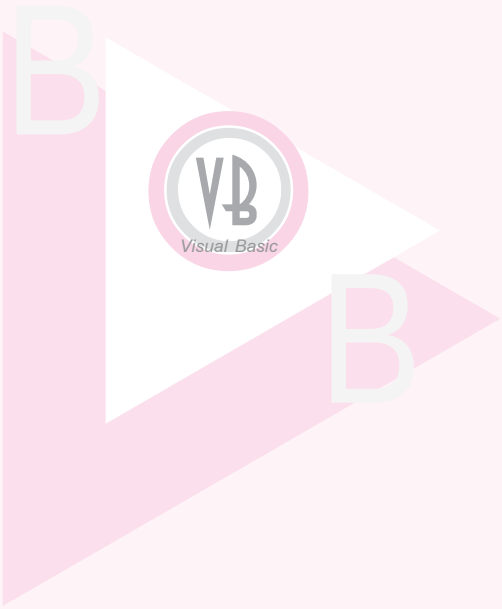
โปรแกรมเมอร์มือใหม่ หัดเขียนโปรแกรม

Microsoft
Visual Basic 6.0
Enterprise Edition

PART 1

"เปิดประตู โปรแกรมวิชวลเบสิก"

เป็นเรื่องของทฤษฎีซึ่งอธิบาย รูปแบบ และวิธีการต่าง ๆ ที่ใช้ในโปรแกรมวิชวลเบสิกขั้นพื้นฐาน แบบนำไปใช้ในงานจริงอย่างไม่มีปัญหา


Visual Basic

บทที่ 1

เริ่มแรกก่อนใช้งานโปรแกรมวิชวลเบสิก

ในการพัฒนาหรือเขียนโปรแกรมขึ้นมา โปรแกรมเมอร์จะต้องมีความรู้ความเข้าใจ ในตัวภาษา ที่จะนำมาใช้ในการพัฒนาเป็นอย่างมาก ไม่ว่าจะพัฒนาด้วยภาษาใดก็ตาม เช่น ภาษาซี ปาสกาล เป็นต้น และจะต้องเขียนโค้ดที่มีความสัมพันธ์กัน ตั้งแต่บรรทัดแรกจนถึงบรรทัดสุดท้าย อีกทั้งยังต้องออกแบบรูปร่างหน้าตาของแอปพลิเคชันให้ตรงกับความต้องการของผู้ใช้งาน อีกด้วย ซึ่งถ้าถูกใจผู้ใช้ก็ดีไป แต่ถ้าต้องมีอันจำเป็นในการแก้ไข ทุกสิ่งทุกอย่างที่ได้ทำมาก็ต้องลงมือทำใหม่ทั้งหมด ทำให้โปรแกรมเมอร์เสียเวลาในการพัฒนาแอปพลิเคชันเป็นอย่างยิ่ง

ต่อมาไมโครซอฟต์ได้นำเสนอ รูปแบบในการเขียนแอปพลิเคชันด้วยการออกโปรแกรมวิชวลเบสิก ถึงแม้ว่าเวอร์ชันแรก จะถูกโปรแกรมเมอร์มองว่าเป็นเวอร์ชันทดลอง แต่ก็ได้สร้างความแปลกใหม่ ในการเขียนโปรแกรมเป็นอย่างยิ่ง ความยุ่งยากซับซ้อนถูกซ่อนไว้เบื้องหลัง มีแต่ความสะดวกสบายไว้เบื้องหน้า ซึ่งเตรียมไว้ให้โปรแกรมเมอร์ เนื่องจากความคิดของการเขียนโปรแกรมเปลี่ยนไปอย่างสิ้นเชิง ด้วยความสามารถของตัวภาษาวิชวลเบสิกเองก็มีมากขึ้น เพราะความก้าวหน้าของเทคโนโลยีในยุคปัจจุบัน ทำให้ไมโครซอฟต์เพิ่มเติมรูปแบบต่าง ๆ เข้าไปมากมาย

จนกระทั่งโปรแกรมวิชวลเบสิกเป็นเครื่องมือในการพัฒนาโปรแกรมที่ไร้เทียมทาน เพราะเทคโนโลยีใหม่ล้วนแล้วแต่มาจากไมโครซอฟต์แทบทั้งสิ้น เช่น ความสามารถในการสร้างแอปพลิเคชัน DHTML ซึ่งใช้ปฏิบัติการบนเว็บเพจ รวมถึงการผนวกเทคโนโลยี ActiveX เข้ากับตัวคอนโทรล ทำให้สามารถเชื่อมโยงเข้ากับเครื่องมือที่สนับสนุนเทคโนโลยีนี้ได้อีกด้วย

โปรแกรมวิชวลเบสิกได้จัดเตรียมเครื่องมือต่าง ๆ ที่เรียกว่า ***คอนโทรล*** ไว้คอยอำนวยความสะดวกให้แก่โปรแกรมเมอร์มากมาย ซึ่งจะต้องศึกษาและทำความเข้าใจกับตัวคอนโทรลให้มากที่สุด ตัวคอนโทรลเหล่านี้ จะอยู่เบื้องหลังที่ทำให้โปรแกรมวิชวลเบสิกประสบผลสำเร็จ เพราะจะช่วยลดขั้นตอนในการพัฒนาไปได้มากทีเดียว

การเขียนโปรแกรมด้วยภาษาวิชวลเบสิก จะเป็นไปในลักษณะการนำคอนโทรลชนิดต่าง ๆ มาวาดเพื่อออกแบบหน้าตาแอปพลิเคชัน ที่เรียกว่า กราฟิกยูสเซอร์อินเตอร์เฟส ซึ่งสามารถที่จะออกแบบอินเตอร์เฟสได้อย่างอิสระ ให้ตรงกับจุดประสงค์และการนำไปใช้งาน แล้วถึงเริ่มเขียนโค้ด เพื่อตอบสนองการกระทำของผู้ใช้งาน ในวิชวลเบสิกเรียกว่า ***เหตุการณ์*** ซึ่งถือว่าเป็นหลักของการเขียนโปรแกรมที่เรียกว่าเขียนโปรแกรมเพื่อตอบสนองเหตุการณ์ที่เกิดขึ้น

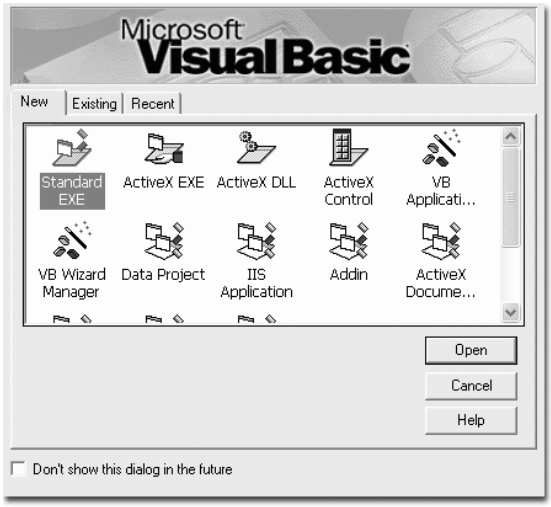
สิ่งที่นำไปใช้ร่วมกันเพื่อเป็นแอปพลิเคชันหนึ่ง ๆ เช่น แถบเมนูบาร์ ไดอะล็อกบล็อก จะถูกมองเสมือนว่าเป็นวัตถุชิ้นหนึ่ง ที่เรียกว่า ***อ็อบเจกต์*** ทุกสิ่งทุกอย่างในแอปพลิเคชัน วิชวลเบสิกจะมองเป็นออบเจ็กต์ที่สามารถควบคุมพฤติกรรมโดยตรงต่ออ็อบเจกต์นั้นได้ด้วยการเขียนโค้ด หรือสามารถเปลี่ยนแปลงคุณสมบัติประจำตัวของอ็อบเจกต์นั้นได้โดยตรง ตัวคอนโทรลก็ถูกมองว่าเป็นอ็อบเจกต์เช่นกัน ในทุก ๆ อ็อบเจกต์ จะมีคุณสมบัติและเมท็อดประจำตัวเสมอ ในแต่ละอ็อบเจกต์อาจจะมีคุณสมบัติและเมท็อดที่เหมือน หรือต่างกันก็ได้ ขึ้นอยู่กับชนิดของอ็อบเจกต์นั้น ๆ

ในการพัฒนาแอปพลิเคชันด้วยโปรแกรมวิชวลเบสิก ***การเขียนโค้ดจะถูกแบ่งออกเป็นส่วน ๆ ที่เรียกว่า โพรซีเยอร์ หรือโปรแกรมย่อย*** แต่ละโพรซีเยอร์ จะประกอบไปด้วยโค้ดที่พิมพ์เข้าไป แล้วทำให้คอนโทรล หรืออ็อบเจกต์ ตอบสนองการกระทำของผู้ใช้งานโดยสมบูรณ์ในตัวมันเอง ซึ่งเรียกว่า ***การเขียนโปรแกรมเชิงวัตถุ*** แต่ตัวภาษาวิชวลเบสิกยังไม่ถือว่า เป็นโปรแกรมเชิงวัตถุอย่างแท้จริง เนื่องจากข้อจำกัดหลาย ๆ อย่างที่โปรแกรมวิชวลเบสิกไม่สามารถทำได้เหมือนกับภาษาซีบวกบวก (C++) การเขียนโปรแกรมเชิงวัตถุมีข้อดี คือ ตัวโค้ดจะถูกแบ่งออกเป็นส่วน ๆ ทำให้ง่ายต่อการตรวจสอบ และดักจับข้อผิดพลาด ซึ่งการแก้ไขดังกล่าวนี้ ไม่ได้ไปกระทบกับโค้ดส่วนอื่น ๆ ในตัวเลย ทำให้โปรแกรมเมอร์สามารถพัฒนาแอปพลิเคชันออกมาได้อย่างสมบูรณ์มากที่สุด โดยที่ไม่ต้องเสียเวลามากมายดังเช่นในอดีต

:: เริ่มต้นกับโปรแกรมวิชวลเบสิก

สิ่งแรกที่จะได้พบเมื่อเข้าสู่โปรแกรมวิชวลเบสิก คือ ไดอะล็อกบล็อก เพื่อให้เลือกพัฒนาชนิดของแอปพลิเคชัน ซึ่งมีมากมายหลายชนิดด้วยกัน เช่น DynamicHTML เป็นแอปพลิเคชันที่ใช้ปฏิบัติการบนเว็บเพจ โดยสร้างเว็บเพจแบบ DynamicHTML ให้ตรงตามมาตรฐานขององค์กรดับเบิลยูสาม ซึ่งสามารถตรวจสอบได้จาก www.w3.org หรือ ชนิด ActiveX.dll ซึ่งเป็นโพรเจกต์ที่ทำงานในลักษณะ In-Process จัดเก็บอยู่ในลักษณะไฟล์ไลบรารี (*.dll) เป็นต้น

ในเบื้องต้น แนะนำให้พัฒนาแอปพลิเคชันชนิด *Standard.EXE* ก่อน เพราะเป็นแอปพลิเคชันที่ใช้งานทั่ว ๆ ไป เมื่อคอมไพล์โพรเจกต์ จะได้แอพพลิเคชันที่มีนามสกุล *.exe เมื่อเลือกชนิดของแอปพลิเคชันแล้ว จะเข้าสู่สภาพแวดล้อมของโปรแกรมวิชวลเบสิก ซึ่งไมโครซอฟต์เรียกว่า *Integrated Development Environment - IDE*



รูปที่ 1-1 แสดงไดอะล็อกบล็อกของแอปพลิเคชัน ที่ให้โปรแกรมเมอร์เลือกพัฒนา

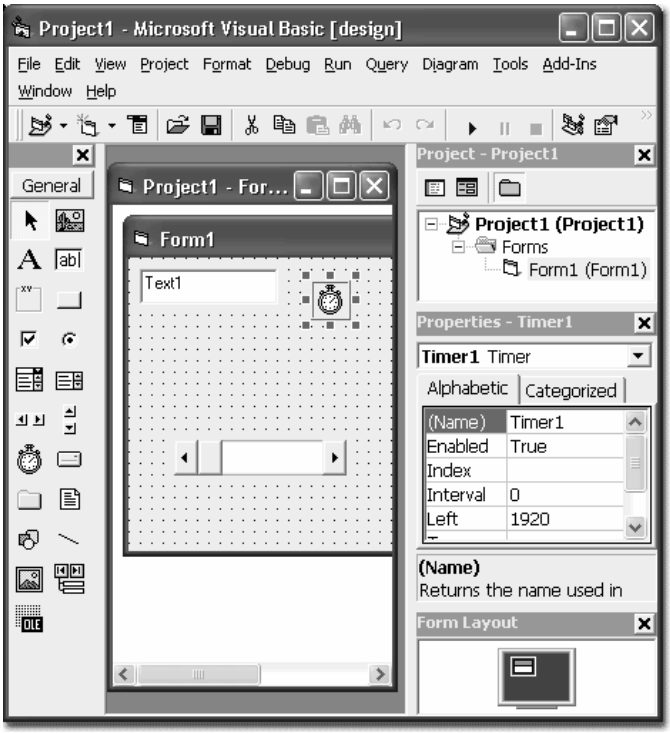
:: ทำความเข้าใจกับสภาพแวดล้อมโปรแกรมวิชวลเบสิก

สภาพแวดล้อมของโปรแกรมวิชวลเบสิก คือ หน้าต่างคุณสมบัติ กลุ่มของเมนู ทูลบาร์ หน้าต่างชิ้นงาน เป็นต้น ที่ประกอบกันขึ้นมาเป็นสภาพแวดล้อมของโปรแกรมวิชวลเบสิก

ในแต่ละส่วนของสภาพแวดล้อมของโปรแกรมวิชวลเบสิก จะมีหน้าที่แตกต่างกันไป ซึ่งในระหว่างการพัฒนาแอปพลิเคชัน จะต้องใช้งาน เช่น เมนูบาร์จะมีคำสั่งต่าง ๆ ที่ครอบคลุมการทำงานทั้งหมดในการพัฒนา แถบทูลบาร์จะประกอบไปด้วยปุ่มต่าง ๆ ที่แทนคำสั่งในเมนูที่ใช้งานบ่อย ๆ เช่น การเปิดปิดโพรเจกต์ การบันทึกโพรเจกต์ เป็นต้น หัวข้อนี้จะเป็นการอธิบายการใช้งานสภาพแวดล้อมของวิชวลเบสิกเบื้องต้นที่ควรทราบ

สำหรับในส่วนที่ไม่ได้อธิบาย จะอธิบายเมื่อเนื้อหามีความสัมพันธ์กันกับสภาพแวดล้อมของวิชวลเบสิก ในส่วนนั้น ๆ จะทำให้ทำความเข้าใจได้ไม่ยาก

และรู้ลึกอย่างต่อเนื่องแถมรวดเร็ว ส่วนเมนูบาร์จะไม่อธิบายโดยตรง เนื่องจากคำสั่งในเมนูบาร์มีการใช้งานเกือบทั้งหมดได้ถูกรวบรวมไว้ในทูลบาร์อยู่แล้ว เมื่อการกล่าวถึงการใช้งานทูลบาร์ใด ผู้เขียนจะอ้างถึงคำสั่งในเมนูบาร์นั้นด้วย

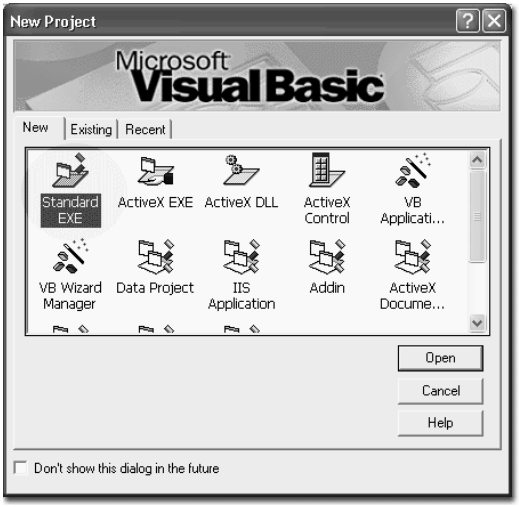


รูปที่ 1-2 แสดงสภาพแวดล้อมของโปรแกรมวิชวลเบสิก 6.0

:: หน้าต่างชิ้นงานใหม่

หน้าต่าง *New Project* จะปรากฏขึ้นมา เมื่อเริ่มต้นใช้งานโปรแกรมวิชวลเบสิก หรือเลือกได้จากเมนู *File>New Project* ไดอะล็อกบล็อกนี้ ก็จะแสดงชนิดของแอปพลิเคชันที่ต้องการพัฒนา

ไอคอนแต่ละตัว จะแทนประเภทของแอปพลิเคชัน ซึ่งสื่อด้วยชื่อ หรือนามสกุล ที่ปรากฏประจำตัวไอคอนนั้น ๆ โดยปกติแล้วแอปพลิเคชันชนิด *Standard.EXE* จะเป็นแอปพลิเคชันพื้นฐานที่ควรจะเริ่มต้นพัฒนา เพราะเป็นแอปพลิเคชันที่ใช้งานทั่วไป ไฟล์ที่ได้จากการคอมไพล์ **จะมีนามสกุล** *.exe*



รูปที่ 1-3 แสดงไดอะล็อกบล็อก New Project

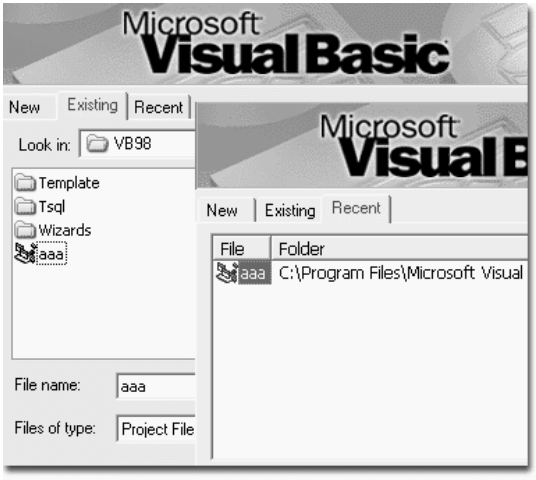
สำหรับรายละเอียดแอปพลิเคชันชนิดอื่น ๆ มีดังนี้

ไอคอน	ความหมาย
Standard EXE	ใช้พัฒนาแอปพลิเคชันทั่วไปและใช้สร้างโปรแกรมแบบกราฟิกทั่วๆ ไป
ActiveX EXE	ใช้พัฒนาแอปพลิเคชันที่สามารถใช้งานและเชื่อมโยงกับแอปพลิเคชันอื่นที่สนับสนุนเทคโนโลยี ActiveX
ActiveX DLL	เป็นแอปพลิเคชันชนิดเดียวกับ ActiveX.EXE แต่จะเก็บอยู่ในลักษณะ ไฟล์ไลบรารี *.dll (In-Process) ซึ่งไม่สามารถรันได้ด้วยตัวมันเอง จะต้องถูกเรียกใช้งานจากแอปพลิเคชันอื่น ๆ

ไอคอน	ความหมาย
 ActiveX Control	ใช้สร้างคอนโทรล ActiveX ขึ้นมาใช้งานเอง
 VB Application Wizard	เป็นเครื่องมือที่อำนวยความสะดวกในการสร้างแอปพลิเคชัน โดยมันจะสร้างองค์ประกอบหลัก ๆ ของแอปพลิเคชันจากขั้นตอนที่ได้เลือกไว้ แอปพลิเคชันที่ได้จะมีความสมบูรณ์ในระดับหนึ่งเท่านั้น จะต้องปรับปรุงเพิ่มเติมและแก้ไข เพื่อทำให้เป็นมาตรฐานยิ่งขึ้น
 VB Wizard Manager	ถือได้ว่าเป็นเครื่องมือที่ช่วยให้โปรแกรมเมอร์ใช้งานโปรแกรมได้เป็นอย่างดี เช่น ติดต่อกับฐานข้อมูล
 Data Project	เป็นชนิดโพรเจกต์ที่เป็นแบบฟอร์ม เพื่อติดต่อกับฐานข้อมูล โดยผ่านทางคอนโทรล ADO Data Control
 IIS Application	แอปพลิเคชันชนิดที่ใช้กับเว็บเซิร์ฟเวอร์
 Addin	เพิ่มเติมโปรแกรมช่วยเข้าไปในสภาวะแวดล้อมของวิชวลเบสิก เพื่อเพิ่ม ประสิทธิภาพ ในการทำงาน
 ActiveX Document Dll	ใช้สร้างแอปพลิเคชันที่ทำงานบนอินเทอร์เน็ต จะเก็บอยู่ในไฟล์ *.dll ไม่สามารถรันด้วยตัวมันเองได้ ต้องให้แอปพลิเคชันที่สนับสนุนเทคโนโลยี ActiveX เรียกใช้งานแทน
 Activex Document Exe	ใช้สร้างแอปพลิเคชันชนิดที่ทำงานบนอินเทอร์เน็ต จะเก็บอยู่ในไฟล์แบบ *.exe ไม่สามารถรันด้วยตัวเองได้ แต่เซิร์ฟเวอร์จะต้องสนับสนุน เทคโนโลยี ActiveX ด้วยเช่นกัน
 DHTML Application	ใช้พัฒนาแอปพลิเคชันรูปแบบของเอกสาร DynamicHTML ซึ่งเป็นมาตรฐานใหม่ของการแสดงผลบนเว็บเพจ
 VB Enterprise Edition Controls	ใช้พัฒนาแอปพลิเคชันในระดับสถานประกอบการ ซึ่งโปรแกรมจะเพิ่ม เติมคอนโทรล ActiveX อีกจำนวนหนึ่งขึ้นมาโดยอัตโนมัติ ซึ่งใช้งานกับโปรแกรมเมอร์มืออาชีพที่มีความเชี่ยวชา-มากพอสมควร

ไดอะล็อกบล็อก New Project มีอยู่อีก 2 แท็บ ดังนี้

- *แท็บ Existing* ใช้สำหรับเปิดโพรเจกต์ที่มีอยู่แล้ว
- *แท็บ Recent* แสดงรายชื่อโพรเจกต์ที่ถูกพัฒนาครั้งหลังสุด เพื่ออำนวยความสะดวกให้กับผู้ใช้งาน



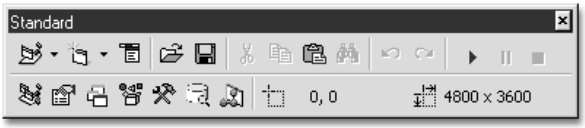
รูปที่ 1.4 แสดงแท็บ Existing และ Recent ในไดอะล็อกบล็อก New Project

:: ทูลบาร์

ทูลบาร์ เป็นส่วนที่รวบรวมคำสั่งที่ใช้งานในการพัฒนาแอปพลิเคชันบ่อยที่สุด จะมีคำสั่งเหมือนกับเมนูบาร์ ซึ่งสามารถที่จะปรับแต่งทูลบาร์นี้ได้ โดยการเลือกเมนู *View>Toolbars* สำหรับชื่อของปุ่มต่าง ๆ ที่อยู่บนทูลบาร์ไม่จำเป็นต้องไปจดจำว่ามันแทนคำสั่งอะไร เพียงแต่ให้เลื่อนเมาส์ไปวางบนปุ่มนั้นซักประมาณ 2 - 3 วินาที จะมีทูลทิปขึ้นมา เพื่อบอกคำสั่งที่ปุ่มนั้นทดแทนอยู่ โดยทูลบาร์สามารถแบ่งออกได้เป็น 4 กลุ่มใหญ่ ๆ คือ

1 ทูลบาร์ Standard

ถือได้ว่าเป็นทูลบาร์ปกติ ที่ต้องใช้งานทุกครั้งและบ่อยที่สุด เนื่องจากประกอบไปด้วย คำสั่งที่เกี่ยวกับการใช้งานทั่ว ๆ ไป ซึ่งรวบรวมคำสั่งมาจากเมนู File, Project, Debug, Run, Tool เป็นต้น



รูปที่ 1.5.1 แสดงรายการทูลบาร์ Standard

2 ทูลบาร์ Edit

จะใช้งานก็ต่อเมื่อเริ่มเขียนโค้ดใน code editor คำสั่งหลักของทูลบาร์กลุ่มนี้คือ Cut, Paste ซึ่งก็คือคำสั่งในเมนู Edit นั่นเอง

รูปที่ 1.5.2 แสดงรายการทูลบาร์ Edit

3 ทูลบาร์ Debug

ใช้ในการตรวจสอบโค้ดว่าทำงานได้ตามความต้องการหรือไม่

รูปที่ 1.5.3 แสดงรายการทูลบาร์ Debug

4 ทูลบาร์ Form Edit

จะใช้งาน เมื่อต้องการปรับขนาด ย้าย เปลี่ยนตำแหน่งคอนโทรลที่อยู่บนฟอร์ม เป็นคำสั่งที่เหมือนกับเมนู Format

รูปที่ 1.5.4 แสดงรายการทูลบาร์ Form Edit

:: ทูลบล็อก

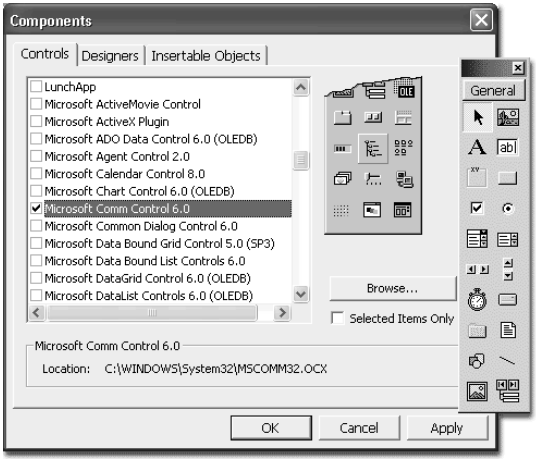
ทูลบล็อก เป็นสิ่งที่รวบรวมคอนโทรล ที่ใช้ในการพัฒนาแอปพลิเคชัน คอนโทรลเป็นสิ่งที่อยู่เบื้องหลังความสำเร็จของโปรแกรมวิชวลเบสิกทุกเวอร์ชัน เพราะมันได้ซ่อนความยุ่งยากด้านเทคนิคไว้ภายใต้การเลือกใช้งานที่แสนง่าย ตัวคอนโทรลสามารถแยกออกเป็นส่วนใหญ่ ๆ ได้ 2 ส่วน คือ

คอนโทรลภายใน

เป็นชุดคอนโทรลมาตรฐานทุกครั้งที่ทำงาน คอนโทรลจะถูกโหลดขึ้นมาอัตโนมัติ ซึ่งสามารถเลือกใช้งานคอนโทรลกลุ่มนี้ได้ทันที โดยที่ไม่ต้องใช้ไฟล์เพิ่มเติม และไม่สามารถถอดคอนโทรลชุดนี้ ออกจากสภาวะแวดล้อมของโปรแกรมได้ เป็นชุดคอนโทรลที่ใช้งานทั่วไปในทุก ๆ แอปพลิเคชัน

คอนโทรล ActiveX

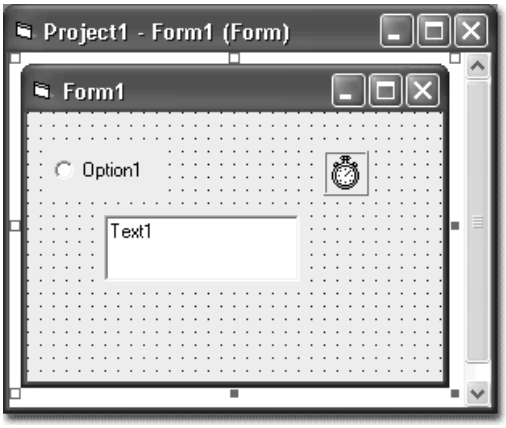
เป็นชุดคอนโทรลที่ใส่เทคโนโลยี ActiveX เข้าไปด้วย ถือว่าเป็นคอนโทรลเพิ่มเติม ถ้าต้องการใช้งาน จะต้องใช้***ไฟล์*** **.ocx* ประจำแต่ละคอนโทรลเพิ่มเข้ามาโดยการเลือกเมนู *Project>Components*



รูปที่ 1.6 แสดงการเพิ่มเติมคอนโทรล ActiveX เข้ามาในทูลบล็อก

:: ฟอร์ม 1 (Form1)

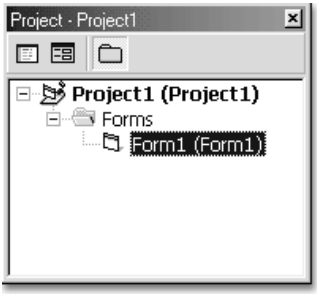
ฟอร์ม 1 ถือได้ว่าเป็นอ็อบเจกต์ตัวแรกที่ได้ใช้งาน จะต้องใช้งานร่วมกับคอนโทรลประกอบกันขึ้นมาเป็นแอปพลิเคชันหนึ่ง ๆ ฟอร์มจัดได้ว่าเป็นตัวบรรจุของคอนโทรลอื่น ๆ และทุกครั้งที่เปิดโปรแกรมวิชวลเบสิกขึ้นมา ฟอร์มว่าง ๆ จะถูกโหลดขึ้นมาเสมอ ซึ่งจะมีความสัมพันธ์กับหน้าต่าง Project Explorer ด้วย ใน 1 โพรเจกต์ จะประกอบไปด้วยอย่างน้อยที่สุด 1 ฟอร์ม แต่มีโพรเจกต์บางชนิดจะไม่มีฟอร์ม เช่น พวกนามสกุล *.dll เป็นต้น



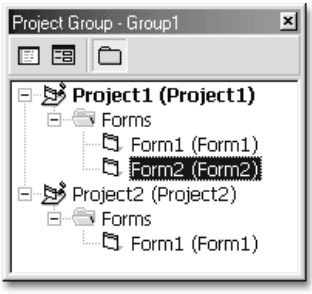
รูปที่ 1.7 แสดงฟอร์มว่าง ๆ ในสภาพแวดล้อมของโปรแกรมวิชวลเบสิก

:: หน้าต่าง Project Explorer

หน้าต่าง Project Explerer ช่วยให้สามารถบริหารจัดการหลาย ๆ โพรเจกต์ได้ ในเวลาเดียวกันอย่างมีประสิทธิภาพ เนื่องจากโปรแกรมวิชวลเบสิกสนับสนุน การพัฒนาแบบหลายชิ้นงาน ซึ่งยังสามารถบันทึกโพรเจกต์เป็นกลุ่มงานที่มีนามสกุลเป็น *.vbg โดยที่โปรแกรมจะจัดกลุ่มโพรเจกต์ต่าง ๆ ให้โดยอัตโนมัติ ซึ่งถ้ามีโพรเจกต์เดียวก็จะมีนามสกุล *.vbp



รูปที่ 1.8
แสดงแบบโพรเจกต์เดียว



รูปที่ 1.9
แสดงแบบหลายโพรเจกต์

หน้าต่าง Project Explorer แสดงรายการองค์ประกอบของแต่ละโพรเจกต์แบบโครงร่างต้นไม้ ตัวโพรเจกต์ถือว่าเป็นตัวแทนแอปพลิเคชันทั้งหมด ซึ่งจะอยู่ส่วนบนสุด ถัดมาจะแสดงองค์ประกอบของโพรเจกต์นั้น ๆ ว่าประกอบไปด้วยอะไรบ้าง เช่น มีฟอร์มกี่ฟอร์ม มีกี่โมดูล เป็นต้น ถ้ามี 2 โพรเจกต์ขึ้นไป ก็จะแสดงแยกออกเป็นส่วนต่างหากอีกโพรเจกต์ ถ้าต้องการใช้งานส่วนใดของโพรเจกต์ สามารถคลิกเลือกได้โดยตรง ซึ่งจะต้องพิจารณาด้วยว่า รายการใดอยู่ภายในโครงสร้างของโพรเจกต์ไหน ก็จะเป็นองค์ประกอบของโพรเจกต์นั้น ซึ่ง***แสดงดังรูปที่ 1.8 และรูปที่ 1.9***

ในโพรเจกต์หนึ่งจะประกอบไปด้วยอย่างน้อยที่สุด 1 ฟอร์ม ยกเว้นกรณีที่สร้างแอปพลิเคชัน ชนิด *.dll ไม่จำเป็นจะต้องมีฟอร์ม ***ถ้าต้องการเพิ่มฟอร์มเข้าไปในโพรเจกต์***ให้ทำดังนี้

- เลื่อนเมาส์ไปบริเวณหน้าต่าง *Project Explorer*
- ***คลิกขวา*** เพื่อแสดง pop-up เมนู
- เลือกคำสั่ง *Add>Form*

สำหรับรายการชนิดของอ็อบเจกต์ที่ปรากฏอยู่ในหน้าต่าง Project Explorer มีหลายชนิดที่เป็นสิ่งจำเป็น ในการเริ่มต้นสร้างแอปพลิเคชันหนึ่ง ๆ ให้มีความสมบูรณ์มากยิ่งขึ้น

ชื่อรูปแบบ	ความหมาย
Project (n)	แอปพลิเคชันที่พัฒนาอยู่ อาจมีโพรเจกต์เดียว หรือหลายโพรเจกต์ก็ได้ โดยปกติมีนามสกุล *.vbp
Form (n)	เป็นฟอร์มที่เรียกใช้งานจากโพรเจกต์ อาจมีมากกว่า 1 ฟอร์มก็ได้ ซึ่งปกติแล้วจะมีนามสกุล *.frm
Modules	ที่เก็บชุดคำสั่ง หรือรูทีนที่เขียนขึ้นมาจะเก็บบันทึกไว้ใช้งานภายหลังได้ อีกด้วย โปรแกรมเมอร์มักจะเก็บคำสั่ง หรือโพรซีเยอร์ที่ใช้งานบ่อย ไว้ในโมดูลชนิดนี้ มีนามสกุล *.bas
Class modules	เป็นโมดูลชนิดพิเศษ ที่มีลักษณะเป็นอ็อบเจกต์ ที่สามารถสร้างขึ้นมาได้ มีนามสกุล *.cls
User controls	เป็นคอนโทรล ActiveX ที่สร้างขึ้นมา มีนามสกุล *.ctl

:: หน้าต่างคุณสมบัติ

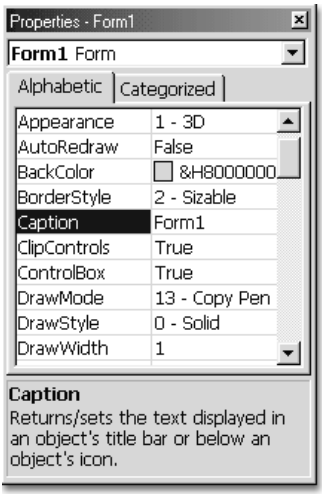
หน้าต่างคุณสมบัติ จะมีความสัมพันธ์กับฟอร์ม คอนโทรล ที่มีสถานะแอกทีฟ หรือได้รับความสนใจในขณะนั้น ซึ่งโปรแกรมจะเปลี่ยนแปลงหน้าต่างคุณสมบัติ ให้ตรงกับอ็อบเจกต์โดยอัตโนมัติ

โดยปกติแล้ว โปรแกรมวิชวลเบสิกจะตั้งค่าปกติให้โดยอัตโนมัติ ซึ่งจะมีทั้งส่วนที่เหมาะสมดีอยู่แล้ว และส่วนที่โปรแกรมตั้งค่ากลาง ๆ เอาไว้ให้ จะมีทั้งที่ต้องปรับแต่งและไม่ต้องยุ่งเกี่ยว คุณสมบัติบางตัวเท่านั้นที่ควรปรับแต่งค่าเริ่มต้นในหน้าต่างคุณสมบัตินี้ อีกส่วนหนึ่งจะใช้วิธีการเขียนโค้ด เพื่อปรับแต่งค่าคุณสมบัติ ซึ่งเป็นเทคนิคที่ทางไมโครซอฟต์แนะนำให้ใช้งาน

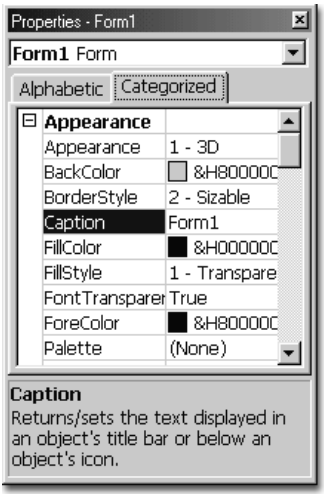
ผู้ใช้งานสามารถเลือกรายการคอนโทรล ได้จากหน้าต่างคุณสมบัติเช่นกัน โดย***คลิกที่*** *drop-down list box* จะแสดงรายการคอนโทรลที่ใช้งานอยู่ และยังสามารถทราบความหมายเบื้องต้นของคุณสมบัติที่เลือก โดยมีคำนิยามสั้น ๆ ที่อยู่ด้านล่างอีกด้วย

ในหน้าต่างคุณสมบัติ ประกอบไปด้วย 2 แท็บ ดังนี้

- ***แท็บ Alphabetic*** เป็นแท็บที่แสดงรายการคุณสมบัติ เรียงตามตัวอักษรในภาษาอังกฤษ
- ***แท็บ Categorized*** เป็นแท็บที่แสดงรายการคุณสมบัติ โดยจัดกลุ่มของคุณสมบัติที่มีหน้าที่คล้ายกัน หรือมีความสัมพันธ์กัน



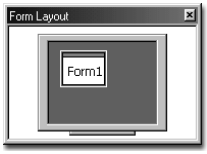
รูปที่ 1.10
หน้าต่างคุณสมบัติแบบ Alphabetic



รูปที่ 1.11
หน้าต่างคุณสมบัติแบบ Categorized

:: หน้าต่าง Form Layout

หน้าต่าง Form Layout ใช้กำหนดตำแหน่งของฟอร์ม ที่จะปรากฏบนจอภาพในขณะทำงาน สามารถกำหนดตำแหน่งของฟอร์ม โดยการย้ายฟอร์มจำลองที่อยู่ในจอภาพ ด้วยการลากเมาส์ไปยังตำแหน่งที่ต้องการ มีข้อสังเกต คือ เมื่อเคลื่อนย้ายฟอร์มจำลองแล้ว ฟอร์มจริงจะไม่ได้เคลื่อนย้ายไปยังตำแหน่งดังกล่าว เพราะว่ามันจะมีผลในขณะรันเท่านั้น ***เรียกว่า ช่วง runtime*** ส่วนฟอร์มใหญ่ ที่เห็นอยู่กลางจอภาพ จะเป็นช่วง *design time* เมื่อทำงาน จะรู้ว่าฟอร์มอยู่ตรงไหนของจอภาพ



รูปที่ 1.12
หน้าต่าง Form Layout

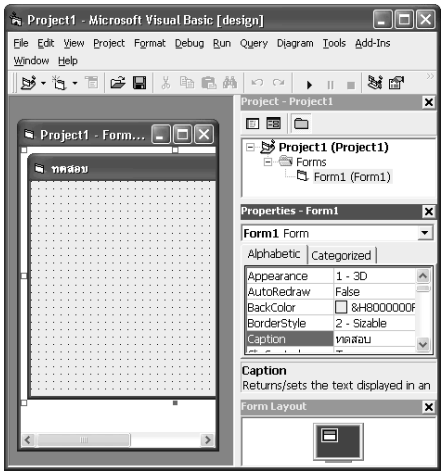
:: หน้าต่างแสดงผลทันที (Immediate Window)

เป็นหน้าต่างที่ให้ประโยชน์ ในกรณีที่ต้องการทราบผลโดยทันที เช่น การทดสอบโพรซีเยอร์ เมื่อสั่งรันโพรเจกต์ หน้าต่างนี้ จะปรากฏขึ้นโดยอัตโนมัติ สามารถเรียกใช้งาน โดยเลือกไปที่เมนู *View>Immediate Window*

รูปที่ 1-13 แสดงหน้าต่าง Immediate

:: สภาพแวดล้อมโดยรวมของโปรแกรมวิชวลเบสิก

สภาพแวดล้อมของโปรแกรมวิชวลเบสิก สามารถแบ่งได้ 2 ประเภท ดังต่อไปนี้



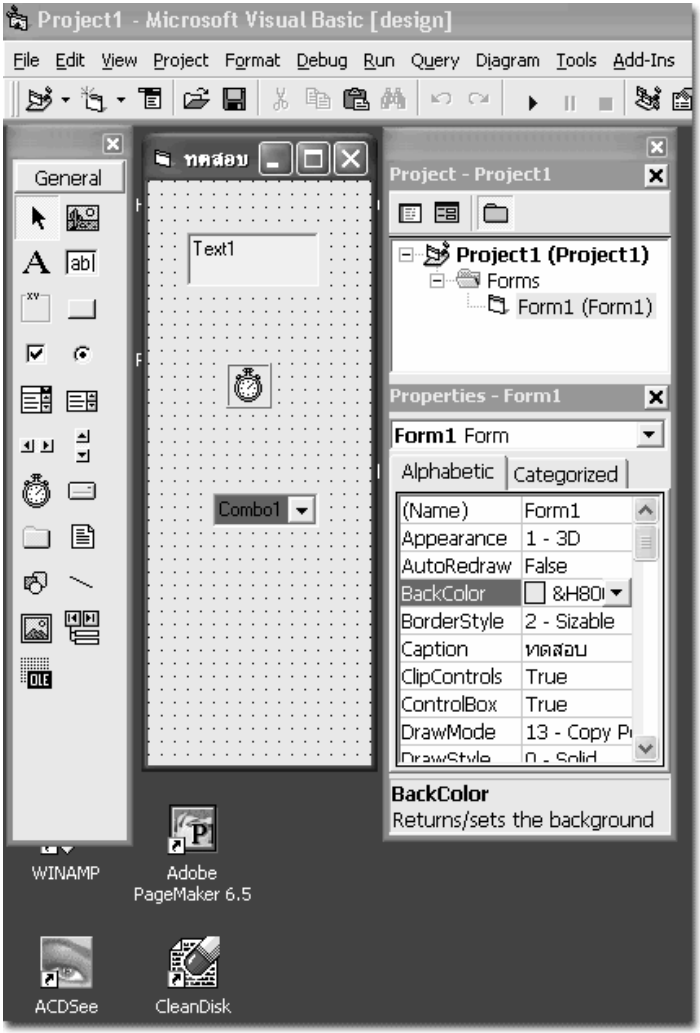
รูปที่ 1-14 แสดงสภาพแวดล้อมแบบ MDI

1 โหมด MDI

โหมด MDI (Multiple Document Interface) จะแสดงหน้าต่างในรูปแบบเป็นหนึ่งเดียวกัน ซึ่งเป็นสภาพแวดล้อมปกติของโปรแกรมวิชวลเบสิก

2 โหมด SDI

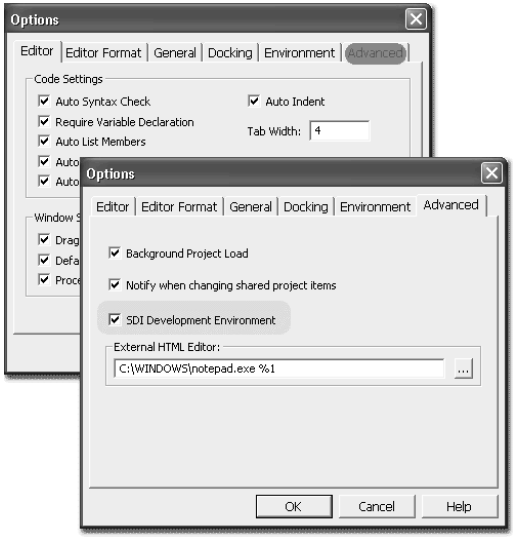
โหมด SDI (Single Document Interface) จะแสดงหน้าต่างที่มีลักษณะเป็นอิสระต่อกัน แต่ยังคงมีความสัมพันธ์กัน เหมือนกับโหมด MDI แต่จะใช้พัฒนาแอปพลิเคชันอีกชนิดหนึ่ง



รูปที่ 1-15 แสดงสภาพแวดล้อมแบบ SDI

:: การเปลี่ยนสภาพแวดล้อมจาก MDI เป็น SDI

- เลือกเมนู *Tools>Options* จะปรากฏไดอะล็อกบล็อก *Options*
- คลิกที่แท็บ *Advanced* เลือก *SDI Development Environment*
- คลิกที่ปุ่ม *OK*



รูปที่ 1-16 แสดงการเปลี่ยนสภาพแวดล้อมจาก MDI เป็น SDI

:: การปรับแต่งสภาพแวดล้อมโปรแกรมให้ตรงกับการใช้งาน

ในการใช้งานจริง ๆ แล้ว ไม่จำเป็นต้องแสดงผลทุกหน้าต่างพร้อมกัน เพราะจะทำให้พื้นที่ในการทำงานมีน้อย เมื่อพัฒนาแอปพลิเคชันไปในระยะหนึ่ง จะพบว่าสภาพแวดล้อมแบบใดเหมาะสมกับโพรเจกต์ และจะแสดงอะไรบ้าง ผู้เขียนจะพยายามปิดหน้าต่างที่ไม่ได้ใช้งานเสมอ เพราะถือว่า ถ้าเราจัดโต๊ะทำงานให้เป็นระเบียบเรียบร้อย มันจะส่งผลให้การพัฒนาแอปพลิเคชัน สะดวกและรวดเร็วขึ้นไปอีกระดับหนึ่งอีกด้วย

ข้อตกลงในการใช้หนังสือเล่มนี้

การที่จะเริ่มเขียนแอปพลิเคชันอย่างแท้จริง จะต้องมีแนวทางการเขียนโปรแกรม ดังนี้

ซอร์สโค้ด

ซอร์สโค้ดของโปรแกรม จะใช้ฟอนต์ที่มีลักษณะแตกต่างจากฟอนต์ปกติ และใช้การเว้นวรรคตัดคำ ให้ตรงกับไวยากรณ์ที่วิชวลเบสิกยอมรับ หรือกำหนดมา เช่น

```
Private Sub Command1_Click ( )
Text1.Text = "วิศวกร นอกคอก"
End Sub
```

ไวยากรณ์

สำหรับไวยากรณ์ จะใช้**ตัวเข้ม**เพื่อแทน รูปแบบ (format) คำสั่ง (statements) ฟังก์ชัน (functions) คำสงวน (keyword) ให้ตรงกับรูปแบบการนำเสนอของ msdn Library เพื่อให้ผู้ใช้งานมีความคุ้นเคย และจะใช้**ตัวธรรมดา** เพื่อแสดงส่วนของ ตัวแปร พารามิเตอร์ อาร์กิวเมนต์ เช่น

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

วงเล็บปิด [ ]

วงเล็บปิด [ ] จะไม่พิมพ์ลงไป และตัวแปรที่อยู่ในวงเล็บ หมายถึง ผู้อ่านจะใส่หรือไม่ใส่ค่าก็ได้ ขึ้นอยู่กับการใช้งาน เช่น

```
Object.Caption [= string]  หรือ  Object.Circle [Step](x, y), radius, [color, start, end]
```

วงเล็บปีกกา { }

วงเล็บปีกกา { } ไม่ต้องพิมพ์ลงไป และตัวแปรที่อยู่ในวงเล็บดังกล่าว จะถูกคั่นด้วยเครื่องหมาย | ซึ่งจะต้องใส่ตัวแปรตัวใดตัวหนึ่งในวงเล็บนั้น เช่น

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

■ *คำนิยาม*

คำนิยาม หมายถึง คำหรือข้อความ ที่ใช้สำหรับกล่าวอ้างอิง ซึ่งมีความหมายดังต่อไปนี้

- *default* ค่าที่โปรแกรมวิชวลเบสิกตั้งไว้ให้แล้ว จะเปลี่ยนหรือไม่ก็ได้
- *optional* ข้อกำหนดเพิ่มเติมของการใช้งานตัวแปรนั้น ๆ จะใส่ หรือไม่ใส่ก็ได้
- *application* เป็นคำที่ผู้เขียนใช้แทนโปรแกรมทั่ว ๆ ไปในท้องตลาด ซึ่งอาจรวมถึงโปรแกรมที่ทำงานภายใต้ทุกระบบปฏิบัติการ
- *compile* การแปลงโพรเจกต์ให้เป็นแอปพลิเคชันที่สมบูรณ์ ซึ่งโดยปกติแล้วจะมี นามสกุลเป็น *.exe
- *argument* ตัวแปร ค่าคงที่ หรือพารามิเตอร์ ที่ใช้สำหรับส่งค่าเข้าไปในฟังก์ชันหรือส่งค่าเข้าไปในโพรซีเยอร์
- *design time* ช่วงเวลาที่โปรแกรมเมอร์ออกแบบอินเตอร์เฟส หรือ เขียนโค้ด อาจใช้คำว่าขณะออกแบบก็ได้
- *keyword* คำสงวน หมายถึงคำ หรือข้อความที่โปรแกรมวิชวลเบสิก นำไปใช้เป็นชุดคำสั่ง คอนโทรล อ็อบเจกต์ ฟังก์ชัน ซึ่งคำเหล่านี้จะไม่สามารถนำไปใช้ตั้งชื่อ ที่ไม่ใช่หน้าที่ หรือไวยากรณ์ของมันได้
- *run time* ช่วงเวลาที่โปรแกรมเมอร์ทดสอบโค้ด หรือ กำลังรันโพรเจกต์อยู่จะไม่สามารถแก้ไขโค้ดได้ โดยการกดปุ่ม F5 หรือกดปุ่มรัน
- *project* ชิ้นงานซอร์สโค้ดของโปรแกรมที่ยังไม่ได้ถูกแปลง

:: การปรับแต่งให้อ่านโค้ดภาษาไทยได้

ลองมาดูโค้ดตามรายละเอียด ดังนี้

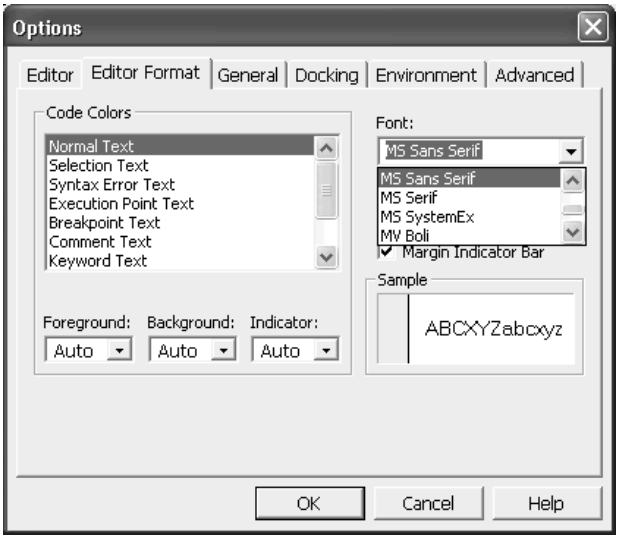
```
Private Sub Command1_Click ( )
        Text1.Text = " Love You..."
End Sub

Private Sub Command2_Click ( )
        Text1.Text = "?×(Y©∇?Y?ƒ/∝∇? "
End Sub
```

พิมพ์ภาษาไทย แต่ผลลัพธ์ออกมาเป็นอักขระ

ขั้นตอนการปรับแต่ง ทำได้ดังนี้

- เลือกเมนู *Tools>Options*
- คลิกที่แท็บ *Editor Format* เลือกรายละเอียดต่าง ๆ ***ดังรูปที่ 1-17***
  - *Font :* ปรับแต่งเป็น *MS Sans Serif* หรือฟอนต์ที่ลงท้ายด้วย *UPC*
  - *Size :* แนะนำให้ปรับแต่งเป็น *10* จะดูดีที่สุด



รูปที่ 1-17 แสดงการปรับแต่งให้โปรแกรมวิชวลเบสิกอ่านภาษาไทยได้

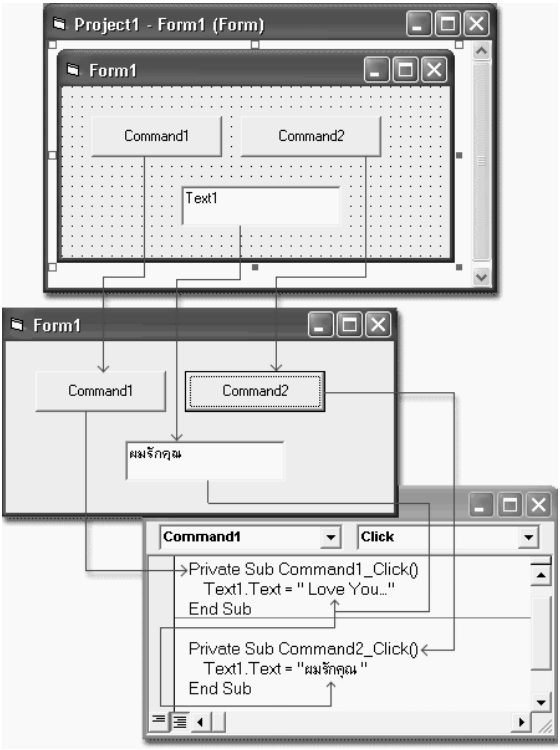
พอปรับแต่งฟอนต์ที่สนับสนุนภาษาไทย ให้โปรแกรมวิชวลเบสิกอ่านภาษาไทยได้ ในส่วนของซอร์สโค้ดก็จะสามารถพิมพ์ภาษาไทยได้ทันทีเช่นกัน

```
Private Sub Command1_Click ( )
        Text1.Text = " Love You..."
End Sub

Private Sub Command2_Click ( )
        Text1.Text = "ผมรักคุณ... "
End Sub
```

พิมพ์ภาษาไทย อ่านภาษาไทย ได้แล้ว

ต่อไปเป็นตัวอย่างที่แสดงให้เห็นการเชื่อมโยง 2 โพรซีเยอร์เข้าด้วยกัน ให้วาดอินเตอร์เฟส และเขียนโค้ดกำกับ ***ดังรูปที่ 1-8***



รูปที่ 1-18 การเชื่อมโยง 2 โพรซีเยอร์เข้าด้วยกัน และเขียนโค้ดภาษาไทย

:: ความสัมพันธ์ของคุณสมบัติ เมท็อด และอิเวนต์

ก่อนที่จะทำการออกแบบแอปพลิเคชันของการทำงานต่าง ๆ จะต้องทำความคุ้นเคยกับความสัมพันธ์ทั้ง 3 ส่วน คือ คุณสมบัติ (properties) เมท็อด (method) และอิเวนต์ (event) ให้ดีเสียก่อน เพื่อจะได้ไม่ต้องสับสน ในการกล่าวอ้างถึง รวมถึงรูปแบบของการใช้งานด้วย

1 คุณสมบัติ

การกำหนดคุณสมบัติของแต่ละชิ้นงานจะไม่เหมือนกัน ขึ้นอยู่กับว่าจะนำคอนโทรลไหนมาวาดลงบนฟอร์ม จะอธิบายให้ทราบได้จากบทต่อ ๆ ไป

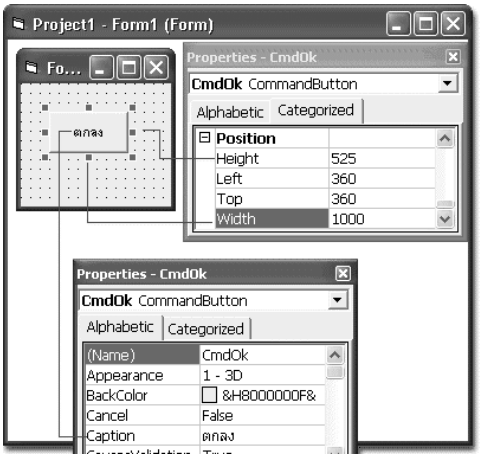
2 เมท็อด

เป็นคำสั่งเฉพาะตัวของอ็อบเจกต์ มีทั้งใช้กับตัวเอง และใช้กับอ็อบเจกต์อื่น เช่น method “clear” ใช้ได้กับ ListBox, ComboBox เป็นต้น

3 อิเวนต์

เป็นเหตุการณ์ที่เกิดขึ้นกับตัวอ็อบเจกต์ เพื่อตอบสนองการใช้งาน

ลองมาดูตัวอย่าง ที่แสดงถึงความสัมพันธ์ทั้ง 3 ส่วนเข้าด้วยกัน คือ คุณสมบัติ (properties) เมท็อด (method) และอิเวนต์ (event) จะใช้คอนโทรลที่ชื่อว่า *CommandButton* เป็นตัวแสดงความสัมพันธ์



รูปที่ 1-19 แสดงการกำหนดคุณสมบัติ

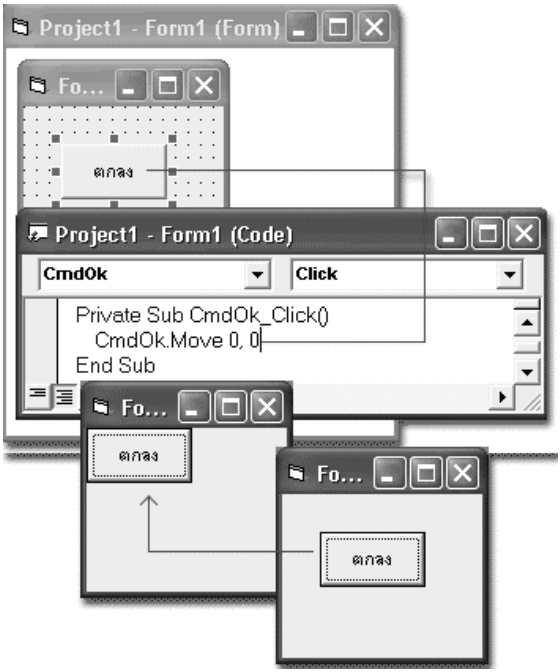
CmdOk.*Caption* = *“ตกลง”* เป็นการกำหนดคุณสมบัติ *Caption* ของ *CmdOk*

CmdOk.*Hight* = *525* เป็นการกำหนดคุณสมบัติ *Hight* ของ *CmdOk*

CmdOk.*Width* = *1000* เป็นการกำหนดคุณสมบัติ *Width* ของ *CmdOk*

Note !

โปรดสังเกตว่าข้อมูลชนิดข้อความ จะต้องมีเครื่องหมาย **"xxxxxx"** ปิดข้อมูลนั้นด้วย แต่กับข้อมูลชนิดตัวเลขไม่จำเป็นต้องมี **"xxxxxx"** ปิดข้อมูลนั้นด้วย



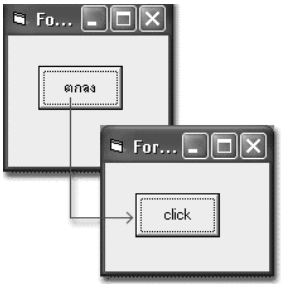
รูปที่ 1-20 แสดงการกำหนดเมท็อด

การสั่งให้คอนโทรลปุ่มคำสั่งชื่อ CmdOk ย้ายไปที่ตำแหน่ง 0, 0 ทำงานตามที่ต้องการ

รูปที่ 1-21 แสดงการเกิดเหตุการณ์ CmdOk_Click ( ) ขึ้นมา

ถ้าต้องการตอบสนองต่อเหตุการณ์ *Click* ของปุ่มคำสั่ง *CmdOk* ซึ่งในการตอบสนองต่อเหตุการณ์ Click ข้างต้น ให้ทดลองใส่คำสั่ง ดังนี้

```
Sub CmdOk_Click ( )          คำสั่งที่ตอบสนองต่อเหตุการณ์ Click ของ CmdOk
  CmdOk.Caption = "click"    เปลี่ยนคุณสมบัติ Caption ของ CmdOk เป็น Click
End Sub                      จบการตอบสนองต่อเหตุการณ์ Click ของ CmdOk
```



จะพบว่าเมื่อคลิกเมาส์ที่ปุ่มคำสั่งแล้ว คำว่า ***ตกลง*** จะเปลี่ยนไปเป็น *Click* แทน

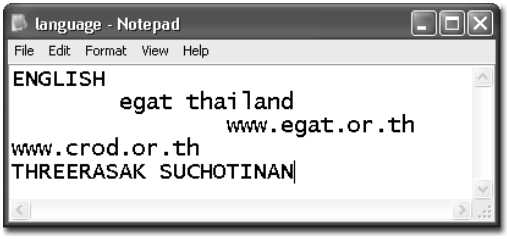
รูปที่ 1-22 แสดงอิเวนต์

:: รู้จักกับแอปพลิเคชันชนิด SDI และ MDI

แอปพลิเคชันทั่ว ๆ ไป ที่ทำงานอยู่ภายใต้ระบบปฏิบัติการของวินโดวส์ จะมีอินเตอร์เฟสหลัก ๆ อยู่ด้วยกัน 3 แบบ คือ

1. อินเตอร์เฟสแบบ Single Document Interface (SDI)
2. อินเตอร์เฟสแบบ Multiple Document Interface (MDI)
3. อินเตอร์เฟสแบบ Windows Explorer

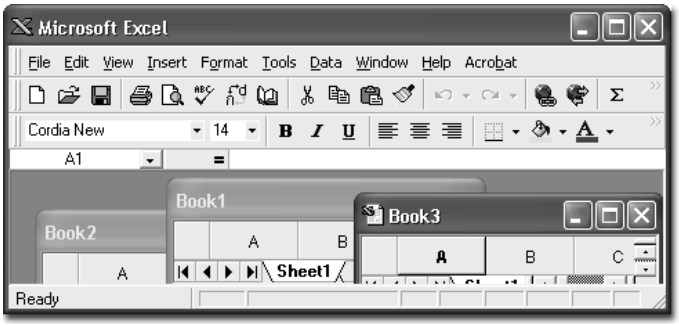
แอปพลิเคชันแบบ SDI ที่พบเห็นกันอยู่เสมอ คือโปรแกรม NotePad ในวินโดวส์เอง ซึ่งเป็นตัวอย่างของแอปพลิเคชันชนิด SDI ได้ดีที่สุด



รูปที่ 1-23 แสดงอินเตอร์เฟสแบบ SDI ของ Notepad

โดยมีลักษณะที่เฉพาะ คือ สามารถเปิดเอกสารมาใช้งานได้เพียงครั้งละ 1 ไฟล์เท่านั้น ถ้าต้องการเปิดเอกสารชุดใหม่ ต้องปิดเอกสารเก่าเสียก่อน ซึ่งจะไม่สามารถดูเอกสารพร้อม ๆ กันได้ จึงถูกเรียกว่าอินเตอร์เฟสแบบ SDI

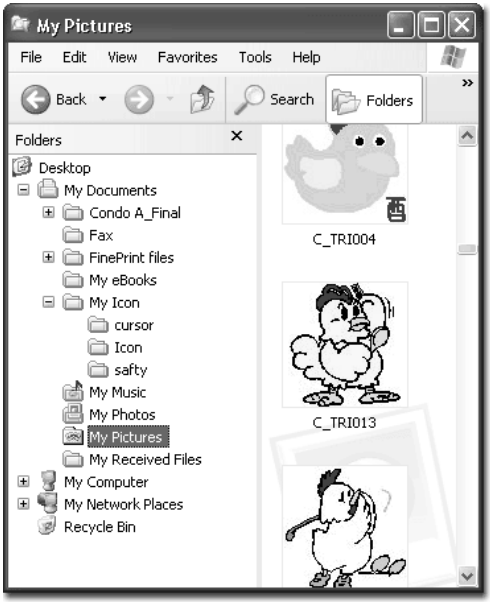
สำหรับแอปพลิเคชันแบบ MDI ที่พบเห็นบ่อย คือโปรแกรมในชุดออฟฟิศทั้งหลาย เช่น Microsoft Word, Microsoft Excel เป็นต้น ซึ่งเป็นตัวอย่างของแอปพลิเคชันชนิด MDI ได้ดีที่สุด



รูปที่ 1-24 แสดงอินเตอร์เฟสแบบ MDI ของ Microsoft Excel

จะมีลักษณะเฉพาะ คือ สามารถเปิดเอกสารใช้งานได้มากกว่า 1 ไฟล์ ถ้าต้องการเปิดเอกสารชุดใหม่ก็สามารถเปิดได้ทันที ซึ่งมีความเป็นอิสระต่อกัน จึงถูกเรียกว่าอินเตอร์เฟสแบบ MDI ซึ่งจะมีหน้าต่างเป็นของตัวเอง ในโปรแกรมวิชวลเบสิกเรียกว่า ChildForm ดูได้จาก***รูปที่ 1-24*** เป็นสภาพแวดล้อมของ Microsoft Excel ที่เปิด worksheet 3 ชุดพร้อมกัน ในแต่ละ worksheet จะไม่มีส่วนเกี่ยวข้องกัน

สำหรับอินเตอร์เฟสแบบ Explorer จะมีลักษณะแบ่งเป็น 2 ฝั่ง คือ ซ้ายและขวา โดยด้านซ้ายจะมีลักษณะแสดงเป็นโครงร่างต้นไม้ ส่วนด้านขวาเป็นรายละเอียดของอ็อบเจกต์ที่ถูกเลือกจากฝั่งซ้าย ซึ่งเป็นที่นิยมกันมากในปัจจุบัน



รูปที่ 1-25 แสดงอินเตอร์เฟสแบบ Explorer ของ Windows Explorer

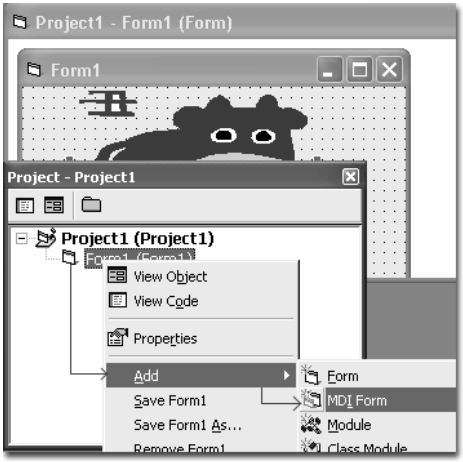
:: การใช้งานฟอร์มในรูปแบบ MDI

ฟอร์มชนิด MDI เป็นรูปแบบแอปพลิเคชัน ที่ประกอบไปด้วยฟอร์มแม่ ซึ่งเป็นตัวบรรจุฟอร์มลูก ถ้าเปรียบเทียบกับ Microsoft Excel แล้ว ดูเหมือนว่าสภาพแวดล้อมของ Microsoft Excel คือ ฟอร์มแม่ ส่วน worksheet ที่เปิดขึ้นมาคือ ฟอร์มลูกนั่นเอง

วิธีการทำฟอร์มชนิด MDI มี 2 ขั้นตอน คือ

1. ให้เพิ่ม *MDI Form* เข้ามาในสภาพแวดล้อมของโปรแกรมวิชวลเบสิก
2. และจะต้องทำให้ฟอร์มที่เห็นในโปรแกรมเป็นฟอร์มลูก

พิจารณาจาก***รูปที่ 1-26*** ให้คลิกขวาที่บริเวณ *Form1 (Form1)* แล้วเลือกคำสั่ง *Add>MDI Form* หรือคำสั่ง *Project>Add MDIForm* เลือกไอคอน *MDI Form*

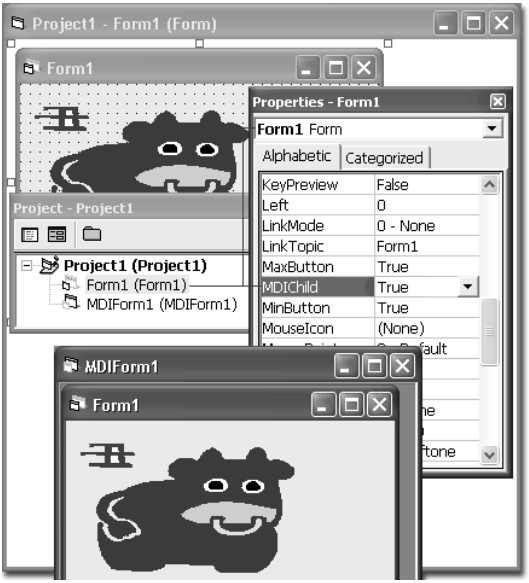


รูปที่ 1-26 แสดงการเพิ่ม MDI Form เข้ามาในโปรแกรม

พิจารณาจาก***รูปที่ 1-27*** ให้สังเกตที่ *Project Explorer* จะพบว่าไอคอนของฟอร์มปกติกับ MDI Form จะแตกต่างกัน

ให้เปลี่ยนคุณสมบัติ *MDI Child* ของ *Form1* = *True* แล้วรันโพรเจกต์ดู จะพบว่า *MDI Form* กลายเป็นฟอร์มแม่ของแอปพลิเคชัน ส่วนฟอร์มปกติ จะเป็นฟอร์มลูก

รูปที่ 1-27 แสดงไอคอนของ Form แบบธรรมดากับชนิด MDI Form



รูปที่ 1-28 แสดงการทำงานของโพรเจกต์ชนิด MDI Form

จะพบว่าไอคอนของฟอร์มปกติเปลี่ยนไป ซึ่งสามารถทราบได้ว่าฟอร์มใดเป็นฟอร์มแม่ หรือฟอร์มใดเป็นฟอร์มลูก โดยดูที่ไอคอนที่กำกับอยู่ จาก***รูปที่ 1-28*** จะพบว่า *Form1* ***เป็นฟอร์มลูก***เพราะ***ไอคอนเล็ก***มีความเด่นชัด ส่วน *MDIForm1* ***เป็นฟอร์มแม่***เพราะว่า***ไอคอนใหญ่***เด่นชัด ข้อสังเกตนี้ จะมีประโยชน์ในกรณีที่เปลี่ยนชื่อฟอร์มใหม่ และในโพรเจกต์ที่มีฟอร์มหลายฟอร์ม ซึ่งจะทำให้ทราบได้ว่าฟอร์มใดเป็นฟอร์มหลัก คุณสมบัติที่สำคัญของฟอร์มลูก คือ จะสามารถเคลื่อนย้ายฟอร์มได้เฉพาะในขอบเขตของฟอร์มแม่เท่านั้น

:: การคิดแบบนักเขียนโปรแกรม

การเขียนโปรแกรมครั้งแรก ผู้เขียนอาจสับสนว่าโปรแกรมมีรูปแบบใด และจะสร้างขึ้นมาได้อย่างไร ในโปรแกรมวิชวลเบสิก ไม่ต้องกังวลใจ ***หนังสือเล่มนี้มีคำตอบ***

โปรแกรมคืออะไร

โปรแกรม คือ ชุดคำสั่งซึ่งทำให้คอมพิวเตอร์ทำงานตามที่ผู้ใช้งานต้องการ ในการเขียนโปรแกรมวิชวลเบสิก ให้ตรวจสอบแต่ละขั้นตอน ดังนี้

การออกแบบโปรแกรม

ขั้นตอนแรกของการเขียนโปรแกรม คือ ต้องรู้ว่าจะให้โปรแกรมทำอะไร และผู้ใช้งานต้องการอะไร ดูเหมือนง่าย แต่ไม่ได้คำนึงถึงประโยคคำสั่งตรง ๆ ทำให้นักเขียนโปรแกรม อาจพบกับปัญหาในการสร้างแอปพลิเคชันได้ การออกแบบโปรแกรมเป็นเหมือนกับการออกแบบหน้าขนม Cake ซึ่งต้องมีการเตรียมการก่อน

การออกแบบส่วนติดต่อผู้ใช้งาน

จากการที่กำหนดเป้าหมาย สำหรับการเขียนโปรแกรมแล้ว ต่อไปสิ่งสำคัญที่จะคิด คือ เกี่ยวกับหน้าตา และการประมวลผลข้อมูล หน้าจอที่สมบูรณ์และรูปภาพที่ใช้ใน โปรแกรม ซึ่งถูกเรียกว่า ***ส่วนติดต่อผู้ใช้ของโปรแกรม*** รวมไปถึงเมนูออบเจกต์ ไดอะล็อกบล็อก ปุ่ม และรูปภาพประกอบ ซึ่งผู้ใช้งานจะได้เห็นขณะโปรแกรมกำลังทำงาน ระบบการเขียนโปรแกรมของโปรแกรมวิชวลเบสิก ให้สร้างองค์ประกอบทั้งหมดบนวินโดวส์ เหมือนผู้เชี่ยวชาญได้อย่างรวดเร็ว

สำรวจตัวเองก่อนออกแบบโปรแกรม

เมื่อออกแบบโครงงานของการเขียนโปรแกรมแล้ว อาจพบว่าการเขียนส่วนติดต่อกับผู้ใช้งาน ที่ต้องการแสดงให้ผู้ใช้เห็นว่ามีประโยชน์มาก ซึ่งยังช่วยให้คิดถึงกระบวนการทำงานของโปรแกรมอีกด้วย และจะทำให้รู้ว่า อะไรคือเป้าหมายของการเขียนโปรแกรม ใครจะเป็นผู้ใช้โปรแกรม โปรแกรมจะมีหน้าตาเป็นอย่างไร เมื่อเริ่มทำงาน ผู้ใช้ต้องป้อนข้อมูลแบบใดเข้าไปในโปรแกรม โปรแกรมจะประมวลผลอย่างไรและได้ผลลัพธ์เป็นแบบไหน ต้องคิดก่อนเสมอ

การสร้างโปรแกรมด้วยโปรแกรมวิชวลเบสิก

การสร้างโปรแกรม มีขั้นตอนอยู่ 3 ขั้นตอน คือ

1. การสร้างส่วนติดต่อผู้ใช้งาน
2. การตั้งค่าคุณสมบัติ
3. การเขียนรหัส หรือ โค้ด

■ **ภาษาของโปรแกรมวิชวลเบสิก**

ภาษาในการเขียนโปรแกรมวิชวลเบสิก จะมีประโยคคำสั่ง ฟังก์ชัน และตัวอักษรพิเศษอีกหลายชนิด กฎเกณฑ์ และเนื้อหาของภาษาการเขียนโปรแกรมวิชวลเบสิกจะเรียกว่า ***รูปแบบการเขียนโปรแกรม*** *(Syntax)* จะต้องมีการจัดลำดับของคำสำหรับตัวแปลภาษา จะพบว่าความง่ายในการใช้งานโปรแกรมวิชวลเบสิกเป็นเหตุผลให้ตัวมันเองถูกเลือกให้มาเป็นตัวเขียนโปรแกรมบนวินโดวส์ที่รู้จักกันดี คือ โปรแกรมด้านตาราง Microsoft Excel

การทดสอบและแจกจ่ายโปรแกรม

เมื่อสร้างโปรแกรมเรียบร้อยตามวัตถุประสงค์ที่ต้องการแล้ว จะต้องทดสอบโปรแกรมที่เขียนอย่างละเอียด โดยการแปลงโค้ดไฟล์โพรเจกต์ให้เป็นไฟล์ใช้งานที่มี***นามสกุล*** **.exe* ***ซึ่งหมายถึงไฟล์*** *Execute* ที่สามารถทำงานได้โดยไม่ต้องอยู่ในสภาพแวดล้อมของโปรแกรมวิชวลเบสิกอีกต่อไป เพื่อนำไปแจกจ่ายให้กับผู้ใช้งานภายหลัง ซึ่งกระบวนการโดยเริ่มต้น จากการออกแบบและวิเคราะห์เป้าหมายใหม่ ขั้นตอนเหล่านี้ จะอยู่ในวงลูปของการพัฒนา การเขียนโปรแกรมที่สมบูรณ์ จะต้องเป็นแบบนี้ทุกโปรแกรม ซึ่งมีขั้นตอน ดังต่อไปนี้

■ **การทดสอบโปรแกรมที่ออกแบบ**

การทดสอบโปรแกรม เป็นการตรวจสอบสถานะของการทำงานจริง เพื่อตัดสินใจว่าทำงานถูกต้องตามวัตถุประสงค์หรือไม่ ซึ่งอาจจะพบปัญหาที่ทำให้โปรแกรมหยุดทำงานกลางคัน ซึ่งจะเรียกว่า***โปรแกรมมีบัก*** เราต้องหาความผิดพลาดนั้นให้พบก่อนจำหน่ายจ่ายแจกไปยังผู้ใช้งาน ในโปรแกรมวิชวลเบสิกมีเครื่องดีบักที่สามารถตรวจจับจุดผิดพลาดในการเขียนโปรแกรมได้เป็นอย่างดี

■ **การแปลงโค้ดไฟล์โพรเจกต์**

การแปลงโค้ดไฟล์โพรเจกต์ เมื่อทำการสร้างและทดสอบโปรแกรมเป็นอย่างดีแล้ว ต้องแปลงโค้ดไฟล์โพรเจกต์ ให้เป็นโปรแกรมที่สามารถทำงานด้วยตัวมันเองได้

■ **การแจกจ่ายโปรแกรมไปยังผู้ใช้งาน**

เมื่อแปลงโค้ดเรียบร้อยแล้ว ต่อไป คือ การแจกจ่ายโปรแกรมไปยังผู้ที่ต้องการใช้งาน

จุดประสงค์ของหนังสือเล่มนี้ จะเน้นการนำไปใช้งานให้มากที่สุด โดยแสดงตัวอย่างประกอบ เพื่อให้ผู้อ่านเห็นภาพได้อย่างชัดเจน แต่ยังไม่ทิ้งแนวคิดพื้นฐานในการใช้งาน และจะละไว้บางส่วนที่ไม่สำคัญ เนื่องจากสามารถทำความเข้าใจเองได้อย่างไม่ยากเย็นนัก

ผู้อ่านจะพบได้ว่า ถ้าเริ่มต้นเขียนโปรแกรมในปัจจุบัน โปรแกรมวิชวลเบสิกเป็นโปรแกรมที่นักเขียนโปรแกรมน่าสนใจ เพราะรูปแบบในการเขียนง่ายมาก หนังสือเล่มนี้ จะเหมาะสมกับผู้ที่จะเป็นโปรแกรมเมอร์มือใหม่ หรือบุคคลที่ต้องการเริ่มเขียนโปรแกรมอย่างจริงจัง

แบบฝึกหัดท้ายบท

1. จงอธิบายถึงสภาพแวดล้อมของโปรแกรมวิชวลเบสิก

2. จงอธิบายถึงความแตกต่างของแท็บ Existing กับแท็บ Recent

3. จงอธิบายถึงความสำคัญของทูลบาร์

4. จงอธิบายถึงความสำคัญของทูลบล็อก

5. จงอธิบายถึงความสำคัญของฟอร์ม

6. จงอธิบายถึงความสำคัญของหน้าต่าง Project Explerer

7. จงอธิบายถึงความสำคัญของหน้าต่างคุณสมบัติ

8. จงอธิบายถึงความสำคัญของหน้าต่าง Form Layout

9. จงอธิบายถึงความแตกต่างของโหมด SDI กับโหมด MDI

10. ถ้าจะทำให้โปรแกรมวิชวลเบสิกอ่านภาษาไทยได้ จะต้องทำอย่างไร

11. จงอธิบายคุณสมบัติ (properties) เมท็อด (method) และอิเวนต์ (event)

12. ทำไมถึงต้องมีการทดสอบโปรแกรมที่ออกแบบ ก่อนแจกจ่ายไปยังผู้ใช้งาน