บทที่ 3

ไมโครซอฟต์วิชวลเบสิก

ลักษณะทั่วไปของภาษาวิชวลเบสิก

ภาษาวิชวลเบสิกเป็นภาษา Basic ประเภทหนึ่งที่ได้รับการพัฒนาให้สามารถทำงานภายใต้ระบบวินโดวส์ (Windows) ได้ และมีตัวประสานกับผู้ใช้ในรูปแบบของกราฟิก ในการพัฒนาโปรแกรมสามารถกำหนดชนิดของข้อมูล กำหนดค่าคงที่ และมีการสร้างฟังก์ชันหรือโปรแกรมย่อยที่มีการส่งผ่านค่าพารามิเตอร์ได้ และเมื่อสิ้นสุดการพัฒนาแล้วก็สามารถแปลโปรแกรมเป็นแฟ้มข้อมูลกระทำการที่สามารถทำงานได้ด้วยตัวเอง และเป็นโปรแกรมประยุกต์หนึ่งบนวินโดวส์

ในการพัฒนาโปรแกรมด้วยภาษาวิชวลเบสิกนั้นจะเริ่มจากการกำหนดส่วนประสานงานกับผู้ใช้ก่อน และเนื่องจากเป็นโปรแกรมประยุกต์บนวินโดวส์ Visual Basic ก็จะมีวัตถุ (Object)ต่างๆ เช่น วินโดวส์ กล่องโต้ตอบ (Dialog Box) ปุ่ม (Button) และกล่องรายการ (List Box) เป็นต้น ให้ผู้พัฒนาเลือกนำมาประกอบกันเป็นส่วนประสานงานกับผู้ใช้ในโปรแกรมประยุกต์นั้น ๆ ซึ่งในการทำงานก็สามารถใช้เมาส์ (Mouse) เลื่อนตัวชี้ไปเลือก เคลื่อนย้าย หรือเปลี่ยนขนาดวัตถุได้ตามต้องการ ในโปรแกรมประยุกต์ของวิชวลเบสิก จะประกอบขึ้นจากแฟ้มข้อมูล 3 ประเภท คือ

1. ฟอร์ม (Form)

เป็นแฟ้มข้อมูลที่เก็บส่วนประกอบต่าง ๆ ซึ่งก็คือ วินโดวส์และการควบคุมในวินโดวส์รวมกับ โปรแกรมสำหรับสั่งงานของส่วนประกอบนั้น ซึ่งแฟ้มข้อมูลประเภทนี้อาจจะมีมากกว่าหนึ่งหรือไม่มีก็ได้

2. โคดโมดูล (Code Module)

ทำหน้าที่เก็บเฉพาะกระบวนการคำสั่ง (Procedure) ที่ใช้ร่วมกันในโปรแกรมประยุกต์ ซึ่งแฟ้มข้อมูลนี้อาจจะมีมากกว่าหนึ่งหรือไม่มีเลยก็ได้

3. โกลบอลโมดูล (Global Module)

ทำหน้าที่เก็บตัวแปรส่วนกลาง (Global Variable) และอื่น ๆ ที่จะใช้ร่วมกันในโปรแกรมประยุกต์ ซึ่งแฟ้มข้อมูลประเภทนี้จะมีได้เพียงประเภทเดียวเท่านั้น โดยทั่วไปแล้วโปรแกรมประยุกต์จะต้องประกอบด้วยอย่างน้อย 2 แฟ้มข้อมูล คือ แฟ้มโกลบอลโมดูลและฟอร์ม 1 ฟอร์ม

ดังนั้นวิชวลเบสิกจะมีการพัฒนาและเขียนโปรแกรมแตกต่างจากรูปแบบเดิม กล่าวคือในรูปแบบเดิมจะต้องมีการออกแบบหน้า ระบุตำแหน่งการแสดงผล คิดหาขั้นตอนการทำงาน และอื่น ๆ จากนั้นจึงทำการเขียนโปรแกรม โปรแกรมที่ได้จะอธิบายและสั่งงานคอมพิวเตอร์เป็นลำดับไป แต่ในวิชวลเบสิกจะใช้หลักของภาพและการมองเห็นได้ โดยเริ่มจากออกแบบวินโดวส์หรือในวิชวลเบสิก เรียกว่า "ฟอร์ม" ในฟอร์มจะประกอบไปด้วยวัตถุต่าง ๆ ที่จะทำงานด้วย เช่น ข้อความ กล่องข้อความ (Text Box) สกรอลบอกซ์ (Scroll Box) หรือปุ่ม เมื่อกำหนดสิ่งเหล่านี้ครบตามความต้องการแล้วจึงจะระบุถึงการทำงานของแต่ละวัตถุ โดยเขียนโปรแกรมเข้าไปเชื่อมโยง (Link) กับวัตถุเหล่านี้ สาเหตุที่เป็นเช่นนี้เนื่องจากการทำงานในวินโดวส์เป็นแบบขึ้นกับเหตุการณ์ (Event Driven) การเขียนโปรแกรมแบบเดิมคือ สั่งงานตามลำดับ จะมีความยุ่งยากและซับซ้อน เพราะวินโดวส์มีการทำงานแบบระบบหลายภารกิจ (Multitasking System) ซึ่งสามารถดำเนินงานหลาย ๆ งานได้ในเวลาเดียวกัน จึงต้องใช้รูปแบบการโปรแกรมในลักษณะที่กล่าวมา ซึ่งวัตถุแต่ละชิ้นสามารถมีเหตุการณ์เกิดขึ้นด้วยหลาย ๆ อย่าง และสามารถเขียนโปรแกรมเพื่อควบคุมเหตุการณ์ที่สนใจได้ เช่น กรณีของวัตถุปุ่มที่ต้องการควบคุม เมื่อมีเหตุการณ์คลิก (Click) หรือดับเบิลคลิก (Double Click) ก็สามารถกำหนดวิธีการทำงานต่าง ๆ การทำงานของวัตถุชิ้นนั้น เมื่อมีเหตุการณ์เหล่านี้เกิดขึ้น เหตุการณ์ที่ไม่ได้ระบุก็จะไม่มีผลต่อวัตถุชิ้นนั้น เช่น กล่องข้อความ จะมีชื่อ ข้อความ ความกว้าง ความสูง และสี เป็นต้น โดยที่สามารถอ้างถึงหรือเปลี่ยนคุณสมบัติเหล่านี้ได้ขณะที่โปรแกรมกำลังดำเนินงานอยู่

<u>ความสามารถของภาษาวิชวลเบสิก</u>

ในการสร้างโปรแกรมประยุกต์โดยใช้ภาษาวิชวลเบสิกนั้น ผู้ใช้จะได้ความสามารถต่าง ๆ ที่มีใน Software Development Kit (SDK) เช่น

1. มีวัตถุต่าง ๆ ให้เลือกใช้ เช่น ปุ่ม กล่องรายการ คอมโบบอกซ์ เป็นต้น
2. สามารถสร้างโปรแกรมประยุกต์แบบหลายวินโดวส์ได้
3. ให้ความยืดหยุ่นต่อเหตุการณ์รวมทั้งยอมให้เคลื่อนย้ายวัตถุได้

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

4. แสดงหรือซ่อนวัตถุ
5. ทำงานกับคลิปบอร์ด (Clip Board) ของวินโดวส์
6. เรียกใช้ฟังก์ชันของวินโดวส์ได้โดยตรง
7. ติดต่อกับโปรแกรมประยุกต์ผ่านทาง DDE (Dynamic Data Exchange)

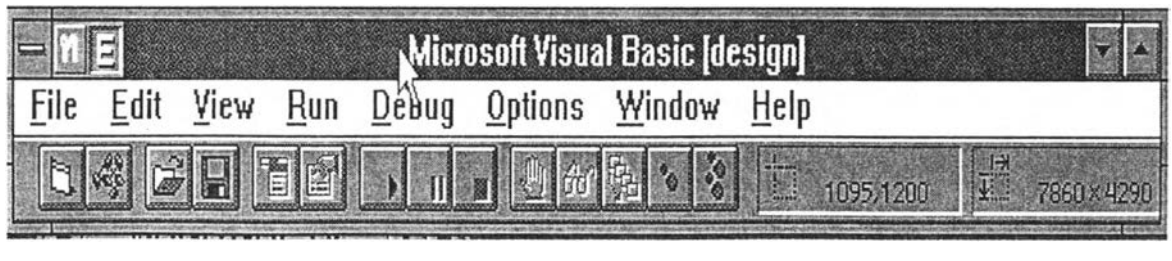
นอกจากนั้นยังขยายขีดความสามารถของภาษาได้โดยการเพิ่มการควบคุมใช้กับวัตถุด้วยการสามารถกำหนดการควบคุมวัตถุนั้นเองได้ นอกเหนือจากที่มีในทูลบอกซ์ (Tool Box) แม้ว่าจะไม่สามารถทำได้โดยตรงกับภาษาวิชวลเบสิกเองก็ตาม แต่ก็ยังมี Microsoft Visual Basic Control Development Kit (CDK) ซึ่งต้องใช้ร่วมกับตัวแปลภาษาของภาษาซีและ SDK เพื่อเพิ่มสัญรูปสำหรับการควบคุมเข้าไปไว้ในทูลบอกซ์ ทำให้ลดขีดจำกัดในการขยายความสามารถของวิชวลเบสิกลงได้

<u>สภาพแวดล้อมในการพัฒนาโปรแกรมของวิชวลเบสิก</u>

หลังจากที่เรียกวิชวลเบสิกขึ้นมาดำเนินงานจะพบว่ามีวินโดวส์เกิดขึ้น 7 วินโดวส์คือ

1. <u>วินโดวส์หลักของวิชวลเบสิก</u>

ใช้สั่งงานด้านต่าง ๆ เช่น เกี่ยวกับการแปลคำสั่ง ดำเนินงาน แฟ้มข้อมูล หรือควบคุมวินโดวส์อื่น ๆ ส่วนบนจะเป็นเมนูเพื่อสั่งงานเหมือนกับโปรแกรมประยุกต์ในวินโดวส์ทั่ว ๆ ไป ดังรูปที่ 3.1 ข้างใต้เมนูคือ แถบเครื่องมือ (Toolbar) ทำให้สั่งงานได้อย่างรวดเร็วโดยเพียงแต่คลิกเมาส์ตรงปุ่มที่ต้องการทางขวามือจะเป็นการแสดงตำแหน่งของวัตถุที่กำลังทำงานด้วยว่าอยู่ ณ ตำแหน่งใดจากมุมบนซ้ายของฟอร์มและมีขนาดเท่าไร ในระบบของวิชวลเบสิกประกอบด้วยวินโดวส์ย่อยหลายอัน ซึ่งสามารถเปิดปิดได้ แต่ถ้าเป็นการปิดวินโดวส์หลักจะเป็นการยกเลิกการทำงานของวิชวลเบสิกไป



รูปที่ 3.1 แสดงวินโดวส์หลักของวิชวลเบสิก

ปุ่มในทูลบาร์มีดังตารางที่ 3.1 นี้

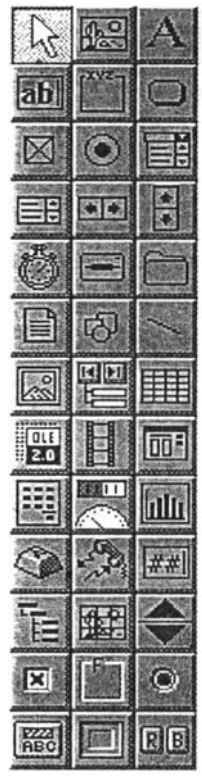
ตารางที่ 3.1 แสดงปุ่มในทูลบาร์

สัญรูป	ชื่อ	รูปแบบการทำงาน
	New Form	สร้างฟอร์มใหม่
	New Module	สร้างโมดูลใหม่
	Open Project	เปิดแฟ้มโปรเจค
	Save Project	บันทึกแฟ้มโปรเจค
	Menu - Design Window	เปิดวินโดวส์เมนูออกแบบ
	Property Window	เปิดวินโดวส์คุณสมบัติ
	Run	สั่งให้โปรแกรมดำเนินงาน
	Break	สั่งให้โปรแกรมหยุดทำงานชั่วขณะ

	End	จบการทำงานของโปรแกรม
	Breakpoint	กำหนดจุดพัก (Breakpoint) ในโปรแกรม
	Instant Watch	ตรวจสอบค่าของตัวแปร หรือนิพจน์
	Calls	แสดงลำดับการเรียกใช้รูทีนย่อยทั้งหมด
	Single Step	ให้ทำงานโปรแกรมทีละคำสั่ง
	Procedure Step	ให้ทำงานโปรแกรมทีละคำสั่ง แต่ไม่เข้าไปในรูทีนย่อย

2. วินโดวส์ทูลบอกซ์ (Toolbox Window)

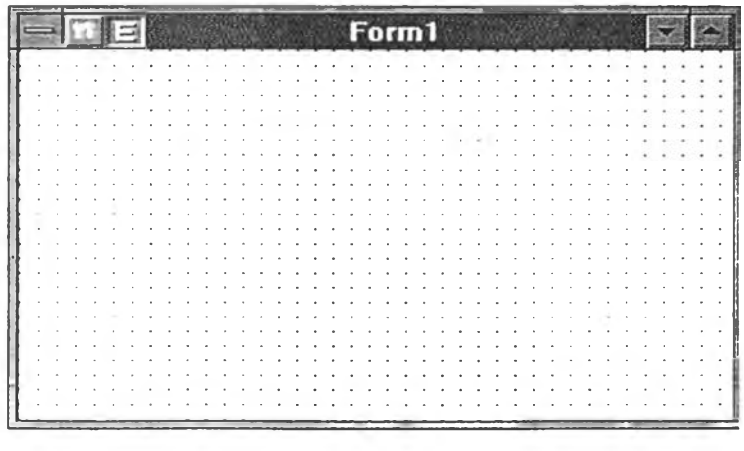
ประกอบด้วยวัตถุหรือตัวควบคุมต่าง ๆ ที่จะนำมาสร้างโปรแกรมประยุกต์ ดังรูปที่ 3.2 ตัวควบคุมเหล่านี้จะมีทั้งตัวควบคุมมาตรฐาน (Standard Control) และตัวควบคุมแบบคัสทอม (Custom Control) ซึ่งเป็นตัวควบคุมที่พัฒนาขึ้นเองโดยผู้ใช้สามารถเพิ่มเข้าไปในวินโดวส์ทูลบอกซ์ โดยใช้คำสั่ง Add File ของรายการเลือก File สำหรับแฟ้มข้อมูลของตัวควบคุมที่พัฒนาขึ้นเอง ซึ่งมีสกุลเป็น .VBX ก็จะปรากฏสัญรูปสำหรับตัวควบคุมนั้น ๆ ในวินโดวส์ทูลบอกซ์



รูปที่ 3.2 แสดงวินโดวส์ทูลบอกซ์

3. วินโดวส์ฟอร์ม (Form Window)

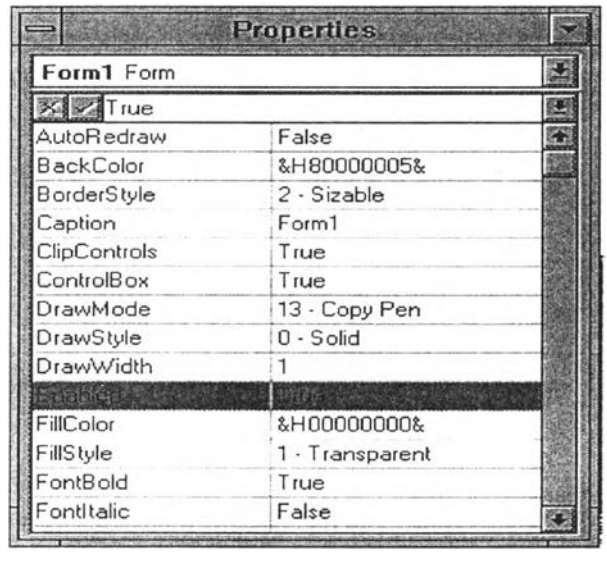
เป็นวินโดวส์สำหรับสร้างองค์ประกอบของโปรแกรมประยุกต์ ดังรูปที่ 3.3 ซึ่งวิชวลเบสิกจะสร้างฟอร์มให้สำหรับบรรจุวัตถุต่าง ๆ ที่ถูกเลือกมาจากทูลบอกซ์



รูปที่ 3.3 แสดงวินโดวส์ฟอร์ม

4. วินโดวส์คุณสมบัติ

วินโดวส์นี้จะแสดงคุณสมบัติทั้งหมดของวัตถุที่ถูกเลือก ถ้าหากวินโดวส์นี้ถูกเปิดอยู่ การคลิกที่วัตถุใดในฟอร์ม จะทำให้คุณสมบัติในวินโดวส์คุณสมบัติเปลี่ยนไปตามวัตถุนั้น (คือไปอ่านคุณสมบัติของวัตถุนั้นมาแสดง) นอกจากการคลิกแล้ว ถ้าเลือกที่ตัววัตถุในช่องบนสุดของวินโดวส์นี้ก็สามารถแสดงรายการของวัตถุขึ้นมาให้เลือกได้ด้วย ในช่องตรงกลางมีไว้กำหนดค่าของคุณสมบัติที่เลือกไว้จากรายการคุณสมบัติในส่วนล่าง ซึ่งในบางคุณสมบัติสามารถแสดงเป็นรายการเลือกได้ โดยคลิกที่ปุ่มรูปลูกศรชี้ลงของช่องกลาง สำหรับส่วนล่างหากมีรายการคุณสมบัติมากกว่าที่จะแสดงได้หมด ก็สามารถใช้แถบเลื่อนในการเลื่อนดูคุณสมบัติต่าง ๆ ดังรูปที่ 3.4



รูปที่ 3.4 วินโดวส์คุณสมบัติแสดงคุณสมบัติของวัตถุ

5. วินโดวส์โปรเจค (Project Window)

เป็นวินโดวส์ที่ใช้ในการควบคุมแฟ้มข้อมูลต่าง ๆ ทั้งหมด ที่จะประกอบกันขึ้นเป็นโปรแกรมประยุกต์หนึ่ง ๆ แฟ้มข้อมูลทั้งหมดที่จะใช้โปรแกรมประยุกต์จะต้องมีอยู่ในวินโดวส์โปรเจคนี้ ซึ่งจะใช้เป็นที่ควบคุมและจัดการแฟ้มข้อมูลต้นฉบับ (Source File) ทั้งหมด ดังรูปที่ 3.5 เช่น แฟ้มที่มีสกุลเป็น .VBX แฟ้มโมดูล หรือแฟ้มแบบฟอร์ม การเพิ่มหรือยกเลิกแฟ้มใดจากโปรเจคจะใช้คำสั่งในเมนู File เช่น การเพิ่มแฟ้มจะทำได้จากหัวข้อ Add File ... และการยกเลิกแฟ้มให้เลือกหัวข้อ Remove File ซึ่งในแฟ้มเหล่านี้จะถูกรวมอยู่ในโปรเจค โดยแบ่งออกได้เป็น 3 ประเภท คือ

1) แฟ้มโมดูล

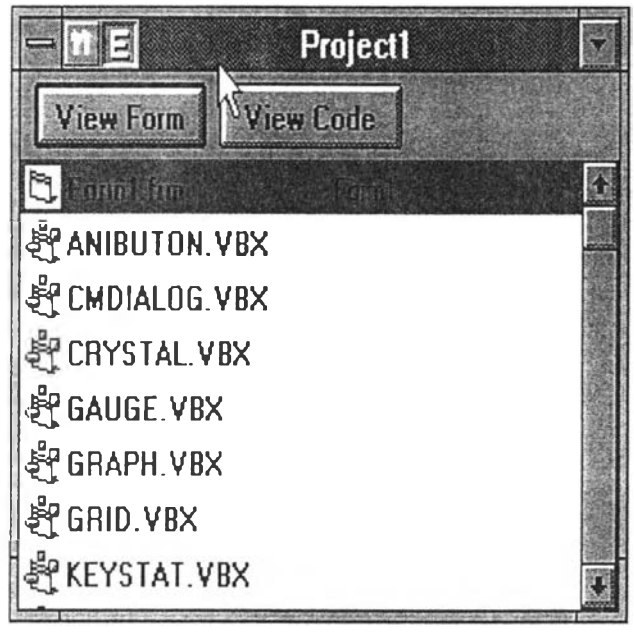
แฟ้มนี้มีนามสกุลเป็น .BAS สำหรับเก็บรูทีนที่จะใช้ร่วมกันในโปรแกรมประยุกต์หนึ่ง ๆ เก็บการประกาศตัวแปรแบบโกลบอลและอื่น ๆ ที่จะใช้ร่วมกัน ซึ่งก็คือรูทีนที่ไม่ขึ้นกับแบบฟอร์มใด ๆ เลย

2) แฟ้มตัวควบคุมแบบคัสทอม

กรณีที่ต้องการใช้ตัวควบคุมแบบคัสทอมอื่นเพิ่มเติม จะต้องมีการเพิ่มแฟ้มที่มีสกุล .VBX ของตัวควบคุมนั้น ๆ เข้าไปในโปรเจคด้วย จึงจะสามารถใช้งานได้

3) แฟ้มแบบฟอร์ม

มีสกุลของแฟ้มเป็น .FRM ซึ่งจะเป็นแฟ้มที่เก็บแต่ละฟอร์มที่ได้ออกแบบไว้แล้ว ในแฟ้มนี้จะรวมคำสั่งและรูทีนของแต่ละวัตถุในฟอร์มไว้ด้วย



รูปที่ 3.5 แสดงวินโดวส์โปรเจค

ปุ่ม View Form และ View Code ในวินโดวส์โปรเจค สำหรับให้แสดงฟอร์มหรือโปรแกรมต้นฉบับของแฟ้มที่เลือกในวินโดวส์นี้ตามลำดับ ถ้าหากเลือกแฟ้มโมดูลแล้วจะเลือกได้เฉพาะปุ่ม View Code เนื่องจากไม่มีฟอร์มอยู่ในโมดูล และถ้าเลือกแฟ้มตัวควบคุมแบบคัสทอมจะไม่สามารถเลือกปุ่มทั้งสองได้เลย เพราะตัวควบคุมแบบคัสทอม เป็นโปรแกรมที่ถูกแปลคำสั่งมาแล้ว จึงไม่สามารถดูโปรแกรมต้นฉบับได้

6. วินโดวส์บรรณาธิกร (Editor Window)

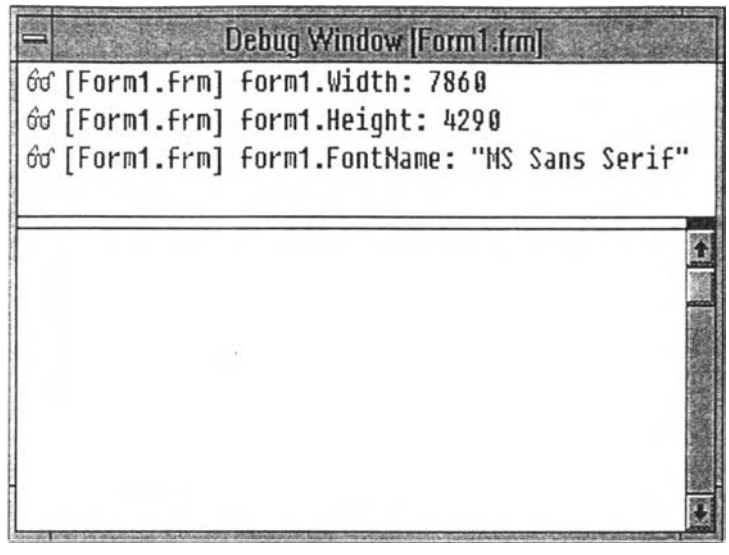
เป็นวินโดวส์สำหรับป้อนโปรแกรม ที่ส่วนบนของวินโดวส์จะมีช่องสำหรับแสดงวัตถุและกระบวนการ (Procedure) โดยที่กระบวนการสามารถเขียนเป็นโปรแกรมหรือรูทีนย่อยประจำเหตุการณ์นั้น ๆ ดังรูป 3.6

รูปที่ 3.6 แสดงวินโดวส์บรรณาธิกร

7. วินโดวส์ตรวจสอบจุดบกพร่อง (Debug Window)

ใช้ในการตรวจสอบค่าของตัวแปรหรือนิพจน์ และทดสอบการทำงานทีละคำสั่ง โดยวินโดวส์นี้จะแสดงเมื่อมีการสั่งดำเนินงานของโปรแกรม ดังรูป 3.7

กล่าวโดยสรุปคือ วัตถุแต่ละชิ้นที่จะนำมาใส่ในฟอร์มนั้น จะถูกเลือกมาจากวินโดวส์ทูลบอกซ์ซึ่งมีส่วนประสานกับผู้ใช้แบบพื้นฐานในระบบวินโดวส์ เช่น ปุ่ม แถบเลื่อน และดังที่ได้กล่าวไปแล้วว่าวัตถุแต่ละชิ้นจะมีคุณสมบัติต่าง ๆ ของตนเอง เช่น ขนาด ตำแหน่งหรือสี การกำหนดคุณสมบัติเหล่านี้สามารถทำได้จากวินโดวส์หลัก โดยวินโดวส์หลักของวิชวลเบสิกจะมีส่วนบนเป็นเมนูเพื่อทำงานด้านแฟ้มข้อมูล การแก้ไข แปลคำสั่งโปรแกรม และจัดการด้านวินโดวส์เหมือนกับโปรแกรมประยุกต์ทั่วไปในวินโดวส์ ในวิชวลเบสิกประกอบด้วยหลาย ๆ วินโดวส์ ซึ่งสามารถเปิด/ปิดได้ แต่การปิดวินโดวส์หลักจะเป็นการเลิกการทำงานของวิชวลเบสิกไปด้วย สำหรับส่วนล่างของวินโดวส์หลักนั้นเรียกว่า "แถบคุณสมบัติ (Properties Bar)" มีไว้กำหนดคุณสมบัติให้กับวัตถุในฟอร์มโดยเมื่อแรกที่สร้างวัตถุขึ้นมาวิชวลเบสิกจะกำหนดคุณสมบัติให้ก่อน ขณะเดียวกันก็สามารถเปลี่ยนแปลงคุณสมบัติของวัตถุได้โดยผ่านทางแถบคุณสมบัตินี้



รูปที่ 3.7 แสดงวินโดวส์ตรวจสอบจุดบกพร่อง

ความหมายของตัวควบคุม

ตัวควบคุมเป็นวินโดวส์ลูก (Child Window) ที่มีลักษณะจำเพาะชนิดหนึ่งที่มีการกำหนดคลาส (Class)ไว้ใช้งานล่วงหน้าอยู่แล้ว ตัวควบคุมเป็นสิ่งอำนวยความสะดวกที่ทำให้การพัฒนาโปรแกรมบนวินโดวส์มีความสะดวกและง่ายยิ่งขึ้น โดยนัยแล้วตัวควบคุมคือปุ่มบังคับชนิดต่าง ๆ เพื่อให้โปรแกรมประยุกต์ต่าง ๆ สามารถติดต่อกับผู้ใช้ได้ด้วยรูปแบบที่ง่าย ๆ ดังนั้นตัวควบคุมจึงทำหน้าที่ในการรับคำสั่งจากผู้ใช้และแสดงผลทางจอภาพ ตัวอย่างของตัวควบคุม เช่น ปุ่ม แถบเลื่อน กล่องรายการ บรรณาธิกร เป็นต้น

การสร้างวินโดวส์ลูกจะต้องมีวิธีเป็นขั้นตอนกล่าวคือจะต้องสร้างคลาสด้วยการลงทะเบียนด้วยฟังก์ชัน RegisterClass ซึ่งคลาสนั้นจะบอกลักษณะจำเพาะของวินโดวส์ เช่น สัญรูป(Icon) สีของวินโดวส์ และกระบวนคำสั่งของวินโดวส์ (Window Procedure) เมื่อลงทะเบียนไปแล้วจะได้ค่าจัดกระทำหรือแฮนเดิล (Handle) ตอบกลับมาเพื่อนำค่าแฮนเดิลนี้ไปสร้างเป็นวินโดวส์ด้วยฟังก์ชัน CreateWindow บานแรก ซึ่งจะเป็นวินโดวส์แม่ (Parent Window) การสร้างวินโดวส์ลูกก็จะทำในลักษณะที่คล้ายคลึงกับวินโดวส์แม่ เพียงแต่ต้องระบุพารามิเตอร์เพิ่มขึ้นเพื่อบอกให้ทราบว่าเป็นวินโดวส์ลูกในขณะที่เรียกใช้ฟังก์ชัน CreateWindow ส่วนรูปแบบในการติดต่อระหว่างวินโดวส์แม่กับตัวควบคุมจะมีลักษณะเหมือนกับวินโดวส์ลูก ทั่ว ๆ ไป กล่าวคือ วินโดวส์แม่สามารถปิดหรือเปิดวินโดวส์ลูกได้ ส่งข้อความไปยังวินโดวส์ลูกได้ ขณะเดียวกันวินโดวส์ลูกสามารถตอบข้อความกลับไปยังวินโดวส์แม่ได้ด้วยเหมือนกัน

ประเภทของตัวควบคุม

ประเภทของตัวควบคุมที่มีในวินโดวส์ สามารถแบ่งออกได้เป็น

1. ตัวควบคุมแบบปุ่มบังคับ (Button Control)

มีลักษณะเป็นวินโดวส์เล็ก ๆ ที่ใช้รับข้อมูลง่าย ๆ จำพวก ใช่/ไม่ใช่ ปิด/เปิด ซึ่งการใช้งานในแต่ละแบบจะแตกต่างกันออกไป โดยทั่วไปแล้วมักจะใช้ตัวควบคุมแบบปุ่มบังคับดังต่อไปนี้

1) ปุ่มกด (Push Button)

เป็นปุ่มบังคับชนิดหนึ่งที่ผู้ใช้จะใช้ทำงานต่าง ๆ โดยเฉพาะเจาะจงในปุ่มกดอันหนึ่ง ๆ จะมีข้อความอธิบายสั้น ๆ ว่าปุ่มกดนั้นใช้ทำอะไร เมื่อผู้ใช้คลิกที่ปุ่มกด จะตอบสนองต่อการสั่งงานของผู้ใช้โดยปฏิบัติการตามทันที

2) ปุ่มกดแบบปริยาย (Default Button)

เป็นปุ่มกดที่เจาะจงสำหรับให้ผู้ใช้ส่งสัญญาณบอกโปรแกรมประยุกต์ เมื่อทำงานที่ต้องการได้เสร็จเรียบร้อยแล้ว

3) ช่องขีด (Check Box)

ช่องขีดได้ถูกสร้างมาสำหรับการเลือกลักษณะการใช้งานต่างๆ ของคำสั่งที่จะกำลังจะสั่งให้โปรแกรมประยุกต์ทำงาน เมื่อนำช่องขีดหลาย ๆ ช่องมาใช้ร่วมกัน ก็สามารถจะให้ผู้ใช้เลือกลักษณะการใช้งานที่ต้องการได้หลาย ๆ ลักษณะ

4) ปุ่มวิทยุ (Radio Button)

ปุ่มวิทยุแต่ละอันจะทำงานคล้าย ๆ กับช่องขีด แต่ว่าการเลือกของปุ่มวิทยุจะเป็นไปในลักษณะที่สามารถเลือกได้เพียงอันหนึ่งอันใดได้เท่านั้น ซึ่งเหมือนกับการเลือกฟังวิทยุได้ทีละช่อง

5) ปุ่มกดวาดเอง (Custom Button)

มีลักษณะเหมือนกับปุ่มกดอื่น ๆ แต่จะแตกต่างที่การแสดงตัวของปุ่มจะต้องให้โปรแกรมประยุกต์เป็นผู้จัดการโดยตลอด ไม่ว่าจะเป็นเมื่อตัวควบคุมปุ่มกดวาดเองนั้นได้โฟกัส (Focus) ถูกปิดทาง (Disable) หรือถูกกดเลือก วินโดวส์เพียงส่งข้อความของตัวควบคุมเป็นสัญญาณให้แก่โปรแกรมประยุกต์เท่านั้น

6) กล่องจัดกลุ่ม (Group Box)

เป็นตัวควบคุมแบบกดชนิดหนึ่ง มีลักษณะเป็นรูปกรอบสี่เหลี่ยมธรรมดาที่จะใช้ล้อมรอบปุ่มกดหรือตัวควบคุมที่มีจำนวนตั้งแต่ 2 ชิ้นขึ้นไป เพื่อบอกแก่วินโดวส์และผู้ใช้ว่าตัวควบคุมเหล่านั้นรวมอยู่เป็นกลุ่มเดียวกัน

2. ตัวควบคุมแบบสถิต (Static Control)

มีลักษณะเป็นวินโดวส์เล็กๆ ที่ใช้แสดงข้อความหรือรูปภาพบางอย่างที่บรรจุไว้เท่านั้น ดังนั้นตัวควบคุมแบบสถิตจึงใช้สำหรับเขียนชื่อให้กับตัวควบคุมอื่นๆ หรือลากกรอบ ลากเส้นตรง หรือระบายสีให้แก่พื้นที่ เพื่อแบ่งกลุ่มของตัวควบคุมให้ผู้ใช้ได้เห็น ตัวควบคุมชนิดนี้จะเหมือนตัวควบคุมแบบกล่องจัดกลุ่มกล่าวคือ ไม่ตอบสนองต่อผู้ใช้ และไม่สร้างสัญญาณหรือข้อความใดๆ แต่จะยังคงสามารถเปลี่ยนการแสดงผลข้อความภายใน รูปภาพ และตำแหน่งของตัวควบคุมได้

3. ตัวควบคุมแบบกล่องรายการ (List Box Control)

มีลักษณะเป็นกล่องบรรจุรายการของบางสิ่งบางอย่างที่ผู้ใช้สามารถเลือกได้ เช่น ภายในกรอบข้อความ open จะมีกล่องรายการบรรจุชื่อแฟ้มข้อมูลสำหรับให้ผู้ใช้เลือกแฟ้มข้อมูลหนึ่งแฟ้มหรือมากกว่าก็ได้ เป็นต้น

4. ตัวควบคุมแบบคอมโบบอกซ์ (Combo Box Control)

เป็นตัวควบคุมที่เกิดจากตัวควบคุมแบบกล่องรายการมาผสมกับตัวควบคุมแบบสถิต หรือตัวควบคุมแบบกล่องรายการมาผสมกับตัวควบคุมแบบบรรณาธิกร ลักษณะการตอบสนองต่อผู้ใช้ก็จะเป็นเช่นเดียวกับตัวควบคุมทั้งสองอย่างทำงานร่วมกัน ต่างกันที่ตัวควบคุมแบบคอมโบบอกซ์จะสามารถซ่อนส่วนที่เป็นกล่องรายการไว้จนกว่าผู้ใช้จะต้องการมาใช้ แต่ในส่วนอื่นๆ แล้วจะมีลักษณะคล้ายกันมาก

5. ตัวควบคุมแบบบรรณาธิกร (Edit Control)

มีลักษณะเป็นวินโดวส์รูปสี่เหลี่ยมที่สำหรับให้ผู้ใช้ใส่และแก้ไขข้อมูลที่เป็นข้อความให้กับโปรแกรมประยุกต์ต่าง ๆ ตัวควบคุมชนิดนี้สามารถใช้แก้ไขข้อความหลาย ๆ บรรทัดรวมทั้งสามารถมีแถบเลื่อน และตอบสนองต่อการใช้งานแถบเลื่อนได้ด้วย ไม่ว่าจะเป็นการเลื่อนข้อความภายในไปทางซ้ายหรือ ขวาขึ้นหรือลง

6. ตัวควบคุมแบบแถบเลื่อน (Scroll Bar Control)

เป็นตัวควบคุมสำหรับติดต่อกับผู้ใช้ในแง่ที่ผู้ใช้จะเลือกใส่ข้อมูลหนึ่งจากข้อมูลที่เป็นช่วงต่อเนื่อง เช่น ค่าความสว่างของจอภาพอาจมีได้ตั้งแต่ 0 1 2 .....255 หรือตำแหน่งของรูปที่จะแสดงผลตัวควบคุมแบบแถบเลื่อนจะทำหน้าที่ในการรับข้อมูลประเภทนี้

โครงสร้างพื้นฐานของตัวควบคุมทั่วไป

จากที่ได้กล่าวมาแล้วว่าตัวควบคุมต้องถูกบรรจุอยู่ในโปรเจคนั้น ๆ จึงจะสามารถเรียกมาใช้งานได้ โดยที่ผู้ใช้สามารถสร้างสำเนาเฉพาะของโปรแกรมที่กำลังดำเนินงานนั้นๆ หรือที่เรียกว่าอินสแตนท์ (Instance) ได้โดยไม่จำกัดจำนวนอินสแตนท์ ทั้งนี้ขึ้นอยู่กับหน่วยความจำที่มีอยู่ ณ ขณะนั้นด้วย ตัวอย่างเช่น แต่ละตัวควบคุมมาตรฐานในวิชวลเบสิกซึ่งจะแสดงเป็นสัญรูปในวินโดวส์ทูลบอกซ์ ยกเว้นสัญรูปตัวชี้จะหมายถึงแต่ละคลาส ดังนั้นผู้ใช้จึงสามารถสร้างกล่องรับข้อความขึ้นมาได้หลายกล่อง โดยที่ทุกกล่องต่างก็เป็นอินสแตนท์ของคลาสกล่องรับข้อความ (ต่อไปขอให้ดูโครงร่างของแฟ้ม SHEET.H และ SHEET.C ในภาคผนวก ข. ประกอบด้วย)

โครงสร้างของตัวควบคุมจะประกอบไปด้วยส่วนที่สำคัญ 2 ส่วน คือ

1. ส่วนตัวแบบ (Control Model)

ส่วนนี้เป็นโครงสร้างของภาษาซีที่กำหนดลักษณะพื้นฐานของตัวควบคุมที่จะพัฒนาเพื่อใช้ในวิชวลเบสิกหรือตัวควบคุมแบบคัสทอม ลักษณะพื้นฐานที่สำคัญของตัวควบคุมแบบคัสทอม (SHEET.H บรรทัดที่ 167-184) เช่น ชื่อโดยปริยายของตัวควบคุม คุณสมบัติของตัวควบคุม และเหตุการณ์ของตัวควบคุม เป็นต้น โดยที่ส่วนตัวแบบจะเก็บตัวชี้ไปยังตาราง 2 ตาราง (SHEET.H บรรทัดที่ 179-180) คือ

1) ตารางข้อมูลคุณสมบัติ (Property Information Table)

เป็นตารางที่เก็บข้อมูลด้านคุณสมบัติทั้งหมดของตัวควบคุมนั้น ๆ (SHEET.H บรรทัดที่ 89-102) เช่น คุณสมบัติด้านความสูง คุณสมบัติด้านความกว้าง เป็นต้น

2) ตารางข้อมูลเหตุการณ์ (Event Information Table)

เป็นตารางที่เก็บข้อมูลด้านเหตุการณ์ทั้งหมดของตัวควบคุมนั้น ๆ (SHEET.H บรรทัดที่ 151-161) เช่น เหตุการณ์คลิก เป็นต้น

2. ส่วนกระบวนงาน (Control Procedure)

ส่วนนี้ (SHEET.C บรรทัดที่ 30-119 ในที่นี้คือ SheetCtlProc) จะคล้ายคลึงกับกระบวนงานของวินโดวส์ในโปรแกรมประยุกต์ของวินโดวส์ โดยจะทำหน้าที่รับข้อความและตอบสนองต่อข้อความนั้น ๆ (SHEET.C บรรทัดที่ 41-117) นอกจากนั้น ยังเป็นตัวกำหนดเหตุการณ์ที่จะเกิดขึ้นอีกด้วย และทำหน้าที่วาดตัวควบคุมบนจอภาพด้วยถ้าตัวควบคุมนั้นไม่ได้เป็นคลาสย่อย (Subclass) ของคลาสตัว ควบคุมของวินโดวส์ (Window Control Class) (SHEET.C บรรทัดที่ 82-94)

เนื่องจากตัวควบคุมแบบคัสทอมนี้มีลักษณะการทำงานแบบคลังเชื่อมโยงแบบพลวัต จึงไม่ต้องมีกระบวนงาน WinMain ในโปรแกรมต้นฉบับ เพราะตัวควบคุมนั้นถูกเชื่อมโยงกับโปรแกรมประยุกต์ที่กำลังดำเนินงานอยู่แล้ว

การลงทะเบียนของตัวควบคุม

เมื่อมีการเรียกใช้ตัวควบคุมแบบคัสทอม จะเกิดลำดับเหตุการณ์ดังนี้คือ

1. วิชวลเบสิกจะสั่งกระทำการกับฟังก์ชันที่ชื่อ VBINITCC ซึ่งควรจะระบุไว้ในแฟ้มของตัวควบคุมแบบคัสทอม (SHEET.C บรรทัดที่ 124-132) เพื่อที่วินโดวส์จะสามารถเรียกใช้งานได้

2. ฟังก์ชัน VBINITCC จะทำการลงทะเบียนส่วนตัวแบบของตัวควบคุมแบบคัสทอม โดยการเรียก ฟังก์ชัน VBRegisterModel พร้อมทั้งส่งผ่านที่อยู่ของโครงสร้างตัวแบบไปด้วย (SHEET.C บรรทัดที่ 131) ขณะเดียวกันโครงสร้าง ของส่วนตัวแบบจะเก็บตัวชี้ไปยังที่อยู่ของส่วนกระบวนงานอีกด้วย (SHEET.H บรรทัดที่ 171) ซึ่งกระบวนงานนี้ก็จะต้องกำหนดให้วินโดวส์สามารถเรียกใช้งานได้เช่นกัน (SHEET.C บรรทัดที่ 30)

3. ผู้พัฒนาสามารถสร้างอินสแตนท์ของตัวควบคุมแบบคัสทอมได้ตามจำนวนที่ต้องการ

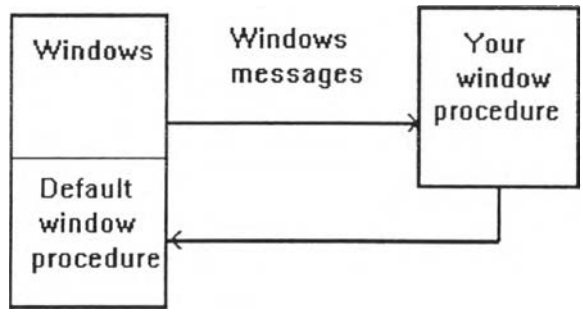
4. เมื่อมีข้อความถูกส่งมายังอินสแตนท์ใด ๆ ของคลาส ส่วนกระบวนการของตัวควบคุมก็จะจัดการกับข้อความนั้น ๆ ในข้อความเหล่านั้น (SHEET.C บรรทัดที่ 30-117) จะมีทั้งพารามิเตอร์ และ แฮนเดิล (SHEET.C บรรทัดที่ 32-36) ที่สามารถระบุได้ว่าเป็นข้อความของอินสแตนท์ตัวไหน

เนื่องจากแฟ้มของตัวควบคุมแบบคัสทอมมีลักษณะเป็น DLL จึงต้องมีรูทีนการกำหนดค่าเริ่มต้น (Initialization routine) ของ DLL ทันทีที่มีการนำแฟ้มของตัวควบคุมเข้าสู่หน่วยความจำในครั้งแรกด้วยฟังก์ชัน LibMain (SHEET.C บรรทัดที่ 146-160) ส่วนฟังก์ชัน VBINITCC อาจจะถูกเรียกใช้ได้หลายครั้ง โดยแต่ละอินสแตนท์ของวิชวลเบสิกที่ใช้แฟ้มตัวควบคุมแบบคัสทอมจะเรียกใช้ฟังก์ชัน VBINITCC เพียง 1 ครั้ง

การส่งข้อความไปยังตัวควบคุมของวิชวลเบสิก

ส่วนใหญ่ในการพัฒนาโปรแกรมในโปรแกรมประยุกต์บนวินโดวส์มักจะเกี่ยวข้องกับการส่ง รับ และประมวลผลข้อความ การเขียนโปรแกรมสำหรับตัวควบคุมแบบคัสทอมมีทั้งลักษณะที่คล้ายคลึงและแตกต่างกับโปรแกรมเหล่านั้น กล่าวคือ ในการเขียนณปรแกรมประยุกต์

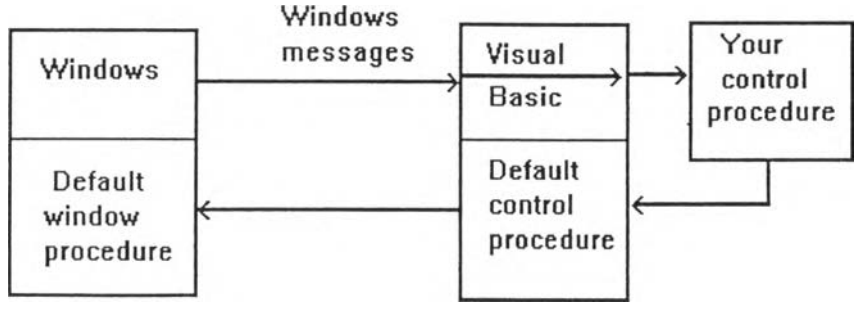
บนวินโดวส์ ตัววินโดวส์จะส่งข้อความไปให้แต่ละโปรแกรมประยุกต์เพื่อนำไปประมวลผล หรือส่งข้อความนั้นไปยังส่วนประมวลผลข้อความโดยปริยาย (Default Message Processor) ดังรูปที่ 3.8



รูปที่ 3.8 แสดงการประมวลผลข้อความในโปรแกรมประยุกต์บนวินโดวส์

ดังนั้นเมื่อมีการสร้างหน้าต่าง การเปลี่ยนแปลงขนาดของหน้าต่าง การเลื่อนเมาส์ และอื่น ๆ ก็จะมีการส่งข้อความไปยังกระบวนงานของวินโดวส์ด้วย

ขณะเดียวกันการเขียนโปรแกรมสำหรับตัวควบคุมแบบคัสทอม เมื่อมีการนำตัวควบคุมเข้ามาในหน่วยความจำ วินโดวส์จะส่งข้อความที่เหมือนกับข้อความที่ส่งให้กับโปรแกรมประยุกต์อื่น ๆ ให้กับตัวควบคุม อย่างไรก็ตาม วิชวลเบสิกจะเป็นผู้รับข้อความนั้นก่อนแล้วจะเลือกว่าจะส่งข้อความนั้นไปให้กับกระบวนงานของตัวควบคุมหรือไม่ โดยที่กระบวนงานของตัวควบคุมอาจจะประมวลผลข้อความนั่นเองหรือส่งต่อไปยังส่วนประมวลผลข้อความโดยปริยายด้วยการเรียกฟังก์ชัน VBDefControlProc (SHEET.C บรรทัดที่ 118) ดังรูปที่ 3.9



รูปที่ 3.9 แสดงการประมวลผลข้อความในวิชวลเบสิก

วิชวลเบสิกจะเป็นตัวจัดการว่าข้อความใดที่ควรส่งไปให้กับกระบวนงานของตัวควบคุม เช่น ข้อความที่เกี่ยวกับเมาส์และคีย์บอร์ดจะไม่ถูกส่งไปในขณะออกแบบ เนื่องจากผู้ใช้มีการใช้คีย์บอร์ดและเมาส์ในการเคลื่อนย้าย เปลี่ยนแปลงขนาด หรือกำหนดคุณสมบัติของตัวควบคุม ข้อความที่วิชวลเบสิกส่งให้กับกระบวนงานของตัวควบคุมจะมีแฮนเดิลของตัวควบคุม (SHEET.C บรรทัดที่ 32) เพื่อใช้เข้าถึงโครงสร้างของอินสแตนท์ของตัวควบคุมนั้น ๆ ซึ่ง โครงสร้างนี้จะใช้เฉพาะในวิชวลเบสิกเท่านั้น

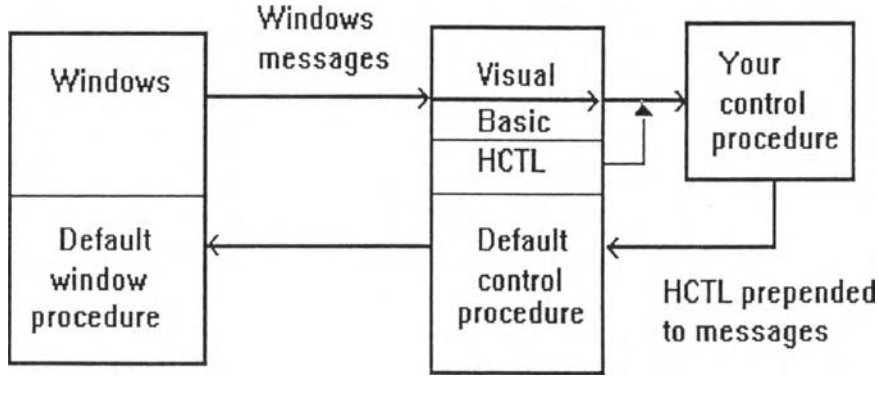
ขณะเดียวกันวิชวลเบสิกก็สามารถสร้างข้อความเพื่อส่งให้กับกระบวนงานของตัวควบคุม เช่น เมื่อมีการกำหนดค่าของคุณสมบัติวิชวลเบสิกจะสร้างและส่งข้อความVBM_SETPROPERTY (SHEET.C บรรทัดที่ 97-102 196-234) ไปยังตัวควบคุมนั้น ๆ ดังรูปที่ 3.10

<u>การอ้างถึงอินสแตนท์ของตัวควบคุม</u>

การอ้างถึงอินสแตนท์ของตัวควบคุมสามารถทำได้ โดยผ่านพารามิเตอร์ดังนี้ คือ

1. HWND

เป็นพารามิเตอร์ที่เป็นแฮนเดิลของวินโดวส์ เพื่อใช้ในการอ้างถึงโครงสร้างของวินโดวส์ ซึ่งมีสารสนเทศเกี่ยวกับวินโดวส์ของตัวควบคุม นอกจากนั้นยังใช้แฮนเดิลตัวนี้ในการติดต่อกับส่วนประสานกับผู้ใช้โปรแกรมประยุกต์ (Application Program Interface : API) ของวินโดวส์ เพื่อจัดการกับตัวควบคุมนั้น ๆ



รูปที่ 3.10 แสดงการประมวลผลข้อความในวิชวลเบสิกด้วยข้อความ VBM_messages

2. HCTL

เป็นพารามิเตอร์ที่เป็นแฮนเดิลของตัวควบคุมเพื่อใช้ในการอ้างถึงโครงสร้างของตัวควบคุมเมื่อมีการสร้างตัวควบคุมวิชวลเบสิกจะสำรองเนื้อที่สำหรับโครงสร้างของตัวควบคุมให้กับแต่ละอินสแตนท์ของตัวควบคุมนั้น (SHEET.H บรรทัดที่ 167-184) โครงสร้างนี้จะประกอบไปด้วยสารสนเทศต่าง ๆ ที่ใช้ในการกำหนดค่าคุณสมบัติของตัวควบคุม โดยที่ API ของวิชวลเบสิกจะต้องใช้โครงสร้างของตัวควบคุมในการจัดการและการกำหนดค่าคุณสมบัติของตัวควบคุมดังนั้นจึงต้องมีการส่งผ่านแฮนเดิลของตัวควบคุมให้กับ API ของวิชวลเบสิกด้วย นอกจากนั้นยังใช้แฮนเดิลของตัวควบคุมในการอ้างถึง (Dereference) โดยใช้ฟังก์ชัน VBDefControlProc (SHEET.C บรรทัดที่ 118) ซึ่งจะคืนค่าตัวชี้ไปยังโครงสร้างของผู้เขียนโปรแกรม (Programmer- defined Structure) ของแต่ละอินสแตนท์ของตัวควบคุมนั้น (SHEET.H บรรทัดที่ 51-62)

<u>การโต้ตอบกับตัวควบคุมของวิชวลเบสิก</u>

วิชวลเบสิกจะทำการโต้ตอบกับตัวควบคุมทั้งในส่วนตัวแบบและส่วนกระบวนงาน โดยเฉพาะการสร้างและการวาดตัวควบคุม การจัดการคุณสมบัติและเหตุการณ์ของตัวควบคุม ดังนี้

1. <u>ด้านการสร้างและการวาดตัวควบคุม</u>

เมื่อมีการสร้างอินสแตนท์ของตัวควบคุม วิชวลเบสิกจะดำเนินการสร้างแฮนเดิลของตัวควบคุมก่อน จากนั้นจึงทำการสร้างโครงสร้างของวินโดวส์ ซึ่งจะทำให้มีการสำรองเนื้อที่ให้กับแฮนเดิลของวินโดวส์ (ยกเว้นกรณีของตัวควบคุมแบบกราฟิก) และทำให้วินโดวส์ทำการส่งข้อความ WM_NCCREATE และ WM_CREATE ให้กับส่วนกระบวนงานของตัวควบคุม (SHEET.C บรรทัดที่ 43-48) และเมื่อสร้างวินโดวส์เสร็จคุณสมบัติต่าง ๆ ของตัวควบคุมก็จะถูกนำเข้าสู่หน่วยความจำ จากนั้นวิชวลเบสิกจะแสดงรูปร่างของตัวควบคุมโดยการใช้ ฟังก์ชัน ShowWindow ซึ่งจะทำให้ส่วนกระบวนงานของตัวควบคุมได้รับข้อความ WM_PAINT (หรือ VBM_PAINT ในกรณีของตัวควบคุมแบบกราฟิก) เมื่อได้รับข้อความ WM_PAINT (SHEET.C บรรทัดที่ 82-94) ก็สามารถจัดรูปแบบของตัวควบคุมได้ตามที่ต้องการซึ่งเหมือนกับโปรแกรมประยุกต์โดยทั่ว ๆ ไปของวินโดวส์ ขณะเดียวกันข้อความ WM_PAINT ไม่ได้ เกี่ยวข้องกับการแสดงสัญรูปของตัวควบคุมบนทูลบอกซ์ แต่ต้องระบุหมายเลขของรูปภาพบิตแมพ (SHEET.H บรรทัดที่ 17-20) ที่จะใช้ในทูลบอกซ์ลงในตัวแบบของตัวควบคุม จากนั้นวิชวลเบสิกจะแสดงภาพบิตแมพนั้นเป็นสัญรูปของตัวควบคุมในทูลบอกซ์ให้เอง

I 17536522

2. ด้านการจัดการเกี่ยวกับคุณสมบัติของตัวควบคุม

วิชวลเบสิกจะจัดการเกี่ยวกับคุณสมบัติต่าง ๆ ของตัวควบคุมนั้นได้ โดยการกำหนดคุณสมบัติที่ต้องการลงในตารางข้อมูลคุณสมบัติ (SHEET.H บรรทัดที่ 89-102) จากนั้นอาจมีชุดคำสั่งที่จัดการเกี่ยวกับคุณสมบัติอยู่ในกระบวนงานของตัวควบคุม (SHEET.C บรรทัดที่ 97-102 196-234) ซึ่งในการกำหนดคุณสมบัติเหล่านั้นควรต้องคำนึงด้วยว่า วิชวลเบสิกจะจัดการกับคุณสมบัติของตัวควบคุมนั้นได้อย่างไรใน 3 ประเด็นหลักต่อไปนี้

1. วิธีการอ่านค่าคุณสมบัติ

2. วิธีการกำหนดค่าคุณสมบัติ

3. วิธีการนำค่าคุณสมบัติเข้าสู่หน่วยความจำและวิธีการบันทึกค่าคุณสมบัติลงในจานบันทึก (Disk) หรือลงในคลิปบอร์ด

โดยที่รูปแบบในการติดต่อกับวิชวลเบสิกของทั้ง 3 ประเด็นดังกล่าว อาจจะอยู่ในรูปแบบของการโอนถ่ายข้อมูลกันโดยตรง รูปแบบของการส่งข้อความ หรือใช้ทั้ง 2 รูปแบบ เช่น ในการกำหนดค่าคุณสมบัติของตัวควบคุมนั้นสามารถกำหนดให้วิชวลเบสิก VBM_SETPROPERTY ทุกครั้งที่มีการกำหนดค่าของคุณสมบัติ เป็นต้น ส่วนการตอบสนองกับข้อความในกรณีที่เลือกให้มีการโอนถ่ายข้อมูลกันโดยตรงไม่ว่าจะเป็นการอ่านและการกำหนดค่าคุณสมบัติ จะต้องกำหนดตำแหน่ง (Offset) ของคุณสมบัตินั้นในโครงสร้างของผู้พัฒนาโปรแกรมด้วย (SHEET.H บรรทัดที่ 86)

ขณะเดียวกัน วิชวลเบสิกได้มีพัฒนาคุณสมบัติไว้จำนวนหนึ่งให้กับตัวควบคุมต่าง ๆ ซึ่งทำให้เขียนชุดคำสั่งในกับคุณสมบัติเหล่านั้นอีกเพียงเล็กน้อย หรือไม่ต้องเขียนเลยก็ได้ ทั้งนี้กระบวนงานควบคุมโดยปริยายของวิชวลเบสิก (Default Control Procedure) จะเป็นตัวจัดการกับคุณสมบัติเหล่านั้นเอง (SHEET.C บรรทัดที่ 118) ซึ่งคุณสมบัติดังกล่าวนี้เรียกว่า "คุณสมบัติมาตรฐาน" (Standard Properties) อย่างไรก็ตามคุณสมบัติเหล่านี้สามารถนำมาใช้ได้ด้วยการกำหนดไว้ในตารางข้อมูลคุณสมบัติ (SHEET.H บรรทัดที่ 91-93)

3. ด้านการจัดการเกี่ยวกับเหตุการณ์ของตัวควบคุม

เหตุการณ์ต่าง ๆ ของตัวควบคุมที่จะใช้จะต้องมีการระบุในตารางข้อมูลเหตุการณ์ (SHEET.H บรรทัดที่ 151-161) เพื่อที่วิชวลเบสิกจะสามารถนำไปดำเนินการได้ กรณีที่ไม่ใช่เหตุการณ์มาตรฐาน (Standard Event) จะต้องเขียนชุดคำสั่งเพื่อจัดการกับเหตุการณ์นั้นไว้ในกระบวนงานของตัวควบคุมด้วย วิธีการให้วิชวลเบสิกรับรู้ว่าได้เกิดเหตุการณ์นั้นขึ้นสามารถทำได้ด้วยการเรียกใช้ฟังก์ชัน VBFireEvent (SHEET.C บรรทัดที่ 228) ซึ่งฟังก์ชันนี้จะทำให้เกิดการ

ทำงานในส่วนของกระบวนงานเหตุการณ์วิชวลเบสิก ขณะเดียวกันเหตุการณ์มาตรฐานของวิชวลเบสิกที่มีอยู่แล้ว กระบวนงานควบคุมโดยปริยายจะเป็นตัวจัดการให้เองโดยที่ไม่ต้องเขียนชุดคำสั่งเพิ่มเติมเพื่อจัดการกับเหตุการณ์เหล่านั้น ดังนั้นจึงสามารถเลือก เหตุการณ์มาตรฐานเหล่านั้นมาใช้กับตัวควบคุมได้ด้วยการระบุไว้ในตารางข้อมูลเหตุการณ์ (SHEET H บรรทัดที่ 153-154) และเมื่อส่วนประมวลผลข้อความได้รับข้อความที่เหมาะสมก็จะทำให้เกิดเหตุการณ์นั้น ๆ อย่างไรก็ตาม เหตุการณ์มาตรฐานเหล่านี้จะเกิดขึ้นก็ต่อเมื่อส่วนประมวลผลข้อความโดยปริยายได้รับข้อความที่เหมาะสมเท่านั้น เช่น ถ้ามีการจัดการกับข้อความของเมาส์เองแล้วคืนค่าทันที เหตุการณ์มาตรฐานคลิก (Click Event) ก็จะไม่เกิดขึ้น เป็นต้น