รายการอ้างอิง

ภาษาไทย

จเร เลิศสุดวิชัย. โครงสร้างข้อมูลพื้นฐานของโปรแกรมสเปรดชีต. ไมโครคอมพิวเตอร์ 44 (พฤศจิกายน 2531): 172-178.

จานจ่าย. อดีตและอนาคตของโปรแกรมสเปรดชีต. ไมโครคอมพิวเตอร์ 60 (กรกฎาคม 2533): 208-216.

จิระ จริงจิตร. Visual Basic เขียนโปรแกรมง่ายเพียงปลายนิ้ว. คอมพิวเตอร์รีวิว 86 (ตุลาคม 2534): 223-228.

_____. การโปรแกรมบนเส้นทาง Visual Basic (ตอน 2). คอมพิวเตอร์รีวิว 95 (กรกฎาคม 2535): 291-296.

ชัยวุฒิ จันมา และ ประเสริฐ ประวัติรุ่งเรือง. โลตัสเวอร์ชัน 2.2. กรุงเทพมหานคร: สำนักพิมพ์ไทยเจริญการพิมพ์, ม.ป.ป.

ชูเกียรติ นาคพิทักษ์กุล. Quattro Pro ผู้มากระทบไหล่ LOTUS. คอมพิวเตอร์รีวิว 80 (เมษายน 2534): 249-258.

ทศเทพ บุญสิลา. 15 เรื่องน่าสนใจในสเปรดชีต. ไมโครคอมพิวเตอร์ 96 (กรกฎาคม 2536): 210-212.

ภาษาอังกฤษ

Lesley Beddie and Scott Raeburn. An Introduction to Computer Integrated Business. London: A. Wheaton and Co. Ltd., 1989.

Microsoft Corperation. Microsoft Visual Basic Programming System for Windows Professional Feature Book 1 Version 3.0 , Microsoft Coperation, 1993.

____. Microsoft Windows 3.1 Guide to Programming. Microsoft Corperation, 1993.

Raymond R. Panko. End User Computing Management, Applications and Technology. New York: John Wiley & Sons,Inc., 1988.

ภาคผนวก

ภาคผนวก ก
คุณสมบัติและเหตุการณ์ของตัวควบคุม

ในส่วนของ รูปแบบ จะมีความหมายของลักษณะอักษรดังนี้

- ตัวอักษรปกติ หมายถึง จะต้องมีคำนั้นอยู่ในรูปแบบเสมอ
- ตัวอักษร *เอียง* ในเครื่องหมาย *[ ]* หมายถึง คำเลือกที่จะมีหรือไม่มีในรูปแบบก็ได้
- form และ CtlName หมายถึง ชื่อของฟอร์มและตัวควบคุมที่ได้กำหนดขึ้นเอง

คุณสมบัติของตัวควบคุม

ตัวควบคุมจะมีทั้งคุณสมบัติมาตรฐาน และคุณสมบัติที่สร้างขึ้นมาใหม่ ดังนี้

1. คุณสมบัติมาตรฐาน ได้แก่

BackColor Property กำหนดสีพื้นหลัง (Background) เมื่อแสดงตัวควบคุม

รูปแบบ *[form.]*CtlName.BackColor *[=Color]*

ค่าที่ใช้เป็นเลขฐานสิบหกจำนวน 6 หลัก ตั้งแต่ &H000000H ถึง &HFFFFFFH เช่น &H000000H หมายถึง สีดำ &HFFFFFFH แทนสีขาว เป็นต้น

ค่าโดยปริยาย ตามค่าของ WINDOW_BACKGROUND ในแฟ้มข้อมูล CONSTANT.TXT

BorderStyle Property กำหนดลักษณะกรอบของตัวควบคุม

รูปแบบ *[form.]*CtlName.BorderStyle *[=numericexpression]*

ตารางที่ ก.1 ค่าที่ใช้ของ BorderStyle Property

ค่าที่ใช้	ลักษณะขอบ
0	ไม่มี
1	เส้นตรง

ค่าโดยปริยาย I

Enabled Property กำหนดให้ตัวควบคุมอยู่ในสภาพใช้งานได้ หรืออยู่ในสภาพใช้งานไม่ได้ กรณีที่ต้องการป้องกันไม่ให้มีการส่งเหตุการณ์ของเมาส์หรือแผงเป็นอักขระ (Keyboard) ไปยังฟอร์ม ก็ต้องกำหนดให้ Enabled Property เป็น False

รูปแบบ *[form.]*CtlName.Enabled *[=booleanexpression]*

ตารางที่ ก.2 ค่าที่ใช้ของ Enabled Property

ค่าที่ใช้	สภาพการใช้งานของตัวควบคุม
True	ใช้ได้
False	ใช้ไม่ได้

ค่าโดยปริยาย True

FontBold Property กำหนดลักษณะของตัวอักขระที่ใช้แสดงในตัวควบคุมว่าต้องการเป็นตัวหนาหรือไม่

รูปแบบ *[form.]*CtlName.FontBold *[=booleanexpression]*

ตารางที่ ก.3 ค่าที่ใช้ของ FontBold Property

ค่าที่ใช้	ความหมาย
True	กำหนดให้ใช้รูปแบบตัวหนาสำหรับตัวอักขระนั้น ๆ
False	กำหนดไม่ให้ใช้รูปแบบตัวหนาสำหรับตัวอักขระนั้น ๆ

ค่าโดยปริยาย True

FontItalic Property กำหนดลักษณะของตัวอักขระที่ใช้แสดงในตัวควบคุมว่าต้องการเป็นตัวเอียงหรือไม่

รูปแบบ *[form.]*CtlName.FontItalic *[=booleanexpression]*

ตารางที่ ก.4 ค่าที่ใช้ของ FontItalic Property

ค่าที่ใช้	ความหมาย
True	กำหนดใช้รูปแบบตัวเอียงสำหรับตัวอักขระนั้น ๆ
False	กำหนดไม่ใช้รูปแบบตัวเอียงสำหรับตัวอักขระนั้น ๆ

ค่าโดยปริยาย False

FontStrikethru Property กำหนดลักษณะของตัวอักขระที่ใช้แสดงในตัวควบคุมว่าต้องการเป็นตัวอักขระที่มีเส้นขีดกลางหรือไม่

รูปแบบ *[form.]*CtlName.FontStrikethru *[=booleanexpression]*

ตารางที่ ก.5 ค่าที่ใช้ของ FontStrikethru Property

ค่าที่ใช้	ความหมาย
True	กำหนดใช้รูปแบบตัวอักขระที่มีเส้นขีดกลาง
False	กำหนดไม่ใช้รูปแบบตัวอักขระที่มีเส้นขีดกลาง

FontUnderline Property กำหนดลักษณะรูแบบตัวอักขระที่ใช้แสดงในตัวควบคุมว่าต้องการเป็นตัวอักขระที่ขีดเส้นใต้หรือไม่

รูปแบบ *[form.]*CtlName.FontUnderline *[=booleanexpression]*

ตารางที่ ก.6 ค่าที่ใช้ของ FontUnderline Property

ค่าที่ใช้	ความหมาย
True	กำหนดใช้รูปแบบตัวอักขระที่ขีดเส้นใต้
False	กำหนดไม่ใช้รูปแบบตัวอักขระที่ขีดเส้นใต้

FontName Property กำหนดลักษณะของตัวอักขระที่ใช้แสดงค่าในตัวควบคุม

รูปแบบ *[form.]*CtlName.FontName *[=Font]*

ค่าที่ใช้เป็นอักขระ เช่น Sheet1.FontName = "Courier"

ค่าโดยปริยาย กำหนดโดยระบบ

FontSize Property กำหนดขนาดของตัวอักขระที่ใช้แสดงในตัวควบคุม

รูปแบบ *[form.]*CtlName.FontSize *[=Point]*

ค่าที่ใช้เป็น Single ค่าสูงสุด คือ 2048 Point (1 Point = 1/ 72 นิ้ว)

ค่าโดยปริยาย กำหนดโดยระบบ

Height Property กำหนดขนาดส่วนสูงของตัวควบคุม หน่วยที่ใช้ขึ้นอยู่กับ ScaleMode Property ของฟอร์มที่ตัวควบคุมนั้นอยู่

รูปแบบ *[form.]*CtlName.Height *[=numericexpression]*

ค่าที่ใช้เป็น Single

Hwnd Property ไมโครซอฟต์วินโดวส์ จะกำหนดแฮนเดิลให้กับตัวควบคุม ซึ่งค่าของ Hwnd นั้น จะใช้เป็นอาร์กิวเมนต์ในการติดต่อกับ API ของวินโดวส์

รูปแบบ *[form.]*CtlName.HWnd

ค่าที่ใช้เป็นเลขจำนวนเต็ม คุณสมบัตินี้ไม่มีในหน้าต่างคุณสมบัติ และเป็นคุณสมบัติที่อ่านค่าได้อย่างเดียว ไม่สามารถเปลี่ยนแปลงค่าได้ในช่วงเวลาดำเนินงาน

Index Property กำหนดตัวควบคุมนั้นเป็นตัวควบคุมแบบแถวลำดับหรือไม่

รูปแบบ *[form.]*CtlName.*[(i).]*Index

ค่าที่ใช้เป็นเลขจำนวนเต็ม ตั้งแต่ 0 ถึง 32,767 เนื่องจากตัวควบคุมแบบแถวลำดับใช้ Name Property เดียวกัน จึงต้องใช้ Index Property ในการอ้างถึงตัวควบคุมแต่ละตัวในแถวลำดับนั้น โดยการเขียนตัวเลขอยู่ในวงเล็บต่อจากชื่อของตัวควบคุม และไม่สามารถเปลี่ยนแปลงค่าของคุณสมบัตินี้ได้ในช่วงเวลาดำเนินงาน

Left Property กำหนดระยะห่างระหว่างขอบซ้ายของตัวควบคุมกับขอบซ้ายของฟอร์ม หน่วยที่ใช้ขึ้นอยู่กับ ScaleMode Property ของฟอร์มที่ตัวควบคุมนั้นอยู่

รูปแบบ *[form.]*CtlName.Left *[=numericexpression]*
ค่าที่ใช้เป็น Single

Name Property กำหนดชื่อที่ใช้แทนตัวควบคุมในการเขียนโปรแกรม
ค่าโดยปริยาย ชื่อของตัวควบคุมแล้วต่อท้ายด้วยเลขจำนวนเต็ม เช่น Sheet1, Sheet2, Sheet3 เรียงตามลำดับที่ตัวควบคุมนั้นถูกสร้างขึ้นมา
ค่าที่ใช้เป็นอักขระ โดยการตั้งชื่อจะต้องขึ้นต้นด้วยตัวอักษร และมีความยาวไม่เกิน 40 ตัวอักษร ซึ่งสามารถประกอบด้วยตัวเลขและเครื่องหมายขีดเส้นใต้ได้ และห้ามใช้เครื่องหมายวรรคตอนอื่นๆ รวมทั้งห้ามเว้นวรรคภายใน Name ของตัวควบคุม

Parent Property หมายถึง ฟอร์มที่ตัวควบคุมนั้นอยู่
รูปแบบ *[form.]*CtlName.Parent
ค่าที่ใช้ เป็นฟอร์ม

สามารถใช้คุณสมบัตินี้ในการเข้าถึงคุณสมบัติหรือวิธีของฟอร์มที่ตัวควบคุมนั้นอยู่ได้ เช่น Sheet1.Parent.Left = 1440 เป็นต้น คุณสมบัตินี้ไม่มีในหน้าต่างคุณสมบัติและไม่สามารถเปลี่ยนแปลงค่าของคุณสมบัตินี้ได้ในช่วงเวลาดำเนินงาน

Top Property กำหนดระยะห่างระหว่างขอบบนของตัวควบคุม กับขอบบนของฟอร์ม หน่วยที่ใช้ขึ้นอยู่กับ ScaleMode Property ของฟอร์มที่ตัวควบคุมนั้นอยู่
รูปแบบ *[form.]*CtlName.Top *[= numericexpression]*
ค่าที่ใช้เป็น Single

Visible Property กำหนดให้แสดงภาพลักษณะของตัวควบคุมในช่วงเวลาดำเนินงานหรือไม่
รูปแบบ *[form.]*CtlName.Visible *[=booleanexpression]*

ตารางที่ ก.7 ค่าที่ใช้ของ Visible Property

ค่าที่ใช้	แสดงลักษณะของตัวควบคุม
True	แสดง
False	ไม่แสดง

Width Property กำหนดขนาดความกว้างของตัวควบคุม หน่วยที่ใช้ขึ้นอยู่กับ ScaleMode Property ของฟอร์มที่ตัวควบคุมนั้นอยู่

รูปแบบ *[form.]*CtlName.Width *[=numericexpression]*

2. คุณสมบัติแบบคัสทอม สามารถแบ่งออกได้เป็น 2 แบบ คือ

2.1) คุณสมบัติที่มีในหน้าค่างคุณสมบัติ

ColHeaderType Property กำหนดประเภทของหัวเรื่องของแถวสดมภ์ที่จะแสดงในตัวควบคุม

รูปแบบ *[form.]*CtlName.ColHeaderType *[=numericexpression]*

ตารางที่ ก.8 ค่าที่ใช้ของ ColHeaderType Property

ค่าที่ใช้	ประเภทของหัวเรื่อง
0	ว่าง
1	ตัวเลข
2	ตัวอักขระ

ค่าโดยปริยาย True

DisplayColHeader Property กำหนดลักษณะการแสดงหัวเรื่องของแถวสดมภ์ในตัวควบคุม

รูปแบบ *[form.]*CtlName.DisplayColHeader *[=booleanexpression]*

ตารางที่ ก.9 ค่าที่ใช้ของ DisplayColHeader Property

ค่าที่ใช้	ลักษณะการแสดงหัวเรื่องของแถวสดมภ์
True	แสดง
False	ไม่แสดง

ค่าโดยปริยาย True

DisplayRowHeader Property กำหนดลักษณะการแสดงหัวเรื่องของแถวนอนในตัวควบคุม

รูปแบบ *[form.]*CtlName.DisplayRowHeader *[=booleanexpression]*

ตารางที่ ก.10 ค่าที่ใช้ของ DisplayRowHeader Property

ค่าที่ใช้	ลักษณะการแสดงหัวเรื่องของแถวนอน
True	แสดง
False	ไม่แสดง

ค่าโดยปริยาย True

GridColor Property กำหนดสีของเส้นแบ่งเซลล์ที่แสดงในตัวควบคุม

รูปแบบ *[form.]*CtlName.GridColor *[=Color]*

ค่าที่ใช้ เป็นเลขฐานสิบหก จำนวน 6 หลัก ตั้งแต่ &H000000H ถึง &HFFFFFFH เช่น &H000000H หมายถึง สีดำ

ค่าโดยปริยาย &H000000H (สีดำ)

GridShowHoriz Property กำหนดลักษณะการแสดงแถบเลื่อนในแนวนอนของตัวควบคุม

รูปแบบ *[form.]*CtlName.GridShowHoriz *[=booleanexpression]*

ตารางที่ ก.11 ค่าที่ใช้ของ GridShowHoriz Property

ค่าที่ใช้	ลักษณะการแสดงแถบเลื่อนในแนวนอน
True	แสดง
False	ไม่แสดง

ค่าโดยปริยาย True

GridShowVert Property กำหนดลักษณะการแสดงแถบเลื่อนในแนวตั้งของตัวควบคุม

รูปแบบ *[form.]*CtlName.GridShowVert *[=booleanexpression]*

ตารางที่ ก.12 ค่าที่ใช้ของ GridShowVert Property

ค่าที่ใช้	ลักษณะการแสดงแถบเลื่อนในแนวตั้ง
True	แสดง
False	ไม่แสดง

ค่าโดยปริยาย True

GridSolid Property กำหนดประเภทของเส้นแบ่งเซลล์ที่แสดงในตัวควบคุม

รูปแบบ *[form.]*CtlName.GridSolid *[=booleanexpression]*

ตารางที่ ก.13 ค่าที่ใช้ของ GridSolid Property

ค่าที่ใช้	ประเภทของเส้นแบ่งเซลล์
True	เส้นทึบ
False	เส้นประ

ค่าโดยปริยาย True

Scrollbars Property กำหนดลักษณะของแถบเลื่อนที่จะแสดงในตัวควบคุม

รูปแบบ *[form.]*CtlName.Scrollbars *[=numericexpression]*

ตารางที่ ก.14 ค่าที่ใช้ของ Scrollbars Property

ค่าที่ใช้	ลักษณะของแถบเลื่อนที่แสดง
0	ไม่แสดง

ตารางที่ ก.14 ค่าที่ใช้ของ Scrollbars Property (ต่อ)

ค่าที่ใช้	ลักษณะของแถบเลื่อนที่แสดง
1	แสดงเฉพาะแนวนอน
2	แสดงเฉพาะแนวตั้ง
3	แสดงทั้งแนวนอนและแนวตั้ง

ค่าโดยปริยาย 3

2.2) คุณสมบัติที่ไม่มีในหน้าต่างคุณสมบัติ เป็นคุณสมบัติที่ใช้ในการเขียนโปรแกรม ซึ่งจะมีผลกับตัวควบคุมในช่วงเวลาดำเนินงานเท่านั้น

Action Property เป็นคุณสมบัติที่ใช้กำหนดลักษณะของการจัดการข้อมูลในตัวควบคุม

รูปแบบ *[form.]*CtlName.Action *[=numericexpression]*

ตารางที่ ก.15 ค่าที่ใช้ของ Action Property

ค่าที่ใช้	ลักษณะของการจัดการ
1	แทรกแถวนอน
2	แทรกแถวสดมภ์
3	ลบแถวนอน
4	ลบแถวสดมภ์
5	คัดลอกข้อมูลในเซลล์
6	เรียงลำดับข้อมูลในแถวนอน
7	เรียงลำดับข้อมูลในแถวสดมภ์

ค่าโดยปริยาย 1

Alignment Property เป็นคุณสมบัติที่ใช้ในการจัดวางตำแหน่งข้อมูลในเซลล์

รูปแบบ *[form.]*CtlName.Alignment *[=numericexpression]*

ตารางที่ ก.16 ค่าที่ใช้ Alignment Property

ค่าที่ใช้	ลักษณะการจัดวางตำแหน่งข้อมูล
0	ชิดซ้าย
1	กึ่งกลาง
2	ชิดขวา

ค่าโดยปริยาย 0 สำหรับข้อมูลชนิดตัวอักษรและสูตร
1 สำหรับข้อมูลชนิดเลขจำนวนเต็ม และจำนวนจริงทศนิยม

ColWidth Property กำหนดขนาดความกว้างของแถวสดมภ์ที่ระบุ หน่วยที่ใช้ขึ้นกับ ScaleMode Property ของฟอร์มที่ตัวควบคุมนั้นอยู่

รูปแบบ *[form.]*CtlName.ColWidth(index) *[=numericexpression]*

ค่าที่ใช้ เป็น Single

โดยที่ค่าของ index เป็นดัชนีของแถวสดมภ์ ซึ่งเป็นเลขจำนวนเต็มตั้งแต่ 0 - 99

CurrentCol Property กำหนดตำแหน่งของเซลล์ปัจจุบันในแถวสดมภ์

รูปแบบ *[form.]*CtlName.CurrentCol *[=numericexpression]*

ค่าที่ใช้ เป็นเลขจำนวนเต็มตั้งแต่ 1 ถึง 99

CurrentRow Property กำหนดตำแหน่งของเซลล์ปัจจุบันในแถวนอน

รูปแบบ *[form.]*CtlName.CurrentRow *[=numericexpression]*

ค่าที่ใช้ เป็นเลขจำนวนเต็มตั้งแต่ 1 ถึง 99

DataType Property เป็นคุณสมบัติที่ใช้กำหนดประเภทของข้อมูลในเซลล์

รูปแบบ *[form.]*CtlName.DataType *[=numericexpression]*

ตารางที่ ก.17 ค่าที่ใช้ของ DataType Property

ค่าที่ใช้	ประเภทของข้อมูล
0	ข้อความ
1	เลขจำนวนเต็ม
2	เลขจำนวนจริงที่มีทศนิม 2 ตำแหน่ง
3	สูตรการคำนวณและผลลัพธ์ที่ได้จากการคำนวณ

DestinationCol Property ใช้กำหนดตำแหน่งจุดหมายปลายทางของแถวสดมภ์ที่ต้องการทำการคัดลอกข้อมูล

รูปแบบ *[form.]*CtlName.DestinationCol *[=numericexpression]*

ค่าที่ใช้ เป็นเลขจำนวนเต็ม ตั้งแต่ 1 ถึง 99

DestinationRow Property ใช้กำหนดตำแหน่งจุดหมายปลายทางของแถวนอนที่ต้องการทำการคัดลอกข้อมูล

รูปแบบ *[form.]*CtlName.DestinationRow *[=numericexpression]*

ค่าที่ใช้ เป็นเลขจำนวนเต็มตั้งแต่ 1 ถึง 99

FloatValue เป็นคุณสมบัติที่กำหนดประเภทของข้อมูลในเซลล์เป็นเลขจำนวนจริงที่มีทศนิยม 2 ตำแหน่ง

รูปแบบ *[form.]*CtlName.FloatValue *[=numericexpression]*

ค่าที่ใช้ เป็นตัวเลขจำนวนจริง ซึ่งมีค่าได้ตั้งแต่ 1.40×10^{-45} ถึง 3.40×10^{38}

Formula Property เป็นคุณสมบัติที่ใช้กำหนดประเภทของข้อมูลให้อยู่ในรูปสูตรของการคำนวณทางคณิตศาสตร์ คือ บวก (+), ลบ (-), คูณ (*) และ หาร (/) โดยที่ไม่มีวงเล็บอยู่ในสูตรการคำนวณ ซึ่งรูปแบบของสูตรนั้นจะประกอบไปด้วย หัวเรื่องแถวสดมภ์ที่เป็นตัวอักขระ และหัวเรื่องแถวนอนที่เป็นตัวเลข เช่น A1 หมายถึง เซลล์ที่อยู่ในแถวสดมภ์ A และแถวนอนที่ 1 , AB20 หมายถึง เซลล์ที่อยู่ในแถวสดมภ์ที่ AB และแถวนอนที่ 20 เป็นต้น ดังนั้นสูตรจึงอาจเป็นการคำนวณที่ได้จากค่าต่าง ๆ ของแต่ละเซลล์ของสูตรที่กำหนดหรือการคำนวณของตัวเลขที่กำหนด เช่น A2+B5*2/3-2 , 3.50*3-1.25 เป็นต้น

รูปแบบ *[form.]*CtlName.Formula *[=stringexpression]*

ค่าที่ใช้ เป็นอักขระและตัวเลข ที่มีความยาวของสูตรไม่เกิน 256 ตัวอักขระ

FResult Property เป็นคุณสมบัติที่สามารถอ่านค่าได้เพียงอย่างเดียวเท่านั้นในขณะดำเนินงาน โดยที่จะทำการอ่านผลลัพธ์ที่ได้จากการคำนวณตามที่กำหนด

รูปแบบ *[form.]*CtlName.FResult

ค่าโดยปริยาย 0

IntValue เป็นคุณสมบัติที่กำหนดประเภทของข้อมูลในเซลล์เป็นเลขจำนวนเต็ม

รูปแบบ *[form.]*CtlName.IntValue *[=numericexpression]*

ค่าที่ใช้ เป็นตัวเลขจำนวนเต็ม ซึ่งมีค่าได้ตั้งแต่ -32,768 ถึง 32,767

Protect Property กำหนดการป้องกันแก้ไขข้อมูลในเซลล์ และยกเลิกการป้องกันแก้ไขข้อมูลในเซลล์

รูปแบบ *[form.]*CtlName.Protect *[=numericexpression]*

ตารางที่ ก.18 ค่าที่ใช้ใน Protect Property

ค่าที่ใช้	ลักษณะการป้องกันการแก้ไขข้อมูลในเซลล์
0	ยกเลิกการป้องกันการแก้ไขข้อมูล
1	ป้องกันการแก้ไขข้อมูล

ค่าโดยปริยาย 0

RowHeight Property กำหนดขนาดความสูงของแถวนอนที่ระบุ หน่วยที่ใช้ขึ้นกับ ScaleMode Property ของฟอร์มที่ตัวควบคุมนั้นอยู่

รูปแบบ *[form.]*CtlName.RowHeight(index) *[=numericexpression]*

ค่าที่ใช้ เป็น Single

โดยที่ค่าของ index เป็นดัชนีของแถวนอน ซึ่งเป็นเลขจำนวนเต็มตั้งแต่ 0 - 99

SelEndCol Property ใช้กำหนดตำแหน่งจุดสิ้นสุดของการเลือกในแถวสดมภ์

รูปแบบ *[form.]*CtlName.SelEndCol *[=numericexpression]*

ค่าที่ใช้ เป็นเลขจำนวนเต็มตั้งแต่ 1 ถึง 99

SelEndRow Property ใช้กำหนดตำแหน่งจุดสิ้นสุดของการเลือกในแถวนอน

รูปแบบ *[form.]*CtlName.SelEndRow *[=numericexpression]*

ค่าที่ใช้ เป็นเลขจำนวนเต็มตั้งแต่ 1 ถึง 99

SelStartCol Property ใช้กำหนดตำแหน่งจุดเริ่มต้นของการเลือกในแถวสดมภ์

รูปแบบ *[form.]*CtlName.SelStartCol *[=numericexpression]*

ค่าที่ใช้ เป็นเลขจำนวนเต็มตั้งแต่ 1 ถึง 99

SelStartRow Property ใช้กำหนดตำแหน่งจุดเริ่มต้นของการเลือกในแถวนอน

รูปแบบ *[form.]*CtlName.SelStartCol *[=numericexpression]*

ค่าที่ใช้ เป็นเลขจำนวนเต็มตั้งแต่ 1 ถึง 99

SortOrder Property ใช้กำหนดลักษณะการเรียงลำดับข้อมูลในช่วงของการเลือกที่กำหนด

รูปแบบ *[form.]*CtlName.SortOrder *[=numericexpression]*

ตารางที่ ก.19 ค่าที่ใช้ของ SortOrder Property

ค่าที่ใช้	ลักษณะการเรียงลำดับข้อมูล
0	การเรียงลำดับขึ้น (Ascending)
1	การเรียงลำดับลง (Descending)

ค่าโดยปริยาย 0

Text Property เป็นคุณสมบัติที่กำหนดประเภทของข้อมูลในเซลล์เป็นแบบข้อความ ซึ่งมีความยาวของข้อมูลได้สูงสุดถึง 256 ตัวอักขระ

รูปแบบ *[form.]*CtlName.Text *[=Stringexpression]*

ค่าที่ใช้ เป็นตัวอักขระ

เหตุการณ์ของตัวควบคุม

ตัวควบคุมจะมีเหตุการณ์ 2 ประเภทคือ

1. เหตุการณ์มาตรฐาน ได้แก่

Click Event เกิดขึ้นเมื่อมีการคลิกเมาส์ที่ตัวควบคุม

รูปแบบ Sub CtlName_Click (*[Index As Integer]*)

อาร์กิวเมนต์ Index เป็นดัชนีของแต่ละตัวควบคุม ในกรณีที่เป็นตัวควบคุมแบบแถวลำดับ หากต้องการทราบด้วยว่า มีการคลิกเมาส์ด้วยปุ่มใด ให้ใช้เหตุการณ์ MouseDown, MouseUp แทน

DblClick Event เกิดขึ้นเมื่อมีการดับเบิ้ลคลิกเมาส์ที่ตัวควบคุม

รูปแบบ Sub CtlName_DblClick (*[Index As Integer]*)

อาร์กิวเมนต์ Index เป็นดัชนีของแต่ละตัวควบคุม

MouseDown, MouseUp Event เกิดขึ้นเมื่อมีการกด (MouseDown) หรือปล่อย (MouseUp) เมาส์ที่ตัวควบคุม

รูปแบบ Sub CtlName_MouseDown (*[Index As Integer,]*Button As Integer, Shift As Integer, X As Single, Y As Single)

Sub CtlName_MouseUp (*[Index As Integer,]*Button As Integer, Shift As Integer, X As Single, Y As Single)

ใช้เหตุการณ์ทั้งสองนี้ควบคุมสิ่งที่จะเกิดขึ้นเมื่อมีการกดหรือปล่อยปุ่มเมาส์ที่ต้องการได้ ซึ่งไม่เหมือนกับเหตุการณ์ Click และ DblClick ที่ไม่สามารถบอกได้ว่ามีการกดเมาส์ด้วยปุ่มใด

ตารางที่ ก.20 ความหมายอาร์กิวเมนต์ของ MouseDown, MouseUp Event

อาร์กิวเมนต์	ความหมาย
Index	เป็นหมายเลขของแต่ละตัวควบคุม ในกรณีที่เป็นตัวควบคุมแบบแถวลำดับ
Button	หมายถึงปุ่มของเมาส์ ถ้าค่าเป็น 1 คือปุ่มซ้าย 2 คือปุ่มขวา 4 คือปุ่มกลาง

Shift	กำหนดโดย Argument Button ค่าของ Shift เป็นในลักษณะของบิตฟิลด์ โดยบิต 0 คือ ปุ่ม Shift บิต 1 คือ ปุ่ม Ctrl และบิต 2 คือ ปุ่ม Alt ปุ่มใดถูกกดบิตนั้นก็จะถูก SET ค่าเป็น 1 ตัวอย่างเช่น ถ้า Shift = 5 ก็หมายความว่า ปุ่ม Shift และ Alt ถูกกดพร้อมกัน เป็นต้น
X, Y	ตำแหน่งปัจจุบันของเมาส์บนหน้าจอ โดยค่าของ X, Y จะเป็นเช่นไรนั้นขึ้นอยู่กับ ScaleHeight, ScaleLeft, and ScaleTop properties ของฟอร์มที่ตัวควบคุมนั้นอยู่

MouseMove Event เกิดขึ้นเมื่อมีการเลื่อนเมาส์บนตัวควบคุม

รูปแบบ Sub CtlName_MouseMove (*[Index As Integer,]*Button As Integer, Shift As Integer, X As Single, Y As Single)

ตารางที่ ก.21 ความหมายอาร์กิวเมนต์ของ MouseMove Event

อาร์กิวเมนต์	ความหมาย
Index	เป็นหมายเลขของแต่ละตัวควบคุม ในกรณีที่เป็นตัวควบคุมแบบแถวลำดับ
Button	ค่าของ Button จะบอกถึงการกดปุ่มเมาส์ โดยเป็นลักษณะของบิตฟิลด์ บิต 0 หมายถึง ปุ่มซ้าย บิต 1 คือ ปุ่มขวา และบิต 2 คือ ปุ่มกลางของเมาส์ ถ้าปุ่มใดถูกกด บิตประจำปุ่มนั้นก็จะถูก SET เป็น 1 เช่นค่าของ Button = 5 ก็หมายความว่า มีการกดทั้งปุ่มซ้ายและปุ่มกลางพร้อม ๆ กัน
Shift	เป็นสถานะของปุ่ม Shift, Ctrl และ Alt ในขณะที่กดปุ่มเมาส์ที่กำหนดโดย Argument Button ค่าของ Shift เป็นลักษณะของบิตฟิลด์ โดยบิต 0 คือ ปุ่ม Shift บิต 1 คือ ปุ่ม Ctrl และ บิต 2 คือ ปุ่ม Alt ปุ่มใดถูกกดบิตนั้นก็จะถูก SET ค่าเป็น 1 ตัวอย่างเช่น ถ้า Shift = 5 ก็หมายความว่า ปุ่ม Shift และ Alt ถูกกดพร้อมกัน เป็นต้น
X, Y	ตำแหน่งปัจจุบันของเมาส์บนหน้าจอ โดยค่าของ X, Y จะเป็นเช่นไรนั้นขึ้นอยู่กับ ScaleHeight, ScaleWiddth, ScaleLeft, and ScaleTop properties ของฟอร์มที่ตัวควบคุมนั้นอยู่

เหตุการณ์ MouseMove จะเกิดขึ้นต่อเนื่องในขณะที่ตัวชี้ตำแหน่งของเมาส์เคลื่อนที่ผ่านตัวควบคุม อาร์กิวเมนต์ Button ของเหตุการณ์ MouseMove แตกต่างจากอาร์กิวเมนต์ Button ของเหตุการณ์ MouseDown, MouseUp โดย Button ของ MouseMove จะสามารถบอกได้ว่ามีการกดเมาส์ปุ่มใดบ้าง แต่ของ MouseDown, MouseUp บอกได้เพียงปุ่มเดียวเท่านั้น

KeyDown , KeyUp Event เกิดขึ้นเมื่อมีการกด (KeyDown) หรือปล่อย (KeyUp) แป้นคีย์บอร์ดที่ตัวควบคุม

รูปแบบ Sub CtlName_KeyDown (*[Index As Integer,]* KeyCode As Integer, Shift As Integer)

Sub CtlName_KeyUp (*[Index As Integer,]* KeyCode As Integer, Shift As Integer)

ตารางที่ ก.22 ความหมายอาร์กิวเมนต์ของ KeyDown และ KeyUp Event

อาร์กิวเมนต์	ความหมาย
Index	เป็นหมายเลขของแต่ละตัวควบคุม ในกรณีที่เป็นตัวควบคุมแบบแถวลำดับ
KeyCode	เป็นรหัสแอสกีสำหรับอักขระที่ผู้ใช้พิมพ์เข้าไป
Shift	เป็นสภานะของปุ่ม Shift, Ctrl และ Alt ในขณะที่กดแป้นคีย์บอร์ดที่กำหนดโดยอาร์กิวเมนต์ KeyCode ค่าของ Shift มีลักษณะเป็นบิตฟิลด์ โดยบิต 0 คือปุ่ม Shift , บิต 1 คือ ปุ่ม Ctrl และบิต 2 คือปุ่ม Alt เมื่อปุ่มใดปุ่มหนึ่งถูกกดบิตนั้นก็จะถูกกำหนดค่าเป็น 1

KeyPress Event เกิดขึ้นเมื่อมีการกดแป้นคีย์บอร์ด

รูปแบบ Sub CtlName_KeyPress (*[Index As Integer,]* KeyAscii As Integer)

อาร์กิวเมนต์ KeyAscii เป็นรหัสแอสกีสำหรับอักขระที่ผู้ใช้พิมพ์เข้าไป

2. เหตุการณ์แบบคัสทอม ได้แก่

ChangeData Event เกิดขึ้นเมื่อเซลล์ข้อมูลมีการเปลี่ยนแปลงค่าด้วยคุณสมบัติ Text, IntValue, FloatValue, Formula และ Action ในช่วงเวลาดำเนินงาน

รูปแบบ Sub CtlName_ChangeData (*[Index As Integer,]* Col As Integer, Row As Integer)

อาร์กิวเมนต์ Col เป็นตำแหน่งแถวสดมภ์ของเซลล์ข้อมูลมีการเปลี่ยนแปลงค่า

อาร์กิวเมนต์ Row เป็นตำแหน่งแถวนอนของเซลล์ข้อมูลมีการเปลี่ยนแปลงค่า

ChangePosition Event เกิดขึ้นเมื่อมีการเปลี่ยนแปลงตำแหน่งปัจจุบันของเซลล์ทั้งในแถวนอนและแถวสดมภ์ ด้วยคุณสมบัติ CurrentCol และ CurrentRow ในช่วงเวลาดำเนินงาน

รูปแบบ Sub CtlName_ChangePosition (*[Index As Integer,]* DestinationCol As Integer, DestinationRow As Integer, CurrentCol As Integer, CurrentRow As Integer)

ตารางที่ ก.23 ความหมายอาร์กิวเมนต์ของ ChangePosition Event

อาร์กิวเมนต์	ความหมาย
DestinationCol	ตำแหน่งจุดหมายปลายทางของแถวสดมภ์
DestinationRow	ตำแหน่งจุดหมายปลายทางของแถวนอน
CurrentCol	ตำแหน่งปัจจุบันของแถวสดมภ์
CurrentRow	ตำแหน่งปัจจุบันของแถวนอน

ColWidthChange Event เกิดขึ้นเมื่อมีการเปลี่ยนแปลงความกว้างของแถวสดมภ์ ด้วยคุณสมบัติ ColWidth ในช่วงเวลาดำเนินงาน

รูปแบบ Sub CtlName_ColWidthChange (*[Index As Integer,]* SelStartCol As Integer, SelEndCol As Integer)

อาร์กิวเมนต์ SelStartCol เป็นจุดเริ่มต้นของการเลือกในแถวสดมภ์

อาร์กิวเมนต์ SelEndCol เป็นจุดสิ้นสุดของการเลือกในแถวสดมภ์

RowHeightChangeEvent เกิดขึ้นเมื่อมีการเปลี่ยนแปลงความสูงของแถวนอน ด้วยคุณสมบัติ RowHeightChange ในช่วงเวลาดำเนินงาน

รูปแบบ Sub CtlName_RowHeightChange (*[Index As Integer,]* SelStartRow As Integer, SelEndRow As Integer)

อาร์กิวเมนต์ SelStartRow เป็นจุดเริ่มต้นของการเลือกในแถวนอน

อาร์กิวเมนต์ SelEndRow เป็นจุดสิ้นสุดของการเลือกในแถวนอน

ภาคผนวก ข

แผ่นแบบโปรแกรมต้นฉบับของตัวควบคุม

```
//----------------------------------------------------------------------
// Sheet.h      TEMPLATE
//----------------------------------------------------------------------
#include "global.h"
//----------------------------------------------------------------------
// Global Defination
//----------------------------------------------------------------------
#define D_MAXROW          100
#define D_MAXCOLUMN       100
        ....
        ....
//----------------------------------------------------------------------
// Resource Information
//----------------------------------------------------------------------
// Toolbox bitmap resource IDs numbers.
//----------------------------------------------------------------------
#define IDBMP_SHEETUP             8000
#define IDBMP_SHEETDOWN           8001
#define IDBMP_SHEETMONO           8003
#define IDBMP_SHEETEGA            8006
//----------------------------------------------------------------------
// Change these for each new VBX file
//----------------------------------------------------------------------
#define VBX_COMPANYNAME      "Microsoft Corporation\0"
#define VBX_FILEDESCRIPTION  "Visual Basic Pix Custom Control Example\0"
....
#define VBX_LEGALCOPYRIGHT   "Copyright \251 Microsoft Corp. 1991-93\0"
#define VBX_ORIGINALFILENAME     "PIX.VBX\0"
#define VBX_PRODUCTNAME      "Microsoft\256 Visual Basic\231 for Windows\231\0"
//----------------------------------------------------------------------
// Update these fields for each build.
//----------------------------------------------------------------------
```

```
35  #define VBX_VERSION             3.00.0.00
36  #define VBX_VERSION_STR     "3.00.0000"
37  #ifndef RC_INVOKED
38  //--------------------------------------------------------------------------
39  // Macro for referencing member of structure
40  //--------------------------------------------------------------------------
41  #define OFFSETIN(struc, field)      ((USHORT)&(((struc *)0)->field))
42  //--------------------------------------------------------------------------
43  // Prototype of Control Procedure
44  //--------------------------------------------------------------------------
45  LONG FAR PASCAL _export SheetCtlProc(HCTL, HWND, USHORT, USHORT, LONG);
46  void near ProcDeleteRow(HCTL hctl, int row);
47  void near SetDeleteRow(HCTL hctl);
48          ....
49          ....
50  //--------------------------------------------------------------------------
51  // Program defined structure (Sheet Structure Defination)
52  //--------------------------------------------------------------------------
53  typedef struct tagSHEET
54  {
55    HFONT   Hfont;
56    COLORREF GridColor;
57    int     CurrentRow;
58    ....
59    ....
60    int    ColWidth[D_MAXCOLUMN];
61    int    RowHeight[D_MAXROW];
62    HLOCAL  pData[D_MAXROW];
63  } SheetStru;
64  typedef SheetStru   FAR * LPSHEET;
65  #define LpSheetDeref(hctl)        ((LPSHEET)VBDerefControl(hctl))
66  //--------------------------------------------------------------------------
67  // Define the consecutive indicies for the properties list
68  //--------------------------------------------------------------------------
```

```
69  PROPINFO Property_Col_Width =
70  {
71    "ColWidth",
72    DT_SHORT | PF_fGetMsg | PF_fSetMsg | PF_fPropArray| PF_fNoShow,
73    0, 0, 0, NULL, 0
74  };
75  PROPINFO ....
76  {
77      .....
78      .....
79  };
80          ....
81          ....
82  PROPINFO Property_Grid_Color =
83  {
84    "GridColor",
85    DT_COLOR | PF_fGetMsg | PF_fSetMsg| PF_fEditable,
86    OFFSETIN(SheetStru, GridColor),
87    0, 0, NULL, 0
88  };
89  PPROPINFO Sheet_Properties[] =
90  {
91  PPROPINFO_STD_CTLNAME,
92  PPROPINFO_STD_INDEX,
93  PPROPINFO_STD_BACKCOLOR,
94          .....
95          .....,
96  &Property_Grid_Color,
97          .....
98          .....
99  &Property_Col_Width,
100         .....
101 NULL
102 };
```

```
103   enum Prop_Index
104   {
105   IPROPINFO_STD_CTLNAME,
106   IPROPINFO_STD_INDEX,
107   IPROPINFO_STD_BACKCOLOR,
108         .....
109         .....
110   IPROPINFO_Grid_Color,
111         .....
112         .....
113   IPROPINFO_Col_Width,
114         .....
115         .....
116   IPROPINFO_END
117   } ;
118   //---------------------------------------------------------------------------
119   // Event info
120   //---------------------------------------------------------------------------
121   WORD Paramtypes_Col_Width_Change[]  = {ET_I2,ET_I2};
122   EVENTINFO Event_Col_Width_Change =
123   {
124   "ColWidthChange",
125   2,
126   4,
127   Paramtypes_Col_Width_Change,
128   "SelStartCol As Integer,SelEndCol As Integer",
129   };
130   WORD ....
131   EVENTINFO ....
132   {
133       .....
134       .....
135   };
136       ....
```

```
137  //------------------------------------------------------------------------
138  // Define the consecutive indicies for the events
139  //------------------------------------------------------------------------
140  enum Event_Index
141  {
142  IEVENTINFO_STD_CLICK,
143  IEVENTINFO_STD_DBLCLICK,
144          .....
145          .....
146  IEVENTINFO_Col_Width_Change,
147          .....
148          .....
149  IEVENTINFO_END
150  } ;
151  PEVENTINFO Sheet_Events[] =
152  {
153  PEVENTINFO_STD_CLICK,
154  PEVENTINFO_STD_DBLCLICK,
155          .....
156          .....
157  &Event_Col_Width_Change,
158          .....
159          .....
160  NULL
161  };
162  //------------------------------------------------------------------------
163  // Model struct
164  //------------------------------------------------------------------------
165  // Define the control model (using the event and property structures).
166  //------------------------------------------------------------------------
167  MODEL modelSheet =
168      {
169      VB_VERSION,                          // VB version being used
170      MODEL_fFocusOk | MODEL_fLoadMsg,     // MODEL flags
```

```
171     (PCTLPROC)SheetCtlProc,                 // Control procedure
172     CS_VREDRAW | CS_HREDRAW,                // Class style
173     WS_BORDER| WS_HSCROLL| WS_VSCROLL,      // Default Windows style
174     sizeof(SheetStru),                      // Size of STD structure
175     IDBMP_SHEETUP,                          // Palette bitmap ID
176     "Sheet",                                // Default control name
177     "SHEET",                                // Visual Basic class name
178     NULL,                                   // Parent class name
179     Sheet_Properties,                       // Property information table
180     Sheet_Events,                           // Event information table
181     IPROPINFO_STD_BACKCOLOR,                // Default property
182     IEVENTINFO_STD_CLICK,                   // Default event
183     NULL                                    // Property representing value of ctl
184     };
185
186 #endif    // RC_INVOKED
187 //------------------ End of Sheet.h Template  ----------------------------
```

```
1   //-------------------------------------------------------------------------
2   // Sheet.c      TEMPLATE
3   //-------------------------------------------------------------------------
4   // Contains control procedure for Sheet control
5   //-------------------------------------------------------------------------
6   #include <windows.h>
7   #include <windowsx.h>
8   #include <vbapi.h>
9   #include <string.h>
10  #include "sheet.h"
11  #include <stdlib.h>
12  #include <stdio.h>
13  #include <malloc.h>
14  //-------------------------------------------------------------------------
15  // Global Variables
16  //-------------------------------------------------------------------------
17  HANDLE hmodDLL;
18  //-------------------------------------------------------------------------
19  // Standard Error Values
20  //-------------------------------------------------------------------------
21  #define ERR_None          0
22  #define ERR_InvPropVal    380    // ErrorS(380) = "Invalid property value"
23  #define ERR_UnableAlloc   999    // ErrorS(999) = "Unable Allocate Memory"
24  #define ERR_Lock          1000
25  #define ERR_Calc          1001
26  #define ERR_Mem           1002
27  //-------------------------------------------------------------------------
28  // Sheet Control Procedure
29  //-------------------------------------------------------------------------
30  LONG FAR PASCAL _export SheetCtlProc
31  (
32      HCTL   hctl,
33      HWND   hWnd,
34      USHORT msg,
```

```
35      USHORT wParam,
36      LONG   lParam
37  )
38  {
39      USHORT Error = 0;
40      LPSHEET LpSheet = LpSheetDeref(hctl);
41      switch (msg)
42      {
43        case WM_NCCREATE:
44            {
45             InitNCCREATE (hctl, hWnd, wParam, lParam);
46             InitMem (hctl, hWnd, &Error);
47            }
48           break;
49        case WM_ACTIVATE:
50            {
51              LPSHEET LpSheet = LpSheetDeref(hctl);
52              SetFocus(hWnd);
53            }
54           break;
55        case WM_LBUTTONDOWN:
56        case WM_LBUTTONDBLCLK:
57             OnLButtonDown(hWnd,hctl,LOWORD(lParam),HIWORD(lParam));
58             break;
59        case WM_SETFONT:
60            {
61           LPSHEET LpSheet = LpSheetDeref(hctl);
62           LpSheet->Hfont = (HFONT)wParam;
63           return 0;
64            }
65        case WM_GETFONT:
66            {
67           LPSHEET LpSheet = LpSheetDeref(hctl);
68           return LpSheet->Hfont;
```

```
69          }
70      case WM_HSCROLL:
71          OnHScroll(hctl, hWnd, wParam, lParam);
72          break;
73      case WM_VSCROLL:
74          OnVScroll(hctl, hWnd, wParam, lParam);
75          break;
76      case WM_CREATE:
77          SetScrollRange(hWnd,SB_VERT,1,D_MAXROW-1,TRUE);
78          SetScrollPos(hWnd,SB_VERT,0,FALSE);
79          SetScrollRange(hWnd,SB_HORZ,1,D_MAXCOLUMN-1,TRUE);
80          SetScrollPos(hWnd,SB_HORZ,0,FALSE);
81          break;
82      case WM_PAINT:
83          {
84          if (wParam)
85            OnPaint(hctl, hWnd, (HDC)wParam);
86          else
87            {
88            PAINTSTRUCT ps;
89            BeginPaint(hWnd, &ps);
90            OnPaint(hctl, hWnd, ps.hdc);
91            EndPaint(hWnd, &ps);
92            }
93          }
94          break;
95      case VBM_GETDEFSIZE:
96          return MAKELONG(200,200);
97      case VBM_SETPROPERTY:
98          {
99          if (SetProperty(hctl, hWnd, wParam, lParam, &Error))
100           return Error;
101         }
102         break;
```

```
103         case VBM_GETPROPERTY:
104             {
105             if (GetProperty(hctl, hWnd, wParam, lParam, &Error))
106               return Error;
107             }
108             break;
109         case WM_NCDESTROY:
110             {
111               int i;
112               for (i=0 ;i < D_MAXROW; i++)
113                {
114                 FreeRow(LpSheet->pData[i]);
115                }
116             }
117     }  // switch
118     return VBDefControlProc(hctl, hWnd, msg, wParam, lParam);
119 }
120 //---------------------------------------------------------------------------
121 // Register custom control. This routine is called by VB when the custom
122 // control DLL is loaded for use.
123 //---------------------------------------------------------------------------
124 int FAR PASCAL _export VBINITCC
125 (
126     USHORT usVersion,
127     int    fRuntime
128 )
129 {
130     // Register Custom Control
131     return VBRegisterModel(hmodDLL, &modelSheet);
132 }
133 //---------------------------------------------------------------------------
134 // Unregister custom control.  This routine is called by VB when the custom
135 // control DLL is being unloaded.
136 //---------------------------------------------------------------------------
```

```
137  VOID FAR PASCAL _export VBTERMCC
138  (  VOID  )
139  {
140      return;
141  }
142  //--------------------------------------------------------------------------
143  // Initialize library.          This routine is called when the first client loads
144  // the DLL.
145  //--------------------------------------------------------------------------
146  int FAR PASCAL LibMain
147  (
148      HANDLE hModule,
149      WORD   wDataSeg,
150      WORD   cbHeapSize,
151      LPSTR  lpszCmdLine
152  )
153  {
154      // Avoid warnings on unused (but required) formal parameters
155      wDataSeg      = wDataSeg;
156      cbHeapSize    = cbHeapSize;
157      lpszCmdLine = lpszCmdLine;
158      hmodDLL = hModule;
159      return 1;
160  }
161  //--------------------------------------------------------------------------
162  // WEP
163  //--------------------------------------------------------------------------
164  // C7 and QCWIN provide default a WEP:
165  //--------------------------------------------------------------------------
166  #if (_MSC_VER < 610)
167  int FAR PASCAL WEP(int fSystemExit);
168  //--------------------------------------------------------------------------
169  // For Windows 3.0 it is recommended that the WEP function reside in a
170  // FIXED code segment and be exported as RESIDENTNAME.  This is
```

```
171    // accomplished using the alloc_text pragma below and the related EXPORTS
172    // and SEGMENTS directives in the .DEF file.
173    //
174    // Read the comments section documenting the WEP function in the Windows
175    // 3.1 SDK "Programmers Reference, Volume 2: Functions" before placing
176    // any additional code in the WEP routine for a Windows 3.0 DLL.
177    //--------------------------------------------------------------------------
178    #pragma alloc_text(WEP_TEXT,WEP)
179    //--------------------------------------------------------------------------
180    // Performs cleanup tasks when the DLL is unloaded.  WEP() is
181    // called automatically by Windows when the DLL is unloaded (no
182    // remaining tasks still have the DLL loaded).  It is strongly
183    // recommended that a DLL have a WEP() function, even if it does
184    // nothing but returns success (1), as in this example.
185    //--------------------------------------------------------------------------
186    int FAR PASCAL WEP
187    (
188        int fSystemExit
189    )
190    {
191        // Avoid warnings on unused (but required) formal parameters
192        fSystemExit = fSystemExit;
193        return 1;
194    }
195    #endif // C6
196    BOOL near SetProperty (HCTL hctl, HWND hWnd, USHORT wParam, LONG lParam, USHORT
197    * Error)
198    {
199        LPSHEET LpSheet = LpSheetDeref(hctl);
200        int nRow,nCol;
201        switch (wParam)
202        {
203        case IPROPINFO_Action:
204            {
```

```
205             SetAction (hctl, hWnd,wParam,lParam);
206             .....
207         case IPROPINFO_Alignment:
208             .....
209         case IPROPINFO_Grid_Show_Horiz:
210             .....
211         case ....
212             .....
213             .....
214
215         case IPROPINFO_Col_Width:
216             {
217             LONG i;
218             LPDATASTRUCT  LpDs = (LPDATASTRUCT)lParam;
219
220             i = LpDs->index[0].data;
221             if (i < 0 || i >= D_MAXCOLUMN)
222                *Error = ERR_BADINDEX;
223             else
224                {
225                LpSheet->ColWidth[i]    = LpDs->data;
226                InvalidateRect (hWnd,NULL,TRUE);
227                }
228             VBFireEvent(hctl,IEVENTINFO_Col_Width_Change,NULL);
229             return TRUE;
230             }
231         default:
232             return FALSE;
233         }
234     }
235     void near SetAction (HCTL hctl,HWND hWnd,USHORT wParam,LONG lParam)
236     {
237         switch ((int) lParam)
238         {
```

```
239
240         case InsertRow :
241             {
242             SetInsertRow(hctl);
243             break;
244             }
245         case InsertCol :
246             {
247             SetInsertColumn(hctl);
248             break;
249             }
250         case DeleteRow :
251             {
252             SetDeleteRow(hctl);
253             break;
254             }
255         case DeleteCol :
256             {
257             SetDeleteColumn(hctl);
258             break;
259             }
260         case Copy :
261             {
262             SetCopyCell(hctl);
263             break;
264             }
265         case SortRow :
266             {
267             SetSortRow(hctl);
268             break;
269             }
270         case SortCol :
271             {
272             SetSortColumn(hctl);
```

```
273             break;
274             }
275         default:
276             break;
277         }
278     }
279     BOOL near GetProperty (HCTL hctl, HWND hWnd, USHORT wParam, LONG lParam, USHORT
280     * Error)
281     {
282         LPSHEET   LpSheet   = LpSheetDeref(hctl);
283         switch (wParam)
284         {
285         case IPROPINFO_Col_Width:
286             {
287             LONG i;
288             LPDATASTRUCT  LpDs = (LPDATASTRUCT)lParam;
289             i = LpDs->index[0].data;
290             if (i < 0 || i >= D_MAXCOLUMN)
291               *Error = ERR_BADINDEX;
292             else
293               {
294               LpSheet->ColWidth[i]    = (LONG)LpDs->data;
295               InvalidateRect (hWnd,NULL,TRUE);
296               }
297             return TRUE;
298             }
299         case ....
300             ....
301             ....
302         case IPROPINFO_Action:
303             *(int *)lParam = LpSheet->Action;
304             return TRUE;
305         default:
306             return FALSE;
```

```
307         }
308     }
309     void near OnPaint(HCTL hctl, HWND hWnd, HDC hDC)
310     {
311         RECT rc;
312         ....
313         GetDrawRect(hctl.hWnd,&rc);
314         ....
315         DrawColTitle(hctl.hWnd,hDC,rc.left,rc.right);
316         DrawRowTitle(hctl,hWnd,hDC,rc.top,rc.bottom);
317         ....
318         DrawGrid(hctl,hDC,rc.left,rc.top,rc.right,rc.bottom);
319         ....
320     }
321     //------------------------- End of Sheet.c TEMPLATE ------------------
```

ประวัติผู้เขียน

นางสาว มาลีรัตน์ มวลประเสริฐ เกิดเมื่อวันที่ 13 มีนาคม พ.ศ. 2509 สำเร็จการศึกษาปริญญาตรีศิลปศาสตรบัณฑิต สาขาสถิติ คณะศิลปศาสตร์ มหาวิยาลัยธรรมศาสตร์ ในปีการศึกษา 2531 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิทยาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมศาสตร์คอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิยาลัย เมื่อพ.ศ .2535