



รายการอ้างอิง

ภาษาไทย

- ทักษิณา สนวนานนท์. 2536. พจนานุกรมศัพท์คอมพิวเตอร์. กรุงเทพมหานคร: ไฮเทค พรินติ้ง จำกัด.
- นิสาชล โตดิเทพย์. 2535. โครงสร้างข้อมูล. กรุงเทพมหานคร: สำนักพิมพ์โอเดียนสโตร์.
- วรชัย เซาว์วิระประสิทธิ์. 2535. Object-Oriented เทคโนโลยีปฏิวัติวงการ. COMPUTER REVIEW 96: 167-172.
- สุชาย ชนวเสถียร และวิชัย จิวงกูร. 2535. โครงสร้างข้อมูลเพื่อการออกแบบโปรแกรมคอมพิวเตอร์. กรุงเทพมหานคร: บริษัท ซีเอ็ดยูเคชั่น จำกัด.
- Petzold, C. 2535. สภาพแวดล้อมเพื่อพัฒนาการโปรแกรมมิง. COMPUTER REVIEW 96: 215-229

ภาษาอังกฤษ

- Cornell, G. 1993. The Visual Basic 3 for Windows Handbook. Berkeley, U.S.A.: Osborne McGraw-Hill.
- Kruse, R.L., Leung, B.P., and Tondo, C.L. 1991. Data Structures and Program Design in C. The Republic of Singapore: PRENTICE-HALL.
- Microsoft Corp. 1993. Microsoft Visual Basic Programming System for Windows , Professional Feature Book 1 Version 3.0. U.S.A.: Microsoft Corp.
- Rector, B.E. 1992. Developing Windows 3.1 Applications with Microsoft C/C++. 2nd ed. Indiana, U.S.A.: Sams Publishing.
- Shiflet, A.B. 1990. Elementary Data Structures with Pascal. St. Paul, U.S.A: West Publishing Company.
- Wilken, P., and Honekamp, D. 1991. Windows System Programming. U.S.A.: Abacus.

ภาคผนวก

ภาคผนวก ก
เพิ่มโปรแกรมของคอนโทรล

คอนโทรลต้นไม้ใบนารี

1. BINARY.H ขนาด 16,240 ไบต์
 2. BINARY.C ขนาด 50,649 ไบต์
 3. BINARY.RC ขนาด 1,431 ไบต์
 4. BINARY.DEF ขนาด 986 ไบต์
 5. BINARY.MAK ขนาด 388 ไบต์
 6. เพิ่มรูปภาพสัญลักษณ์ที่ใช้แทนลักษณะของบัพ 11 เพิ่ม ได้แก่ BOX.ICO , CIRCLE.ICO , CLOCK.ICO, FACE.ICO, LIGHT.ICO , PHONE1.ICO , PHONE2.ICO , PLANE.ICO , ROCKET.ICO , TRAFFIC1.ICO , TRAFFIC2.ICO แต่ละเพิ่มมีขนาด 766 ไบต์
 7. เพิ่มรูปภาพปิดแม่พิมพ์ที่ใช้แทนคอนโทรลในหน้าต่างกล่องเครื่องมือจำนวน 4 เพิ่ม ได้แก่ BIN_EU.BMP ขนาด 470 ไบต์ BIN_MU.BMP , BIN_VU.BMP และ BIN_VD.BMP แต่ละเพิ่มมีขนาดละ 566 ไบต์
- เมื่อแปลแล้วเชื่อมโยงเพิ่มทั้งหมดของคอนโทรลต้นไม้ใบนารีแล้วได้เพิ่ม BINARY.VBX ขนาด 39,936 ไบต์

คอนโทรลต้นไม้ใบนารีค้นหาข้อมูล

1. BISEARCH.H ขนาด 21,519 ไบต์
2. BISEARCH.C ขนาด 90,900 ไบต์
3. BISEARCH.RC ขนาด 2,076 ไบต์
4. BISEARCH.DEF ขนาด 1,020 ไบต์
5. BISEARCH.MAK ขนาด 390 ไบต์
6. เพิ่มรูปภาพสัญลักษณ์ที่ใช้แทนลักษณะของบัพ 11 เพิ่ม และใช้ในการแสดงการเปลี่ยนแปลงภายในต้นไม้อีก 1 เพิ่ม ได้แก่ BOX.ICO , CIRCLE.ICO , CLOCK.ICO, FACE.ICO , LIGHT.ICO , PHONE1.ICO , PHONE2.ICO , PLANE.ICO , ROCKET.ICO , TRAFFIC1.ICO , TRAFFIC2.ICO และ BLANK.ICO แต่ละเพิ่มมีขนาด 766 ไบต์

7. เพิ่มรูปภาพบิตแมพที่ใช้แทนคอนโทรลในหน้าต่างกล่องเครื่องมือจำนวน 4 เพิ่ม ได้แก่ BISCH_EU.BMP ขนาด 470 ไบต์ BISCH_MU.BMP , BISCH_VU.BMP และ BISCH_VD.BMP แต่ละเพิ่มมีขนาดเพิ่มละ 566 ไบต์

เมื่อแปลแล้วเชื่อมโยงเพิ่มทั้งหมดของคอนโทรลต้นไม้ไบนารีค้นหาข้อมูลแล้วได้เพิ่ม BISEARCH.VBX ขนาด 54,272 ไบต์

คอนโทรลต้นไม้ที่มีความสูงสมดุล

1. AVL.H ขนาด 21,412 ไบต์
2. AVL.C ขนาด 94,003 ไบต์
3. AVL.RC ขนาด 2,028 ไบต์
4. AVL.DEF ขนาด 1,010 ไบต์
5. AVL.MAK ขนาด 385 ไบต์
6. เพิ่มรูปภาพสัญลักษณ์ที่ใช้แทนลักษณะของบัพ 11 เพิ่ม และใช้ในการแสดงการเปลี่ยนแปลงภายในต้นไม้อีก 1 เพิ่ม ได้แก่ BOX.ICO , CIRCLE.ICO , CLOCK.ICO , FACE.ICO , LIGHT.ICO , PHONE1.ICO , PHONE2.ICO , PLANE.ICO , ROCKET.ICO , TRAFFIC1.ICO , TRAFFIC2.ICO และ BLANK.ICO แต่ละเพิ่มมีขนาด 766 ไบต์

7. เพิ่มรูปภาพบิตแมพที่ใช้แทนคอนโทรลในหน้าต่างกล่องเครื่องมือจำนวน 4 เพิ่ม ได้แก่ AVL_EU.BMP ขนาด 470 ไบต์ AVL_MU.BMP , AVL_VU.BMP และ AVL_VD.BMP แต่ละเพิ่มมีขนาดละ 566 ไบต์

เมื่อแปลแล้วเชื่อมโยงเพิ่มทั้งหมดของคอนโทรลต้นไม้ที่มีความสูงสมดุลแล้วได้เพิ่ม AVL.VBX ขนาด 56,320 ไบต์

หน้าที่ของเพิ่มสกุลต่าง ๆ นั้นได้กล่าวไว้ในบทที่ 5 หัวข้อการพัฒนาคอนโทรลแล้ว

เพิ่มความช่วยเหลือ (HELP FILE)

1. TREECTL.HPJ ขนาด 876 ไบต์
2. TREECTL.RTF ขนาด 46,898 ไบต์
3. เพิ่มภาพบิตแมพ NODESHAP.BMP ขนาด 44,822 ไบต์ และ NUMBERED.BMP ขนาด 18,374 ไบต์

เมื่อแปลด้วย Microsoft Help Compiler รุ่น 3.1 แล้วได้เพิ่ม TREECTL.HLP ขนาด 41,070 ไบต์

ภาคผนวก ข
กระบวนงานของคอนโทรล

กระบวนงานของคอนโทรลทั้งสามประกอบไปด้วยฟังก์ชันที่สำคัญ ๆ ดังตารางที่ ข.1 โดยเครื่องหมาย ✓ หมายถึง คอนโทรลมีฟังก์ชันนั้น

ตารางที่ ข.1 ฟังก์ชันของคอนโทรลทั้งสาม

ชื่อฟังก์ชัน	ต้นไม้ใบนารี	ต้นไม้ใบนารี ค้นหาข้อมูล	ต้นไม้ที่มี ความสูงสมดุล
1. AnimateHandle ()		✓	✓
2. AppendListOfValues ()		✓	✓
3. AssignValueOk ()		✓	✓
4. BalancedHandle ()			✓
5. CalDistance ()	✓	✓	✓
6. CalFirstPos ()	✓	✓	✓
7. ClickInNode ()		✓	✓
8. CountHeight ()			✓
9. CountLevel ()	✓	✓	✓
10. CreateTree ()		✓	✓
11. DelayTime ()		✓	✓
12. DisplayValue ()	✓	✓	✓
13. DrawNode ()	✓	✓	✓
14. FindMaxLeft ()		✓	✓
15. FindMinRight ()		✓	✓
16. FindPosAnimate ()		✓	✓
17. FindReplacedNode ()		✓	✓
18. FireNodeDecrease ()		✓	✓
19. FireNodeIncrease ()		✓	✓
20. FireTreeChange ()		✓	✓
21. FreeAllNodes ()	✓	✓	✓

ตารางที่ ข.1 (ต่อ) ฟังก์ชันของคอนโทรลทั้งสาม

22. InsertNode()		✓	✓
23. InsertNodeDlg()		✓	✓
24. IsNumberType ()		✓	✓
25. NumberedNode ()	✓	✓	✓
26. PaintTree()	✓	✓	✓
27. ReplaceValue ()		✓	✓
28. ReplaceWithLeftSon ()		✓	✓
29. ReplaceWithRightSon ()		✓	✓
30. ScrollBarHandle ()	✓	✓	✓
31. SetFirstPos ()	✓	✓	✓
32. SetNodeShape ()	✓	✓	✓
33. ShowErrorMsg ()	✓	✓	✓
34. StrToNum ()		✓	✓
35. TreeCtlProc()	✓	✓	✓
36. TreeTravel ()	✓	✓	✓

รายละเอียดของแต่ละฟังก์ชันเป็นดังนี้

1. VOID NEAR **AnimateHandle**(HWND hwnd, HCTL hctl, int IdxToAnimate1, int IdxToAnimate2, int Process, int Step);

- แสดงการเปลี่ยนแปลงของบัพภายในคอนโทรล

- ความหมายของอาร์กิวเมนต์

1. hwnd แชนเดิลของหน้าต่างของคอนโทรล

2. hctl แชนเดิลของคอนโทรล

3. IdxToAnimate1 หมายเลขบัพแรก

4. IdxtoAnimate2 หมายเลขบัพที่สอง

5. Process ตัวบ่งชี้ว่าจะแสดงการเปลี่ยนแปลงแบบใด เช่น เพิ่มบัพ หรือลบบัพ

6. Step ขั้นตอนของการแสดง

2. VOID NEAR **AppendListOfValues**(HCTL hctl);
 - ทำการเพิ่มคุณสมบัติ ListOfKeys เมื่อมีการกำหนดคุณสมบัติ InsertedKey เช่น เดิม ListOfKeys เป็น 12, 40, 30 เมื่อกำหนด InsertedKey เป็น 42 ListOfKeys ก็จะเปลี่ยนเป็น 12, 40, 30, 42
 - ความหมายของอาร์กิวเมนต์
 1. hctl แชนเดิลของคอนโทรล

3. BOOL NEAR **AssignValueOk**(HCTL hctl, LPSTR lpstrAV, LONG Idx);
 - ตรวจสอบความถูกต้องของการกำหนดค่า (อักขระ) ใหม่ของแต่ละบัพด้วย Key Property
 - ความหมายของอาร์กิวเมนต์
 1. hctl แชนเดิลของคอนโทรล
 2. lpstrAV ค่า (อักขระ) ที่ต้องการเปลี่ยนแปลง
 3. Idx หมายเลขบัพ

4. VOID NEAR **BalancedHandle**(HCTL hctl, HWND hwnd)
 - ตรวจสอบความสูงของแต่ละบัพในคอนโทรล หากต้นไม่เกิดไม่สมดุลขึ้นก็จะจัดการทำให้ต้นไม้ นั้นเป็นต้นไม้ที่สมดุลอยู่เสมอ
 - ความหมายของอาร์กิวเมนต์
 1. hctl แชนเดิลของคอนโทรล
 2. hwnd แชนเดิลของหน้าต่างของคอนโทรล

5. SHORT NEAR **CalDistance**(HCTL hctl, int Level, int WantRight);
 - คำนวณตำแหน่งจริงที่จะวาดแต่ละบัพ โดยใช้ค่าของคุณสมบัติ LevelDistance และ NodeDistance ในการคำนวณด้วย
 - ความหมายของอาร์กิวเมนต์
 1. hctl แชนเดิลของคอนโทรล
 2. Level ระดับของบัพนั้น
 3. WantRight ตัวบ่งชี้ กำหนดว่าต้องการระยะของลูกทางซ้ายหรือลูกทางขวา

6. VOID NEAR **CalFirstPos**(HCTL hctl, int Idx, int Level, int PofX, int PofY);
 - คำนวณตำแหน่งเริ่มต้นเพื่อใช้ในการวาดตัวคอนโทรลใหม่
 - ความหมายของอาร์กิวเมนต์
 1. hctl แชนเดิลของค

2. Idx หมายเลขบัพ
3. Level ระดับของบัพนั้น
4. PofX ตำแหน่งในแนวนอนของบัพ หน่วยเป็น Pixel
5. PofY ตำแหน่งในแนวตั้งของบัพ หน่วยเป็น Pixel

7. BOOL NEAR **ClickInNode**(HCTL hctl, int Idx, SHORT PofX, SHORT PofY, int CheckX, int CheckY, int Level);

- ทำการตรวจสอบว่า ในขณะที่ดำเนินงานมีการคลิกเมาส์ที่บัพใด ๆ ในคอนโทรลหรือไม่
- ความหมายของอาร์กิวเมนต์

1. hctl แชนเดิลของคอนโทรล
2. Idx หมายเลขบัพ
3. PofX ตำแหน่งในแนวนอนของการคลิกเมาส์ หน่วยเป็น Pixel
4. PofY ตำแหน่งในแนวตั้งของการคลิกเมาส์ หน่วยเป็น Pixel
5. CheckX ตำแหน่งของบัพในแนวนอน หน่วยเป็น Pixel
6. CheckY ตำแหน่งของบัพในแนวตั้ง หน่วยเป็น Pixel
7. Level ระดับของบัพนั้น

8. VOID NEAR **CountHeight**(HCTL hctl, int Idx, int HeightNow)

- หาคความสูงของต้นไม้ย่อยทางซ้าย หรือทางขวาของบัพใด ๆ
 - ความหมายของอาร์กิวเมนต์
1. hctl แชนเดิลของคอนโทรล
 2. Idx หมายเลขบัพ
 3. HeightNow ความสูงขณะนั้น

```
VOID NEAR CountHeight(HCTL hctl, int Idx, int HeightNow)
{
    LPAVL lpavi = lpavIDEREF(hctl);
    if (Idx > MAXNODE)
        return;
    if (lpavi->OccupiedArr[Idx]) {
        HeightNow++;
        if (HeightNow > lpavi->HeightOfNode)
            lpavi->HeightOfNode = HeightNow;
        CountHeight(hctl, Idx * 2, HeightNow);
        CountHeight(hctl, Idx * 2 + 1, HeightNow);
    }
} // CountHeight
```

เป็นฟังก์ชันเรียกตัวเอง ผลลัพธ์ที่ได้จะเป็นความสูงของบัพพ่อของ Idx ค่าแรกที่เป็นพารามิเตอร์

ของฟังก์ชันนี้



9. VOID NEAR **CountLevel**(HCTL hctl, int Idx, int Level);

- ฟังก์ชันนี้จะถูกเรียกใช้เมื่อต้องการทำงานวนระดับ (สูงสุด) ของต้นไม้ในขณะนั้น
- ความหมายของอาร์กิวเมนต์
 1. hctl แชนเดิลของคอนโทรล
 2. Idx หมายเลขบัพ
 3. Level ระดับของบัพนั้น

10. SHORT NEAR **CreateTree**(HCTL hctl);

- ทำการสร้างต้นไม้ขึ้นใหม่ โดยการนำแต่ละค่าจากคุณสมบัติ ListOfKeys ไปเก็บในช่องลำดับของอักขระที่ปิดท้ายด้วย "\0" ของคอนโทรล
- ความหมายของอาร์กิวเมนต์
 1. hctl แชนเดิลของคอนโทรล

11. VOID NEAR **DelayTime**(HCTL hctl, int Interval);

- ทำหน้าที่หน่วงเวลา ในการแสดงการเปลี่ยนแปลงของบัพภายในคอนโทรล
- ความหมายของอาร์กิวเมนต์
 1. hctl แชนเดิลของคอนโทรล
 2. Interval ช่วงเวลาที่ใช้หน่วง

12. VOID NEAR **DisplayValue**(HWND hwnd, HCTL hctl, HDC hdc, int PofX, int PofY, LPSTR lpstr, BOOL WantBrush);

- แสดงค่าของแต่ละบัพ ซึ่งตำแหน่งที่ใช้แสดงนั้น ก็กำหนดโดยคุณสมบัติ ValuePosition
- ความหมายของอาร์กิวเมนต์
 1. hwnd แชนเดิลของหน้าต่างของคอนโทรล
 2. hctl แชนเดิลของคอนโทรล
 3. hdc เนื้อหาอุปกรณ์
 4. PofX ตำแหน่งในแนวนอน หน่วยเป็น Pixel
 5. PofY ตำแหน่งในแนวตั้ง หน่วยเป็น Pixel
 7. lpstr ค่าที่จะใช้แสดง
 8. WantBrush ตัวบ่งชี้ว่า ต้องการแสดงโดยใช้สีพื้นหลัง ซึ่งกำหนดโดยคุณสมบัติ

BackColor ด้วยหรือไม่

13. VOID NEAR **DrawNode**(HWND hwnd, HCTL hctl, HDC hdc, HWND hIcon, int Idx, int PofX, int PofY, int Level);

- ใช้เส้นเคิลจากฟังก์ชัน SetNodeShape () มาวาดเป็นลักษณะของบัพ
- ความหมายของอาร์กิวเมนต์
 1. hwnd เส้นเคิลของหน้าต่างของคอนโทรล
 2. hctl เส้นเคิลของคอนโทรล
 3. hdc เนื้อหาอุปกรณ์
 4. hIcon เส้นเคิลของสัญลักษณ์
 5. Idx หมายเลขบัพที่จะวาด
 6. PofX ตำแหน่งในแนวนอนของบัพ หน่วยเป็น Pixel
 7. PofY ตำแหน่งในแนวตั้งของบัพ หน่วยเป็น Pixel
 8. Level ระดับของบัพที่จะวาด

14. SHORT NEAR **FindMaxLeft**(HCTL hctl, LONG Idx)

- เป็นฟังก์ชันเรียกตัวเอง ที่ใช้ค้นหาเพื่อนบ้านถัดไปแบบตามลำดับของบัพ Idx
- ความหมายของอาร์กิวเมนต์
 1. hctl เส้นเคิลของคอนโทรล
 2. Idx หมายเลขบัพที่ต้องการหาเพื่อนบ้าน

SHORT NEAR FindMaxLeft(HCTL hctl, LONG Idx)

```
{
    LPAVL lpavl = lpavIDEREF(hctl);

    if (Idx * 2 + 1 <= MAXNODE) {
        if (lpavl->OccupiedArr[Idx * 2 + 1])
            return FindMaxLeft(hctl, Idx * 2 + 1);
    }
    else
        return FALSE;

    return Idx;
} // FindMaxLeft0
```

15. SHORT NEAR **FindMinRight**(HCTL hctl, LONG Idx)

- ใช้ค้นหาว่าบัพ Idx มีลูกทางซ้ายหรือไม่
- ความหมายของอาร์กิวเมนต์
 1. hctl เส้นเคิลของคอนโทรล
 2. Idx หมายเลขบัพที่ต้องการหาว่ามีลูกทางซ้ายหรือไม่

```

SHORT NEAR FindMinRight(HCTL hctl, LONG idx)
{
    LPAVL lpavl = LpavlDEREF(hctl);

    if (idx * 2 <= MAXNODE) {
        if (lpavl->OccupiedArr[idx * 2])
            return FindMinRight(hctl, idx * 2);
    }
    else
        return FALSE;

    return idx;
} // FindMinRight0

```

16. BOOL NEAR **FindPosAnimate**(HCTL hctl, int Idx, int PofX, int PofY, int IdxToFind, int Level);

- ค้นหาตำแหน่งของบัพที่ใช้ในการแสดงการเปลี่ยนแปลง
- ความหมายของอาร์กิวเมนต์
 1. hctl แชนเดิลของคอนโทรล
 2. Idx หมายเลขบัพ
 3. PofX ตำแหน่งของบัพในแนวนอน หน่วยเป็น Pixel
 4. PofY ตำแหน่งของบัพในแนวตั้ง หน่วยเป็น Pixel
 5. IdxToFind หมายเลขบัพที่ต้องการค้นหา
 6. Level ระดับของบัพ Idx

17. SHORT NEAR **FindReplacedNode**(HCTL hctl, int Idx, int FindGT);

- หาบัพที่จะมาแทนที่บัพที่จะถูกลบออกไป
- ความหมายของอาร์กิวเมนต์
 1. hctl แชนเดิลของคอนโทรล
 2. Idx หมายเลขบัพ
 3. FindGT ตัวบ่งชี้ว่าจะให้หาจากต้นไม้อย่างน้อยทางซ้ายหรือทางขวา (GT - Greater Than หมายถึงการหาจากต้นไม้อย่างน้อยทางขวา)

18. VOID NEAR **FireNodeDecrease**(HCTL hctl, int Idx);

- ทำให้เกิดเหตุการณ์ NodeDecrease ขึ้น พร้อมทั้งส่งค่าอาร์กิวเมนต์ของเหตุการณ์ให้กับกระบวนการเหตุการณ์ NodeDecrease ด้วย (ดูรายละเอียดเพิ่มเติมของเหตุการณ์ NodeDecrease ได้จาก ภาคผนวก ค)

- ความหมายของอาร์กิวเมนต์
 1. hctl แชนเดิลของคอนโทรล
 2. Idx หมายเลขบัพที่ถูกลบออกไป

19. VOID NEAR **FireNodeIncrease**(HCTL hctl, int Idx);

- ทำให้เกิดเหตุการณ์ NodeIncrease ขึ้น พร้อมทั้งส่งค่าอาร์กิวเมนต์ของเหตุการณ์ให้กับกระบวนการเหตุการณ์ NodeIncrease ด้วย (ดูรายละเอียดเพิ่มเติมของเหตุการณ์ NodeIncrease ได้จากภาคผนวก ค)

- ความหมายของอาร์กิวเมนต์
 1. hctl แชนเดิลของคอนโทรล
 2. Idx หมายเลขบัพที่เพิ่มเข้ามาในคอนโทรล

20. VOID NEAR **FireTreeChange**(HCTL hctl, int Type, int Idx);

- ทำให้เกิดเหตุการณ์ TreeChange ขึ้น พร้อมทั้งส่งค่าอาร์กิวเมนต์ของเหตุการณ์ให้กับกระบวนการเหตุการณ์ TreeChange ด้วย (ดูรายละเอียดเพิ่มเติมของเหตุการณ์ TreeChange ได้จากภาคผนวก ค)

- ความหมายของอาร์กิวเมนต์
 1. hctl แชนเดิลของคอนโทรล
 2. Type ชนิดของการเปลี่ยนแปลงจำนวนบัพในคอนโทรล ถ้า Type = 0 คือการลบบัพ Type = 1 คือการเพิ่มบัพ
 3. Idx หมายเลขบัพที่เปลี่ยนแปลงนั้น

21. VOID NEAR **FreeAllNodes**(HCTL hctl);

- ทำหน้าที่ลบบัพทั้งหมดออกจากคอนโทรล เช่น เมื่อผู้พัฒนาโปรแกรมด้วยวิซวลเบสิกใช้วิธี Clear กับคอนโทรล (ดูรายละเอียดของวิธี Clear ได้จากภาคผนวก ค)

- ความหมายของอาร์กิวเมนต์
 1. hctl แชนเดิลของคอนโทรล

22. SHORT NEAR **InsertNode**(HCTL hctl, int Idx, LPSTR TmpValue, BOOL InsertProcess);

- เพิ่มหรือค้นหาข้อมูลในช่องลำดับของอักขระที่ปิดท้ายด้วย "\0" ของคอนโทรล

- ความหมายของอาร์กิวเมนต์
 1. hctl แชนเดิลของคอนโทรล
 2. Idx หมายเลขบัพ
 3. TmpValue ค่าที่ต้องการเพิ่มหรือค้นหาจากคอนโทรล
 4. InsertProcess ตัวยบ่งชี้ว่า เป็นการเพิ่มหรือค้นหาบัพ

23. **BOOL FAR PASCAL InsertNodeDlg**(HWND hDlg, unsigned message, WORD wParam, LONG lParam);

- ควบคุมการทำงานของกรอบข้อความที่ใช้ในการเพิ่มบัพให้กับคอนโทรล
- ความหมายของอาร์กิวเมนต์
 1. hDlg แฮนเดิลของกรอบข้อความ
 2. message ข้อความที่ส่งมายังกรอบข้อความ
 3. wParam ข้อมูลเพิ่มเติมของข้อความ
 4. lParam ข้อมูลเพิ่มเติมของข้อความ

24. **SHORT NEAR IsNumberType**(LPSTR lpstr);

- เมื่อได้กำหนดคุณสมบัติ ValueType ให้เป็น 0 (เลขจำนวนเต็ม) ก็จะมีการเรียกใช้ฟังก์ชันนี้ เพื่อตรวจสอบค่าของบัพใหม่ว่า เป็นจำนวนเต็มหรือไม่ (ดูรายละเอียดเพิ่มเติมของคุณสมบัติ ValueType ได้จากภาคผนวก ค)

- ความหมายของอาร์กิวเมนต์
 1. lpstr ค่า (อักขระ) ที่ต้องการตรวจสอบ

25. **VOID NEAR NumberedNode**(HCTL hctl, HDC hdc, int Idx, int PofX, int PofY);

- แสดงหมายเลขประจำบัพที่บริเวณใต้บัพ เมื่อมีการกำหนดคุณสมบัติ NumberedNode เป็น True และในคอนโทรลต้นไม้ที่มีความสูงสมดุลง ฟังก์ชันนี้จะทำหน้าที่แสดงน้ำหนักของแต่ละบัพด้วย เมื่อมีการกำหนดคุณสมบัติ ShowWeight เป็น True

- ความหมายของอาร์กิวเมนต์
 1. hctl แฮนเดิลของคอนโทรล
 2. hdc เนื้อหาอุปกรณ์
 3. Idx หมายเลขบัพ
 4. PofX ตำแหน่งของบัพในแนวนอน หน่วยเป็น Pixel
 5. PofY ตำแหน่งของบัพในแนวตั้ง หน่วยเป็น Pixel

26. **VOID NEAR PaintTree**(HCTL hctl, HWND hwnd, HDC hdc);

- ควบคุมการแสดงผลลักษณะของคอนโทรล
- ความหมายของอาร์กิวเมนต์
 1. hctl แฮนเดิลของคอนโทรล
 2. hwnd แฮนเดิลของหน้าต่างของคอนโทรล

3. hdc เนื้อหาอุปกรณ์

27. VOID NEAR **ReplaceValue**(HCTL hctl, int DeIdx, int ReIdx);

- ทำการเคลื่อนย้ายค่าจากบัพหนึ่งไปยังอีกบัพหนึ่ง
- ความหมายของอาร์กิวเมนต์
 1. hctl แชนเนลของคอนโทรล
 2. DeIdx หมายเลขบัพที่ต้องลบค่าเดิมออก
 3. ReIdx หมายเลขบัพที่นำค่าไปแทนที่บัพ DeIdx

28. VOID NEAR **ReplaceWithLeftSon**(HCTL hctl, int DeIdx, int ReIdx);

- ควบคุมการเคลื่อนย้ายค่าของแต่ละบัพ เมื่อต้องใช้ลูกทางซ้ายของบัพที่จะถูกลบไปแทนที่บัพที่จะถูกลบนั้น

- ความหมายของอาร์กิวเมนต์
 1. hctl แชนเนลของคอนโทรล
 2. DeIdx หมายเลขบัพที่ถูกลบ
 3. ReIdx หมายเลขบัพที่จะมาแทนที่

```
VOID NEAR ReplaceWithLeftSon(HCTL hctl, int DeIdx, int ReIdx)
{
  LPBSEARCH lpbsearch = lpbsearchDEREF(hctl);
  if (DeIdx > MAXNODE || ReIdx > MAXNODE)
    return;
  if (lpbsearch->OccupiedArr(ReIdx)) {
    if ((ReIdx * 2 + 1) <= MAXNODE) {
      if (lpbsearch->OccupiedArr(ReIdx * 2 + 1))
        ReplaceWithLeftSon(hctl, DeIdx * 2 + 1, ReIdx * 2 + 1);
    }
    ReplaceValue(hctl, DeIdx, ReIdx);
    if (ReIdx * 2 <= MAXNODE) {
      if (lpbsearch->OccupiedArr(ReIdx * 2))
        ReplaceWithLeftSon(hctl, DeIdx * 2, ReIdx * 2);
    }
  }
} // ReplaceWithLeftSon0
```

ฟังก์ชันนี้เป็นฟังก์ชันเรียกตัวเอง โดยพารามิเตอร์ตัวที่สองเป็นบัพที่จะถูกลบ DeIdx แล้วให้บัพ ReIdx มาแทนที่ ซึ่งจะทำการแทนที่ในต้นไม้ย่อยทางขวาก่อน แล้วจึงแทนที่รากบัพ จากนั้นจึงจะแทนที่ต้นไม้ย่อยทางซ้าย ลำดับของการแทนที่นี้มีความสำคัญ เพราะป้องกันไม่ให้เกิดการทับกันของค่าในแต่ละบัพขึ้น ฟังก์ชันนี้ใช้กับขั้นตอนวิธีในหัวข้อขั้นตอนวิธีที่สำคัญ ของบทที่ 5 ในเรื่องการลบบัพที่มีลูกทางซ้ายและ/หรือ มีลูกทางขวา

29. VOID NEAR **ReplaceWithRightSon**(HCTL hctl, int DelIdx, int RepIdx);

- ความคุมการเคลื่อนย้ายค่าของแต่ละบัพ เมื่อต้องใช้เพื่อนบ้านถัดไปแบบตามลำดับของบัพที่จะถูกลบไปแทนที่บัพที่จะถูกลบนั้น

- ความหมายของอาร์กิวเมนต์

1. hctl แชนเดิลของคอนโทรล
2. DelIdx หมายเลขบัพที่ถูกลบ
3. RepIdx หมายเลขบัพที่จะมาแทนที่

VOID NEAR **ReplaceWithRightSon**(HCTL hctl, int DelIdx, int RepIdx)

```
{
  LPBSEARCH lpbsearch = LpbsearchDEREF(hctl);
  if (DelIdx > MAXNODE || RepIdx > MAXNODE)
    return;
  if (lpbsearch->OccupiedArr[RepIdx]) {
    if (RepIdx * 2 <= MAXNODE) {
      if (lpbsearch->OccupiedArr[RepIdx * 2])
        ReplaceWithRightSon(hctl, DelIdx * 2, RepIdx * 2);
    }
    ReplaceValue(hctl, DelIdx, RepIdx);
    if (RepIdx * 2 + 1 <= MAXNODE) {
      if (lpbsearch->OccupiedArr[RepIdx * 2 + 1])
        ReplaceWithRightSon(hctl, DelIdx * 2 + 1, RepIdx * 2 + 1);
    }
  }
} // ReplaceWithRightSon0
```

ฟังก์ชันนี้เป็นฟังก์ชันเรียกตัวเอง โดยพารามิเตอร์ตัวที่สองเป็นบัพที่จะถูกลบ DelIdx แล้วให้บัพ RepIdx มาแทนที่ ซึ่งจะทำการแทนที่ในต้นไม้ย่อยทางซ้ายก่อน แล้วจึงแทนที่รากบัพ จากนั้นจึงจะแทนที่ต้นไม้ย่อยทางขวา ลำดับของการแทนที่นี้มีความสำคัญ เพราะป้องกันไม่ให้เกิดการทับกันของค่าในแต่ละบัพขึ้น ฟังก์ชันนี้ใช้กับขั้นตอนวิธีของหัวข้อขั้นตอนวิธีที่สำคัญ ในบทที่ 5 ในเรื่องการลบบัพที่มีลูกทางซ้ายและ/หรือ มีลูกทางขวา

30. VOID NEAR **ScrollBarHandle**(HWND hwnd, HCTL hctl);

- เมื่อมีการกำหนดให้คอนโทรลมี Scroll Bar โดยใช้คุณสมบัติ ScrollBar ก็จะมีการเรียกใช้ฟังก์ชันนี้ เพื่อให้ควบคุมการทำงานของ Scroll Bar (ดูรายละเอียดเพิ่มเติมของคุณสมบัติ ScrollBar ได้จาก ภาคผนวก ค)

- ความหมายของอาร์กิวเมนต์

1. hwnd แชนเดิลของหน้าต่างของคอนโทรล
2. hctl แชนเดิลของคอนโทรล

31. VOID NEAR **SetFirstPos**(HCTL hctl, int FAR *PofX, int FAR *PofY);

- กำหนดตำแหน่งเริ่มต้นของการวาดตัวคอนโทรลภายในพื้นที่ทำงาน (Client Area)

- ความหมายของอาร์กิวเมนต์
 1. hctl แขนงเดิลของคอนโทรล
 2. PofX ตำแหน่งในแนวนอนของบัพ 109วายเป็น Pixel
 3. PofY ตำแหน่งในแนวตั้งของบัพ หน่วยเป็น Pixel

32. **HWND SetNodeShape**(HCTL hctl, HWND hIcon);

- ทำหน้าที่กำหนดลักษณะของบัพ โดยจะคืนค่าเป็นแฮนเดิลของสัญลักษณ์ที่จะใช้เป็นลักษณะบัพ ซึ่งกำหนดโดยคุณสมบัติ NodeShape

- ความหมายของอาร์กิวเมนต์
 1. hctl แขนงเดิลของคอนโทรล
 2. hIcon แฮนเดิลของสัญลักษณ์

33. **VOID NEAR ShowErrorMsg**(LPSTR lpstr);

- แสดงกรอบข้อความ เพื่อบอกข่าวสารหรือข้อผิดพลาดต่าง ๆ เช่น ไม่สามารถเพิ่มบัพใหม่ได้ เพราะเหตุใด

- ความหมายของอาร์กิวเมนต์
 1. lpstr ข้อความที่ใช้แสดงในกรอบข้อความ

34. **LONG NEAR StrToNum**(LPSTR lpstr);

- ทำการคำนวณหาค่าเลขจำนวนเต็มจากอักขระ เช่น "123" ฟังก์ชันนี้จะคืนค่า 123 มาให้ มีการเรียกใช้ฟังก์ชันนี้เมื่อกำหนดคุณสมบัติ ValueType เป็น 0 (เลขจำนวนเต็ม)

- ความหมายของอาร์กิวเมนต์
 1. ค่าอักขระที่ต้องการคำนวณ

35. **LONG FAR PASCAL _export TreeCtlProc** (HCTL hctl, HWND hwnd, USHORT msg, USHORT wp, LONG lp);

- เป็นฟังก์ชันหลักของคอนโทรลทั้งสาม ควบคุมการทำงานทั้งหมดภายในคอนโทรล และต้องกำหนดให้วินโดวส์สามารถเรียกใช้ฟังก์ชันนี้ได้ด้วย (EXPORT)

- ความหมายของอาร์กิวเมนต์
 1. hctl แขนงเดิลของคอนโทรล
 2. hwnd แฮนเดิลของหน้าต่างของคอนโทรล
 3. msg ข้อความที่ส่งมายังฟังก์ชันนี้
 4. wp ข้อมูลเพิ่มเติมของข้อความ

5. lp ข้อมูลเพิ่มเติมของข้อความ

36. VOID NEAR **TreeTravel**(HCTL hctl, int Idx, int Code);

- ทำหน้าที่กำหนดค่าคุณสมบัติ InOrder , PreOrder และ PostOrder

- ความหมายของอาร์กิวเมนต์

1. hctl แชนเดิลของคอนโทรล

2. Idx หมายเลขบัพ

3. Code ตัวบ่งชี้ว่าจะให้กำหนดคุณสมบัติ InOrder , PreOrder หรือ PostOrder

ภาคผนวก ค
คุณสมบัติ, วิธีและเหตุการณ์ของคอนโทรล

ในส่วนของ รูปแบบ จะมีความหมายของลักษณะอักษรดังนี้

- ตัวอักษรปกติ หมายถึง จะต้องมียกขึ้นอยู่ในรูปแบบเสมอ
- ตัวอักษรเอียง ในเครื่องหมาย [] หมายถึง จะมีหรือไม่มียกขึ้นในรูปแบบก็ได้
- *form* และ *CtlName* หมายถึง ชื่อของฟอร์มและคอนโทรลที่ได้กำหนดขึ้นเอง

คุณสมบัติของคอนโทรล

คอนโทรลจะมีทั้งคุณสมบัติมาตรฐาน และคุณสมบัติที่สร้างขึ้นใหม่ ดังนี้

1. คุณสมบัติมาตรฐาน ได้แก่

BackColor Property กำหนดสีพื้นหลัง (Background) เมื่อแสดงค่าของแต่ละบัพในคอนโทรล

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

ค่าที่ใช้ เป็นเลขฐานสิบหก จำนวน 6 หลัก ตั้งแต่ &H000000H ถึง &HFFFFFFH (หรือ 0 ถึง 16,777,215)

เช่น &H000000H หมายถึง สีดำ &HFFFFFFH แทนสีขาว &H00FFFFH หมายถึงสีเหลือง เป็นต้น

ค่าโดยปริยาย ตามค่าของ WINDOW_BACKGROUND ในแฟ้ม CONSTANT.TXT

BorderStyle Property กำหนดลักษณะกรอบของคอนโทรล

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

ตารางที่ ค.1 ค่าที่ใช้ของ BorderStyle Property

ค่าที่ใช้	ลักษณะขอบ
0	ไม่มี
1	เส้นตรง

DragIcon Property กำหนดสัญลักษณ์ที่จะใช้แทนลักษณะของคอนโทรลเมื่อเกิดการลากและวาง (drag-and-drop)

รูปแบบ `[form.] CtlName. DragIcon [= icon]`

ตารางที่ ค.2 ค่าที่ใช้ของ DragIcon Property

ค่าที่ใช้	ลักษณะคอนโทรล
(none)	ลูกศรในกรอบสี่เหลี่ยมขนาดเท่ากับคอนโทรล
Icon	กำหนดลักษณะจากแฟ้มสัญลักษณ์ (สกุล ICO)

ค่าโดยปริยาย (none)

โดยที่ค่าของ *icon* นั้นสามารถใช้ชุดของคำสั่งที่ให้ค่าเป็นแอสเคิลของสัญลักษณ์ก็ได้ เช่น `Form1.Icon`, `Text1.DragIcon` หรือฟังก์ชัน `LoadPicture()` ของวิซวลเบสิก

DragMode Property กำหนดวิธีที่ใช้ในการลากและวาง

รูปแบบ `[form.] CtlName. DragMode [= numericexpression]`

ตารางที่ ค.3 ค่าที่ใช้ของ DragMode Property

ค่าที่ใช้	วิธีลากและวาง
0	จะต้องใช้ Drag Method เพื่อให้เกิดการลากกับคอนโทรลได้ (Manual)
1	สามารถลากได้ โดยการคลิกเมาส์ซ้ายที่คอนโทรล (Automatic)

ค่าโดยปริยาย 0

FontBold, FontItalic, FontStrikethru, FontUnderLine Property กำหนดลักษณะของตัวอักษรที่ใช้แสดงค่าของบัพในคอนโทรล โดย `FontBold` หมายถึง แสดงด้วยตัวหนาหรือไม่ `FontItalic` หมายถึง แสดงด้วยตัวเอียงหรือไม่ `FontStrikethru` หมายถึง แสดงด้วยอักษรที่มีเส้นขีดกลางหรือไม่ `FontUnderline` หมายถึง แสดงด้วยตัวอักษรขีดเส้นใต้หรือไม่

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

`[form.] CtlName. FontItalic [= booleanexpression]`

`[form.] CtlName. FontStrikethru [= booleanexpression]`

`[form.] CtlName. FontUnderLine [= booleanexpression]`

ตารางที่ ค.4 ค่าที่ใช้ของ FontBold , FontItalic , FontStrikethru , FontUnderline Property

ค่าที่ใช้	ความหมาย
True	กำหนดให้ใช้รูปแบบของตัวอักษรนั้น ๆ (เป็นค่าโดยปริยายสำหรับ FontBold)
False	กำหนดให้ไม่ใช้รูปแบบของตัวอักษรนั้น ๆ (เป็นค่าโดยปริยายสำหรับ FontItalic, FontStrikethru และ FontUnderline)

FontName Property กำหนดลักษณะของอักขระที่ใช้ในการแสดงค่าของบัพในคอนโทรล

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

ค่าที่ใช้เป็น อักขระ เช่น `BiSearch1.FontName = "Courier"`

ค่าโดยปริยาย กำหนดโดยระบบ

ในช่วงเวลาดำเนินงาน สามารถหารายชื่อลักษณะของอักขระต่าง ๆ ที่มีในระบบได้จาก `FontCount` และ `Fonts Property` ตัวอย่างเช่น

```
For I = 0 To Screen.FontCount - 1 ' Determine number of fonts.
  Print Screen.Fonts(I) ' Put each font into list box.
Next I
```

FontSize Property กำหนดขนาดของของตัวอักษรที่ใช้แสดงค่าของบัพในคอนโทรล

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

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

ค่าโดยปริยาย กำหนดโดยระบบ

Height, Width Property กำหนดขนาดของคอนโทรล หน่วยที่ใช้ขึ้นอยู่กับ `ScaleMode Property` ของฟอร์มที่คอนโทรลนั้นอยู่ โดย Height กำหนดส่วนสูง Width กำหนดความกว้าง

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

`[form.] CtlName. Width [= numericexpression]`

ค่าที่ใช้เป็น Single

HelpContextID Property กำหนดหมายเลขจำนวนเต็ม เพื่อใช้ในการทำ Context-sensitive Help ให้กับคอนโทรลในโปรแกรมประยุกต์ต่าง ๆ ที่สร้างขึ้น

รูปแบบ `[form.] CtlName. HelpContextID [= numericexpression]`

ค่าที่ใช้ เป็นจำนวนเต็ม 32 บิตแบบคิดเครื่องหมาย (Long Integer)

hWnd Property ไมโครซอฟต์วินโดวส์ จะกำหนดแฮนเดิลให้กับคอนโทรล ซึ่งค่าของ hWnd นั้น จะใช้เป็นอาร์กิวเมนต์ในการติดต่อกับ API ของวินโดวส์

รูปแบบ [form.] CtlName. hWnd

ค่าที่ใช้ เป็น เลขจำนวนเต็ม

คุณสมบัตินี้ไม่มีในหน้าต่างคุณสมบัติและเป็นคุณสมบัติที่อ่านค่าได้อย่างเดียว (Read-Only Property) จะเปลี่ยนแปลงค่าไม่ได้ในช่วงเวลาดำเนินงาน

Index Property กำหนดให้คอนโทรลนั้นเป็นคอนโทรลแบบแถวลำดับหรือไม่

รูปแบบ [form.] CtlName [(i)]. Index

ค่าที่ใช้ เป็นเลขจำนวนเต็ม 0 ถึง 32,767

เพราะคอนโทรลแบบแถวลำดับใช้ Name Property เดียวกัน จึงต้องใช้ Index Property ในการอ้างถึงคอนโทรลแต่ละตัวในแถวลำดับนั้น โดยการเขียนตัวเลขอยู่ในวงเล็บต่อจากชื่อของคอนโทรล เช่น BiSearch(3) และไม่สามารถเปลี่ยนแปลงค่าของคุณสมบัตินี้ได้ในช่วงเวลาดำเนินงาน

Left Property กำหนดระยะห่างระหว่างขอบซ้ายของคอนโทรล กับขอบซ้ายของฟอร์ม หน่วยที่ใช้ขึ้นอยู่กับ

ScaleMode Property ของฟอร์มที่คอนโทรลนั้นอยู่






รูปแบบ [form.] CtlName. Left [= numericexpression]

ค่าที่ใช้ เป็น Single








MousePointer Property กำหนดลักษณะของตัวชี้ตำแหน่งของเมาส์ เมื่อตัวชี้ตำแหน่งอยู่บนคอนโทรล

รูปแบบ [form.] CtlName. MousePointer [= numericexpression]

ตารางที่ ค.5 ค่าที่ใช้ของ MousePointer Property

ค่าที่ใช้	ลักษณะของตัวชี้ตำแหน่ง
0	ลักษณะโดยปริยายของระบบขณะนั้น
1	 (Arrow)
2	 (Cross)
3	 (I-Beam)
4	 (Icon)
5	 (Size)

ตารางที่ ค.5 (ต่อ) ค่าที่ใช้ของ MousePointer Property

6	 (Size NE SW)
7	 (Size NS)
8	 (Size NW SE)
9	 (Size WE)
10	 (Up Arrow)
11	 (Hourglass)
12	 (No Drop)

Name Property กำหนดชื่อที่ใช้แทนคอนโทรลในการเขียนโปรแกรม

ค่าที่ใช้ เป็น อักขระ

ค่าโดยปริยาย คือ ชนิดของคอนโทรลแล้วต่อท้ายด้วยเลขจำนวนเต็ม เช่น BiSearch1, BiSearch2, BiSearch3 ไปเรื่อย ๆ ตามลำดับที่คอนโทรลนั้นถูกสร้างขึ้นมา

การตั้ง Name ต้องขึ้นต้นด้วยตัวอักษร และมีความยาวไม่เกิน 40 ตัวอักษร ซึ่งสามารถประกอบด้วยตัวเลขและเครื่องหมายขีดเส้นใต้ (Underscore) ได้ และห้ามใช้เครื่องหมายวรรคตอนอื่น ๆ รวมทั้งห้ามเว้นวรรคภายใน Name ของคอนโทรล คุณสมบัตินี้ไม่สามารถอ้างถึงได้ในช่วงเวลาดำเนินงาน

Parent Property หมายถึง ฟอรัมที่คอนโทรลนั้นอยู่

รูปแบบ [form.] CtlName. Parent

ค่าที่ใช้ เป็น ฟอรัม

สามารถใช้คุณสมบัตินี้ในการเข้าถึงคุณสมบัติหรือวิธีของฟอรัมที่คอนโทรลนั้นอยู่ได้ เช่น BiSearch1. Parent. MousePointer = 4 เป็นต้น คุณสมบัตินี้ไม่มีในหน้าต่างคุณสมบัติและไม่สามารถเปลี่ยนแปลงค่าของคุณสมบัตินี้ได้ในช่วงเวลาดำเนินงาน

Tag Property กำหนดค่าพิเศษให้กับคอนโทรล เพื่อใช้แทนคอนโทรลในการเขียนโปรแกรม

รูปแบบ [form.] CtlName. Tag [= stringexpression]

ค่าที่ใช้ เป็น อักขระ

ค่าโดยปริยาย อักขระว่าง ("")

Top Property กำหนดระยะห่างระหว่างขอบบนของคอนโทรล กับขอบบนของฟอร์ม หน่วยที่ใช้ขึ้นอยู่กับ ScaleMode Property ของฟอร์มที่คอนโทรลนั้นอยู่

รูปแบบ [form.] CtlName. Top [= numericexpression]

ค่าที่ใช้ เป็น Single

Visible Property กำหนดให้แสดงภาพลักษณะของคอนโทรลในช่วงเวลาดำเนินงานหรือไม่

รูปแบบ [form.] CtlName. Visible [= booleanexpression]

ตารางที่ ค.6 ค่าที่ใช้ของ Visible Property

ค่าที่ใช้	แสดงลักษณะของคอนโทรล
True	แสดง
False	ไม่แสดง

ค่าโดยปริยาย True

2. คุณสมบัติที่สร้างขึ้นใหม่ จากตารางที่ 4.2 ผู้เขียนได้แบ่งคุณสมบัติออกเป็น 2 แบบคือ

2.1 คุณสมบัติที่มีในหน้าต่างคุณสมบัติ บางคุณสมบัติผู้ใช้คอนโทรลก็ไม่สามารถเปลี่ยนแปลงค่าได้ในช่วงเวลาดำเนินงาน

AnimationSpeed Property กำหนดให้แสดงการเปลี่ยนแปลงจำนวนบัพภายในคอนโทรลหรือไม่ในช่วงเวลาดำเนินงาน

รูปแบบ [form.] CtlName. AnimationSpeed [= numericexpression]

ตารางที่ ค.7 ค่าที่ใช้ของ AnimationSpeed Property

ค่าที่ใช้	ลักษณะของการแสดงการเปลี่ยนแปลง
0	ไม่แสดง
1	แสดงด้วยความเร็วช้า
2	แสดงด้วยความเร็วปานกลาง
3	แสดงด้วยความเร็วมาก

ค่าโดยปริยาย 0

เมื่อมีการเพิ่มบัพใหม่เข้ามาในคอนโทรล จะสังเกตเห็นกิ่งของบัพใหม่ค่อย ๆ งอกออกมา แล้วค่อยมีบัพใหม่นั้นเกิดขึ้น เมื่อมีการลบบัพ บัพนั้นจะค่อย ๆ กระทบหายไป บัพที่จะมาแทนที่จะค่อย ๆ เคลื่อนมา ณ ตำแหน่งของบัพที่ถูกลบออกไปนั้น

BranchWidth Property กำหนดความกว้างของกิ่งของคอนโทรล

รูปแบบ `[form.] CtlName. BranchWidth [= numericexpression]`

ค่าที่ใช้ เป็น จำนวนเต็ม 1 - 20

ค่าโดยปริยาย 1

DeletedKey Property กำหนดค่าที่ต้องการลบออกจากคอนโทรล

รูปแบบ `[form.] CtlName. Deletedkey [= stringexpression]`

`[form.] CtlName. Deletedkey [= numericexpression]`

ค่าที่ใช้ เป็น ข้อความ (String) หรือ ตัวเลข (Integer) ซึ่งกำหนดโดย Value Type Property

ค่าโดยปริยาย อักขระว่าง ""

หากมีค่าที่ต้องการลบอยู่ในคอนโทรล ค่านั้นก็จะหายไปจากคอนโทรล จำนวนบัพก็ลดลง 1 บัพ และถ้าเป็นในช่วงเวลาการออกแบบค่านั้นใน ListOfKeys Property จะหายไปด้วย เช่น เดิม ListOfKeys เป็น 24,32,55 แล้วกำหนดค่า DeletedKey = 32 ListOfKeys จะเปลี่ยนเป็น 24,55 เป็นต้น ส่วนรูปร่างของต้นไม้ก็จะเปลี่ยนแปลงไป แต่ก็ยังคงคุณสมบัติของโครงสร้างต้นไม้โบราณค้นหาข้อมูล

EnableInsDelBox Property กำหนดให้ ในช่วงเวลาดำเนินงาน ผู้ใช้คอนโทรลสามารถที่จะดับเบิลคลิกเมาส์ซ้ายที่บัพใด ๆ ในคอนโทรล เพื่อเพิ่มบัพใหม่ หรือคลิกเมาส์ขวาที่บัพใด ๆ เพื่อลบบัพนั้นได้หรือไม่

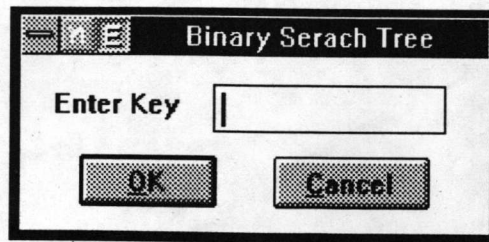
รูปแบบ `[form.] CtlName. EnableInsDelBox [= Booleanexpression]`

ตารางที่ ค.8 ค่าที่ใช้ของ EnableInsDelBox Property

ค่าที่ใช้	สามารถคลิกเมาส์เพื่อเพิ่มหรือลบบัพ
True	ได้
False	ไม่ได้

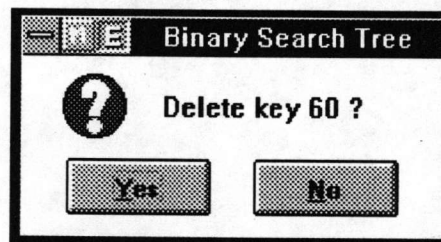
ค่าโดยปริยาย True

กรณีของการเพิ่มบัพใหม่ จะปรากฏกรอบข้อความ ให้ผู้ใช้คอนโทรลใส่ค่าคีย์ที่ต้องการเพิ่มเข้าไปในคอนโทรล ดังรูปที่ ค.1



รูปที่ ค.1 กรอบข้อความของการเพิ่มบัพใหม่

หากต้องการลบบัพใด ๆ ในคอนโทรล ก็จะมีกรอบข้อความถามความแน่ใจของผู้ใช้ครั้งหนึ่ง ดังรูป สมมติต้องการลบคีย์ 60



รูปที่ ค.2 กรอบข้อความของการลบบัพ

Insertedkey Property เป็นค่าที่ใช้ในการเพิ่มบัพใหม่เข้าไปภายในคอนโทรล

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

[form.] CtlName. Insertedkey [= numericexpression]

ค่าที่ใช้ ถ้าเป็นอักขระ จะต้องมีความยาวไม่เกิน 100 อักขระ ถ้าเป็นเลขจำนวนเต็ม จะมีค่าตั้งแต่ 0 - 2,147,483,647 (2^{31})

ค่าโดยปริยาย อักขระว่าง ""

ทุกครั้งที่มีการกำหนดค่าคุณสมบัตินี้ นอกจากจำนวนบัพจะเพิ่มขึ้นแล้ว ในช่วงเวลาการออกแบบค่านั้นจะไปเพิ่มอยู่ใน ListOfKeys Property ด้วย เช่น เมื่อกำหนดค่า InsertedKey Property = 30 ถ้าเดิม ListOfKeys เป็น 12,24 ก็จะเปลี่ยนเป็น 12,24,30 การกำหนดค่าของ InsertedKey นั้น จะต้องกำหนดตามชนิดของข้อมูล หรือตาม ValueType Property หากกำหนดค่าไม่ถูกต้องตามชนิดของข้อมูล หรือค่าของ InsertedKey นั้น จะทำให้ได้ต้นไม้ที่มีระดับความสูงเกินข้อจำกัดของคอนโทรล (LevelLimit Property) หรือค่าของ InsertedKey นั้นซ้ำกับค่าที่มีอยู่แล้วภายในคอนโทรล คอนโทรลก็จะไม่เพิ่มบัพให้ตามค่าที่กำหนดใน InsertedKey และค่าของ ListOfKeys Property ก็ไม่เปลี่ยนแปลง

LevelDistance Property กำหนดระยะห่างของแต่ละระดับภายในคอนโทรล

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

ค่าที่ใช้ เป็นจำนวนเต็ม ตั้งแต่ -32 - 128

ค่าโดยปริยาย 20

ListOfKeys Property เป็นคุณสมบัติที่แสดงชุดลำดับของข้อมูลทั้งหมดในคอนโทรล โดยคั่นแต่ละค่าด้วยเครื่องหมายจุลภาค (,)

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

ค่าที่ใช้ เป็นอักขระ ความยาวไม่เกิน 1,000 อักขระ

ค่าโดยปริยาย อักขระว่าง ""

ในการกำหนดค่าของคุณสมบัตินี้ ตัวคอนโทรลจะทำการสร้างต้นไม้ขึ้นใหม่ โดยทำการลบข้อมูลเดิมในทุกบัพภายในคอนโทรลก่อน จากนั้นจะนำแต่ละค่าที่คั่นด้วยเครื่องหมายจุลภาคของคุณสมบัตินี้ มาสร้างเป็นต้นไม้ใหม่ตามลำดับ

เมื่อมีการกำหนดค่า InsertedKey Property เพื่อเพิ่มบัพให้กับคอนโทรล ค่าของ ListOfKeys ก็จะเปลี่ยนแปลงด้วย (เฉพาะในช่วงเวลาการออกแบบเท่านั้น) เช่น InsertedKey = 20 หากคีย์ 20 สามารถเพิ่มเข้าไปในคอนโทรลได้ ดังนั้น ถ้าเดิมค่าของ ListOfKeys Property เป็น 15,50,32 ก็จะเปลี่ยนเป็น 15,50,32,20 หาก InsertedKey Property จะทำให้ได้ ListOfKeys ที่มีขนาดเกิน 1,000 อักขระ ค่าของ ListOfKeys จะไม่เปลี่ยนแปลง

LoadNodeShape Property กำหนดลักษณะของแต่ละบัพของคอนโทรลโดยใช้รูปภาพจากแฟ้มสัญญาณ

รูปแบบ *[form.] CtlName. LoadNodeShape [= picture]*

ค่าโดยปริยาย NULL

ในช่วงเวลาดำเนินงาน สามารถกำหนดค่าของคุณสมบัตินี้ได้ โดยใช้ฟังก์ชัน LoadPicture() ของวิซวลเบสิก

NodeDistance Property กำหนดระยะห่างของแต่ละบัพในระดับล่างสุดของคอนโทรล

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

ค่าที่ใช้ เป็นจำนวนเต็ม ตั้งแต่ -64 - 128

ค่าโดยปริยาย 16













ไม่สามารถเปลี่ยนแปลงค่าของคุณสมบัตินี้ได้ในช่วงเวลาดำเนินงาน หากกำหนดคุณสมบัตินี้ด้วยจำนวนเต็มลบ จะทำให้บัพในระดับล่างสุดทับซ้อนกัน



NodeShape Property ใช้กำหนดลักษณะของแต่ละบัพของคอนโทรล

รูปแบบ `[form.] CtlName. NodeShape [= numericexpression]`

ตารางที่ ค.9 ค่าที่ใช้ของ NodeShape Property

ค่าที่ใช้	ลักษณะของบัพ
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	

ค่าโดยปริยาย 0

ในการกำหนดลักษณะของบัพนั้น ยังสามารถใช้รูปภาพจากแฟ้มสัญลักษณ์มาแสดงแทนแต่ละบัพได้ด้วย โดยใช้ LoadNodeShape Property หากมีการกำหนดค่าของ LoadNodeShape Property ค่าของ NodeShape Property จะไม่มีผลต่อลักษณะบัพของคอนโทรล

NumberedNode Property กำหนดให้แสดงหมายเลขประจำบัพที่บริเวณด้านล่างของแต่ละบัพในคอนโทรลหรือไม่

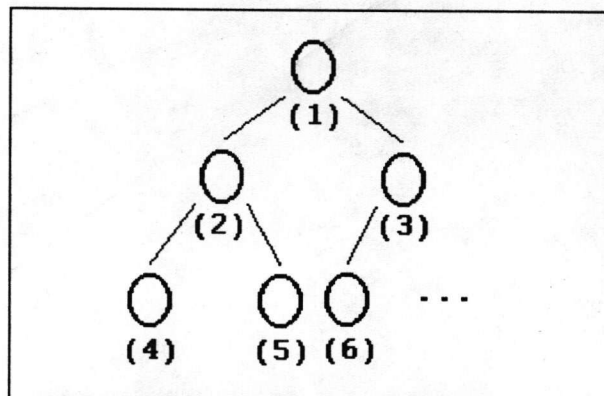
รูปแบบ *[form.] CtlName. NumberedNode [= Booleanexpression]*

ตารางที่ ค.10 ค่าที่ใช้ของ NumberedNode Property

ค่าที่ใช้	แสดงหมายเลขประจำบัพ
True	แสดง
False	ไม่แสดง

ค่าโดยปริยาย False

ตัวอย่างเช่น บัพแรกเป็นหมายเลข 1 บัพลูกทางซ้ายของบัพแรกเป็น 2 ทางขวาเป็น 3 ลูกทางซ้ายของบัพที่ 2 เป็นหมายเลข 4 ลูกทางขวาของบัพที่ 2 เป็น 5 ไปเรื่อย ๆ แบบซีเควนเซียล จนถึง 1,023 ดังรูปที่ ค.3



รูปที่ ค.3 การให้หมายเลขประจำบัพ

ResponseMsg Property กำหนดให้แสดงกรอบข้อความโต้ตอบกับผู้ใช้คอนโทรลหรือไม่

รูปแบบ *[form.] CtlName. ResponseMsg [= Booleanexpression]*

ตารางที่ ค.11 ค่าที่ใช้ของ ResponseMsg Property

ค่าที่ใช้	แสดงกรอบข้อความโต้ตอบกับผู้ใช้
True	แสดง
False	ไม่แสดง

ค่าโดยปริยาย True

กรอบข้อความโต้ตอบอาจเป็นการแจ้งข้อผิดพลาดที่เกิดขึ้น เช่น ไม่มีค่าของ DeletedKey อยู่ในคอนโทรล หรือเป็นการบอกข่าวสารบางอย่าง เช่น การที่ไม่สามารถเพิ่มบัพใหม่ได้ เพราะเหตุใด

ScrollBar Property กำหนดให้คอนโทรลมี Scroll Bar หรือไม่

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

ตารางที่ ค.12 ค่าที่ใช้ของ ScrollBar Property

ค่าที่ใช้	ตำแหน่งของ Scroll Bar
0	ไม่มี
1	มีเฉพาะแนวนอน
2	มีเฉพาะแนวตั้ง
3	มีทั้งแนวนอนและแนวตั้ง

ค่าโดยปริยาย 3

คุณสมบัตินี้ไม่สามารถเปลี่ยนแปลงได้ในช่วงเวลาดำเนินงาน

SearchKey Property กำหนดค่าข้อมูลที่ต้องการหาจากคอนโทรล

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

[form.] CtlName. Searchkey [= numericexpression]

ค่าที่ใช้ เป็น อักขระ หรือ เลขจำนวนเต็ม ซึ่งกำหนดโดย ValueType Property

ค่าโดยปริยาย อักขระว่าง ""

ถ้ามีค่าของ SearchKey อยู่ภายในคอนโทรล FoundNode Property จะเป็นหมายเลขของบัพที่มีค่าของ SearchKey อยู่ ถ้าไม่มีค่าของ SearchKey อยู่ในคอนโทรล ค่าของ FoundNode จะเป็น 0

ShowWeight Property กำหนดให้แสดงน้ำหนักของแต่ละบัพที่บริเวณด้านขวาของแต่ละบัพในคอนโทรลหรือไม่

รูปแบบ *[form.] CtlName. ShowWeight [= Booleanexpression]*

ตารางที่ ค.13 ค่าที่ใช้ของ ShowWeight Property

ค่าที่ใช้	แสดงน้ำหนักแต่ละบัพ
True	แสดง
False	ไม่แสดง

ค่าโดยปริยาย False

ValuePosition Property กำหนดตำแหน่งในการแสดงค่าของข้อมูลของแต่ละบัพในคอนโทรล

รูปแบบ [form.] CtlName. ValuePosition [= numericexpression]

ตารางที่ ค.14 ค่าที่ใช้ของ ValuePosition Property

ค่าที่ใช้	ตำแหน่งค่าของข้อมูล
0	ตรงกลางบัพ
1	ด้านซ้ายของบัพ
2	ด้านบนของบัพ
3	ด้านขวาของบัพ
4	ด้านล่างของบัพ
5	ไม่แสดงค่าของข้อมูล

ค่าโดยปริยาย 0

ValueType Property กำหนดชนิดของข้อมูลในทุกบัพของคอนโทรล

รูปแบบ [form.] CtlName. ValueType [= numericexpression]

ตารางที่ ค.15 ค่าที่ใช้ของ ValueType Property

ค่าที่ใช้	ชนิดของข้อมูล
0	เลขจำนวนเต็ม
1	อักขระ

ค่าโดยปริยาย 0

เมื่อมีการเปลี่ยนแปลงค่าของคุณสมบัตินี้ ตัวคอนโทรลจะลบข้อมูลทั้งหมดในต้นไม้ทิ้งไป แล้วนำข้อมูลทุก ๆ ค่าในชุดลำดับของ ListOfKeys Property มาสร้างต้นไม้ค้นหาข้อมูลขึ้นมาใหม่ ดังนั้น รูปร่างของต้นไม้อาจเปลี่ยนแปลงไปจากเดิมขึ้นกับชนิดของการเปรียบเทียบข้อมูล

2.2 คุณสมบัติที่ไม่มีในหน้าต่างคุณสมบัติ เป็นคุณสมบัติที่ใช้ในการเขียนโปรแกรม ซึ่งจะ มีผลกับคอนโทรลในช่วงเวลาดำเนินงานเท่านั้น

FoundNode Property (Read-Only Property) เป็นหมายเลขของบัพที่มีข้อมูลที่ต้องการหาอยู่

รูปแบบ [form.] CtlName. FoundNode

ค่าที่ใช้ เป็นจำนวนเต็ม ตั้งแต่ 0 - 1,023

ค่าโดยปริยาย 0

เมื่อมีการกำหนดค่า SearchKey Property เพื่อหาข้อมูลจากคอนโทรล หากมีค่านั้นอยู่ใน คอนโทรล คุณสมบัติ FoundNode นี้ก็จะเป็นหมายเลขของบัพที่มีข้อมูลที่ต้องการ ถ้าไม่มีค่านั้นอยู่ใน คอนโทรล FoundNode Property จะมีค่าเป็น 0

InOrder, PreOrder, PostOrder Property (Read-Only Property) เป็นคุณสมบัติที่แสดงชุดลำดับ ของหมายเลขบัพที่ได้จากการท่องไปในต้นไม้แบบตามลำดับ , ก่อนลำดับ และ หลังลำดับ

รูปแบบ [form.] CtlName. InOrder

[form.] CtlName. PreOrder

[form.] CtlName. PostOrder

ค่าที่ใช้ เป็น อักขระ ความยาวไม่เกิน 1,000 อักขระ

ค่าโดยปริยาย อักขระว่าง ""

โดยเป็นข้อความที่คั่นแต่ละค่าด้วยเครื่องหมายจุลภาค (,) เช่น 1,2,4,5,11,3 เป็นต้น หากการท่องไปในต้นไม้แล้วทำให้ได้ชุดลำดับที่เกิน 1,000 อักขระก็จะตัดชุดลำดับที่เกินนั้นทิ้งไป

Key Property ค่าของข้อมูลในแต่ละบัพจะถูกเก็บอยู่ในคุณสมบัตินี้ โดยเริ่มตั้งแต่บัพที่ 1,2,3.... เรื่อยไป จนถึงบัพที่ 1,023

รูปแบบ [form.] CtlName. Key (index) [= numericexpression]

[form.] CtlName. Key (index) [= stringexpression]

ค่าที่ใช้ ถ้าเป็น จำนวนเต็มจะมีค่าได้ตั้งแต่ 0 - 2,147,483,647 (2^{31}) ถ้าเป็นอักขระ จะมีความยาวของ อักขระของแต่ละบัพขึ้นอยู่กับขนาดของหน่วยความจำ

ค่าโดยปริยาย อักขระว่าง ""

ค่าของ index เป็นหมายเลขประจำบัพ ใช้จำนวนเต็มตั้งแต่ 1 - 1,023 เช่น Key(12)

LeftNode Property (Read-Only Property) กำหนดหมายเลขของบัพลูกทางซ้ายของแต่ละบัพในคอนโทรล

รูปแบบ [form.] CtlName. LeftNode (index)

ค่าที่ใช้ เป็นจำนวนเต็มตั้งแต่ 2, 4, 6, ... 2046

ค่าโดยปริยาย $index * 2$

ค่าของ index เป็นจำนวนเต็มตั้งแต่ 1 - 1,023 ตัวอย่างเช่น LeftNode(3) คือ บัพที่ 6
LeftNode(8) คือ บัพที่ 16 เป็นต้น

LevelLimit Property (Read-Only Property) กำหนดจำนวนระดับสูงสุดของคอนโทรล

รูปแบบ [form.] CtlName. LevelLimit

ค่าที่ใช้ เป็นเลขจำนวนเต็ม 9

ค่าโดยปริยาย 9

ในคอนโทรลนี้สามารถมีระดับของต้นไม้สูงสุดได้ 9 ระดับ โดยที่บัพรากจะมีความสูงเท่ากับ 0 ในกรณีที่มีการเพิ่มบัพใหม่แล้วจะทำให้ได้ต้นไม้ที่มีระดับเกิน LevelLimit จะไม่สามารถเพิ่มบัพนั้นได้

ParentNode Property (Read-Only Property) กำหนดหมายเลขของบัพพ่อของแต่ละบัพในคอนโทรล

รูปแบบ [form.] CtlName. ParentNode (index)

ค่าที่ใช้ เป็นจำนวนเต็มตั้งแต่ 1 - 511

ค่าโดยปริยาย $\lfloor index / 2 \rfloor$

ค่าของ index เป็นจำนวนเต็มตั้งแต่ 1 - 1,023 ตัวอย่างเช่น ParentNode(3) คือ บัพที่ 1
ParentNode(8) คือ บัพที่ 4 เป็นต้น และผู้ใช้ไม่สามารถเปลี่ยนค่าคุณสมบัตินี้ได้ช่วงเวลาดำเนินงาน

RightNode Property (Read-Only Property) กำหนดหมายเลขของบัพลูกทางขวาของแต่ละบัพในคอนโทรล

รูปแบบ [form.] CtlName. RightNode (index)

ค่าที่ใช้ เป็นจำนวนเต็มตั้งแต่ 3, 5, 7, ... 2047

ค่าโดยปริยาย $index * 2 + 1$

ค่าของ index เป็นจำนวนเต็มตั้งแต่ 1 - 1,023 เช่น RightNode(3) คือ บัพที่ 7
RightNode(8) คือ บัพที่ 17 เป็นต้น

Root Property (Read-Only Property) กำหนดหมายเลขของบัพรากของคอนโทรล

รูปแบบ [form.] CtlName. Root

ค่าที่ใช้ เป็นเลขจำนวนเต็ม 1

ค่าโดยปริยาย 1

TreeLevel Property (Read-Only Property) เป็นคุณสมบัติที่บอกความสูงของต้นไม้ในขณะนั้น

รูปแบบ `[form.] CtlName. TreeLevel`

ค่าที่ใช้ เป็นจำนวนเต็ม ตั้งแต่ 0 - 9

ค่าโดยปริยาย 0

เมื่อมีการเพิ่มบัพหรือลบบัพ ก็อาจทำให้ความสูงของต้นไม้เปลี่ยนแปลงไป คุณสมบัตินี้จะเป็นค่าของความสูงต้นไม้ขณะนั้น แต่จะมีค่าไม่เกินค่าของ LevelLimit Property (ที่บัพรากจะมีความสูงเท่ากับ 0)

เหตุการณ์ของคอนโทรล

คอนโทรลจะมีทั้งเหตุการณ์มาตรฐาน และเหตุการณ์ที่สร้างขึ้นใหม่ ดังนี้

1. เหตุการณ์มาตรฐาน ได้แก่

Click Event เกิดขึ้นเมื่อมีการคลิกเมาส์ที่ตัวคอนโทรล

รูปแบบ `Sub ctlname_Click ([Index As Integer])`

อาร์กิวเมนต์ *Index* เป็นหมายเลขของแต่ละคอนโทรล ในกรณีที่เป็นคอนโทรลแบบแถวลำดับ หากต้องการทราบว่า มีการคลิกเมาส์ด้วยปุ่มใด ให้ใช้เหตุการณ์ `MouseDown` , `MouseUp` แทน

DbClick Event เกิดขึ้นเมื่อมีการดับเบิลคลิกเมาส์ที่ตัวคอนโทรล

รูปแบบ `Sub ctlname_DbClick ([Index As Integer])`

อาร์กิวเมนต์ *Index* เป็นหมายเลขของแต่ละคอนโทรล ในกรณีที่เป็นคอนโทรลแบบแถวลำดับ หากต้องการทราบว่า มีการคลิกเมาส์ด้วยปุ่มใด ให้ใช้เหตุการณ์ `MouseDown` , `MouseUp` แทน

DragDrop Event เกิดขึ้นเมื่อการลากและวางคอนโทรลเสร็จสมบูรณ์

รูปแบบ `Sub ctlname_DragDrop ([Index As Integer ,] Source As Control, X As Single, Y As Single)`

ตารางที่ ค.16 ความหมายอาร์กิวเมนต์ของ DragDrop Event

อาร์กิวเมนต์	ความหมาย
<i>Index</i>	เป็นหมายเลขของแต่ละคอนโทรล ในกรณีที่เป็นคอนโทรลแบบแถวลำดับ
<i>Source</i>	คอนโทรลที่กำลังถูกลาก ซึ่งเราสามารถอ้างถึงคุณสมบัติและวิธีของคอนโทรลนั้นได้ด้วยอาร์กิวเมนต์นี้ เช่น <code>Source.Visible = 0</code>
<i>X, Y</i>	ตำแหน่งปัจจุบันของเมาส์ในคอนโทรลหรือฟอร์มเป้าหมาย โดยค่าของ <i>X, Y</i> จะเป็นเซนไรรันขึ้นอยู่กับ <code>ScaleHeight, Scale Width, ScaleLeft, and ScaleTop</code> properties ของคอนโทรลหรือฟอร์มเป้าหมายนั้น

ให้ใช้เหตุการณ์นี้ในการควบคุมสิ่งที่จะเกิดขึ้นหลังจากการลากและวางคอนโทรลเสร็จสมบูรณ์ เช่น ทำสำเนาเพิ่มข้อมูล หรือเคลื่อนย้ายเพิ่มข้อมูล เป็นต้น

DragOver Event เกิดขึ้นในขณะที่มีการลากและวางตัวคอนโทรล

รูปแบบ `Sub ctlname_DragOver ([Index As Integer ,] Source As Control, X As Single, Y As Single, State As Integer)`

ตารางที่ ค.17 ความหมายอาร์กิวเมนต์ของ DragOver Event

อาร์กิวเมนต์	ความหมาย
<i>Index</i>	เป็นหมายเลขของแต่ละคอนโทรล ในกรณีที่เป็นคอนโทรลแบบแถวลำดับ
<i>Source</i>	คอนโทรลที่กำลังถูกลาก ซึ่งเราสามารถอ้างถึงคุณสมบัติและวิธีของคอนโทรลนั้นได้ด้วยอาร์กิวเมนต์นี้ เช่น <code>Source.Visible = False</code>
<i>X, Y</i>	ตำแหน่งปัจจุบันของเมาส์ในคอนโทรลหรือฟอร์มเป้าหมาย โดยค่าของ <i>X, Y</i> จะเป็นเซนไรรันขึ้นอยู่กับ <code>ScaleHeight, Scale Width, ScaleLeft, and ScaleTop</code> properties ของคอนโทรลหรือฟอร์มเป้าหมายนั้น
<i>State</i>	สถานะของคอนโทรลที่กำลังถูกลากและวางภายในฟอร์ม หรือคอนโทรลเป้าหมาย โดยมีค่าเป็นดังนี้ 0 - เข้าไปอยู่ในฟอร์มหรือคอนโทรลเป้าหมาย 1 - ออกจากฟอร์มหรือคอนโทรลเป้าหมาย 2 - อยู่ภายในฟอร์มหรือคอนโทรลเป้าหมาย

ให้ใช้เหตุการณ์นี้ในการตรวจสอบตำแหน่งของตัวชี้ตำแหน่งเมาส์ว่า อยู่ที่ใดในคอนโทรลเป้าหมาย และใช้เหตุการณ์นี้ในการกำหนดสิ่งที่จะเกิดขึ้นหลังจากการลากได้เริ่มขึ้น และกำหนดสิ่งที่จะเกิดก่อนที่จะมีการ

วาง (Drop) คอนโทรลในฟอร์มหรือคอนโทรลเป้าหมาย และถ้าหากคอนโทรลได้ถูกวางลงในตำแหน่งเป้าหมายแล้ว เหตุการณ์ DragDrop ก็เกิดขึ้น แต่ถ้าถูกวางลงในตำแหน่งอื่น เหตุการณ์ DragOver ก็จะเกิดขึ้นอีกด้วยอาร์กิวเมนต์ *State* จะมีค่าเป็น 1

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

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)`

ตารางที่ ค.18 ความหมายอาร์กิวเมนต์ของ MouseDown , MouseUp Event

อาร์กิวเมนต์	ความหมาย
<i>Index</i>	เป็นหมายเลขของแต่ละคอนโทรล ในกรณีที่เป็นคอนโทรลแบบแถวลำดับ
<i>Button</i>	หมายถึง ปุ่มของเมาส์ ถ้าค่าเป็น 1 คือ ปุ่มซ้าย 2 คือ ปุ่มขวา 4 คือ ปุ่มกลาง
<i>Shift</i>	กำหนดโดย Argument <i>Button</i> ค่าของ <i>Shift</i> เป็นในลักษณะของบิตฟิลด์ โดยบิต 0 คือ ปุ่ม Shift บิต 1 คือ ปุ่ม Ctrl และบิต 2 คือ ปุ่ม Alt ปุ่มใดถูกกด บิตนั้นก็จะถูก SET ค่าเป็น 1 ตัวอย่างเช่น ถ้า <i>Shift</i> = 5 ก็หมายความว่า ปุ่ม Shift และ Alt ถูกกดพร้อมกัน เป็นต้น
<i>X, Y</i>	ตำแหน่งปัจจุบันของเมาส์บนหน้าจอ โดยค่าของ X, Y จะเป็นเซนโรนนั้นขึ้นอยู่กับ <i>ScaleHeight, Scale Width, ScaleLeft, and ScaleTop properties</i> ของฟอร์มที่คอนโทรลนั้นอยู่

ใช้เหตุการณ์ทั้งสองนี้ควบคุมสิ่งที่จะเกิดขึ้นเมื่อมีการกดหรือปล่อยปุ่มเมาส์ที่ต้องการได้ ซึ่งไม่เหมือนกับเหตุการณ์ Click และ DblClick ที่ไม่สามารถบอกได้ว่ามีการกดเมาส์ด้วยปุ่มใด

MouseMove Event เกิดขึ้นเมื่อมีการเลื่อนเมาส์บนตัวคอนโทรล

รูปแบบ `Sub ctlname_MouseMove ([Index As Integer,] Button As Integer, Shift As Integer, X As Single, Y As Single)`

ตารางที่ ค.19 ความหมายอาร์กิวเมนต์ของ MouseMove Event

อาร์กิวเมนต์	ความหมาย
<i>Index</i>	เป็นหมายเลขของแต่ละคอนโทรล ในกรณีที่เป็นคอนโทรลแบบแถวลำดับ
<i>Button</i>	ค่าของ <i>Button</i> จะบอกถึงการกดปุ่มเมาส์ โดยเป็นลักษณะของบิตฟิลด์ บิต 0 หมายถึง ปุ่มซ้าย บิต 1 คือ ปุ่มขวา และบิต 2 คือ ปุ่มกลางของเมาส์ ถ้าปุ่มใดถูกกด บิตประจำปุ่มนั้นก็จะถูก SET เป็น 1 เช่น ค่าของ <i>Button</i> = 5 ก็หมายความว่า มีการกดทั้งปุ่มซ้ายและปุ่มกลางพร้อม ๆ กัน
<i>Shift</i>	เป็นสถานะของปุ่ม Shift, Ctrl และ Alt ในขณะที่กดปุ่มเมาส์ที่กำหนดโดย Argument <i>Button</i> ค่าของ <i>Shift</i> เป็นในลักษณะของบิตฟิลด์ โดยบิต 0 คือ ปุ่ม Shift บิต 1 คือ ปุ่ม Ctrl และบิต 2 คือ ปุ่ม Alt ปุ่มใดถูกกด บิตนั้นก็จะถูก SET ค่าเป็น 1 ตัวอย่างเช่น ถ้า <i>Shift</i> = 5 ก็หมายความว่า ปุ่ม Shift และ Alt ถูกกดพร้อมกัน เป็นต้น
<i>X, Y</i>	ตำแหน่งปัจจุบันของเมาส์บนหน้าจอ โดยค่าของ <i>X, Y</i> จะเป็นเซนไรรันขึ้นอยู่ กับ <i>ScaleHeight, Scale Width, ScaleLeft, and ScaleTop</i> properties ของฟอร์มที่คอนโทรลนั้นอยู่

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

2. เหตุการณ์ที่สร้างขึ้นใหม่ จากตารางที่ 4.3 มีรายละเอียดของแต่ละเหตุการณ์ดังนี้

NodeDecrease Event เกิดขึ้นเมื่อมี 1 บัพถูกลบออกจากคอนโทรลด้วยวิธี `RemoveItem` หรือด้วยคุณสมบัติ `DeletedKey` ในช่วงเวลาดำเนินงาน

รูปแบบ `Sub CtlName_NodeDecrease ([Index As Integer,] NodeID As Integer, Key As String)`

ตารางที่ ค.20 ความหมายอาร์กิวเมนต์ของ NodeDecrease Event

อาร์กิวเมนต์	ความหมาย
<i>Index</i>	เป็นหมายเลขของแต่ละคอนโทรล ในกรณีที่เป็นคอนโทรลแบบแถวลำดับ
<i>NodeID</i>	หมายเลขประจำบัพที่ถูกลบออกไป
<i>Key</i>	ค่าของบัพที่ถูกลบออกไป

ใช้เหตุการณ์นี้ควบคุมสิ่งที่เกิดขึ้นหลังจากมีการลบบัพออกจากคอนโทรลในช่วงเวลาดำเนินงาน โดยที่เหตุการณ์นี้จะเกิดก่อนเหตุการณ์ `TreeChange`

NodeIncrease Event เกิดขึ้นเมื่อมีบัพใหม่ 1 บัพเพิ่มขึ้นในคอนโทรลด้วยวิธี `AddItem` หรือด้วยคุณสมบัติ `InsertedKey` ในช่วงเวลาดำเนินงาน

รูปแบบ `Sub CtlName_NodeIncrease ([Index As Integer,] NodeID As Integer, Key As String)`

ตารางที่ ค.21 ความหมายอาร์กิวเมนต์ของ NodeIncrease Event

อาร์กิวเมนต์	ความหมาย
<i>Index</i>	เป็นหมายเลขของแต่ละคอนโทรล ในกรณีที่เป็นคอนโทรลแบบแถวลำดับ
<i>NodeID</i>	หมายเลขประจำบัพที่เพิ่มเข้ามา
<i>Key</i>	ค่าของบัพที่เพิ่มเข้ามา

ใช้เหตุการณ์นี้ควบคุมสิ่งที่เกิดขึ้นหลังจากมีบัพใหม่เกิดขึ้นในช่วงเวลาดำเนินงาน โดยที่เหตุการณ์นี้จะเกิดก่อนเหตุการณ์ `TreeChange`

TreeChange Event เกิดขึ้นเมื่อจำนวนบัพในคอนโทรลมีการเปลี่ยนแปลงไป 1 บัพ ซึ่งอาจเป็นการเพิ่มหรือลบบัพออกจากคอนโทรล โดยการใช้วิธี AddItem , RemoveItem หรือด้วยคุณสมบัติ InsertedKey , DeletedKey ในช่วงเวลาดำเนินงาน

รูปแบบ `Sub CtlName_TreeChange ([Index As Integer,] OprType As Integer, NodeID As Integer, Key As String)`

ตารางที่ ค.22 ความหมายอาร์กิวเมนต์ของ TreeChange Event

อาร์กิวเมนต์	ความหมาย
<i>Index</i>	เป็นหมายเลขของแต่ละคอนโทรล ในกรณีที่เป็นคอนโทรลแบบแถวลำดับ
<i>OprType</i>	ชนิดของการเปลี่ยนแปลง ถ้าเป็นการเพิ่มบัพ <i>OprType</i> = 1 ถ้าเป็นการลบบัพ <i>OprType</i> = 0
<i>NodeID</i>	หมายเลขประจำบัพนั้น
<i>Key</i>	ค่าของบัพนั้น

ใช้เหตุการณ์นี้ควบคุมสิ่งที่เกิดขึ้นหลังจากมีการเปลี่ยนแปลงของจำนวนบัพในคอนโทรลในช่วงเวลาดำเนินงาน เหตุการณ์นี้จะเกิดหลังเหตุการณ์ NodeDecrease และ NodeIncrease

วิธีของคอนโทรล

AddItem Method ทำการเพิ่มบัพให้กับคอนโทรล

รูปแบบ `CtlName.AddItem item [, index]`

ตารางที่ ค.23 ความหมายแต่ละส่วนของ AddItem Method

ส่วน	ความหมาย
<i>CtlName</i>	ชื่อของคอนโทรล
<i>item</i>	นิพจน์อักขระหรือเลขจำนวนเต็มที่จะเพิ่มเข้าไปในคอนโทรล
<i>index</i>	หมายเลขของบัพที่จะมี <i>item</i> นั้นอยู่ เป็นจำนวนเต็มตั้งแต่ 1 - 1,023

ในการใช้วิธีนี้กับคอนโทรลต้นไม้นั้นมีขั้นตอนหาข้อมูลและคอนโทรลต้นไม้นั้นที่มีความสูงสมดุล

คอนโทรลทั้งสองจะไม่สนใจส่วน *index* ว่าจะมีหรือไม่ แต่ถ้าใช้กับคอนโทรลต้นไม้นั้นจะต้องมีส่วน *index* ด้วยเสมอ

Clear Method ลบับทุกบัพภายในคอนโทรล

รูปแบบ *CtlName*. Clear

CtlName หมายถึงชื่อของคอนโทรล

Drag Method ใช้เริ่มต้น , ลั้นสุด หรือยกเลิกการลากที่ตัวคอนโทรล

รูปแบบ *CtlName*. Drag [*action*]

ตารางที่ ค.24 ความหมายแต่ละส่วนของ Drag Method

ส่วน	ความหมาย
<i>CtlName</i>	ชื่อคอนโทรลที่จะทำการลาก
<i>action</i>	เลขจำนวนเต็ม 0 - 2 ใช้กำหนดการลาก 0 - ยกเลิกการลาก 1 - เริ่มการลาก 2 - จบการลาก (หรือทำการวางคอนโทรล)

ใช้วิธี Drag เมื่อได้กำหนดค่าคุณสมบัติ DragMode เป็น 0 แต่ก็สามารถใช้วิธีนี้เมื่อกำหนดค่าคุณสมบัติ DragMode เป็น 1 ได้เช่นกัน

Move Method ทำการเคลื่อนย้ายคอนโทรลไปยังตำแหน่งใหม่ภายในฟอร์ม

รูปแบบ [*CtlName.*] Move left [, top [, width [, height]]]

ตารางที่ ค.25 ความหมายแต่ละส่วนของ Move Method

ส่วน	ความหมาย
<i>CtlName</i>	ชื่อของคอนโทรลที่จะเคลื่อนย้าย
<i>left</i>	เลข Single-precision ของตำแหน่งแนวนอนของกรอบด้านซ้ายของคอนโทรล
<i>top</i>	เลข Single-precision ของตำแหน่งแนวตั้งของกรอบด้านบนของคอนโทรล
<i>width</i>	เลข Single-precision ของความกว้างใหม่ของคอนโทรล
<i>height</i>	เลข Single-precision ของความยาวใหม่ของคอนโทรล

สามารถกำหนดเฉพาะ *left* ก็ได้ แต่อย่างไรก็ตามจะต้องกำหนดอาร์กิวเมนต์อื่นด้วยก่อนที่จะกำหนดอาร์กิวเมนต์ที่ต้องการ ตัวอย่างเช่น จะต้องกำหนด *top* และ *width* ด้วย ถ้าต้องการกำหนด *height*

Refresh Method ทำการวาดลักษณะของตัวคอนโทรลใหม่ (Repaint)

รูปแบบ *CtlName.* Refresh

CtlName หมายถึงชื่อของคอนโทรล

RemoveItem Method ทำการลบข้อออกจากคอนโทรล

รูปแบบ *CtlName.* RemoveItem *index*

ตารางที่ ค.26 ความหมายแต่ละส่วนของ RemoveItem Method

ส่วน	ความหมาย
<i>CtlName</i>	ชื่อของคอนโทรล
<i>index</i>	หมายเลขของข้อที่ต้องการลบออกเป็นจำนวนเต็มตั้งแต่ 1 - 1,023

ZOrder Method ให้วิซวลเบสิกวางคอนโทรลที่ข้างหน้าหรือหลังตามลำดับ Z (z-order) ในระดับทางด้านรูปภาพ (Graphical Level)

รูปแบบ [*object.*] ZOrder [*position*]

ตารางที่ ค.27 ความหมายแต่ละส่วนของ ZOrder Method

ส่วน	ความหมาย
<i>CtlName</i>	ชื่อคอนโทรลที่จะวาง
<i>position</i>	ตำแหน่งที่จะวาง เป็นเลขจำนวนเต็มกำหนดตำแหน่งสัมพัทธ์ของคอนโทรลที่จะวางกับคอนโทรลอื่น หากไม่มีการกำหนดหรือกำหนดเป็น 0 คอนโทรลจะถูกวางข้างหน้าตามลำดับ Z ถ้ากำหนดเป็น 1 จะถูกวางข้างหลังตามลำดับ Z

เราสามารถกำหนดลำดับ Z ของคอนโทรลในช่วงเวลาการออกแบบได้ โดยใช้คำสั่ง Bring To Front และ Send To Back ของรายการเลือก Edit



ประวัติผู้เขียน

นายบุญศิริ บุญยก เกิดวันที่ 1 กุมภาพันธ์ พ.ศ. 2514 ที่อำเภอเมือง จังหวัดพัทลุง
สำเร็จการศึกษาปริญญาตรีวิทยาศาสตร์บัณฑิต สาขาศาสตร์คอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี
มหาวิทยาลัยธรรมศาสตร์ ในปีการศึกษา 2534 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต
สาขาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย เมื่อ พ.ศ. 2535