รายการอ้างอิง

ภาษาไทย

กิติ เลิศพิริยะสุวัฒน์. โปรแกรมด้วย Visual Basic for Windows. ไอทีซอฟท์แวร์ 1 (พฤษภาคม 2536): 165-170.

กิติ เลิศพิริยะสุวัฒน์. โปรแกรมด้วย Visual Basic for Windows. ไอทีซอฟท์แวร์ 1 (มิถุนายน 2536): 167-172.

นิสาชล โตอดิเทพย์. โครงสร้างข้อมูล. กรุงเทพมหานคร: สำนักพิมพ์โอเดียนสโตร์, 2535.

บุญศิริ บุญยก. การสร้างวัตถุที่เป็นโครงสร้างข้อมูลแบบต้นไม้. วิทยานิพนธ์ปริญญามหาบัณฑิต จุฬาลงกรณ์มหาวิทยาลัย, 2538.

ประกาศิต ชาติบุรุษ และอาทิตย์ จิตต์จุฬานนท์. โครงสร้างข้อมูลและอัลกอริทึม. กรุงเทพมหานคร: บริษัท ซีเอ็ดยูเคชั่น จำกัด, 2533.

วีเนอร์, ริชาด์ เอส. และพินสัน, ลูอิส เจ. หลักการโปรแกรม Object-Oriented และภาษา C++. แปลโดย นพดล ตังควรรณวานิช และสรรค์ เสกขุนทด. กรุงเทพมหานคร: บริษัท ซีเอ็ดยูเคชั่น จำกัด, 2533.

สุชาย ธนวเสถียร และวิชัย จิวังกูร. โครงสร้างข้อมูลเพื่อการออกแบบโปรแกรมคอมพิวเตอร์. กรุงเทพมหานคร: บริษัท ซีเอ็ดยูเคชั่น จำกัด, 2529.

ภาษาอังกฤษ

Chang, S., K. Principles of Visual Programming Systmes. New Jersey: Prentice-Hall Inc., 1990.

Cilwa, P., and Duntemann, J. Windows Programming Power with Custom Controls. Arizona: Coriolis Group Books, 1994.

Kruse, R., L., Leung, B., P., and Tondo, C., L. Data Structures and Program Design in C. The Republic of Singapor: Prentice-Hall Inc., 1991.

Microsoft Corporation. Microsoft Visual Basic Programming System for Windows, Lanuage Reference Version 3.0. U.S.A.: Microsoft Corp., 1993.

Microsoft Corporation. Microsoft Visual Basic Programming System for Windows, Professional FeatureBook 1 Version 3.0. U.S.A.: Microsoft Corp., 1993.

Rector, B., E. Developing Windows™ 3.1 Application with Microsoft® C/C++. 2nd ed. Indiana: Sams Publishing, 1993.

Weiss, M., A. Data Structures and Algorithm Analysis in C. California: The Benjamin/Cummings Publishing Inc., 1992.

ภาคผนวก

ภาคผนวก ก

แฟ้มโปรแกรมของตัวควบคุม

แฟ้มโปรแกรมที่ใช้แปลและเชื่อมโยงเป็นแฟ้มของตัวควบคุม HPCTL.VBX มีดังต่อไปนี้

1. VBX.H
2. HPCTL.H
3. HPCTL.CPP
4. HPMGT. CPP
5. PAINTLN.CPP
6. PAINTTR.CPP
7. DIALOG.CPP
8. HPFTN.CPP
9. HPCTL.RC
10. HPCTL.DEF
11. HPCTLCU.BMP
12. HPCTLCD.BMP
13. HPCTLEU.BMP
14. HPCTLMU.BMP
15. MAKEFILE

ความสำคัญของแต่ละแฟ้มสามารถดูรายละเอียดได้ในบทที่ 5

แฟ้มข้อความที่ใช้แปลให้เป็นแฟ้มความช่วยเหลือ HPCTL.HLP ได้แก่

1. HPCTL.RTF เป็นแฟ้มข้อความประเภท Rich text file (RTF) ซึ่งจะใช้ในการแปลให้เป็นแฟ้มความช่วยเหลือ แฟ้มนี้ใช้เก็บหัวข้อและข้อความต่าง ๆ ที่จะใช้เป็นเนื้อหาของแฟ้มความช่วยเหลือ เราสามารถใช้โปรแกรม Microsoft Word for Windows ในการสร้างแฟ้มข้อความประเภทนี้ได้

2. HPCTL.HPJ เป็นแฟ้มที่ใช้กำหนดข้อมูลที่จำเป็นต้องใช้ในการแปลให้เป็นแฟ้มความช่วยเหลือ เช่นกำหนดชื่อของแฟ้มข้อความที่จะใช้เป็นเนื้อหาของแฟ้มความช่วยเหลือ กำหนดแฟ้มรูปภาพที่จะใช้ในการแปลให้เข้าไปรวมอยู่ในแฟ้มความช่วยเหลือ และกำหนดวิธีการขั้นตอนในการแปลให้เป็นแฟ้มความช่วยเหลือ เป็นต้น

ในวิทยานิพนธ์นี้ใช้ชุดโปรแกรม Microsoft Help Compiler รุ่น 3.1 เป็นโปรแกรมในการแปลให้เป็นแฟ้มความช่วยเหลือ HPCTL.HLP

แฟ้มโปรแกรมตัวอย่างสำหรับวิชวลเบสิค ประกอบด้วย

1. GBL.BAS เป็นแฟ้มโมดูล (Module) ของภาษาเบสิค แฟ้มนี้ใช้เฉพาะในโปรแกรมตัวอย่างทั้งสามโปรแกรม เพื่อนิยามฟังก์ชัน API ต่าง ๆ ที่ใช้อยู่ในโปรแกรมตัวอย่างและกำหนดข้อมูลชนิดใหม่ ซึ่งใช้เป็นโครงสร้างของบัพ

2. แฟ้มประเภท .FRM เป็นแฟ้มที่เก็บฟอร์มและตัวควบคุมแบบต่าง ๆ ที่จำเป็นต้องใช้ในโปรแกรมของวิชวลเบสิค

3. แฟ้มประเภท .MAK เป็นแฟ้มที่ใช้ในการกำหนดแฟ้มชนิดต่าง ๆ ที่ใช้ในโปรแกรม รวมทั้งรายละเอียดอื่น ๆ ที่ใช้ในการสร้างโปรแกรม เช่น ขนาดของกรอบของโปรแกรม, สัญรูปของโปรแกรม และชื่อของโปรแกรม เป็นต้น

โปรแกรม HPSM1.EXE สร้างจากแฟ้มดังต่อไปนี้

1. GBL.BAS
2. HPSM1.FRM
3. HPSM1.MAK

โปรแกรม HPSM2.EXE สร้างจากแฟ้มดังต่อไปนี้

1. GBL.BAS
2. HPSM2.FRM
3. HPSM2.MAK

โปรแกรม HPSM3.EXE สร้างจากแฟ้มดังต่อไปนี้

1. GBL.BAS
2. HPSM3.FRM
3. HPSM3.MAK

ภาคผนวก ข

ฟังก์ชันที่ใช้ทำงานภายในตัวควบคุม

การทำงานต่าง ๆ ภายในตัวควบคุมมีการใช้ฟังก์ชันที่อยู่ในแฟ้มโปรแกรมต่าง ๆ ดังนี้

แฟ้ม HPCTL.CPP

1. LONG FAR PASCAL _export HeapCtlProc(HCTL hctl, HWND hwnd, USHORT msg, USHORT wp, LONG lp)

เป็นฟังก์ชันหลักที่ใช้ในการควบคุมการทำงานต่าง ๆ ภายในตัวควบคุม ฟังก์ชันนี้ต้องกำหนดให้มีที่อยู่แบบ 32 บิต (FAR) และกำหนดให้วินโดว์สามารถเรียกใช้ฟังก์ชันนี้ได้ (_export)

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hctl แฮนเดิลของตัวควบคุม

hwnd แฮนเดิลของวินโดว์ของตัวควบคุม

msg ข้อความที่ส่งมายังฟังก์ชันนี้

wp ข้อมูลเพิ่มเติมของข้อความ

lp ข้อมูลเพิ่มเติมของข้อความ

แฟ้ม DIALOG.CPP

1. HWND NEAR HwndInitFlashPopup(VOID)

ใช้สร้างหน้าต่างสำหรับการทดลองเพิ่มบัพข้อมูลใหม่เข้าไปในโครงสร้างข้อมูลในช่วงของการออกแบบโปรแกรม (Design Time)

2. LONG FAR PASCAL _export FlashPopupWndProc(HWND hwnd, USHORT msg, USHORT wp, LONG lp)

เป็นฟังก์ชันกระบวนงานของหน้าต่างสำหรับการทดลองเพิ่มบัพข้อมูลใหม่ในช่วงของการออกแบบโปรแกรม (Design Time) ใช้ควบคุมการทำงานของหน้าต่างของการทดลองเพิ่มบัพข้อมูลใหม่

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hwnd แฮนเดิลของวินโดว์ของหน้าต่างสำหรับการทดลองเพิ่มบัพข้อมูล

msg ข้อความที่ส่งมายังฟังก์ชันนี้

wp ข้อมูลเพิ่มเติมของข้อความ

lp ข้อมูลเพิ่มเติมของข้อความ

3. BOOL FAR PASCAL _export InputNode(HWND hdlg, UNIT msg, WPARAM wParam, LPARAM lParam)

เป็นฟังก์ชันที่ใช้ในการแสดงกรอบรับข้อความ (Dialog Box) เพื่อรอรับบัพข้อมูลใหม่เข้าไปในโครงสร้างข้อมูล ซึ่งจะมีการตรวจสอบความถูกต้องของข้อมูลทั้งสองส่วนของบัพข้อมูลด้วย

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hdlg แฮนเดิลของวินโดว์ของกรอบรับข้อความ

msg ข้อความที่ส่งมายังฟังก์ชันนี้

wParam ข้อมูลเพิ่มเติมของข้อความ

lParam ข้อมูลเพิ่มเติมของข้อความ

4. BOOL FAR PASCAL _export ChgNode(HWND hdlg, UNIT msg, WPARAM wParam, LPARAM lParam)

เป็นฟังก์ชันที่ใช้ในการแสดงบัพข้อมูลไว้ในกรอบรับข้อความ ซึ่งเป็นบัพที่ถูกเลือกด้วยปุ่มเมาส์ข้างขวาจากโครงสร้างข้อมูลที่แสดงอยู่ในกรอบของตัวควบคุม เพื่อรอรับการแก้ไขเปลี่ยนแปลงหรือลบบัพข้อมูลนั้นออกจากโครงสร้างข้อมูล และการนำบัพข้อมูลที่ มีการแก้ไขเปลี่ยนแปลงเข้าไปเก็บไว้ในโครงสร้างข้อมูล จะต้องมีการตรวจสอบความถูกต้องของข้อมูลทั้ง 2 ส่วนก่อน

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hdlg แฮนเดิลของวินโดว์ของกรอบรับข้อความ

msg ข้อความที่ส่งมายังฟังก์ชันนี้

wParam ข้อมูลเพิ่มเติมของข้อความ

lParam ข้อมูลเพิ่มเติมของข้อความ

แฟ้ม HPMGT.CPP

1. VOID BuildHeap(HWND hwnd, HCTL hctl, BOOL DrawL, char opr, int noden)

ใช้ในการจัดโครงสร้างข้อมูลให้เป็นฮีปโดยให้มีวิธีการจัดเรียงของข้อมูลตามอาร์กิวเมนต์ที่กำหนดเข้ามา ฟังก์ชันนี้จะถูกเรียกใช้เมื่อค่าของคุณสมบัติ BuildHeap เป็น True

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hwnd แฮนเดิลของวินโดว์ของตัวควบคุม

hctl แฮนเดิลของตัวควบคุม

Drawl ใช้กำหนดว่าต้องการให้วาดบัพสุดท้ายในโครงสร้างข้อมูลที่เพิ่งถูกเพิ่มเข้ามาก่อนทำการจัดโครงสร้างข้อมูลหรือไม่

opr ตัวบ่งชี้ว่ามีกระทำอะไรเกิดขึ้นในโครงสร้างข้อมูล

noden ตำแหน่งของบัพในแถวลำดับ ซึ่งเป็นบัพที่ถูกกระทำ เช่น เป็นบัพใหม่ที่เพิ่มเข้ามา หรืออาจเป็นบัพที่ถูกเปลี่ยนแปลงข้อมูล

2. VOID InsertMaxHpAn(List_type FAR * hp, Item_type item, Item_rep itemrep, int start, int maxheap, HWND hwnd, HCTL hctl)

ใช้ในการจัดต้นไม้ย่อยที่กำหนดให้เป็นฮีปที่บัพรากมีค่าของคีย์มากกว่าบัพลูก (ข้อมูลที่มีค่าของคีย์มากกว่าอยู่ข้างบน) และให้มีการแสดงการเปลี่ยนแปลงของบัพในแต่ละขั้นตอนการทำงาน (Animation) ฟังก์ชันนี้จะถูกเรียกใช้เมื่อค่าของคุณสมบัติ Animation เป็น True และ คุณสมบัติ HeapOrder เป็น Max on Top

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hp โครงสร้างข้อมูลที่จะให้จัดเป็นฮีปซึ่งมีการแทนที่ข้อมูลไว้ในหน่วยความจำเป็นแบบแถวลำดับ

item ส่วนที่เป็นคีย์ของบัพข้อมูล

itemrep ส่วนที่เป็นตัวแทนของบัพข้อมูล

start ตำแหน่งในแถวลำดับ ซึ่งเป็นบัพรากของต้นไม้ย่อย

maxheap ตำแหน่งในแถวลำดับ ซึ่งเป็นบัพสุดท้ายของต้นไม้ย่อย

hwnd แฮนเดิลของวินโดว์ของตัวควบคุม

hctl แฮนเดิลของตัวควบคุม

3. VOID InsertMinHpAn(List_type FAR * hp, Item_type item, Item_rep itemrep, int start, int maxheap, HWND hwnd, HCTL hctl)

ใช้ในการจัดต้นไม้ย่อยที่กำหนดให้เป็นฮีปที่บัพรากมีค่าของคีย์น้อยกว่าบัพลูก (ข้อมูลที่มีค่าของคีย์น้อยกว่าอยู่ข้างบน) และให้มีการแสดงการเปลี่ยนแปลงของบัพในแต่ละขั้นตอนการทำงาน ฟังก์ชันนี้จะถูกเรียกใช้เมื่อค่าของคุณสมบัติ Animation เป็น True และ คุณสมบัติ HeapOrder เป็น Min on Top

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hp โครงสร้างข้อมูลที่จะให้จัดเป็นฮีปซึ่งมีการแทนที่ข้อมูลไว้ในหน่วยความจำเป็นแบบแถวลำดับ

item ส่วนที่เป็นคีย์ของบัพข้อมูล

itemrep ส่วนที่เป็นตัวแทนของบัพข้อมูล

start ตำแหน่งในแถวลำดับ ซึ่งเป็นบัพรากของต้นไม้ย่อย

maxheap ตำแหน่งในแถวลำดับ ซึ่งเป็นบัพสุดท้ายของต้นไม้ย่อย

hwnd แฮนเดิลของวินโดว์ของตัวควบคุม

hctl แฮนเดิลของตัวควบคุม

4. VOID InsertMaxHp(List_type FAR * hp, Item_type item, Item_rep itemrep, int start, int maxheap)

ใช้ในการจัดต้นไม้ย่อยที่กำหนดให้เป็นฮีปที่บัพรากมีค่าของคีย์มากกว่าบัพลูก (ข้อมูลที่มีค่าของคีย์มากกว่าอยู่ข้างบน) แต่ไม่มีการแสดงการเปลี่ยนแปลงของบัพในระหว่างการทำงานของฟังก์ชัน ฟังก์ชันนี้จะถูกเรียกใช้เมื่อค่าของคุณสมบัติ HeapOrder เป็น Max on Top

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hp โครงสร้างข้อมูลที่จะให้จัดเป็นฮีปซึ่งมีการแทนที่ข้อมูลไว้ในหน่วยความจำเป็นแบบแถวลำดับ

item ส่วนที่เป็นคีย์ของบัพข้อมูล

itemrep ส่วนที่เป็นตัวแทนของบัพข้อมูล

start ตำแหน่งในแถวลำดับ ซึ่งเป็นบัพรากของต้นไม้ย่อย

maxheap ตำแหน่งในแถวลำดับ ซึ่งเป็นบัพสุดท้ายของต้นไม้ย่อย

5. VOID InsertMinHp(List_type FAR * hp, Item_type item, Item_rep itemrep, int start, int maxheap)

ใช้ในการจัดต้นไม้ย่อยที่กำหนดให้เป็นฮีปที่บัพรากมีค่าของคีย์น้อยกว่าบัพลูก (ข้อมูลที่มีค่าของคีย์น้อยกว่าอยู่ข้างบน) แต่ไม่มีการแสดงการเปลี่ยนแปลงของบัพในระหว่างการทำงานของฟังก์ชัน ฟังก์ชันนี้จะถูกเรียกใช้เมื่อค่าของคุณสมบัติ HeapOrder เป็น Min on Top

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hp โครงสร้างข้อมูลที่จะให้จัดเป็นฮีปซึ่งมีการแทนที่ข้อมูลไว้ในหน่วยความจำเป็นแบบแถวลำดับ

item ส่วนที่เป็นคีย์ของบัพข้อมูล

itemrep ส่วนที่เป็นตัวแทนของบัพข้อมูล

start ตำแหน่งในแถวลำดับ ซึ่งเป็นบัพรากของต้นไม้ย่อย

maxheap ตำแหน่งในแถวลำดับ ซึ่งเป็นบัพสุดท้ายของต้นไม้ย่อย

6. VOID PercolUpMin(List_type FAR * hp, int noden)

ใช้ในการเคลื่อนย้ายบัพที่กำหนดขึ้นข้างบนเพื่อจัดโครงสร้างข้อมูลให้เป็นฮีปที่บัพรากมีค่าของคีย์น้อยกว่าบัพลูก (ข้อมูลที่มีค่าของคีย์น้อยกว่าอยู่ข้างบน) แต่ไม่มีการแสดงการเปลี่ยนแปลงของบัพในระหว่างการทำงานของฟังก์ชัน ฟังก์ชันนี้จะถูกเรียกใช้เมื่อค่าของคุณสมบัติ HeapOrder เป็น Min on Top

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hp โครงสร้างข้อมูลที่จะให้จัดเป็นฮีปซึ่งมีการแทนที่ข้อมูลไว้ในหน่วยความจำเป็นแบบแถวลำดับ

noden ตำแหน่งในแถวลำดับ ซึ่งเป็นบัพที่จะถูกทำการเคลื่อนย้ายขึ้นข้างบน

7. VOID PercolUpMax(List_type FAR * hp, int noden)

ใช้ในการเคลื่อนย้ายบัพที่กำหนดขึ้นข้างบนเพื่อจัดโครงสร้างข้อมูลให้เป็นฮีปที่บัพรากมีค่าของคีย์มากกว่าบัพลูก (ข้อมูลที่มีค่าของคีย์มากกว่าอยู่ข้างบน) แต่ไม่มีการแสดงการเปลี่ยนแปลงของบัพในระหว่างการทำงานของฟังก์ชัน ฟังก์ชันนี้จะถูกเรียกใช้เมื่อค่าของคุณสมบัติ HeapOrder เป็น Max on Top

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hp โครงสร้างข้อมูลที่จะให้จัดเป็นฮีปซึ่งมีการแทนที่ข้อมูลไว้ในหน่วยความจำเป็นแบบแถวลำดับ

noden ตำแหน่งในแถวลำดับ ซึ่งเป็นบัพที่จะถูกทำการเคลื่อนย้ายขึ้นข้างบน

8. VOID PercolUpMinAn(List_type FAR * hp, int noden, HWND hwnd, HCTL hctl)

ใช้ในการเคลื่อนย้ายบัพที่กำหนดขึ้นข้างบนเพื่อจัดโครงสร้างข้อมูลให้เป็นฮีปที่บัพรากมีค่าของคีย์น้อยกว่าบัพลูก (ข้อมูลที่มีค่าของคีย์น้อยกว่าอยู่ข้างบน) และให้มีการแสดงการเปลี่ยนแปลงของบัพในแต่ละขั้นตอนการทำงาน ฟังก์ชันนี้จะถูกเรียกใช้เมื่อค่าของคุณสมบัติ Animation เป็น True และคุณสมบัติ HeapOrder เป็น Min on Top

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hp โครงสร้างข้อมูลที่จะให้จัดเป็นฮีปซึ่งมีการแทนที่ข้อมูลไว้ในหน่วยความจำเป็นแบบแถวลำดับ

noden ตำแหน่งในแถวลำดับ ซึ่งเป็นบัพที่จะถูกทำการเคลื่อนย้ายขึ้นข้างบน

hwnd แฮนเดิลของวินโดว์ของตัวควบคุม

hctl แฮนเดิลของตัวควบคุม

9. VOID PercolUpMaxAn(List_type FAR * hp, int noden, HWND hwnd, HCTL hctl)

ใช้ในการเคลื่อนย้ายบัพที่กำหนดขึ้นข้างบนเพื่อจัดโครงสร้างข้อมูลให้เป็นฮีปที่บัพรากมีค่าของคีย์มากกว่าบัพลูก (ข้อมูลที่มีค่าของคีย์มากกว่าอยู่ข้างบน) และให้มีการแสดงการเปลี่ยนแปลงของบัพในแต่ละขั้นตอนการทำงาน ฟังก์ชันนี้จะถูกเรียกใช้เมื่อค่าของคุณสมบัติ Animation เป็น True และคุณสมบัติ HeapOrder เป็น Max on Top

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hp โครงสร้างข้อมูลที่จะให้จัดเป็นฮีปซึ่งมีการแทนที่ข้อมูลไว้ในหน่วยความจำเป็นแบบแถวลำดับ

noden ตำแหน่งในแถวลำดับ ซึ่งเป็นบัพที่จะถูกทำการเคลื่อนย้ายขึ้นข้างบน

hwnd แฮนเดิลของวินโดว์ของตัวควบคุม

hctl แฮนเดิลของตัวควบคุม

10.VOID NEAR PaintDelNod(HWND hwnd, HCTL hctl, int slnode)

ใช้แสดงการเปลี่ยนแปลงของโครงสร้างข้อมูล เมื่อมีการลบบัพข้อมูลออกจากโครงสร้างข้อมูล โดยจะแสดงการนำบัพสุดท้ายเข้าไปแทนที่บัพที่จะถูกลบออก และลบบัพสุดท้ายออกจากโครงสร้างข้อมูลแทน

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hwnd แฮนเดิลของวินโดว์ของตัวควบคุม

hctl แฮนเดิลของตัวควบคุม

slnode ตำแหน่งของบัพในแถวลำดับ ซึ่งเป็นบัพที่ต้องการจะลบออก

11.VOID NEAR PaintChgNod(HWND hwnd, HCTL hctl, int slnode, int iValue, LPSTR szTextN)

ใช้แสดงบัพที่ถูกแก้ไขเปลี่ยนแปลง โดยจะแสดงข้อมูลเดิมของบัพก่อนที่จะถูกแก้ไข และแสดงข้อมูลใหม่ของบัพหลังจากแก้ไขเสร็จแล้ว

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hwnd แฮนเดิลของวินโดว์ของตัวควบคุม

hctl แฮนเดิลของตัวควบคุม

slnode ตำแหน่งของบัพในแถวลำดับ ซึ่งเป็นบัพที่จะทำการแก้ไข

iValue ค่าคีย์ของบัพที่ถูกแก้ไข ค่านี้จะเป็นค่าใหม่ที่จะเข้าไปแทนที่ค่าเดิม

szTextN ค่าของส่วนที่เป็นตัวแทนของบัพ ซึ่งเป็นค่าใหม่ที่จะเข้าไปแทนที่ค่าเดิม

12.BOOL NEAR ChgLevelTr(HCTL hctl, HWND hwnd, HDC hdc)

ใช้ตรวจสอบว่า การเพิ่มขึ้นหรือลดลงของจำนวนบัพ จะทำให้จำนวนระดับของโครงสร้างข้อมูลเปลี่ยนไปหรือไม่ และถ้าจำนวนระดับยังไม่เปลี่ยน ฟังก์ชันนี้จะทำการเตรียมพื้นที่สำหรับการวาดบัพสุดท้ายของโครงสร้างข้อมูลให้ด้วย ฟังก์ชันนี้จะถูกเรียกใช้เมื่อคุณสมบัติ DisplayTree เป็น True

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hctl แฮนเดิลของตัวควบคุม

hwnd แฮนเดิลของวินโดว์ของตัวควบคุม

hdc แฮนเดิลของลักษณะอุปกรณ์ (Device Context)

13.LONG NEAR PASCAL Methods (HCTL hctl, HWND hwnd, USHORT meth, LONG lp)

เมื่อมีการใช้วิธี จะเรียกใช้ฟังก์ชันนี้เพื่อจัดการการทำงานของวิธี ซึ่งตัวควบคุมนี้จะมีการกำหนดการทำงานขึ้นมาใหม่เฉพาะวิธี CLEAR เท่านั้น โดยฟังก์ชันจะตรวจสอบดูว่าวิธีที่ใช้นั้นเป็นวิธี CLEAR ใช่หรือไม่ ถ้าใช่ ก็จะทำการลบบัพข้อมูลทั้งหมดออกจากโครงสร้างข้อมูล

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hctl แฮนเดิลของตัวควบคุม

hwnd แฮนเดิลของวินโดว์ของตัวควบคุม

meth ดัชนีเพื่อบ่งชี้ว่าเป็นวิธีอะไร

lp ตัวชี้ที่อยู่ของค่าอาร์กิวเมนต์ของวิธี

14.VOID NEAR FireNodeEvt(HCTL hctl, int n1, int k1, LPSTR Text1, char opr)

ใช้ฟังก์ชันนี้เพื่อทำให้เหตุการณ์ต่าง ๆ ของตัวควบคุมเกิดขึ้น พร้อมทั้งเตรียมข้อมูลเพื่อจะส่งเป็นอาร์กิวเมนต์สำหรับกระบวนงานเหตุการณ์ในวิชวลเบสิค (ดูรายละเอียดเพิ่มเติมสำหรับเหตุการณ์ต่าง ๆ ได้ในภาคผนวก ค.)

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hctl แฮนเดิลของตัวควบคุม

n1 ตำแหน่งของบัพในแถวลำดับที่เกี่ยวข้องกับเหตุการณ์ที่เกิดขึ้น

k1 ค่าคีย์ของบัพที่เกี่ยวข้องกับเหตุการณ์ที่เกิดขึ้น

Text1 ค่าส่วนที่เป็นตัวแทนของบัพที่เกี่ยวข้องกับเหตุการณ์ที่เกิดขึ้น

opr ตัวบ่งชี้ว่าเป็นเหตุการณ์อะไรที่เกิดขึ้น

แฟ้ม PAINTTR.CPP

1. VOID NEAR PaintTimTre(HCTL hctl, HWND hwnd, HDC hdc, BOOL chglvl)

ใช้ในการแสดงผลของโครงสร้างข้อมูล ในรูปแบบแผนภาพของต้นไม้ โดยระยะเวลาที่ใช้ในการวาดบัพแต่ละบัพ จะถูกกำหนดโดยค่าคุณสมบัติ NodeTime และเรียกใช้ฟังก์ชันนี้เมื่อค่าคุณสมบัติ DisplayTree เป็น True

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hctl แฮนเดิลของตัวควบคุม

hwnd แฮนเดิลของวินโดว์ของตัวควบคุม

hdc แฮนเดิลของลักษณะอุปกรณ์ (Device Context)

chglvl ตัวบ่งชี้ว่ามีการเปลี่ยนแปลงจำนวนระดับในโครงสร้างข้อมูลเกิดขึ้นหรือไม่

2. BOOL NEAR InNodeTre (HCTL hctl, HWND hwnd, SHORT xcoord, SHORT ycoord, LPLONG slnode)

ใช้ฟังก์ชันนี้ในการตรวจสอบว่า ตำแหน่งบนตัวควบคุมที่ถูกเลือกด้วยการใช้เมาส์ปุ่มข้างขวานั้น มีบัพปรากฏอยู่ที่ตำแหน่งที่เลือกนั้นจริงหรือไม่ และถ้ามีบัพอยู่จริง ก็จะเก็บตำแหน่งของบัพในแถวลำดับไว้ในอาร์กิวเมนต์ slnode ฟังก์ชันนี้จะถูกเรียกใช้งานเมื่อค่าคุณสมบัติ DisplayTree ในขณะนั้นเป็น True

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hctl แฮนเดิลของตัวควบคุม

hwnd แฮนเดิลของวินโดว์ของตัวควบคุม

xcoord ตำแหน่งในแนวนอนที่เกิดจากการใช้เมาส์เลือกลงบนตัวควบคุม มีหน่วยเป็น Pixel

ycoord ตำแหน่งในแนวตั้งที่เกิดจากการใช้เมาส์เลือกลงบนตัวควบคุม มีหน่วยเป็น Pixel

slnode ตำแหน่งของบัพในแถวลำดับ ซึ่งเป็นบัพที่ถูกเลือกด้วยเมาส์

3. VOID NEAR PaintTree(HCTL hctl, HWND hwnd, HDC hdc)

ใช้ในการแสดงผลของโครงสร้างข้อมูลออกมาในรูปแบบแผนภาพของต้นไม้ โดยวาดแต่ละบัพออกมาอย่างต่อเนื่อง ไม่มีการเว้นช่วงระยะเวลาระหว่างบัพ ฟังก์ชันนี้ถูกเรียกใช้เมื่อค่าคุณสมบัติ DisplayTree เป็น True

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hctl แฮนเดิลของตัวควบคุม

hwnd แฮนเดิลของวินโดว์ของตัวควบคุม

hdc แฮนเดิลของลักษณะอุปกรณ์ (Device Context)

4. VOID NEAR PaintNodTre(HCTL hctl, HWND hwnd, HDC hdc, int nodei, LPSTR Text1, LPSTR Text2)

ใช้ฟังก์ชันนี้ในการวาดบัพที่กำหนด โดยวาดในรูปแบบของโครงสร้างข้อมูลแบบต้นไม้ ฟังก์ชันนี้ใช้ในการวาดบัพเพื่อแสดงให้เห็นถึงการเปลี่ยนแปลงของข้อมูลในบัพนั้นและฟังก์ชันนี้จะถูกเรียกใช้เมื่อกำหนดค่าของคุณสมบัติ DisplayTree เป็น True

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hctl แฮนเดิลของตัวควบคุม

hwnd แฮนเดิลของวินโดว์ของตัวควบคุม

hdc แฮนเดิลของลักษณะอุปกรณ์ (Device Context)

nodei ตำแหน่งของบัพในแถวลำดับ ซึ่งเป็นบัพที่จะถูกวาดในฟังก์ชันนี้

Text1 ค่าเดิมของบัพที่จะถูกวาดออกมาก่อน

Text2 ค่าใหม่ของบัพที่จะถูกวาดออกมาทีหลัง

<u>แฟ้ม PAINTLN.CPP</u>

1. VOID NEAR PaintTimLin(HCTL hctl, HWND hwnd, HDC hdc, BOOL dtime)

ใช้ในการแสดงผลของโครงสร้างข้อมูลในรูปแบบของแถวลำดับ โดยมีระยะเวลาในการแสดงระหว่างบัพแต่ละบัพ ซึ่งจะถูกกำหนดโดยค่าคุณสมบัติ NodeTime และฟังก์ชันนี้จะถูกเรียกใช้เมื่อกำหนดค่าคุณสมบัติ DisplayTree เป็น False

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hctl แฮนเดิลของตัวควบคุม

hwnd แฮนเดิลของวินโดว์ของตัวควบคุม

hdc แฮนเดิลของลักษณะอุปกรณ์ (Device Context)

dtime ตัวบ่งชี้ว่าในการวาดบัพแต่ละบัพ ต้องการให้มีระยะเวลาระหว่างบัพหรือไม่

2. BOOL NEAR InNodeLin(HCTL hctl, HWND hwnd, SHORT xcoord, SHORT ycoord, LPLONG slnode)

ใช้ฟังก์ชันนี้ในการตรวจสอบว่า ตำแหน่งบนตัวควบคุมที่ถูกเลือกโดยการใช้เมาส์ปุ่มข้างขวานั้น มีบัพปรากฏอยู่ที่ตำแหน่งที่เลือกจริงหรือไม่ และถ้ามีบัพอยู่จริง ก็จะเก็บตำแหน่งของบัพในแถวลำดับไว้ในอาร์กิวเมนต์ slnode ฟังก์ชันนี้จะถูกเรียกใช้งานเมื่อค่าคุณสมบัติ DisplayTree ในขณะนั้นเป็น False

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hctl แฮนเดิลของตัวควบคุม

hwnd แฮนเดิลของวินโดว์ของตัวควบคุม

xcoord ตำแหน่งในแนวนอนที่เกิดจากการใช้เมาส์เลือกลงบนตัวควบคุม มีหน่วยเป็น Pixel

ycoord ตำแหน่งในแนวตั้งที่เกิดจากการใช้เมาส์เลือกลงบนตัวควบคุม มีหน่วยเป็น Pixel

slnode ตำแหน่งของบัพในแถวลำดับ ซึ่งเป็นบัพที่ถูกเลือกด้วยเมาส์

3. VOID NEAR PaintLine(HCTL hctl, HWND hwnd, HDC hdc)

ใช้ในการแสดงโครงสร้างข้อมูลออกมาในรูปแบบของแถวลำดับ โดยวาดแต่ละบัพออกมาอย่างต่อเนื่อง ไม่มีการเว้นช่วงระยะเวลาระหว่างบัพ ฟังก์ชันนี้ถูกเรียกใช้เมื่อค่าคุณสมบัติ DisplayTree เป็น False

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hctl แฮนเดิลของตัวควบคุม

hwnd แฮนเดิลของวินโดว์ของตัวควบคุม

hdc แฮนเดิลของลักษณะอุปกรณ์ (Device Context)

4. VOID NEAR PaintNodLin(HCTL hctl, HWND hwnd, HDC hdc, int nodei, LPSTR Text1, LPSTR Text2)

ใช้ฟังก์ชันนี้ในการวาดบัพที่กำหนด โดยวาดโครงสร้างข้อมูลในรูปแบบของแถวลำดับ ฟังก์ชันมีการทำงานเพื่อแสดงให้เห็นถึงการเปลี่ยนแปลงของข้อมูลในบัพนั้น และฟังก์ชันนี้จะถูกเรียกใช้เมื่อค่าของคุณสมบัติ DisplayTree เป็น False

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hctl แฮนเดิลของตัวควบคุม

hwnd แฮนเดิลของวินโดว์ของตัวควบคุม

hdc แฮนเดิลของลักษณะอุปกรณ์ (Device Context)

nodei ตำแหน่งของบัพในแถวลำดับ ซึ่งเป็นบัพที่จะถูกวาดในฟังก์ชันนี้

Text1 ค่าเดิมของบัพที่จะถูกวาดออกมาก่อน

Text2 ค่าใหม่ของบัพที่จะถูกวาดออกมาทีหลัง

5. VOID NEAR InVldNodLn(HCTL hctl, HWND hwnd, HDC hdc)

ใช้ฟังก์ชันนี้ช่วยเตรียมพื้นที่ในการวาดบัพสุดท้ายของโครงสร้างข้อมูล ฟังก์ชันจะถูกเรียกใช้งานเมื่อกำหนดให้การวาดโครงสร้างข้อมูลเป็นแบบแถวลำดับ ซึ่งก็คือค่าคุณสมบัติ DisplayTree เป็น False

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hctl แฮนเดิลของตัวควบคุม

hwnd แฮนเดิลของวินโดว์ของตัวควบคุม

hdc แฮนเดิลของลักษณะอุปกรณ์ (Device Context)

แฟ้ม HPFTN.CPP

ฟังก์ชันที่อยู่ในแฟ้มนี้จะแตกต่างจากฟังก์ชันในแฟ้มอื่น ๆ ที่ได้กล่าวมาแล้ว กล่างคือ ฟังก์ชันที่อยู่ในแฟ้มนี้จะถูกเรียกใช้งานจากภายนอกตัวควบคุม ซึ่งก็คือถูกเรียกใช้จากโปรแกรมในวิชวลเบสิคโดยตรง เพื่อให้ฟังก์ชันเหล่านี้ทำงานแทนบางวิธีที่ไม่สามารถใช้ได้กับตัวควบคุมนี้ รายละเอียดของฟังก์ชันที่อยู่ในแฟ้มนี้สามารถดูจากภาคผนวก ค หัวข้อฟังก์ชันของตัวควบคุม

ภาคผนวก ค

คุณสมบัติ เหตุการณ์ วิธี และฟังก์ชันของตัวควบคุม

ในส่วนของ "วิธีใช้" และ "รูปแบบ" ที่กล่าวถึงในหัวข้อต่าง ๆ จะมีความหมายของลักษณะตัวอักษรดังนี้

- ตัวอักษรปกติ หมายถึง จะต้องมีคำนั้นอยู่ในประโยคเสมอ
- *ตัวอักษรเอียง* หมายถึง ค่าหรือข้อมูลที่ผู้ใช้จะเป็นผู้กำหนดให้เอง เช่น *Ctlname* คือ ชื่อของตัวควบคุมที่ใช้งานอยู่ในขณะนั้น
- *ตัวอักษรเอียง* อยู่ภายในเครื่องหมาย [...] หมายถึง จะใช้หรือไม่ใช้คำนั้นในประโยคก็ได้

คุณสมบัติของตัวควบคุม

คุณสมบัติของตัวควบคุมประกอบด้วย คุณสมบัติมาตรฐาน และคุณสมบัติที่สร้างขึ้นมาใหม่ตามที่ได้กล่าวไว้ในบทที่ 4 ซึ่งมีรายละเอียดของแต่ละคุณสมบัติ ดังนี้

1. คุณสมบัติมาตรฐาน ได้แก่

BackColor Property ใช้กำหนดสีพื้นหลัง (Background) ของตัวควบคุม

วิธีใช้ : *Ctlname*.BackColor = *Color*

การกำหนดค่าของ *Color* จะมีวิธีกำหนดโดยใช้ระบบสี RGB ของวินโดว์ ค่าของตัวเลขที่ใช้ในการกำหนดจะเป็นตัวเลขฐานสิบหกจำนวน 4 ไบต์ ค่าของระบบสี RGB โดยทั่วไปจะมีค่าอยู่ระหว่าง 0 ถึง 16,777,215 (&H00FFFFFF) โดยค่าของไบต์สูงทางด้านซ้าย (High byte) จะมีค่าเป็น 0 และถัดลงไปอีก 3 ไบต์ จะใช้ในการกำหนดค่าของสีแดง เขียว และ น้ำเงินตามลำดับ และค่าขององค์ประกอบแต่ละสีจะมีค่าอยู่ระหว่าง 0 ถึง 255 (&HFF) เช่น &H00000000 แทนสีดำ &H00FFFFFF แทนสีขาว และ &H0000FFFF แทนสีเหลือง เป็นต้น

ค่าโดยปริยาย : ตามค่าของ WINDOW_BACKGROUND ที่กำหนดอยู่ในแฟ้ม CONSTANT.TXT

ForeColor Property ใช้กำหนดสีของตัวอักษรและรูปภาพที่จะแสดงบนตัวควบคุม

วิธีใช้ : *Ctlname*.ForeColor = *Color*

การกำหนดค่าสีของคุณสมบัติ ForeColor จะมีวิธีการกำหนดโดยใช้ระบบสี RGB ของวินโดว์เช่นเดียวกันกับคุณสมบัติ BackColor

ค่าโดยปริยาย : ตามค่าของ WINDOW_TEXT ที่กำหนดอยู่ในแฟ้ม CONSTANT.TXT

Ctlname Property ใช้กำหนดชื่อของตัวควบคุมเพื่อการอ้างอิงในการเขียนโปรแกรมวิชวลเบสิค คุณสมบัตินี้ไม่สามารถใช้ได้ในขณะดำเนินงานของโปรแกรม (Run time)

ค่าโดยปริยาย : ใช้ชื่อของชนิดของตัวควบคุมตามด้วยตัวเลขที่ไม่ซ้ำกัน ซึ่งแสดงถึงลำดับที่การสร้างขึ้นมาใช้งานของตัวควบคุมชนิดนั้น

DragIcon Property ใช้กำหนดสัญรูปที่จะแสดงเป็นตัวชี้เมื่อมีการใช้เมาส์ทำการลากและวางตัวควบคุม

วิธีใช้ : *Ctlname*.DragIcon = *icon*

การกำหนดค่า	ความหมาย
(none)	ลูกศรอยู่ภายในกรอบสี่เหลี่ยม
Icon	กำหนดตัวชี้จากแฟ้มสัญรูปที่เป็นแฟ้มประเภท .ICO

ค่าโดยปริยาย : (none)

DragMode Property กำหนดวิธีที่ใช้ในการลากและวาง

วิธีใช้ : *Ctlname*.DragMode = *mode*

การกำหนดค่า	ความหมาย
0	ต้องใช้ Drag Method เพื่อให้เกิดการลากกับตัวควบคุมได้ (Manual)
1	สามารถลากได้ทันที โดยการคลิกเมาส์ซ้ายที่ตัวควบคุม (Automatic)

FontBold, FontItalic, FontStrikethru, FontUnderline Properties กำหนดลักษณะของตัวอักขระที่ใช้แสดงข้อมูลของบัฟในตัวควบคุม โดย FontBold ใช้กำหนดให้แสดงด้วยตัวหนา FontItalic ใช้กำหนดให้แสดงตัวเอียง FontStrikethru ใช้กำหนดให้แสดงเส้นขีดกลางตัวอักขระ FontUnderline ใช้กำหนดให้แสดงด้วยตัวอักขระขีดเส้นใต้

วิธีใช้ : *Ctlname*.FontBold = *boolean*

Ctlname.FontItalic = *boolean*

Ctlname.FontStrikethru = *boolean*

Ctlname.FontUnderline = *boolean*

การกำหนดค่า	ความหมาย
True	กำหนดให้ใช้เป็นลักษณะของตัวอักขระในการแสดงผล
False	กำหนดให้ไม่ใช้เป็นลักษณะของตัวอักขระในการแสดงผล

ค่าโดยปริยาย : True สำหรับ FontBold

False สำหรับ FontItalic, FontStrikethru และ FontUnderline

FontName Property กำหนดชื่อของรูปแบบของตัวอักขระที่ใช้ในการแสดงข้อมูลของบัฟในตัวควบคุม

วิธีใช้ : *Ctlname*.FontName = *font*

จำนวนรูปแบบของตัวอักขระที่จะใช้ได้ขึ้นอยู่กับจำนวนรูปแบบของตัวอักขระที่มีอยู่ในระบบของวินโดว์ ในช่วงดำเนินงานของโปรแกรม เราสามารถหารายชื่อของรูปแบบของตัวอักขระที่จะใช้ได้จากคุณสมบัติ FontCount และ Fonts

ค่าโดยปริยาย : กำหนดโดยระบบของวินโดว์

FontSize Property กำหนดขนาดของตัวอักขระที่ใช้แสดงข้อมูลของบัฟในตัวควบคุม

วิธีใช้ : *Ctlname*.FontSize = *points*

ใช้ตัวเลขในการกำหนดขนาดของตัวอักขระได้ตามต้องการ โดยมีค่าสูงสุดคือ 2048 points (1 Point = 1/72 นิ้ว)

ค่าโดยปริยาย : กำหนดโดยระบบของวินโดว์

Height, Width Property ใช้กำหนดขนาดของตัวควบคุม การวัดขนาดจะวัดจากจุดกึ่งกลางของกรอบของตัวควบคุม หน่วยที่ใช้ขึ้นอยู่กับหน่วยวัดขนาดของฟอร์มที่ตัวควบคุมนั้นตั้งอยู่ โดย Height ใช้กำหนดส่วนสูง Width ใช้กำหนดความกว้างของตัวควบคุม

วิธีใช้ : *Ctlname*.Height = *numericexpression*

Ctlname.Width = *numericexpression*

Left, Top Property ใช้กำหนดตำแหน่งหัวมุมซ้ายบนของตัวควบคุมบนฟอร์ม โดย Left ใช้กำหนดระยะห่างระหว่างขอบซ้ายของตัวควบคุมกับขอบซ้ายของฟอร์ม Top ใช้กำหนดระยะห่างระหว่างขอบบนของตัวควบคุมกับขอบบนของฟอร์ม หน่วยของค่าของตัวเลขที่ใช้ในการกำหนดจะขึ้นอยู่กับระบบพิกัดของฟอร์มที่ตัวควบคุมนั้นตั้งอยู่

วิธีใช้ : *Ctlname*.Left = *X*

Ctlname.Top = *Y*

hWnd Property เป็นค่าแฮนเดิลของวินโดว์ของตัวควบคุม ซึ่งถูกกำหนดค่าโดยวินโดว์ ค่าของ hWnd จะถูกใช้เป็นอาร์กิวเมนต์ในการติดต่อกับฟังก์ชัน API ต่าง ๆ คุณสมบัตินี้ไม่สามารถใช้ได้ในช่วงการออกแบบโปรแกรม (Design time) และสามารถอ่านค่าได้เท่านั้นในช่วงของการดำเนินงาน

วิธีใช้ : *Ctlname*.hWnd

Visible Property กำหนดตัวควบคุมให้เป็นที่มองเห็นได้หรือไม่

วิธีใช้ : *Ctlname*.Visible = *boolean*

การกำหนดค่า	ความหมาย
True	แสดงตัวควบคุมให้มองเห็นได้
False	ซ่อนตัวควบคุมไว้ไม่ให้มองเห็น

ค่าโดยปริยาย : True

2. คุณสมบัติที่สร้างขึ้นมาใหม่ ได้แก่

Animation Property ใช้กำหนดว่า ต้องการให้มีการแสดงการเปลี่ยนแปลงของบัพข้อมูลในแต่ละขั้นตอนการทำงานของการจัดโครงสร้างข้อมูลให้เป็นฮีปหรือไม่

วิธีใช้ : *Ctlname*.Animation = *boolean*

การกำหนดค่า	ความหมาย
False	ไม่มีการแสดงการเปลี่ยนแปลงของบัพในระหว่างขั้นตอนการทำงาน
True	มีการแสดงการเปลี่ยนแปลงของบัพข้อมูลในระหว่างขั้นตอนการทำงานของการจัดโครงสร้างข้อมูล

ค่าโดยปริยาย : False

เมื่อกำหนดให้คุณสมบัตินี้เป็น True แล้ว ระยะเวลาที่ใช้ในการแสดงการเปลี่ยนแปลงของแต่ละบัพข้อมูล จะถูกกำหนดโดยค่าคุณสมบัติ NodeTime

BuildHeap Property ใช้กำหนดว่า ต้องการให้มีการจัดโครงสร้างข้อมูลให้เป็นฮีปหรือไม่

วิธีใช้ : *Ctlname*.BuildHeap = *boolean*

การกำหนดค่า	ความหมาย
False	ไม่มีการจัดโครงสร้างข้อมูลให้เป็นฮีป
True	กำหนดให้จัดโครงสร้างข้อมูลให้เป็นฮีป

เมื่อมีการเพิ่ม, เปลี่ยนแปลงหรือลบบัพข้อมูล จะมีการจัดให้โครงสร้างข้อมูลเป็นฮีป เมื่อกำหนดค่าคุณสมบัตินี้เป็น True และการกำหนดค่าคุณสมบัติ HeapOrder ซึ่งใช้ในการกำหนดวิธีการจัดลำดับภายในโครงสร้างข้อมูล จะถูกใช้งานก็ต่อเมื่อคุณสมบัติ BuildHeap ได้ถูกกำหนดให้เป็น True ด้วย

ค่าโดยปริยาย : False

CharPerNode Property ใช้กำหนดจำนวนตัวอักขระที่จะนำมาใช้ในการแสดงผลบนตัวควบคุมต่อหนึ่งบัพข้อมูล

วิธีใช้ : *Ctlname*.CharPerNode = *numericexpression*

คุณสมบัตินี้จะถูกนำมาใช้พิจารณาในการแสดงผลเมื่อค่าคุณสมบัติ DisplayTree เป็น False ซึ่งคือการกำหนดให้แสดงโครงสร้างข้อมูลเป็นแบบแถวลำดับ

DisplayByKey Property ใช้เลือกว่าจะให้นำข้อมูลส่วนที่เป็นคีย์หรือข้อมูลส่วนที่เป็นตัวแทนของบัพมาใช้ในการแสดงผลบนตัวควบคุม

วิธีใช้ : *Ctlname*.DisplayByKey = *boolean*

การกำหนดค่า	ความหมาย
False	ให้นำข้อมูลส่วนที่เป็นตัวแทนของบัพมาใช้ในการแสดงผล
True	ให้นำข้อมูลที่เป็นคีย์ของบัพมาใช้ในการแสดงผล

ค่าโดยปริยาย : True

ในกรณีที่กำหนดให้นำข้อมูลส่วนที่เป็นคีย์มาใช้ในการแสดงผล ตัวควบคุมจะนำรูปแบบของตัวเลขที่ไม่เกินจำนวน 5 หลักมาใช้ในการแสดงผล

DisplayTree Property ใช้กำหนดรูปแบบของการแสดงผลโครงสร้างข้อมูลบนตัวควบคุม

วิธีใช้ : *Ctlname*.DisplayTree = *boolen*

การกำหนดค่า	ความหมาย
False	กำหนดให้แสดงผลโครงสร้างข้อมูลเป็นแบบแถวลำดับ
True	กำหนดให้แสดงผลโครงสร้างข้อมูลเป็นแผนภาพต้นไม้

ค่าโดยปริยาย : True

Del_val Property ใช้ลบบัพข้อมูลออกจากโครงสร้างข้อมูล โดยบัพที่ถูกลบจะเป็นบัพแรกที่พบว่ามีค่าคีย์เท่ากับค่าที่กำหนดอยู่ในคุณสมบัตินี้

วิธีใช้ : *Ctlname*.Del_val = *numericexpression*

คุณสมบัตินี้ใช้ได้เฉพาะในช่วงการดำเนินงานของโปรแกรมเท่านั้น

HeapOrder Property ใช้กำหนดลำดับในการจัดโครงสร้างข้อมูลให้เป็นฮีป

วิธีใช้ : *Ctlname*.HeapOrder = *order*

การกำหนดค่า	ความหมาย
0	กำหนดให้บัพรากมีค่าคีย์มากที่สุด หรือ ค่าคีย์ที่มากกว่าอยู่บน
1	กำหนดให้บัพรากมีค่าคีย์น้อยที่สุด หรือ ค่าคีย์ที่น้อยกว่าอยู่บน

ค่าโดยปริยาย : 0

คุณสมบัตินี้จะถูกนำมาใช้กำหนดการจัดลำดับก็ต่อเมื่อคุณสมบัติ BuildHeap มีค่าเป็น True

InputNode Property ใช้ทดลองเพิ่มบัพข้อมูลใหม่เข้าไปในโครงสร้างข้อมูลในช่วงของการออกแบบโปรแกรม ใช้คุณสมบัติได้โดยการใช้เมาส์คลิกที่ข้อคุณสมบัตินี้ในหน้าต่างคุณสมบัติโดยตรง ตัวควบคุมจะแสดงกรอบรับข้อความ (Dialog Box) ขึ้นมาเพื่อรอรับข้อมูลของบัพใหม่

Node_rep Property ใช้กำหนดค่าของส่วนที่เป็นตัวแทนของบัพใหม่ที่จะเพิ่มเข้าไปในโครงสร้างข้อมูล

วิธีใช้ : *Ctlname*.Node_rep = *stringexpression*

คุณสมบัตินี้ใช้กำหนดค่าได้เฉพาะในช่วงการดำเนินงานของโปรแกรม

Node_val Property ใช้กำหนดค่าคีย์ของบัพใหม่ที่จะเพิ่มเข้าไปในโครงสร้างข้อมูล

วิธีใช้ : *Ctlname*.Node_val = *numericexpression*

คุณสมบัตินี้ใช้กำหนดค่าได้เฉพาะในช่วงการดำเนินงานของโปรแกรม คุณสมบัตินี้ต้องใช้กำหนดค่าคู่กับคุณสมบัติ Node_rep โดยต้องกำหนดค่าคุณสมบัติ Node_rep ก่อน แล้วตามด้วยการกำหนดค่าคุณสมบัติ Node_val จึงจะเป็นการเพิ่มบัพข้อมูลใหม่เข้าไปในโครงสร้างข้อมูล

NodeTime Property ใช้กำหนดระยะเวลาที่ใช้ในการแสดงผลบัพข้อมูล เช่น การแสดงบัพข้อมูลใหม่ที่ค่อย ๆ กระพริบและปรากฏขึ้นมาบนตัวควบคุม หรือการกระพริบแสดงการเปลี่ยนแปลงของข้อมูลในบัพจากข้อมูลเก่ามาเป็นข้อมูลใหม่ เป็นต้น

วิธีใช้ : *Ctlname*.NodeTime = *numericexpression*

ค่าโดยปริยาย : 0

เหตุการณ์ของตัวควบคุม

เหตุการณ์ของตัวควบคุมประกอบด้วย เหตุการณ์มาตรฐาน และ เหตุการณ์ที่สร้างขึ้นใหม่ตามที่ได้กล่าวไว้ในบทที่ 4 ซึ่งมีรายละเอียดของแต่ละเหตุการณ์ดังนี้

1. เหตุการณ์มาตรฐาน ได้แก่

Click Event เกิดขึ้นเมื่อมีการคลิกเมาส์ที่ตัวควบคุม

รูปแบบ : Sub *Ctlname*_Click (*Index* As Integer)

อาร์กิวเมนต์ *Index* เป็นหมายเลขของแต่ละตัวควบคุม จะถูกส่งเข้ามาในกรณีที่เป็นตัวควบคุมแบบแถวลำดับ ถ้าต้องการทราบว่า มีการคลิกเมาส์ด้วยปุ่มใด ให้ใช้เหตุการณ์ MouseDown, MouseUp แทน

DblClick Event เกิดขึ้นเมื่อมีการคลิกเมาส์ 2 ครั้ง ติดต่อกันที่ตัวควบคุม

รูปแบบ : Sub *Ctlname*_DblClick (*Index* As Integer)

อาร์กิวเมนต์ *Index* เป็นหมายเลขของแต่ละตัวควบคุม จะถูกส่งเข้ามาในกรณีที่เป็นตัวควบคุมแบบแถวลำดับ ถ้าต้องการทราบว่า มีการคลิกเมาส์ด้วยปุ่มใด ให้ใช้เหตุการณ์ MouseDown, MouseUp แทน

DragDrop Event เกิดขึ้นเมื่อการลากและวางตัวควบคุมเสร็จสมบูรณ์

รูปแบบ : Sub *Ctlname*_DragDrop ([*Index* As Integer,] *Source* As Control, *X* As Single, *Y* As Single)

อาร์กิวเมนต์	ความหมาย
Index	หมายเลขของแต่ละตัวควบคุม ถูกส่งเข้ามาในกรณีที่เป็นตัวควบคุมแบบแถวลำดับ
Source	ตัวควบคุมที่กำลังถูกลาก เราสามารถอ้างถึงคุณสมบัติและวิธีของตัวควบคุมนั้นด้วยอาร์กิวเมนต์นี้ เช่น *Source*.Visible = 0
X, *Y*	ตำแหน่งปัจจุบันในแนวนอนและแนวตั้งของเมาส์ภายในตัวควบคุมเป้าหมายโดยค่าของ *X*, *Y* จะขึ้นอยู่กับระบบพิกัดของตัวควบคุม ซึ่งถูกกำหนดโดยคุณสมบัติ ScaleHeight, ScaleWidth, ScaleLeft และ ScaleTop

ใช้เหตุการณ์นี้ในการควบคุมการทำงานที่จะให้เกิดขึ้นหลังจากการลากตัวควบคุมเสร็จสมบูรณ์ เช่นเคลื่อนย้ายตัวควบคุมไปที่ตำแหน่งใหม่ หรือทำสำเนาแฟ้มข้อมูลจากที่หนึ่งไปยังอีกที่หนึ่ง เป็นต้น

DragOver Event เกิดขึ้นเมื่อกำลังลากและวางตัวควบคุม

รูปแบบ : Sub *Ctlname*_DragOver ([*Index* As Integer,] *Source* As Control, *X* As Single, *Y* As Single, *State* As Integer)

อาร์กิวเมนต์	ความหมาย
Index	หมายเลขของแต่ละตัวควบคุม ถูกส่งเข้ามาในกรณีที่เป็นตัวควบคุมแบบแถวลำดับ
Source	ตัวควบคุมที่กำลังถูกลาก เราสามารถอ้างถึงคุณสมบัติและวิธีของตัวควบคุมนั้นด้วยอาร์กิวเมนต์นี้ เช่น *Source*.Visible = 0

X, Y	ตำแหน่งปัจจุบันในแนวนอนและแนวตั้งของเมาส์ภายในตัวควบคุมเป้าหมายโดยค่าของ *X, Y* จะขึ้นอยู่กับระบบพิกัดของตัวควบคุม ซึ่งถูกกำหนดโดยคุณสมบัติ ScaleHeight, ScaleWidth, ScaleLeft และ ScaleTop
State	สถานะของตัวควบคุมที่กำลังถูกลากและวางภายในฟอร์มที่ตัวควบคุมนั้นตั้งอยู่ 0 - ตัวควบคุมกำลังถูกลากอยู่ในฟอร์มเป้าหมาย 1 - ตัวควบคุมถูกลากออกจากฟอร์มเป้าหมาย 2 - ตัวควบคุมได้ถูกลากจากตำแหน่งหนึ่งไปยังอีกตำแหน่งหนึ่งภายในฟอร์มเป้าหมายแล้ว

ใช้เหตุการณ์นี้ในการพิจารณาว่าจะทำอะไร เมื่อเริ่มทำการลากตัวควบคุม และก่อนที่ตัวควบคุมจะถูกวางลงตำแหน่งเป้าหมาย เช่น ตรวจสอบว่าตัวควบคุมถูกลากเข้าไปในตำแหน่งที่ถูกต้องหรือไม่ หรือแสดงสัญรูปของตัวชี้ให้เป็นรูปที่แตกต่างไปจากเดิมในขณะที่กำลังลากตัวควบคุมอยู่ เป็นต้น

นอกจากนั้น ยังสามารถใช้อาร์กิวเมนต์ *State* มากำหนดการทำงานในสถานะต่าง ๆ ได้ เช่น เมื่อค่า State เป็น 0 เราสามารถทำแสงสว่างบนพื้นที่ที่ตัวควบคุมจะสามารถไปตั้งอยู่ได้ และกลับคืนตามลักษณะเดิมเมื่อค่า State เป็น 1

เมื่อเกิดเหตุการณ์ DragOver โดยมีค่า *State* = 0 และถ้าตัวควบคุมได้ถูกวางลงบนเป้าหมายที่ถูกต้อง ก็จะทำให้เหตุการณ์ DragDrop เกิดขึ้น แต่ถ้าตัวควบคุมถูกวางในตำแหน่งที่ไม่ถูกต้อง เหตุการณ์ DragOver ก็จะเกิดขึ้นอีกด้วยอาร์กิวเมนต์ *State* มีค่าเป็น 1

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)

อาร์กิวเมนต์	ความหมาย
Index	หมายเลขของแต่ละตัวควบคุม ถูกส่งเข้ามาในกรณีที่เป็นตัวควบคุมแบบแถวลำดับ

Button	หมายถึงปุ่มของเมาส์ที่ได้กดหรือปล่อย - ค่าเป็น 1 คือปุ่มซ้าย - ค่าเป็น 2 คือปุ่มขวา - ค่าเป็น 4 คือปุ่มกลาง
Shift	สถานะของปุ่ม SHIFT, CTRL และ ALT บนแป้นพิมพ์เมื่อปุ่มเมาส์ที่กำหนดโดย Button ได้ถูกกด - ปุ่ม SHIFT มีค่าเป็น 1 - ปุ่ม CTRL มีค่าเป็น 2 - ปุ่ม ALT มีค่าเป็น 4 ค่าของ *Shift* สามารถเป็นผลรวมของการกดปุ่มมากกว่า 1 ปุ่มก็ได้ เช่น ถ้า *Shift* = 6 ก็หมายความว่า ปุ่ม CTRL และ ALT ถูกกดพร้อมกัน เป็นต้น
X, Y	ตำแหน่งปัจจุบันในแนวนอนและแนวตั้งของเมาส์ภายในตัวควบคุมเป้าหมายโดยค่าของ *X, Y* จะขึ้นอยู่กับระบบพิกัดของตัวควบคุม ซึ่งถูกกำหนดโดยคุณสมบัติ ScaleHeight, ScaleWidth, ScaleLeft และ ScaleTop

เราสามารถใช้เหตุการณ์ทั้งสองนี้ควบคุมให้มีการทำงานตามที่ต้องการเมื่อมีการกดหรือปล่อยปุ่มเมาส์ที่กำหนดได้ ซึ่งจะต่างกับเหตุการณ์ Click และ DblClick ที่ไม่สามารถบอกได้ว่ามีการกดเมาส์ด้วยปุ่มใด นอกจากนั้น ยังสามารถเขียนโปรแกรมควบคุมการทำงานที่มีการใช้ปุ่มเมาส์ร่วมกับการกดบางปุ่มบนแป้นพิมพ์ได้ด้วย

MouseMove Event เกิดขึ้นเมื่อมีการเลื่อนเมาส์บนตัวควบคุม

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

อาร์กิวเมนต์	ความหมาย
Index	หมายเลขของแต่ละตัวควบคุม ถูกส่งเข้ามาในกรณีที่เป็นตัวควบคุมแบบแถวลำดับ

Button ค่าของ Button จะบอกถึงสถานะของการกดปุ่มเมาส์ โดยเป็นลักษณะของบิตฟิลด์ ถ้าปุ่มใดถูกกด บิตประจำปุ่มนั้น จะถูก SET เป็น 1

- บิต 0 หมายถึงปุ่มซ้าย มีค่าเป็น 1
- บิต 1 หมายถึงปุ่มขวา มีค่าเป็น 2
- บิต 2 หมายถึงปุ่มกลาง มีค่าเป็น 4

ค่าของ *Button* อาจเป็นผลรวมที่เกิดจากการกดปุ่มเมาส์มากกว่า 1 ปุ่มหรืออาจไม่มีการกดปุ่มเมาส์เลยก็ได้

Shift เป็นสถานะของการกดปุ่ม SHIFT, CTRL และ ALT ค่าของ *Shift* เป็นลักษณะของบิตฟิลด์ โดยค่าของบิตประจำปุ่มจะถูก SET เป็น 1 เมื่อปุ่มนั้นถูกกด

- ปุ่ม SHIFT มีค่าเป็น 1
- ปุ่ม CTRL มีค่าเป็น 2
- ปุ่ม ALT มีค่าเป็น 4

ค่าของ *Shift* สามารถเป็นผลรวมของการกดปุ่มมากกว่า 1 ปุ่ม เข้ามาได้ เช่น ถ้า *Shift* = 6 ก็หมายความว่า ปุ่ม CTRL และ ALT ถูกกดพร้อมกันเป็นต้น

X, Y ตำแหน่งปัจจุบันในแนวนอนและแนวตั้งของเมาส์ภายในตัวควบคุมเป้าหมาย โดยค่าของ *X, Y* จะขึ้นอยู่กับระบบพิกัดของตัวควบคุม ซึ่งถูกกำหนดโดยคุณสมบัติ ScaleHeight, ScaleWidth, ScaleLeft และ ScaleTop

เหตุการณ์ MouseMove จะเกิดขึ้นอย่างต่อเนื่องในขณะที่ตัวชี้ตำแหน่งของเมาส์เคลื่อนที่ผ่านตัวควบคุม อาร์กิวเมนต์ *Button* ของเหตุการณ์ MouseMove จะแตกต่างจากอาร์กิวเมนต์ *Button* ของเหตุการณ์ MouseDown, MouseUp โดย *Button* ของ MouseMove สามารถบอกได้ว่ามีการกดเมาส์ปุ่มใดบ้าง แต่ *Button* ของ MouseDown, MouseUp บอกได้เพียงปุ่มเดียวเท่านั้น

2. เหตุการณ์ที่สร้างขึ้นมาใหม่ ได้แก่

HpNodeAdd Event เกิดขึ้นเมื่อมีการเพิ่มบัพใหม่เข้าไปในโครงสร้างข้อมูล

รูปแบบ : Sub *Ctlname*_HpNodeAdd ([*Index* As Integer,] *Key* As Integer, *Rep* As String)

อาร์กิวเมนต์	ความหมาย
Index	หมายเลขของแต่ละตัวควบคุม ถูกส่งเข้ามาในกรณีที่เป็นตัวควบคุมแบบแถวลำดับ
Key	ค่าส่วนที่เป็นคีย์ของบัพข้อมูลใหม่ที่ได้เพิ่มเข้าไป
Rep	ค่าส่วนที่เป็นตัวแทนของบัพข้อมูลใหม่ที่ได้เพิ่มเข้าไป

การเพิ่มบัพข้อมูลใหม่สามารถทำได้ 3 วิธีคือ

- การคลิกที่เมาส์ปุ่มข้างซ้ายติดกัน 2 ครั้งที่ตัวควบคุม เพื่อเรียกกรอบรับข้อความขึ้นมาให้ใส่บัพข้อมูลใหม่เพิ่มเข้าไปในโครงสร้างข้อมูล
- การเพิ่มบัพข้อมูลใหม่โดยการกำหนดค่าคุณสมบัติ Node_rep และ Node_val
- การเพิ่มบัพข้อมูลใหม่โดยการเรียกใช้ฟังก์ชัน HpAddNode

การเพิ่มบัพข้อมูลใหม่ตามที่กล่าวมาทั้ง 3 วิธี ยังไม่นับรวมถึงการเพิ่มบัพใหม่โดยการรวมบัพข้อมูลเข้ามาจากโครงสร้างข้อมูลอื่น ซึ่งวิธีการนี้จะไม่ทำให้เกิดเหตุการณ์ HpNodeAdd

HpNodeChg Event เกิดขึ้นเมื่อมีการแก้ไขเปลี่ยนแปลงบัพข้อมูลที่อยู่ในโครงสร้างข้อมูล

รูปแบบ : Sub *Ctlname*_HpNodeChg ([*Index* As Integer,] *Key* As Integer, *Rep* As String)

อาร์กิวเมนต์	ความหมาย
Index	หมายเลขของแต่ละตัวควบคุม ถูกส่งเข้ามาในกรณีที่เป็นตัวควบคุมแบบแถวลำดับ
Key	ค่าคีย์ใหม่ของบัพข้อมูลที่ถูกเปลี่ยนแปลง
Rep	ค่าของส่วนที่เป็นตัวแทนของบัพที่ได้รับการแก้ไขเปลี่ยนแปลงแล้ว

การแก้ไขเปลี่ยนแปลงบัพโดยตรงทำได้ ดังนี้

- การใช้เมาส์ปุ่มข้างขวาคลิกติดกัน 2 ครั้ง ที่บัพข้อมูลที่ต้องการจะแก้ไขเปลี่ยนแปลงทำให้มีกรอบรับข้อความนำบัพข้อมูลที่ต้องการขึ้นมารอรับการแก้ไข
- การแก้ไขบัพข้อมูล โดยระบุหมายเลขบัพที่ต้องการแก้ไขผ่านทางฟังก์ชัน HpChgNode

HpNodeDel Event เกิดขึ้นเมื่อมีการลบบัพข้อมูลออกจากโครงสร้างข้อมูล

รูปแบบ : Sub *Ctlname*_HpNodeDel ([*Index* As Integer,] *Key* As Integer, *Rep* As String)

อาร์กิวเมนต์	ความหมาย
Index	หมายเลขของแต่ละตัวควบคุม ถูกส่งเข้ามาในกรณีที่เป็นตัวควบคุมแบบแถวลำดับ

Key	ค่าคีย์ของบัพข้อมูลที่ถูกลบออกจากโครงสร้างข้อมูล
Rep	ค่าของส่วนที่เป็นตัวแทนของบัพที่ถูกลบออกจากโครงสร้างข้อมูล

การลบบัพข้อมูลออกจากโครงสร้างข้อมูลสามารถทำได้ ดังนี้

- การใช้เมาส์ปุ่มข้างขวาคลิกติดต่อกัน 2 ครั้ง ที่บัพข้อมูลที่ต้องการจะลบออก ทำให้มีกรอบรับข้อความ นำบัพข้อมูลที่ต้องการ แสดงขึ้นมาเพื่อรอให้ลบออกจากโครงสร้างข้อมูล
- การลบบัพข้อมูลโดยการใช้คุณสมบัติ Del_val กำหนดค่าคีย์ของบัพที่ต้องการจะลบออกจากโครงสร้างข้อมูล
- การลบบัพข้อมูลโดยใช้ฟังก์ชัน HpPopNode ซึ่งเป็นการลบและอ่านบัพรากออกจากโครงสร้างข้อมูล และฟังก์ชัน HpDelNode ซึ่งเป็นการลบโดยการระบุค่าคีย์ของบัพที่ต้องการจะลบ

การลบบัพข้อมูลตามที่กล่าวมาข้างต้น จะไม่รวมถึงการลบบัพข้อมูลทั้งหมดออกจากโครงสร้างข้อมูลโดยการใช้วิธี Clear ซึ่งจะไม่ทำให้เกิดเหตุการณ์ HpNodeDel

HpNodeRead Event เกิดขึ้นเมื่อมีการอ่านบัพข้อมูลที่อยู่ในโครงสร้างข้อมูล

รูปแบบ : Sub *Ctlname*_HpNodeRead ([*Index* As Integer,]*Nodei* As Integer, *Key* As Integer, *Rep* As String)

อาร์กิวเมนต์	ความหมาย
Index	หมายเลขของแต่ละตัวควบคุม ถูกส่งเข้ามาในกรณีที่เป็นตัวควบคุมแบบแถวลำดับ
Nodei	หมายเลขของบัพหรือตำแหน่งของบัพที่อยู่ในแถวลำดับที่อ่านมาได้
Key	ค่าคีย์ของบัพที่อ่านได้
Rep	ค่าของส่วนที่เป็นตัวแทนของบัพที่อ่านได้

การอ่านบัพข้อมูลทำได้โดยการใช้ฟังก์ชันสำหรับการอ่านบัพของตัวควบคุม ซึ่งได้แก่ ฟังก์ชัน HpGetTopVal, HpGetTopRep, HpGetNode, HpGetLChild, HpGetRChild และ HpGetParent

HpClear Event เกิดขึ้นเมื่อมีการลบบัพข้อมูลทั้งหมดออกจากโครงสร้างข้อมูล ซึ่งทำได้โดยการใช้วิธี Clear

รูปแบบ : Sub *Ctlname*_HpClear (*Index* As Integer)

อาร์กิวเมนต์ *Index* เป็นหมายเลขของแต่ละตัวควบคุม จะถูกส่งเข้ามาในกรณีที่เป็นตัวควบคุมแบบแถวลำดับ

HpMergeIn Event เกิดขึ้นเมื่อมีการใช้ฟังก์ชัน HpMerge ในการคัดลอกโครงสร้างข้อมูลจากตัวควบคุมตัวหนึ่งเข้าไปยังโครงสร้างข้อมูลที่อยู่ในตัวควบคุมอีกตัวหนึ่ง

รูปแบบ : Sub *Ctlname*_HpMergeIn (*Index* As Integer)

อาร์กิวเมนต์ *Index* เป็นหมายเลขของแต่ละตัวควบคุม จะถูกส่งเข้ามาในกรณีที่เป็นตัวควบคุมแบบแถวลำดับ

วิธีของตัวควบคุม

Clear Method ลบบัพข้อมูลทั้งหมดออกจากโครงสร้างข้อมูล

รูปแบบ : *Ctlname*.Clear

Drag Method ใช้เริ่มต้น, สิ้นสุด หรือยกเลิกการลากที่ตัวควบคุม

รูปแบบ : *Ctlname*.Drag [*action*]

ส่วน	ความหมาย
Ctlname	ชื่อของตัวควบคุมที่จะทำการลาก
action	เลขจำนวนเต็ม 0-2 ใช้กำหนดการลาก
	0 - ยกเลิกการลาก
	1 - เริ่มการลาก
	2 - จบการลาก (หรือทำการวางตัวควบคุม)

Move Method ทำการเคลื่อนย้ายตัวควบคุมไปยังตำแหน่งใหม่ภายในฟอร์ม

รูปแบบ : *Ctlname*.Move *left* [, *top* [, *width* [, *height*]]]

ส่วน	ความหมาย
Ctlname	ชื่อของตัวควบคุมที่จะถูกเคลื่อนย้าย
left	เลขจำนวนเต็มของตำแหน่งแนวนอนของกรอบด้านซ้ายของตัวควบคุม
top	เลขจำนวนเต็มของตำแหน่งแนวตั้งของกรอบด้านบนของตัวควบคุม
width	เลขจำนวนเต็มของความกว้างใหม่ของตัวควบคุม
height	เลขจำนวนเต็มของความสูงใหม่ของตัวควบคุม

เราสามารถกำหนดเฉพาะค่า *left* เท่านั้นก็ได้ แต่ถ้าต้องการกำหนดค่าอาร์กิวเมนต์ตัวอื่น ก็จะต้องกำหนดค่าอาร์กิวเมนต์ตัวอื่น ๆ ที่อยู่ก่อนหน้าอาร์กิวเมนต์ที่ต้องการกำหนดค่าด้วย เช่น ถ้าต้องการกำหนดค่า height ก็ต้องกำหนดค่า top และ width ด้วย

Refresh Method ทำการวาดตัวควบคุมอีกครั้งหนึ่ง (Repaint)
รูปแบบ : *Ctlname*.Refresh

ZOrder Method ให้วิชวลเบสิควางตัวควบคุมที่ข้างหน้าหรือข้างหลังตามลำดับ Z (z-order) ในระดับทางด้านรูปภาพ (Graphical Level)
รูปแบบ : *Ctlname*.ZOrder [*position*]

ส่วน	ความหมาย
Ctlname	ชื่อตัวควบคุมที่จะวาง
position	ตำแหน่งที่จะวาง เป็นเลขจำนวนเต็มกำหนดตำแหน่งสัมพัทธ์ของตัวควบคุมที่จะวางกับตัวควบคุมอื่น หากไม่มีการกำหนด หรือกำหนดเป็น 0 ตัวควบคุมจะถูกวางข้างหน้าตามลำดับ Z ถ้ากำหนดเป็น 1 จะถูกวางข้างหลังตามลำดับ Z

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

ฟังก์ชันของตัวควบคุม

ฟังก์ชันที่สร้างขึ้นมานี้สามารถถูกเรียกใช้ได้โดยตรงจากโปรแกรมวิชวลเบสิค เพื่อให้ทำงานแทนบางวิธีซึ่งไม่สามารถใช้ได้กับตัวควบคุมนี้ ฟังก์ชันที่สร้างขึ้นมานี้ ได้แก่

1. HpGetTopVal (HWND hwnd)
   ใช้อ่านค่าคีย์ของบัพราก โดยค่าที่อ่านได้จะคืนออกมาเป็นค่าของฟังก์ชัน
   อาร์กิวเมนต์ของฟังก์ชันได้แก่
   hwnd ค่าแฮนเดิลของวินโดว์ของตัวควบคุม หาค่านี้ได้จากคุณสมบัติ hWnd
2. HpGetTopRep (HWND hwnd, LPSTR noderep)
   ใช้ฟังก์ชันนี้อ่านค่าส่วนที่เป็นตัวแทนของบัพราก

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hwnd ค่าแฮนเดิลของวินโดว์ของตัวควบคุม หาค่านี้ได้จากคุณสมบัติ hWnd

noderep ตัวแปรประเภทสายอักขระ ใช้สำหรับเก็บค่าส่วนที่เป็นตัวแทนของบัพรากที่อ่านได้

3. HpGetCount (HWND hwnd)

ใช้หาค่าจำนวนบัพทั้งหมดที่มีอยู่ในโครงสร้างข้อมูล โดยค่าที่หาได้จะถูกคืนออกมาเป็นค่าของฟังก์ชัน

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hwnd ค่าแฮนเดิลของวินโดว์ของตัวควบคุม หาค่านี้ได้จากคุณสมบัติ hWnd

4. HpGetNode (HWND hwnd, int noden, Node_type FAR * node)

ใช้อ่านบัพข้อมูลโดยการระบุหมายเลขที่ต้องการ

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hwnd ค่าแฮนเดิลของวินโดว์ของตัวควบคุม หาค่านี้ได้จากคุณสมบัติ hWnd

noden หมายเลขของบัพหรือตำแหน่งของบัพในแถวลำดับ ซึ่งเป็นบัพที่ต้องการจะอ่านค่า

node บัพเปล่าใช้สำหรับเก็บค่าข้อมูลของบัพที่อ่านมาได้

5. HpGetLChild (HWND hwnd, int noden, Node_type FAR * node)

ใช้อ่านบัพลูกด้านซ้ายของหมายเลขบัพที่กำหนดเข้ามา

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hwnd ค่าแฮนเดิลของวินโดว์ของตัวควบคุม หาค่านี้ได้จากคุณสมบัติ hWnd

noden หมายเลขบัพที่ต้องการจะให้อ่านบัพลูกทางด้านซ้าย

node บัพเปล่าใช้สำหรับเก็บค่าข้อมูลของบัพที่อ่านมาได้

6. HpGetRChild (HWND hwnd, int noden, Node_type FAR * node)

ใช้อ่านบัพลูกด้านขวาของหมายเลขบัพที่กำหนดเข้ามา

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hwnd ค่าแฮนเดิลของวินโดว์ของตัวควบคุม หาค่านี้ได้จากคุณสมบัติ hWnd

noden หมายเลขบัพที่ต้องการจะให้อ่านบัพลูกทางด้านขวา

node บัพเปล่าใช้สำหรับเก็บค่าข้อมูลของบัพที่อ่านมาได้

7. HpGetParent (HWND hwnd, int noden, Node_type FAR * node)

ใช้อ่านบัพแม่ของหมายเลขบัพที่กำหนดเข้ามา

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hwnd ค่าแฮนเดิลของวินโดว์ของตัวควบคุม หาค่านี้ได้จากคุณสมบัติ hWnd

noden หมายเลขบัพที่ต้องการจะให้อ่านบัพแม่ที่อยู่ด้านบน

node บัพเปล่าใช้สำหรับเก็บค่าข้อมูลของบัพที่อ่านมาได้

8. HpAddNode (HWND hwnd, Node_type FAR * node)

ใช้เพิ่มบัพข้อมูลใหม่เข้าไปในโครงสร้างข้อมูล

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hwnd ค่าแฮนเดิลของวินโดว์ของตัวควบคุม หาค่านี้ได้จากคุณสมบัติ hWnd

node บัพข้อมูลใหม่ที่ต้องการจะเพิ่มเข้าไปในโครงสร้างข้อมูล

9. HpChgNode (HWND hwnd, int noden, Node_type FAR * node)

ใช้แก้ไขเปลี่ยนแปลงบัพข้อมูลโดยกำหนดหมายเลขบัพที่ต้องการจะแก้ไข

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hwnd ค่าแฮนเดิลของวินโดว์ของตัวควบคุม หาค่านี้ได้จากคุณสมบัติ hWnd

noden หมายเลขของบัพข้อมูลที่ต้องการจะแก้ไข

node บัพที่เก็บข้อมูลใหม่ ซึ่งจะถูกคัดลอกเข้าไปแทนที่ข้อมูลเก่าของบัพที่กำหนด

10. HpDelNode (HWND hwnd, int nodekey)

ใช้ลบบัพข้อมูลออกจากโครงสร้างข้อมูลโดยระบุค่าคีย์ของบัพที่ต้องการจะลบออก

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hwnd ค่าแฮนเดิลของวินโดว์ของตัวควบคุม หาค่านี้ได้จากคุณสมบัติ hWnd

nodekey ค่าคีย์ของบัพที่ต้องการจะลบออก โดยบัพแรกที่มีค่าคีย์เท่ากับค่าที่กำหนดเข้ามา จะถูกลบออกจากโครงสร้างข้อมูล

11. HpPopNode (HWND hwnd, Node_type FAR * node)

ใช้อ่านและลบบัพรากออกจากโครงสร้างข้อมูล

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hwnd ค่าแฮนเดิลของวินโดว์ของตัวควบคุม หาค่านี้ได้จากคุณสมบัติ hWnd

node บัพเปล่า ใช้เก็บค่าข้อมูลของบัพรากที่อ่านมาได้

12. HpMerge (HWND hwnd1, HWND hwnd2)

ใช้คัดลอกบัพข้อมูลจากโครงสร้างข้อมูลของตัวควบคุมต้นฉบับ (hwnd2) ไปยังโครงสร้างข้อมูลของตัวควบคุมปลายทาง (hwnd1) โดยจำนวนบัพที่จะคัดลอกมาจะคัดลอกเฉพาะในจำนวนที่ไม่ทำให้จำนวนบัพรวมของโครงสร้างข้อมูลปลายทางมีมากกว่า 1024 บัพ

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hwnd1 แฮนเดิลของวินโดว์ของตัวควบคุมปลายทาง

hwnd2 แฮนเดิลของวินโดว์ของตัวควบคุมต้นฉบับ

13.HpSort (HWND hwnd)

ใช้จัดเรียงลำดับข้อมูลที่อยู่ภายในโครงสร้างข้อมูล โดยใช้ลำดับในการจัดเรียงจากคุณสมบัติ HeapOrder ฟังก์ชันนี้จะทำการจัดเรียงลำดับได้อย่างถูกต้องก็ต่อเมื่อได้กำหนดให้คุณสมบัติ BuildHeap เป็น True ก่อนที่จะเรียกใช้ฟังก์ชันนี้

อาร์กิวเมนต์ของฟังก์ชัน ได้แก่

hwnd ค่าแฮนเดิลของวินโดว์ของตัวควบคุม หาค่านี้ได้จากคุณสมบัติ hWnd

ประวัติผู้เขียน

นายวรชาติ พุทธชาติเสวี เกิดวันที่ 15 กุมภาพันธ์ พ.ศ. 2510 ที่อำเภอบางรัก จังหวัดกรุงเทพฯ สำเร็จการศึกษาปริญญาตรีบริหารธุรกิจบัณฑิต สาขาการจัดการเชิงปริมาณ ภาควิชาพาณิชยศาสตร์ คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2531 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อ พ.ศ. 2535