

การใช้ **คอนโทรล DriveListBox**

ใช้สำหรับแสดงข้อมูลเกี่ยวกับระบบ

คล้ายกับ Drop-down ComboBox ใช้แสดง Drive

ติดตั้งในเครื่องคอมพิวเตอร์ ในการใช้งานตามปกติมักจะใช้ควบคู่กับ

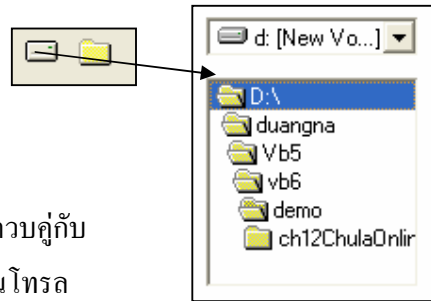
คอนโทรลอีก 2 ตัวเสมอ คือ คอนโทรล FileListBox และคอนโทรล

DirListBox เมื่อผู้ใช้เปลี่ยนแปลงไดรฟ์ (คอนโทรล DriveListBox) ก็จะทำให้ไฟล์เดอร์เปลี่ยน

(คอนโทรล DirListBox) เมื่อไฟล์เดอร์เปลี่ยน รายการเพิ่มข้อมูลต่างๆ ที่อยู่ในไฟล์เดอร์

(คอนโทรล FileListBox) ก็ต้องเปลี่ยนแปลงตามไปด้วย ในการใช้งานคอนโทรล DriveListBox

ควรตั้งชื่อ โดยใช้คำนำหน้าว่า drv

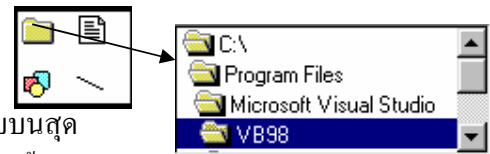


การใช้คอนโทรล **DirListBox**

ใช้แสดงข้อมูลเกี่ยวกับระบบFile Common Dialog Control และ

แสดงโครงสร้างของ Directory ของ Drive ที่กำหนด เริ่มจากระดับบนสุด

และแสดง Sub Directory เยื้องเข้าจาก Directory ที่เหลืออยู่ ในการตั้งชื่อ ควรใช้คำนำหน้าว่า dir



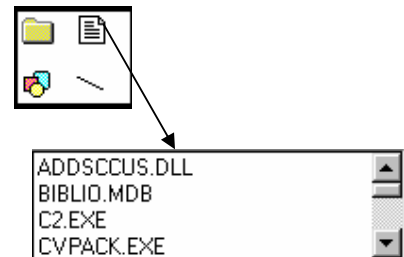
การใช้คอนโทรล **FileListBox**

ใช้สำหรับแสดงข้อมูลเกี่ยวกับระบบFile และ Common Dialog Control

คล้ายกับ DirListBox และจะแสดงชื่อ file ใน Directory ที่กำหนด

Property ของ *FileListBox Control* ที่สำคัญ

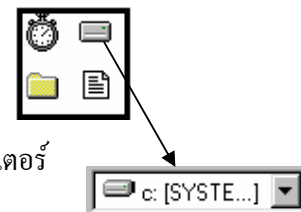
Pattern	แสดง Pattern ของ File ที่จะแสดงใน Control
Hidden	แสดงชื่อ File ที่ File Attribute เป็น Hidden
System	แสดงชื่อ File ที่ File Attribute เป็น System
ReadOnly	แสดงชื่อ File ที่ File Attribute เป็น ReadOnly



การใช้ **DriveListBox**

ใช้สำหรับแสดงข้อมูลเกี่ยวกับระบบ

คล้ายกับ Drop-down ComboBox ใช้แสดง Drive ที่ติดตั้งในเครื่องคอมพิวเตอร์



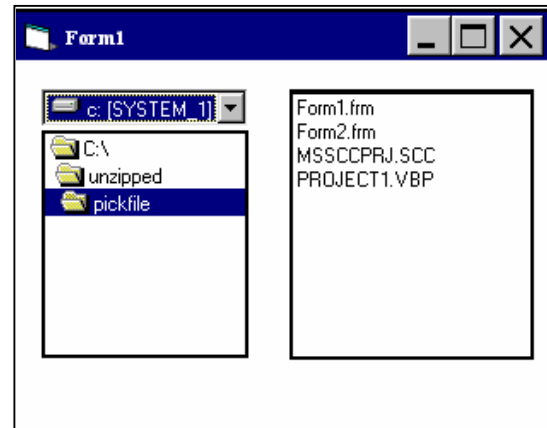
Demo15

Object	Property	Setting
Drive List Box	Name	DrvList
Dir List Box	Name	DirList
File List Box	Name	FilList

```
Private Sub dirList_Change()
```

```
    filList.Path = dirList.Path
```

```
End Sub
```



```
Private Sub drvList_Change()
```

```
    dirList.Path = drvList.Drive
```

```
End Sub
```

Demo 16

```
Private Sub Form_Load()
```

```
    cboPattern.AddItem "Bitmaps (*.bmp)"
```

```
    cboPattern.AddItem "GIF (*.gif)"
```

```
    cboPattern.AddItem "JPEG (*.jpg)"
```

```
    cboPattern.AddItem "Icons (*.ico)"
```

```
    cboPattern.AddItem "Metafiles (*.wmf)"
```

```
    cboPattern.AddItem "DIBs (*.dib)"
```

```
    cboPattern.AddItem "Graphic (*.gif;*.jpg;*.ico;*.bmp;*.wmf;*.dib)"
```

```
    cboPattern.AddItem "All Files (*.*)"
```

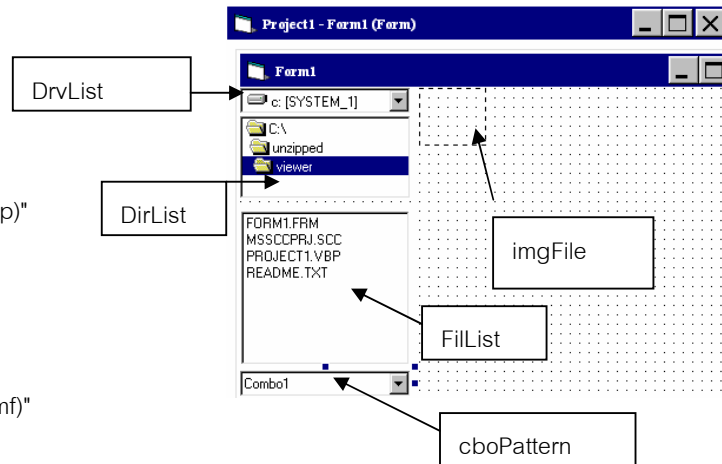
```
    cboPattern.ListIndex = 0
```

```
End Sub
```

```
Private Sub dirList_Change()
```

```
    FilList.Path = dirList.Path
```

```
End Sub
```



```
Private Sub DrvList_Change()
```

```
dirList.Path = DrvList.Drive
```

```
End Sub
```

```
Private Sub FilList_Click()
```

```
Dim fname As String
```

```
On Error GoTo LoadPictureError
```

```
fname = FilList.Path + "\" + FilList.FileName
```

```
Caption = "Viewer [" & fname & "]"
```

```
MousePointer = vbHourglass
```

```
DoEvents
```

```
imgFile.Picture = LoadPicture(fname)
```

```
MousePointer = vbDefault
```

```
Exit Sub
```

```
LoadPictureError:
```

```
Beep
```

```
MousePointer = vbDefault
```

```
Caption = "Viewer [Invalid picture]"
```

```
Exit Sub
```

```
End Sub
```

```
Private Sub cboPattern_Click()
```

```
Dim pat As String
```

```
Dim p1 As Integer
```

```
Dim p2 As Integer
```

```
pat = cboPattern.List(cboPattern.ListIndex)
```

```
p1 = InStr(pat, "(")
```

```
p2 = InStr(pat, ")")
```

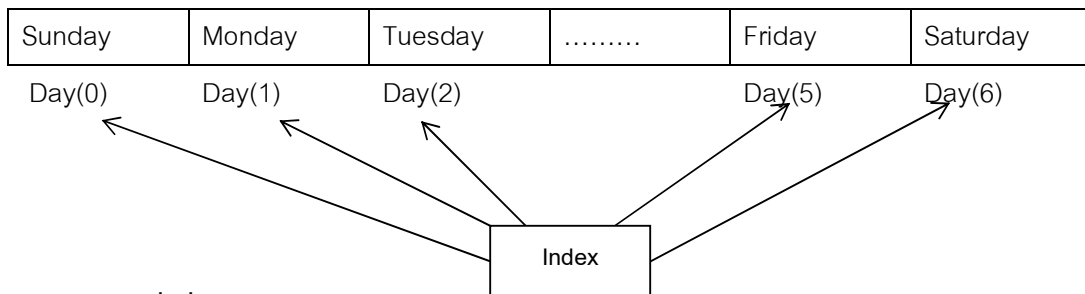
```
FilList.Pattern = Mid$(pat, p1 + 1, p2 - p1 - 1)
```

```
End Sub
```

Array

ข้อมูลแบบ Array เป็น โครงสร้างข้อมูลที่เป็นกลุ่มของข้อมูลชนิดเดียวกัน โดยใน Array จะใช้ index ในการอ้างอิงถึงสมาชิกในกลุ่มของข้อมูล การกำหนดตัวแปรของ Array จะต้องมีการกำหนดขอบเขต เช่น Dim A(4) as Integer แต่ถ้าไม่กำหนดขอบเขต จะมีค่าเท่ากับ 0 ตัวแปรแบบ Array มักใช้ในกรณีที่ต้องการเก็บข้อมูลชนิดเดียวกันจำนวนมาก เช่น เก็บรหัสสมาชิกจำนวน 500 คน ข้อมูลที่เก็บเป็นประเภทเดียวกัน ถ้าใช้ตัวแปรแบบ Array เก็บข้อมูลจะใช้ชื่อตัวแปรตัวเดียว

ลักษณะโครงสร้างแบบ array จะแบ่งตารางเป็นช่องๆ แต่ละช่องสามารถเก็บข้อมูลได้ 1 หน่วย ช่องทุกช่องต้องเก็บข้อมูลที่มีชนิดเดียวกัน และขนาดของแต่ละช่องต้องเท่ากันหมด



จากตัวอย่างสิ่งที่จะต้องกำหนดใน Array คือ

1. ชื่อของ Array คือ Day
2. ขนาดของ Array แต่ละช่อง คือ 9 ตัวอักษร
3. มิติของ Array คือ 1 มิติ
4. ค่าสูงสุด (Upper bound) คือ 6
ค่าต่ำสุด (Lower bound) คือ 0

การเก็บค่าใน Array เช่น

Day(2) = "Tuesday" คือการกำหนดให้สมาชิกตัวที่3 (เริ่มจาก 0) ของ Array Day มีค่าเท่ากับ Tuesday

มิติของ Array มี 3 มิติคือ 1. One Dimension Array 2. Two Dimension Array และ 3. Three Dimension Array

Array 1 มิติ (One Dimension Array)

Array 1 มิติ เป็นการเก็บข้อมูลเรียงต่อกันเป็นแนวเดียว และสามารถกำหนดชื่อตัวแปรเป็นชื่อเดียวกันได้ เช่น 1 dimation มี 5 ค่า แทนที่จะกำหนดเป็น 5 ค่า เช่น test1 test2 test3 test4 test5 จะกำหนดเป็นแบบ Array ได้เช่น

Dim Test(5) as integer

For I = 1 to 5

Test(I) = I * 1

Demo 17 จงสร้าง Form ดังตัวอย่าง

Private Sub cmdOk_Click()

Dim test(5) As Integer

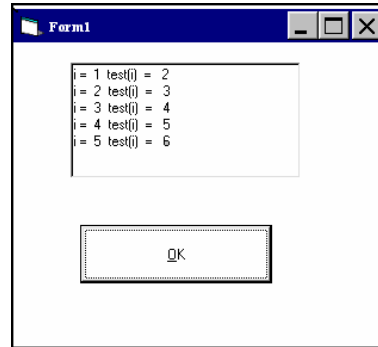
For i = 1 To 5

test(i) = i + 1

picCal.Print "i = " & i & " test(i) = " & test(i)

Next i

End Sub



Array 2 มิติ

Array 2 มิติ คือการเก็บข้อมูลในลักษณะตารางคือมี 2 แนว ในด้านแถว และคอลัมน์

ตัวอย่างการเก็บข้อมูลแบบ Array 2 มิติ

ประเภท \ สาขา	นวนิยาย	หนังสือเด็ก	ตำรา	สารคดี
บางนา	2546	6521	5554	5484
ปิ่นเกล้า	5457	7877	7824	1157
สยามสแควร์	4578	4771	4787	7874

Sale(1,1) = 2,546 Sale(1,2) = 6,521 (ลำดับที่เริ่มต้นในแต่ละด้านเริ่มที่ 0 แต่ในที่นี้

ข้อกำหนดให้ Array เริ่มที่ 1 ใช้คำสั่ง Option Base 1)

การกำหนดตัวแปร Array 2 มิติ สามารถตั้งชื่อได้ เช่น test (2,3)

เป็น integer กำหนดค่า A และ B ดังตัวอย่าง

Dim test(2, 3) As Integer

For A = 1 To 2

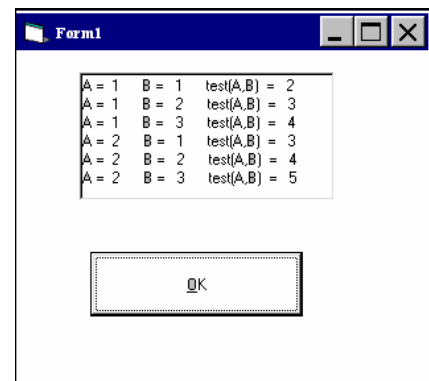
For B = 1 To 3

test(A, B) = A + B

picCal.Print "A = " & A & " B = " & B & " test(A,B) = " & test(A, B)

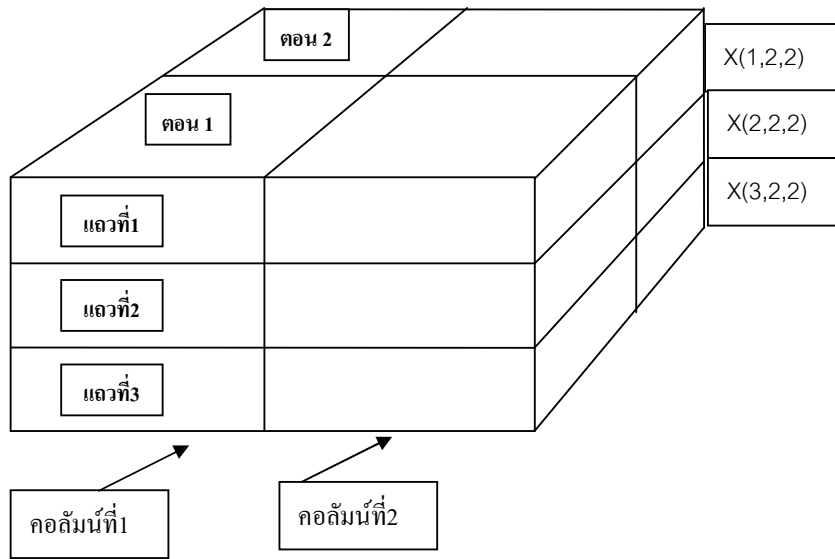
Next B

Next A



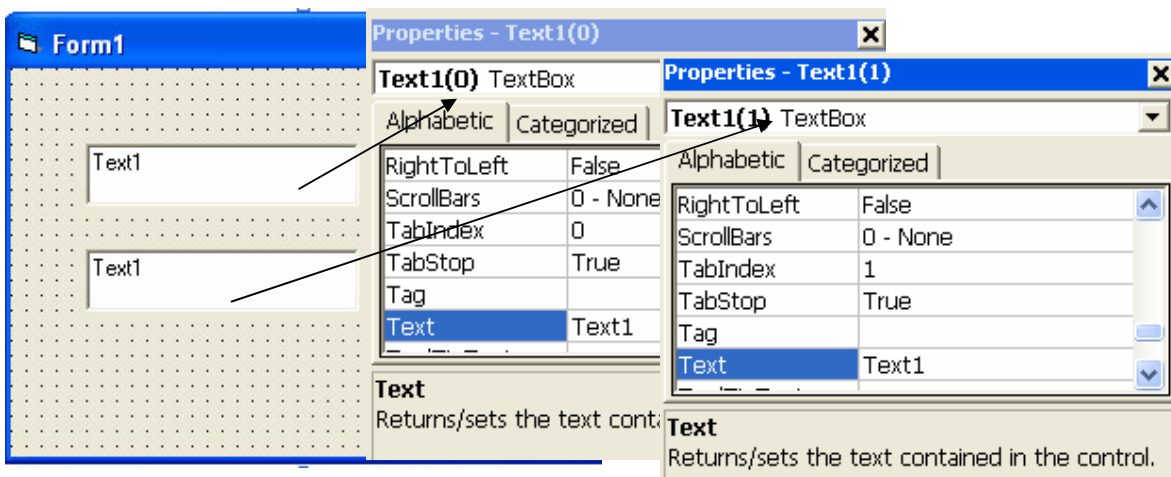
Array 3 มิติ

Array 3 มิติ คือการเก็บข้อมูลแบบ 3 แถว 2 คอลัมน์ 2 ตอน



Control Array

ในการใช้งานตัวแปร สามารถใช้งานในลักษณะแบบปกติ และใช้งานแบบอาร์เรย์ ซึ่งหมายถึง การใช้งานตัวแปรชุด ที่มีการตั้งชื่อเหมือนกัน ต่างกันตรงที่ ตัวแปรแต่ละตัวมีลำดับ (ค่า index) ต่างกัน ซึ่งแนวความคิดนี้เอง สามารถนำมาใช้กับคอนโทรลได้เช่นกัน นั่นคือ สามารถใช้งานคอนโทรลประเภทเดียวกัน โดยที่ตั้งชื่อเหมือนกัน แต่ต่างกันที่คอนโทรลแต่ละตัว จะมีค่า index ที่ทำให้คอนโทรลแต่ละตัวไม่เหมือนกัน ดังตัวอย่าง การสร้าง คอนโทรล text box โดย สร้าง TextBox ที่ 1 แล้วเลือกคำสั่ง Copy , Paste ซึ่งจะได้ไอคอนบ็อกซ์ ยืนยันการใช้งานคอนโทรลอาร์เรย์



Demo 18 จงสร้าง Form และ Control โดยมี property ตามที่กำหนด

Object	Property	Setting
Form	Name	FrmDemo17
	Caption	Control GUI
Label	Name	LblCaption (สร้างเป็น control array 0-3)
	Caption	First Name Last Name Status Occupation
	Name	LblDisplay (สำหรับแสดงข้อมูลที่เลือก)
	Border style	1-fixed Single
	Back Color	White
Text Box	Name	TxtFirstName
	Name	TxtLastName
List Box	Name	LstStatus
	List	Father Item Data: 0 Mother 1 Son 2 Daughter 3 Grandson 4 Granddaughter 5
Frame	Name	FraSex
Option Button	Name	OptSex (สร้างเป็น Control Array 0-1)
	Caption	Male Female
Frame	Name	FraOwner
Check Box	Name	ChkOwner (สร้างเป็น Control Array 0-2)
	Caption	House Condominium Car
Frame	Name	FraAge
Option button	Name	OptAge (สร้างเป็น Control Array 0-3)
	Caption	1-10 years

		11-20 years 21-40 years 41 up
Combo box	Name	CboOccupation
Command button	Name	CmdGUI1 (สร้างเป็น Control Array 0-2)
	Style	1= graphical
	Caption	Display data Clear data Exit
	Picture	รูปภาพ .bmp ตามต้องการ

เมื่อ load Form ต้องสร้างข้อมูลให้แก่ Combo box โดยใช้ method ของ Combo box คือ Additem เช่น

cboOccupation.AddItem "Student" โดยให้สร้างเป็น Sub Procedure เพื่อแยกกลุ่มของงานประเภทเดียวกันไว้ด้วยกัน

```
Private intIndex As Integer      'variable for use as index
Private strSex(0 To 1) As String 'variable for store value of Sex
Private strOwner(0 To 2) As String 'variable for store value of Owner
Private strAge(0 To 3) As String 'variable for store value of Age
```

Private Sub cmdGUI1_Click(Index As Integer)

```
Select Case Index
Case 0      'Display data button
    Call DisplaySelected
Case 1      'Clear data button
    Call ClearAllFields
Case 2      'Exit button
    Unload Me
End Select
```

End Sub

Private Sub Form_Load()

'create display value for Sex,Owner,Age


```

strSex(0) = "Sex=Male"
strSex(1) = "Sex=Female"
strOwner(0) = "Owner of House"
strOwner(1) = "Owner of Condomidium"
strOwner(2) = "Owner of Car"
strAge(0) = "Age = 1-10 years"
strAge(1) = "Age = 11-20 years"
strAge(2) = "Age = 21-40 years"
strAge(3) = "Age = 41 up"

```

'initial variable for use as argument for load occupation in combo box

```

intIndex = 0      'set variable for display first item in the list
Call LoadOccupation(intIndex)

```

End Sub

Private Sub DisplaySelected()

'display selected data process : use chr(10) for new line

```

lblDisplay.Caption = txtFirstName.Text & " " & txtLastName.Text & Chr(10)
lblDisplay.Caption = lblDisplay.Caption & lstStatus.List(lstStatus.ListIndex) & Chr(10)
For intIndex = 0 To 1
    If optSex(intIndex).Value = True Then
        lblDisplay.Caption = lblDisplay.Caption & strSex(intIndex) & Chr(10)
    Exit For
End If
Next intIndex
For intIndex = 0 To 2
    If chkOwner(intIndex).Value = vbChecked Then
        lblDisplay.Caption = lblDisplay.Caption & strOwner(intIndex) & Chr(10)
    End If
Next intIndex
For intIndex = 0 To 3
    If OptAge(intIndex).Value = True Then
        lblDisplay.Caption = lblDisplay.Caption & strAge(intIndex) & Chr(10)
    Exit For
End If
Next intIndex

```

```
lblDisplay.Caption = lblDisplay.Caption & cboOccupation.Text
```

End Sub

Private Sub LoadOccupation(pIndex As Integer)

```
'load occupation value in combo box process
```

```
cboOccupation.AddItem "None"
```

```
cboOccupation.AddItem "Student"
```

```
cboOccupation.AddItem "Business Owner"
```

```
cboOccupation.AddItem "Doctor"
```

```
cboOccupation.AddItem "Nurse"
```

```
cboOccupation.AddItem "Accountant"
```

```
cboOccupation.AddItem "Housewife"
```

```
cboOccupation.AddItem "Programmer"
```

```
cboOccupation.ListIndex = pIndex      'define value to show in combo text
```

End Sub

Private Sub ClearAllFields()

```
'clear all fields on form process
```

```
lblDisplay.Caption = ""
```

```
txtFirstName = ""
```

```
txtLastName = ""
```

```
lstStatus.ListIndex = -1
```

```
For intIndex = 0 To 1
```

```
    optSex(intIndex).Value = False
```

```
Next intIndex
```

```
For intIndex = 0 To 2
```

```
    chkOwner(intIndex).Value = vbUnchecked
```

```
Next intIndex
```

```
For intIndex = 0 To 3
```

```
    OptAge(intIndex).Value = False
```

```
Next intIndex
```

```
cboOccupation.ListIndex = -1
```

End Sub

ชนิดของแฟ้มข้อมูล

ประเภทของแฟ้มข้อมูล มี 3 ประเภท คือ

- **แฟ้มข้อมูลแบบเรียงลำดับ (Sequential access)** การเข้าถึงข้อมูลในแฟ้มข้อมูลแบบนี้ ระบบจะต้องอ่านข้อมูลที่ละตัวอักษร (Byte) ตั้งแต่ต้นแฟ้มข้อมูล จนกระทั่งสิ้นสุดส่วนของแฟ้มข้อมูล (End of file) ซึ่งไม่เหมาะสมกับแฟ้มข้อมูลที่มีขนาดใหญ่หลายๆ เพราะเนื่องจากว่า ถ้าข้อมูลที่ต้องการ อยู่ส่วนท้าย จะทำให้เสียเวลาในการเข้าถึงข้อมูล
- **แฟ้มข้อมูลแบบสุ่ม (Random access)** การเข้าถึงข้อมูล ระบบจะดึงข้อมูลขึ้นมาทีละชุด หรือทีละบล็อก ทำให้การเข้าถึงข้อมูลแบบนี้ มีความยืดหยุ่นมาก เพราะถ้าต้องการข้อมูลตัวใด ก็จะต้องออกมาทั้งบล็อก โดยที่ไม่ต้องอ่านข้อมูลตั้งแต่จุดเริ่มต้นของแฟ้มข้อมูล แต่มีข้อเสียก็คือ ถ้าในกรณี Record นั้นๆ มีการใช้พื้นที่ในการจัดเก็บมากเกินความจำเป็น ก็อาจทำให้สูญเสียพื้นที่ไปโดยเปล่าประโยชน์
- **แฟ้มข้อมูลแบบไบนารี (Binary access)** การเข้าถึงข้อมูลแบบนี้ ระบบจะมองแฟ้มข้อมูลทั้งหมดเป็นบิต (Bit) ซึ่งมีความหมายว่า จะมองทุกๆ ตัวอักษรในข้อมูล เป็นรหัสฐาน 2 ทั้งหมด เมื่อมีการดึงข้อมูล ก็จะอาศัยการเปรียบเทียบในระดับบิตว่า ข้อมูลที่ต้องการอยู่ส่วนใด ก็จะดึงออกมา มีข้อดีคือ ใช้พื้นที่ในการจัดเก็บน้อยมาก

การเปิดแฟ้มข้อมูล

1. ต้องเปิดแฟ้มข้อมูลขึ้นมาก่อน เพื่อให้ระบบทราบก่อนว่า ต้องการดึงข้อมูลจากแฟ้มข้อมูลใด
2. กำหนดวัตถุประสงค์การเปิดแฟ้มข้อมูล เพื่อ อ่าน,เขียน หรือทั้งอ่านและเขียน
3. กำหนดหมายเลขให้กับแฟ้มข้อมูล เพื่อให้ระบบจองหน่วยความจำ ซึ่งเรียกว่า **buffer** ให้กับแฟ้มข้อมูลนั้นๆ ได้ถูกต้อง ซึ่งถ้าในภายหลังมีการประมวลผลกับแฟ้มข้อมูลใด ก็จะสามารถอ้างอิงถึงแฟ้มข้อมูลนั้นได้อย่างถูกต้อง

- การใช้งานคำสั่ง **Open ()**

รูปแบบการใช้งาน

Open pathname **For** mode [**Access access**] [**lock**] **As** [#]filename [**Len=reclength**]

ตัวแปร pathname หมายถึง พาทและชื่อแฟ้มข้อมูลที่ต้องการเปิด

ตัวแปร mode หมายถึง โหมดในการเปิดแฟ้มข้อมูล

- **Append** หมายถึง ต้องการเปิดแฟ้มข้อมูลเพื่อเขียนต่อท้ายแฟ้มข้อมูล

- **Binary** หมายถึง ต้องการเปิดเพิ่มข้อมูลแบบไบนารี
- **Input** หมายถึง ต้องการอ่านข้อมูลในเพิ่มข้อมูลออกมา
- **Output** หมายถึง ต้องการเปิดเพิ่มข้อมูลเพื่อเขียนข้อมูลเข้าไปในเพิ่มข้อมูล
- **Random** หมายถึง ต้องการเปิดเพิ่มข้อมูลแบบสุ่ม

ถ้าไม่กำหนด mode ระบบจะกำหนดการเปิดเพิ่มข้อมูลแบบ Random

ตัวแปร access ใช้กำหนดจุดประสงค์ในการเปิดเพิ่มข้อมูล

- **Read** กำหนดให้เพิ่มข้อมูลที่เปิดขึ้นมา อ่านได้อย่างเดียว
- **Write** กำหนดให้เพิ่มข้อมูลที่เปิดขึ้นมา เขียนข้อมูลได้อย่างเดียว
- **Read Write** กำหนดให้เพิ่มข้อมูลที่เปิดขึ้นมา ทั้งอ่านและเขียน

ตัวแปร lock ใช้กำหนดสิทธิ์ในการใช้งานเพิ่มข้อมูลนั้น (จะกำหนดหรือไม่กำหนดก็ได้)

- **Shared** หมายถึง กำหนดให้เพิ่มข้อมูลนั้น สามารถถูกอ้างอิงหรือเข้าถึงข้อมูลในเพิ่มข้อมูล ได้มากกว่า 1 ครั้ง ในเวลาเดียวกัน ซึ่งเรียกว่า Shared file
- **Lock Read** หมายถึง กำหนดให้เพิ่มข้อมูลนั้น ถูกอ้างอิงหรือเข้าถึงข้อมูลในเพิ่มข้อมูลได้ 1 ครั้ง จากผู้ใช้คนเดียวเท่านั้น เพื่ออ่านข้อมูลจากเพิ่มข้อมูลอย่างเดียว
- **Lock Write** หมายถึง กำหนดให้เพิ่มข้อมูลนั้น ถูกอ้างอิงหรือเข้าถึงข้อมูลในเพิ่มข้อมูลได้ 1 ครั้ง จากผู้ใช้คนเดียวเท่านั้น เพื่อเขียนข้อมูลเข้าไปในเพิ่มข้อมูลอย่างเดียว
- **Lock Read Write** หมายถึง กำหนดให้เพิ่มข้อมูลนั้น ถูกอ้างอิงหรือเข้าถึงข้อมูลในเพิ่มข้อมูลได้ 1 ครั้ง จากผู้ใช้คนเดียวเท่านั้น ทั้งอ่านและเขียนเพิ่มข้อมูล

ตัวแปร # filename ใช้กำหนดหมายเลขให้กับเพิ่มข้อมูล เช่น #1, #2 เป็นต้น ซึ่งสามารถกำหนดหมายเลขได้ตั้งแต่ 1-511 เช่น

Open "name.txt" For Input As #1

หรือใช้ฟังก์ชัน FreeFile () เพื่อให้ระบบ ค้นหาหมายเลขที่ว่างอยู่ในขณะนั้นโดยอัตโนมัติรูปแบบการใช้งานมีดังนี้

FreeFile[(rangenumber)]

ตัวแปร rangenumber หมายถึง เลขจำนวนเต็ม Integer ซึ่งใช้กำหนดระยะสำหรับหาหมายเลขที่ว่างอยู่ โดยที่

- 0 หมายถึง (default) หาหมายเลขที่อยู่ระหว่าง 1-255

- 1 หมายถึง หาหมายเลขที่อยู่ระหว่าง 256-511 เช่น

`FileNumber = FreeFile(1)` 'ค้นหาหมายเลขที่ว่างอยู่'
`Open "test.txt" For Output As #FileNumber` 'กำหนดหมายเลขที่ได้ให้กับแฟ้มข้อมูล'

การปิดแฟ้มข้อมูล

ในทุกๆ ครั้งที่เปิดแฟ้มข้อมูลขึ้นมา จะต้องปิดแฟ้มข้อมูลนั้นเสมอ เพื่อคืนทรัพยากรให้กับระบบ ซึ่งจะใช้คำสั่ง `Close`

- การใช้งานคำสั่ง `Close`

รูปแบบการใช้งาน

`Close [# filenumberlist]`

ตัวแปร `filenumberlist` หมายถึง หมายเลขแฟ้มข้อมูลที่ต้องการปิด อาจจะระบุหรือไม่ก็ได้ ถ้าไม่ระบุ VB จะปิดแฟ้มข้อมูลทุกแฟ้มข้อมูลที่อยู่ในโปรเซสนั้นๆ เช่น