

From libertybasic.conforums.com by Brent Thorn:

This little program demonstrates two concepts.

1. The CascadePosition sub gives your program the ability to open multiple windows arranged in the classic cascaded arrangement. CascadePosition must be called before a window is opened.

2. The TrapCloseCommon\$() function and **CommonClose** sub provide a simple means of multiple window management by maintaining a count of the number of instances of open windows. To use these procedures, you must place the declaration for **g.WindowCount** near the top of your program and instead of doing a #handle "TrapClose etc" you must use #handle TrapCloseCommon\$().

```
'* Cascading windows and instance counting
'* By Brent D. Thorn (http://www.b6sw.com/)
'* PUBLIC DOMAIN

Global g.WindowCount

Call CascadePosition
Open "1" For Window As #1
#1 TrapCloseCommon$()

Call CascadePosition
Open "2" For Window As #2
#2 TrapCloseCommon$()

Call CascadePosition
Open "3" For Window As #3
#3 TrapCloseCommon$()

Wait

Sub CommonClose handle$
    Close #handle$
    g.WindowCount = g.WindowCount - 1

    Print "Closed ";handle$;". ";g.WindowCount;" window(s) open."

    If g.WindowCount = 0 Then
        Print "End of program."
        End
    End If
End Sub
```

```
Function TrapCloseCommon$( )
    TrapCloseCommon$ = "TrapClose CommonClose"
    g.WindowCount      = g.WindowCount + 1
End Function

Sub CascadePosition
    offset = 0

    CallDLL #user32, "GetSystemMetrics", _
        _SM_CYCAPTION As ULong, _
        cy              As Long

    offset = offset + cy

    CallDLL #user32, "GetSystemMetrics", _
        _SM_CYSIZEFRAME As ULong, _
        cy              As Long

    offset = offset + cy

    UpperLeftX = UpperLeftX + offset
    UpperLeftY = UpperLeftY + offset

    If (UpperLeftX + WindowWidth > DisplayWidth) _
        Or (UpperLeftY + WindowHeight > DisplayHeight) Then
        UpperLeftX = 1
        UpperLeftY = 1
    End If
End Sub
```