The FlashWindow Function

The FlashWindow function flashes the specified window one time. This visual color change is done by toggling the window state between active and inactive. The window's caption color reflects this toggling. In addition, the taskbar window button flashes a contrasting color.

```
' Flashing Window Demo #1a
' Flashing the Window Once

    WindowWidth = 320
    WindowHeight = 350
    Button #demo.btn, " Flash Caption ", [FlashCaption], UL, 100, 50,
 100, 30
    Open "Flashing Window Caption" for Window as #demo
    #demo "Trapclose [XbyTrap]"
    hDemo = hWnd(#demo)

Wait

[FlashCaption]
    CallDLL #user32, "FlashWindow", _
        hDemo as uLong, _
            1 as Long, _
            FlashWindow as Long
Wait


[XbyTrap]
    Close #demo
End
```

Flashing a Window

Repeatedly flashing a window grabs the user's attention. From the Microsoft Developer Network (MSDN)

- Flashing a window means changing the appearance of its caption bar as if the window were changing from inactive to active status, or vice versa. (An inactive caption bar changes to an active caption bar; an active caption bar changes to an inactive caption bar.)

- Typically, a window is flashed to inform the user that the window requires attention but that it does not currently have the keyboard focus.

- The FlashWindow function flashes the window only once; for repeated flashing, the application should create a system timer.

Repeated flashing requires a timer. The time intervals may vary according to your preference. Delays of 250 milliseconds to 500 milliseconds work well. This next demo causes Window flashing every 300 milliseconds.

```
' Flashing Window repeatedly using a timer
    WindowWidth = 320
    WindowHeight = 350
    Button #demo.btn, " Stop Flashing ", [StopFlashing], UL, 100, 50,
 100, 30
    Open "Flashing Window Caption" for Window as #demo
    #demo "Trapclose [XbyTrap]"
    hDemo = hWnd(#demo)
    Timer 300, [FlashCaption]

Wait

[FlashCaption]
    CallDLL #user32, "FlashWindow", _
        hDemo as uLong, _
            1 as Long, _
            FlashWindow as Long
Wait

[StopFlashing]
    Timer 0
Wait

[XbyTrap]
    Timer 0
    Close #demo
End
```

Conditional Flashing Using a Timer

A practical use of the FlashWindow function is to get the user's attention *when the application doesn't have keyboard focus*. This final snippet demonstrates the use of flashing when the window is minimized. A timer checks for the iconic state of the window. While the iconic state is 1 (the window has been minimized), the taskbar window button flashes. during an iconic state of 0 (non-minized), no flashing occurs. The first demo uses branch labels as event handlers. The second demo uses subs as event handlers.

The use of WAIT inside the CheckIconic Sub prevents the timer from causing the window to lock-up.

```
' Flashing Window Demo #2a - [BranchLabels] as Event Handlers
' Taskbar Window Button flashes whenever window is minized

    Open "Flashes When Minized" for Window as #demo
    #demo "Trapclose [XbyTrap]"
    hDemo = hWnd(#demo)

    Timer 300, [CheckIconic]

Wait

[CheckIconic]
    IsIconic = IsIconic(hDemo)
    If IsIconic <> 0 Then
        CallDLL #user32, "FlashWindow", _
            hDemo as uLong, _
            1 as Long, _
            result as Long
    End If
Wait

Function IsIconic(hApp)
    CallDLL #user32, "IsIconic", _
        hApp as uLong, _
        0 as Long, _
        IsIconic as Long
End Function

[XbyTrap]
    Timer 0
    Close #demo
End

' Flashing Window Demo #2b - Subs as Event Handlers
' Taskbar Window Button flashes whenever window is minized

    Open "Flashes When Minized" for Window as #demo
    #demo, "Trapclose XbyTrap"
    hDemo = hWnd(#demo)

    Timer 300, CheckIconic
```

```
Wait

Sub CheckIconic
    hApp = hWnd(#demo)
    IsIconic = IsIconic(hApp)
    If IsIconic <> 0 Then
        CallDLL #user32, "FlashWindow", _
            hApp as uLong, _
            1 as Long, _
            result as Long
    End If
Wait ' Prevents timer from causing window lockup
End Sub

Function IsIconic(hApp)
    CallDLL #user32, "IsIconic", _
        hApp as uLong, _
        0 as Long, _
        IsIconic as Long
End Function

Sub XbyTrap handle$
    Timer 0
    Close #demo
End
End Sub
```

Grabbing the User's Attention

If your Liberty BASIC program is working in the background and has encountered a situation where you require user intervention, use the FlashWindow function. Check to see if the handle of your application window matches the return of the GetActiveWindow function. If no match, then set up a timer to flash the window until your application window becomes the active window.