

LỜI NÓI ĐẦU

Khi đọc những lời nói này, tôi nghĩ rằng dù bạn là một người lập trình viên giỏi hay là một người mới chập chững bước vào thế giới của những đoạn code thì chắc hẳn tất cả các bạn đều là những người yêu thích công nghệ thông tin nói chung và lập trình nói riêng. Và bản thân tôi cũng vậy, là một người yêu thích công nghệ thông tin đứng ở khía cạnh này tôi tự nhận thấy để nghiên cứu, học tập, làm việc được tốt thì phải có nhiều tư liệu, sách báo phù hợp với mình. Chính bởi vì lẽ đó, tôi đã soạn cuốn sách này với hy vọng đây là một cuốn sách tra cứu hữu ích bổ sung thêm vào kho tư liệu của mình.

Cuốn sách này được biên soạn từ rất nhiều nguồn tư liệu, và chủ yếu tôi chỉ dùng cho riêng bản thân mình do đó khó tránh khỏi những sai sót, vì thế tôi rất mong nhận được nhiều ý kiến đóng góp của các bạn để những phiên bản sau ngày càng hoàn thiện hơn.

Đồng thời tôi cũng xin tặng cuốn sách này cho tất cả các bạn yêu thích công nghệ thông tin với hy vọng nó sẽ giúp ích một phần nào đó trong công việc của các bạn, nhưng tôi mong các bạn hãy tôn trọng tác giả bằng cách **không chỉnh sửa nội dung, xuất xứ** của cuốn sách. Cuốn sách này hoàn toàn miễn phí, do đó các bạn có thể cho, tặng, biếu bất kỳ người nào nhưng tuyệt đối **cấm thương mại (mua, bán) dưới bất kỳ hình thức nào**.

Mọi chi tiết thắc mắc, góp ý xin vui lòng liên hệ : xla0hu@yahoo.com

Xin cảm ơn !

Phần I
SƠ LƯỢC VỀ WINDOWS API

1. API là gì :

API là viết tắt của **A**pplication **P**rogramming **I**nterface (giao diện lập trình ứng dụng).

API cung cấp hầu hết các tính năng thông dụng cho tất cả các chương trình chạy trên nền Window. Hầu hết các hàm API thường được chứa trong file DLL trong thư mục hệ thống Window (thường là C:/Windows/System).

2. Các thành phần của Windows API :

Có 4 thành phần tạo nên Windows API :

- Các hàm (function) : là thành phần chính của Windows API, chúng được chứa trong các file DLL và có thể được truy xuất một cách dễ dàng bởi các chương trình trên nền Window.
- Các cấu trúc (structure) : nhiều hàm API đòi hỏi một cấu trúc phải được truyền cho chúng để có thể vận chuyển một lượng lớn thông tin mà không cần phải dùng quá nhiều đối số. Các cấu trúc này được dùng trong các hàm API nhưng các bạn phải tự định nghĩa.
- Các hằng được đặt tên (named constant) : cũng như cấu trúc, các hằng phải được định nghĩa rõ ràng trong chương trình.
- Các hàm callback (callback function) : về mặt khái niệm, các hàm callback ngược với các hàm API. Một hàm callback được định nghĩa hoàn toàn trong chương trình của bạn. Sau đó hàm này sẽ được một hàm API khác gọi khi nó thực thi một tác vụ nào đó. Các hàm callback cung cấp một cách thức để chương trình của bạn có thể can thiệp trực tiếp vào một tác vụ nào đó.

3. Cách dùng các hàm API :

a. Cách khai báo :

Các hàm API có 2 dạng: hàm (Function) có trị trả về và thủ tục (Sub) không có trị trả về. Khai báo cho hàm có trị trả về như sau:

Declare Function < tên hàm API > Lib < Tên thư viện > [Alias < tên bí danh >] ([danh sách các đối số]) as < kiểu dữ liệu >

Khai báo cho các thủ tục:

Declare Sub < tên hàm API > Lib < tên thư viện > [Alias < tên bí danh >] ([danh sách các đối số])

Trong đó :

- < tên hàm API > là tên hàm trong các file thư viện DLL.
- < tên thư viện > tên file thư viện DLL để Visual Basic tìm các hàm API. Các file thư viện này phải có đầy đủ tên cùng phần mở rộng, riêng đối với 3 thư viện USER, KERNEL, và GUI thì không cần phải có phần mở rộng. Tên này là một String nên cần phải báo trong dấu "".
- [Alias < tên bí danh >] có thể có hay không cũng được. Bạn cần khai báo bí danh khi muốn triệu gọi hàm API với một cái tên khác do chính bạn đặt, hoặc trong tên hàm chuẩn có chứa ký tự bị cấm sử dụng trong Visual Basic, lúc này bạn hãy đặt bí danh cho nó để Visual Basic sử dụng được.

Ví dụ :

Hàm API "AddfontResource " sau đây được đặt lại bí danh là AddFont cho ngắn gọn mỗi lần gọi hàm.

```
Declare Function AddFontResource Lib "gdi32" Alias "AddFont" (ByVal lpFileName As String) As Long
```

Phạm vi sử dụng của hàm API cũng phụ thuộc vào các vị trí khai báo nó như cách khai báo các biến trong Visual Basic.

Phần sau mệnh đề Lib sẽ báo cho VisualBasic biết file .dll có chứa thủ tục đã khai báo. Đối với các thư viện User32, Kernel32, GDI32 bạn không cần phải ghi thêm phần tên mở rộng của tập tin:

Ví dụ :

```
Declare Function GetTickCount Lib "kernel32" Alias "GetTickCount" () As Long
```

Trong khi đối với những file .dll khác, bạn phải khai báo đường dẫn đầy đủ :

Ví dụ:

```
Declare Function IzCopy Lib "c:\windows\lzexpand.dll" (ByVal S As Integer, ByVal D As Integer) As Long
```

Nếu không khai báo đường dẫn đầy đủ, VisualBasic sẽ tìm lần lượt trong:

1. Thư mục chứa file thực thi .exe
2. Thư mục hiện hành
3. \Windows\System32
4. \Windows\System
5. \Windows
6. Đường dẫn đã được khai báo trước

b. Các file thư viện phổ biến :

Advapi32.dll : Advanced API services library supporting numerous APIs including many security and Registry calls

Comdlg32.dll : Thư viện API dùng cho các hộp thoại (Common dialog API library)

Gdi32.dll : Thư viện API dùng cho giao diện ứng dụng đồ họa

Kernel32.dll : Core Windows 32-bit base API support

Lz32.dll : 32-bit compression routines

Mpr.dll : Multiple Provider Router library

Netapi32.dll : 32-bit Network API library

Shell32.dll : 32-bit Shell API library

User32.dll : Library for user interface routines

Version.dll : Thư viện các phiên bản (Version library)

Winmm.dll : Windows multimedia library

Bốn thư viện chính của Windows :

KERNEL32 : The main DLL, Kernel32, handles memory management, multitasking of the programs that are running, and most other functions which directly affect how Windows actually runs.

USER32 : Windows management library. Contains functions which deal with menus, timers, communications, files and many other non-display areas of Windows.

GDI32 : Graphics Device Interface. Provides the functions necessary to draw things on the screen, as well as checking which areas of forms need to be redrawn.

WINMM : Provides multimedia functions for dealing with sound, music, real-time video, sampling and more. This is a 32-bit only DLL. The 16 bit equivalent is called MMSYSTEM.

4. Handle là gì :

Handle : (cán) tạm gọi là địa chỉ

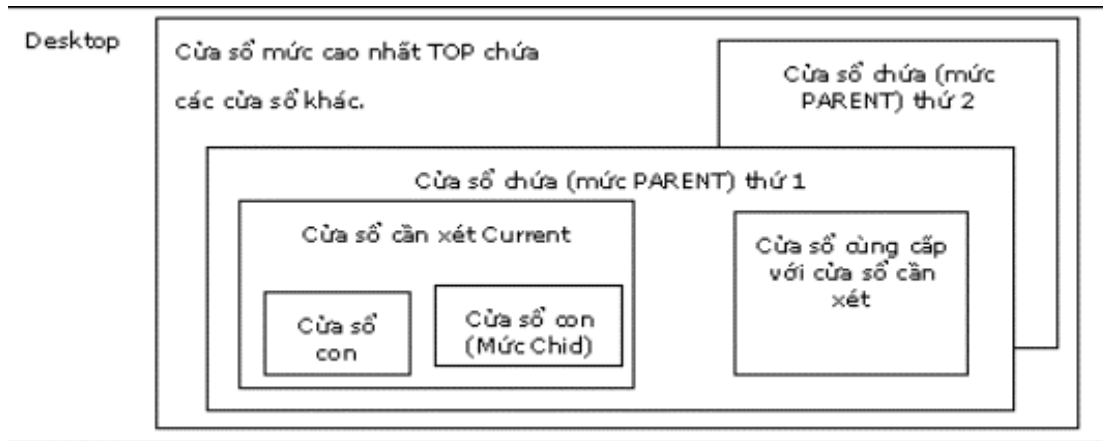
Là một biến kiểu Long có giá trị nhận biết duy nhất dùng để định nghĩa một đối tượng. Và trong Windows thì mỗi đối tượng (control) sẽ được gán cho 1 địa chỉ riêng, giống như số CMND của mình vậy đó, và khi chúng ta muốn làm việc với đối tượng nào thì phải trở tới địa chỉ của đối tượng đó, cái địa chỉ đó được gọi là handle của mỗi đối tượng. Mỗi cửa sổ trong HĐH Windows thì được định nghĩa bởi một handle. Bạn có thể lấy được tất cả các thông tin về một đối tượng sau khi bạn lấy được handle của nó.

5. Có 3 vấn đề chính khi sử dụng và khai thác WinAPI

- Kỹ thuật Subclass: Để cải tổ các đối tượng Visual Basic.
- Kỹ thuật Hook: Câu móc từ chương trình Visual Basic với các chương trình khác. Lấy giá trị nhập vào các chương trình khác của người sử dụng đưa vào chương trình của mình để xử lý.
- Kỹ thuật Multicasting: Dùng một đối tượng tạo lập để theo dõi, chi phối các đối tượng khác của Visual Basic.

Phần II **CÁC HÀM API**

CHƯƠNG I - Các hàm liên quan đến cửa sổ :



(trích dẫn từ trung tâm tin học ABC)

1. AdjustWindowRect

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

Declare Function AdjustWindowRect Lib "user32" Alias "AdjustWindowRect" (lpRect As RECT, ByVal dwStyle As Long, ByVal bMenu As Long) As Long

- Các tham số :
 - **lpRect** : con trỏ tới một cấu trúc RECT chứa vùng làm việc client.
 - **dwStyle** : kiểu cửa sổ
 - **bMenu** : trả về TRUE (khác 0) nếu cửa sổ có menu.
- Mô tả : hàm **AdjustWindowRect** điều chỉnh kích thước của cửa sổ khi có vùng làm việc client (không tính kích thước của thanh tiêu đề, đường viền và các phần thêm). Ngoài ra kích thước của cửa sổ có thể sử dụng hàm CreateWindow để tạo một cửa sổ có diện tích vùng client tùy ý.
- Các hàm liên quan : AdjustWindowRectEx
- Ví dụ minh họa : DeferWindowPos

```
Const WS_BORDER = &H800000
Const WS_DLGFRAME = &H400000
Const WS_THICKFRAME = &H40000
Const WS_CAPTION = &HC00000 ' WS_BORDER Or WS_DLGFRAME
Const HWND_BOTTOM = 1
Const HWND_TOP = 0
Const HWND_TOPMOST = -1
Const HWND_NOTOPMOST = -2
Const SWP_SHOWWINDOW = &H40

Private Type RECT
    Left As Long
    Top As Long
    Right As Long
    Bottom As Long
End Type
```

```
Private Declare Function AdjustWindowRect Lib "user32" (lpRect As RECT, ByVal dwStyle As Long, ByVal bMenu As Long) As Long
Private Declare Function BeginDeferWindowPos Lib "user32" (ByVal nNumWindows As Long) As Long
Private Declare Function DeferWindowPos Lib "user32" (ByVal hWinPosInfo As Long, ByVal hwnd As Long, ByVal hwndInsertAfter As Long, ByVal x As Long, ByVal y As Long, ByVal cx As Long, ByVal cy As Long, ByVal wFlags As Long) As Long
Private Declare Function EndDeferWindowPos Lib "user32" (ByVal hWinPosInfo As Long) As Long

Private Sub Form_Load()
    'KPD-Team 2000
    'URL: http://www.allapi.net/
    'E-Mail: KPDTTeam@Allapi.net
    Dim R As RECT, hDWP As Long
    R.Left = 30
    R.Top = 30
    R.Bottom = 200
    R.Right = 120
    AdjustWindowRect R, WS_THICKFRAME Or WS_CAPTION, False
    hDWP = BeginDeferWindowPos(1)
    DeferWindowPos hDWP, Me.hwnd, HWND_TOP, R.Left, R.Top, R.Right - R.Left, R.Bottom - R.Top, SWP_SHOWWINDOW
    EndDeferWindowPos hDWP
End Sub
```

2. [AdjustWindowRectEx](#)

- *Thư viện* : user32.dll
- *Hệ điều hành* : Windows NT 3.1 or later; Windows 95 or later
- *Khai báo* :

```
Declare Function AdjustWindowRectEx Lib "user32" Alias "AdjustWindowRectEx" (lpRect As RECT, ByVal dwStyle As Long, ByVal bMenu As Long, ByVal dwExStyle As Long) As Long
```

- *Các tham số* :
 - **lpRect** : con trỏ tới một cấu trúc RECT chứa vùng làm việc client.
 - **dwStyle** : kiểu cửa sổ
 - **bMenu** : trả về TRUE (khác 0) nếu cửa sổ có menu.
 - **dwExStyle** : kiểu mở rộng của cửa sổ
- *Mô tả* : hàm **AdjustWindowRectEx** điều chỉnh kích thước của cửa sổ khi có vùng làm việc client (không tính kích thước của thanh tiêu đề, đường viền và các phần thêm). Ngoài ra kích thước của cửa sổ có thể sử dụng hàm CreateWindowEx để tạo một cửa sổ có diện tích vùng client tùy ý.
- *Các hàm liên quan* : AdjustWindowRect
- *Ví dụ minh họa* : AdjustWindowRectEx

```
Const WS_BORDER = &H800000
Const WS_DLGFRAME = &H400000
Const WS_THICKFRAME = &H40000
Const WS_CAPTION = &HC00000 ' WS_BORDER Or WS_DLGFRAME
Const WS_EX_CLIENTEDGE = &H200
Private Type RECT
```



```
Left As Long
Top As Long
Right As Long
Bottom As Long
End Type

Private Declare Function AdjustWindowRectEx Lib "user32" (lpRect As RECT, ByVal dsStyle As Long, ByVal bMenu As Long, ByVal dwExStyle As Long) As Long
Private Declare Function MoveWindow Lib "user32" (ByVal hwnd As Long, ByVal x As Long, ByVal y As Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal bRepaint As Long) As Long

Private Sub Form_Load()
    'KPD-Team 2000
    'URL: http://www.allapi.net/
    'E-Mail: KPDTeam@Allapi.net
    Dim R As RECT, hDWP As Long
    R.Left = 30
    R.Top = 30
    R.Bottom = 200
    R.Right = 120
    AdjustWindowRectEx R, WS_THICKFRAME Or WS_CAPTION, False, WS_EX_CLIENTEDGE
    MoveWindow Me.hwnd, R.Left, R.Top, R.Right - R.Left, R.Bottom - R.Top, False
End Sub
```

3. AnyPopup

- *Thư viện* : user32.dll
- *Hệ điều hành* : Windows NT 3.1 or later; Windows 95 or later
- *Khai báo* :

```
Public Declare Function AnyPopup Lib "user32" Alias "AnyPopup" () As Long
```

- *Mô tả* : Đưa ra chỉ số cửa sổ popup hiện đang tồn tại trên màn hình.
- *Trị trả về* : Hàm trả về TRUE (khác 0) nếu có cửa sổ popup.

4. ArrangelconicWindows

- *Thư viện* : user32.dll
- *Hệ điều hành* : Windows NT 3.1 or later; Windows 95 or later
- *Khai báo*

```
Public Declare Function ArrangelconicWindows Lib "user32" Alias "ArrangelconicWindows" (ByVal hwnd As Long) As Long
```

- *Các tham số* :
 - **hwnd** : handle của cửa sổ chứa (mức parent)
- *Mô tả* : Xếp các biểu tượng cửa sổ trong một cửa sổ chứa (mức parent).
- *Trị trả về* : Hàm trả về số long chiều cao của hàng biểu tượng (bằng 0 nếu thất bại)

5. BeginDeferWindowPos

- *Thư viện* : user32.dll
- *Hệ điều hành* : Windows NT 3.1 or later; Windows 95 or later

- Khai báo :

```
Declare Function BeginDeferWindowPos Lib "user32" Alias "BeginDeferWindowPos" (ByVal nNumWindows As Long) As Long
```

- Các tham số :
 - **nNumWindows** : Số cửa sổ ban đầu để cấp phát vùng nhớ trống.
- Mô tả : Bắt đầu xây dựng danh sách vị trí các cửa sổ mới thành cấu trúc bản đồ nội bộ chứa vị trí các cửa sổ
- Trị trả về : số long – handle của cấu trúc bản đồ. 0 nếu thất bại
- Các hàm liên quan :
 - DeferWindowPos
 - EndDeferWindowPos
- Các ví dụ minh họa : xem ví dụ DeferWindowPos của hàm AdjustWindowRect trang 7.

6. BringWindowToTop

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

```
Declare Function BringWindowToTop Lib "user32" Alias "BringWindowToTop" (ByVal hwnd As Long) As Long
```

- Các tham số :
 - **hwnd** : handle của cửa sổ muốn đưa lên trên cùng (cửa sổ topmost) trong danh sách Z-order
- Mô tả : đưa một cửa sổ chỉ định lên trên cùng trong danh sách thứ tự Z-order (thứ tự hiển thị các cửa sổ), làm cho nó nằm trên tất cả các cửa sổ khác (topmost). Hàm này có tác dụng tương tự như hàm SetWindowPos để đặt cửa sổ lên trên cùng trong Z-order.
- Các hàm liên quan : SetWindowPos
- Các ví dụ minh họa : Window to top

```
'This program needs two forms, two buttons and a module
'KPD-Team 1998
'URL: http://www.allapi.net/
'E-Mail: KPDTTeam@Allapi.net
Private Declare Function BringWindowToTop Lib "user32" (ByVal hwnd As Long) As Long

Private Sub Command1_Click()
    'in form1
    'Set Form2 on top
    BringWindowToTop Form2.hwnd
End Sub

Private Sub Form_Load()
    'in form1
    Form2.Visible = True
End Sub

Private Sub Command1_Click()
    'in form2
    'Set Form1 on top
    BringWindowToTop Form1.hwnd
```

End Sub

7. BrowseCallbackProc

- *Khai báo :*

Public Function BrowseCallbackProc (ByVal hwnd As Long, ByVal uMsg As Long, ByVal
IParam As Long, ByVal lpData As Long) As Long
 ' Đoạn mã xác định ứng dụng đặt tại đây
End Function

- *Các tham số :*

- **hwnd** : handle của cửa duyệt thư mục của hộp thoại Folder đang gọi hàm này. Handle này dùng để giữ các thông điệp cho hộp thoại.
- **uMsg** : một trong các cờ dưới đây xác định các sự kiện
 - **BFFM_INITIALIZED** : hộp thoại hoàn tất khởi tạo, **IParam = 0**
 - **BFFM_SELCHANGED** : người dùng đã thay đổi lựa chọn hiện thời, **IParam là một PIDL** đến lựa chọn hiện thời.
 - **BFFM_VALIDATEFAILED** : từ Internet Explorer 4.0 trở đi : thông báo rằng người dùng nhập một đường dẫn sai vào hộp soạn thảo. **IParam là một con trỏ** trỏ tới một chuỗi (kết thúc bằng ký tự NULL) chứa tên đường dẫn sai này.
- **IParam** : phụ thuộc vào giá trị uMsg .
- **lpData** : giá trị do ứng dụng định nghĩa được trong cấu trúc BROWSEINFO dùng để tạo hộp thoại.

- *Mô tả :* hàm callback BrowseCallbackProc xử lý các thông điệp của cửa sổ duyệt của các hộp thoại Folder. Cụ thể, hàm này sẽ đưa ra những thông điệp thông báo khi hộp thoại đang được khởi chạy và khi người dùng thay đổi các lựa chọn hiện thời. Hàm callback có thể làm việc với hộp thoại đang được khởi chạy và khi người dùng thay đổi các lựa chọn hiện thời. Hàm callback có thể làm việc với hộp thoại bằng cách giữ cho nó một trong những thông điệp sau bằng hàm SendMessage

- **BFFM_ENABLEOK** : Enable hoặc Disable nút OK của hộp thoại. Để Enable nút OK, thiết lập tham số thông điệp **IParam là một giá trị khác 0**. Để Disable nút OK, thiết lập **tham số IParam bằng 0**.
- **BFFM_SETSELECTION** : thiết lập lựa chọn hiện thời trong hộp thoại. Để định rõ đường dẫn như mong muốn bằng cách dùng một chuỗi, thiết lập tham số thông điệp IParam là chuỗi và tham số thông điệp wParam là một số khác 0. Để định rõ đường dẫn theo mong muốn bằng cách sử dụng một con trỏ tới một cấu trúc ITEMIDLIST, thiết lập tham số thông điệp IParam là PIDL và tham số thông điệp wParam là 0.
- **BFFM_SETSTATUSTEXT** : thiết lập trạng thái của văn bản được hiển thị bởi hộp thoại nếu có. Thiết lập tham số thông điệp IParam là chuỗi chứa văn bản mong muốn.

- *Giá trị trả về :*

- Hàm luôn luôn trả về 0 nếu hộp dialog không xử lý thông điệp BFFM_VALIDATEFAILED.
- Hàm trả về 0 để đóng hộp thoại, hoặc trả về giá trị khác 0 để giữ cho nó vẫn được hiển thị.

- *Hằng định nghĩa :*

- Const BFFM_ENABLEOK = &H465
- Const BFFM_SETSELECTION = &H466
- Const BFFM_SETSTATUSTEXT = &H464

- Const BFFM_INITIALIZED = 1
- Const BFFM_SELCHANGED = 2
- Const BFFM_VALIDATEFAILED = 3
- *Ghi chú* : giống như tất cả các hàm callback khác, hàm BrowseCallbackProc phải được khai báo public trong một module.
- *Sử dụng bởi* : BROWSEINFO

8. CallWindowProc

- *Thư viện* : user32.dll
- *Hệ điều hành* : Windows NT 3.1 or later; Windows 95 or later
- *Khai báo* :

```
Declare Function CallWindowProc Lib "user32" Alias "CallWindowProcA" (ByVal  
lpPrevWndFunc As Long, ByVal hWnd As Long, ByVal Msg As Long, ByVal wParam As  
Long, ByVal lParam As Long) As Long
```

- *Các tham số* :
 - **lpPrevWndFunc** : Con trỏ tới hàm thủ tục Window để gọi một cách tường minh. Hàm này dùng để xử lý thông tin. Nếu giá trị này được gọi bởi hàm GetWindowLong với tham số nIndex được thiết lập là GWL_WNDPROC hoặc DWL_DLGPROC, thậm chí nó còn là địa chỉ của cửa sổ hoặc thủ tục của hộp thoại hoặc là handle đại diện cho địa chỉ đó.
 - **hWnd** : một handle cho cửa sổ xử lý thông điệp.
 - **Msg** : thông điệp để xử lý
 - **wParam** : thông tin bổ sung về thông điệp, nội dung của tham số này phụ thuộc vào tham số Msg.
 - **lParam** : thông tin bổ sung về thông điệp, nội dung tham số này phụ thuộc vào tham số Msg.
- *Mô tả* : (gọi tường minh là hàm hook) hoạt động như là một thủ tục của cửa sổ để xử lý một thông điệp. Nó cho phép một thông điệp của cửa sổ được xử lý bằng một thủ tục cửa sổ mà không nhất thiết là thủ tục thường được gọi bởi cửa sổ.
- *Trị trả về* : hàm trả về giá trị được tạo ra sau khi xử lý thông điệp được giữ
- *Các hàm liên quan* :
 - DefMDIChildProc
 - DefWindowProc
- *Các ví dụ minh họa* :
 - + Ví dụ 1 : Clipboard Viewer

```
'Create a new project, add a module to it  
'Add a command button to Form1  
'In the form  
Private Sub Form_Load()  
    'KPD-Team 1999  
    'URL: http://www.allapi.net/  
    'E-Mail: KPDTTeam@Allapi.net  
    'Subclass this form  
    HookForm Me  
    'Register this form as a Clipboardviewer  
    SetClipboardViewer Me.hwnd  
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
    'Unhook the form
    UnHookForm Me
End Sub

Private Sub Command1_Click()
    'Change the clipboard
    Clipboard.Clear
    Clipboard.SetText "Hello !"
End Sub

'In a module
'These routines are explained in our subclassing tutorial.
'http://www.allapi.net/vbtutor/subclass.htm
Declare Function SetWindowLong Lib "user32" Alias "SetWindowLongA" (ByVal hwnd As Long,
ByVal nIndex As Long, ByVal dwNewLong As Long) As Long
Declare Function CallWindowProc Lib "user32" Alias "CallWindowProcA" (ByVal lpPrevWndFunc
As Long, ByVal hwnd As Long, ByVal Msg As Long, ByVal wParam As Long, ByVal lParam As
Long) As Long
Declare Function SetClipboardViewer Lib "user32" (ByVal hwnd As Long) As Long

Public Const WM_DRAWCLIPBOARD = &H308
Public Const GWL_WNDPROC = (-4)

Dim PrevProc As Long

Public Sub HookForm(F As Form)
    PrevProc = SetWindowLong(F.hwnd, GWL_WNDPROC, AddressOf WindowProc)
End Sub

Public Sub UnHookForm(F As Form)
    SetWindowLong F.hwnd, GWL_WNDPROC, PrevProc
End Sub

Public Function WindowProc(ByVal hwnd As Long, ByVal uMsg As Long, ByVal wParam As Long,
ByVal lParam As Long) As Long
    WindowProc = CallWindowProc(PrevProc, hwnd, uMsg, wParam, lParam)
    If uMsg = WM_DRAWCLIPBOARD Then
        MsgBox "Clipboard changed ..."
    End If
End Function
End Function
```

+ Ví dụ 2: Call Procedure

```
'Create a new project and add this code to Form1
Private Declare Function FreeLibrary Lib "kernel32" (ByVal hLibModule As Long) As Long
Private Declare Function LoadLibrary Lib "kernel32" Alias "LoadLibraryA" (ByVal lpLibFileName As
String) As Long
Private Declare Function GetProcAddress Lib "kernel32" (ByVal hModule As Long, ByVal
lpProcName As String) As Long
Private Declare Function CallWindowProc Lib "user32" Alias "CallWindowProcA" (ByVal
lpPrevWndFunc As Long, ByVal hWnd As Long, ByVal Msg As Any, ByVal wParam As Any, ByVal
lParam As Any) As Long

Private Sub Form_Load()
```

```
On Error Resume Next
'KPD-Team 1999
'URL: http://www.allapi.net/
'E-Mail: KPDTTeam@Allapi.net
'We're going to call an API-function, without declaring it!
Dim lb As Long, pa As Long
'map 'user32' into the address space of the calling process.
lb = LoadLibrary("user32")
'retrieve the address of 'SetWindowTextA'
pa = GetProcAddress(lb, "SetWindowTextA")
'Call the SetWindowTextA-function
CallWindowProc pa, Me.hWnd, "Hello !", ByVal 0&, ByVal 0&
'unmap the library's address
FreeLibrary lb
End Sub
```

+ Ví dụ 3: Low Level Audio Stream

```
'This Project needs a module and a form. The Form must contain two buttons, a horizontal scrollbar
and a timer
'Add this code to the form:

Dim fMovingSlider As Boolean

Private Sub Command1_Click()
    Timer1.Interval = 100
    Timer1.Enabled = True
    FileSeek 0
    Play
End Sub

Private Sub Command2_Click()
    Timer1.Enabled = False
    StopPlay
End Sub

Private Sub Form_Load()
    'KPD-Team 1999
    'URL: http://www.allapi.net/
    'E-Mail: KPDTTeam@Allapi.net
    Const sWavFile = "c:\windows\msremind.wav"
    'Check if the file exists
    If Dir(sWavFile) = "" Or sWavFile = "" Then
        HScroll1.Enabled = False
        MsgBox "File not found !" + vbCrLf + "Please choose a valid file !", vbCritical + vbOKOnly
        Exit Sub
    End If
    HScroll1.Max = 100
    HScroll1.SmallChange = 1
    HScroll1.LargeChange = 5
    Command1.Caption = "Play"
    Command2.Caption = "Stop"
    MsgBox "This method uses subclassing, so never press the Stop-button." + vbCrLf + "Unload this
form properly by pressing the 'X'", vbInformation + vbOKOnly
    'Initialize
```

```
Initialize Me.hwnd
'Open the file
OpenFile sWavFile
End Sub

Private Sub Form_Unload(Cancel As Integer)
'If we're playing then stop
If Playing() Then StopPlay
'Close the file
CloseFile
End Sub

Private Sub HScroll1_Change()
If fMovingSlider = True Then
fMovingSlider = False
'move to a new position
FileSeek (HScroll1.Value / 100) * Length
End If
End Sub

Private Sub HScroll1_Scroll()
fMovingSlider = True
End Sub

Private Sub Timer1_Timer()
On Error Resume Next
If (fMovingSlider) Then Exit Sub
If (Playing() = False) Then
Timer1.Enabled = False
End If
HScroll1.Value = (Position() / Length()) * 100
End Sub

'Add this code to a module:
Public Const CALLBACK_WINDOW = &H10000
Public Const MMIO_READ = &H0
Public Const MMIO_FINDCHUNK = &H10
Public Const MMIO_FINDRIFF = &H20
Public Const MM_WOM_DONE = &H3BD
Public Const MMSYSERR_NOERROR = 0
Public Const SEEK_CUR = 1
Public Const SEEK_END = 2
Public Const SEEK_SET = 0
Public Const TIME_BYTES = &H4
Public Const WHDR_DONE = &H1
Type mmioinfo
dwFlags As Long
fccIOProc As Long
pIOProc As Long
wErrorRet As Long
htask As Long
cchBuffer As Long
pchBuffer As String
pchNext As String
pchEndRead As String
pchEndWrite As String
```

```
    lBufOffset As Long
    lDiskOffset As Long
    adwInfo(4) As Long
    dwReserved1 As Long
    dwReserved2 As Long
    hmmio As Long
End Type

Type WAVEHDR
    lpData As Long
    dwBufferLength As Long
    dwBytesRecorded As Long
    dwUser As Long
    dwFlags As Long
    dwLoops As Long
    lpNext As Long
    Reserved As Long
End Type

Type WAVEINCAPS
    wMid As Integer
    wPid As Integer
    vDriverVersion As Long
    szPname As String * 32
    dwFormats As Long
    wChannels As Integer
End Type

Type WAVEFORMAT
    wFormatTag As Integer
    nChannels As Integer
    nSamplesPerSec As Long
    nAvgBytesPerSec As Long
    nBlockAlign As Integer
    wBitsPerSample As Integer
    cbSize As Integer
End Type

Type MMCKINFO
    ckid As Long
    ckSize As Long
    fccType As Long
    dwDataOffset As Long
    dwFlags As Long
End Type

Type MMTIME
    wType As Long
    u As Long
    x As Long
End Type

Declare Function waveOutGetPosition Lib "winmm.dll" (ByVal hWaveOut As Long, lpInfo As MMTIME, ByVal uSize As Long) As Long
Declare Function waveOutOpen Lib "winmm.dll" (hWaveOut As Long, ByVal uDeviceID As Long, ByVal format As String, ByVal dwCallback As Long, ByRef fPlaying As Boolean, ByVal dwFlags As
```



```

Long) As Long
Declare Function waveOutPrepareHeader Lib "winmm.dll" (ByVal hWaveIn As Long, lpWaveInHdr
As WAVEHDR, ByVal uSize As Long) As Long
Declare Function waveOutReset Lib "winmm.dll" (ByVal hWaveIn As Long) As Long
Declare Function waveOutUnprepareHeader Lib "winmm.dll" (ByVal hWaveIn As Long,
lpWaveInHdr As WAVEHDR, ByVal uSize As Long) As Long
Declare Function waveOutClose Lib "winmm.dll" (ByVal hWaveIn As Long) As Long
Declare Function waveOutGetDevCaps Lib "winmm.dll" Alias "waveInGetDevCapsA" (ByVal
uDeviceID As Long, lpCaps As WAVEINCAPS, ByVal uSize As Long) As Long
Declare Function waveOutGetNumDevs Lib "winmm.dll" () As Long
Declare Function waveOutGetErrorText Lib "winmm.dll" Alias "waveInGetErrorTextA" (ByVal err As
Long, ByVal lpText As String, ByVal uSize As Long) As Long
Declare Function waveOutWrite Lib "winmm.dll" (ByVal hWaveOut As Long, lpWaveOutHdr As
WAVEHDR, ByVal uSize As Long) As Long
Declare Function mmioClose Lib "winmm.dll" (ByVal hmmio As Long, ByVal uFlags As Long) As
Long
Declare Function mmioDescend Lib "winmm.dll" (ByVal hmmio As Long, lpck As MMCKINFO,
lpckParent As MMCKINFO, ByVal uFlags As Long) As Long
Declare Function mmioDescendParent Lib "winmm.dll" Alias "mmioDescend" (ByVal hmmio As
Long, lpck As MMCKINFO, ByVal x As Long, ByVal uFlags As Long) As Long
Declare Function mmioOpen Lib "winmm.dll" Alias "mmioOpenA" (ByVal szFileName As String,
lpmmioinfo As mmioinfo, ByVal dwOpenFlags As Long) As Long
Declare Function mmioRead Lib "winmm.dll" (ByVal hmmio As Long, ByVal pch As Long, ByVal cch
As Long) As Long
Declare Function mmioReadString Lib "winmm.dll" Alias "mmioRead" (ByVal hmmio As Long, ByVal
pch As String, ByVal cch As Long) As Long
Declare Function mmioSeek Lib "winmm.dll" (ByVal hmmio As Long, ByVal lOffset As Long, ByVal
iOrigin As Long) As Long
Declare Function mmioStringToFOURCC Lib "winmm.dll" Alias "mmioStringToFOURCCA" (ByVal
sz As String, ByVal uFlags As Long) As Long
Declare Function mmioAscend Lib "winmm.dll" (ByVal hmmio As Long, lpck As MMCKINFO, ByVal
uFlags As Long) As Long
Declare Function GlobalAlloc Lib "kernel32" (ByVal wFlags As Long, ByVal dwBytes As Long) As
Long
Declare Function GlobalLock Lib "kernel32" (ByVal hmem As Long) As Long
Declare Function GlobalFree Lib "kernel32" (ByVal hmem As Long) As Long
Declare Sub CopyStructFromPtr Lib "kernel32" Alias "RtlMoveMemory" (struct As Any, ByVal ptr As
Long, ByVal cb As Long)
Declare Sub CopyPtrFromStruct Lib "kernel32" Alias "RtlMoveMemory" (ByVal ptr As Long, struct
As Any, ByVal cb As Long)
Declare Sub CopyStructFromString Lib "kernel32" Alias "RtlMoveMemory" (dest As Any, ByVal
source As String, ByVal cb As Long)
Declare Function PostWavMessage Lib "user32" Alias "PostMessageA" (ByVal hwnd As Long,
ByVal wParam As Long, ByVal wParam As Long, ByRef hdr As WAVEHDR) As Long
Declare Function CallWindowProc Lib "user32" Alias "CallWindowProcA" (ByVal lpPrevWndFunc
As Long, ByVal hwnd As Long, ByVal msg As Long, ByVal wParam As Long, ByRef lParam As
WAVEHDR) As Long
Declare Function SetWindowLong Lib "user32" Alias "SetWindowLongA" (ByVal hwnd As Long,
ByVal nIndex As Long, ByVal dwNewLong As Long) As Long

Public Const GWL_WNDPROC = -4

Dim lpPrevWndProc As Long

Const NUM_BUFFERS = 5
Const BUFFER_SECONDS = 0.1

```

```
Dim rc As Long      ' Return code
Dim hmmioIn As Long  ' file handle
Dim dataOffset As Long ' start of audio data in wave file
Dim audioLength As Long ' number of bytes in audio data
Dim pFormat As Long   ' pointer to wave format
Dim formatBuffer As String * 50 ' buffer to hold the wave format
Dim startPos As Long  ' sample where we started playback from
Dim format As WAVEFORMAT ' waveformat structure
Dim i As Long         ' loop control variable
Dim j As Long         ' loop control variable
Dim hmem(1 To NUM_BUFFERS) As Long ' memory handles
Dim pmem(1 To NUM_BUFFERS) As Long ' memory pointers
Dim hdr(1 To NUM_BUFFERS) As WAVEHDR ' wave headers
Dim bufferSize As Long ' size of output buffers
Dim fPlaying As Boolean ' is file currently playing
Dim fFileOpen As Boolean ' is file currently open
Dim hWaveOut As Long    ' waveout handle
Dim msg As String * 250 ' message buffer
Dim hwnd As Long        ' window handle

Public Sub Initialize(hwndIn As Long)
    hwnd = hwndIn
    lpPrevWndProc = SetWindowLong(hwnd, GWL_WNDPROC, AddressOf WindowProc)
    fPlaying = False
    fFileOpen = False
    startPos = 0
End Sub

Public Sub CloseFile()
    mmioClose hmmioIn, 0
    fFileOpen = False
End Sub

Public Sub OpenFile(soundfile As String)
    Dim mmckinfoParentIn As MMCKINFO
    Dim mmckinfoSubchunkIn As MMCKINFO
    Dim mmioinf As mmioinfo

    ' close previously open file (if any)
    CloseFile
    If (soundfile = "") Then
        Exit Sub
    End If

    ' Open the input file
    hmmioIn = mmioOpen(soundfile, mmioinf, MMIO_READ)
    If (hmmioIn = 0) Then
        MsgBox "Error opening input file, rc = " & mmioinf.wErrorRet
        Exit Sub
    End If

    ' Check if this is a wave file
    mmckinfoParentIn.fccType = mmioStringToFOURCC("WAVE", 0)
    rc = mmioDescendParent(hmmioIn, mmckinfoParentIn, 0, MMIO_FINDRIFF)
    If (rc <> MMSYSERR_NOERROR) Then
```

```

    CloseFile
    MsgBox "Not a wave file"
    Exit Sub
End If

' Get format info
mmckinfoSubchunkIn.ckid = mmioStringToFOURCC("fmt", 0)
rc = mmioDescend(hmmioIn, mmckinfoSubchunkIn, mmckinfoParentIn, MMIO_FINDCHUNK)
If (rc <> MMSYSERR_NOERROR) Then
    CloseFile
    MsgBox "Couldn't get format chunk"
    Exit Sub
End If
rc = mmioReadString(hmmioIn, formatBuffer, mmckinfoSubchunkIn.ckSize)
If (rc = -1) Then
    CloseFile
    MsgBox "Error reading format"
    Exit Sub
End If
rc = mmioAscend(hmmioIn, mmckinfoSubchunkIn, 0)
CopyStructFromString format, formatBuffer, Len(format)

' Find the data subchunk
mmckinfoSubchunkIn.ckid = mmioStringToFOURCC("data", 0)
rc = mmioDescend(hmmioIn, mmckinfoSubchunkIn, mmckinfoParentIn, MMIO_FINDCHUNK)
If (rc <> MMSYSERR_NOERROR) Then
    CloseFile
    MsgBox "Couldn't get data chunk"
    Exit Sub
End If
dataOffset = mmioSeek(hmmioIn, 0, SEEK_CUR)

' Get the length of the audio
audioLength = mmckinfoSubchunkIn.ckSize

' Allocate audio buffers
bufferSize = format.nSamplesPerSec * format.nBlockAlign * format.nChannels *
BUFFER_SECONDS
bufferSize = bufferSize - (bufferSize Mod format.nBlockAlign)
For i = 1 To (NUM_BUFFERS)
    GlobalFree hmem(i)
    hmem(i) = GlobalAlloc(0, bufferSize)
    pmem(i) = GlobalLock(hmem(i))
Next

fFileOpen = True
End Sub

Public Function Play() As Boolean
    If (fPlaying) Then
        Play = True
        Exit Function
    End If

    rc = waveOutOpen(hWaveOut, 0, formatBuffer, hwnd, True, CALLBACK_WINDOW)

```

```
If (rc <> MMSYSERR_NOERROR) Then
    waveOutGetErrorText rc, msg, Len(msg)
    MsgBox msg
    Play = False
    Exit Function
End If

For i = 1 To NUM_BUFFERS
    hdr(i).lpData = pmem(i)
    hdr(i).dwBufferLength = bufferSize
    hdr(i).dwFlags = 0
    hdr(i).dwLoops = 0
    rc = waveOutPrepareHeader(hWaveOut, hdr(i), Len(hdr(i)))
    If (rc <> MMSYSERR_NOERROR) Then
        waveOutGetErrorText rc, msg, Len(msg)
        MsgBox msg
    End If
Next

fPlaying = True
Play = True

startPos = mmioSeek(hmmioIn, 0, SEEK_CUR) - dataOffset

For i = 1 To NUM_BUFFERS
    PostWavMessage hwnd, MM_WOM_DONE, 0, hdr(i)
Next
End Function

Public Sub StopPlay()
    fPlaying = False
    FileSeek Position()
    waveOutReset hWaveOut
End Sub

Public Function Length() As Long
    Length = audioLength \ format.nBlockAlign
End Function

Public Function FileSeek(Position As Long) As Boolean
    Dim bytupos As Long
    FileSeek = False
    bytupos = Position * format.nBlockAlign
    If (fFileOpen = False) Or (bytupos < 0) Or (bytupos >= audioLength) Then
        Exit Function
    End If
    rc = mmioSeek(hmmioIn, bytupos + dataOffset, SEEK_SET)
    If (rc = MMSYSERR_NOERROR) Then
        FileSeek = True
    End If
    startPos = rc
End Function

Public Function Position() As Long
    Dim tm As MMTIME
    tm.wType = TIME_BYTES
```

```

rc = waveOutGetPosition(hWaveOut, tm, Len(tm))
If (rc = MMSYSERR_NOERROR) Then
    Position = (startPos + tm.u) \ format.nBlockAlign
Else
    Position = (mmioSeek(hmmioIn, 0, SEEK_CUR) - dataOffset + bufferSize * NUM_BUFFERS) \
format.nBlockAlign
End If
End Function

Public Function Playing() As Boolean
    Dim tm As MMTIME
    tm.wType = TIME_BYTES
    rc = waveOutGetPosition(hWaveOut, tm, Len(tm))
    If (rc = MMSYSERR_NOERROR) Then
        Playing = True
    Else
        Playing = False
    End If
End Function

Function WindowProc(ByVal hw As Long, ByVal uMsg As Long, ByVal wParam As Long, ByRef
wavhdr As WAVEHDR) As Long
    Static dataRemaining As Long
    If (uMsg = MM_WOM_DONE) Then
        If (fPlaying = True) Then
            dataRemaining = (dataOffset + audioLength - mmioSeek(hmmioIn, 0, SEEK_CUR))
            If (bufferSize < dataRemaining) Then
                rc = mmioRead(hmmioIn, wavhdr.lpData, bufferSize)
            Else
                rc = mmioRead(hmmioIn, wavhdr.lpData, dataRemaining)
                fPlaying = False
            End If
            wavhdr.dwBufferLength = rc
            rc = waveOutWrite(hWaveOut, wavhdr, Len(wavhdr))
        Else
            For i = 1 To NUM_BUFFERS
                waveOutUnprepareHeader hWaveOut, hdr(i), Len(hdr(i))
            Next
            waveOutClose hWaveOut
        End If
    End If
    WindowProc = CallWindowProc(lpPrevWndProc, hw, uMsg, wParam, wavhdr)
End Function

```

+ Ví dụ 4: Classical

```

'This project needs one form
' Also set StartupObject to 'Sub Main'
' (-> Project Properties -> General Tab -> Startup Object)
'---- Declarations
Declare Function RegisterClass Lib "user32" Alias "RegisterClassA" (Class As WNDCLASS) As
Long
Declare Function UnregisterClass Lib "user32" Alias "UnregisterClassA" (ByVal lpClassName As
String, ByVal hInstance As Long) As Long
Declare Function CreateWindowEx Lib "user32" Alias "CreateWindowExA" (ByVal dwExStyle As

```

```
Long, ByVal lpClassName As String, ByVal lpWindowName As String, ByVal dwStyle As Long,
ByVal x As Long, ByVal y As Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal
hWndParent As Long, ByVal hMenu As Long, ByVal hInstance As Long, lpParam As Any) As Long
Declare Function DefWindowProc Lib "user32" Alias "DefWindowProcA" (ByVal hWnd As Long,
ByVal wParam As Long, ByVal lParam As Long) As Long
Declare Sub PostQuitMessage Lib "user32" (ByVal nExitCode As Long)
Declare Function GetMessage Lib "user32" Alias "GetMessageA" (lpMsg As Msg, ByVal hWnd As
Long, ByVal wParamFilterMin As Long, ByVal wParamFilterMax As Long) As Long
Declare Function TranslateMessage Lib "user32" (lpMsg As Msg) As Long
Declare Function DispatchMessage Lib "user32" Alias "DispatchMessageA" (lpMsg As Msg) As
Long
Declare Function ShowWindow Lib "user32" (ByVal hWnd As Long, ByVal nCmdShow As Long) As
Long
Declare Function LoadCursor Lib "user32" Alias "LoadCursorA" (ByVal hInstance As Long, ByVal
lpCursorName As Any) As Long
Declare Function DefMDIChildProc Lib "user32" Alias "DefMDIChildProcA" (ByVal hWnd As Long,
ByVal wParam As Long, ByVal lParam As Long, ByVal lParam As Long) As Long
' Define information of the window (pointed to by hWnd)
Declare Function GetWindowLong Lib "user32" Alias "GetWindowLongA" (ByVal hWnd As Long,
ByVal nIndex As Long) As Long
Declare Function SetWindowLong Lib "user32" Alias "SetWindowLongA" (ByVal hWnd As Long,
ByVal nIndex As Long, ByVal dwNewLong As Long) As Long
Declare Function CallWindowProc Lib "user32" Alias "CallWindowProcA" (ByVal lpPrevWndFunc
As Long, ByVal hWnd As Long, ByVal Msg As Long, ByVal wParam As Long, ByVal lParam As
Long) As Long

Type WNDCLASS
    style As Long
    lpfnwndproc As Long
    cbClsextra As Long
    cbWndExtra2 As Long
    hInstance As Long
    hIcon As Long
    hCursor As Long
    hbrBackground As Long
    lpszMenuName As String
    lpszClassName As String
End Type

Type POINTAPI
    x As Long
    y As Long
End Type

Type Msg
    hWnd As Long
    message As Long
    wParam As Long
    lParam As Long
    time As Long
    pt As POINTAPI
End Type

' Class styles
Public Const CS_VREDRAW = &H1
Public Const CS_HREDRAW = &H2
```

```
Public Const CS_KEYCVTWINDOW = &H4
Public Const CS_DBLCLKS = &H8
Public Const CS_OWNDC = &H20
Public Const CS_CLASSDC = &H40
Public Const CS_PARENTDC = &H80
Public Const CS_NOKEYCVT = &H100
Public Const CS_NOCLOSE = &H200
Public Const CS_SAVEBITS = &H800
Public Const CS_BYTEALIGNCLIENT = &H1000
Public Const CS_BYTEALIGNWINDOW = &H2000
Public Const CS_PUBLICCLASS = &H4000

' Window styles
Public Const WS_OVERLAPPED = &H0&
Public Const WS_POPUP = &H80000000
Public Const WS_CHILD = &H40000000
Public Const WS_MINIMIZE = &H20000000
Public Const WS_VISIBLE = &H10000000
Public Const WS_DISABLED = &H80000000
Public Const WS_CLIPSIBLINGS = &H40000000
Public Const WS_CLIPCHILDREN = &H20000000
Public Const WS_MAXIMIZE = &H1000000
Public Const WS_CAPTION = &HC00000 ' WS_BORDER Or WS_DLGFRAME
Public Const WS_BORDER = &H800000
Public Const WS_DLGFRAME = &H400000
Public Const WS_VSCROLL = &H200000
Public Const WS_HSCROLL = &H100000
Public Const WS_SYSMENU = &H80000
Public Const WS_THICKFRAME = &H40000
Public Const WS_GROUP = &H20000
Public Const WS_TABSTOP = &H10000
Public Const WS_MINIMIZEBOX = &H20000
Public Const WS_MAXIMIZEBOX = &H10000
Public Const WS_TILED = WS_OVERLAPPED
Public Const WS_ICONIC = WS_MINIMIZE
Public Const WS_SIZEBOX = WS_THICKFRAME
Public Const WS_OVERLAPPEDWINDOW = (WS_OVERLAPPED Or WS_CAPTION Or
WS_SYSMENU Or WS_THICKFRAME Or WS_MINIMIZEBOX Or WS_MAXIMIZEBOX)
Public Const WS_TILEDWINDOW = WS_OVERLAPPEDWINDOW
Public Const WS_POPUPWINDOW = (WS_POPUP Or WS_BORDER Or WS_SYSMENU)
Public Const WS_CHILDWINDOW = (WS_CHILD)

' ExWindowStyles
Public Const WS_EX_DLGMODALFRAME = &H1&
Public Const WS_EX_NOPARENTNOTIFY = &H4&
Public Const WS_EX_TOPMOST = &H8&
Public Const WS_EX_ACCEPTFILES = &H10&
Public Const WS_EX_TRANSPARENT = &H20&

' Color constants
Public Const COLOR_SCROLLBAR = 0
Public Const COLOR_BACKGROUND = 1
Public Const COLOR_ACTIVECAPTION = 2
Public Const COLOR_INACTIVECAPTION = 3
Public Const COLOR_MENU = 4
Public Const COLOR_WINDOW = 5
```

```
Public Const COLOR_WINDOWFRAME = 6
Public Const COLOR_MENUTEXT = 7
Public Const COLOR_WINDOWTEXT = 8
Public Const COLOR_CAPTIONTEXT = 9
Public Const COLOR_ACTIVEBORDER = 10
Public Const COLOR_INACTIVEBORDER = 11
Public Const COLOR_APPWORKSPACE = 12
Public Const COLOR_HIGHLIGHT = 13
Public Const COLOR_HIGHLIGHTTEXT = 14
Public Const COLOR_BTNFACE = 15
Public Const COLOR_BTNSHADOW = 16
Public Const COLOR_GRAYTEXT = 17
Public Const COLOR_BTNTEXT = 18
Public Const COLOR_INACTIVECAPTIONTEXT = 19
Public Const COLOR_BTNHIGHLIGHT = 20
```

' Window messages

```
Public Const WM_NULL = &H0
Public Const WM_CREATE = &H1
Public Const WM_DESTROY = &H2
Public Const WM_MOVE = &H3
Public Const WM_SIZE = &H5
```

' ShowWindow commands

```
Public Const SW_HIDE = 0
Public Const SW_SHOWNORMAL = 1
Public Const SW_NORMAL = 1
Public Const SW_SHOWMINIMIZED = 2
Public Const SW_SHOWMAXIMIZED = 3
Public Const SW_MAXIMIZE = 3
Public Const SW_SHOWNOACTIVATE = 4
Public Const SW_SHOW = 5
Public Const SW_MINIMIZE = 6
Public Const SW_SHOWMINNOACTIVE = 7
Public Const SW_SHOWNA = 8
Public Const SW_RESTORE = 9
Public Const SW_SHOWDEFAULT = 10
Public Const SW_MAX = 10
```

' Standard ID's of cursors

```
Public Const IDC_ARROW = 32512&
Public Const IDC_IBEAM = 32513&
Public Const IDC_WAIT = 32514&
Public Const IDC_CROSS = 32515&
Public Const IDC_UPARROW = 32516&
Public Const IDC_SIZE = 32640&
Public Const IDC_ICON = 32641&
Public Const IDC_SIZENWSE = 32642&
Public Const IDC_SIZENESW = 32643&
Public Const IDC_SIZEWE = 32644&
Public Const IDC_SIZENS = 32645&
Public Const IDC_SIZEALL = 32646&
Public Const IDC_NO = 32648&
Public Const IDC_APPSTARTING = 32650&
Public Const GWL_WNDPROC = -4
```



```
Dim hwnd2 As Long, hwnd3 As Long, old_proc As Long, new_proc As Long

Public Sub Main()
    'KPD-Team 1999
    'URL: http://www.allapi.net/
    'E-Mail: KPDTeam@Allapi.net
    Dim lngTemp As Long
    ' Register class
    If MyRegisterClass Then
        ' Window created?
        If MyCreateWindow Then
            ' Change the button's procedures
            ' Point to new address
            new_proc = GetMyWndProc(AddressOf ButtonProc)
            old_proc = SetWindowLong(hwnd2, GWL_WNDPROC, new_proc)
            ' Message loop
            MyMessageLoop
        End If
        ' Unregister Class
        MyUnregisterClass
    End If
End Sub

Private Function MyRegisterClass() As Boolean
    ' WNDCLASS-structure
    Dim wndcls As WNDCLASS
    wndcls.style = CS_HREDRAW + CS_VREDRAW
    wndcls.lpfwndproc = GetMyWndProc(AddressOf MyWndProc)
    wndcls.cbClsExtra = 0
    wndcls.cbWndExtra2 = 0
    wndcls.hInstance = App.hInstance
    wndcls.hIcon = 0
    wndcls.hCursor = LoadCursor(0, IDC_ARROW)
    wndcls.hbrBackground = COLOR_WINDOW
    wndcls.lpszMenuName = 0
    wndcls.lpszClassName = "myWindowClass"
    ' Register class
    MyRegisterClass = (RegisterClass(wndcls) <> 0)
End Function

Private Sub MyUnregisterClass()
    UnregisterClass "myWindowClass", App.hInstance
End Sub

Private Function MyCreateWindow() As Boolean
    Dim hWnd As Long
    ' Create the window
    hWnd = CreateWindowEx(0, "myWindowClass", "My Window", WS_OVERLAPPEDWINDOW, 0,
    0, 400, 300, 0, 0, App.hInstance, ByVal 0&)
    ' The Button and Textbox are child windows
    hwnd2 = CreateWindowEx(0, "Button", "My button", WS_CHILD, 50, 55, 100, 25, hWnd, 0,
    App.hInstance, ByVal 0&)
    hwnd3 = CreateWindowEx(0, "edit", "My textbox", WS_CHILD, 50, 25, 100, 25, hWnd, 0,
    App.hInstance, ByVal 0&)
    If hWnd <> 0 Then ShowWindow hWnd, SW_SHOWNORMAL
    ' Show them
```

```
ShowWindow hwnd2, SW_SHOWNORMAL
ShowWindow hwnd3, SW_SHOWNORMAL
' Go back
MyCreateWindow = (hwnd <> 0)
End Function

Private Function MyWndProc(ByVal hwnd As Long, ByVal message As Long, ByVal wParam As Long, ByVal lParam As Long) As Long
    Select Case message
        Case WM_DESTROY
            ' Destroy window
            PostQuitMessage (0)
    End Select
    ' calls the default window procedure
    MyWndProc = DefWindowProc(hwnd, message, wParam, lParam)
End Function

Function GetMyWndProc(ByVal lWndProc As Long) As Long
    GetMyWndProc = lWndProc
End Function

Private Sub MyMessageLoop()
    Dim aMsg As Msg
    Do While GetMessage(aMsg, 0, 0, 0)
        DispatchMessage aMsg
    Loop
End Sub

Private Function ButtonProc(ByVal hwnd As Long, ByVal message As Long, ByVal wParam As Long, ByVal lParam As Long) As Long
    Dim x As Integer
    If (message = 533) Then
        x = MsgBox("You clicked on the button", vbOKOnly)
    End If
    ' calls the window procedure
    ButtonProc = CallWindowProc(old_proc, hwnd, message, wParam, lParam)
End Function
```

+ Ví dụ 5 : Register Server (2)

```
' Add 2 Commandbuttons and a textbox to the form, and paste this code into the form
Option Explicit

Private Declare Function FreeLibrary Lib "kernel32" (ByVal hLibModule As Long) As Long
Private Declare Function LoadLibrary Lib "kernel32" Alias "LoadLibraryA" (ByVal lpLibFileName As String) As Long
Private Declare Function GetProcAddress Lib "kernel32" (ByVal hModule As Long, ByVal lpProcName As String) As Long
Private Declare Function CallWindowProc Lib "user32" Alias "CallWindowProcA" (ByVal lpPrevWndFunc As Long, ByVal hwnd As Long, ByVal Msg As Any, ByVal wParam As Any, ByVal lParam As Any) As Long
Private Const ERROR_SUCCESS = &H0

Private Sub Form_Load()
    Text1.Text = "C:\WINDOWS\SYSTEM\COMCTL32.OCX"
```

```
Command1.Caption = "Register server"
Command2.Caption = "Unregister server"
End Sub

Private Sub Command1_Click()
    Call RegisterServer(Me.hWnd, Text1.Text, True)
End Sub

Private Sub Command2_Click()
    Call RegisterServer(Me.hWnd, Text1.Text, False)
End Sub

Public Function RegisterServer(hWnd As Long, DllServerPath As String, bRegister As Boolean)
    On Error Resume Next

    'KPD-Team 2000
    'URL: http://www.allapi.net/
    'E-Mail: KPDTeam@Allapi.net
    'We're going to call an API-function, without declaring it!

    ' Modified by G. Kleijer
    ' gkleijer@casema.net
    ' going to call the DllRegisterServer/DllUnRegisterServer API of the specified library.
    ' there's no need to use the Regsvr32.exe anymore.

    ' Make sure the path is correct and that the file exists, otherwise VB will crash.

    Dim lb As Long, pa As Long
    lb = LoadLibrary(DllServerPath)

    If bRegister Then
        pa = GetProcAddress(lb, "DllRegisterServer")
    Else
        pa = GetProcAddress(lb, "DllUnregisterServer")
    End If

    If CallWindowProc(pa, hWnd, ByVal 0&, ByVal 0&, ByVal 0&) = ERROR_SUCCESS Then
        MsgBox If(bRegister = True, "Registration", "Unregistration") + " Successful"
    Else
        MsgBox If(bRegister = True, "Registration", "Unregistration") + " Unsuccessful"
    End If
    'unmap the library's address
    FreeLibrary lb
End Function
```

+ Ví dụ 6: Download File

```
'This project needs a TextBox
'-> (Name)=Text1
'-> MultiLine=True
'in a form
Private Sub Form_Load()
    'KPD-Team 2000
    'URL: http://www.allapi.net/
    'E-Mail: KPDTeam@Allapi.net
```

```
Dim sSave As String
Me.AutoRedraw = True
Set Obj = Me.Text1
'Start subclassing
HookForm Me
'create a new winsock session
StartWinsock sSave
'show the winsock version on this form
If InStr(1, sSave, Chr$(0)) > 0 Then sSave = Left$(sSave, InStr(1, sSave, Chr$(0)) - 1)
Me.Print sSave
'connect to Microsoft.com
ISocket = ConnectSock("www.microsoft.com", 80, 0, Me.hwnd, False)
End Sub

Private Sub Form_Unload(Cancel As Integer)
'close our connection to microsoft.com
closesocket ISocket
'end winsock session
EndWinsock
'stop subclassing
UnHookForm Me
End Sub

'in a module
Public Const AF_INET = 2
Public Const INVALID_SOCKET = -1
Public Const SOCKET_ERROR = -1
Public Const FD_READ = &H1&
Public Const FD_WRITE = &H2&
Public Const FD_CONNECT = &H10&
Public Const FD_CLOSE = &H20&
Public Const PF_INET = 2
Public Const SOCK_STREAM = 1
Public Const IPPROTO_TCP = 6
Public Const GWL_WNDPROC = (-4)
Public Const WINSOCKMSG = 1025
Public Const WSA_DESCRIPTIONLEN = 256
Public Const WSA_DescriptionSize = WSA_DESCRIPTIONLEN + 1
Public Const WSA_SYS_STATUS_LEN = 128
Public Const WSA_SysStatusSize = WSA_SYS_STATUS_LEN + 1
Public Const INADDR_NONE = &HFFFF
Public Const SOL_SOCKET = &HFFFF&
Public Const SO_LINGER = &H80&
Public Const hostent_size = 16
Public Const sockaddr_size = 16

Type WSADatatype
wVersion As Integer
wHighVersion As Integer
szDescription As String * WSA_DescriptionSize
szSystemStatus As String * WSA_SysStatusSize
iMaxSockets As Integer
iMaxUdpDg As Integer
lpVendorInfo As Long
End Type
```

```
Type HostEnt
    h_name As Long
    h_aliases As Long
    h_addrtype As Integer
    h_length As Integer
    h_addr_list As Long
End Type

Type sockaddr
    sin_family As Integer
    sin_port As Integer
    sin_addr As Long
    sin_zero As String * 8
End Type

Type LingerType
    l_onoff As Integer
    l_linger As Integer
End Type

Public Declare Function setsockopt Lib "wsck32.dll" (ByVal s As Long, ByVal Level As Long,
ByVal optname As Long, optval As Any, ByVal optlen As Long) As Long
Public Declare Function getsockopt Lib "wsck32.dll" (ByVal s As Long, ByVal Level As Long,
ByVal optname As Long, optval As Any, optlen As Long) As Long
Public Declare Function WSAGetLastError Lib "wsck32.dll" () As Long
Declare Function SetWindowLong Lib "user32" Alias "SetWindowLongA" (ByVal hwnd As Long,
ByVal nIndex As Long, ByVal dwNewLong As Long) As Long
Declare Function CallWindowProc Lib "user32" Alias "CallWindowProcA" (ByVal lpPrevWndFunc
As Long, ByVal hwnd As Long, ByVal Msg As Long, ByVal wParam As Long, ByVal lParam As
Long) As Long
Public Declare Function WSAsIsBlocking Lib "wsck32.dll" () As Long
Public Declare Function WSACleanup Lib "wsck32.dll" () As Long
Public Declare Function Send Lib "wsck32.dll" Alias "send" (ByVal s As Long, buf As Any, ByVal
buflen As Long, ByVal flags As Long) As Long
Public Declare Function recv Lib "wsck32.dll" (ByVal s As Long, buf As Any, ByVal buflen As Long,
ByVal flags As Long) As Long
Public Declare Function WSASStartup Lib "wsck32.dll" (ByVal wVR As Long, lpWSAD As
WSADATAType) As Long
Public Declare Function htons Lib "wsck32.dll" (ByVal hostshort As Long) As Integer
Public Declare Function ntohs Lib "wsck32.dll" (ByVal netshort As Long) As Integer
Public Declare Function socket Lib "wsck32.dll" (ByVal af As Long, ByVal s_type As Long, ByVal
protocol As Long) As Long
Public Declare Function closesocket Lib "wsck32.dll" (ByVal s As Long) As Long
Public Declare Function Connect Lib "wsck32.dll" Alias "connect" (ByVal s As Long, addr As
sockaddr, ByVal namelen As Long) As Long
Public Declare Function WSAAsyncSelect Lib "wsck32.dll" (ByVal s As Long, ByVal hwnd As
Long, ByVal wParam As Long, ByVal lEvent As Long) As Long
Public Declare Function inet_addr Lib "wsck32.dll" (ByVal cp As String) As Long
Public Declare Function gethostbyname Lib "wsck32.dll" (ByVal host_name As String) As Long
Public Declare Sub MemCopy Lib "kernel32" Alias "RtlMoveMemory" (Dest As Any, Src As Any,
ByVal cb&)
Public Declare Function inet_ntoa Lib "wsck32.dll" (ByVal inn As Long) As Long
Public Declare Function strlen Lib "kernel32" Alias "strlenA" (ByVal lpString As Any) As Long
Public Declare Function WSACancelBlockingCall Lib "wsck32.dll" () As Long
Public saZero As sockaddr
Public WSASetup As Boolean, Obj As TextBox
```

```
Public PrevProc As Long, ISocket As Long

'subclassing functions
'for more information about subclassing,
'check out the subclassing tutorial at http://www.allapi.net/
Public Sub HookForm(F As Form)
    PrevProc = SetWindowLong(F.hwnd, GWL_WNDPROC, AddressOf WindowProc)
End Sub

Public Sub UnHookForm(F As Form)
    If PrevProc <> 0 Then
        SetWindowLong F.hwnd, GWL_WNDPROC, PrevProc
        PrevProc = 0
    End If
End Sub

Public Function WindowProc(ByVal hwnd As Long, ByVal uMsg As Long, ByVal wParam As Long,
ByVal lParam As Long) As Long
    If uMsg = WINSOCKMSG Then
        ProcessMessage wParam, lParam
    Else
        WindowProc = CallWindowProc(PrevProc, hwnd, uMsg, wParam, lParam)
    End If
End Function

'our Winsock-message handler
Public Sub ProcessMessage(ByVal IFromSocket As Long, ByVal lParam As Long)
    Dim X As Long, ReadBuffer(1 To 1024) As Byte, strCommand As String
    Select Case lParam
        Case FD_CONNECT 'we are connected to microsoft.com
        Case FD_WRITE 'we can write to our connection
            'this is a part of the HTTP protocol
            'for more information about this protocol, visit http://www.w3c.org/
            strCommand = "GET http://www.microsoft.com/ HTTP/1.0" + vbCrLf
            strCommand = strCommand + "Pragma: no-cache" + vbCrLf
            strCommand = strCommand + "Accept: */*" + vbCrLf
            strCommand = strCommand + "Accept: text/html" + vbCrLf + vbCrLf
            'send the data to our microsoft.com-connection
            SendData IFromSocket, strCommand
        Case FD_READ 'we have data waiting to be processed
            'start reading the data
            Do
                X = recv(IFromSocket, ReadBuffer(1), 1024, 0)
                If X > 0 Then
                    Obj.Text = Obj.Text + Left$(StrConv(ReadBuffer, vbUnicode), X)
                End If
                If X <> 1024 Then Exit Do
            Loop
            Case FD_CLOSE 'the connection with microsoft.com is closed
        End Select
End Sub

'the following functions are standard WinSock functions
'from the wsksock.bas-file
Public Function StartWinsock(sDescription As String) As Boolean
    Dim StartupData As WSADATAType
```

```
If Not WSAShutdown Then
    If Not WSAStartup(&H101, StartupData) Then
        WSAShutdown = True
        sDescription = StartupData.szDescription
    Else
        WSAShutdown = False
    End If
End If
StartWinsock = WSAShutdown
End Function

Sub EndWinsock()
    Dim Ret&
    If WSALoadLibrary() Then
        Ret = WSACancelBlockingCall()
    End If
    Ret = WSACleanup()
    WSAShutdown = False
End Sub

Public Function SendData(ByVal s&, vMessage As Variant) As Long
    Dim TheMsg() As Byte, sTemp$
    TheMsg = ""
    Select Case VarType(vMessage)
        Case 8209 'byte array
            sTemp = vMessage
            TheMsg = sTemp
        Case 8 'string, if we receive a string, its assumed we are linemode
            sTemp = StrConv(vMessage, vbFromUnicode)
        Case Else
            sTemp = CStr(vMessage)
            sTemp = StrConv(vMessage, vbFromUnicode)
    End Select
    TheMsg = sTemp
    If UBound(TheMsg) > -1 Then
        SendData = Send(s, TheMsg(0), (UBound(TheMsg) - LBound(TheMsg) + 1), 0)
    End If
End Function

Function ConnectSock(ByVal Host$, ByVal Port&, retIpPort$, ByVal HWndToMsg&, ByVal Async%)
As Long
    Dim s&, SelectOps&, Dummy&
    Dim sockin As sockaddr
    SockReadBuffer$ = ""
    sockin = saZero
    sockin.sin_family = AF_INET
    sockin.sin_port = htons(Port)
    If sockin.sin_port = INVALID_SOCKET Then
        ConnectSock = INVALID_SOCKET
        Exit Function
    End If

    sockin.sin_addr = GetHostByNameAlias(Host$)

    If sockin.sin_addr = INADDR_NONE Then
        ConnectSock = INVALID_SOCKET
    End If
End Function
```

```
Exit Function
End If
retIpPort$ = getascip$(sockin.sin_addr) & ":" & ntohs(sockin.sin_port)

s = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP)
If s < 0 Then
    ConnectSock = INVALID_SOCKET
    Exit Function
End If
If SetSockLinger(s, 1, 0) = SOCKET_ERROR Then
    If s > 0 Then
        Dummy = closesocket(s)
    End If
    ConnectSock = INVALID_SOCKET
    Exit Function
End If
If Not Async Then
    If Connect(s, sockin, sockaddr_size) <> 0 Then
        If s > 0 Then
            Dummy = closesocket(s)
        End If
        ConnectSock = INVALID_SOCKET
        Exit Function
    End If
    SelectOps = FD_READ Or FD_WRITE Or FD_CONNECT Or FD_CLOSE
    If WSAAsyncSelect(s, HWndToMsg, ByVal 1025, ByVal SelectOps) Then
        If s > 0 Then
            Dummy = closesocket(s)
        End If
        ConnectSock = INVALID_SOCKET
        Exit Function
    End If
Else
    SelectOps = FD_READ Or FD_WRITE Or FD_CONNECT Or FD_CLOSE
    If WSAAsyncSelect(s, HWndToMsg, ByVal 1025, ByVal SelectOps) Then
        If s > 0 Then
            Dummy = closesocket(s)
        End If
        ConnectSock = INVALID_SOCKET
        Exit Function
    End If
    If Connect(s, sockin, sockaddr_size) <> -1 Then
        If s > 0 Then
            Dummy = closesocket(s)
        End If
        ConnectSock = INVALID_SOCKET
        Exit Function
    End If
End If
ConnectSock = s
End Function

Function GetHostByNameAlias(ByVal hostname$) As Long
    On Error Resume Next
    Dim phe&
    Dim heDestHost As HostEnt
```



```
Dim addrList&
Dim retIP&
retIP = inet_addr(hostname)
If retIP = INADDR_NONE Then
    phe = gethostbyname(hostname)
    If phe <> 0 Then
        MemCopy heDestHost, ByVal phe, hostent_size
        MemCopy addrList, ByVal heDestHost.h_addr_list, 4
        MemCopy retIP, ByVal addrList, heDestHost.h_length
    Else
        retIP = INADDR_NONE
    End If
End If
GetHostByNameAlias = retIP
If Err Then GetHostByNameAlias = INADDR_NONE
End Function

Function getascip(ByVal inn As Long) As String
    On Error Resume Next
    Dim lpStr&
    Dim nStr&
    Dim retString$
    retString = String(32, 0)
    lpStr = inet_ntoa(inn)
    If lpStr = 0 Then
        getascip = "255.255.255.255"
        Exit Function
    End If
    nStr = strlen(lpStr)
    If nStr > 32 Then nStr = 32
    MemCopy ByVal retString, ByVal lpStr, nStr
    retString = Left(retString, nStr)
    getascip = retString
    If Err Then getascip = "255.255.255.255"
End Function

Public Function SetSockLinger(ByVal SockNum&, ByVal OnOff%, ByVal LingerTime%) As Long
    Dim Linger As LingerType
    Linger.l_onoff = OnOff
    Linger.l_linger = LingerTime
    If setsockopt(SockNum, SOL_SOCKET, SO_LINGER, Linger, 4) Then
        Debug.Print "Error setting linger info: " & WSAGetLastError()
        SetSockLinger = SOCKET_ERROR
    Else
        If getsockopt(SockNum, SOL_SOCKET, SO_LINGER, Linger, 4) Then
            Debug.Print "Error getting linger info: " & WSAGetLastError()
            SetSockLinger = SOCKET_ERROR
        End If
    End If
End Function
```

9. [ChildWindowFromPoint](#)

- *Thư viện* : user32.dll
- *Hệ điều hành* : Windows NT 3.1 or later; Windows 95 or later

- Khai báo :

```
Declare Function ChildWindowFromPoint Lib "user32" Alias "ChildWindowFromPoint" (ByVal hWndParent As Long, ByVal pt As POINTAPI) As Long
```

- Các tham số
 - **hWndParent** : Cán (handles) của cửa sổ chứa (mức Parent).
 - **pt** : Trị của điểm
- Mô tả : Lấy cán của cửa sổ con (mức Child) đầu tiên thoả mãn.
- Trị trả về : Nếu không thấy cửa sổ con (mức Child) nào trả về cán của cửa sổ chứa (mức Parent). Trả về 0 nếu điểm nằm ngoài cửa sổ chứa (mức Parent).
- Các hàm liên quan : ChildWindowFromPointEx

10. ChildWindowFromPointEx

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

```
Declare Function ChildWindowFromPointEx Lib "user32" Alias "ChildWindowFromPointEx" (ByVal hWnd As Long, ByVal pt As POINTAPI, ByVal un As Long) As Long
```

- Các tham số
 - **hWndParent** : Cán (handles) của cửa sổ chứa (mức Parent).
 - **pt** : Trị của điểm
- Mô tả : Lấy cán của cửa sổ con (mức Child) đầu tiên thoả mãn.
- Trị trả về : Nếu không thấy cửa sổ con (mức Child) nào trả về cán của cửa sổ chứa (mức Parent). Trả về 0 nếu điểm nằm ngoài cửa sổ chứa (mức Parent).
- Các hàm liên quan : ChildWindowFromPoint:

11. ClientToScreen

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

```
Declare Function ClientToScreen Lib "user32" Alias "ClientToScreen" (ByVal hwnd As Long, lpPoint As POINTAPI) As Long
```

- Các tham số
 - **hWnd** : Cán (handles) của cửa sổ làm căn cứ xác định toạ độ.
 - **lpPoint** : Là biến cấu trúc kiểu POINTAPI chứa toạ độ cửa sổ chuyển đổi. Nếu hàm thực hiện thành công thì nó sẽ copy toạ độ của màn hình mới vào trong cấu trúc này.
- Mô tả : Chuyển toạ độ theo cửa sổ sang toạ độ theo màn hình.
- Các hàm liên quan : ScreenToClient
- Các ví dụ minh hoạ :
 - + Ví dụ 1 : Move Cursor

```
'This project needs 2 Buttons
Private Type POINTAPI
    x As Long
    y As Long
End Type
Private Declare Function ClientToScreen Lib "user32" (ByVal hwnd As Long, lpPoint As
```

```
POINTAPI) As Long
Private Declare Function SetCursorPos Lib "user32" (ByVal x As Long, ByVal y As Long) As Long
Private Declare Function GetDeviceCaps Lib "gdi32" (ByVal hdc As Long, ByVal nIndex As Long)
As Long

Dim P As POINTAPI
Private Sub Form_Load()
    'KPD-Team 1998
    'URL: http://www.allapi.net/
    'E-Mail: KPDTTeam@Allapi.net

    Command1.Caption = "Screen Middle"
    Command2.Caption = "Form Middle"
    'API uses pixels
    Me.ScaleMode = vbPixels
End Sub

Private Sub Command1_Click()
    'Get information about the screen's width
    P.x = GetDeviceCaps(Form1.hdc, 8) / 2
    'Get information about the screen's height
    P.y = GetDeviceCaps(Form1.hdc, 10) / 2
    'Set the mouse cursor to the middle of the screen
    ret& = SetCursorPos(P.x, P.y)
End Sub

Private Sub Command2_Click()
    P.x = 0
    P.y = 0
    'Get information about the form's left and top
    ret& = ClientToScreen&(Form1.hwnd, P)
    P.x = P.x + Me.ScaleWidth / 2
    P.y = P.y + Me.ScaleHeight / 2
    'Set the cursor to the middle of the form
    ret& = SetCursorPos&(P.x, P.y)
End Sub
```

+ Vi dụ 2: ClipCursor

```
Private Type RECT
    left As Long
    top As Long
    right As Long
    bottom As Long
End Type

Private Type POINT
    x As Long
    y As Long
End Type

Private Declare Sub ClipCursor Lib "user32" (lpRect As Any)
Private Declare Sub GetClientRect Lib "user32" (ByVal hWnd As Long, lpRect As RECT)
Private Declare Sub ClientToScreen Lib "user32" (ByVal hWnd As Long, lpPoint As POINT)
```

```
Private Declare Sub OffsetRect Lib "user32" (lpRect As RECT, ByVal x As Long, ByVal y As Long)

Private Sub Form_Load()
    'KPD-Team 1999
    'URL: http://www.allapi.net/
    'E-Mail: KPDTTeam@Allapi.net
    Command1.Caption = "Limit Cursor Movement"
    Command2.Caption = "Release Limit"
End Sub

Private Sub Command1_Click()
    'Limits the Cursor movement to within the form.
    Dim client As RECT
    Dim upperleft As POINT
    'Get information about our window
    GetClientRect Me.hWnd, client
    upperleft.x = client.left
    upperleft.y = client.top
    'Convert window coördinates to screen coördinates
    ClientToScreen Me.hWnd, upperleft
    'move our rectangle
    OffsetRect client, upperleft.x, upperleft.y
    'limit the cursor movement
    ClipCursor client
End Sub

Private Sub Command2_Click()
    'Releases the cursor limits
    ClipCursor ByVal 0&
End Sub

Private Sub Form_Unload(Cancel As Integer)
    'Releases the cursor limits
    ClipCursor ByVal 0&
End Sub
```

+ Ví dụ 3: Window Placement

```
Private Const SW_MINIMIZE = 6

Private Type POINTAPI
    x As Long
    y As Long
End Type

Private Type RECT
    Left As Long
    Top As Long
    Right As Long
    Bottom As Long
End Type

Private Type WINDOWPLACEMENT
    Length As Long
```

```
    flags As Long
    showCmd As Long
    ptMinPosition As POINTAPI
    ptMaxPosition As POINTAPI
    rcNormalPosition As RECT
End Type

Private Declare Function ClientToScreen Lib "user32" (ByVal hwnd As Long, lpPoint As POINTAPI) As Long
Private Declare Function GetWindowPlacement Lib "user32" (ByVal hwnd As Long, lpwndpl As WINDOWPLACEMENT) As Long
Private Declare Function SetWindowPlacement Lib "user32" (ByVal hwnd As Long, lpwndpl As WINDOWPLACEMENT) As Long

Dim Rectan As RECT

Private Sub Form_Load()
    'Tip submitted by pyp99 (pyp99@hotmail.com)
    Dim WinEst As WINDOWPLACEMENT
    Dim rtn As Long
    WinEst.Length = Len(WinEst)
    'get the current window placement
    rtn = GetWindowPlacement(Me.hwnd, WinEst)
    Rectan = WinEst.rcNormalPosition
End Sub

Private Sub Command1_Click()
    Dim WinEst As WINDOWPLACEMENT
    Dim Punto As POINTAPI
    Dim rtn As Long
    'set the new min/max positions
    Punto.x = 100
    Punto.y = 100
    'initialize the structure
    WinEst.Length = Len(WinEst)
    WinEst.showCmd = SW_MINIMIZE
    WinEst.ptMinPosition = Punto
    WinEst.ptMaxPosition = Punto
    WinEst.rcNormalPosition = Rectan
    'set the new window placement (minimized)
    rtn = SetWindowPlacement(Me.hwnd, WinEst)
End Sub
```

12. CloseWindow

- *Thư viện* : user32.dll
- *Hệ điều hành* : Windows NT 3.1 or later; Windows 95 or later
- *Khai báo* :

```
Declare Function CloseWindow Lib "user32" Alias "CloseWindow" (ByVal hwnd As Long) As Long
```

- *Các tham số*
 - **hWnd** : Cán (handles) của cửa sổ cần thu nhỏ.
- *Mô tả* : Thu nhỏ cửa sổ .

- Các hàm liên quan : ShowWindow
- Các ví dụ minh họa : CloseWindow

```
Private Declare Function CloseWindow Lib "user32" (ByVal hwnd As Long) As Long

Private Sub Form_Load()
    'KPD-Team 2000
    'URL: http://www.allapi.net/
    'E-Mail: KPDTTeam@Allapi.net
    'Minimize this Window
    CloseWindow Me.hwnd
End Sub
```

13. [CommDlgExtendedError](#)

- Thư viện : comdlg32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

```
Public Declare Function CommDlgExtendedError Lib "comdlg32.dll" Alias  
"CommDlgExtendedError" () As Long
```

- Mô tả : Hàm CommDlgExtendedError trả về mã lỗi từ chức năng cuối cùng của một hộp thoại common dialog nào đó. Hàm không trả về mã lỗi cho bất kỳ hàm API nào khác (trong trường hợp này, dùng GetLastError để thay thế). Giá trị trả về của hàm không được xác định nếu chức năng được gọi sau cùng của hộp thoại common dialog thành công. Nếu có một lỗi xảy ra với chức năng này, giá trị trả về chính xác là một trong những cờ lỗi của hộp thoại common dialog sau đây :

CDERR_DIALOGFAILURE = &HFFFF

Không thể mở hộp thoại.

CDERR_FINDRESFAILURE = &H6

Thất bại khi muốn tìm tài nguyên cần thiết.

CDERR_GENERALCODES = &H0

Lỗi liên quan đến một thuộc tính tổng quát của hộp thoại common.

CDERR_INITIALIZATION = &H2

Thất bại trong suốt quá trình khởi tạo (thường là bộ nhớ không đủ).

CDERR_LOADRESFAILURE = &H7

Thất bại khi nạp tài nguyên yêu cầu.

CDERR_LOADSTRFAILURE = &H5

Thất bại khi nạp chuỗi yêu cầu.

CDERR_LOCKRESFAILURE = &H8

Thất bại khi khoá tài nguyên yêu cầu.

CDERR_MEMALLOCFAILURE = &H9

Thất bại khi xác định khối bộ nhớ.

CDERR_MEMLOCKFAILURE = &HA

Thất bại khi khoá bộ nhớ yêu cầu.

CDERR_NOINSTANCE = &H4

Không được cung cấp một handles hợp lệ (nếu handles được yêu cầu).

CDERR_NOHOOK = &HB

Không được cung cấp một handles tới hàm hook hợp lệ (nếu handles được yêu cầu).

CDERR_NOTEMPLATE = &H3

Không được cung cấp màu ban đầu hợp lệ (nếu màu được yêu cầu).

CDERR_REGISTERMSGFAIL = &HC
Không thể đăng ký một thông điệp cửa sổ thành công.

CDERR_STRUCTSIZE = &H1
Được cung cấp một kích thước cấu trúc không hợp lệ.

CFERR_CHOOSEFONTCODES = &H2000
Lỗi liên quan đến hộp thoại Choose Font.

CFERR_MAXLESSTHANMIN = &H2002
Được cung cấp giá trị kích thước font lớn nhất nhỏ hơn kích thước font nhỏ nhất đã được cung cấp.

CFERR_NOFONTS = &H2001
Không thể tìm thấy các font đang tồn tại.

FNERR_BUFFERTOOSMALL = &H3003
Được cung cấp một bộ đệm tên tập tin quá nhỏ.

FNERR_FILENAMECODES = &H3000
Lỗi liên quan đến hộp thoại Open File hoặc Save File.

FNERR_INVALIDFILENAME = &H3002
Được cung cấp hay nhận một tên tập tin không hợp lệ.

FNERR_SUBCLASSFAILURE = &H3001
Không đủ bộ nhớ để phân lớp hộp danh sách.

FRERR_BUFFERLENGTHZERO = &H4001
Được cung cấp một bộ đệm không hợp lệ.

FRERR_FINDREPLACECODES = &H4000
Lỗi liên quan đến hộp thoại Find hoặc Replace.

PDERR_CREATEICFAILURE = &H100A
Không thể tạo một ngữ cảnh thông tin.

PDERR_DEFAULTDIFFERENT = &H100C
Đang có thông tin được cung cấp của máy in mặc định, nhưng thực sự thì các thiết lập máy in mặc định lại khác.

PDERR_DNDMMISMATCH = &H1009
Dữ liệu trong hai cấu trúc dữ liệu mô tả các máy in khác nhau (tức là chúng chứa các thông tin mâu thuẫn).

PDERR_GETDEVMODEFAIL = &H1005
Driver máy in thất bại khi khởi tạo cấu trúc DEVMODE.

PDERR_INITFAILURE = &H1006
Thất bại trong quá trình khởi tạo.

PDERR_LOADDRVFAILURE = &H1004
Thất bại khi nạp driver thiết bị yêu cầu.

PDERR_NODEFAULTPRN = &H1008
Không thể tìm thấy máy in mặc định.

PDERR_NODEVICES = &H1007
Không thể tìm thấy bất kỳ máy in nào.

PDERR_PARSEFAILURE = &H1002
Thất bại khi phân tích các chuỗi quan hệ với máy in trong WIN.INI

PDERR_PRINTERCODES = &H1000
Lỗi liên quan đến hộp thoại Print.

PDERR_PRINTERNOTFOUND = &H100B
Không thể tìm thấy thông tin trong WIN.INI về máy in được yêu cầu.

PDERR_RETDEFFAILURE = &H1003
Các handles tới cấu trúc dữ liệu được cung cấp là khác không dù hàm được yêu cầu trả thông tin về máy in mặc định.

PDERR_SETUPFAILURE = &H1001

Thất lại khi nạp những tài nguyên yêu cầu.

- *Các hàm liên quan :*
 - + CHOOSECOLOR
 - + GetOpenFileName
 - + CHOOSEFONT
 - + GetSaveFileName
 - + PrintDialog
 - + PAGESETUPDLG
 - + GetLastError
- *Các ví dụ minh họa :*
 - + Ví dụ 1 : CommDlgExtendedError

```
Const CDERR_DIALOGFAILURE = &HFFFF
Const CDERR_FINDRESFAILURE = &H6
Const CDERR_GENERALCODES = &H0
Const CDERR_INITIALIZATION = &H2
Const CDERR_LOADRESFAILURE = &H7
Const CDERR_LOADSTRFAILURE = &H5
Const CDERR_LOCKRESFAILURE = &H8
Const CDERR_MEMALLOCFAILURE = &H9
Const CDERR_MEMLOCKFAILURE = &HA
Const CDERR_NOHINSTANCE = &H4
Const CDERR_NOHOOK = &HB
Const CDERR_REGISTERMSGFAIL = &HC
Const CDERR_NOTEMPLATE = &H3
Const CDERR_STRUCTSIZE = &H1

Private Declare Function GetOpenFileName Lib "comdlg32.dll" Alias "GetOpenFileNameA"
(pOpenfilename As Any) As Long
Private Declare Function CommDlgExtendedError Lib "comdlg32.dll" () As Long

Private Sub Form_Load()
'KPD-Team 1999
'URL: http://www.allapi.net/
'E-Mail: KPDTeam@Allapi.net
'We're letting the GetOpenFileName-function crash
GetOpenFileName ByVal 0&
Select Case CommDlgExtendedError
Case CDERR_DIALOGFAILURE
MsgBox "The dialog box could not be created."
Case CDERR_FINDRESFAILURE
MsgBox "The common dialog box function failed to find a specified resource."
Case CDERR_INITIALIZATION
MsgBox "The common dialog box function failed during initialization."
Case CDERR_LOADRESFAILURE
MsgBox "The common dialog box function failed to load a specified resource."
Case CDERR_LOADSTRFAILURE
MsgBox "The common dialog box function failed to load a specified string."
Case CDERR_LOCKRESFAILURE
MsgBox "The common dialog box function failed to lock a specified resource."
Case CDERR_MEMALLOCFAILURE
MsgBox "The common dialog box function was unable to allocate memory for internal
structures."
Case CDERR_MEMLOCKFAILURE
```



```
        MsgBox "The common dialog box function was unable to lock the memory associated with a
handle."
    Case CDERR_NOHINSTANCE
        MsgBox "The ENABLETEMPLATE flag was set in the Flags member of the initialization
structure for the corresponding common dialog box, but you failed to provide a corresponding
instance handle."
    Case CDERR_NOHOOK
        MsgBox "The ENABLEHOOK flag was set in the Flags member of the initialization structure
for the corresponding common dialog box, but you failed to provide a pointer to a corresponding
hook procedure."
    Case CDERR_REGISTERMSGFAIL
        MsgBox "The RegisterWindowMessage function returned an error code when it was called
by the common dialog box function."
    Case CDERR_NOTEMPLATE
        MsgBox "The ENABLETEMPLATE flag was set in the Flags member of the initialization
structure for the corresponding common dialog box, but you failed to provide a corresponding
template."
    Case CDERR_STRUCTSIZE
        MsgBox "The IStructSize member of the initialization structure for the corresponding
common dialog box is invalid."
    Case Else
        MsgBox "Undefined error ..."
    End Select
End Sub
```

- + Ví dụ 2 : Cho hộp thoại Open File một kích thước bộ đệm không đủ. Sau đó hiển thị mã lỗi đã cung cấp.

```
'Trích từ Cam Nang Lap Trinh Windows API - NXB Giao Thong Van Tai
'Cho hop thoai Open File mot kích thước bộ đệm không đủ. Sau đó
'Hien thi ma loi đã cung cấp cho hop thoai OpenFile
```

```
Private Type OPENFILENAME
    IStructSize As Long
    hwndOwner As Long
    hInstance As Long
    lpstrFilter As String
    lpstrCustomFilter As String
    nMaxCustFilter As Long
    nFilterIndex As Long
    lpstrFile As String
    nMaxFile As Long
    lpstrFileName As String
    nMaxFileName As Long
    lpstrInitialDir As String
    lpstrTitle As String
    flags As Long
    nFileOffset As Integer
    nFileExtension As Integer
    lpstrDefExt As String
    lCustData As Long
    lpfnHook As Long
    lpTemplateName As String
End Type
```

```

Private Declare Function GetOpenFileName Lib "comdlg32.dll" Alias "GetOpenFileNameA"
(pOpenfilename As OPENFILENAME) As Long
Private Declare Function CommDlgExtendedError Lib "comdlg32.dll" () As Long

Private Const FNERR_BUFFERTOOSMALL = &H3003
Private Const FNERR_INVALIDFILENAME = &H3002
Private Const OFN_PATHMUSTEXIST = &H800
Private Const OFN_FILEMUSTEXIST = &H1000
Private Const OFN_HIDEREADONLY = &H4

Private Sub Form_Load()
    Dim filebox As OPENFILENAME 'Cau truc thiet lap hop thoai
    Dim fname As String 'se nhan vao ten tap tin duoc chon
    Dim retval As Long 'gia tri tra ve
    Dim errcode As Long 'nhan ma loi

    'Cau hinh hinh dang hop thoai
    filebox.lStructSize = Len(filebox) 'kich thuc cua cau truc
    filebox.hwndOwner = Me.hWnd

    'Van ban hien thi trong thanh tieu de cua hop
    filebox.lpstrFile = "Open File"

    'Dong ke tiep thiet lap tap tin kieu drop-box
    filebox.lpstrFilter = "Text Files" & vbNullChar & "*.txt" & vbNullChar & "All Files" & vbNullChar &
    "*.*" & vbNullChar & vbNullChar
    filebox.lpstrFile = "" 'Loi : bo dem rong!
    filebox.nMaxFile = 0 'Chieu dai cua tap tin la duong dan bo dem

    'Khoi tao bo dem nhan ten tap tin
    filebox.lpstrFileTitle = Space(255)
    'Chieu dai cua bo dem ten tap tin chi cho phep cac tap tin ton tai
    'va che giao hop check chi doc
    filebox.nMaxFileTitle = 255
    filebox.flags = OFN_PATHMUSTEXIST Or OFN_FILEMUSTEXIST Or OFN_HIDEREADONLY

    'Thuc thi hop thoai
    retval = GetOpenFileName(filebox)
    If retval = 0 Then 'Vai loi da xay ra hoac nut Cancel bi nhan
        errcode = CommDlgExtendedError()
        'Lay ma loi cua ham GetOpenFileName
        If errcode = FNERR_BUFFERTOOSMALL Then
            MsgBox "The buffer provider was too small to " + "hold the file name"
        ElseIf errcode = FNERR_INVALIDFILENAME Then
            MsgBox "An invalid filename was provider"
        Else
            MsgBox "The common dialog box function was unable to allocate memory for internal
structures."
        End If
    End If
End Sub

```

14. CopyRect

- Thư viện : user32.dll

- *Hệ điều hành* : Windows NT 3.1 or later; Windows 95 or later
- *Khai báo* :

```
Public Declare Function CopyRect Lib "user32" Alias "CopyRect" (lpDestRect As RECT, lpSourceRect As RECT) As Long
```

- *Các tham số*
 - **lpDestRect** : Hình chữ nhật đích để thiết lập (sẽ nhận kết quả).
 - **lpSourceRect** : Hình chữ nhật nguồn (bị copy).
- *Mô tả* : Hàm CopyRect sao nội dung hình chữ nhật. Hàm này gán một hình chữ nhật bằng với một hình chữ nhật khác. Điều này được thực hiện bằng cách gấp đôi tất cả giá trị thành phần của hình chữ nhật nguồn tới những giá trị tương ứng trong hình chữ nhật đích. Việc này nhanh hơn là phải gán bốn tọa độ chính bằng mã.
- *Trị trả về* : Hàm trả về giá trị 0 nếu có một lỗi xảy ra, hoặc 1 nếu thành công.
- *Các hàm liên quan* :
 - + EqualRect
 - + SetRect
 - + SetRectEmpty
- *Các ví dụ minh họa* : CopyRect

```
Private Type RECT
    Left As Long
    Top As Long
    Right As Long
    Bottom As Long
End Type
Private Declare Function CopyRect Lib "user32" (lpDestRect As RECT, lpSourceRect As RECT) As Long
Private Declare Function GetWindowRect Lib "user32" (ByVal hwnd As Long, lpRect As RECT) As Long
Private Sub Form_Load()
    'KPD-Team 1999
    'URL: http://www.allapi.net/
    'E-Mail: KPDTTeam@Allapi.net
    Dim rectWindow As RECT, rectCopy As RECT
    'Get the bounding rectangle of this window
    GetWindowRect Me.hwnd, rectWindow
    'Copy the rectangle
    CopyRect rectCopy, rectWindow
    MsgBox "This form's width:" + Str$(rectCopy.Right - rectCopy.Left) + " pixels"
End Sub
```

15. DeferWindowPos

- *Thư viện* : user32.dll
- *Hệ điều hành* : Windows NT 3.1 or later; Windows 95 or later
- *Khai báo* :

```
Public Declare Function DeferWindowPos Lib "user32" Alias "DeferWindowPos" (ByVal hWinPosInfo As Long, ByVal hwnd As Long, ByVal hWndInsertAfter As Long, ByVal x As Long, ByVal y As Long, ByVal cx As Long, ByVal cy As Long, ByVal wFlags As Long) As Long
```

- *Các tham số*
 - **hWinPosInfo** : Cán của cấu trúc bản đồ, nó chứa đựng thông tin kích thước và vị trí của một hoặc nhiều cửa sổ. Cấu trúc này được trả về (return) bởi hàm `BeginDeferWindowPos` hoặc bởi lời gọi hàm `DeferWindowPos`.
 - **hWnd** : Cửa sổ cần định vị.
 - **hWndInsertAfter** : Cán của cửa sổ mà cửa sổ hWnd đặt sau nó trong danh sách. Nó có thể là một trong các hằng sau :
 - **HWND_BOTTOM** : Đặt cửa sổ về cuối danh sách. Nếu tham số hWnd xác định một cửa sổ nằm trên cùng và được đặt ở cuối tất cả các cửa sổ khác.
 - **HWND_NOTOPMOST** : Đặt cửa sổ nằm trên tất cả các cửa sổ khác trừ cửa sổ topmost (nghĩa là đặt đằng sau tất cả các cửa sổ topmost). Cờ này sẽ không có hiệu lực nếu cửa sổ này đã là cửa sổ non-topmost.
 - **HWND_TOP** : Đặt cửa sổ ở đầu danh sách.
 - **HWND_TOPMOST** : Đặt cửa sổ ở đầu danh sách lên trên cùng nhìn thấy được. Cửa sổ này sẽ luôn luôn nằm trên tất cả các cửa sổ khác thậm trí khi nó ở trạng thái không hoạt động, tham số này sẽ không có hiệu lực nếu cờ **SWP_NOZORDER** được bật lên trong tham số wFlags.
 - **x** : Hoành độ của cửa sổ hWnd theo toạ độ của cửa sổ chứa (mức Parent) nó.
 - **y** : Tung độ của cửa sổ hWnd theo toạ độ của cửa sổ chứa (mức Parent) nó.
 - **cx** : Chiều rộng của cửa sổ mới.
 - **cy** : Chiều cao của cửa sổ mới.
 - **wFlags** : Cờ xác định kích thước và vị trí của cửa sổ, được kết hợp bởi những hằng số sau :
 - **SWP_DRAWFRAME** : Vẽ khung bao quanh cửa sổ.
 - **SWP_FRAMECHANGED** : Gửi thông điệp `WM_NCCALCSIZE` đến cửa sổ cho dù kích thước của cửa sổ không thay đổi. Nếu cờ này chưa được chỉ rõ (không sử dụng) thì thông điệp `WM_NCCALCSIZE` chỉ được gửi đi khi kích thước của cửa sổ thay đổi.
 - **SWP_HIDEWINDOW** : Ẩn cửa sổ.
 - **SWP_NOACTIVATE** : Không kích hoạt cửa sổ. Nếu không thiết lập cờ này, thì cửa sổ sẽ được kích hoạt và di chuyển lên đầu của cửa sổ topmost hoặc non-topmost (phụ thuộc vào sự thiết lập của tham số `hWndInsertAfter`).
 - **SWP_NOCOPYBITS** : Huỷ bỏ toàn bộ nội dung của vùng Client. Nếu cờ này không được thiết lập thì nội dung của vùng Client sẽ được lưu lại và copied sau vào trong vùng Client sau cửa sổ được xác định.
 - **SWP_NOMOVE** : Giữ nguyên vị trí hiện tại (bỏ qua các tham số x và y).
 - **SWP_NOOWNERZORDER** : Không thay đổi vị trí của cửa sổ cha mẹ trong danh sách.
 - **SWP_NOREDRAW** : Không tự động vẽ lại. Nếu cờ này được thiết lập thì nó sẽ không vẽ lại bất kì cửa sổ nào xuất hiện. Nó được áp dụng trong vùng client và nonclient (bao gồm cả thanh tiêu đề và thanh cuộn), và bất kì phần nào của cửa sổ cha mẹ khi có cửa sổ khác che lấp.
 - **SWP_NOREPOSITION** : Giống như cờ **SWP_NOOWNERZORDER**.
 - **SWP_NOSENDCHANGING** : Ngăn cản cửa sổ nhận thông điệp `WM_WINDOWPOSCHANGING`.
 - **SWP_NOSIZE** : Giữ nguyên kích thước (bỏ qua các tham số cx và cy).
 - **SWP_NOZORDER** : Giữ nguyên vị trí hiện hành trong danh sách (bỏ qua tham số `hWndInsertAfter`).
 - **SWP_SHOWWINDOW** : Hiển thị cửa sổ.

- *Mô tả* : Hàm DeferWindowPos định nghĩa vị trí của cửa sổ mới qua cửa sổ khai báo và đưa vào cấu trúc bản đồ nội bộ chứa vị trí các cửa sổ.
- *Trị trả về* : Long – Cản (handle) mới đối với cấu trúc bản đồ chứa thông tin cập nhật vị trí. Trả về 0 nếu thất bại.
- *Các hàm liên quan* :
 - + BeginDeferWindowPos : tạo ra cấu trúc
 - + EndDeferWindowPos : sử dụng thông tin trong cấu trúc này để thay đổi vị trí và kích thước của một số cửa sổ.
- *Các ví dụ minh họa* : DeferWindowPos

```
Const WS_BORDER = &H800000
Const WS_DLGFRAME = &H400000
Const WS_THICKFRAME = &H40000
Const WS_CAPTION = &HC00000 ' WS_BORDER Or WS_DLGFRAME
Const HWND_BOTTOM = 1
Const HWND_TOP = 0
Const HWND_TOPMOST = -1
Const HWND_NOTOPMOST = -2
Const SWP_SHOWWINDOW = &H40

Private Type RECT
    Left As Long
    Top As Long
    Right As Long
    Bottom As Long
End Type

Private Declare Function AdjustWindowRect Lib "user32" (lpRect As RECT, ByVal dwStyle As Long,
ByVal bMenu As Long) As Long
Private Declare Function BeginDeferWindowPos Lib "user32" (ByVal nNumWindows As Long) As Long
Private Declare Function DeferWindowPos Lib "user32" (ByVal hWinPosInfo As Long, ByVal hwnd
As Long, ByVal hWndInsertAfter As Long, ByVal x As Long, ByVal y As Long, ByVal cx As Long,
ByVal cy As Long, ByVal wFlags As Long) As Long
Private Declare Function EndDeferWindowPos Lib "user32" (ByVal hWinPosInfo As Long) As Long

Private Sub Form_Load()
    'KPD-Team 2000
    'URL: http://www.allapi.net/
    'E-Mail: KPDTTeam@Allapi.net
    Dim R As RECT, hDWP As Long
    R.Left = 30
    R.Top = 30
    R.Bottom = 200
    R.Right = 120
    AdjustWindowRect R, WS_THICKFRAME Or WS_CAPTION, False
    hDWP = BeginDeferWindowPos(1)
    DeferWindowPos hDWP, Me.hwnd, HWND_TOP, R.Left, R.Top, R.Right - R.Left, R.Bottom -
R.Top, SWP_SHOWWINDOW
    EndDeferWindowPos hDWP
End Sub
```

16. DefWindowProc

- *Thư viện* : user32.dll
- *Hệ điều hành* : Windows NT 3.1 or later; Windows 95 or later
- *Khai báo* :

```
Public Declare Function DefWindowProc Lib "user32" Alias "DefWindowProcA" (ByVal hwnd As Long, ByVal wParam As Long, ByVal lParam As Long) As Long
```

- *Các tham số*
 - **hwnd** : Cán (handle) của cửa sổ xử lý thông điệp.
 - **wParam** : Thông điệp cần xử lý.
 - **wParam** : Thông tin mở rộng về thông điệp. Nội dung của tham số này phụ thuộc vào giá trị của tham số wParam.
 - **lParam** : Thông tin mở rộng về thông điệp. Nội dung của tham số này phụ thuộc vào giá trị của tham số wParam.
- *Mô tả* : Hàm DefWindowProc gọi tường minh thủ tục window mặc định của hệ điều hành để xử lý một thông điệp cho một cửa sổ. Thủ tục window mặc định này cung cấp chức năng cần thiết tối thiểu cho một thủ tục window và nên được dùng để cung cấp hiện thực mặc định của thông điệp cửa sổ.
- *Trị trả về* : Giá trị trả về của hàm này là giá trị trả về của thông điệp được xử lý.
- *Các hàm liên quan* : CallWindowProc
- *Các ví dụ minh họa* :
 - + Ví dụ 1 : Hotkey

```
Private Declare Function SendMessage Lib "user32" Alias "SendMessageA" (ByVal hwnd As Long, ByVal wParam As Long, ByVal lParam As Long, ByVal wMsg As Long) As Long
Private Declare Function DefWindowProc Lib "user32" Alias "DefWindowProcA" (ByVal hwnd As Long, ByVal wParam As Long, ByVal lParam As Long, ByVal wMsg As Long) As Long
```

```
Const WM_SETHOTKEY = &H32
Const WM_SHOWWINDOW = &H18
Const HK_SHIFTA = &H141 'Shift + A
Const HK_SHIFTB = &H142 'Shift + B
Const HK_CONTROL = &H241 'Control + A
Const HK_ALTF = &H45A
```

```
'The value of the key-combination has to
'declared in lowbyte/highbyte-format
'That means as a hex-number: the last two
'characters specify the lowbyte (e.g.: 41 = a),
'the first the highbyte (e.g.: 01 = 1 = Shift)
```

```
Private Sub Form_Load()
    'KPD-Team 1999
    'URL: http://www.allapi.net/
    'E-Mail: KPDTTeam@Allapi.net
    Me.WindowState = vbMinimized
    'Let windows know what hotkey you want for
    'your app, setting of lParam has no effect
    erg = SendMessage(Me.hwnd, WM_SETHOTKEY, HK_ALTF, 0)
    'Check if successful
    If erg <> 1 Then
        MsgBox "You need another hotkey", vbOKOnly, "Error"
    End If
```

```
'Tell windows what it should do, when the hotkey  
'is pressed -> show the window!  
'The setting of wParam and lParam has no effect  
    erg& = DefWindowProc(Me.hwnd, WM_SHOWWINDOW, 0, 0)  
End Sub
```

+ Ví dụ 2: Classical

```
'This project needs one form  
' Also set StartupObject to 'Sub Main'  
' (-> Project Properties -> General Tab -> Startup Object)  
'---- Declarations  
Declare Function RegisterClass Lib "user32" Alias "RegisterClassA" (Class As WNDCLASS) As Long  
Declare Function UnregisterClass Lib "user32" Alias "UnregisterClassA" (ByVal lpClassName As String, ByVal hInstance As Long) As Long  
Declare Function CreateWindowEx Lib "user32" Alias "CreateWindowExA" (ByVal dwExStyle As Long, ByVal lpClassName As String, ByVal lpWindowName As String, ByVal dwStyle As Long, ByVal x As Long, ByVal y As Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal hWndParent As Long, ByVal hMenu As Long, ByVal hInstance As Long, lParam As Any) As Long  
Declare Function DefWindowProc Lib "user32" Alias "DefWindowProcA" (ByVal hWnd As Long, ByVal wParam As Long, ByVal lParam As Long) As Long  
Declare Sub PostQuitMessage Lib "user32" (ByVal nExitCode As Long)  
Declare Function GetMessage Lib "user32" Alias "GetMessageA" (lpMsg As Msg, ByVal hWnd As Long, ByVal wParam As Long, ByVal lParam As Long) As Long  
Declare Function TranslateMessage Lib "user32" (lpMsg As Msg) As Long  
Declare Function DispatchMessage Lib "user32" Alias "DispatchMessageA" (lpMsg As Msg) As Long  
Declare Function ShowWindow Lib "user32" (ByVal hWnd As Long, ByVal nCmdShow As Long) As Long  
Declare Function LoadCursor Lib "user32" Alias "LoadCursorA" (ByVal hInstance As Long, ByVal lpCursorName As Any) As Long  
Declare Function DefMDIChildProc Lib "user32" Alias "DefMDIChildProcA" (ByVal hWnd As Long, ByVal wParam As Long, ByVal lParam As Long) As Long  
' Define information of the window (pointed to by hWnd)  
Declare Function GetWindowLong Lib "user32" Alias "GetWindowLongA" (ByVal hWnd As Long, ByVal nIndex As Long) As Long  
Declare Function SetWindowLong Lib "user32" Alias "SetWindowLongA" (ByVal hWnd As Long, ByVal nIndex As Long, ByVal dwNewLong As Long) As Long  
Declare Function CallWindowProc Lib "user32" Alias "CallWindowProcA" (ByVal lpPrevWndFunc As Long, ByVal hWnd As Long, ByVal Msg As Long, ByVal wParam As Long, ByVal lParam As Long) As Long  
  
Type WNDCLASS  
    style As Long  
    lpfnwndproc As Long  
    cbClsextra As Long  
    cbWndExtra2 As Long  
    hInstance As Long  
    hIcon As Long  
    hCursor As Long  
    hbrBackground As Long  
    lpzMenuName As String  
    lpzClassName As String  
End Type
```

```

Type POINTAPI
    x As Long
    y As Long
End Type

Type Msg
    hWnd As Long
    message As Long
    wParam As Long
    lParam As Long
    time As Long
    pt As POINTAPI
End Type

' Class styles
Public Const CS_VREDRAW = &H1
Public Const CS_HREDRAW = &H2
Public Const CS_KEYCVTWINDOW = &H4
Public Const CS_DBLCLKS = &H8
Public Const CS_OWNDC = &H20
Public Const CS_CLASSDC = &H40
Public Const CS_PARENTDC = &H80
Public Const CS_NOKEYCVT = &H100
Public Const CS_NOCLOSE = &H200
Public Const CS_SAVEBITS = &H800
Public Const CS_BYTEALIGNCLIENT = &H1000
Public Const CS_BYTEALIGNWINDOW = &H2000
Public Const CS_PUBLICCLASS = &H4000

' Window styles
Public Const WS_OVERLAPPED = &H0&
Public Const WS_POPUP = &H80000000
Public Const WS_CHILD = &H40000000
Public Const WS_MINIMIZE = &H20000000
Public Const WS_VISIBLE = &H10000000
Public Const WS_DISABLED = &H80000000
Public Const WS_CLIPSIBLINGS = &H40000000
Public Const WS_CLIPCHILDREN = &H20000000
Public Const WS_MAXIMIZE = &H1000000
Public Const WS_CAPTION = &HC00000 ' WS_BORDER Or WS_DLGFRAME
Public Const WS_BORDER = &H800000
Public Const WS_DLGFRAME = &H400000
Public Const WS_VSCROLL = &H200000
Public Const WS_HSCROLL = &H100000
Public Const WS_SYSMENU = &H80000
Public Const WS_THICKFRAME = &H40000
Public Const WS_GROUP = &H20000
Public Const WS_TABSTOP = &H10000
Public Const WS_MINIMIZEBOX = &H20000
Public Const WS_MAXIMIZEBOX = &H10000
Public Const WS_TILED = WS_OVERLAPPED
Public Const WS_ICONIC = WS_MINIMIZE
Public Const WS_SIZEBOX = WS_THICKFRAME
Public Const WS_OVERLAPPEDWINDOW = (WS_OVERLAPPED Or WS_CAPTION Or
WS_SYSMENU Or WS_THICKFRAME Or WS_MINIMIZEBOX Or WS_MAXIMIZEBOX)
Public Const WS_TILEDWINDOW = WS_OVERLAPPEDWINDOW

```



```
Public Const WS_POPUPWINDOW = (WS_POPUP Or WS_BORDER Or WS_SYSMENU)
Public Const WS_CHILDWINDOW = (WS_CHILD)
```

```
' ExWindowStyles
```

```
Public Const WS_EX_DLGMODALFRAME = &H1&
Public Const WS_EX_NOPARENTNOTIFY = &H4&
Public Const WS_EX_TOPMOST = &H8&
Public Const WS_EX_ACCEPTFILES = &H10&
Public Const WS_EX_TRANSPARENT = &H20&
```

```
' Color constants
```

```
Public Const COLOR_SCROLLBAR = 0
Public Const COLOR_BACKGROUND = 1
Public Const COLOR_ACTIVECAPTION = 2
Public Const COLOR_INACTIVECAPTION = 3
Public Const COLOR_MENU = 4
Public Const COLOR_WINDOW = 5
Public Const COLOR_WINDOWFRAME = 6
Public Const COLOR_MENUTEXT = 7
Public Const COLOR_WINDOWTEXT = 8
Public Const COLOR_CAPTIONTEXT = 9
Public Const COLOR_ACTIVEBORDER = 10
Public Const COLOR_INACTIVEBORDER = 11
Public Const COLOR_APPWORKSPACE = 12
Public Const COLOR_HIGHLIGHT = 13
Public Const COLOR_HIGHLIGHTTEXT = 14
Public Const COLOR_BTNFACE = 15
Public Const COLOR_BTNSHADOW = 16
Public Const COLOR_GRAYTEXT = 17
Public Const COLOR_BTNTEXT = 18
Public Const COLOR_INACTIVECAPTIONTEXT = 19
Public Const COLOR_BTNHIGHLIGHT = 20
```

```
' Window messages
```

```
Public Const WM_NULL = &H0
Public Const WM_CREATE = &H1
Public Const WM_DESTROY = &H2
Public Const WM_MOVE = &H3
Public Const WM_SIZE = &H5
```

```
' ShowWindow commands
```

```
Public Const SW_HIDE = 0
Public Const SW_SHOWNORMAL = 1
Public Const SW_NORMAL = 1
Public Const SW_SHOWMINIMIZED = 2
Public Const SW_SHOWMAXIMIZED = 3
Public Const SW_MAXIMIZE = 3
Public Const SW_SHOWNOACTIVATE = 4
Public Const SW_SHOW = 5
Public Const SW_MINIMIZE = 6
Public Const SW_SHOWMINNOACTIVE = 7
Public Const SW_SHOWNA = 8
Public Const SW_RESTORE = 9
Public Const SW_SHOWDEFAULT = 10
Public Const SW_MAX = 10
```

```
' Standard ID's of cursors
Public Const IDC_ARROW = 32512&
Public Const IDC_IBEAM = 32513&
Public Const IDC_WAIT = 32514&
Public Const IDC_CROSS = 32515&
Public Const IDC_UPARROW = 32516&
Public Const IDC_SIZE = 32640&
Public Const IDC_ICON = 32641&
Public Const IDC_SIZENWSE = 32642&
Public Const IDC_SIZENESW = 32643&
Public Const IDC_SIZEWE = 32644&
Public Const IDC_SIZENS = 32645&
Public Const IDC_SIZEALL = 32646&
Public Const IDC_NO = 32648&
Public Const IDC_APPSTARTING = 32650&
Public Const GWL_WNDPROC = -4

Dim hwnd2 As Long, hwnd3 As Long, old_proc As Long, new_proc As Long

Public Sub Main()
    'KPD-Team 1999
    'URL: http://www.allapi.net/
    'E-Mail: KPDTeam@Allapi.net
    Dim lngTemp As Long
    ' Register class
    If MyRegisterClass Then
        ' Window created?
        If MyCreateWindow Then
            ' Change the button's procedures
            ' Point to new address
            new_proc = GetMyWndProc(AddressOf ButtonProc)
            old_proc = SetWindowLong(hwnd2, GWL_WNDPROC, new_proc)
            ' Message loop
            MyMessageLoop
        End If
        ' Unregister Class
        MyUnregisterClass
    End If
End Sub

Private Function MyRegisterClass() As Boolean
    ' WNDCLASS-structure
    Dim wndcls As WNDCLASS
    wndcls.style = CS_HREDRAW + CS_VREDRAW
    wndcls.lpfwndproc = GetMyWndProc(AddressOf MyWndProc)
    wndcls.cbClsExtra = 0
    wndcls.cbWndExtra2 = 0
    wndcls.hInstance = App.hInstance
    wndcls.hIcon = 0
    wndcls.hCursor = LoadCursor(0, IDC_ARROW)
    wndcls.hbrBackground = COLOR_WINDOW
    wndcls.lpszMenuName = 0
    wndcls.lpszClassName = "myWindowClass"
    ' Register class
    MyRegisterClass = (RegisterClass(wndcls) <> 0)
End Function
```

```
Private Sub MyUnregisterClass()  
    UnregisterClass "myWindowClass", App.hInstance  
End Sub  
  
Private Function MyCreateWindow() As Boolean  
    Dim hWnd As Long  
    ' Create the window  
    hWnd = CreateWindowEx(0, "myWindowClass", "My Window", WS_OVERLAPPEDWINDOW, 0,  
0, 400, 300, 0, 0, App.hInstance, ByVal 0&)  
    ' The Button and Textbox are child windows  
    hWnd2 = CreateWindowEx(0, "Button", "My button", WS_CHILD, 50, 55, 100, 25, hWnd, 0,  
App.hInstance, ByVal 0&)  
    hWnd3 = CreateWindowEx(0, "edit", "My textbox", WS_CHILD, 50, 25, 100, 25, hWnd, 0,  
App.hInstance, ByVal 0&)  
    If hWnd <> 0 Then ShowWindow hWnd, SW_SHOWNORMAL  
    ' Show them  
    ShowWindow hWnd2, SW_SHOWNORMAL  
    ShowWindow hWnd3, SW_SHOWNORMAL  
    ' Go back  
    MyCreateWindow = (hWnd <> 0)  
End Function  
  
Private Function MyWndProc(ByVal hWnd As Long, ByVal message As Long, ByVal wParam As  
Long, ByVal lParam As Long) As Long  
    Select Case message  
        Case WM_DESTROY  
            ' Destroy window  
            PostQuitMessage (0)  
    End Select  
    ' calls the default window procedure  
    MyWndProc = DefWindowProc(hWnd, message, wParam, lParam)  
End Function  
  
Function GetMyWndProc(ByVal lWndProc As Long) As Long  
    GetMyWndProc = lWndProc  
End Function  
Private Sub MyMessageLoop()  
    Dim aMsg As Msg  
    Do While GetMessage(aMsg, 0, 0, 0)  
        DispatchMessage aMsg  
    Loop  
End Sub  
  
Private Function ButtonProc(ByVal hWnd As Long, ByVal message As Long, ByVal wParam As  
Long, ByVal lParam As Long) As Long  
    Dim x As Integer  
    If (message = 533) Then  
        x = MsgBox("You clicked on the button", vbOKOnly)  
    End If  
    ' calls the window procedure  
    ButtonProc = CallWindowProc(old_proc, hWnd, message, wParam, lParam)  
End Function
```

+ Ví dụ 3 : Tạo thủ tục VB

‘Minh hoạ cách Visual Basic cung cấp cho lập trình viên một thủ tục window mạnh hơn so với ‘thủ tục window mặc định của hệ điều hành window

*** Đặt đoạn mã sau vào trong một module. ***

```
Public Declare Function DefWindowProc Lib "user32" Alias "DefWindowProcA" (ByVal hwnd As Long, ByVal wParam As Long, ByVal lParam As Long) As Long
Public Declare Function SetWindowLong Lib "user32" Alias "SetWindowLongA" (ByVal hwnd As Long, ByVal nIndex As Long, ByVal dwNewLong As Long) As Long
```

```
Public Const GWL_WNDPROC = (-4)
```

‘Con trở tới một thủ tục window của Visual Basic

```
Public pVBProc as long ‘ (Biến trên mặc định mang giá trị 0).
```

‘Hàm sau thực hiện chức năng bọc. Tất cả công việc nó làm là gọi thủ tục window mặc định.

```
Public Function WindowProc (ByVal hwnd As Long, ByVal uMsg As Long, ByVal wParam As Long, ByVal lParam As Long) As Long
```

‘Gọi thủ tục window mặc định và trả về giá trị.

```
WindowProc = DefWindowProc (hwnd, uMsg, wParam, lParam)
```

```
End Function
```

*** Đặt đoạn mã sau vào nơi bạn muốn. ***

```
Dim retval As Long ‘giá trị trả về
```

```
If pVBProc = 0 Then
```

‘Cửa sổ Form1 đang dùng thủ tục của VB cung cấp. Chuyển qua dùng thủ tục mặc định

```
pVBProc = SetWindowLong (Form1.hwnd, GWL_WNDPROC, AddressOf WindowProc)
```

```
Else
```

‘Cửa sổ Form1 đang dùng thủ tục mặc định (qua hàm bao bọc). Chuyển qua dùng thủ tục của VB.

```
RetVal = SetWindowLong (Form1.hwnd, GWL_WNDPROC, pVBProc)
```

‘Lập pVBProc về 0 để chúng ta biết được thủ tục nào đang được dùng.

```
pVBProc = 0
```

```
End If
```

‘Bằng cách cho phép người dùng chuyển tới lui giữa các thủ tục, sự khác nhau trở nên rõ ràng

17. [DestroyWindow](#)

- *Thư viện* : user32.dll
- *Hệ điều hành* : Windows NT 3.1 or later; Windows 95 or later
- *Khai báo* :

```
Public Declare Function DestroyWindow Lib "user32" Alias "DestroyWindow" (ByVal hwnd As Long) As Long
```

- *Các tham số*
 - **hwnd** : Cn của cửa sổ sẽ phá hủy.
- *Mô tả* : Hàm DestroyWindow phá hủy cửa sổ (kể cả các cửa sổ con của nó). Hàm này sẽ gửi thông điệp WM_DESTROY và WM_NCDESTROY đến cửa sổ nhằm ngưng hoạt động cửa sổ đó và xoá bỏ focus tới nó. Hàm này cũng phá hủy menu, thông điệp trong hàng đợi, phá hủy timers, xoá bỏ quyền sở hữu clipboard,
- *Trị trả về* : Số nguyên khác 0 nếu thành công và bằng 0 nếu thất bại.
- *Các hàm liên quan* :
 - + MoveWindow
 - + IsWindowEnabled
- *Các ví dụ minh hoạ* :

+ Ví dụ 1: Move Window

```
Private Declare Function DestroyWindow Lib "user32" (ByVal hwnd As Long) As Long
Private Declare Function IsWindowEnabled Lib "user32" (ByVal hwnd As Long) As Long
Private Declare Function MoveWindow Lib "user32" (ByVal hwnd As Long, ByVal x As Long, ByVal y As Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal bRepaint As Long) As Long

Private Sub Form_Load()
    Dim bl As Boolean
    'Is the window enabled?
    bl = IsWindowEnabled(Me.hwnd)
    MsgBox "Is the form enabled? " + Str$(bl)
    'Move the window
    MoveWindow Me.hwnd, 0, 0, 200, 200, 1
    'Show the window
    Me.Show
    'Wait 5 seconds
    t = Timer
    Do
        'Show the remaining time in the form's caption
        Me.Caption = 5 - Int(Timer - t)
        DoEvents
    Loop Until Timer > t + 5
    'Destroy the window
    DestroyWindow Me.hwnd
End Sub
```

+ Ví dụ 2: New Start-button

```
Const WS_CHILD = &H40000000
Const WM_LBUTTONDOWN = &H201
Const WM_LBUTTONUP = &H202
Const SW_HIDE = 0
Const SW_NORMAL = 1

Private Type RECT
    Left As Long
    Top As Long
    Right As Long
    Bottom As Long
End Type

Private Declare Function FindWindowEx Lib "user32" Alias "FindWindowExA" (ByVal hWnd1 As Long, ByVal hWnd2 As Long, ByVal lpsz1 As String, ByVal lpsz2 As String) As Long
Private Declare Function FindWindow Lib "user32" Alias "FindWindowA" (ByVal lpClassName As String, ByVal lpWindowName As String) As Long
Private Declare Function ShowWindow Lib "user32" (ByVal hwnd As Long, ByVal nCmdShow As Long) As Long
Private Declare Function GetWindowRect Lib "user32" (ByVal hwnd As Long, lpRect As RECT) As Long
Private Declare Function CreateWindowEx Lib "user32" Alias "CreateWindowExA" (ByVal dwExStyle As Long, ByVal lpClassName As String, ByVal lpWindowName As String, ByVal dwStyle As Long, ByVal x As Long, ByVal y As Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal hWndParent As Long, ByVal hMenu As Long, ByVal hInstance As Long, lpParam As Any) As Long
Private Declare Function DestroyWindow Lib "user32" (ByVal hwnd As Long) As Long
```

```
Dim tWnd As Long, bWnd As Long, ncWnd As Long

Private Sub Form_Load()
    'KPD-Team 1998
    'URL: http://www.allapi.net/
    'E-Mail: KPDTeam@Allapi.net
    Dim R As RECT
    'Get the taskbar's window handle
    tWnd = FindWindow("Shell_TrayWnd", vbNullString)
    'Get the start-button's window handle
    bWnd = FindWindowEx(tWnd, ByVal 0&, "BUTTON", vbNullString)
    'Get the start button's position
    GetWindowRect bWnd, R
    'Create a new button
    ncWnd = CreateWindowEx(ByVal 0&, "BUTTON", "Hello !", WS_CHILD, 0, 0, R.Right - R.Left,
    R.Bottom - R.Top, tWnd, ByVal 0&, App.hInstance, ByVal 0&)
    'Show our button
    ShowWindow ncWnd, SW_NORMAL
    'Hide the start button
    ShowWindow bWnd, SW_HIDE
End Sub

Private Sub Form_Unload(Cancel As Integer)
    'show the start button
    ShowWindow bWnd, SW_NORMAL
    'destroy our button
    DestroyWindow ncWnd
End Sub
```

+ Ví dụ 3: Start In

```
Private Declare Function FindWindow Lib "user32" Alias "FindWindowA" (ByVal lpClassName As Long, ByVal lpWindowName As Long) As Long
Private Declare Function GetParent Lib "user32" (ByVal hwnd As Long) As Long
Private Declare Function SetParent Lib "user32" (ByVal hWndChild As Long, ByVal hWndNewParent As Long) As Long
Private Declare Function GetWindowThreadProcessId Lib "user32" (ByVal hwnd As Long, lpdwProcessId As Long) As Long
Private Declare Function GetWindow Lib "user32" (ByVal hwnd As Long, ByVal wCmd As Long) As Long
Private Declare Function LockWindowUpdate Lib "user32" (ByVal hwndLock As Long) As Long
Private Declare Function GetDesktopWindow Lib "user32" () As Long
Private Declare Function DestroyWindow Lib "user32" (ByVal hwnd As Long) As Long
Private Declare Function TerminateProcess Lib "kernel32" (ByVal hProcess As Long, ByVal uExitCode As Long) As Long
Private Declare Function GetCurrentProcess Lib "kernel32" () As Long
Private Declare Function Putfocus Lib "user32" Alias "SetFocus" (ByVal hwnd As Long) As Long

Const GW_HWNDNEXT = 2
Dim mWnd As Long

Function InstanceToWnd(ByVal target_pid As Long) As Long
    Dim test_hwnd As Long, test_pid As Long, test_thread_id As Long
    'Find the first window
    test_hwnd = FindWindow(ByVal 0&, ByVal 0&)
```

```
Do While test_hwnd <> 0
    'Check if the window isn't a child
    If GetParent(test_hwnd) = 0 Then
        'Get the window's thread
        test_thread_id = GetWindowThreadProcessId(test_hwnd, test_pid)
        If test_pid = target_pid Then
            InstanceToWnd = test_hwnd
            Exit Do
        End If
    End If
    'retrieve the next window
    test_hwnd = GetWindow(test_hwnd, GW_HWNDNEXT)
Loop
End Function

Private Sub Form_Load()
    'KPD-Team 1999
    'URL: http://www.allapi.net/
    'E-Mail: KPDTeam@Allapi.net
    Dim Pid As Long
    'Lock the window update
    LockWindowUpdate GetDesktopWindow
    'Execute notepad.Exe
    Pid = Shell("c:\windows\notepad.exe", vbNormalFocus)
    If Pid = 0 Then MsgBox "Error starting the app"
    'retrieve the handle of the window
    mWnd = InstanceToWnd(Pid)
    'Set the notepad's parent
    SetParent mWnd, Me.hwnd
    'Put the focus on notepad
    Putfocus mWnd
    'Unlock windowupdate
    LockWindowUpdate False
End Sub

Private Sub Form_Unload(Cancel As Integer)
    'Unload notepad
    DestroyWindow mWnd
    'End this program
    TerminateProcess GetCurrentProcess, 0
End Sub
```

18. EnableWindow

- *Thư viện* : user32.dll
- *Hệ điều hành* : Windows NT 3.1 or later; Windows 95 or later
- *Khai báo* :

Public Declare Function EnableWindow Lib "user32" Alias "EnableWindow" (ByVal hwnd As Long, ByVal fEnable As Long) As Long

- *Các tham số*
 - **hwnd** : một handle của cửa sổ (hoặc control) được enable hay bị disable.
 - **fEnable** : Nếu bằng 0, cửa sổ hoặc control sẽ bị disable. Nếu khác 0, cửa sổ sẽ được enable

- *Mô tả* : Hàm EnableWindow dùng để làm tích cực (enable) hoặc bị động (disable) một cửa sổ hoặc một điều khiển người dùng (control). Nếu một cửa sổ hoặc một control bị disable, nó sẽ không nhận focus tức không thể tương tác với người dùng. Một vài loại control như nút nhấn và các loại control khác sẽ có màu xám khi bị disable, mặc dù bất kỳ cửa sổ nào cũng được enable hay bị disable.
- *Trị trả về* : Hàm sẽ trả về 0 nếu trước đó cửa sổ (hoặc control) đang enable; trả về giá trị khác 0 nếu trước đó nó bị disable.
- *Các hàm liên quan* : IsWindowEnabled
- *Các ví dụ minh họa* :
 - + Ví dụ 1 : EnableWindow

```
'This project needs two command buttons
Private Declare Function IsWindowEnabled Lib "user32" (ByVal hwnd As Long) As Long
Private Declare Function EnableWindow Lib "user32" (ByVal hwnd As Long, ByVal fEnable As Long) As Long

Private Sub Command2_Click()
    'KPD-Team 1999
    'URL: http://www.allapi.net/
    'E-Mail: KPDTTeam@Allapi.net

    ' Reverse the enabled status of Command1. If the window is
    ' disabled, enable it; if it is enabled, disable it.
    Dim wasenabled As Long ' receives enabled/disabled status of Command1
    Dim retval As Long ' return value
    ' Determine if the window Command1 is currently enabled or not.
    wasenabled = IsWindowEnabled(Command1.hwnd)
    If wasenabled = 0 Then ' if not enabled, enable it
        retval = EnableWindow(Command1.hwnd, 1)
    Else ' if enabled, disable it
        retval = EnableWindow(Command1.hwnd, 0)
    End If
End Sub
```

- + Ví dụ 2 : Đảo trạng thái enable của control Command1

```
'Đảo trạng thái enable của control command1, nếu control đang bị
' disable thì cho phép hoạt động; nếu đang hoạt động thì disable nó .
Dim wasenabled as long 'chứa trạng thái enable/disable của command1
Dim retval as long 'giá trị trả về'
'Xác định control command1 hiện tại được enable hay không.
Wasenabled = IsWindowEnabled(command1.hwnd)
If wasenabled = 0 then 'nếu không được enable thì enable nó retval = EnableWindow
(command1.hwnd,1)
Else 'nếu đang được enable thì disable nó
    Retval = EnableWindow (command1.hwnd,0)
End if.
```

19. EndDeferWindowPos

- *Thư viện* : user32.dll
- *Hệ điều hành* : Windows NT 3.1 or later; Windows 95 or later
- *Khai báo* :


```
Public Declare Function EndDeferWindowPos Lib "user32" Alias "EndDeferWindowPos"
(ByVal hWndPosInfo As Long) As Long
```

- Các tham số
 - **hWnd**
- Mô tả : Hàm EndDeferWindowPos cập nhật vị trí, kích thước và tình trạng của một hoặc nhiều cửa sổ trong một chu trình là tươi (refreshing) của màn hình đơn.
- Các hàm liên quan :
 - + BeginDeferWindowPos
 - + DeferWindowPos
- Các ví dụ minh họa : Xem ví dụ hàm DeferWindowPos (hàm số 15).

20. EnumChildWindows

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

```
Public Declare Function EnumChildWindows Lib "user32" Alias "EnumChildWindows"
(ByVal hWndParent As Long, ByVal lpEnumFunc As Long, ByVal lParam As Long) As Long
```

- Các tham số
 - **hWndParent** : Một handle của cửa sổ cha mà ta cần liệt kê các cửa sổ con của nó.
 - **lpEnumFunc** : Con trỏ chỉ tới hàm callback EnumChildProc.
 - **lParam** : Giá trị mở rộng truyền vào hàm callback.
- Mô tả : Hàm EnumChildWindows liệt kê và cung cấp các handle cho tất cả các cửa sổ con của một cửa sổ. Hàm này cũng liệt kê bất kỳ cửa sổ con nào của các cửa sổ con. Mỗi lần một cửa sổ con được xác định, hàm sẽ truyền handle của nó vào một hàm callback được định nghĩa trong chương trình. Hàm tiếp tục thực hiện cho đến khi tất cả các cửa sổ con đều được liệt kê, hoặc cho đến khi tiến trình bị ngừng (hàm callback trả về FALSE). Phải có Custom Control CBK.VBX mới sử dụng được.
- Trị trả về : Nếu xảy ra lỗi, hàm trả về 0 (dùng hàm GetLastError để nhận mã lỗi. Nếu thành công hàm trả về một giá trị khác 0.
- Các hàm liên quan :
 - + EnumThreadWindows
 - + EnumWindows
- Các ví dụ minh họa : EnumChildWindows

```
'in a form
Private Sub Form_Load()
    'KPD-Team 2000
    'URL: http://www.allapi.net/
    'E-Mail: KPDTTeam@Allapi.net
    Me.AutoRedraw = True
    EnumChildWindows GetDesktopWindow, AddressOf EnumChildProc, ByVal 0&
End Sub

'in a module
Declare Function GetDesktopWindow Lib "user32" () As Long
Declare Function EnumChildWindows Lib "user32" (ByVal hWndParent As Long, ByVal lpEnumFunc As Long, ByVal lParam As Long) As Long
Declare Function GetWindowText Lib "user32" Alias "GetWindowTextA" (ByVal hwnd As Long,
```

```
ByVal lpString As String, ByVal cch As Long) As Long
Declare Function GetWindowTextLength Lib "user32" Alias "GetWindowTextLengthA" (ByVal hwnd
As Long) As Long

Public Function EnumChildProc(ByVal hwnd As Long, ByVal lParam As Long) As Long
    Dim sSave As String
    'Get the windowtext length
    sSave = Space$(GetWindowTextLength(hwnd) + 1)
    'get the window text
    GetWindowText hwnd, sSave, Len(sSave)
    'remove the last Chr$(0)
    sSave = Left$(sSave, Len(sSave) - 1)
    If sSave <> "" Then Form1.Print sSave
    'continue enumeration
    EnumChildProc = 1
End Function
```

21. EnumThreadWindows

- *Thư viện* : user32.dll
- *Hệ điều hành* : Windows NT 3.1 or later; Windows 95 or later
- *Khai báo* :

```
Public Declare Function EnumThreadWindows Lib "user32" Alias "EnumThreadWindows"
(ByVal dwThreadId As Long, ByVal lpfn As Long, ByVal lParam As Long) As Long
```

- *Các tham số*
 - **dwThreadId** : Một định danh của thread cần liệt kê các cửa sổ của nó.
 - **lpfn** : Một con trỏ của hàm callback EnumThreadWndProc.
 - **lParam** : Giá trị mở rộng được truyền vào hàm callback.
- *Mô tả* : Hàm EnumThreadWindows liệt kê và cung cấp các handle cho tất cả các cửa sổ được sở hữu và điều khiển bởi một thread. (Chú ý, các cửa sổ này bao gồm cả các cửa sổ không nhìn thấy được đối với người dùng). Mỗi lần một cửa sổ được định vị, hàm truyền handle của nó cho hàm callback. Hàm tiếp tục thực hiện cho đến khi tất cả các cửa sổ đều được liệt kê, hoặc cho đến khi tiến trình bị bỏ dở.
- *Trị trả về* : Nếu xảy ra lỗi, hàm trả về giá trị 0 (dùng hàm GetLastError để lấy mã lỗi). Nếu thành công, hàm trả về một giá trị khác 0.
- *Các hàm liên quan* :
 - + EnumChildWindows
 - + EnumWindows
 - + GetCurrentThread
- *Các ví dụ minh họa* : Enum Classnames

```
'in a form
Private Sub Form_Load()
    'KPD-Team 2000
    'URL: http://www.allapi.net/
    'E-Mail: KPDTTeam@Allapi.net
    Dim ThreadID As Long, ProcessID As Long ' receive id to thread and process of Form1
    ' Determine the thread which owns this window
    ThreadID = GetWindowThreadProcessId(Me.hWnd, ProcessID)
    ' Use the callback function to list all of the enumerated thrad windows
    EnumThreadWindows ThreadID, AddressOf EnumThreadWndProc, 0
    'Show the results
```

```
Me.AutoRedraw = True
Me.Print sClasses
End Sub

'In a module
Declare Function GetWindowThreadProcessId Lib "user32" (ByVal hWnd As Long, lpdwProcessId As Long) As Long
Declare Function EnumThreadWindows Lib "user32" (ByVal dwThreadId As Long, ByVal lpfn As Long, ByVal lParam As Long) As Long
Declare Function GetClassName Lib "user32" Alias "GetClassNameA" (ByVal hWnd As Long, ByVal lpClassName As String, ByVal nMaxCount As Long) As Long

'variable used to list all the classnames
Public sClasses As String

Public Function EnumThreadWndProc(ByVal hWnd As Long, ByVal lParam As Long) As Long
    Dim Ret As Long, sText As String
    'create a string-buffer
    sText = Space(255)
    'get the classname of the window handle
    Ret = GetClassName(hWnd, sText, 255)
    'cut off the unnecessary part of Chr$(0)'s
    sText = Left$(sText, Ret)
    'add this classname to the list of classnames
    sClasses = sClasses + sText + vbCrLf
    'continue the enumeration
    EnumThreadWndProc = 1
End Function
```

22. [EnumWindows](#)

- *Thư viện* : user32.dll
- *Hệ điều hành* : Windows NT 3.1 or later; Windows 95 or later
- *Khai báo* :

```
Public Declare Function EnumWindows Lib "user32.dll" ( ByVal lpEnumFunc As Long, ByVal lParam As Long) As Long
```

- *Các tham số*
 - **lpEnumFunc** : Một con trỏ chỉ đến hàm callback EnumWindowsProc
 - **lParam** : Một giá trị mở rộng được truyền vào hàm callback.
- *Mô tả* : Hàm EnumWindows liệt kê và cung cấp handle cho tất cả các cửa sổ top-level hiện đang mở. Hàm này sẽ bỏ qua các cửa sổ con. (Chú ý các cửa sổ top-level bao gồm cả các cửa sổ không hiện ra). Mỗi lần một cửa sổ được định vị, hàm sẽ truyền handle đó cho một hàm callback. Hàm này liên tục thực hiện cho đến khi tất cả các cửa sổ đều được liệt kê hay cho đến khi tiến trình được bỏ qua.
- *Trị trả về* : Nếu xảy ra lỗi, hàm trả về 0 (dùng hàm GetLastError để nhận mã lỗi). Nếu thành công, hàm trả về một giá trị khác 0.
- *Các hàm liên quan* :
 - + EnumChildWindows
 - + EnumThreadWindows
- *Các ví dụ minh họa* : EnumWindows

```
'Add this code to a form
Private Sub Form_Load()
    'KPD-Team 2000
    'URL: http://www.allapi.net/
    'E-Mail: KPDTeam@Allapi.net
    'Set the form's graphics mode to persistent
    Me.AutoRedraw = True
    'call the Enumwindows-function
    EnumWindows AddressOf EnumWindowsProc, ByVal 0&
End Sub

'Add this code to a module
Declare Function EnumWindows Lib "user32" (ByVal lpEnumFunc As Long, ByVal lParam As Long)
As Boolean
Declare Function GetWindowText Lib "user32" Alias "GetWindowTextA" (ByVal hwnd As Long,
ByVal lpString As String, ByVal cch As Long) As Long
Declare Function GetWindowTextLength Lib "user32" Alias "GetWindowTextLengthA" (ByVal hwnd
As Long) As Long

Public Function EnumWindowsProc(ByVal hwnd As Long, ByVal lParam As Long) As Boolean
    Dim sSave As String, Ret As Long
    Ret = GetWindowTextLength(hwnd)
    sSave = Space(Ret)
    GetWindowText hwnd, sSave, Ret + 1
    Form1.Print Str$(hwnd) + " " + sSave
    'continue enumeration
    EnumWindowsProc = True
End Function
```

23. EnumWindowStations

- *Thư viện* : user32.dll
- *Hệ điều hành* : Windows NT 3.1 or later; Windows 95 or later
- *Khai báo* :

Public	Declare	Function	EnumWindowStations	Lib	"user32"	Alias
"EnumWindowStationsA" (ByVal lpEnumFunc As Long, ByVal lParam As Long) As Long						

- *Các tham số*
 - **lpEnumFunc** : Biến con trỏ chỉ đến hàm để gọi đối với mỗi cửa sổ con (mức Child). Sử dụng tính chất ProcAddress của Custom Control CBK.VBX để nhận hàm biến trỏ (function pointer) để gọi lại (callbacks).
 - **lParam** : Giá trị chuyển đến cho sự kiện EnumWindows của Custom Control trong lúc liệt kê. Ý nghĩa của trị này do lập trình viên xác định.
- *Mô tả* : Liệt kê danh sách cửa sổ cấp trên, chứa cửa sổ khai báo. Phải có Custom Control CBK.VBX mới sử dụng được.
- *Trị trả về* : Hàm trả về TRUE (khác 0) nếu thành công, ngược lại hàm trả về FALSE (bằng 0) nếu thất bại.
- *Các hàm liên quan* :
 - + EnumChildWindows
 - + EnumThreadWindows
 - + EnumWindows

24. FindWindow

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

```
Public Declare Function FindWindow Lib "user32" Alias "FindWindowA" (ByVal lpClassName As String, ByVal lpWindowName As String) As Long
```

- Các tham số
 - **lpClassName** : Tên của lớp cửa sổ hoặc cửa sổ cần tìm.
 - **lpWindowName** : Tên tựa của cửa sổ cần tìm, nếu tham số này là NULL thì tất cả các tên cửa sổ đều hợp lệ.
- Mô tả : Hàm FindWindow tìm tất cả các cửa sổ phù hợp với một tên lớp cửa sổ và / hoặc tên cửa sổ (chính là tiêu đề của cửa sổ). Hàm tìm kiếm không phân biệt chữ hoa hay chữ thường. Nếu bạn không muốn chỉ rõ một không số nào đó, hãy truyền chuỗi rỗng cho nó (để truyền chuỗi rỗng cho một thông số, dùng hằng vbNullString).
- Trị trả về : Nếu xảy ra lỗi, hoặc không tìm thấy cửa sổ thích hợp, hàm trả về 0 (dùng hàm GetLastError để nhận mã lỗi). Nếu thành công hàm sẽ trả về một handle của cửa sổ tìm thấy.
- Các hàm liên quan :
 - + FindWindowEx
 - + GetActiveWindow
 - + GetForegroundWindow
- Các ví dụ minh họa :
 - + Ví dụ 1 : Get Classname

```
Private Declare Function FindWindow Lib "user32" Alias "FindWindowA" (ByVal lpClassName As String, ByVal lpWindowName As String) As Long
Private Declare Function PostMessage Lib "user32" Alias "PostMessageA" (ByVal hwnd As Long, ByVal wParam As Long, ByVal lParam As Any) As Long
Private Declare Function GetClassName Lib "user32" Alias "GetClassNameA" (ByVal hwnd As Long, ByVal lpClassName As String, ByVal nMaxCount As Long) As Long
Private Declare Function ShowWindow Lib "user32" (ByVal hwnd As Long, ByVal nCmdShow As Long) As Long

Const SW_SHOWNORMAL = 1
Const WM_CLOSE = &H10
Const gcClassNameMSWord = "OpusApp"
Const gcClassNameMSExcel = "XLMAIN"
Const gcClassNameMSIExplorer = "IEFrame"
Const gcClassNameMSVBasic = "wndclass_desked_gsk"
Const gcClassNameNotePad = "Notepad"
Const gcClassNameMyVBApp = "ThunderForm"

Private Sub Form_Load()
    'KPD-Team 1998
    'URL: http://www.allapi.net/
    'E-Mail: KPDTTeam@Allapi.net
    Dim WinWnd As Long, Ret As String, RetVal As Long, lpClassName As String
    'Ask for a Window title
    Ret = InputBox("Enter the exact window title:" + Chr$(13) + Chr$(10) + "Note: must be an exact match")
    'Search the window
    WinWnd = FindWindow(vbNullString, Ret)
```

```
If WinWnd = 0 Then MsgBox "Couldn't find the window ...": Exit Sub
'Show the window
ShowWindow WinWnd, SW_SHOWNORMAL
'Create a buffer
lpClassName = Space(256)
'retrieve the class name
RetVal = GetClassName(WinWnd, lpClassName, 256)
'Show the classname
MsgBox "Classname: " + Left$(lpClassName, RetVal)
'Post a message to the window to close itself
PostMessage WinWnd, WM_CLOSE, 0&, 0&
End Sub
```

+ *Vi du 2*: Hide Start-button

```
Const SWP_HIDEWINDOW = &H80
Const SWP_SHOWWINDOW = &H40
Const GW_CHILD = 5
Const GW_HWNDNEXT = 2

Private Declare Function SetWindowPos Lib "user32" (ByVal hwnd As Long, ByVal hWndInsertAfter As Long, ByVal x As Long, ByVal y As Long, ByVal cx As Long, ByVal cy As Long, ByVal wFlags As Long) As Long
Private Declare Function FindWindow Lib "user32" Alias "FindWindowA" (ByVal lpClassName As String, ByVal lpWindowName As String) As Long
Private Declare Function GetWindow Lib "user32" (ByVal hwnd As Long, ByVal wCmd As Long) As Long
Private Declare Function GetClassName Lib "user32" Alias "GetClassNameA" (ByVal hwnd As Long, ByVal lpClassName As String, ByVal nMaxCount As Long) As Long

Dim tWnd As Long, bWnd As Long, sSave As String * 250

Private Sub Form_Load()
    'KPD-Team 1999
    'URL: http://www.allapi.net/
    'E-Mail: KPDTeam@Allapi.net

    'This code will hide the Start-button
    'Find the taskbar's handle
    tWnd = FindWindow("Shell_traywnd", vbNullString)
    'Search for a child window
    bWnd = GetWindow(tWnd, GW_CHILD)
    Do
        'get the child window's classname
        GetClassName bWnd, sSave, 250
        'We have the handle of the Start button If the classname is 'button'
        If LCase(Left$(sSave, 6)) = "button" Then Exit Do
        'Search the next child
        bWnd = GetWindow(bWnd, GW_HWNDNEXT)
    Loop
    'Hide the start button
    SetWindowPos bWnd, 0, 0, 0, 0, 0, SWP_HIDEWINDOW
End Sub

Private Sub Form_Unload(Cancel As Integer)
```

```
'Show the start button
SetWindowPos bWnd, 0, 0, 0, 0, 0, SWP_SHOWWINDOW
End Sub
```

- + Ví dụ 3 : New Start-button (xem ví dụ 2 hàm DestroyWindow)
- + Ví dụ 4 : Start In (xem ví dụ 3 hàm DestroyWindow)
- + Ví dụ 5 : ForeGround Window

```
Private Declare Function FindWindow Lib "user32" Alias "FindWindowA" (ByVal lpClassName As String, ByVal lpWindowName As String) As Long
Private Declare Function SetForegroundWindow Lib "user32" (ByVal hwnd As Long) As Long

Private Sub Form_Load()
    'KPD-Team 1999
    'URL: http://www.allapi.net/
    'E-Mail: KPDTTeam@Allapi.net
    Dim lHandle As Long
    'First we're going to retrieve the handle of this window
    ' "ThunderRT5Form" is the classname of a VB-window
    lHandle = FindWindow("ThunderRT5Form", Me.Caption)
    'Set this window to the foreground
    lHandle = SetForegroundWindow(lHandle)
End Sub
```

25. FindWindowEx

- *Thư viện* : user32.dll
- *Hệ điều hành* : Windows NT 3.1 or later; Windows 95 or later
- *Khai báo* :

```
Public Declare Function FindWindowEx Lib "user32" Alias "FindWindowExA" (ByVal hWnd1 As Long, ByVal hWnd2 As Long, ByVal lpsz1 As String, ByVal lpsz2 As String) As Long
```

- *Các tham số*
 - **hWnd**
- *Mô tả* :
- *Trị trả về* :
- *Các hàm liên quan* :
- *Các ví dụ minh họa* :

26. FlashWindow

- *Thư viện* : user32.dll
- *Hệ điều hành* : Windows NT 3.1 or later; Windows 95 or later
- *Khai báo* :

```
Public Declare Function FlashWindow Lib "user32" Alias "FlashWindow" (ByVal hwnd As Long, ByVal bInvert As Long) As Long
```

- *Các tham số*
 - **hWnd**
- *Mô tả* :

- *Trị trả về :*
- *Các hàm liên quan :*
- *Các ví dụ minh hoạ :*

27. FlashWindowEx

- *Thư viện :* user32.dll
- *Hệ điều hành :* Windows NT 3.1 or later; Windows 95 or later
- *Khai báo :*

```
Public Declare Function FlashWindowEx Lib "user32.dll" (ByRef pfw As PFLASHWINFO) As Long
```

- *Các tham số*
 - **hWnd**
- *Mô tả :*
- *Trị trả về :*
- *Các hàm liên quan :*
- *Các ví dụ minh hoạ :*

28. GetActiveWindow

- *Thư viện :* user32.dll
- *Hệ điều hành :* Windows NT 3.1 or later; Windows 95 or later
- *Khai báo :*

```
Public Declare Function GetActiveWindow Lib "user32" Alias "GetActiveWindow" () As Long
```

- *Các tham số*
 - **hWnd**
- *Mô tả :*
- *Trị trả về :*
- *Các hàm liên quan :*
- *Các ví dụ minh hoạ :*

29. GetClassInfo

- *Thư viện :* user32.dll
- *Hệ điều hành :* Windows NT 3.1 or later; Windows 95 or later
- *Khai báo :*

```
Public Declare Function GetClassInfo Lib "user32" Alias "GetClassInfoA" (ByVal hInstance As Long, ByVal lpClassName As String, lpWndClass As WNDCLASS) As Long
```

- *Các tham số*
 - **hWnd**
- *Mô tả :*
- *Trị trả về :*
- *Các hàm liên quan :*
- *Các ví dụ minh hoạ :*

30. GetClassInfoEx

- *Thư viện :* user32.dll
- *Hệ điều hành :* Windows NT 3.1 or later; Windows 95 or later
- *Khai báo :*


```
Public Declare Function GetClassInfoEx Lib "user32.dll" Alias "GetClassInfoExA" ( ByVal hinstance As Long, ByVal lpctr As String, ByRef lpwndclassex As WNDCLASSEX) As Long
```

- Các tham số
 - **hWnd**
- Mô tả :
- Trị trả về :
- Các hàm liên quan :
- Các ví dụ minh họa :

31. GetClassLong

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

```
Public Declare Function GetClassLong Lib "user32" Alias "GetClassLongA" (ByVal hwnd As Long, ByVal nIndex As Long) As Long
```

- Các tham số
 - **hWnd**
- Mô tả :
- Trị trả về :
- Các hàm liên quan :
- Các ví dụ minh họa :

32. GetClassName

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

```
Public Declare Function GetClassName Lib "user32" Alias "GetClassNameA" (ByVal hwnd As Long, ByVal lpClassName As String, ByVal nMaxCount As Long) As Long
```

- Các tham số
 - **hWnd**
- Mô tả :
- Trị trả về :
- Các hàm liên quan :
- Các ví dụ minh họa :

33. GetDesktopWindow

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

```
Public Declare Function GetDesktopWindow Lib "user32" Alias "GetDesktopWindow" () As Long
```

- Các tham số
 - **hWnd**
- Mô tả :
- Trị trả về :
- Các hàm liên quan :

- Các ví dụ minh họa :

34. GetParent

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

```
Public Declare Function GetParent Lib "user32" Alias "GetParent" (ByVal hwnd As Long) As Long
```

- Các tham số
 - **hwnd**
- Mô tả :
- Trị trả về :
- Các hàm liên quan :
- Các ví dụ minh họa :

35. GetWindow

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

```
Public Declare Function GetWindow Lib "user32" Alias "GetWindow" (ByVal hwnd As Long, ByVal wCmd As Long) As Long
```

- Các tham số
 - **hwnd**
- Mô tả :
- Trị trả về :
- Các hàm liên quan :
- Các ví dụ minh họa :

36. GetWindowLong

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

```
Public Declare Function GetWindowLong Lib "user32" Alias "GetWindowLongA" (ByVal hwnd As Long, ByVal nIndex As Long) As Long
```

- Các tham số
 - **hwnd**
- Mô tả :
- Trị trả về :
- Các hàm liên quan :
- Các ví dụ minh họa :

37. GetWindowRect

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

```
Public Declare Function GetWindowRect Lib "user32" Alias "GetWindowRect" (ByVal hwnd As Long, lpRect As RECT) As Long
```

- Các tham số
 - **hWnd**
- Mô tả :
- Trị trả về :
- Các hàm liên quan :
- Các ví dụ minh họa :

38. GetWindowText

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

```
Public Declare Function GetWindowText Lib "user32" Alias "GetWindowTextA" (ByVal  
hwnd As Long, ByVal lpString As String, ByVal cch As Long) As Long
```

- Các tham số
 - **hWnd**
- Mô tả :
- Trị trả về :
- Các hàm liên quan :
- Các ví dụ minh họa :

39. GetWindowTextLength

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

```
Public Declare Function GetWindowTextLength Lib "user32" Alias  
"GetWindowTextLengthA" (ByVal hwnd As Long) As Long
```

- Các tham số
 - **hWnd**
- Mô tả :
- Trị trả về :
- Các hàm liên quan :
- Các ví dụ minh họa :

40. GetWindowThreadProcessId

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

```
Public Declare Function GetWindowThreadProcessId Lib "user32" Alias  
"GetWindowThreadProcessId" (ByVal hwnd As Long, lpdwProcessId As Long) As Long
```

- Các tham số
 - **hWnd**
- Mô tả :
- Trị trả về :
- Các hàm liên quan :
- Các ví dụ minh họa :

41. GetWindowWord

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

Public Declare Function GetWindowWord Lib "user32" Alias "GetWindowWord" (ByVal hwnd As Long, ByVal nIndex As Long) As Integer

- Các tham số
 - hWnd
- Mô tả :
- Trị trả về :
- Các hàm liên quan :
- Các ví dụ minh họa :

42. InvalidateRect

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

Public Declare Function InvalidateRect Lib "user32" Alias "InvalidateRect" (ByVal hwnd As Long, lpRect As RECT, ByVal bErase As Long) As Long

- Các tham số
 - hWnd
- Mô tả :
- Trị trả về :
- Các hàm liên quan :
- Các ví dụ minh họa :

43. IsChild

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

Public Declare Function IsChild Lib "user32" Alias "IsChild" (ByVal hwndParent As Long, ByVal hwnd As Long) As Long

- Các tham số
 - hWnd
- Mô tả :
- Trị trả về :
- Các hàm liên quan :
- Các ví dụ minh họa :

44. IsIconic

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

Public Declare Function IsIconic Lib "user32" Alias "IsIconic" (ByVal hwnd As Long) As Long

- Các tham số
 - hWnd
- Mô tả :

- *Trị trả về :*
- *Các hàm liên quan :*
- *Các ví dụ minh hoạ :*

45. IsRectEmpty

- *Thư viện :* user32.dll
- *Hệ điều hành :* Windows NT 3.1 or later; Windows 95 or later
- *Khai báo :*

```
Public Declare Function IsRectEmpty Lib "user32" Alias "IsRectEmpty" (lpRect As RECT) As Long
```

- *Các tham số*
 - **hWnd**
- *Mô tả :*
- *Trị trả về :*
- *Các hàm liên quan :*
- *Các ví dụ minh hoạ :*

46. IsWindow

- *Thư viện :* user32.dll
- *Hệ điều hành :* Windows NT 3.1 or later; Windows 95 or later
- *Khai báo :*

```
Public Declare Function IsWindow Lib "user32" Alias "IsWindow" (ByVal hwnd As Long) As Long
```

- *Các tham số*
 - **hWnd**
- *Mô tả :*
- *Trị trả về :*
- *Các hàm liên quan :*
- *Các ví dụ minh hoạ :*

47. IsWindowEnabled

- *Thư viện :* user32.dll
- *Hệ điều hành :* Windows NT 3.1 or later; Windows 95 or later
- *Khai báo :*

```
Public Declare Function IsWindowEnabled Lib "user32" Alias "IsWindowEnabled" (ByVal hwnd As Long) As Long
```

- *Các tham số*
 - **hWnd**
- *Mô tả :*
- *Trị trả về :*
- *Các hàm liên quan :*
- *Các ví dụ minh hoạ :*

48. IsWindowVisible

- *Thư viện :* user32.dll
- *Hệ điều hành :* Windows NT 3.1 or later; Windows 95 or later
- *Khai báo :*

Public Declare Function IsWindowVisible Lib "user32" Alias "IsWindowVisible" (ByVal hwnd As Long) As Long

- Các tham số
 - hWnd
- Mô tả :
- Trị trả về :
- Các hàm liên quan :
- Các ví dụ minh họa :

49. IsZoomed

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

Public Declare Function IsZoomed Lib "user32" Alias "IsZoomed" (ByVal hwnd As Long) As Long

- Các tham số
 - hWnd
- Mô tả :
- Trị trả về :
- Các hàm liên quan :
- Các ví dụ minh họa :

50. LockWindowUpdate

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

Public Declare Function LockWindowUpdate Lib "user32" Alias "LockWindowUpdate" (ByVal hwndLock As Long) As Long

- Các tham số
 - hWnd
- Mô tả :
- Trị trả về :
- Các hàm liên quan :
- Các ví dụ minh họa :

51. MapWindowPoints

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

Public Declare Function MapWindowPoints Lib "user32" Alias "MapWindowPoints" (ByVal hwndFrom As Long, ByVal hwndTo As Long, lppt As Any, ByVal cPoints As Long) As Long

- Các tham số
 - hWnd
- Mô tả :
- Trị trả về :
- Các hàm liên quan :

- Các ví dụ minh họa :

52. MoveWindow

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

Public Declare Function MoveWindow Lib "user32" Alias "MoveWindow" (ByVal hwnd As Long, ByVal x As Long, ByVal y As Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal bRepaint As Long) As Long

- Các tham số
 - **hwnd**
- Mô tả :
- Trị trả về :
- Các hàm liên quan :
- Các ví dụ minh họa :

53. PostMessage

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

Public Declare Function PostMessage Lib "user32" Alias "PostMessageA" (ByVal hwnd As Long, ByVal wParam As Long, ByVal lParam As Long, ByVal IPParam As Long) As Long

- Các tham số
 - **hwnd**
- Mô tả :
- Trị trả về :
- Các hàm liên quan :
- Các ví dụ minh họa :

54. RedrawWindow

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

Public Declare Function RedrawWindow Lib "user32" Alias "RedrawWindow" (ByVal hwnd As Long, lprcUpdate As RECT, ByVal hrgnUpdate As Long, ByVal fuRedraw As Long) As Long

- Các tham số
 - **hwnd**
- Mô tả :
- Trị trả về :
- Các hàm liên quan :
- Các ví dụ minh họa :

55. RegisterClass

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

```
Public Declare Function RegisterClass Lib "user32" Alias "RegisterClassA" Alias  
"RegisterClass" (Class As WNDCLASS) As Long
```

- Các tham số
 - **hWnd**
- Mô tả :
- Trị trả về :
- Các hàm liên quan :
- Các ví dụ minh họa :

56. RegisterClassEx

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

```
Public Declare Function RegisterClassEx Lib "user32" Alias "RegisterClassExA"  
(pcWndClassEx As WNDCLASSEX) As Integer
```

- Các tham số
 - **hWnd**
- Mô tả :
- Trị trả về :
- Các hàm liên quan :
- Các ví dụ minh họa :

57. ScreenToClient

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

```
Public Declare Function ScreenToClient Lib "user32" Alias "ScreenToClient" (ByVal hwnd  
As Long, lpPoint As POINTAPI) As Long
```

- Các tham số
 - **hWnd**
- Mô tả :
- Trị trả về :
- Các hàm liên quan :
- Các ví dụ minh họa :

58. SendMessage

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

```
Public Declare Function SendMessage Lib "user32" Alias "SendMessageA" (ByVal hwnd  
As Long, ByVal wParam As Long, ByVal lParam As Any) As Long
```

- Các tham số
 - **hWnd**
- Mô tả :
- Trị trả về :
- Các hàm liên quan :
- Các ví dụ minh họa :

59. SetActiveWindow

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

```
Public Declare Function SetActiveWindow Lib "user32" Alias "SetActiveWindow" (ByVal hwnd As Long) As Long
```

- Các tham số
 - **hwnd**
- Mô tả :
- Trị trả về :
- Các hàm liên quan :
- Các ví dụ minh họa :

60. SetClassLong

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

```
Public Declare Function SetClassLong Lib "user32" Alias "SetClassLongA" (ByVal hwnd As Long, ByVal nIndex As Long, ByVal dwNewLong As Long) As Long
```

- Các tham số
 - **hwnd**
- Mô tả :
- Trị trả về :
- Các hàm liên quan :
- Các ví dụ minh họa :

61. SetForegroundWindow

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

```
Public Declare Function SetForegroundWindow Lib "user32" Alias "SetForegroundWindow" (ByVal hwnd As Long) As Long
```

- Các tham số
 - **hwnd**
- Mô tả :
- Trị trả về :
- Các hàm liên quan :
- Các ví dụ minh họa :

62. SetParent

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

```
Public Declare Function SetParent Lib "user32" Alias "SetParent" (ByVal hWndChild As Long, ByVal hWndNewParent As Long) As Long
```

- Các tham số

- **hWnd**
- Mô tả :
- Trị trả về :
- Các hàm liên quan :
- Các ví dụ minh họa :

63. SetWindowLong

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

```
Public Declare Function SetWindowLong Lib "user32" Alias "SetWindowLongA" (ByVal hwnd As Long, ByVal nIndex As Long, ByVal dwNewLong As Long) As Long
```

- Các tham số
 - **hWnd**
- Mô tả :
- Trị trả về :
- Các hàm liên quan :
- Các ví dụ minh họa :

64. SetWindowPos

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

```
Public Declare Function SetWindowPos Lib "user32" Alias "SetWindowPos" (ByVal hwnd As Long, ByVal hWndInsertAfter As Long, ByVal x As Long, ByVal y As Long, ByVal cx As Long, ByVal cy As Long, ByVal wFlags As Long) As Long
```

- Các tham số
 - **hWnd**
- Mô tả :
- Trị trả về :
- Các hàm liên quan :
- Các ví dụ minh họa :

65. SetWindowRgn (tạo hình dáng cho cửa sổ)

- Thư viện : user32.dll
- Hệ điều hành : Windows NT 3.1 or later; Windows 95 or later
- Khai báo :

```
Declare Function SetWindowRgn Lib "user32" Alias "SetWindowRgn" (ByVal hWnd As Long, ByVal hRgn As Long, ByVal bRedraw As Boolean) As Long
```

- Các tham số
 - **hWnd** : Handle của cửa sổ.
 - **hRgn** : Handle của vùng làm việc. Hàm sẽ thiết lập vùng làm việc (hình dạng) của cửa sổ dựa vào thông số này. Nếu hRgn là Null thì hàm sẽ thiết lập vùng làm việc là NULL.
 - **bRedraw** : nhận vào giá trị Boolean thông báo cho hệ điều hành vẽ lại sau khi thiết lập vùng làm việc. Ví dụ : bạn đặt bRedraw là TRUE nếu như cửa sổ nhìn thấy được (visible).

- *Mô tả* : hàm **SetWindowRgn** dùng để thiết lập miền làm việc (hình dáng) của cửa sổ. Miền làm việc này là phần diện tích bên trong cửa sổ nơi hệ điều hành cho phép vẽ. Hệ điều hành sẽ không hiển thị bất kỳ vùng nào bên ngoài miền làm việc này.
- *Các hàm liên quan* : CreateEllipticRgn
- *Các ví dụ minh họa* :
 - + Ví dụ 1 : Create Font

```
'used with fnWeight
Const FW_DONTCARE = 0
Const FW_THIN = 100
Const FW_EXTRALIGHT = 200
Const FW_LIGHT = 300
Const FW_NORMAL = 400
Const FW_MEDIUM = 500
Const FW_SEMIBOLD = 600
Const FW_BOLD = 700
Const FW_EXTRABOLD = 800
Const FW_HEAVY = 900
Const FW_BLACK = FW_HEAVY
Const FW_DEMIBOLD = FW_SEMIBOLD
Const FW_REGULAR = FW_NORMAL
Const FW_ULTRABOLD = FW_EXTRABOLD
Const FW_ULTRALIGHT = FW_EXTRALIGHT
```

```
'used with fdwCharSet
Const ANSI_CHARSET = 0
Const DEFAULT_CHARSET = 1
Const SYMBOL_CHARSET = 2
Const SHIFTJIS_CHARSET = 128
Const HANGEUL_CHARSET = 129
Const CHINESEBIG5_CHARSET = 136
Const OEM_CHARSET = 255
```

```
'used with fdwOutputPrecision
Const OUT_CHARACTER_PRECIS = 2
Const OUT_DEFAULT_PRECIS = 0
Const OUT_DEVICE_PRECIS = 5
```

```
'used with fdwClipPrecision
Const CLIP_DEFAULT_PRECIS = 0
Const CLIP_CHARACTER_PRECIS = 1
Const CLIP_STROKE_PRECIS = 2
```

```
'used with fdwQuality
Const DEFAULT_QUALITY = 0
Const DRAFT_QUALITY = 1
Const PROOF_QUALITY = 2
```

```
'used with fdwPitchAndFamily
Const DEFAULT_PITCH = 0
Const FIXED_PITCH = 1
Const VARIABLE_PITCH = 2
```

```
'used with SetBkMode
Const OPAQUE = 2
```

```
Const TRANSPARENT = 1

Const LOGPIXELSY = 90
Const COLOR_WINDOW = 5
Const Message = "Hello !"

Private Type RECT
    Left As Long
    Top As Long
    Right As Long
    Bottom As Long
End Type

Private Declare Function GetDeviceCaps Lib "gdi32" (ByVal hdc As Long, ByVal nIndex As Long) As Long
Private Declare Function BitBlt Lib "gdi32" (ByVal hDestDC As Long, ByVal x As Long, ByVal y As Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal hSrcDC As Long, ByVal xSrc As Long, ByVal ySrc As Long, ByVal dwRop As Long) As Long
Private Declare Function CreateCompatibleBitmap Lib "gdi32" (ByVal hdc As Long, ByVal nWidth As Long, ByVal nHeight As Long) As Long
Private Declare Function CreateCompatibleDC Lib "gdi32" (ByVal hdc As Long) As Long
Private Declare Function GetDC Lib "user32" (ByVal hWnd As Long) As Long
Private Declare Function DeleteDC Lib "gdi32" (ByVal hdc As Long) As Long
Private Declare Function DeleteObject Lib "gdi32" (ByVal hObject As Long) As Long
Private Declare Function CreateEllipticRgn Lib "gdi32" (ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long) As Long
Private Declare Function SetWindowRgn Lib "user32" (ByVal hWnd As Long, ByVal hRgn As Long, ByVal bRedraw As Boolean) As Long
Private Declare Function CreateFont Lib "gdi32" Alias "CreateFontA" (ByVal nHeight As Long, ByVal nWidth As Long, ByVal nEscapement As Long, ByVal nOrientation As Long, ByVal fnWeight As Long, ByVal fdwItalic As Boolean, ByVal fdwUnderline As Boolean, ByVal fdwStrikeOut As Boolean, ByVal fdwCharSet As Long, ByVal fdwOutputPrecision As Long, ByVal fdwClipPrecision As Long, ByVal fdwQuality As Long, ByVal fdwPitchAndFamily As Long, ByVal lpszFace As String) As Long
Private Declare Function SelectObject Lib "gdi32" (ByVal hdc As Long, ByVal hObject As Long) As Long
Private Declare Function TextOut Lib "gdi32" Alias "TextOutA" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long, ByVal lpString As String, ByVal nCount As Long) As Long
Private Declare Function MulDiv Lib "kernel32" (ByVal nNumber As Long, ByVal nNumerator As Long, ByVal nDenominator As Long) As Long
Private Declare Function SetBkMode Lib "gdi32" (ByVal hdc As Long, ByVal nBkMode As Long) As Long
Private Declare Function GetSysColorBrush Lib "user32" (ByVal nIndex As Long) As Long
Private Declare Function FillRect Lib "user32" (ByVal hdc As Long, lpRect As RECT, ByVal hBrush As Long) As Long
Private Declare Function SetRect Lib "user32" (lpRect As RECT, ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long) As Long

Dim mDC As Long, mBitmap As Long

Private Sub Form_Click()
    Unload Me
End Sub

Private Sub Form_Load()
    'KPD-Team 1999
```

```

'URL: http://www.allapi.net/
'E-Mail: KPDTTeam@Allapi.net
Dim mRGN As Long, Cnt As Long, mBrush As Long, R As RECT
'Create a device context, compatible with the screen
mDC = CreateCompatibleDC(GetDC(0))
'Create a bitmap, compatible with the screen
mBitmap = CreateCompatibleBitmap(GetDC(0), Me.Width / Screen.TwipsPerPixelX, Me.Height /
Screen.TwipsPerPixelY)
'Select the bitmap into the device context
SelectObject mDC, mBitmap
'Set the bitmap's backmode to transparent
SetBkMode mDC, TRANSPARENT
'Set the rectangles' values
SetRect R, 0, 0, Me.Width / Screen.TwipsPerPixelX, Me.Height / Screen.TwipsPerPixelY
'Fill the rect with the default window-color
FillRect mDC, R, GetSysColorBrush(COLOR_WINDOW)

For Cnt = 0 To 350 Step 30
    'Select the new font into the form's device context and delete the old font
    DeleteObject SelectObject(mDC, CreateMyFont(24, Cnt))
    'Print some text
    TextOut mDC, (Me.Width / Screen.TwipsPerPixelX) / 2, (Me.Height / Screen.TwipsPerPixelY) /
2, Message, Len(Message)
Next Cnt

'Create an elliptical region
mRGN = CreateEllipticRgn(0, 0, Me.Width / Screen.TwipsPerPixelX, Me.Height /
Screen.TwipsPerPixelY)

'Set the window region
SetWindowRgn Me.hWnd, mRGN, True

'delete our elliptical region
DeleteObject mRGN
End Sub

Function CreateMyFont(nSize As Integer, nDegrees As Long) As Long
    'Create a specified font
    CreateMyFont = CreateFont(-MulDiv(nSize, GetDeviceCaps(GetDC(0), LOGPIXELSY), 72), 0,
nDegrees * 10, 0, FW_NORMAL, False, False, False, DEFAULT_CHARSET,
OUT_DEFAULT_PRECIS, CLIP_DEFAULT_PRECIS, PROOF_QUALITY, DEFAULT_PITCH,
"Times New Roman")
End Function

Private Sub Form_Paint()
    'Copy the picture to the form
    BitBlt Me.hdc, 0, 0, Me.Width / Screen.TwipsPerPixelX, Me.Height / Screen.TwipsPerPixelY,
mDC, 0, 0, vbSrcCopy
End Sub

Private Sub Form_Unload(Cancel As Integer)
    'clean up
    DeleteDC mDC
    DeleteObject mBitmap
End Sub

```

+ Ví dụ 2 : Path2Region

```
Private Declare Function BeginPath Lib "gdi32" (ByVal hdc As Long) As Long
Private Declare Function EndPath Lib "gdi32" (ByVal hdc As Long) As Long
Private Declare Function PathToRegion Lib "gdi32" (ByVal hdc As Long) As Long
Private Declare Function TextOut Lib "gdi32" Alias "TextOutA" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long, ByVal lpString As String, ByVal nCount As Long) As Long
Private Declare Function SetWindowRgn Lib "user32" (ByVal hWnd As Long, ByVal hRgn As Long, ByVal bRedraw As Boolean) As Long
Private Declare Function DeleteObject Lib "gdi32" (ByVal hObject As Long) As Long

Private Sub Form_Click()
    'end..
    Unload Me
End Sub

Private Sub Form_Load()
    'KPD-Team 2000
    'URL: http://www.allapi.net/
    'E-Mail: KPDTTeam@Allapi.net
    Dim hRgn As Long
    Const sText = "Click Here!"
    'set the font to 'Times New Roman, size 72'
    Me.FontName = "Times New Roman"
    Me.FontSize = 72
    'set the backcolor to Red
    Me.BackColor = vbRed
    'open a path bracket
    BeginPath Me.hdc
    'draw the text
    TextOut Me.hdc, 0, 0, sText, Len(sText)
    'close the path bracket
    EndPath Me.hdc
    'convert the path to a region
    hRgn = PathToRegion(Me.hdc)
    'set the Window-region
    SetWindowRgn Me.hWnd, hRgn, True
    'destroy our region
    DeleteObject hRgn
End Sub
```

+ Ví dụ 3 : Clipping Regions

```
Private Declare Function GetClipRgn Lib "gdi32" (ByVal hdc As Long, ByVal hRgn As Long) As Long
Private Declare Function IntersectClipRect Lib "gdi32" (ByVal hdc As Long, ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long) As Long
Private Declare Function SelectClipRgn Lib "gdi32" (ByVal hdc As Long, ByVal hRgn As Long) As Long
Private Declare Function OffsetClipRgn Lib "gdi32" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long) As Long
Private Declare Function CreateEllipticRgn Lib "gdi32" (ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long) As Long
```

```
Private Declare Function DeleteObject Lib "gdi32" (ByVal hObject As Long) As Long
Private Declare Function SetWindowRgn Lib "user32" (ByVal hWnd As Long, ByVal hRgn As Long, ByVal bRedraw As Boolean) As Long

Dim hRgn As Long

Private Sub Form_Load()
    'KPD-Team 2000
    'URL: http://www.allapi.net/
    'E-Mail: KPDTTeam@Allapi.net
    Me.ScaleMode = vbPixels
End Sub

Private Sub Form_Paint()
    Form_Resize
End Sub

Private Sub Form_Resize()
    Dim Ret As Long
    'destroy the previous region
    DeleteObject hRgn
    'create an elliptic region
    hRgn = CreateEllipticRgn(0, 0, Me.ScaleWidth, Me.ScaleHeight)
    'select this elliptic region into the form's device context
    SelectClipRgn Me.hdc, hRgn
    'move the clipping region
    OffsetClipRgn Me.hdc, 10, 10
    'generate a new clipping region
    IntersectClipRect Me.hdc, 10, 10, 500, 300
    'clear the form
    Me.Cls
    'draw a Black rectangle over the entire form
    Me.Line (0, 0)-(Me.ScaleWidth, Me.ScaleHeight), vbBlack, BF
    'create a temporary region
    Ret = CreateEllipticRgn(0, 0, 1, 1)
    'copy the current clipping region into the temporary region
    GetClipRgn Me.hdc, Ret
    'set the new window region
    SetWindowRgn Me.hWnd, Ret, True
End Sub

Private Sub Form_Unload(Cancel As Integer)
    'clean up
    DeleteObject hRgn
End Sub

Private Sub Form_Click()
    'unload the form when the user clicks on it
    Unload Me
End Sub
```

TÀI LIỆU THAM KHẢO

- www.allapi.net
- www.caulacbovb.com
- Website Lê Hoàn
- Cẩm nang lập trình Windows API – nhà xuất bản Giao Thông Vận Tải – 2005