

Windows API 的 VB 编程控制应用程序行为

朱从旭 (中南工业大学信息工程学院 410083)

摘要:本文探讨用 VB 控制外部应用程序行为的 Windows API 编程方法,介绍了有关 API 函数的使用技巧并提供示例程序。

关键词: VB Windows 编程 API 函数 控制

一、引言

用程序实现对其他应用程序行为的控制,这在软件开发中是常常必要的。尤其是对于大型应用软件的开发,常常需要将一些其他外来应用软件集成起来,如果这些外来软件缺乏原开发人员的技术支持,要集成起来是比较困难的。对于 Windows 环境下的软件开发,由于 VB 具有其界面友好和快速高效的特点,已成为首选的最有效开发工具之一。在用 VB 开发大型应用软件时,如果能用 VB 编制一个自己的程序实现对其他应用程序运行状态的控制,即控制其他程序的启动、关闭及程序的窗口大小和位置,那么对实现软件集成将带来许多帮助。但由于 VB 是高级语言,控制程序要借助 Windows 操作系统去实现,而 Windows API 正是联系 VB 和 Windows 的桥梁,利用 API 函数可为我们大大拓展 VB 编程功能。因此,本文探讨解决上述问题的 Windows API 编程方法,并提供有关示例程序。

二、控制外部应用程序的启动和关闭

1. 在软件中直接启动应用程序

Windows API 函数 WinExec(lpszCmdLine, fuCmdShow)可以用来控制外部应用程序的启动。参数 lpszCmdLine 是命令行字符串指针,指向一个以 Null 结尾的字符串,这个字符串含有将要运行的应用程序的命令行,若字符串不含路径则 Windows 将在缺省目录中搜索。参数 fuCmdShow 表示启动应用程序的窗口形式,可取一些描述窗体风格的以 SW-开头的常量。(有关 API 函数、常量和结构类型须在 VB 程序中声明,可放在一模块中,本文略)。

例:用 WinExec 控制“记事本”程序的启动,可参见下列事件代码:

```
Private Sub CmdWinExe_Click()
```

```
Dim cmdline As String, x As Long
cmdline = InputBox("请输入应用程序可执行文件名:")
'若在输入框中输入"notepad"
x = WinExec(cmdline, SW_SHOWNORMAL)
'则启动记事本程序
'记事本程序的窗口形式将由 SW_SHOWNORMAL 参数决定。
End Sub
```

2. 在软件中关闭运行中的应用程序

利用 API 函数 FindWindow 和 PostMessage,可实现用软件关闭运行中的外部程序。

函数原型:HWND FindWindow(lpszClass, lpszWindow)

函数原型:BOOL PostMessage(hwnd, uMsg, wParam, lParam)

函数 FindWindow 返回一个窗口的句柄;该窗口的类由参数 lpWindowClass 给出,窗口的名字或标题由参数 lpWindowName 给出。而函数 PostMessage 在指定的窗口消息队列中放置一条消息;参数 hwnd 为指定窗口的句柄;uMsg 为消息号参数,当取常量 WM_CLOSE 时,即为关闭该外部程序;wParam、lParam 分别为消息的字参数和长参数。

例:下列 WinClose 函数先由窗口标题 txt 得到相应的窗口句柄,然后关闭该窗体:

Function WinClose(ByVal txt As String) 'txt 为欲关闭窗口的标题

```
Dim retval As Long, Index As Integer, winHwnd As Long
```

winHwnd = FindWindow(vbNullString, txt) '得到欲关闭窗口的句柄,txt 为窗口标题

```
If winHwnd <> 0 Then
```

```
retval = PostMessage(winHwnd, WM_CLOSE, 0&, 0&) '关闭 winHwnd 标识的窗口
```

```

End If
End Function
Private Sub CmdClos_Click()
    WinClose "计算器" '关闭"计算器"程序的示例
End Sub

```

三、枚举并列表活动程序窗口

利用 API 函数 EnumWindows 可以得到屏幕上存在的所有窗口的名字或窗口标题:

```
BOOL EnumWindows ( lpEnumFunc , lParam )
```

函数通过把每一个窗口的句柄依次传递给 lpEnumFunc 参数指定的回调函数来枚举屏幕上的所有父窗口,子窗口不被枚举。EnumWindows 继续枚举窗体,直到回调函数返回零或枚举完所有的窗体。lpEnumFunc 是回调函数的指针,VB5.0 以上版本提供了一个函数指针运算符 AddressOf,利用它可实现从 VB 到 DLL 函数的回调。

例:以下程序枚举屏幕上所有父窗体并将各窗口的名字或标题列于列表框 List1 中。

```

Private Sub CmdEnum_Click()
    List1.Clear
    x # = EnumWindows ( AddressOf EnumWin-
dowsProc, 0)
    '枚举屏幕上所有父窗体,EnumWindowsProc 为回调函数
End Sub

```

下面是上述回调函数的一个示例程序:

```
Function EnumWindowsProc(ByVal hwnd As Long, ByVal lParam As Long) As Long
```

```
    Dim l As Long
```

```
    Dim TempStr As String * 128 '字符串变量 TempStr 用于存放窗口的标题
```

```
    l = GetWindowText(hwnd, TempStr, 128) 'API 函数 GetWindowText 获取由 hwnd 标识的窗体的窗口标题,并将窗口标题存入字符串变量 TempStr 中
```

```
    If (IsWindowVisible(hwnd)) Then 'API 函数 IsWindowVisible 判断由 hwnd 标识的窗体是否存在于屏幕上。如果指定窗口存在于屏幕上,则将该窗口的标题添加到列表框中
```

```
        Form1.List1.AddItem TempStr
```

```
    End If
```

```
    EnumWindowsProc = True
```

```
End Function
```

四、应用程序的运行状态控制

1. 使某个窗体始终位于所有窗体之上

比如在进行文字输入时,我们总希望输入法窗口恒位于所有窗体之上;又如在编写屏幕取词软件时,也希望实现如此要求。要使一个窗口失去焦点后还能处在其他窗口的前面,我们可以用 API 函数 SetWindowPos() 实现这项功能。SetWindowPos() 函数的使用方法是:

```
SetWindowPos hwnd, hwndInsertAfter, x, y, cx, cy, fuflags
```

其中,参数 hwnd 为窗体句柄(Handle),hwndInsertAfter 为窗体所放置的位置。当我们想要窗体“总在最前”时,可将此参数设为常量 HWND_TOPMOST,当我们想要窗体恢复正常时,应设为常量 HWND_NOTOPMOST。参数 x,y,cx,cy 表示窗体的位置和大小,如果我们并不想在此改变这些参数,就可将 wFlags 参数设为 SWP_NOMOVE 和 SWP_NOSIZE 的组合,以使 SetWindowPos 函数忽略 x,y,cx,cy 参数的设置。

2. 使所有窗口最小化

利用前述方法,首先遍历桌面上的所有窗口;然后判断窗口类型;就可对类型符合的窗口进行最小化(但桌面上的某些窗口,如中文输入法窗口是无法最小化的;而还有一些窗口则是不可见的,这些类型窗口也不能最小化)。这里我们除要用到上述的 EnumWindows 函数来遍历所有窗口外,还要利用 GetWindowLong 函数来获得并判断窗口的类型;最后用 ShowWindow 函数实现窗口最小化。此处将用到的常量参数有 WS_MINIMIZEBOX, WS_VISIBLE, GWL_STYLE 及 SW_MINIMIZE。示例程序如下:

'函数 EnumWindowsProc 是 EnumWindow 的回调函数,EnumWindow 函数将遍历的窗

'口句柄传递到参数 hwnd 中。

```
Function EnumWindowsProc(ByVal hwnd As Long, ByVal lParam As Long) As Long
```

```
    Dim l As Long, ld As Long
```

```
    ld = GetWindowLong(hwnd, GWL_STYLE) '获得并判断窗口的类型
```

```
    If ((ld AND WS_MINIMIZEBOX) = WS_MINIMIZEBOX) Then
```

```
        If ((ld And WS_VISIBLE)) THEN
```

```
            ShowWindow(hwnd, SW_MINIMIZE)
```

```

End If
End If
EnumWindowsProc = True
End Function
Private Sub Command1_Click() '遍历所有窗口的事件过程:
    l = EnumWindows(AddressOf EnumWindowsProc, 0)
End Sub

```

3. 应用程序窗口大小及位置的任意控制

首先从枚举的窗口列表中选择要进行窗口大小和位置控制的窗口标题,再由窗口标题得到窗口的句柄(方法同上)。有了窗口句柄,就为下一步窗口在桌面上的任意放置做好了准备。窗口在桌面上的任意放置,可利用下列函数:

SetWindowPlacement(hwnd, lpwndlp) 该函数将窗口句柄为 hwnd 的窗口根据 lpwndlp 结构中的内容放在桌面上,关于该结构类型的详细说明可参阅文献[2]。下面函数将窗口句柄为 winHwnd 的窗口,按结构 r 决定的位置和大小放置,其中窗口左上角坐标 x、y,窗口宽度 cx、高度 cy 均为 r 的成员。

```
Function SetWindow(winHwnd As Long, x As Long, y As Long, cx As Long, cy As Long)
```

```
    Dim lpwndlp As WINDOWPLACEMENT '定义结构类型变量 lpwndlp
```

```
    Dim poin As POINTAPI : r As RECT '定义结构类型变量 poin 和结构类型变量 r
```

```
    r.Left = x : r.Top = y : r.Right = x + cx : r.Bottom = y + cy '填充结构 r
```

```
    CloseWindow(winHwnd) '调用 API 函数 CloseWindow,先将窗口句柄为 winHwnd 的窗口最小化,以便统一处理。以下填充 WINDOWPLACEMENT 型结构变量 lpwndlp
```

```
    lpwndlp.Length = Len(lpwndlp) '该类型结构变量 lpwndlp 的长度
```

```
    lpwndlp.flags = WPFL_RESTORETOMAXIMIZED
    lpwndlp.showCmd = SW_RESTORE: lpwndlp.ptMinPosition = poin
```

```
    lpwndlp.ptMaxPosition = poin: lpwndlp.rcNormalPosition = r '存放窗口位置坐标
```

```
    SetWindowPlacement winHwnd, lpwndlp '重新放置
```

窗口位置

End Function

Private Sub CmdCtrl_Click() 'VB 窗体的事件过程代码:

```

..... '按前述方法获取窗口句柄 winHwnd
returval = SetWindow(winHwnd, 20, 30, 500, 400) '重置窗体位置和大小

```

End Sub

4. 使窗体透明化

使窗体透明化,就是使窗体上除了控件外,其他部分都是透明的,这样就不挡住后面的窗口。实现此功能要用到 API 函数 SetWindowsLong,该函数原型为:

```
SetWindowsLong(hWnd As Long, nIndex As Long, dwNewLong As Long)
```

同时,还需使用常量 WS_EX_TRANSPARENT 和 GWL_EXSTYLE。

可以编写一个变窗体为透明的通用过程 MakeTransparent(),在需要的地方调用此过程。此通用过程的示例代码如下:

```
Public Sub MakeTransparent(ByVal hWnd As Long) 'hWnd 为窗体句柄
```

```
    Dim RetVal As Long
```

```
    RetVal = SetWindowsLong(hWnd, GWL_EXSTYLE, WS_EX_TRANSPARENT)
```

```
    ' WS_EX_TRANSPARENT 参数使窗体透明
```

```
End Sub
```

以上方法,我们在用多窗口显示技术开发 CAI 教学软件时得到成功的应用。

参考文献

- [1] 王旭等译,Microsoft Windows3.1 程序员参考大全(二)函数,北京:清华大学工出版社,1993 年 12 月。
- [2] 蒋维杜等译,Microsoft Windows3.1 程序员参考大全(三)消息、数据结构和宏,北京:清华大学工出版社,1993 年 12 月。
- [3] 王克己主编,Visual Basic 5.0 中文版用户手册,人民邮电出版社,1998 年 10 月。

(来稿时间:1999 年 7 月)