首先，我对于本文的作者骗人的行为很不爽，从开篇来看是详细分析 API 的文章，但是这部分内容只有 18 面，后边的大量篇幅全是 API 函数大全。很明显，这是一篇拼凑出来的东西，为了刷分用的，所以我特意重新发出来，免费之！不过前面这段文字对于 API 入门确实挺有用，值得看看。如果想深入点学习 API，强烈推荐 VB 学习之 API 教程系列（共七课）。网上有单独的章节，我在另一个文档中已经将七课合并，并且加入补充章节----《逻辑坐标与设备坐标》，免去大家来回找的麻烦。

一、API 是什么？

这个我本来不想说的，不过也许你知道其它人不知道，这里为了照顾一下新手，不得不说些废话，请大家谅解。

Win32 API 即为 Microsoft 32 位平台的应用程序编程接口（Application Programming Interface）。所有在Win32 平台上运行的应用程序都可以调用这些函数。

使用 Win32 API，应用程序可以充分挖掘 Windows 的 32 位操作系统的潜力。Mircrosoft 的所有 32 位平台都支持统一的 API，包括函数、结构、消息、宏及接口。使用 Win32 API 不但可以开发出在各种平台上都能成功运行的应用程序，而且也可以充分利用每个平台特有的功能和属性。

以上为 API 的相关介绍，不过有些新手看了以后可能还是不怎么明白 API 到底有什么用？这里请不要着急，如果你有足够耐心的话，请慢慢往下看。

二、如何使用 API？

估计这才是大家真正关心的，那么如何使用 API 呢？在了解 API 之前，先打开你的 VB 书，翻到过程函数这章来，在搞清楚 API 之前应该先搞懂过程函数是怎么一回事！如果你还不知道过程的工作方式，那么请先不要急着往下看，那样容易走很多弯路。

好了，当你理解了过程函数时，也就是你可以使用 API 的时候了，别把 API 看得太难，你就像使用过程函数一样使用 API 就可以了。首先，让我们看看一个简单的 API，以下：

```
Private Declare Sub Sleep Lib "kernel32" Alias "Sleep" (ByVal dwMilliseconds As Long)
```

以上这个 API 的呢是起一个延时作用。你如果是刚接触 API 的话可能会感到 API 的书写及其复杂，而且会感到很不适应。其实这没什么的，慢慢习惯就好了。至于 API 这些复杂的书写你就不用操心了，在你安装 VB 的时候微软已经帮我们带上了 API 浏览器，这些全部都可以利用 API 浏览器帮我们自动生成。API 浏览器的位置位于[开始菜单－程序－Microsoft Visual Basic 6.0 中文版－Microsoft Visual Basic 6.0 中文版工具－API 文本浏览器]。打开 API 浏览器，在最上面的一个文本框中输入 Sleep，这时下面列表框中就会自动显示相应的 API 函数，然后点右边添加按钮即可，接着点击复制按钮，这时你就可以用 Ctrl+V 把声明的 API 添加到 VB 代码窗口中了。

这里我要说一下，有些新手可能还弄不明白。API 的声明范围一般有两种模式，一种是 Private(私有的),一种是 Public(公用的)。一般 Private 是声明在类模块或窗体类中，Public 声明在模块中。你在添加 API 的时候，添加按钮下面就有 API 的声明范围，可以根据自己的需要进行添加。这里我们一般选择私有的(Private)就可以了。

经过上面，我们知道如何添加 API，接着我们分析一下 API 声明，这是你了解 API 必备的。首先看第一个单词 Private，很显然，我上面刚刚讲过，这是申明一个私有的 API 变量。再看第二个 Declare，这个单词帮我们告诉 VB 是在申明 API 函数，一般申明外在的 API 函数时都必须带上这个单词。第三个 Sub，别告诉我你不知道什么意思？这就是我叫你先学

习 VB 中过程函数的意思，这个说白了就是没有反回值，一般如果不是 Sub 而是 Function 都带有反回值的。第四个 Lib，这个是告诉 VB 我们要声明哪一个 DLL 中的 API 函数，也就是告诉 VB 我们要申明第五个单词 kernel32.dLL 中的 API，一般写 DLL 名称时都要用双引号括起来，如"user32"、"shell32.dll"等，至于后面的.dll 这个可以带可不带。再来看第六个 Alias，这个也是需要同后面一个一起用的，我们应该把第六个和第七个连起来一起看 Alias "Sleep"，这个意思表示将被调用的过程在 DLL 中还有另外的名称，这个是可选的。最后括号里面的，也就是和过程函数一样，你传入相应的值就可以了。

上面我们分析完 API 函数声明以后，接着我们就要自己动手写代码了。先把这个 API 复制到 Form1 代码窗口中，然后写如下代码：

```
Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)
Private Sub Form_Load()
    Sleep 2000
End Sub
```

解释一下，也就是在窗体启动时使用 Sleep API 进行延时 2 秒，后面的参数 dwMilliseconds 是表示你要延时的秒数，基本上和设置 Timer 中的秒数一样。你再看一下 Sleep 2000 的使用方式，是不是和使用 VB 过程函数一样呢？好了，我们的第一个 VB API 程序写完了，可以看到使用 API 并不是一件很难的事。

三、如何才能提升你对 API 的学习兴趣？

API，我常把它看做成过程函数，不过每人都有每人的见解和理解方式，自己的理解方式只要可以帮助自己更好的学习和掌握 API，也没必要一定要学习他人的。

1,自己做 MsgBox

了解 API 参数的使用方法是很重要的，这里我们不用 VB 的 MsgBox，直接使用 API 弹出 MsgBox 消息框。首先，打开 API 浏览器，选择 MessageBox，大家可以用这个 API 和 VB 内置的 MsgBox 比较一下，其实 MsgBox 也就是 MessageBox 的缩写，只不过一个是 API，一个是 VB 内置的，但两者都是通过 API 进行工作的。好了，选择私有声明方式，粘贴到 VB 代码编辑窗口中，然后新建一个 CommandButton，写入以下代码：

```
Private Declare Function MessageBox Lib "user32" Alias "MessageBoxA"
(ByVal hwnd As Long, ByVal lpText As String, ByVal lpCaption As String,
ByVal wType As Long) As Long
Private Sub Command1_Click()
    MessageBox Me.hwnd, "这里是内容", "标题", 0
End Sub
```

先让我们来分析一下，首先看第一个参数 Byval hWnd As Long，很显然这是一个长整形变量，所以我们这里需要传递的是数字，你可能会发现我们传递的并不是数字啊，而是 Me.hwnd？？很奇怪是吗？如果你真的有此疑问说明你是真心想要学习好 API 的，现在就让

我们来看看 Me.hwnd 到底是什么东西？以下摘自 VB 帮助文档：

hWnd 属性：返回窗体或控件的句柄。

句 柄：是由操作环境定义的一个唯一的整数值，它被程序用来标识或者切换到对象，如窗体或控件等。

现在估计你差不多就已经明白了，我们调用的 hwnd 其实是一个句柄整数值，你可以用 Msgbox Me.hwnd 看一下就知道了。至于 Me 这是一个关键字，代表当前 Form 窗体对象。如：Me.Caption="标题"、Me.BackColor=vbRed 等。

（“print Me.hwnd”输出 不关掉窗体每次相同，关掉重新运行后不同）

接上面的，首先我们传入了 Me.hwnd，表示是当前窗口调用 MessageBox，这里告诉大家一个技巧，也就是以后凡是看到 Byval hwnd As Long，一般都是需要传入句柄的，至于传入哪个对象句柄，那就要看你是怎么实现的了。

ByVal lpText As String，这个是字符串变量，标识着叫我们需要传入字符串进去，可以看里面的变量字符 lpText,属于文本的意思，也就是说是用来显示 MsgBox 中的消息文本的。

ByVal lpCaption As String，也是字符串变量，还是传入字符串进去。在看里面的变量字符 lpCaption，其实就是显示 MsgBox 标题的。

ByVal wType As Long，这是一个整形变量，需要传递整形数字，还是看里面的变量字符 wType，标识着显示 MsgBox 类型，这里可以像 VB 的 MsgBox 一样使用，如这里可以传入：vbYesNo,vbOkCancel 等，如果忽略那就传入 0 即可。

好了，按 F5 启动程序，点击 Command1，接着就会弹出一个消息框，这里我们制作以及分析 MsgBox 已经完成了。希望你能在这段学习到一些知识。

2,来点实用的吧

就拿隐藏 Windows 任务管理器来说吧，这里只能隐藏任务管理器中的窗口，不能隐藏进程。(问：有没有隐藏进程的?答：你想干什么？)，当程序运行后你无法从任务管理器的窗口中关闭程序，只能从进程中进行终止。好了，还是老规矩，打开 API 浏览器，输入 GetWindow 和 ShowWindow 两个 API,声明范围还是私有的，复制粘贴到 Form 代码窗口中，嗯，好了？别急，还是 API 浏览器，选择 Combox 中的常数，输入 GW_OWNER 和 SW_HIDE 这两个 API 常数，然后粘贴到代码窗口中，问我这两个是干什么的？那就接着往下看吧。写入以下代码：

```
Private Declare Function GetWindow Lib "user32" (ByVal hwnd As Long,
ByVal wCmd As Long) As Long
Private Declare Function ShowWindow Lib "user32" (ByVal hwnd As Long,
ByVal nCmdShow As Long) As Long
Private Const GW_OWNER = 4
Private Const SW_HIDE = 0
Private Sub Form_Load()
    Dim lphWnd As Long
    lphWnd = GetWindow(Me.hwnd, GW_OWNER)
    ShowWindow lphWnd, SW_HIDE
End Sub
```

又到了分析的时候了，这对刚入门的新手可谓是最激动的时候了。好了，还是老子，看

看两个 API 的表面意思和传递值变量。

先看 GetWindow,表面意思：获取窗口。传递值变量：hWnd 整形句柄，wCmd 整形命令值。

再看 ShowWindow，表面意思：显示窗口。传递值变量：hWnd 整形句柄，nCmdShow 整形命令值。

然后是使用代码，先看 lphWnd = GetWindow(Me.hwnd, GW_OWNER)这句，这句意思是获取当前窗口的所有者窗口句柄（“print Me.hwnd”输出 不关掉窗体每次相同，关掉重新运行后不同）。。。（“print lphwnd”输出 不关掉窗体每次相同，关掉重新运行每次输出也相同），可以看到 GetWindow 是 Function 过程函数，执行以后会返回相应的窗口句柄值，这个值为 Long 整形(同句柄)。接着调用 ShowWindow lphWnd, SW_HIDE，这句意思是显示 lphwnd 这个句柄的窗口，关键一句是最后的 SW_HIDE,这是 API 函数的常量。通过设置常量能让系统知道 API 到底应该怎么执行显示窗口，是显示？还是隐藏？Hide 当然是隐藏的意思。好了，编译成 Exe，运行后打开任务管理器，查看程序窗口，还有吗？

我又要说一下了，有些人可能不懂为什么要用 GW_OWNER 这些常量，这些到底有什么用？还有就是我怎么知道哪些 API 对应哪些的常量？其实这些常量你只要稍微注意一下就知道它们是怎么回事了，如在 GetWindow 中我使用 GW_OWNER，在 ShowWindow 中我使用 SW_HIDE 这些常量都有一个共同的特点，就是他们都是以 API 的单词第一个字母为标准。如 GetWindow 相对应的常量就是 Get(G)Window(W)=GW，ShowWindow 相对应的常就是 Show(S)Window(W)=SW，这些常量可以自己在 VB 的 API 浏览器中找找看。（讲了这么多还是没讲常量到底有什么作用呵，GW_OWNER 寻找窗口所有者，SW_HIDE 隐藏窗口，活动状态给另一个窗口）

3.继续往下学吧。。

上面两个我们讲到了一般 API 的使用方法，和一些 API 常量的使用方法，接着我们来看看 API 类型的使用方法，在了解这一小节前请先搞懂 VB 中的自定义类型(Type)这章，否则你可能会稀里糊涂的，到时别怪我没提醒你哦！

这次让我们来获取一下鼠标指针的位置。这里教大家一个技巧，当你想用 API 去实现某一特定的功能时，却又不知道该用哪个 API，这时你可以就表面的意思到 API 浏览器找找，有 70%以上的机率可以找到哦！现在就拿这个 API 开刀，那我们应该如何找？别着急，往下看：

如我们现在要获取鼠标指针位置，可以这样翻译一下：Get（获取）Cursor（指针）Pos（位置），组合起来：GetCursorPos，呵呵，一条 API 就这样出来了，到 API 浏览器输入这个组合单词，呵，有吧？见以下：

```
Private Declare Function GetCursorPos Lib "user32" Alias "GetCursorPos" (lpPoint As POINTAPI) As Long
```

好了好了，高兴一下就算了，现在让我们分析一下这个 API，看其它的没啥不同的，其中只有一个参数，就是最后一个变量有些不懂？在 VB 中好像没有见过这个变量？不明白么？那就再继续往下看。

lpPoint As POINTAPI，POINTAPI？很显然，在 VB 中并没有此类型，一般都是 String、Integer、Long、Byte 等变量类型，那么这个也就理所当然的是自定义类型(问：什么是自定义类型？答：不知道,自己不会看书啊)。既然是自定义类型，那么我们如何才能知道它是如

何定义的呢？这里也就不用你操心啦，还是 API 浏览器，在最上面的 Combox 中选择类型，这时下面 List 中也就自然的把 API 的相关类型显示出来了，现在我们开始在 Text 文本框中输入我们需要的自定义类型，POINTAPI，点击添加，出来了吧？如下：

```
Private Type POINTAPI
        x As Long
        y As Long
End Type
```

好了，现在开始写代码，添加一个 Timer 控件，设置属性见以下：

Interval = 100

Enabled = True

双击 Timer 控件，转到代码环境中写入以下代码：

```
Private Declare Function GetCursorPos Lib "user32" (lpPoint As POINTAPI) As Long
Private Type POINTAPI
        x As Long
        y As Long
End Type
Private Sub Timer1_Timer()
    Dim lpPoint As POINTAPI

    GetCursorPos lpPoint

    Me.Caption = "X = " & lpPoint.x & " Y = " & lpPoint.y
End Sub
```

好了，分析开始，紧张不？别紧张，没啥值得紧张的！见以下：

Dim lpPoint As POINTAPI，申明一个 POINTAPI 类型变量，我们学过自定义类型的朋友都知道，一般使用自定义类型时都需要先申明一个相关的类型变量方可使用。

GetCursorPos lpPoint，这一步我不说你都知道，调用 API 呗。通过这个 API 获取鼠标指针的相关信息。这里我们使用了自己声明的 lpPoint 变量，那为啥要使用这个变量呢？这里我们回过头来就前两节我们所分析的那样进行分析，可以看到 GetCursorPos 所需要传递的值，如果是 Long，我们就传入整形数字，如果是 String，我们就传入字符串，这里是 POINTAPI，所以理所当然是要传入 POINTAPI 类型，但是！VB 中的自定义类型不可以直接使用，所以我们需要先声明一个相同类型的变量。不知道说了这么多你懂了没？

Me.Caption = "X = " & lpPoint.x & " Y = " & lpPoint.y，最后一句，也就是用来显示当前鼠标的坐标值的，我们通过声明的 lpPoint 变量来获取相应的鼠标坐标值，如果你不懂，那就请你先把 VB 自定义类型这章学完再说。

这里关于 API 的一些使用方法及范例就先介绍到这里，如果你还有耐心往下看下去的话，那我们就接着往下聊！

四、如何慢慢提升自己的 API 功力？

何为 API 功力？其实没必要搞那么清楚，首先需要搞明白的就是，你应该知道在什么

环境下使用什么 API，实现哪些功能应该使用哪些 API！这才是我们需要的。

1,试着自己从小程序开始写起。

写小程序？对！在你写小程序时应该拣你最感兴趣的程序写，否则有可能你写到一半以后会觉得自己这个程序写得毫无价值，简直是在浪费时间，最后到头来还是功亏一篑。这里我拿什么当题材呢？我在这里也想了很久，最后还是决定选择一个注销 Windows 程序来做题材(其实这是我当初学 API 最想实现的功能)。

注销 Windows 也就是退出 Windows(重启，关机等都一样，不都是退出的意思吗？)，根据表面意思在 API 浏览器中输入 Exit(退出)Windows，看看有没有这个 API？这里提醒一下，你在查找这个 API 的时候还会看到 ExitWindowsEX 这个 API，其实这两个 API 实现的功能一样，前者是用在 16 位操作系统上，只不过在 Win32 位操作系统上一般都使用 ExitWindowsEX。所以这里就使用后者。API 见以下：

```
Private Declare Function ExitWindowsEx Lib "user32" Alias "ExitWindowsEx" (ByVal uFlags As Long, ByVal dwReserved As Long) As Long
```

看看里面的两个参数，ByVal uFlags As Long？这里我们需要传入一个整形数字，可是应该传入什么数字呢？这里说下，API 中的参数可以传入不同的值，不同的值从而导致产生不同的结果。分析 API 中参数应该传递哪些值其实是有技巧的，以后大家只要是看到参数字符中包函 Flags 字符的话那就说明该参数可以被传入一个或多个标志，并且大部分都是传入 API 常数(什么是 API 常数就不用我说了吧)。说白话点，就是我们可以传入多个 API 常量，并且可以在 API 浏览器中找到，当然，不一定所有的 API 常数都可以在 API 浏览器中找到，不过大部分都可以。

在 API 浏览器查 API 常量时我前面就教过大家技巧，现在该是我们实践的时候了，分析如下：

```
Exit   ：头一个大写字符 E
Windows：头一个大写字符 W
Ex     ：头一个大写字符 E
组合   ：EWE_
```

好了，现在在 API 浏览器的中常数中找找，咦？发现好像没有以 EWE 开头的常数？？只发现以 EWX 开头的？现在先别着急，咱们回过头来再分析下，咱们是失败在最后一步 Ex 上，这里我不得不否决我前面教过大家的技巧，但是又不能完全否决，出现这种情况时就需要大家灵活运用 API 常数的分析法，可以看到 EWX 最后一个 X 是以 Ex 的 X 作结尾的，以这种方法做 API 常数开头的不止这一个，所以这里我特意留了一个陷阱，希望给大家带来一些经验将来能够灵活运用。现在我把关机 uFlags 所能用到的相关常数发上来，如下：

```
Private Const EWX_FORCE = 4
Private Const EWX_LOGOFF = 0
Private Const EWX_REBOOT = 2
Private Const EWX_SHUTDOWN = 1
```

怎么样？看得懂吧？英语稍微好一点基本上没问题。不过这里我还是要解释一番，照顾

新手嘛！

EWX_FORCE 前面的 EWX_ 我就不说了，关键是看 _ 符号后面的,Force 单词翻译：强制，强迫。人工在翻译一下(我英文不好，翻译错了请别见怪，呵呵 ^_^)，意思是说：强制执行 ExitWindowsEx API 关机函数。不知道这样解释你能不能明白。那到底这个常数有什么用呢？这里我们先回忆一下以前关机的时候，当 Windows 无法关闭某些窗口的时候就停止继续关机了，最后还得把无法关闭的窗口手动关闭方可，现在，如果我们使用这个常数进行关机，那 Windows 不管你窗口能不能关闭，直接强制关闭。希望你懂了。

EWX_LOGOFF 这个嘛，貌似组合单词，不可直接翻译，那样就不是那个意思了。Logout Off，是这样写吗？注销的意思。

EWX_REBOOT 不浪费时间了，直接说明意思：重新启动。

EWX_SHUTDOWN 关机。

至于第二个 ByVal dwReserved As Long，为保留整形，一般为 0 即可。至于为什么为 0,大家可以到网上下载一些专门讲解 API 函数的电子书看看，里面有大部分 API 函数的详细讲解。或者下载 VS.MSDN 看看，在 MSDN 中说 Windows 2000/95/98/Me 中此参数忽略，XP 中是指定关机消息说明。

最后看看这个 API 为 Function 声明，说明该函数有返回值，返回值为 Long，MSDN 中说：如果执行成功，则返回非零，否则为零。

现在上面已经把这个关机 API 和相关参数常量都给你分析透了，你可别告诉我你还不知道怎么写？好了，这里我们做一个定时注销程序，呵呵，虽然很简单，不过很多时候用得上哦！在 Form 窗口上添加 Timer 控件，Interval 设置为 1000，Enabled 设置为 True。好了，代码如下：

```
Private Declare Function ExitWindowsEx Lib "user32" (ByVal uFlags As Long, ByVal dwReserved As Long) As Long
Private Const EWX_LOGOFF = 0
Private Sub Timer1_Timer()
    Static i As Integer

    i = i + 1

    Me.Caption = i '这一步纯粹是想看看当前已经执行到几秒了？可不要

    If i = 10 Then '秒数判断，可以根据自己的需要进行运算
        ExitWindowsEx EWX_LOGOFF, 0
    End If
End Sub
```

其实我都有点不想分析了，不过为了照顾大众，不得不说下，Static i As Integer 静态变量(问：啥叫静态变量？答：我晕！)，i = i + 1 是每执行 Timer 一次 i 就加 1,Timer 的 interval 设置为 1000，1000 为一秒，2000 为二秒。。。。后面一个 If i = 10 Then 是判断当 i=10 以后，也就是 10 秒，就执行注销，这个时候你可别忘了保存好你的其它没有保存的文件哦，如果没保存资料丢失的话偶不承担任何法律责任的。其实这里我们可以自己做一个，如可以写成这样：ExitWindowsEx EWX_LOGOFF Or EWX_FORCE, 0，其中用了 Or 运算，整体的意思是强制 Windows 注销。这样理解就够了，只要能让你明白。

现在我又要说一下了，不说不行的！就是在 API 中使用 Or 运算，关于 Or 运算符 VB 书中都有详细解释的，别告诉我你没看？没看马上去看！上面 EWX_LOGOFF Or EWX_FORCE 的使用是把 注销 和 强制 进行 Or 位运算，对两个数值执行按位析取，这里涉及到二进制运算，说多了你可能不明白(如果你还是想追根到底的想知道到底是怎么一回事的话，我也没办法，给个网址你慢慢看 http://book.csdn.net/bookfiles/110/1001103366.shtml)，我就说简单点的吧，以后如果你想组合两个 API 常数的功能，一般都是用 Or 进行运行的。如上面写的。

好了，保存其它文件，然后 F5 运行之，看着 Form 标题的数字慢慢添加，当为 10 时，Windows 开始注销。。。

小提示：在使用 EWX_SHUTDOWN 的时候你可能会感觉没有作用，主要是 NT 系统的安全性提高，需要用其它 API 进行提升自己的权限才可以。关于如何提升应用程序权限请百度一下。

2,先从一些最简单的 API 开始

无疑自己试着写程序是最好的提升方法，学完一些知识以后自己试着写写，这样能让你理解的更快更好，好了不说废话了，接着往下看。

最简单的 API，呵呵，哪些最简单呢？这个我也说不好，这样吧，咱们就从 Get(获取)开始，那 Get 什么呢？Window(窗口)，还是从窗口下手吧，这样更接近我们日常的编程，谁叫这是一个 Windows 操作系统呢？先列几个常用的 API：

GetWindow 、 GetWindowDC 、 GetWindowLong 、 GetWindowRect 、 GetWindowRgn 、 GetWindowsDirectory、GetWindowText、GetWindowThreadProcessId

还有很多，我就先列举几个简单点的，咱们就从这几个中间随便抽几个来讲讲吧。

先从 GetWindowText 下手，大家就表面的意思进行理解下，Get(获取)Window(窗口)Text(文本)，Very Good!这个 API 以前不错的，可以获取密码框中的密码，呵呵，说到这里，我估计有些人开始兴奋起来了！那好，Follow Me!

新建一个 Form 窗口，然后添加一个 CommandButton，Caption 设置为:显示密码。接着

添加两个 TextBox，Text1 属性设置：PasswordChar=*;Text=123456789，Text2 的属性基本上没有什么需要设置的，只需要把 Text 属性为空就可以了，它主要是用来帮助咱们显示出密码的。好了，在 Form1 代码框中填入以下代码：

```
Private Declare Function GetWindowText Lib "user32" Alias "GetWindowTextA" (ByVal hwnd As Long, ByVal lpString As String, ByVal cch As Long) As Long
Private Sub Command1_Click()
    Dim sBuffer As String

    sBuffer = Space(255)
    GetWindowText Text1.hwnd, sBuffer, 255

    Text2.Text = sBuffer
End Sub
```

OK，F5 运行，点击 Command1，怎么样？Text1 中的密码字符显示在 Text2 中了吧？你可以再更改下 Text1 中的密码，然后再点击 Command1 试试。也许你觉得会多此一举，为何不 Text2.Text=Text1.Text 这样？如果真的这样的话看似简单，那你就学不到 API 了。

又到了开始分析的时候了，打起精神来，先看第一句：Dim sBuffer As String，不用说，声明一个字符串变量呗！接着看第二句：sBuffer = Space(255) 那这一句呢？有些人可能不知道了，没事，我会仔细讲的。Space 是 VB 内置的字符串处理函数，VB 中的帮助文件中有说明：

```
开始 {

本示例使用 Space 函数来生成一个字符串，字符串的内容为空格，长度为指定的长度。
Dim MyString
' 返回 10 个空格的字符串。
MyString = Space(10)
' 将 10 个空格插入两个字符串中间。
MyString = "Hello" & Space(10) & "World"

} 结束
```

很显然，我这一句是要分配 255 个空格字符串内存，为啥要用分配？这都是为后面所要用到打定的基础。接着往下：

GetWindowText Text1.hwnd, sBuffer, 255 这一步是关键，通过它来获取咱们想要的窗口文本，看第一个参数，我前面讲过 hwnd 一般都是需要传入句柄的，这时咱们传入了 Text1.hwnd(Text1 控件的句柄)，第二个参数，lpSting 为字符串变量，所以这里咱们传入 sBuffer 字符串变量。最后一个 cch 为 Long 整形，所以理应传入数字，这里我们传入了 255。现在又有人想问了，为什么需要这么传入值？貌似和以前的传入不一样？确实！一刚开始你可能搞不懂，这时候我先讲讲大概的意思，我们用 GetWindowText 来获取窗口中的文本，当获

取成功以后，理所当然会返回窗口中的字符串，但是当我们用这个 API 进行获取时，必须需要一个缓冲来保存我们所获取的字符串，你如果不信去试试把 sBuffer = Space(255)去掉，后面的 255 其实就是告诉这个 API 我们缓冲字符串的大小，这里再告诉大家一个技巧，以后只要是看见包函有 cch 字符时，大部分都是输入相关类型的大小。

再附加一点，就里我说过，hwnd 是用来传句柄的，你也可以传入其它窗口句柄，只要其它窗口有文本，都是可以通过这个 API 获取的。还有 Text2.Text = sBuffer 其实是可以先把 sBuffer 处理一下再传给 Text2.Text 的，关于字符串处理这里不讲。

好了，分析结束，来个小提示：在 Windows 操作系统中，任何有句柄的东东都可被看作为一个窗口。另外你可能会去试试 QQ 的密码框，^_^ 这里我要告诉你一下，无法成功，为什么无法成功呢？这是一个技术问题目前不提！

接着再来试试 GetWindowsDirectory，大家看表面意思吧！Get(获取)Windows(就是 Windows 目录)Directory(目录)，也就是获取咱们那个系统目录，如：C:\Windows。可能我的 Windows 目录中在 C 盘，而其它人的可能在 D 盘、E 盘也说不定，所以有的时候软件需要这个 API 进行获取操作系统具体的 Windows 目录。

好了，还是新建一个标准 EXE，添加一个 CommandButton，属性 Caption=显示 Windows 目录，OK，写入以下代码：

```
Private Declare Function GetWindowsDirectory Lib "kernel32" Alias "GetWindowsDirectoryA" (ByVal lpBuffer As String, ByVal nSize As Long) As Long
Private Sub Command1_Click()
    Dim sBuffer As String

    sBuffer = Space(255)
    GetWindowsDirectory sBuffer, 255
    MsgBox "Windows 目录在：  " & sBuffer
End Sub
```

分析！第一个 Dim sBuffer As String 字符串变量，sBuffer = Space(255)缓冲字符串，GetWindowsDirectory sBuffer, 255 这个和上面所讲的一样，最后一个参数 nSize 为 Long 整形，所以传入数值，那传入什么数值呢？Size？？？当然是缓冲字符串大小了，以后遇到这个 nSize 一般也是传入相关类型的大小的。MsgBox "Windows 目录在： " & sBuffer，是用 MsgBox 消息框显示出 Windows 目录的位置。

OK，恭喜你，你又会使用了一个 API，还要继续吗？(问：当然还要啦！答：最后一次哦！)

GetWindowThreadProcessId，这次玩玩窗口进程，我估计有些人只要看见与进程有关的东东也会变得兴奋，呵呵！好了，先看看这个 API 是什么样的？如下：

```
Private Declare Function GetWindowThreadProcessId Lib "user32" Alias
```

"GetWindowThreadProcessId" (ByVal hwnd As Long, lpdwProcessId As Long) As Long

看表面意思：Get(获取)Window(窗口)Thread(线程)Process(程序)Id(ID)，组合：获取当前线程的窗口进程 ID。至于进程 ID 要着有什么用，自己以后深入 32 编程就知道了。

看看参数，ByVal hwnd As Long，哈哈，熟悉吧，一个 hWnd 句柄。lpdwProcessId As Long 这个就是咱们需要的进程 ID,老规矩，新建标准 EXE，添加一个 CommandButton,属性：Caption=获取窗口进程 ID。代码如下：

```
Private Declare Function GetWindowThreadProcessId Lib "user32" (ByVal hwnd As Long, lpdwProcessId As Long) As Long
Private Sub Command1_Click()
    Dim PID As Long

    GetWindowThreadProcessId Me.hwnd, PID

    MsgBox "窗口进程的 ID 是：" & PID
End Sub
```

我已经习惯了给大家分析了。首先看看第一个参数，ByVal hwnd As Long，又是句柄来的(问：废话！答：教会了你也别这样啊)，lpdwProcessId As Long，这个就要注意了，看看这个参数的传递方式，是以 ByRef 进行传递的(问：呵呵，不懂什么意思？答：不懂？转回去看过程函数这章)，也就是说 ByRef 是以地址进行传递的，过程中可以改变传递的参数值。明白了吗？还不明白的话回去乖乖看书吧！现在明白了传递方式，也就是说我们声明的 PID 是用来获取窗口进程 ID 的，厉害啊。

F5，运行之，点击 Command1，PID 出来了吧？没出来我马上从十楼跳下去。

温馨小提示^_^：hWnd 可以传入其它窗口句柄，同样可以获取其它窗口进程 ID。

接下来我们再来看看 Set(设置)，Set 什么呢？当然还是 Window(窗口)容易些，先列出几个常用的 API：

SetWindowLong、SetWindowPos、SetWindowRgn、SetWindowText

接上面的。

首先咱们先看 SetWindowText，咱们在上面讲过 GetWindowText 这个 API，GetWindowText 是用来获取窗口文本的，而这个正好相反。现在可以看看表面意思 Set(设置)Window(窗口)Text(文本)，好了这样理解就够了，我们已经知道这个 API 是设置窗口文本的，接着咱们就到 API 浏览器中找找这个 API，如下：

Private Declare Function SetWindowText Lib "user32" Alias "SetWindowTextA" (ByVal hwnd As Long, ByVal lpString As String) As Long

接着咱们看里面所需要传递的参数，一共有两，第一个 ByVal hwnd As Long 我就不用说了，传入句柄呗，第二个 ByVal lpString As String，其中声明的 lpString 是字符串变量，可想而知，这里需要传入字符串，好了，开始实践。新建一个标准 EXE，然后添加一个 TextBox

控件，然后再添加一个 CommandButton,写入以下代码：

```
Private Declare Function SetWindowText Lib "user32" Alias "SetWindowTextA" (ByVal hwnd As Long, ByVal lpString As String) As Long
Private Sub Command1_Click()
   SetWindowText Text1.hwnd, "这是咱们设置的文本"
End Sub
```

呵呵，这个看似比前面的更简单，不过我还是要罗嗦一下，首先把 Text1 的句柄传入第一个参数，这样 API 知道咱们需要操作哪个窗口，第二个是一个字符串变量，所以这里就是我们需要传入的文本。好了，F5 运行，点击 Command1，OK。

再看 SetWindowPos，可以说这个 API 可以看成设置窗口位置，但是最终的实现效果取决于咱们传递的参数，好了，在 API 浏览器中找到这个 API，如下：

```
Private 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
```

呵！好家伙，这个 API 看起来有些复杂啊？不过别担心，有我在嘛，我会帮你好好分析的，这里还请大家别光我一个人分析，必须把自己融入进来，咱们一起分析这样不更有趣？好了，废话少说，先看第一个参数：

ByVal hwnd As Long　这里我就不讲了，传入窗口句柄

ByVal hWndInsertAfter As Long 好了，看看这个！hwndInstrAfter，可以看到里面包函有 hwnd 字符，这时你可能会说我前面不是已经说过嘛，只要看见包函有 hwnd 字符的都应该传入句柄嘛？呵呵，没错，你很聪明，记得我说的话呢！在这里夸一下你，别骄傲啊！现在咱们好好分析一下这个地方应该传入哪些参数！打开 MSDN，不好意思是英文，这里我就把翻译过来的说明放上来，如下：

hWndInsertAfter - Long，窗口句柄。在窗口列表中，窗口 hwnd 会置于这个窗口句柄的后面。也可能选用下述值之一：

HWND_BOTTOM　将窗口置于窗口列表底部

HWND_TOP　将窗口置于 Z 序列的顶部；Z 序列代表在分级结构中，窗口针对一个给定级别的窗口显示的顺序

HWND_TOPMOST　将窗口置于列表顶部，并位于任何最顶部窗口的前面

HWND_NOTOPMOST　将窗口置于列表顶部，并位于任何最顶部窗口的后面

可以看到这个地方有四个参数供我们选择，一般我们会使用第三个 API 常数和第四个 API 常数，这几个 API 常数都可以在 API 浏览器中找到，至于具体实现什么功能我相信大

家都知道吧，后面有写呢！

再看看后面的几个 x,y,cx,cy 分别为 Long 变量，我上面讲过，SetWindowPos 可以看成设置窗口位置嘛，所以这里理所当然是传入相关的坐标值，如果忽略则为 0,自己可以试下。

ByVal wFlags As Long,这个参数，我又说过，看看字符 Flags，呵呵，熟悉吧，所以这里咱们需要传入相关的标识常数，利用咱们以前学过的常数分析法进行分析，Set(S)Window(W)Pos(P)=SWP_ ，可以看到相关的常数了吧？这里我把相关常数的说明发上来大家看下，如下：

SWP_DRAWFRAME 围绕窗口画一个框

SWP_HIDEWINDOW 隐藏窗口

SWP_NOACTIVATE 不激活窗口

SWP_NOMOVE 保持当前位置（x 和 y 设定将被忽略）

SWP_NOREDRAW 窗口不自动重画

SWP_NOSIZE 保持当前大小（cx 和 cy 会被忽略）

SWP_NOZORDER 保持窗口在列表的当前位置（hWndInsertAfter 将被忽略）

SWP_SHOWWINDOW 显示窗口

SWP_FRAMECHANGED 强迫一条 WM_NCCALCSIZE 消息进入窗口，即使窗口的大小没有改变

所以我说过，一个这样的 API 他具体实现的功能取决于你所传递的参数。假设这里咱们需要实现一个窗口永远置前的功能，首先新建一个标准 EXE，输入以下代码：

```
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 Const HWND_TOPMOST = -1
  Private Const SWP_NOMOVE = &H2
  Private Const SWP_NOSIZE = &H1
  Private Sub Form_Load()
      SetWindowPos Me.hwnd, HWND_TOPMOST, 0, 0, 0, 0, SWP_NOMOVE Or SWP_NOSIZE
  End Sub
```

现在咱们开始分析，第一个参数传入句柄，第二个我上面讲过，实现什么功能传入什么参数，这里咱们是实现的窗口永久置前的功能，所以传入 HWND_TOPMOST 常数，现在看看其实坐标，如果你不想改变窗口的具体位置的话，这里可不设为 0，再看看后面的 wFlags,

我传入了两个常数，这两个常数的相关说明请大家看看上面就知道，主要是不改变窗口位置和不改变窗口大小的前提下把窗口置前，其它常数如果大家有兴趣可以自己试试。

最后一个，看看 SetWindowRgn，这里我要解释一番，这个 API 所实现的功能呢就是改变窗口外观，也就是咱们所说的异形窗口等，通过这个 API 咱们可以把窗口改变成任何形状，在 API 浏览器找到这个 API，如下：

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

好了，第一个参数，句柄。第二个参数，Long 变量，这里需要传入什么咱们下面会讲到。第三个，Boolean 变量，可以说明这里需要传入布尔值，Redraw 为重画的意思，所以如果我们用这个 API 改变窗口形状，这里需要为 True，表示重画窗口。

现在新建一个标准 EXE，然后把 Form 的 ScaleMode 设置成 3-Pixel，我们知道 Windows 是以像素为单位的，所以使用这个 API 进行设置的时候是以像素为单位进行处理窗口外观。然后把 BorderStyle 设置为 0-None，这样看得更明显。好了，写入以下代码：

```
Private Declare Function SetWindowRgn Lib "user32" (ByVal hWnd As Long, ByVal hRgn As Long, ByVal bRedraw As Boolean) As Long
Private Declare Function CreateRoundRectRgn Lib "gdi32" (ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long, ByVal X3 As Long, ByVal Y3 As Long) As Long
Private Sub Form_Load()
    Dim hRgn As Long

    hRgn = CreateRoundRectRgn(0, 0, Me.ScaleWidth, Me.ScaleHeight, 10, 10)
    SetWindowRgn Me.hWnd, hRgn, True
End Sub
```

我不得不说一下这里我又用了一个 API，主要是因为使用 SetWindowRgn API 是需要和其它 API 一起进行工作的，首先让我们先看看 CreateRoundRectRgn 这个 API。分析如下：

整体的意思是:创建圆角矩形。这里提示大家一个技巧，一般 API 中包函 Rgn 字符的都是代表可以改变对象外观的。可以看看我们使用的两个 API，一个是 SetWindowRgn(Rgn)，一个是 CreateRoundRectRgn(Rgn),希望你能明白其中的共同点。

参数：x1,y1,x2,y2,x3,y3 这些都是坐标值，具体说明见以下：

X1,Y1 ---------- Long，矩形左上角的 X，Y 坐标
X2,Y2 ---------- Long，矩形右下角的 X，Y 坐标
X3 ---------- Long，圆角椭圆的宽。其范围从 0（没有圆角）到矩形宽（全圆）
Y3 ---------- Long，圆角椭圆的高。其范围从 0（没有圆角）到矩形高（全圆）

所以上面的代码具体是先通过 CreateRoundRectRgn 创建一个圆角矩形对象，然后通过 SetWindowRgn 来改变窗口的外观。

小提示：使用 CreateRoundRectRgn 可以创建圆角矩形，也可以使用 CreateEllipticRgn 创建椭圆形，CreatePolyPolygonRgn 创建多边形，CreateRectRgn 矩形等，细心观察它们最后

三个字符 Rgn 呵呵，明白了吧。

3，获取其它窗口的句柄

这个我本来打算不讲的，不过网友们既然提出来了，我也只好详细说说。一般获取其它窗口的句柄使用以下 API：

FindWindow，FindWindowEx，WindowFromPoint

这两个 API 就足矣，先看看第一个 API 的原型：

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

里面一共有两个参数，先看第一个：ByVal lpClassName As String,字符串变量，所以这里需要传入字符串，第二个 ByVal lpWindowName As String，同样一个字符串变量，这里也需要传入字符串。再看这个 API 为 Function，有返回值的，那返回值就是我们需要的句柄了。好了，现在了解了两个参数的具体传递类型，那我们现在就要知道这两个参数中到底应该传入哪些值？如下：

ByVal lpClassName As String，lpClassName：类名。指窗口类名，如果忽略则传入 vbNullString。

ByVal lpWindowName As String,lpWindowName：窗口名称。指窗口文本，如果忽略则传入 vbNullString。

现在明白了两个参数需要传入哪些值就好办了，一个窗口的类名咱们有可能不知道，但是一个窗口的名称就好办了。如：咱们打开记事本程序，可以看到窗口标题显示为“无标题－记事本”。好了这就是咱们需要的，现在咱们就要通过这个窗口标题来获取记事本的句柄。新建一个标准 EXE，然后输入以下代码：

```
Private Declare Function FindWindow Lib "user32" Alias "FindWindowA" (ByVal lpClassName As String, ByVal lpWindowName As String) As Long
  Private Sub Form_Load()
      Dim WindowHandle As Long

      WindowHandle = FindWindow(vbNullString, "无标题 - 记事本")

      MsgBox WindowHandle
  End Sub
```

好了，F5 运行，显示 MsgBox 消息框，如果不为 0，那么咱们就获取成功了，如果为 0，那么表示获取失败，这个时候你有必要检查一下你所要获取的窗口文本是否符合你所要获取的那个窗口文本(呵，这句话还真长！)。具体代码意思我就不讲了，大家可以自己分析下。

小提示：这个时候咱们已经得到句柄了，具体得到这个句柄干什么？那就看你了。给个例子，如下：

SetWindowText WindowHandle, "哈哈"

看看把这个代码放在上面代码中试下，呵呵！注意，SetWindowText 你要先声明这个 API。别忘了。

再看第二个 FindWindowEx，这个 API 是在窗口列表中寻找与指定条件相符的第一个子窗口，原型如下：

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

看里面的参数，第一个和第二个：ByVal hWnd1 As Long，ByVal hWnd2 As Long，这里都需要传入句柄，再看第三个和第四个：ByVal lpsz1 As String, ByVal lpsz2 As String，这里所要传入的是字符串。具体意思如下：

hWnd1 ---------- Long，在其中查找子的父窗口。如设为零，表示使用桌面窗口（通常说的顶级窗口都被认为是桌面的子窗口，所以也会对它们进行查找）

hWnd2 ---------- Long，从这个窗口后开始查找。这样便可利用对 FindWindowEx 的多次调用找到符合条件的所有子窗口。如设为零，表示从第一个子窗口开始搜索

lpsz1 ---------- String，欲搜索的类名。零表示忽略，注意一般传入 vbNullString

lpsz2 ---------- String，欲搜索的类名。零表示忽略，注意一般传入 vbNullString

用实践帮我们分析，这里还是拿记事本开刀。打开一个记事本，新建一个标准 EXE，接着新建一个 CommandButton，Caption 设置为：设置文本。OK，写入以下代码：

```
Private Declare Function FindWindow Lib "user32" Alias "FindWindowA" (ByVal lpClassName As String, ByVal lpWindowName As String) As Long
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 EnableWindow Lib "user32" (ByVal hwnd As Long, ByVal fEnable As Long) As Long
Private Sub Command1_Click()
    Dim WindowHandle As Long, ChildWindowHandle As Long

    WindowHandle = FindWindow(vbNullString, "无标题 - 记事本")

    If WindowHandle Then '如果获取句柄成功

        ChildWindowHandle = FindWindowEx(WindowHandle, 0, "Edit", vbNullString)

        If ChildWindowHandle Then '如果成功获取子句柄
            EnableWindow ChildWindowHandle, False '禁用子窗口
        Else
```

```
            MsgBox "无法获取子窗口"
        End If
    End If
End Sub
```

好了，帮大家分析。看第一行：Dim WindowHandle As Long, ChildWindowHandle As Long，用于储存获取的句柄的。WindowHandle = FindWindow(vbNullString, "无标题 - 记事本")这个就不用讲了，上面已经讲过。

ChildWindowHandle = FindWindowEx(WindowHandle, 0, "Edit", vbNullString)，这一段是通过我们已经获取的记事本句柄获取其中的子窗口句柄。大家可以用 Spy++查看到记事本的 TextBox 类，然后根据类名写入即可。

EnableWindow ChildWindowHandle, False 这又是一个新的 API，虽然前面我没有前过，但是这个 API 使用起来及其简单。这个 API 中有两个参数，第一个理所当然是传入窗口句柄，第二个为 Long 变量，其实这里应该设为 Boolean 变量好些，主要是用来处理当前窗口是否可用。True 可用，False 禁用。

现在 F5 运行，记得打开记事本哦，然后点击 Command1，看看能不能在记事本的文本框中输入字符串？是否被禁用了？

小提示：EnableWindow 之所有讲出来，是希望提高大家使用 API 的兴趣，有些被禁用的窗口你可以使用这个 API 把它激活，至于怎么使用就看你自己了，这里给大家布置一个作业，呵呵，自己去完成吧。

最后一个 API，WindowFromPoint，这个 API 主要是获取当前坐标的窗口句柄，不是有人想知道当前鼠标指针位置的窗口句柄吗？用这个是不错的选择，原型如下：

Private Declare Function WindowFromPoint Lib "user32" Alias "WindowFromPoint" (ByVal xPoint As Long, ByVal yPoint As Long) As Long

两个参数，一个是 xPoint(x 坐标值)，一个是 yPoint(y 坐标值)，现在你可以在这个两个参数分别传入其它窗口的坐标值就可以获取其它窗口的句柄了。可以看到为 Function 声明，返回值就是咱们需要的句柄。

咱们想实现的功能是获取当前鼠标指针位置的句柄，所以这里当然需要用到 GetCursorPos 了，结合前面所讲的，新建一个标准 EXE，添加一个 Timer 控件，Interval 设置为 100，Enabled=True，OK，写如以下代码：

```
Private Declare Function WindowFromPoint Lib "user32" (ByVal xPoint As Long, ByVal yPoint As Long) As Long
Private Declare Function GetCursorPos Lib "user32" (lpPoint As POINTAPI) As Long
    Private Type POINTAPI
            x As Long
            y As Long
    End Type
Private Sub Timer1_Timer()
        Dim lpPoint As POINTAPI
        Dim WindowHandle As Long

        GetCursorPos lpPoint '获取当前鼠标指针坐标
        WindowHandle = WindowFromPoint(lpPoint.x, lpPoint.y)
```

```
    Me.Caption = "当前鼠标指针位置句柄：" & WindowHandle
End Sub
```

好了，最后一次给大家分析了，至于 GetCursorPos 的使用与说明前面已经讲过，这里不再分析。看看 WindowHandle = WindowFromPoint(lpPoint.x, lpPoint.y)这句，它是通过 GetCursorPos 获取的鼠标坐标值获取当前鼠标坐标位置的句柄。最后一句我就不用说了，在程序窗口显示获取的句柄。

好了，API 入门已经告一段落，其实我还想写下去，不过似乎看的人多，响应的人少，很是打击我写下去的心情。不过还是希望大家能从上面学到一些知识。具体的 API 应用我就不多说，大家可以自己慢慢体会。如果你把以上我讲的全部都搞懂的话，那么证明你已经基本了解 API 的使用方法了，那下面就靠你自己了。至此，我希望我带了一个好头帮助你了解 API。

VB API 大全

所有类别
类别
控件与消息函数共 91 个函数
硬件与系统函数共 98 个函数
设备场景函数共 73 个函数
绘图函数共 105 个函数
位图、图标和光栅运算函数共 39 个函数
菜单函数共 37 个函数
文本和字体函数共 41 个函数
打印函数共 66 个函数
文件处理函数共 118 个函数
进程和线程函数共 40 个函数
Windows 消息函数共 11 个函数
网络函数共 14 个函数
Windows 消息函数
Windows 消息函数，共一页。第一页
BroadcastSystemMessage 将一条系统消息广播给系统中所有的顶级窗口
GetMessagePos 取得消息队列中上一条消息处理完毕时的鼠标指针屏幕位置
GetMessageTime 取得消息队列中上一条消息处理完毕时的时间
PostMessage 将一条消息投递到指定窗口的消息队列
PostThreadMessage 将一条消息投递给应用程序
RegisterWindowMessage 获取分配给一个字串标识符的消息编号
ReplyMessage 答复一个消息
SendMessage 调用一个窗口的窗口函数，将一条消息发给那个窗口
SendMessageCallback 将一条消息发给窗口

SendMessageTimeout 向窗口发送一条消息
SendNotifyMessage 向窗口发送一条消息
完
第一页
Windows 消息函数，共一页。第一页
BroadcastSystemMessage 将一条系统消息广播给系统中所有的顶级窗口
GetMessagePos 取得消息队列中上一条消息处理完毕时的鼠标指针屏幕位置
GetMessageTime 取得消息队列中上一条消息处理完毕时的时间
PostMessage 将一条消息投递到指定窗口的消息队列
PostThreadMessage 将一条消息投递给应用程序
RegisterWindowMessage 获取分配给一个字串标识符的消息编号
ReplyMessage 答复一个消息
SendMessage 调用一个窗口的窗口函数，将一条消息发给那个窗口
SendMessageCallback 将一条消息发给窗口
SendMessageTimeout 向窗口发送一条消息
SendNotifyMessage 向窗口发送一条消息
完
文件处理函数
文件处理函数，共八页。第一页，第二页，第三页，第四页，第五页，第六页，第七页，第八页
CloseHandle 关闭一个内核对象。其中包括文件、文件映射、进程、线程、安全和同步对象等
CompareFileTime 对比两个文件的时间
CopyFile 复制文件
CreateDirectory 创建一个新目录
CreateFile 打开和创建文件、管道、邮槽、通信服务、设备以及控制台
CreateFileMapping 创建一个新的文件映射对象
DeleteFile 删除指定文件
DeviceIoControl 对设备执行指定的操作
DosDateTimeToFileTime 将 DOS 日期和时间值转换成一个 win32 FILETIME 值
FileTimeToDosDateTime 将一个 win32 FILETIME 值转换成 DOS 日期和时间值
FileTimeToLocalFileTime 将一个 FILETIME 结构转换成本地时间
FileTimeToSystemTime 根据一个 FILETIME 结构的内容，装载一个 SYSTEMTIME 结构
FindClose 关闭由 FindFirstFile 函数创建的一个搜索句柄
FindFirstFile 根据文件名查找文件
FindNextFile 根据调用 FindFirstFile 函数时指定的一个文件名查找下一个文件
FlushFileBuffers 针对指定的文件句柄，刷新内部文件缓冲区
第一页
文件处理函数，共八页。第一页，第二页，第三页，第四页，第五页，第六页，第七页，第八页
CloseHandle 关闭一个内核对象。其中包括文件、文件映射、进程、线程、安全和同步对象等
CompareFileTime 对比两个文件的时间
CopyFile 复制文件

CreateDirectory 创建一个新目录
CreateFile 打开和创建文件、管道、邮槽、通信服务、设备以及控制台
CreateFileMapping 创建一个新的文件映射对象
DeleteFile 删除指定文件
DeviceIoControl 对设备执行指定的操作
DosDateTimeToFileTime 将 DOS 日期和时间值转换成一个 win32 FILETIME 值
FileTimeToDosDateTime 将一个 win32 FILETIME 值转换成 DOS 日期和时间值
FileTimeToLocalFileTime 将一个 FILETIME 结构转换成本地时间
FileTimeToSystemTime 根据一个 FILETIME 结构的内容，装载一个 SYSTEMTIME 结构
FindClose 关闭由 FindFirstFile 函数创建的一个搜索句柄
FindFirstFile 根据文件名查找文件
FindNextFile 根据调用 FindFirstFile 函数时指定的一个文件名查找下一个文件
FlushFileBuffers 针对指定的文件句柄，刷新内部文件缓冲区
第二页
文件处理函数，共八页。第一页，第二页，第三页，第四页，第五页，第六页，第七页，第八页
FlushViewOfFile 将写入文件映射缓冲区的所有数据都刷新到磁盘
GetBinaryType 判断文件是否可以执行
GetCompressedFileSize 判断一个压缩文件在磁盘上实际占据的字节数
GetCurrentDirectory 在一个缓冲区中装载当前目录
GetDiskFreeSpace 获取与一个磁盘的组织有关的信息，以及了解剩余空间的容量
GetDiskFreeSpaceEx 获取与一个磁盘的组织以及剩余空间容量有关的信息
GetDriveType 判断一个磁盘驱动器的类型
GetExpandedName 取得一个压缩文件的全名
GetFileAttributes 判断指定文件的属性
GetFileInformationByHandle 这个函数提供了获取文件信息的一种机制
GetFileSize 判断文件长度
GetFileTime 取得指定文件的时间信息
GetFileType 在给出文件句柄的前提下，判断文件类型
GetFileVersionInfo 从支持版本标记的一个模块里获取文件版本信息
GetFileVersionInfoSize 针对包含了版本资源的一个文件，判断容纳文件版本信息需要一个多大的缓冲区
GetFullPathName 获取指定文件的完整路径名
第三页
文件处理函数，共八页。第一页，第二页，第三页，第四页，第五页，第六页，第七页，第八页
GetLogicalDrives 判断系统中存在哪些逻辑驱动器字母
GetLogicalDriveStrings 获取一个字串，其中包含了当前所有逻辑驱动器的根驱动器路径
GetOverlappedResult 判断一个重叠操作当前的状态
GetPrivateProfileInt 为初始化文件（.ini 文件）中指定的条目获取一个整数值
GetPrivateProfileSection 获取指定小节（在.ini 文件中）所有项名和值的一个列表
GetPrivateProfileString 为初始化文件中指定的条目取得字串
GetProfileInt 取得 win.ini 初始化文件中指定条目的一个整数值
GetProfileSection 获取指定小节（在 win.ini 文件中）所有项名和值的一个列表

GetProfileString 为 win.ini 初始化文件中指定的条目取得字串
GetShortPathName 获取指定文件的短路径名
GetSystemDirectory 取得 Windows 系统目录（即 System 目录）的完整路径名
GetTempFileName 这个函数包含了一个临时文件的名字，它可由应用程序使用
GetTempPath 获取为临时文件指定的路径
GetVolumeInformation 获取与一个磁盘卷有关的信息
GetWindowsDirectory 获取 Windows 目录的完整路径名
hread 参考 lread
第四页
文件处理函数，共八页。第一页，第二页，第三页，第四页，第五页，第六页，第七页，第八页
hwrite 参考 lwrite 函数
lclose 关闭指定的文件
lcreat 创建一个文件
llseek 设置文件中进行读写的当前位置
LockFile 锁定文件的某一部分，使其不与其他应用程序共享
LockFileEx 与 LockFile 相似，只是它提供了更多的功能
lopen 以二进制模式打开指定的文件
lread 将文件中的数据读入内存缓冲区
lwrite 将数据从内存缓冲区写入一个文件
LZClose 关闭由 LZOpenFile 或 LZInit 函数打开的一个文件
LZCopy 复制一个文件
LZInit 这个函数用于初始化内部缓冲区
LZOpenFile 该函数能执行大量不同的文件处理，而且兼容于压缩文件
LZRead 将数据从文件读入内存缓冲区
LZSeek 设置一个文件中进行读写的当前位置
MapViewOfFile 将一个文件映射对象映射到当前应用程序的地址空间
第五页
文件处理函数，共八页。第一页，第二页，第三页，第四页，第五页，第六页，第七页，第八页
MoveFile 移动文件
OpenFile 这个函数能执行大量不同的文件操作
OpenFileMapping 打开一个现成的文件映射对象
QueryDosDevice 在 Windows NT 中，DOS 设备名会映射成 NT 系统设备名。该函数可判断当前的设备映射情况
ReadFile 从文件中读出数据
ReadFileEx 与 ReadFile 相似，只是它只能用于异步读操作，并包含了一个完整的回调
RegCloseKey 关闭系统注册表中的一个项（或键）
RegConnectRegistry 访问远程系统的部分注册表
RegCreateKey 在指定的项下创建或打开一个项
RegCreateKeyEx 在指定项下创建新项的更复杂的方式。在 Win32 环境中建议使用这个函数
RegDeleteKey 删除现有项下方一个指定的子项
RegDeleteValue 删除指定项下方的一个值
RegEnumKey 枚举指定项的子项。在 Win32 环境中应使用 RegEnumKeyEx

RegEnumKeyEx 枚举指定项下方的子项
RegEnumValue 枚举指定项的值
RegFlushKey 将对项和它的子项作出的改动实际写入磁盘
第六页
文件处理函数，共八页。第一页，第二页，第三页，第四页，第五页，第六页，第七页，第八页
RegGetKeySecurity 获取与一个注册表项有关的安全信息
RegLoadKey 从以前用 RegSaveKey 函数创建的一个文件里装载注册表信息
RegNotifyChangeKeyValue 注册表项或它的任何一个子项发生变化时，用这个函数提供一种通知机制
RegOpenKey 打开一个现有的注册表项
RegOpenKeyEx 打开一个现有的项。在 win32 下推荐使用这个函数
RegQueryInfoKey 获取与一个项有关的信息
RegQueryValue 取得指定项或子项的默认（未命名）值
RegQueryValueEx 获取一个项的设置值
RegReplaceKey 用一个磁盘文件保存的信息替换注册表信息；并创建一个备份，在其中包含当前注册表信息
RegRestoreKey 从一个磁盘文件恢复注册表信息
RegSaveKey 将一个项以及它的所有子项都保存到一个磁盘文件
RegSetKeySecurity 设置指定项的安全特性
RegSetValue 设置指定项或子项的默认值
RegSetValueEx 设置指定项的值
RegUnLoadKey 卸载指定的项以及它的所有子项
RemoveDirectory 删除指定目录
第七页
文件处理函数，共八页。第一页，第二页，第三页，第四页，第五页，第六页，第七页，第八页
SearchPath 查找指定文件
SetCurrentDirectory 设置当前目录
SetEndOfFile 针对一个打开的文件，将当前文件位置设为文件末尾
SetFileAttributes 设置文件属性
SetFilePointer 在一个文件中设置当前的读写位置
SetFileTime 设置文件的创建、访问及上次修改时间
SetHandleCount 这个函数不必在 win32 下使用；即使使用，也不会有任何效果
SetVolumeLabel 设置一个磁盘的卷标（Label）
SystemTimeToFileTime 根据一个 FILETIME 结构的内容，载入一个 SYSTEMTIME 结构
UnlockFile 解除对一个文件的锁定
UnlockFileEx 解除对一个文件的锁定
UnmapViewOfFile 在当前应用程序的内存地址空间解除对一个文件映射对象的映射
VerFindFile 用这个函数决定一个文件应安装到哪里
VerInstallFile 用这个函数安装一个文件
VerLanguageName 这个函数能根据 16 位语言代码获取一种语言的名称
VerQueryValue 这个函数用于从版本资源中获取信息
第八页

文件处理函数，共八页。第一页，第二页，第三页，第四页，第五页，第六页，第七页，第八页
WriteFile 将数据写入一个文件
WriteFileEx 与 WriteFile 类似，只是它只能用于异步写操作，并包括了一个完整的回调
WritePrivateProfileSection 为一个初始化文件（.ini）中指定的小节设置所有项名和值
WritePrivateProfileString 在初始化文件指定小节内设置一个字串
WriteProfileSection 为 Win.ini 初始化文件中一个指定的小节设置所有项名和值
WriteProfileString 在 Win.ini 初始化文件指定小节内设置一个字串
完
网络函数
网络函数，共一页。第一页
WNetAddConnection 创建同一个网络资源的永久性连接
WNetAddConnection2 创建同一个网络资源的连接
WNetAddConnection3 创建同一个网络资源的连接
WNetCancelConnection 结束一个网络连接
WNetCancelConnection2 结束一个网络连接
WNetCloseEnum 结束一次枚举操作
WNetConnectionDialog 启动一个标准对话框，以便建立同网络资源的连接
WNetDisconnectDialog 启动一个标准对话框，以便断开同网络资源的连接
WNetEnumResource 枚举网络资源
WNetGetConnection 获取本地或已连接的一个资源的网络名称
WNetGetLastError 获取网络错误的扩展错误信息
WNetGetUniversalName 获取网络中一个文件的远程名称以及/或者 UNC（统一命名规范）名称
WNetGetUser 获取一个网络资源用以连接的名字
WNetOpenEnum 启动对网络资源进行枚举的过程
完
第一页
网络函数，共一页。第一页
WNetAddConnection 创建同一个网络资源的永久性连接
WNetAddConnection2 创建同一个网络资源的连接
WNetAddConnection3 创建同一个网络资源的连接
WNetCancelConnection 结束一个网络连接
WNetCancelConnection2 结束一个网络连接
WNetCloseEnum 结束一次枚举操作
WNetConnectionDialog 启动一个标准对话框，以便建立同网络资源的连接
WNetDisconnectDialog 启动一个标准对话框，以便断开同网络资源的连接
WNetEnumResource 枚举网络资源
WNetGetConnection 获取本地或已连接的一个资源的网络名称
WNetGetLastError 获取网络错误的扩展错误信息
WNetGetUniversalName 获取网络中一个文件的远程名称以及/或者 UNC（统一命名规范）名称
WNetGetUser 获取一个网络资源用以连接的名字
WNetOpenEnum 启动对网络资源进行枚举的过程

完
菜单函数
菜单函数：共三页。第一页，第二页，第三页
AppendMenu 在指定的菜单里添加一个菜单项
CheckMenuItem 复选或撤消复选指定的菜单条目
CheckMenuRadioItem 指定一个菜单条目被复选成“单选”项目
CreateMenu 创建新菜单
CreatePopupMenu 创建一个空的弹出式菜单
DeleteMenu 删除指定的菜单条目
DestroyMenu 删除指定的菜单
DrawMenuBar 为指定的窗口重画菜单
EnableMenuItem 允许或禁止指定的菜单条目
GetMenu 取得窗口中一个菜单的句柄
GetMenuCheckMarkDimensions 返回一个菜单复选符的大小
GetMenuContextHelpId 取得一个菜单的帮助场景 ID
GetMenuDefaultItem 判断菜单中的哪个条目是默认条目
GetMenuItemCount 返回菜单中条目（菜单项）的数量
GetMenuItemID 返回位于菜单中指定位置处的条目的菜单 ID
GetMenuItemInfo 取得（接收）与一个菜单条目有关的特定信息
第一页
菜单函数：共三页。第一页，第二页，第三页
AppendMenu 在指定的菜单里添加一个菜单项
CheckMenuItem 复选或撤消复选指定的菜单条目
CheckMenuRadioItem 指定一个菜单条目被复选成“单选”项目
CreateMenu 创建新菜单
CreatePopupMenu 创建一个空的弹出式菜单
DeleteMenu 删除指定的菜单条目
DestroyMenu 删除指定的菜单
DrawMenuBar 为指定的窗口重画菜单
EnableMenuItem 允许或禁止指定的菜单条目
GetMenu 取得窗口中一个菜单的句柄
GetMenuCheckMarkDimensions 返回一个菜单复选符的大小
GetMenuContextHelpId 取得一个菜单的帮助场景 ID
GetMenuDefaultItem 判断菜单中的哪个条目是默认条目
GetMenuItemCount 返回菜单中条目（菜单项）的数量
GetMenuItemID 返回位于菜单中指定位置处的条目的菜单 ID
GetMenuItemInfo 取得（接收）与一个菜单条目有关的特定信息
第二页
菜单函数：共三页。第一页，第二页，第三页
GetMenuItemRect 在一个矩形中装载指定菜单条目的屏幕坐标信息
GetMenuState 取得与指定菜单条目状态有关的信息
GetMenuString 取得指定菜单条目的字串
GetSubMenu 取得一个弹出式菜单的句柄，它位于菜单中指定的位置
GetSystemMenu 取得指定窗口的系统菜单的句柄

HiliteMenuItem 控制顶级菜单条目的加亮显示状态
InsertMenu 在菜单的指定位置处插入一个菜单条目，并根据需要将其他条目向下移动
InsertMenuItem 插入一个新菜单条目
IsMenu 判断指定的句柄是否为一个菜单的句柄
LoadMenu 从指定的模块或应用程序实例中载入一个菜单
LoadMenuIndirect 载入一个菜单
MenuItemFromPoint 判断哪个菜单条目包含了屏幕上一个指定的点
ModifyMenu 改变菜单条目
RemoveMenu 删除指定的菜单条目
SetMenu 设置窗口菜单
SetMenuContextHelpId 设置一个菜单的帮助场景 ID
第三页
菜单函数：共三页。第一页，第二页，第三页
SetMenuDefaultItem 将一个菜单条目设为默认条目
SetMenuItemBitmaps 设置一幅特定位图，令其在指定的菜单条目中使用，代替标准的复选符号（√）
SetMenuItemInfo 为一个菜单条目设置指定的信息
TrackPopupMenu 在屏幕的任意地方显示一个弹出式菜单
TrackPopupMenuEx 与 TrackPopupMenu 相似，只是它提供了额外的功能
完
文本和字体函数
文本和字体函数，共三页。第一页，第二页，第三页
AddFontResource 在 Windows 系统中添加一种字体资源
CreateFont 用指定的属性创建一种逻辑字体
CreateFontIndirect 用指定的属性创建一种逻辑字体
CreateScalableFontResource 为一种 TureType 字体创建一个资源文件，以便能用 API 函数 AddFontResource 将其加入 Windows 系统
DrawText 将文本描绘到指定的矩形中
DrawTextEx 与 DrawText 相似，只是加入了更多的功能
EnumFontFamilies 列举指定设备可用的字体
EnumFontFamiliesEx 列举指定设备可用的字体
EnumFonts 列举指定设备可用的字体
ExtTextOut 经过扩展的文本描绘函数。也请参考 SetTextAlign 函数
GetAspectRatioFilterEx 用 SetMapperFlags 要求 Windows 只选择与设备当前纵横比相符的光栅字体时，本函数可判断纵横比大小
GetCharABCWidths 判断 TureType 字体中一个或多个字符的 A-B-C 大小
GetCharABCWidthsFloat 查询一种字体中一个或多个字符的 A-B-C 尺寸
GetCharacterPlacement 该函数用于了解如何用一个给定的字符显示一个字串
GetCharWidth 调查字体中一个或多个字符的宽度
GetFontData 接收一种可缩放字体文件的数据
第一页
文本和字体函数，共三页。第一页，第二页，第三页
AddFontResource 在 Windows 系统中添加一种字体资源
CreateFont 用指定的属性创建一种逻辑字体

CreateFontIndirect 用指定的属性创建一种逻辑字体
CreateScalableFontResource 为一种 TureType 字体创建一个资源文件，以便能用 API 函数 AddFontResource 将其加入 Windows 系统
DrawText 将文本描绘到指定的矩形中
DrawTextEx 与 DrawText 相似，只是加入了更多的功能
EnumFontFamilies 列举指定设备可用的字体
EnumFontFamiliesEx 列举指定设备可用的字体
EnumFonts 列举指定设备可用的字体
ExtTextOut 经过扩展的文本描绘函数。也请参考 SetTextAlign 函数
GetAspectRatioFilterEx 用 SetMapperFlags 要求 Windows 只选择与设备当前纵横比相符的光栅字体时，本函数可判断纵横比大小
GetCharABCWidths 判断 TureType 字体中一个或多个字符的 A-B-C 大小
GetCharABCWidthsFloat 查询一种字体中一个或多个字符的 A-B-C 尺寸
GetCharacterPlacement 该函数用于了解如何用一个给定的字符显示一个字串
GetCharWidth 调查字体中一个或多个字符的宽度
GetFontData 接收一种可缩放字体文件的数据

GetFontLanguageInfo 返回目前选入指定设备场景中的字体的信息
GetGlyphOutline 取得 TureType 字体中构成一个字符的曲线信息
GetKerningPairs 取得指定字体的字距信息
GetOutlineTextMetrics 接收与 TureType 字体内部特征有关的详细信息
GetRasterizerCaps 了解系统是否有能力支持可缩放的字体
GetTabbedTextExtent 判断一个字串占据的范围，同时考虑制表站扩充的因素
GetTextAlign 接收一个设备场景当前的文本对齐标志
GetTextCharacterExtra 判断额外字符间距的当前值
GetTextCharset 接收当前选入指定设备场景的字体的字符集标识符
GetTextCharsetInfo 获取与当前选定字体的字符集有关的详细信息
GetTextColor 判断当前字体颜色。通常也称为“前景色”
GetTextExtentExPoint 判断要填入指定区域的字符数量。也用一个数组装载每个字符的范围信息
GetTextExtentPoint 判断一个字串的大小（范围）
GetTextFace 获取一种字体的字样名
GetTextMetrics 获取与选入一种设备场景的物理字体有关的信息
GrayString 描绘一个以灰色显示的字串。通常由 Windows 用于标识禁止状态

PolyTextOut 描绘一系列字串
RemoveFontResource 从 Windows 系统中删除一种字体资源
SetMapperFlagsWindows 对字体进行映射时，可用该函数选择与目标设备的纵横比相符的光栅字体
SetTextAlign 设置文本对齐方式，并指定在文本输出过程中使用设备场景的当前位置
SetTextCharacterExtra 描绘文本的时候，指定要在字符间插入的额外间距
SetTextColor 设置当前文本颜色。这种颜色也称为“前景色”

SetTextJustification 通过指定一个文本行应占据的额外空间，可用这个函数对文本进行两端对齐处理
TabbedTextOut 支持制表站的一个文本描绘函数
TextOut 文本绘图函数
完
硬件与系统函数
硬件与系统函数：共七页。第一页，第二页，第三页，第四页，第五页，第六页，第七页
ActivateKeyboardLayout 激活一个新的键盘布局。键盘布局定义了按键在一种物理性键盘上的位置与含义
Beep 用于生成简单的声音
CharToOem 将一个字串从 ANSI 字符集转换到 OEM 字符集
ClipCursor 将指针限制到指定区域
ConvertDefaultLocale 将一个特殊的地方标识符转换成真实的地方 ID
CreateCaret 根据指定的信息创建一个插入符（光标），并将它选定为指定窗口的默认插入符
DestroyCaret 清除（破坏）一个插入符
EnumCalendarInfo 枚举在指定“地方”环境中可用的日历信息
EnumDateFormats 列举指定的“当地”设置中可用的长、短日期格式
EnumSystemCodePages 枚举系统中已安装或支持的代码页
EnumSystemLocales 枚举系统已经安装或提供支持的“地方”设置
EnumTimeFormats 枚举一个指定的地方适用的时间格式
ExitWindowsEx 退出 windows，并用特定的选项重新启动
ExpandEnvironmentStrings 扩充环境字串
FreeEnvironmentStrings 翻译指定的环境字串块
GetACP 判断目前正在生效的 ANSI 代码页
第一页
硬件与系统函数：共七页。第一页，第二页，第三页，第四页，第五页，第六页，第七页
ActivateKeyboardLayout 激活一个新的键盘布局。键盘布局定义了按键在一种物理性键盘上的位置与含义
Beep 用于生成简单的声音
CharToOem 将一个字串从 ANSI 字符集转换到 OEM 字符集
ClipCursor 将指针限制到指定区域
ConvertDefaultLocale 将一个特殊的地方标识符转换成真实的地方 ID
CreateCaret 根据指定的信息创建一个插入符（光标），并将它选定为指定窗口的默认插入符
DestroyCaret 清除（破坏）一个插入符
EnumCalendarInfo 枚举在指定“地方”环境中可用的日历信息
EnumDateFormats 列举指定的“当地”设置中可用的长、短日期格式
EnumSystemCodePages 枚举系统中已安装或支持的代码页
EnumSystemLocales 枚举系统已经安装或提供支持的“地方”设置
EnumTimeFormats 枚举一个指定的地方适用的时间格式
ExitWindowsEx 退出 windows，并用特定的选项重新启动
ExpandEnvironmentStrings 扩充环境字串
FreeEnvironmentStrings 翻译指定的环境字串块
GetACP 判断目前正在生效的 ANSI 代码页
第二页

GetAsyncKeyState 判断函数调用时指定虚拟键的状态
GetCaretBlinkTime 判断插入符光标的闪烁频率
GetCaretPos 判断插入符的当前位置
GetClipCursor 取得一个矩形，用于描述目前为鼠标指针规定的剪切区域
GetCommandLine 获得指向当前命令行缓冲区的一个指针
GetComputerName 取得这台计算机的名称
GetCPInfo 取得与指定代码页有关的信息
GetCurrencyFormat 针对指定的“地方”设置，根据货币格式格式化一个数字
GetCursor 获取目前选择的鼠标指针的句柄
GetCursorPos 获取鼠标指针的当前位置
GetDateFormat 针对指定的“当地”格式，对一个系统日期进行格式化
GetDoubleClickTime 判断连续两次鼠标单击之间会被处理成双击事件的间隔时间
GetEnvironmentStrings 为包含了当前环境字串设置的一个内存块分配和返回一个句柄
GetEnvironmentVariable 取得一个环境变量的值
GetInputState 判断是否存在任何待决（等待处理）的鼠标或键盘事件
GetKBCodePage 由 GetOEMCP 取代，两者功能完全相同

GetKeyboardLayout 取得一个句柄，描述指定应用程序的键盘布局
GetKeyboardLayoutList 获得系统适用的所有键盘布局的一个列表
GetKeyboardLayoutName 取得当前活动键盘布局的名称
GetKeyboardState 取得键盘上每个虚拟键当前的状态
GetKeyboardType 了解与正在使用的键盘有关的信息
GetKeyNameText 在给出扫描码的前提下，判断键名
GetKeyState 针对已处理过的按键，在最近一次输入信息时，判断指定虚拟键的状态
GetLastError 针对之前调用的 api 函数，用这个函数取得扩展错误信息
GetLocaleInfo 取得与指定“地方”有关的信息
GetLocalTime 取得本地日期和时间
GetNumberFormat 针对指定的“地方”，按特定的格式格式化一个数字
GetOEMCP 判断在 OEM 和 ANSI 字符集间转换的 windows 代码页
GetQueueStatus 判断应用程序消息队列中待决（等待处理）的消息类型
GetSysColor 判断指定 windows 显示对象的颜色
GetSystemDefaultLangID 取得系统的默认语言 ID
GetSystemDefaultLCID 取得当前的默认系统“地方”

GetSystemInfo 取得与底层硬件平台有关的信息
GetSystemMetrics 返回与 windows 环境有关的信息
GetSystemPowerStatus 获得与当前系统电源状态有关的信息
GetSystemTime 取得当前系统时间，这个时间采用的是“协同世界时间”（即 UTC，也叫做 GMT）格式
GetSystemTimeAdjustment 使内部系统时钟与一个外部的时钟信号源同步
GetThreadLocale 取得当前线程的地方 ID

GetTickCount 用于获取自 windows 启动以来经历的时间长度（毫秒）
GetTimeFormat 针对当前指定的“地方”，按特定的格式格式化一个系统时间
GetTimeZoneInformation 取得与系统时区设置有关的信息
GetUserDefaultLangID 为当前用户取得默认语言 ID
GetUserDefaultLCID 取得当前用户的默认“地方”设置
GetUserName 取得当前用户的名字
GetVersion 判断当前运行的 Windows 和 DOS 版本
GetVersionEx 取得与平台和操作系统有关的版本信息
HideCaret 在指定的窗口隐藏插入符（光标）
IsValidCodePage 判断一个代码页是否有效
第五页
硬件与系统函数：共七页。第一页，第二页，第三页，第四页，第五页，第六页，第七页
IsValidLocale 判断地方标识符是否有效
keybd_event 这个函数模拟了键盘行动
LoadKeyboardLayout 载入一个键盘布局
MapVirtualKey 根据指定的映射类型，执行不同的扫描码和字符转换
MapVirtualKeyEx 根据指定的映射类型，执行不同的扫描码和字符转换
MessageBeep 播放一个系统声音。系统声音的分配方案是在控制面板里决定的
mouse_event 模拟一次鼠标事件
OemKeyScan 判断 OEM 字符集中的一个 ASCII 字符的扫描码和 Shift 键状态
OemToChar 将 OEM 字符集的一个字串转换到 ANSI 字符集
SetCaretBlinkTime 指定插入符（光标）的闪烁频率
SetCaretPos 指定插入符的位置
SetComputerName 设置新的计算机名
SetCursor 将指定的鼠标指针设为当前指针
SetCursorPos 设置指针的位置
SetDoubleClickTime 设置连续两次鼠标单击之间能使系统认为是双击事件的间隔时间
SetEnvironmentVariable 将一个环境变量设为指定的值
第六页
硬件与系统函数：共七页。第一页，第二页，第三页，第四页，第五页，第六页，第七页
SetKeyboardState 设置每个虚拟键当前在键盘上的状态
SetLocaleInfo 改变用户“地方”设置信息
SetLocalTime 设置当前地方时间
SetSysColors 设置指定窗口显示对象的颜色
SetSystemCursor 改变任何一个标准系统指针
SetSystemTime 设置当前系统时间
SetSystemTimeAdjustment 定时添加一个校准值使内部系统时钟与一个外部的时钟信号源同步
SetThreadLocale 为当前线程设置地方
SetTimeZoneInformation 设置系统时区信息
ShowCaret 在指定的窗口里显示插入符（光标）
ShowCursor 控制鼠标指针的可视性
SwapMouseButton 决定是否互换鼠标左右键的功能
SystemParametersInfo 获取和设置数量众多的 windows 系统参数

SystemTimeToTzSpecificLocalTime 将系统时间转换成地方时间
ToAscii 根据当前的扫描码和键盘信息，将一个虚拟键转换成 ASCII 字符
ToUnicode 根据当前的扫描码和键盘信息，将一个虚拟键转换成 Unicode 字符
第七页
硬件与系统函数：共七页。第一页，第二页，第三页，第四页，第五页，第六页，第七页
UnloadKeyboardLayout 卸载指定的键盘布局
VkKeyScan 针对 Windows 字符集中一个 ASCII 字符，判断虚拟键码和 Shift 键的状态
完
控件与消息函数
控件与消息函数：共六页。第一页，第二页，第三页，第四页，第五页，第六页
AdjustWindowRect 给定一种窗口样式，计算获得目标客户区矩形所需的窗口大小
AnyPopup 判断屏幕上是否存在任何弹出式窗口
ArrangeIconicWindows 排列一个父窗口的最小化子窗口
AttachThreadInput 连接线程输入函数
BeginDeferWindowPos 启动构建一系列新窗口位置的过程
BringWindowToTop 将指定的窗口带至窗口列表顶部
CascadeWindows 以层叠方式排列窗口
ChildWindowFromPoint 返回父窗口中包含了指定点的第一个子窗口的句柄
ClientToScreen 判断窗口内以客户区坐标表示的一个点的屏幕坐标
CloseWindow 最小化指定的窗口
CopyRect 矩形内容复制
DeferWindowPos 该函数为特定的窗口指定一个新窗口位置
DestroyWindow 清除指定的窗口以及它的所有子窗口
DrawAnimatedRects 描绘一系列动态矩形
EnableWindow 指定的窗口里允许或禁止所有鼠标及键盘输入
EndDeferWindowPos 同时更新 DeferWindowPos 调用时指定的所有窗口的位置及状态
第一页
控件与消息函数：共六页。第一页，第二页，第三页，第四页，第五页，第六页
AdjustWindowRect 给定一种窗口样式，计算获得目标客户区矩形所需的窗口大小
AnyPopup 判断屏幕上是否存在任何弹出式窗口
ArrangeIconicWindows 排列一个父窗口的最小化子窗口
AttachThreadInput 连接线程输入函数
BeginDeferWindowPos 启动构建一系列新窗口位置的过程
BringWindowToTop 将指定的窗口带至窗口列表顶部
CascadeWindows 以层叠方式排列窗口
ChildWindowFromPoint 返回父窗口中包含了指定点的第一个子窗口的句柄
ClientToScreen 判断窗口内以客户区坐标表示的一个点的屏幕坐标
CloseWindow 最小化指定的窗口
CopyRect 矩形内容复制
DeferWindowPos 该函数为特定的窗口指定一个新窗口位置
DestroyWindow 清除指定的窗口以及它的所有子窗口
DrawAnimatedRects 描绘一系列动态矩形
EnableWindow 指定的窗口里允许或禁止所有鼠标及键盘输入
EndDeferWindowPos 同时更新 DeferWindowPos 调用时指定的所有窗口的位置及状态

第二页
控件与消息函数：共六页。第一页，第二页，第三页，第四页，第五页，第六页
EnumChildWindows 为指定的父窗口枚举子窗口
EnumThreadWindows 枚举与指定任务相关的窗口
EnumWindows 枚举窗口列表中的所有父窗口
EqualRect 判断两个矩形结构是否相同
FindWindow 寻找窗口列表中第一个符合指定条件的顶级窗口
FindWindowEx 在窗口列表中寻找与指定条件相符的第一个子窗口
FlashWindow 闪烁显示指定窗口
GetActiveWindow 获得活动窗口的句柄
GetCapture 获得一个窗口的句柄，这个窗口位于当前输入线程，且拥有鼠标捕获（鼠标活动由它接收）
GetClassInfo 取得 WNDCLASS 结构（或 WNDCLASSEX 结构）的一个副本，结构中包含了与指定类有关的信息
GetClassLong 取得窗口类的一个 Long 变量条目
GetClassName 为指定的窗口取得类名
GetClassWord 为窗口类取得一个整数变量
GetClientRect 返回指定窗口客户区矩形的大小
GetDesktopWindow 获得代表整个屏幕的一个窗口（桌面窗口）句柄
GetFocus 获得拥有输入焦点的窗口的句柄
第三页
控件与消息函数：共六页。第一页，第二页，第三页，第四页，第五页，第六页
GetForegroundWindow 获得前台窗口的句柄
GetLastActivePopup 获得在一个给定父窗口中最近激活过的弹出式窗口的句柄
GetParent 判断指定窗口的父窗口
GetTopWindow 搜索内部窗口列表，寻找隶属于指定窗口的头一个窗口的句柄
GetUpdateRect 获得一个矩形，它描叙了指定窗口中需要更新的那一部分
GetWindow 获得一个窗口的句柄，该窗口与某源窗口有特定的关系
GetWindowContextHelpId 取得与窗口关联在一起的帮助场景 ID
GetWindowLong 从指定窗口的结构中取得信息
GetWindowPlacement 获得指定窗口的状态及位置信息
GetWindowRect 获得整个窗口的范围矩形，窗口的边框、标题栏、滚动条及菜单等都在这个矩形内
GetWindowText 取得一个窗体的标题（caption）文字，或者一个控件的内容
GetWindowTextLength 调查窗口标题文字或控件内容的长短
GetWindowWord 获得指定窗口结构的信息
InflateRect 增大或减小一个矩形的大小
IntersectRect 这个函数在 lpDestRect 里载入一个矩形，它是 lpSrc1Rect 与 lpSrc2Rect 两个矩形的交集
InvalidateRect 屏蔽一个窗口客户区的全部或部分区域
第四页
控件与消息函数：共六页。第一页，第二页，第三页，第四页，第五页，第六页
IsChild 判断一个窗口是否为另一窗口的子或隶属窗口
IsIconic 判断窗口是否已最小化

IsRectEmpty 判断一个矩形是否为空
IsWindow 判断一个窗口句柄是否有效
IsWindowEnabled 判断窗口是否处于活动状态
IsWindowUnicode 判断一个窗口是否为 Unicode 窗口。这意味着窗口为所有基于文本的消息都接收 Unicode 文字
IsWindowVisible 判断窗口是否可见
IsZoomed 判断窗口是否最大化
LockWindowUpdate 锁定指定窗口，禁止它更新
MapWindowPoints 将一个窗口客户区坐标的点转换到另一窗口的客户区坐标系统
MoveWindow 改变指定窗口的位置和大小
OffsetRect 通过应用一个指定的偏移，从而让矩形移动起来
OpenIcon 恢复一个最小化的程序，并将其激活
PtInRect 判断指定的点是否位于矩形内部
RedrawWindow 重画全部或部分窗口
ReleaseCapture 为当前的应用程序释放鼠标捕获
第五页
控件与消息函数：共六页。第一页，第二页，第三页，第四页，第五页，第六页
ScreenToClient 判断屏幕上一个指定点的客户区坐标
ScrollWindow 滚动窗口客户区的全部或一部分
ScrollWindowEx 根据附加的选项，滚动窗口客户区的全部或部分
SetActiveWindow 激活指定的窗口
SetCapture 将鼠标捕获设置到指定的窗口
SetClassLong 为窗口类设置一个 Long 变量条目
SetClassWord 为窗口类设置一个条目
SetFocusAPI 将输入焦点设到指定的窗口。如有必要，会激活窗口
SetForegroundWindow 将窗口设为系统的前台窗口
SetParent 指定一个窗口的新父
SetRect 设置指定矩形的内容
SetRectEmpty 将矩形设为一个空矩形
SetWindowContextHelpId 为指定的窗口设置帮助场景（上下文）ID
SetWindowLong 在窗口结构中为指定的窗口设置信息
SetWindowPlacement 设置窗口状态和位置信息
SetWindowPos 为窗口指定一个新位置和状态
第六页
控件与消息函数：共六页。第一页，第二页，第三页，第四页，第五页，第六页
SetWindowText 设置窗口的标题文字或控件的内容
SetWindowWord 在窗口结构中为指定的窗口设置信息
ShowOwnedPopups 显示或隐藏由指定窗口所有的全部弹出式窗口
ShowWindow 控制窗口的可见性
ShowWindowAsync 与 ShowWindow 相似
SubtractRect 装载矩形 lprcDst，它是在矩形 lprcSrc1 中减去 lprcSrc2 得到的结果
TileWindows 以平铺顺序排列窗口
UnionRect 装载一个 lpDestRect 目标矩形，它是 lpSrc1Rect 和 lpSrc2Rect 联合起来的结果
UpdateWindow 强制立即更新窗口

ValidateRect 校验窗口的全部或部分客户区
WindowFromPoint 返回包含了指定点的窗口的句柄。忽略屏蔽、隐藏以及透明窗口
完
位图、图标和光栅运算函数
位图、图标和光栅运算函数，共三页。第一页，第二页，第三页
BitBlt 将一幅位图从一个设备场景复制到另一个
CopyIcon 制作指定图标或鼠标指针的一个副本。这个副本从属于发出调用的应用程序
CopyImage 复制位图、图标或指针，同时在复制过程中进行一些转换工作
CreateBitmap 按照规定的格式创建一幅与设备有关位图
CreateBitmapIndirect 创建一幅与设备有关位图
CreateCompatibleBitmap 创建一幅与设备有关位图，它与指定的设备场景兼容
CreateCursor 创建一个鼠标指针
CreateDIBitmap 根据一幅与设备无关的位图创建一幅与设备有关的位图
CreateDIBSection 创建一个 DIBSection
CreateIcon 创建一个图标
CreateIconIndirect 创建一个图标
DestroyCursor 清除指定的鼠标指针，并释放它占用的所有系统资源
DestroyIcon 清除图标
DrawIcon 在指定的位置画一个图标
DrawIconEx 描绘一个图标或鼠标指针。与 DrawIcon 相比，这个函数提供了更多的功能
ExtractAssociatedIcon 判断一个可执行程序或 DLL 中是否存在图标，或是否有图标与系统注册表中指定的文件存在关联并提取之
第一页
位图、图标和光栅运算函数，共三页。第一页，第二页，第三页
BitBlt 将一幅位图从一个设备场景复制到另一个
CopyIcon 制作指定图标或鼠标指针的一个副本。这个副本从属于发出调用的应用程序
CopyImage 复制位图、图标或指针，同时在复制过程中进行一些转换工作
CreateBitmap 按照规定的格式创建一幅与设备有关位图
CreateBitmapIndirect 创建一幅与设备有关位图
CreateCompatibleBitmap 创建一幅与设备有关位图，它与指定的设备场景兼容
CreateCursor 创建一个鼠标指针
CreateDIBitmap 根据一幅与设备无关的位图创建一幅与设备有关的位图
CreateDIBSection 创建一个 DIBSection
CreateIcon 创建一个图标
CreateIconIndirect 创建一个图标
DestroyCursor 清除指定的鼠标指针，并释放它占用的所有系统资源
DestroyIcon 清除图标
DrawIcon 在指定的位置画一个图标
DrawIconEx 描绘一个图标或鼠标指针。与 DrawIcon 相比，这个函数提供了更多的功能
ExtractAssociatedIcon 判断一个可执行程序或 DLL 中是否存在图标，或是否有图标与系统注册表中指定的文件存在关联并提取之
第二页
位图、图标和光栅运算函数，共三页。第一页，第二页，第三页
ExtractIcon 判断一个可执行文件或 DLL 中是否有图标存在，并将其提取出来

GetBitmapBits 将来自位图的二进制位复制到一个缓冲区
GetBitmapDimensionEx 取得一幅位图的宽度和高度
GetDIBColorTable 从选入设备场景的 DIBSection 中取得颜色表信息
GetDIBits 将来自一幅位图的二进制位复制到一幅与设备无关的位图里
GetIconInfo 取得与图标有关的信息
GetStretchBltMode 判断 StretchBlt 和 StretchDIBits 函数采用的伸缩模式
LoadBitmap 从指定的模块或应用程序实例中载入一幅位图
LoadCursor 从指定的模块或应用程序实例中载入一个鼠标指针
LoadCursorFromFile 在一个指针文件或一个动画指针文件的基础上创建一个指针
LoadIcon 从指定的模块或应用程序实例中载入一个图标
LoadImage 载入一个位图、图标或指针
MaskBlt 执行复杂的图象传输，同时进行掩模（MASK）处理
PatBlt 在当前选定的刷子的基础上，用一个图案填充指定的设备场景
PlgBlt 复制一幅位图，同时将其转换成一个平行四边形。利用它可对位图进行旋转处理
SetBitmapBits 将来自缓冲区的二进制位复制到一幅位图
第三页
位图、图标和光栅运算函数，共三页。第一页，第二页，第三页
SetBitmapDimensionEx 设置一幅位图的宽度。以一毫米的十分之一为单位
SetDIBColorTable 设置选入设备场景的一个 DIBSection 的颜色表信息
SetDIBits 将来自与设备无关位图的二进制位复制到一幅与设备有关的位图里
SetDIBitsToDevice 将一幅与设备无关位图的全部或部分数据直接复制到一个设备
SetStretchBltMode 指定 StretchBlt 和 StretchDIBits 函数的伸缩模式
StretchBlt 将一幅位图从一个设备场景复制到另一个
StretchDIBits 将一幅与设备无关位图的全部或部分数据直接复制到指定的设备场景
完
绘图函数
绘图函数：共七页。第一页，第二页，第三页，第四页，第五页，第六页，第七页
AbortPath 抛弃选入指定设备场景中的所有路径。也取消目前正在进行的任何路径的创建工作
AngleArc 用一个连接弧画一条线
Arc 画一个圆弧
BeginPath 启动一个路径分支
CancelDC 取消另一个线程里的长时间绘图操作
Chord 画一个弦
CloseEnhMetaFile 关闭指定的增强型图元文件设备场景，并将新建的图元文件返回一个句柄
CloseFigure 描绘到一个路径时，关闭当前打开的图形
CloseMetaFile 关闭指定的图元文件设备场景，并向新建的图元文件返回一个句柄
CopyEnhMetaFile 制作指定增强型图元文件的一个副本（拷贝）
CopyMetaFile 制作指定（标准）图元文件的一个副本
CreateBrushIndirect 在一个 LOGBRUSH 数据结构的基础上创建一个刷子
CreateDIBPatternBrush 用一幅与设备无关的位图创建一个刷子，以便指定刷子样式（图案）
CreateEnhMetaFile 创建一个增强型的图元文件设备场景
CreateHatchBrush 创建带有阴影图案的一个刷子
CreateMetaFile 创建一个图元文件设备场景

第一页
绘图函数：共七页。第一页，第二页，第三页，第四页，第五页，第六页，第七页
AbortPath 抛弃选入指定设备场景中的所有路径。也取消目前正在进行的任何路径的创建工作
AngleArc 用一个连接弧画一条线
Arc 画一个圆弧
BeginPath 启动一个路径分支
CancelDC 取消另一个线程里的长时间绘图操作
Chord 画一个弦
CloseEnhMetaFile 关闭指定的增强型图元文件设备场景，并将新建的图元文件返回一个句柄
CloseFigure 描绘到一个路径时，关闭当前打开的图形
CloseMetaFile 关闭指定的图元文件设备场景，并向新建的图元文件返回一个句柄
CopyEnhMetaFile 制作指定增强型图元文件的一个副本（拷贝）
CopyMetaFile 制作指定（标准）图元文件的一个副本
CreateBrushIndirect 在一个 LOGBRUSH 数据结构的基础上创建一个刷子
CreateDIBPatternBrush 用一幅与设备无关的位图创建一个刷子，以便指定刷子样式（图案）
CreateEnhMetaFile 创建一个增强型的图元文件设备场景
CreateHatchBrush 创建带有阴影图案的一个刷子
CreateMetaFile 创建一个图元文件设备场景
第二页
绘图函数：共七页。第一页，第二页，第三页，第四页，第五页，第六页，第七页
CreatePatternBrush 用指定了刷子图案的一幅位图创建一个刷子
CreatePen 用指定的样式、宽度和颜色创建一个画笔
CreatePenIndirect 根据指定的 LOGPEN 结构创建一个画笔
CreateSolidBrush 用纯色创建一个刷子
DeleteEnhMetaFile 删除指定的增强型图元文件
DeleteMetaFile 删除指定的图元文件
DeleteObject 删除 GDI 对象，对象使用的所有系统资源都会被释放
DrawEdge 用指定的样式描绘一个矩形的边框
DrawEscape 换码（Escape）函数将数据直接发至显示设备驱动程序
DrawFocusRect 画一个焦点矩形
DrawFrameControl 描绘一个标准控件
DrawState 为一幅图象或绘图操作应用各式各样的效果
Ellipse 描绘一个椭圆，由指定的矩形围绕
EndPath 停止定义一个路径
EnumEnhMetaFile 针对一个增强型图元文件，列举其中单独的图元文件记录
EnumMetaFile 为一个标准的 windows 图元文件枚举单独的图元文件记录
第三页
绘图函数：共七页。第一页，第二页，第三页，第四页，第五页，第六页，第七页
EnumObjects 枚举可随同指定设备场景使用的画笔和刷子
ExtCreatePen 创建一个扩展画笔（装饰或几何）
ExtFloodFill 在指定的设备场景里，用当前选择的刷子填充一个区域
FillPath 关闭路径中任何打开的图形，并用当前刷子填充
FillRect 用指定的刷子填充一个矩形

FlattenPath 将一个路径中的所有曲线都转换成线段
FloodFill 用当前选定的刷子在指定的设备场景中填充一个区域
FrameRect 用指定的刷子围绕一个矩形画一个边框
GdiComment 为指定的增强型图元文件设备场景添加一条注释信息
GdiFlush 执行任何未决的绘图操作
GdiGetBatchLimit 判断有多少个 GDI 绘图命令位于队列中
GdiSetBatchLimit 指定有多少个 GDI 绘图命令能够进入队列
GetArcDirection 画圆弧的时候，判断当前采用的绘图方向
GetBkColor 取得指定设备场景当前的背景颜色
GetBkMode 针对指定的设备场景，取得当前的背景填充模式
GetBrushOrgEx 判断指定设备场景中当前选定刷子起点
第四页
绘图函数：共七页。第一页，第二页，第三页，第四页，第五页，第六页，第七页
GetCurrentObject 获得指定类型的当前选定对象
GetCurrentPositionEx 在指定的设备场景中取得当前的画笔位置
GetEnhMetaFile 取得磁盘文件中包含的一个增强型图元文件的图元文件句柄
GetEnhMetaFileBits 将指定的增强型图元文件复制到一个内存缓冲区里
GetEnhMetaFileDescription 返回对一个增强型图元文件的说明
GetEnhMetaFileHeader 取得增强型图元文件的图元文件头
GetEnhMetaFilePaletteEntries 取得增强型图元文件的全部或部分调色板
GetMetaFile 取得包含在一个磁盘文件中的图元文件的图元文件句柄
GetMetaFileBitsEx 将指定的图元文件复制到一个内存缓冲区
GetMiterLimit 取得设备场景的斜率限制（Miter）设置
GetNearestColor 根据设备的显示能力，取得与指定颜色最接近的一种纯色
GetObjectAPI 取得对指定对象进行说明的一个结构
GetObjectType 判断由指定句柄引用的 GDI 对象的类型
GetPath 取得对当前路径进行定义的一系列数据
GetPixel 在指定的设备场景中取得一个像素的 RGB 值
GetPolyFillMode 针对指定的设备场景，获得多边形填充模式
第五页
绘图函数：共七页。第一页，第二页，第三页，第四页，第五页，第六页，第七页
GetROP2 针对指定的设备场景，取得当前的绘图模式
GetStockObject 取得一个固有对象（Stock）
GetSysColorBrush 为任何一种标准系统颜色取得一个刷子
GetWinMetaFileBits 通过在一个缓冲区中填充用于标准图元文件的数据，将一个增强型图元文件转换成标准 windows 图元文件
InvertRect 通过反转每个像素的值，从而反转一个设备场景中指定的矩形
LineDDA 枚举指定线段中的所有点
LineTo 用当前画笔画一条线，从当前位置连到一个指定的点
MoveToEx 为指定的设备场景指定一个新的当前画笔位置
PaintDesktop 在指定的设备场景中描绘桌面墙纸图案
PathToRegion 将当前选定的路径转换到一个区域里
Pie 画一个饼图
PlayEnhMetaFile 在指定的设备场景中画一个增强型图元文件

PlayEnhMetaFileRecord 回放单独一条增强型图元文件记录
PlayMetaFile 在指定的设备场景中回放一个图元文件
PlayMetaFileRecord 回放来自图元文件的单条记录
PolyBezier 描绘一条或多条贝塞尔（Bezier）曲线
第六页
绘图函数：共七页。第一页，第二页，第三页，第四页，第五页，第六页，第七页
PolyDraw 描绘一条复杂的曲线，由线段及贝塞尔曲线组成
Polygon 描绘一个多边形
Polyline 用当前画笔描绘一系列线段
PolyPolygon 用当前选定画笔描绘两个或多个多边形
PolyPolyline 用当前选定画笔描绘两个或多个多边形
Rectangle 用当前选定的画笔描绘矩形，并用当前选定的刷子填充
RoundRect 用当前选定的画笔画一个圆角矩形，并用当前选定的刷子在其中填充
SelectClipPath 将设备场景当前的路径合并到剪切区域里
SelectObject 为当前设备场景选择图形对象
SetArcDirection 设置圆弧的描绘方向
SetBkColor 为指定的设备场景设置背景颜色
SetBkMode 指定阴影刷子、虚线画笔以及字符中的空隙的填充方式
SetBrushOrgEx 为指定的设备场景设置当前选定刷子的起点
SetEnhMetaFileBits 用指定内存缓冲区内包含的数据创建一个增强型图元文件
SetMetaFileBitsEx 用包含在指定内存缓冲区内的数据结构创建一个图元文件
SetMiterLimit 设置设备场景当前的斜率限制
第七页
绘图函数：共七页。第一页，第二页，第三页，第四页，第五页，第六页，第七页
SetPixel 在指定的设备场景中设置一个像素的 RGB 值
SetPixelV 在指定的设备场景中设置一个像素的 RGB 值
SetPolyFillMode 设置多边形的填充模式
SetROP2 设置指定设备场景的绘图模式。与 vb 的 DrawMode 属性完全一致
SetWinMetaFileBits 将一个标准 Windows 图元文件转换成增强型图元文件
StrokeAndFillPath 针对指定的设备场景，关闭路径上打开的所有区域
StrokePath 用当前画笔描绘一个路径的轮廓。打开的图形不会被这个函数关闭
UnrealizeObject 将一个刷子对象选入设备场景之前，如刷子的起点准备用 SetBrushOrgEx 修改，则必须先调用本函数
WidenPath 根据选定画笔的宽度，重新定义当前选定的路径
完
打印函数
打印函数，共五页。第一页，第二页，第三页，第四页，第五页
AbortDoc 取消一份文档的打印
AbortPrinter 删除与一台打印机关联在一起的缓冲文件
AddForm 为打印机的表单列表添加一个新表单
AddJob 用于获取一个有效的路径名，以便用它为作业创建一个后台打印文件。它也会为作业分配一个作业编号
AddMonitor 为系统添加一个打印机监视器
AddPort 启动“添加端口”对话框，允许用户在系统可用端口列表中加入一个新端口

AddPrinter 在系统中添加一台新打印机
AddPrinterConnection 连接指定的打印机
AddPrinterDriver 为指定的系统添加一个打印驱动程序
AddPrintProcessor 为指定的系统添加一个打印处理器
AddPrintProvidor 为系统添加一个打印供应商
AdvancedDocumentProperties 启动打印机文档设置对话框
ClosePrinter 关闭一个打开的打印机对象
ConfigurePort 针对指定的端口，启动一个端口配置对话框
ConnectToPrinterDlg 启动连接打印机对话框，用它同访问网络的打印机连接
DeleteForm 从打印机可用表单列表中删除一个表单
第一页
打印函数，共五页。第一页，第二页，第三页，第四页，第五页
AbortDoc 取消一份文档的打印
AbortPrinter 删除与一台打印机关联在一起的缓冲文件
AddForm 为打印机的表单列表添加一个新表单
AddJob 用于获取一个有效的路径名，以便用它为作业创建一个后台打印文件。它也会为作业分配一个作业编号
AddMonitor 为系统添加一个打印机监视器
AddPort 启动“添加端口”对话框，允许用户在系统可用端口列表中加入一个新端口
AddPrinter 在系统中添加一台新打印机
AddPrinterConnection 连接指定的打印机
AddPrinterDriver 为指定的系统添加一个打印驱动程序
AddPrintProcessor 为指定的系统添加一个打印处理器
AddPrintProvidor 为系统添加一个打印供应商
AdvancedDocumentProperties 启动打印机文档设置对话框
ClosePrinter 关闭一个打开的打印机对象
ConfigurePort 针对指定的端口，启动一个端口配置对话框
ConnectToPrinterDlg 启动连接打印机对话框，用它同访问网络的打印机连接
DeleteForm 从打印机可用表单列表中删除一个表单
第二页
打印函数，共五页。第一页，第二页，第三页，第四页，第五页
DeleteMonitor 删除指定的打印监视器
DeletePort 启动“删除端口”对话框，允许用户从当前系统删除一个端口
DeletePrinter 将指定的打印机标志为从系统中删除
DeletePrinterConnection 删除与指定打印机的连接
DeletePrinterDriver 从系统删除一个打印机驱动程序
DeletePrintProcessor 从指定系统删除一个打印处理器
DeletePrintProvidor 从系统中删除一个打印供应商
DeviceCapabilities 利用这个函数可获得与一个设备的能力有关的信息
DocumentProperties 打印机配置控制函数
EndDocAPI 结束一个成功的打印作业
EndDocPrinter 在后台打印程序的级别指定一个文档的结束
EndPage 用这个函数完成一个页面的打印，并准备设备场景，以便打印下一个页
EndPagePrinter 指定一个页在打印作业中的结尾

EnumForms 枚举一台打印机可用的表单
EnumJobs 枚举打印队列中的作业
EnumMonitors 枚举可用的打印监视器
第三页
打印函数，共五页。第一页，第二页，第三页，第四页，第五页
EnumPorts 枚举一个系统可用的端口
EnumPrinterDrivers 枚举指定系统中已安装的打印机驱动程序
EnumPrinters 枚举系统中安装的打印机
EnumPrintProcessorDatatypes 枚举由一个打印处理器支持的数据类型
EnumPrintProcessors 枚举系统中可用的打印处理器
Escape 设备控制函数
FindClosePrinterChangeNotification 关闭用 FindFirstPrinterChangeNotification 函数获取的一个打印机通告对象
FindFirstPrinterChangeNotification 创建一个新的改变通告对象，以便我们注意打印机状态的各种变化
FindNextPrinterChangeNotification 用这个函数判断触发一次打印机改变通告信号的原因
FreePrinterNotifyInfo 释放由 FindNextPrinterChangeNotification 函数分配的一个缓冲区
GetForm 取得与指定表单有关的信息
GetJob 获取与指定作业有关的信息
GetPrinter 取得与指定打印机有关的信息
GetPrinterData 为打印机设置注册表配置信息
GetPrinterDriver 针对指定的打印机，获取与打印机驱动程序有关的信息
GetPrinterDriverDirectory 判断指定系统中包含了打印机驱动程序的目录是什么
第四页
打印函数，共五页。第一页，第二页，第三页，第四页，第五页
GetPrintProcessorDirectory 判断指定系统中包含了打印机处理器驱动程序及文件的目录
OpenPrinter 打开指定的打印机，并获取打印机的句柄
PrinterMessageBox 在拥有指定打印作业的系统上显示一个打印机出错消息框
PrinterProperties 启动打印机属性对话框，以便对打印机进行配置
ReadPrinter 从打印机读入数据
ResetDC 重设一个设备场景
ResetPrinter 改变指定打印机的默认数据类型及文档设置
ScheduleJob 提交一个要打印的作业
SetAbortProc 为 Windows 指定取消函数的地址
SetForm 为指定的表单设置信息
SetJob 对一个打印作业的状态进行控制
SetPrinter 对一台打印机的状态进行控制
SetPrinterData 设置打印机的注册表配置信息
StartDoc 开始一个打印作业
StartDocPrinter 在后台打印的级别启动一个新文档
StartPage 打印一个新页前要先调用这个函数
第五页
打印函数，共五页。第一页，第二页，第三页，第四页，第五页
StartPagePrinter 在打印作业中指定一个新页的开始

WritePrinter 将发送目录中的数据写入打印机
完
设备场景函数
设备场景函数，共五页。第一页，第二页，第三页，第四页，第五页
CombineRgn 将两个区域组合为一个新区域
CombineTransform 驱动世界转换。它相当于依顺序进行两次转换
CreateCompatibleDC 创建一个与特定设备场景一致的内存设备场景
CreateDC 为专门设备创建设备场景
CreateEllipticRgn 创建一个椭圆
CreateEllipticRgnIndirect 创建一个内切于特定矩形的椭圆区域
CreateIC 为专用设备创建一个信息场景
CreatePolygonRgn 创建一个由一系列点围成的区域
CreatePolyPolygonRgn 创建由多个多边形构成的区域。每个多边形都应是封闭的
CreateRectRgn 创建一个矩形区域
CreateRectRgnIndirect 创建一个矩形区域
CreateRoundRectRgn 创建一个圆角矩形
DeleteDC 删除专用设备场景或信息场景，释放所有相关窗口资源
DPtoLP 将点阵从设备坐标转换到专用设备场景逻辑坐标
EqualRgn 确定两个区域是否相等
ExcludeClipRect 从专用设备场景的剪裁区中去掉一个矩形区。矩形内不能进行绘图
第一页
设备场景函数，共五页。第一页，第二页，第三页，第四页，第五页
CombineRgn 将两个区域组合为一个新区域
CombineTransform 驱动世界转换。它相当于依顺序进行两次转换
CreateCompatibleDC 创建一个与特定设备场景一致的内存设备场景
CreateDC 为专门设备创建设备场景
CreateEllipticRgn 创建一个椭圆
CreateEllipticRgnIndirect 创建一个内切于特定矩形的椭圆区域
CreateIC 为专用设备创建一个信息场景
CreatePolygonRgn 创建一个由一系列点围成的区域
CreatePolyPolygonRgn 创建由多个多边形构成的区域。每个多边形都应是封闭的
CreateRectRgn 创建一个矩形区域
CreateRectRgnIndirect 创建一个矩形区域
CreateRoundRectRgn 创建一个圆角矩形
DeleteDC 删除专用设备场景或信息场景，释放所有相关窗口资源
DPtoLP 将点阵从设备坐标转换到专用设备场景逻辑坐标
EqualRgn 确定两个区域是否相等
ExcludeClipRect 从专用设备场景的剪裁区中去掉一个矩形区。矩形内不能进行绘图
第二页
设备场景函数，共五页。第一页，第二页，第三页，第四页，第五页
ExcludeUpdateRgn 从专用设备场景剪裁区去掉指定窗口的刷新区域
ExtCreateRegion 根据世界转换修改区域
ExtSelectClipRgn 将指定区域组合到设备场景的当前剪裁区
FillRgn 用指定刷子填充指定区域

FrameRgn 用指定刷子围绕指定区域画一个外框
GetBoundsRect 获取指定设备场景的边界矩形
GetClipBox 获取完全包含指定设备场景剪裁区的最小矩形
GetClipRgn 获取设备场景当前剪裁区
GetDC 获取指定窗口的设备场景
GetDCEx 为指定窗口获取设备场景。相比 GetDC，本函数提供了更多的选项
GetDCOrgEx 获取指定设备场景起点位置（以屏幕坐标表示）
GetDeviceCaps 根据指定设备场景代表的设备的功能返回信息
GetGraphicsMode 确定是否允许增强图形模式（世界转换）
GetMapMode 为特定设备场景调入映象模式
GetRegionData 装入描述一个区域信息的 RgnData 结构或缓冲区
GetRgnBox 获取完全包含指定区域的最小矩形
第三页
设备场景函数，共五页。第一页，第二页，第三页，第四页，第五页
GetUpdateRgn 确定指定窗口的刷新区域。该区域当前无效，需要刷新
GetViewportExtEx 获取设备场景视口（viewport）范围
GetViewportOrgEx 获取设备场景视口起点
GetWindowDC 获取整个窗口（包括边框、滚动条、标题栏、菜单等）的设备场景
GetWindowExtEx 获取指定设备场景的窗口范围
GetWindowOrgEx 获取指定设备场景的逻辑窗口的起点
GetWindowRgn 获取窗口区域
GetWorldTransform 如果有世界转换，为设备场景获取当前世界转换
IntersectClipRect 为指定设备定义一个新的剪裁区
InvalidateRgn 使窗口指定区域不活动，并将它加入窗口刷新区，使之可随后被重画
InvertRgn 通过颠倒每个像素值反转设备场景指定区域
LPtoDP 将点阵从指定设备场景逻辑坐标转换为设备坐标
ModifyWorldTransform 根据指定的模式修改世界转换
OffsetClipRgn 按指定量平移设备场景剪裁区
OffsetRgn 按指定偏移量平移指定区域
OffsetViewportOrgEx 平移设备场景视口区域
第四页
设备场景函数，共五页。第一页，第二页，第三页，第四页，第五页
OffsetWindowOrgEx 平移指定设备场景窗口起点
PaintRgn 用当前刷子背景色填充指定区域
PtInRegion 确定点是否在指定区域内
PtVisible 确定指定点是否可见（即，点是否在设备场景剪裁区内）
RectInRegion 确定矩形是否有部分在指定区域内
RectVisible 确定指定矩形是否有部分可见（是否在设备场景剪裁区内）
ReleaseDC 释放由调用 GetDC 或 GetWindowDC 函数获取的指定设备场景
RestoreDC 从设备场景堆栈恢复一个原先保存的设备场景
SaveDC 将指定设备场景状态保存到 Windows 设备场景堆栈
ScaleViewportExtEx 缩放设备场景视口的范围
ScaleWindowExtEx 缩放指定设备场景窗口范围
ScrollDC 在窗口（由设备场景代表）中水平和（或）垂直滚动矩形

SelectClipRgn 为指定设备场景选择新的剪裁区
SetBoundsRect 设置指定设备场景的边界矩形
SetGraphicsMode 允许或禁止增强图形模式，以提供某些支持（包括世界转换）
SetMapMode 设置指定设备场景的映射模式
第五页
设备场景函数，共五页。第一页，第二页，第三页，第四页，第五页
SetRectRgn 设置区域为指定的矩形
SetViewportExtEx 设置设备场景视口范围
SetViewportOrgEx 设置设备场景视口起点
SetWindowExtEx 设置指定设备场景窗口范围
SetWindowOrgEx 设置指定设备场景窗口起点
SetWindowRgn 设置窗口区域
SetWorldTransform 设置世界转换
ValidateRgn 激活窗口中指定区域，把它从刷新区移走
WindowFromDC 取回与某一设备场景相关的窗口的句柄
完
进程和线程函数
进程与线程函数，共五页。第一页，第二页，第三页，第四页，第五页
CancelWaitableTimer 这个函数用于取消一个可以等待下去的计时器操作
CallNamedPipe 这个函数由一个希望通过管道通信的一个客户进程调用
ConnectNamedPipe 指示一台服务器等待下去，直至客户机同一个命名管道连接
CreateEvent 创建一个事件对象
CreateMailslot 创建一个邮路。返回的句柄由邮路服务器使用（收件人）
CreateMutex 创建一个互斥体（MUTEX）
CreateNamedPipe 创建一个命名管道。返回的句柄由管道的服务器端使用
CreatePipe 创建一个匿名管道
CreateProcess 创建一个新进程（比如执行一个程序）
CreateSemaphore 创建一个新的信号机
CreateWaitableTimer 创建一个可等待的计时器对象
DisconnectNamedPipe 断开一个客户与一个命名管道的连接
DuplicateHandle 在指出一个现有系统对象当前句柄的情况下，为那个对象创建一个新句柄
ExitProcess 中止一个进程
FindCloseChangeNotification 关闭一个改动通知对象
FindExecutable 查找与一个指定文件关联在一起的程序的文件名
第一页
进程与线程函数，共五页。第一页，第二页，第三页，第四页，第五页
CancelWaitableTimer 这个函数用于取消一个可以等待下去的计时器操作
CallNamedPipe 这个函数由一个希望通过管道通信的一个客户进程调用
ConnectNamedPipe 指示一台服务器等待下去，直至客户机同一个命名管道连接
CreateEvent 创建一个事件对象
CreateMailslot 创建一个邮路。返回的句柄由邮路服务器使用（收件人）
CreateMutex 创建一个互斥体（MUTEX）
CreateNamedPipe 创建一个命名管道。返回的句柄由管道的服务器端使用
CreatePipe 创建一个匿名管道

CreateProcess 创建一个新进程（比如执行一个程序）
CreateSemaphore 创建一个新的信号机
CreateWaitableTimer 创建一个可等待的计时器对象
DisconnectNamedPipe 断开一个客户与一个命名管道的连接
DuplicateHandle 在指出一个现有系统对象当前句柄的情况下，为那个对象创建一个新句柄
ExitProcess 中止一个进程
FindCloseChangeNotification 关闭一个改动通知对象
FindExecutable 查找与一个指定文件关联在一起的程序的文件名

FindFirstChangeNotification 创建一个文件通知对象。该对象用于监视文件系统发生的变化
FindNextChangeNotification 重设一个文件改变通知对象，令其继续监视下一次变化
FreeLibrary 释放指定的动态链接库
GetCurrentProcess 获取当前进程的一个伪句柄
GetCurrentProcessId 获取当前进程一个唯一的标识符
GetCurrentThread 获取当前线程的一个伪句柄
GetCurrentThreadId 获取当前线程一个唯一的线程标识符
GetExitCodeProces 获取一个已中断进程的退出代码
GetExitCodeThread 获取一个已中止线程的退出代码
GetHandleInformation 获取与一个系统对象句柄有关的信息
GetMailslotInfo 获取与一个邮路有关的信息
GetModuleFileName 获取一个已装载模板的完整路径名称
GetModuleHandle 获取一个应用程序或动态链接库的模块句柄
GetPriorityClass 获取特定进程的优先级别
GetProcessShutdownParameters 调查系统关闭时一个指定的进程相对于其它进程的关闭早迟情况
GetProcessTimes 获取与一个进程的经过时间有关的信息

GetProcessWorkingSetSize 了解一个应用程序在运行过程中实际向它交付了多大容量的内存
GetSartupInfo 获取一个进程的启动信息
GetThreadPriority 获取特定线程的优先级别
GetTheardTimes 获取与一个线程的经过时间有关的信息
GetWindowThreadProcessId 获取与指定窗口关联在一起的一个进程和线程标识符
LoadLibrary 载入指定的动态链接库，并将它映射到当前进程使用的地址空间
LoadLibraryEx 装载指定的动态链接库，并为当前进程把它映射到地址空间
LoadModule 载入一个 Windows 应用程序，并在指定的环境中运行
MsgWaitForMultipleObjects 等侯单个对象或一系列对象发出信号。如返回条件已经满足，则立即返回
SetPriorityClass 设置一个进程的优先级别
SetProcessShutdownParameters 在系统关闭期间，为指定进程设置他相对于其它程序的关闭顺序
SetProcessWorkingSetSize 设置操作系统实际划分给进程使用的内存容量
SetThreadPriority 设定线程的优先级别

ShellExecute 查找与指定文件关联在一起的程序的文件名
TerminateProcess 结束一个进程
WinExec 运行指定的程序
第四页
进程与线程函数，共五页。第一页，第二页，第三页，第四页，第五页
第五页
进程与线程函数，共五页。第一页，第二页，第三页，第四页，第五页
Windows 消息函数
Windows 消息函数，共一页。第一页
BroadcastSystemMessage 将一条系统消息广播给系统中所有的顶级窗口
GetMessagePos 取得消息队列中上一条消息处理完毕时的鼠标指针屏幕位置
GetMessageTime 取得消息队列中上一条消息处理完毕时的时间
PostMessage 将一条消息投递到指定窗口的消息队列
PostThreadMessage 将一条消息投递给应用程序
RegisterWindowMessage 获取分配给一个字串标识符的消息编号
ReplyMessage 答复一个消息
SendMessage 调用一个窗口的窗口函数，将一条消息发给那个窗口
SendMessageCallback 将一条消息发给窗口
SendMessageTimeout 向窗口发送一条消息
SendNotifyMessage 向窗口发送一条消息
完
BroadcastSystemMessage
BroadcastSystemMessage

VB 声明
Declare Function BroadcastSystemMessage Lib "user32" Alias "BroadcastSystemMessage" (ByVal dw As Long, pdw As Long, ByVal un As Long, ByVal wParam As Long, ByVal lParam As Long) As Long
说明
将一条系统消息广播给系统中所有的顶级窗口
返回值
Long，大于零表示成功；-1 表示出错。如设置了 BSF_QUERY，而且至少有一个消息接收者返回零，那么这个函数返回零
参数表
参数类型及说明
dwLong，下述常数的一个或多个
BSF_FLUSHDISK 每次处理完一条消息后，都对磁盘进行刷新（将未存盘的数据存下来
BSF_FORCEIFHUNG 如目标处于挂起状态，则在设定的超时后到期返回
BSF_IGNORECURRENTTASK 发送任务不接收消息
BSF_LPARAMBUFFERlParam 指向一个内存缓冲区
BSF_NOHANG 跳过被挂起的所有进程
BSF_POSTMESSAGE 投递消息。不与 BSF_LPARAMBUFFER 和 BSF_QUERY 兼容
BSF_QUERY 将消息顺序发给进程，只有前一个返回 TRUE 时，才进入下一个进程
pdwLong，下述常数的一个或多个

BSF_ALLCOMPONENTS 消息进入能够接收消息的每一个系统组件
BSF_APPLICATIONS 消息到达应用程序
BSF_INSTALLABLEDRIVERS 消息到达可安装的驱动程序
BSF_NETDRIVERS 消息到达网络驱动程序
BSF_VXDS 消息到达系统设备驱动程序
unLong，消息编号
wParamLong，由消息决定
lParamLong，由消息决定。如指定了 BSF_LPARAMBUFFER，这就是位于调用进程地址空间的一个内存缓冲区的地址，而且缓冲区的第一个 16 位字包含了缓冲区的长度

Top
GetMessagePos
GetMessagePos

VB 声明
Declare Function GetMessagePos Lib "user32" Alias "GetMessagePos" () As Long
说明
取得消息队列中上一条消息处理完毕时的鼠标指针屏幕位置
返回值
Long，X 坐标对应于结果值的低字，Y 坐标对应于高字

Top
GetMessageTime
GetMessageTime

VB 声明
Declare Function GetMessageTime Lib "user32" Alias "GetMessageTime" () As Long
说明
取得消息队列中上一条消息处理完毕时的时间
返回值
Long，返回一个时间，表示为自系统启动以来经历的毫秒数
原文：The time is specified in milliseconds from the time the system was started.

Top
PostMessage
PostMessage, PostMessageBynum, PostMessageBystring

VB 声明
Declare Function PostMessage& Lib "user32" Alias "PostMessageA" (ByVal hwnd As Long, ByVal wMsg As Long, ByVal wParam As Long, lParam As Any)
Declare Function PostMessageByNum& Lib "user32" Alias "PostMessageA" (ByVal hwnd As Long, ByVal wMsg As Long, ByVal wParam As Long, ByVal lParam As Long)
Declare Function PostMessageByString& Lib "user32" Alias "PostMessageA" (ByVal hwnd As Long, ByVal wMsg As Long, ByVal wParam As Long, ByVal lParam As String)

说明
将一条消息投递到指定窗口的消息队列。投递的消息会在 Windows 事件处理过程中得到处理。在那个时候，会随同投递的消息调用指定窗口的窗口函数。特别适合那些不需要立即处理的窗口消息的发送
返回值
Long，如消息投递成功，则返回 TRUE（非零）。会设置 GetLastError
参数表
参数类型及说明
hwndLong，接收消息的那个窗口的句柄。如设为 HWND_BROADCAST，表示投递给系统中的所有顶级窗口。如设为零，表示投递一条线程消息（参考 PostThreadMessage）
wMsgLong，消息标识符
wParamLong，具体由消息决定
lParamAny，具体由消息决定

Top
PostThreadMessage
PostThreadMessage

VB 声明
Declare Function PostThreadMessage Lib "user32" Alias "PostThreadMessageA" (ByVal idThread As Long, ByVal msg As Long, ByVal wParam As Long, ByVal lParam As Long) As Long
说明
将一条消息投递给应用程序。这条消息由应用程序的内部 GetMessage 循环获得，但不会传给一个特定的窗口
返回值
Long，如消息投递成功，则返回 TRUE（非零）。会设置 GetLastError
参数表
参数类型及说明
idThreadLong，用于接收消息的那个线程的标识符
msgLong，消息标识符
wParamLong，具体由消息决定
ByValLong，具体由消息决定

Top
RegisterWindowMessage
RegisterWindowMessage

VB 声明
Declare Function RegisterWindowMessage Lib "user32" Alias "RegisterWindowMessageA" (ByVal lpString As String) As Long
说明
获取分配给一个字串标识符的消息编号
返回值

Long，&C000 到 &FFFF 之间的一个消息编号。零意味着出错
参数表
参数类型及说明
lpStringString，注册消息的名字
注解
如果没有一个子类处理程序的帮助，这个函数就没有什么用

Top
ReplyMessage
ReplyMessage

VB 声明
Declare Function ReplyMessage Lib "user32" Alias "ReplyMessage" (ByVal lReply As Long) As Long
说明
如将消息传送给位于不同进程的一个窗口，通常第一个进程会暂时挂起，直到另一个进程中的窗口函数完成操作为止。在目标进程的窗口函数完成之前，另一个进程可用这个函数向第一个进程返回一个结果，使之能继续进行
返回值
Long，如准备答复的消息是由另一个进程发来的，则返回 TRUE。如果它是从同一个进程中发出来的，则返回 FALSE（此时，该函数没有任何效果）
参数表
参数类型及说明
lReplyLong，指定发回调用进程的一个结果

Top
SendMessage
SendMessage, SendMessageBynum, SendMessageByString

VB 声明
Declare Function SendMessage& Lib "user32" Alias "SendMessageA" (ByVal hwnd As Long, ByVal wMsg As Long, ByVal wParam As Long, lParam As Any)
Declare Function SendMessageBynum& Lib "user32" Alias "SendMessageA" (ByVal hwnd As Long, ByVal wMsg As Long, ByVal wParam As Long, ByVal lParam As Long)
Declare Function SendMessageByString& Lib "user32" Alias "SendMessageA" (ByVal hwnd As Long, ByVal wMsg As Long, ByVal wParam As Long, ByVal lParam As String)
说明
调用一个窗口的窗口函数，将一条消息发给那个窗口。除非消息处理完毕，否则该函数不会返回。SendMessageBynum， SendMessageByString 是该函数的“类型安全”声明形式
返回值
Long，由具体的消息决定
参数表
参数类型及说明
hwndLong，要接收消息的那个窗口的句柄

wMsgLong，消息的标识符
wParamLong，具体取决于消息
lParamAny，具体取决于消息

Top
SendMessageCallback
SendMessageCallback

VB 声明
Declare Function SendMessageCallback Lib "user32" Alias "SendMessageCallbackA" (ByVal hwnd As Long, ByVal msg As Long, ByVal wParam As Long, ByVal lParam As Long, ByVal lpResultCallBack As Long, ByVal dwData As Long) As Long
说明
将一条消息发给窗口。该函数最大的特定是可以立即返回。目标窗口函数执行完毕后，会用回调函数的形式将结果返回
返回值
Long，TRUE 表示成功，FALSE 表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，要接收消息的那个窗口的句柄
msgLong，消息的标识符
wParamLong，取决于消息
lParamLong，取决于消息
lpResultCallBackLong，指定函数地址。在 vb5 中可用 AddressOf 操作符获得
dwDataLong，用户自定义值
注解
回调函数声明如下：
Public Function WndProc(ByVal hwnd&, ByVal msg&, ByVal wp&, ByVal lp&) As Long
其中，wp 参数是作为 dwData 参数传递的值。lp 参数包含了来自窗口函数的结果

Top
SendMessageTimeout
SendMessageTimeout

VB 声明
Declare Function SendMessageTimeout Lib "user32" Alias "SendMessageTimeoutA" (ByVal hwnd As Long, ByVal msg As Long, ByVal wParam As Long, ByVal lParam As Long, ByVal fuFlags As Long, ByVal uTimeout As Long, lpdwResult As Long) As Long
说明
向窗口发送一条消息。如窗口位于不同的线程中，则利用这个函数可以指定一个超时值，以便在另一个进程挂起的时候防止调用进程也永远挂起
返回值
Long，成功时返回 TRUE，失败时返回 FALSE。会设置 GetLastError
参数表

参数类型及说明
hwndLong，要接收消息的一个窗口的句柄
msgLong，消息的标识符
wParamLong，由消息决定
lParamLong，由消息决定
fuFlagsLong，下述常数的一个或多个
SMTO_ABORTIFHUNG 如目标进程挂起，则函数立即返回
SMTO_BLOCK 除非函数返回，否则调用线程不能处理消息
SMTO_NORMAL 允许调用线程处理消息，同时保持函数继续执行
uTimeoutLong，超时值，采用毫秒为单位
lpdwResultLong，用于装载函数结果的一个变量

Top
SendNotifyMessage
SendNotifyMessage

VB 声明
Declare Function SendNotifyMessage Lib "user32" Alias "SendNotifyMessageA" (ByVal hwnd As Long, ByVal msg As Long, ByVal wParam As Long, ByVal lParam As Long) As Long
说明
向窗口发送一条消息。如目标窗口位于同调用方相同的线程内，则这个函数会表现为 SendMessage 函数。而且除非消息得到处理，否则函数不会返回。如目标窗口从属于一个不同的线程，则函数会立即返回
返回值
Long，TRUE 表示成功，FALSE 表示出错。会设置 GetLastError
参数表
参数类型及说明
hwndLong，用于接收消息的一个窗口的句柄
msgLong，消息的标识符
wParamLong，具体由消息决定
lParamLong，具体由消息决定

Top
文件处理函数
文件处理函数，共八页。第一页，第二页，第三页，第四页，第五页，第六页，第七页，第八页
CloseHandle 关闭一个内核对象。其中包括文件、文件映射、进程、线程、安全和同步对象等
CompareFileTime 对比两个文件的时间
CopyFile 复制文件
CreateDirectory 创建一个新目录
CreateFile 打开和创建文件、管道、邮槽、通信服务、设备以及控制台
CreateFileMapping 创建一个新的文件映射对象
DeleteFile 删除指定文件

DeviceIoControl 对设备执行指定的操作
DosDateTimeToFileTime 将 DOS 日期和时间值转换成一个 win32 FILETIME 值
FileTimeToDosDateTime 将一个 win32 FILETIME 值转换成 DOS 日期和时间值
FileTimeToLocalFileTime 将一个 FILETIME 结构转换成本地时间
FileTimeToSystemTime 根据一个 FILETIME 结构的内容，装载一个 SYSTEMTIME 结构
FindClose 关闭由 FindFirstFile 函数创建的一个搜索句柄
FindFirstFile 根据文件名查找文件
FindNextFile 根据调用 FindFirstFile 函数时指定的一个文件名查找下一个文件
FlushFileBuffers 针对指定的文件句柄，刷新内部文件缓冲区
CloseHandle
CloseHandle

VB 声明
Declare Function CloseHandle Lib "kernel32" Alias "CloseHandle" (ByVal hObject As Long) As Long
说明
关闭一个内核对象。其中包括文件、文件映射、进程、线程、安全和同步对象等。涉及文件处理时，这个函数通常与 vb 的 close 命令相似。应尽可能的使用 close，因为它支持 vb 的差错控制。注意这个函数使用的文件句柄与 vb 的文件编号是完全不同的
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hObjectLong，欲关闭的一个对象的句柄
注解
除非对内核对象的所有引用都已关闭，否则该对象不会实际删除

Top
CompareFileTime
CompareFileTime

VB 声明
Declare Function CompareFileTime Lib "kernel32" Alias "CompareFileTime" (lpFileTime1 As FILETIME, lpFileTime2 As FILETIME) As Long
说明
根据 FILETIME 结构的信息，对比两个文件的时间
返回值
Long，如两个时间相等，就返回零；如 lpFileTime1 小于 lpFileTime2，返回-1；如 lpFileTime2 小于 lpFileTime1，返回 1
参数表
参数类型及说明
lpFileTime1FILETIME，参考 FILETIME
lpFileTime2

Top
CopyFile
CopyFile

VB 声明
Declare Function CopyFile Lib "kernel32" Alias "CopyFileA" (ByVal lpExistingFileName As String, ByVal lpNewFileName As String, ByVal bFailIfExists As Long) As Long
说明
复制文件。与 vb 的 filecopy 命令相似
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpExistingFileNameString，源文件名
lpNewFileNameString，目标文件名
bFailIfExistsLong，如果设为 TRUE（非零），那么一旦目标文件已经存在，则函数调用会失败。否则目标文件被改写

Top
CreateDirectory
CreateDirectory, CreateDirectoryEx

VB 声明
Declare Function CreateDirectory& Lib "kernel32" Alias "CreateDirectoryA" (ByVal lpNewDirectory As String, lpSecurityAttributes As SECURITY_ATTRIBUTES)
Declare Function CreateDirectoryEx& Lib "kernel32" Alias "CreateDirectoryExA" (ByVal lpTemplateDirectory As String, ByVal lpNewDirectory As String, lpSecurityAttributes As SECURITY_ATTRIBUTES)
说明
创建一个新目录
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpTemplateDirectoryString，指定一个模板目录的名字，从中复制默认属性（比如目录中文件的默认压缩方式）。如设为 vbNullString，则表示不使用模板
lpNewDirectoryString，新目录的名字
lpSecurityAttributesSECURITY_ATTRIBUTES，这个结构定义了目录的安全特性——如果操作系统支持的话

Top
CreateFile
CreateFile

VB 声明
Declare Function CreateFile Lib "kernel32" Alias "CreateFileA" (ByVal lpFileName As String, ByVal dwDesiredAccess As Long, ByVal dwShareMode As Long, lpSecurityAttributes As SECURITY_ATTRIBUTES, ByVal dwCreationDisposition As Long, ByVal dwFlagsAndAttributes As Long, ByVal hTemplateFile As Long) As Long
说明
这是一个全功能的例程，可打开和创建文件、管道、邮槽、通信服务、设备以及控制台
返回值
Long，如执行成功，则返回文件句柄。INVALID_HANDLE_VALUE 表示出错，会设置 GetLastError。即使函数成功，但若文件存在，且指定了 CREATE_ALWAYS 或 OPEN_ALWAYS，GetLastError 也会设为 ERROR_ALREADY_EXISTS
参数表
参数类型及说明
lpFileNameString，要打开的文件的名字
dwDesiredAccessLong，如果为 GENERIC_READ 表示允许对设备进行读访问；如果为 GENERIC_WRITE 表示允许对设备进行写访问（可组合使用）；如果为零，表示只允许获取与一个设备有关的信息
dwShareModeLong，零表示不共享； FILE_SHARE_READ 和/或 FILE_SHARE_WRITE 表示允许对文件进行共享访问
lpSecurityAttributesSECURITY_ATTRIBUTES，指向一个 SECURITY_ATTRIBUTES 结构的指针，定义了文件的安全特性（如果操作系统支持的话）
dwCreationDispositionLong，下述常数之一：
CREATE_NEW 创建文件；如文件存在则会出错
CREATE_ALWAYS 创建文件，会改写前一个文件
OPEN_EXISTING 文件必须已经存在。由设备提出要求
OPEN_ALWAYS 如文件不存在则创建它
TRUNCATE_EXISTING 讲现有文件缩短为零长度
dwFlagsAndAttributesLong，一个或多个下述常数
FILE_ATTRIBUTE_ARCHIVE 标记归档属性
FILE_ATTRIBUTE_COMPRESSED 将文件标记为已压缩，或者标记为文件在目录中的默认压缩方式
FILE_ATTRIBUTE_NORMAL 默认属性
FILE_ATTRIBUTE_HIDDEN 隐藏文件或目录
FILE_ATTRIBUTE_READONLY 文件为只读
FILE_ATTRIBUTE_SYSTEM 文件为系统文件
FILE_FLAG_WRITE_THROUGH 操作系统不得推迟对文件的写操作
FILE_FLAG_OVERLAPPED 允许对文件进行重叠操作
FILE_FLAG_NO_BUFFERING 禁止对文件进行缓冲处理。文件只能写入磁盘卷的扇区块
FILE_FLAG_RANDOM_ACCESS 针对随机访问对文件缓冲进行优化
FILE_FLAG_SEQUENTIAL_SCAN 针对连续访问对文件缓冲进行优化
FILE_FLAG_DELETE_ON_CLOSE 关闭了上一次打开的句柄后，将文件删除。特别适合临时文件
也可在 Windows NT 下组合使用下述常数标记：
SECURITY_ANONYMOUS ， SECURITY_IDENTIFICATION ，

SECURITY_IMPERSONATION ， SECURITY_DELEGATION ， SECURITY_CONTEXT_TRACKING， SECURITY_EFFECTIVE_ONLY
hTemplateFileLong，如果不为零，则指定一个文件句柄。新文件将从这个文件中复制扩展属性
注解
打开一个通信端口时（如 COM1），无论如何都要设置成 OPEN_EXISTING
这个函数代替了 lOpen 和 lCreate 函数，应该是我们的首选

Top
CreateFileMapping
CreateFileMapping

VB 声明
Declare Function CreateFileMapping Lib "kernel32" Alias "CreateFileMappingA" (ByVal hFile As Long, lpFileMappigAttributes As SECURITY_ATTRIBUTES, ByVal flProtect As Long, ByVal dwMaximumSizeHigh As Long, ByVal dwMaximumSizeLow As Long, ByVal lpName As String) As Long
说明
创建一个新的文件映射对象
返回值
Long，新建文件映射对象的句柄；零意味着出错。会设置 GetLastError。即使函数成功，但倘若返回的句柄属于一个现成的文件映射对象，那么 GetLastError 也会设置成 ERROR_ALREADY_EXISTS。在这种情况下，文件映射的长度就是现有对象的长度，而不是这个函数指定的尺寸
参数表
参数类型及说明
hFileLong，指定欲在其中创建映射的一个文件句柄。&HFFFFFFFF&表示在内存中创建一个文件映射
lpFileMappigAttributesSECURITY_ATTRIBUTES，指定一个安全对象，在创建文件映射时使用。如果为 NULL（用 ByVal As Long 传递零），表示使用默认安全对象
flProtectLong，下述常数之一：
PAGE_READONLY 以只读方式打开映射
PAGE_READWRITE 以可读、可写方式打开映射
PAGE_WRITECOPY 为写操作留下备份
可组合使用下述一个或多个常数
SEC_COMMIT 为文件映射一个小节中的所有页分配内存
SEC_IMAGE 文件是个可执行文件
SEC_RESERVE 为没有分配实际内存的一个小节保留虚拟内存空间
dwMaximumSizeHighLong，文件映射的最大长度（高 32 位）
dwMaximumSizeLowLong，文件映射的最小长度（低 32 位）。如这个参数和 dwMaximumSizeHigh 都是零，就用磁盘文件的实际长度
lpNameString，指定文件映射对象的名字。如存在这个名字的一个映射，函数就会打开它。用 vbNullString 创建一个无名的文件映射

Top
DeleteFile
DeleteFile

VB 声明
Declare Function DeleteFile Lib "kernel32" Alias "DeleteFileA" (ByVal lpFileName As String) As Long
说明
删除指定文件
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpFileNameString，欲删除文件的名字
注解
与 vb 的 kill 语句相似，在 windows 95 下使用这个函数要小心——即使文件当前正由一个应用程序打开，该函数也会将其删除

Top
DeviceIoControl
DeviceIoControl

VB 声明
Declare Function DeviceIoControl Lib "kernel32" Alias "DeviceIoControl" (ByVal hDevice As Long, ByVal dwIoControlCode As Long, lpInBuffer As Any, ByVal nInBufferSize As Long, lpOutBuffer As Any, ByVal nOutBufferSize As Long, lpBytesReturned As Long, lpOverlapped As OVERLAPPED) As Long
说明
对设备执行指定的操作
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hDeviceLong，设备句柄
dwIoControlCodeLong，带有 FSCTL_ 前缀的常数。参考设备控制选项的部分列表
lpInBufferAny，具体取决于 dwIoControlCode 参数。参考设备控制选项的部分列表
nInBufferSizeLong，输入缓冲区的长度
lpOutBufferAny，具体取决于 dwIoControlCode 参数。参考设备控制选项的部分列表
nOutBufferSizeLong，输出缓冲区的长度
lpBytesReturnedLong，实际装载到输出缓冲区的字节数量
lpOverlappedOVERLAPPED，这个结构用于重叠操作。针对同步操作，请用 ByVal As Long 传递零值
注解
可用于 windows 95 和 windows nt，但并非所有的操作都得到了两种操作系统的同时支持

Top
DosDateTimeToFileTime
DosDateTimeToFileTime

VB 声明
Declare Function DosDateTimeToFileTime Lib "kernel32" Alias "DosDateTimeToFileTime" (ByVal wFatDate As Long, ByVal wFatTime As Long, lpFileTime As FILETIME) As Long
说明
将 DOS 日期和时间值转换成一个 win32 FILETIME 值
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
wFatDateLong，16 位 DOS 日期值
wFatTimeLong，16 位 DOS 时间值
lpFileTimeFILETIME，用于装载 win32 时间的一个结构

Top
FileTimeToDosDateTime
FileTimeToDosDateTime

VB 声明
Declare Function FileTimeToDosDateTime Lib "kernel32" Alias "FileTimeToDosDateTime" (lpFileTime As FILETIME, ByVal lpFatDate As Long, ByVal lpFatTime As Long) As Long
说明
将一个 win32 FILETIME 值转换成 DOS 日期和时间值
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpFileTimeFILETIME，包含了欲转换时间的一个结构
lpFatDateLong，16 位 DOS 日期值
lpFatTimeLong，16 位 DOS 时间值

Top
FileTimeToLocalFileTime
FileTimeToLocalFileTime

VB 声明
Declare Function FileTimeToLocalFileTime Lib "kernel32" Alias "FileTimeToLocalFileTime" (lpFileTime As FILETIME, lpLocalFileTime As FILETIME) As Long
说明
将一个 FILETIME 结构转换成本地时间

返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpFileTimeFILETIME，包含了 UTC 时间信息的一个结构
lpLocalFileTimeFILETIME，用于装载转换过后的本地时间的结构

Top
FileTimeToSystemTime
FileTimeToSystemTime

VB 声明
Declare Function FileTimeToSystemTime Lib "kernel32" Alias "FileTimeToSystemTime" (lpFileTime As FILETIME, lpSystemTime As SYSTEMTIME) As Long
说明
根据一个 FILETIME 结构的内容，装载一个 SYSTEMTIME 结构
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpFileTimeFILETIME，包含了文件时间的一个结构
lpSystemTimeSYSTEMTIME，用于装载系统时间信息的一个结构

Top
FindClose
FindClose

VB 声明
Declare Function FindClose Lib "kernel32" Alias "FindClose" (ByVal hFindFile As Long) As Long
说明
关闭由 FindFirstFile 函数创建的一个搜索句柄
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hFindFileLong，由 FindFirstFile 函数提供的搜索句柄

Top
FindFirstFile
FindFirstFile

VB 声明
Declare Function FindFirstFile Lib "kernel32" Alias "FindFirstFileA" (ByVal lpFileName As

String, lpFindFileData As WIN32_FIND_DATA) As Long
说明
根据文件名查找文件
返回值
Long，如执行成功，返回一个搜索句柄。如果出错，返回一个 INVALID_HANDLE_VALUE 常数，一旦不再需要，应该用 FindClose 函数关闭这个句柄
参数表
参数类型及说明
lpFileNameString，欲搜索的文件名。可包含通配符，并可包含一个路径或相对路径名
lpFindFileDataWIN32_FIND_DATA，这个结构用于装载与找到的文件有关的信息。该结构可用于后续的搜索
注解
由这个函数返回的句柄可以作为一个参数用于 FindNextFile 函数。这样一来，就可以方便的枚举出与 lpFileName 参数指定的文件名相符的所有文件

Top
FindNextFile
FindNextFile

VB 声明
Declare Function FindNextFile Lib "kernel32" Alias "FindNextFileA" (ByVal hFindFile As Long, lpFindFileData As WIN32_FIND_DATA) As Long
说明
根据调用 FindFirstFile 函数时指定的一个文件名查找下一个文件
返回值
Long，非零表示成功，零表示失败。如不再有与指定条件相符的文件，会将 GetLastError 设置成 ERROR_NO_MORE_FILES
参数表
参数类型及说明
hFindFileLong，由 FindFirstFile 函数返回的搜索句柄
lpFindFileDataWIN32_FIND_DATA，这个结构用于装载与找到的文件有关的信息

Top
FlushFileBuffers
FlushFileBuffers

VB 声明
Declare Function FlushFileBuffers Lib "kernel32" Alias "FlushFileBuffers" (ByVal hFile As Long) As Long
说明
针对指定的文件句柄，刷新内部文件缓冲区
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表

参数类型及说明
hFileLong，文件句柄

Top
FlushViewOfFile
FlushViewOfFile

VB 声明
Declare Function FlushViewOfFile Lib "kernel32" Alias "FlushViewOfFile" (lpBaseAddress As Any, ByVal dwNumberOfBytesToFlush As Long) As Long
说明
将写入文件映射缓冲区的所有数据都刷新到磁盘
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpBaseAddressAny，包含了刷新基本地址的一个 Long 值（参考注解）
dwNumberOfBytesToFlushLong，欲刷新的字节数
注解
如与远程系统建立了文件映射，那么虽然这个函数可保证数据已在当前系统写入，但不能保证数据实际写入远程系统的磁盘——除非用 FILE_FLAG_WRITE_THROUGH 选项打开文件。该选项的作用是禁止写延迟，所有更新的数据都必须立即写入磁盘
这个函数的另一种声明形式：Declare Function FlushViewOfFile& Lib "kernel32" (ByVal lpBaseAddress As Long, ByVal dwNumberOfBytesToFlush As Long)

Top
GetBinaryType
GetBinaryType

VB 声明
Declare Function GetBinaryType Lib "kernel32" Alias "GetBinaryTypeA" (ByVal lpApplicationName As String, lpBinaryType As Long) As Long
说明
判断文件是否可以执行
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
lpApplicationNameString，欲测试文件的完整路径名
lpBinaryTypeLong，用于装载文件类型的一个变量。这些类型由下述任何一个常数定义：
SCS_32BIT_BINARYwin32 执行程序
SCS_DOS_BINARYDOS 执行程序
SCS_OS216_BINARY16 位 OS/2 执行程序
SCS_PIF_BINARY 用于执行 DOS 程序的一个 pif 文件

SCS_POSIX_BINARY 一个 Posix 应用
SCS_WOW_BINARY16 位 windows 执行程序

Top
GetCompressedFileSize
GetCompressedFileSize

VB 声明
Declare Function GetCompressedFileSize Lib "kernel32" Alias "GetCompressedFileSizeA" (ByVal lpFileName As String, lpFileSizeHigh As Long) As Long
说明
判断一个压缩文件在磁盘上实际占据的字节数
返回值
Long，返回文件长度。&HFFFFFFFF 表示出错。注意如 lpFileSizeHigh 不为 NULL，且结果为&HFFFFFFFF，那么必须调用 GetLastError，判断是否实际发生了一个错误，因为这是一个有效的结果
参数表
参数类型及说明
lpFileNameString，欲测试的文件名
lpFileSizeHighLong，指定一个 Long 值，用于装载一个 64 位文件尺寸的高 32 位。如长度没有超过 2 ^ 32 字节，则可设为 NULL（变成 ByVal）
注解
如磁盘卷已被压缩，可检查这个函数的结果是否与 GetFileSize 函数的结果有异，从而判断文件是否也被压缩（如有异，表明文件已被压缩）

Top
GetCurrentDirectory
GetCurrentDirectory

VB 声明
Declare Function GetCurrentDirectory Lib "kernel32" Alias "GetCurrentDirectory" (ByVal nBufferLength As Long, ByVal lpBuffer As String) As Long
说明
在一个缓冲区中装载当前目录
返回值
Long，装载到 lpBuffer 的字节数。如 nBufferLength 的长度不够，不足以容纳目录，则返回值是必要的缓冲区长度（要求至少这个长度），其中包括空中止字符。零表示失败。会设置 GetLastError
参数表
参数类型及说明
nBufferLengthLong，lpBuffer 缓冲区的长度
lpBufferString，指定一个预定义字串，用于装载当前目录

Top

GetDiskFreeSpace
GetDiskFreeSpace

VB 声明
Declare Function GetDiskFreeSpace Lib "kernel32" Alias "GetDiskFreeSpaceA" (ByVal lpRootPathName As String, lpSectorsPerCluster As Long, lpBytesPerSector As Long, lpNumberOfFreeClusters As Long, lpTtoalNumberOfClusters As Long) As Long
说明
获取与一个磁盘的组织有关的信息，以及了解剩余空间的容量
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpRootPathNameString，不包括卷名的一个磁盘根路径
lpSectorsPerClusterLong，用于装载一个簇内扇区数的变量
lpBytesPerSectorLong，用于装载一个扇区内字节数的变量
lpNumberOfFreeClustersLong，用于装载磁盘上剩余簇数的变量
lpTtoalNumberOfClustersLong，用于装载磁盘上总簇数的变量
注解
在采用 FAT16 格式的 windows95 系统中，如一个驱动器（分区）的容量超过了 2GB，则不应使用这个函数。此时，这个函数能识别的最大分区容量只有 2GB

Top
GetDiskFreeSpaceEx
GetDiskFreeSpaceEx

VB 声明
Declare Function GetDiskFreeSpaceEx Lib "kernel32" Alias "GetDiskFreeSpaceExA" (ByVal lpRootPathName As String, lpFreeBytesAvailableToCaller As LARGE_INTEGER, lpTotalNumberOfBytes As LARGE_INTEGER, lpTotalNumberOfFreeBytes As LARGE_INTEGER) As Long
说明
获取与一个磁盘的组织以及剩余空间容量有关的信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpRootPathNameString，不包括卷名的磁盘根路径名
lpFreeBytesAvailableToCallerLARGE_INTEGER，指定一个变量，用于容纳调用者可用的字节数量
lpTotalNumberOfBytesLARGE_INTEGER，指定一个变量，用于容纳磁盘上的总字节数
lpTotalNumberOfFreeBytesLARGE_INTEGER，指定一个变量，用于容纳磁盘上可用的字节数
适用平台

Windows 95 OSR2，Windows NT 4.0
注解
LARGE_INTEGER 结构与 FILETIME 结构在内部完全一致。正式调用前，用 GetVersionEx 判断函数是否得到了支持。在 Windows 95 OSR2 环境中，OSVERSIONINFO 结构的 dwBuildNumbe 字段会大于 1000

Top
GetDriveType
GetDriveType

VB 声明
Declare Function GetDriveType Lib "kernel32" Alias "GetDriveTypeA" (ByVal nDrive As String) As Long
说明
判断一个磁盘驱动器的类型
返回值
Long，如驱动器不能识别，则返回零。如指定的目录不存在，则返回 1。如执行成功，则用下述任何一个常数指定驱动器类型： DRIVE_REMOVABLE， DRIVE_FIXED，DRIVE_REMOTE， DRIVE_CDROM 或 DRIVE_RAMDISK
参数表
参数类型及说明
nDriveString，包含了驱动器根目录路径的一个字串

Top
GetExpandedName
GetExpandedName

VB 声明
Declare Function GetExpandedName Lib "lz32.dll" Alias "GetExpandedNameA" (ByVal lpszSource As String, ByVal lpszBuffer As String) As Long
说明
取得一个压缩文件的全名。文件必须是用 COMPRESS.EXE 程序压缩的，而且在压缩时适用/r 选项
返回值
Long，1 表示成功，LZERROR_BADVALUE 表示失败
参数表
参数类型及说明
lpszSourceString，压缩文件的名字
lpszBufferString，指定一个缓冲区，用于装载文件全名
注解
注意事先将 lpszBuffer 字串初始化成一个合适的长度，使其足以容纳文件名

Top
GetFileAttributes

GetFileAttributes

VB 声明
Declare Function GetFileAttributes Lib "kernel32" Alias "GetFileAttributesA" (ByVal lpFileName As String) As Long
说明
判断指定文件的属性
返回值
Long，-1 表示出错。如返回包含了标志的一个 Long 值，则指定文件的属性。其中的标志对应于带有 FILE_ATTRIBUTE_??? 前缀的常数。具体参考 BY_HANDLE_FILE_INFORMATION 结构的 File Attribute Types table 表格
参数表
参数类型及说明
lpFileNameString，指定欲获取属性的一个文件的名字

Top
GetFileInformationByHandle
GetFileInformationByHandle

VB 声明
Declare Function GetFileInformationByHandle Lib "kernel32" Alias "GetFileInformationByHandle" (ByVal hFile As Long, lpFileInformation As BY_HANDLE_FILE_INFORMATION) As Long
说明
这个函数提供了获取文件信息的一种机制——在一个 BY_HANDLE_FILE_INFORMATION 结构中装载与文件有关的信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hFileLong，文件的句柄
lpFileInformationBY_HANDLE_FILE_INFORMATION，用于容纳文件信息的结构

Top
GetFileSize
GetFileSize

VB 声明
Declare Function GetFileSize Lib "kernel32" Alias "GetFileSize" (ByVal hFile As Long, lpFileSizeHigh As Long) As Long
说明
判断文件长度
返回值
Long，返回文件长度。&HFFFFFFFF 表示出错。注意如 lpFileSizeHigh 不为 NULL，且结果

为&HFFFFFFFF，那么必须调用 GetLastError，判断是否实际发生了一个错误，因为这是一个有效的结果
参数表
参数类型及说明
hFileLong，文件的句柄
lpFileSizeHighLong，指定一个长整数，用于装载一个 64 位文件长度的头 32 位。如这个长度没有超过 2 ^ 32 字节，则该参数可以设为 NULL（变成 ByVal）

Top
GetFileTime
GetFileTime

VB 声明
Declare Function GetFileTime Lib "kernel32" Alias "GetFileTime" (ByVal hFile As Long, lpCreationTime As FILETIME, lpLastAccessTime As FILETIME, lpLastWriteTime As FILETIME) As Long
说明
取得指定文件的时间信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hFileLong，文件的句柄
lpCreationTimeFILETIME，用于装载文件的创建时间
lpLastAccessTimeFILETIME，用于装载文件上一次访问的时间（FAT 文件系统不支持这一特性）
lpLastWriteTimeFILETIME，用于装载文件上一次修改的时间
注解
如果不需要特定的信息，那么 lpCreationTime，lpLastAccessTime，lpLastWriteTime 都可以设置为零（用 ByVal As Long）。这个函数返回的文件时间采用 UTC 格式

Top
GetFileType
GetFileType

VB 声明
Declare Function GetFileType Lib "kernel32" Alias "GetFileType" (ByVal hFile As Long) As Long
说明
在给出文件句柄的前提下，判断文件类型
返回值
Long，下述常数之一：
FILE_TYPE_UNKNOWN 文件类型未知
FILE_TYPE_DISK 属于磁盘文件

FILE_TYPE_CHAR 文件是一个控制台或打印机
FILE_TYPE_PIPE 文件是个管道
参数表
参数类型及说明
hFileLong，要检查的文件的句柄

Top
GetFileVersionInfo
GetFileVersionInfo

VB 声明
Declare Function GetFileVersionInfo& Lib "version.dll" Alias "GetFileVersionInfoA" (ByVal lptstrFilename As String, ByVal dwHandle As Long, ByVal dwLen As Long, lpData As Byte)
说明
从支持版本标记的一个模块里获取文件版本信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lptstrFilenameString，欲从中载入版本信息的一个文件的名字
dwHandleLong，win32 中未用
dwLenLong，由 lpData 参数指定的字节数组或缓冲区的大小。用 GetFileVersionInfoSize 函数判断要求的缓冲区长度有多大
lpDataByte，指定一个字节缓冲区的第一个字节。该缓冲区用于装载文件的版本信息
注解
在 win32 下不用 dwHandle 参数
其他
请看 vb 的 api 文本查看器中的声明：Declare Function GetFileVersionInfo Lib "version.dll" Alias "GetFileVersionInfoA" (ByVal lptstrFilename As String, ByVal dwHandle As Long, ByVal dwLen As Long, lpData As Any) As Long

Top
GetFileVersionInfoSize
GetFileVersionInfoSize

VB 声明
Declare Function GetFileVersionInfoSize Lib "version.dll" Alias "GetFileVersionInfoSizeA" (ByVal lptstrFilename As String, lpdwHandle As Long) As Long
说明
针对包含了版本资源的一个文件，判断容纳文件版本信息需要一个多大的缓冲区
返回值
Long，容纳文件的版本资源所需的缓冲区长度。如文件不包含版本信息，则返回一个 0 值。会设置 GetLastError
参数表

参数类型及说明
lptstrFilenameString，包含了版本资源的一个文件的名字
lpdwHandleLong，在这个变量中载入 0 值
注解
lpdwHandle 参数在 win32 中已经放弃

Top
GetFullPathName
GetFullPathName

VB 声明
Declare Function GetFullPathName& Lib "kernel32" Alias "GetFullPathNameA" (ByVal lpFileName As String, ByVal nBufferLength As Long, ByVal lpBuffer As String, lpFilePart As Long)
说明
获取指定文件的完整路径名
返回值
Long，装载到 lpBuffer 中的字符数量（排除空中止字符）。如缓冲区的长度不足以容下完整的路径，则返回值就是要求的缓冲区大小。零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpFileNameString，指定文件名（长文件名或 8.3 格式的 DOS 文件名）
nBufferLengthLong，lpBuffer 字串的长度
lpBufferString，指定一个预先定义好的字串，用于装载目标文件的驱动器及路径名称。如存在长文件名，那么这个参数保存的就肯定是长文件名
lpFilePartLong，指定一个长整数变量，用于装载文件名起始的地方。参考注解
注解
lpFilePart 参数在 vb 里很难使用。它的问题在于：尽管 windows 在这个 Long 值中装载 lpBuffer 字串中的地址，用它表示路径信息文件名部分的起始处。但非常不幸，由 vb 创建的、传递给 api 的 ANSI 字串缓冲区也会使用这个地址。等这个函数返回的时候，vb 已将返回的（lpBuffer）字串复制回它的内部 Unicode 字串缓冲区，所以 lpFilePart 地址已没有任何意义。因此，我们面临两种选择。首先，可以简单的不使用 lpFilePart 信息（忽略 windows 装载在参数中的值）。其次，可以将 lpBuffer 参数变成一个字节数组（lpFilePart As Byte——将数组的第一个元素作为参数传递）
其他
在 vb 的 api 文本查看器中复制的声明为：Declare Function GetFullPathName Lib "kernel32" Alias "GetFullPathNameA" (ByVal lpFileName As String, ByVal nBufferLength As Long, ByVal lpBuffer As String, ByVal lpFilePart As String) As Long

Top
GetLogicalDrives
GetLogicalDrives

VB 声明

Declare Function GetLogicalDrives Lib "kernel32" Alias "GetLogicalDrives" () As Long
说明
判断系统中存在哪些逻辑驱动器字母
返回值
Long，这个结构中的二进制位标志着存在哪些驱动器。其中，位 0 设为 1 表示驱动器 A:存在于系统中；位 1 设为 1 表示存在 B:驱动器；以次类推

Top
GetLogicalDriveStrings
GetLogicalDriveStrings

VB 声明
Declare Function GetLogicalDriveStrings Lib "kernel32" Alias "GetLogicalDriveStringsA" (ByVal nBufferLength As Long, ByVal lpBuffer As String) As Long
说明
获取一个字串，其中包含了当前所有逻辑驱动器的根驱动器路径
返回值
Long，装载到 lpBuffer 的字符数量（排除空中止字符）。如缓冲区的长度不够，不能容下路径，则返回值就变成要求的缓冲区大小。零表示失败。会设置 GetLastError
参数表
参数类型及说明
nBufferLengthLong，lpBuffer 字串的长度
lpBufferString，用于装载逻辑驱动器名称的字串。每个名字都用一个 NULL 字符分隔，在最后一个名字后面用两个 NULL 表示中止（空中止）

Top
GetOverlappedResult
GetOverlappedResult

VB 声明
Declare Function GetOverlappedResult Lib "kernel32" Alias "GetOverlappedResult" (ByVal hFile As Long, lpOverlapped As OVERLAPPED, lpNumberOfBytesTransferred As Long, ByVal bWait As Long) As Long
说明
判断一个重叠操作当前的状态
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError。如 bWait 为 FALSE，而且异步操作仍在执行，则函数回返回零，而 GetLastError 会设置成 ERROR_IO_INCOMPLETE
参数表
参数类型及说明
hFileLong，指定一个文件、管道或通信设备的句柄
lpOverlappedOVERLAPPED，为欲检查的 I/O 操作指定的一个结构
lpNumberOfBytesTransferredLong，用于容纳传输字节数量的一个变量
bWaitLong，如果为 TRUE，就一直等到异步操作结束才返回。FALSE 表示立即返回

Top
GetPrivateProfileInt
GetPrivateProfileInt

VB 声明
Declare Function GetPrivateProfileInt Lib "kernel32" Alias "GetPrivateProfileIntA" (ByVal lpApplicationName As String, ByVal lpKeyName As String, ByVal nDefault As Long, ByVal lpFileName As String) As Long
说明
为初始化文件中指定的条目获取一个整数值
返回值
Long，找到的条目的值；如指定的条目未找到，就返回默认值。如找到的数字不是一个合法的整数，函数会返回其中合法的一部分。如，对于“xyz=55zz”这个条目，函数返回 55。这个函数也能理解采用标准 C 语言格式的十六进制数字：用 0x 作为一个十六进制数字的前缀——所以 0x55ab 等价于 vb 的&H55AB
参数表
参数类型及说明
lpApplicationNameString，指定在其中查找条目的小节。注意这个字串是不区分大小写的
lpKeyNameString，欲获取的设置项或条目。这个支持不区分大小写
nDefaultLong，指定条目未找到时返回的默认值
lpFileNameString，初始化文件的名字。如果没有指定完整的路径名，windows 就会在 Windows 目录中搜索文件
注解
在 Windows NT 中，有些初始化文件实际是在注册表中。可在注册表的下面这个项处找到这些文件的一个列表： HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\IniFileMapping

Top
GetPrivateProfileSection
GetPrivateProfileSection

VB 声明
Declare Function GetPrivateProfileSection Lib "kernel32" Alias "GetPrivateProfileSectionA" (ByVal lpAppName As String, ByVal lpReturnedString As String, ByVal nSize As Long, ByVal lpFileName As String) As Long
说明
获取指定小节所有项名和值的一个列表
返回值
Long，装载到 lpReturnedString 缓冲区的字符数量。如缓冲区的容量不够大，不能容下所有信息，就返回 nSize-2
参数表
参数类型及说明
lpAppNameString，欲获取的小节。注意这个字串不区分大小写

lpReturnedStringString，项和值字串的列表。每个字串都由一个 NULL 字符分隔，最后一个字串后面用两个 NULL 字符中止
nSizeLong，lpReturnedString 缓冲区的大小。在 windows 系统中最大值为 32767
lpFileNameString，初始化文件的名字。如没有指定完整路径名，windows 就在 Windows 目录中查找文件
注解
参考对 GetPrivateProfileInt 函数的注解

Top
GetPrivateProfileString
GetPrivateProfileString

VB 声明
Declare Function GetPrivateProfileString& Lib "kernel32" Alias "GetPrivateProfileStringA" (ByVal lpApplicationName As String, ByVal lpKeyName As String, ByVal lpDefault As String, ByVal lpReturnedString As String, ByVal nSize As Long, ByVal lpFileName As String)
说明
为初始化文件中指定的条目取得字串
返回值
Long，复制到 lpReturnedString 缓冲区的字节数量，其中不包括那些 NULL 中止字符。如 lpReturnedString 缓冲区不够大，不能容下全部信息，就返回 nSize-1（若 lpApplicationName 或 lpKeyName 为 NULL，则返回 nSize-2）
参数表
参数类型及说明
lpApplicationNameString，欲在其中查找条目的小节名称。这个字串不区分大小写。如设为 vbNullString，就在 lpReturnedString 缓冲区内装载这个 ini 文件所有小节的列表
lpKeyNameString，欲获取的项名或条目名。这个字串不区分大小写。如设为 vbNullString，就在 lpReturnedString 缓冲区内装载指定小节所有项的列表
lpDefaultString，指定的条目没有找到时返回的默认值。可设为空（""）
lpReturnedStringString，指定一个字串缓冲区，长度至少为 nSize
nSizeLong，指定装载到 lpReturnedString 缓冲区的最大字符数量
lpFileNameString，初始化文件的名字。如没有指定一个完整路径名，windows 就在 Windows 目录中查找文件
注解
如 lpKeyName 参数为 vbNullString，那么 lpReturnedString 缓冲区会载入指定小节所有设置项的一个列表。每个项都用一个 NULL 字符分隔，最后一个项用两个 NULL 字符中止。也请参考 GetPrivateProfileInt 函数的注解
其他
在 vb 的 api 文本查看器中复制的声明为：Declare Function GetPrivateProfileString Lib "kernel32" Alias "GetPrivateProfileStringA" (ByVal lpApplicationName As String, ByVal lpKeyName As Any, ByVal lpDefault As String, ByVal lpReturnedString As String, ByVal nSize As Long, ByVal lpFileName As String) As Long

Top

GetProfileInt
GetProfileInt

VB 声明
Declare Function GetProfileInt Lib "kernel32" Alias "GetProfileIntA" (ByVal lpAppName As String, ByVal lpKeyName As String, ByVal nDefault As Long) As Long
说明
取得 win.ini 初始化文件中指定条目的一个整数值
返回值
Long，找到条目的值；如指定的条目未找到，就返回默认值。如找到的数字不是一个合法的整数，函数就会返回其中合法的一部分。例如，对于“xyz=55zz”这个条目，函数会返回 55。这个函数也能理解采用标准 C 语言格式的十六进制数字：用 0x 作为一个十六进制数字的前缀——所以 0x55ab 等价于 vb 的&H55AB
参数表
参数类型及说明
lpAppNameString，欲在其中搜索条目的小节名。这个字串不区分大小写
lpKeyNameString，欲获取的项名或条目名。这个字串不区分大小写
nDefaultLong，指定在条目未找到时返回的默认值
注解
参考对 GetPrivateProfileInt 函数的注解

Top
GetProfileSection
GetProfileSection

VB 声明
Declare Function GetProfileSection Lib "kernel32" Alias "GetProfileSectionA" (ByVal lpAppName As String, ByVal lpReturnedString As String, ByVal nSize As Long) As Long
说明
获取指定小节（在 win.ini 文件中）所有项名和值的一个列表
返回值
Long，装载到 lpReturnedString 缓冲区的字符数量。如缓冲区的长度不足以容下所有信息，则返回 nSize-2
参数表
参数类型及说明
lpAppNameString，欲获取的小节。这个字串不区分大小写
lpReturnedStringString，用于容纳项和值字串列表的一个缓冲区。每个字串都用一个 NULL 分隔，最后一个字串用两个 NULL 字符中止
nSizeLong，lpReturnedString 缓冲区的大小，在 windows 95 中最大为 32767
注解
参考 GetPrivateProfileInt 函数的注解

Top
GetProfileString

GetProfileString

VB 声明
Declare Function GetProfileString Lib "kernel32" Alias "GetProfileStringA" (ByVal lpAppName As String, ByVal lpKeyName As String, ByVal lpDefault As String, ByVal lpReturnedString As String, ByVal nSize As Long) As Long
说明
为 win.ini 初始化文件中指定的条目取得字串
返回值
Long，复制到 lpReturnedString 缓冲区的字节数量，其中不包括那些 NULL 中止字符。如 lpReturnedString 缓冲区不够大，不能容下全部信息，就返回 nSize-1（若 lpAppName 或 lpKeyName 为 NULL，则返回 nSize-2）
参数表
参数类型及说明
lpAppNameString，要在其中查找条目的小节名。这个字串不区分大小写。如果为 vbNullString，则在 lpReturnedString 缓冲区装载这个.ini 文件的所有小节的一个列表
lpKeyNameString，欲获取的项名或条目名。这个字串不区分大小写。如果为 vbNullString，则在 lpReturnedString 缓冲区装载指定小节内所有项的一个列表
lpDefaultString，指定条目未找到时返回的默认值。可设为空（""）
lpReturnedStringString，指定一个预先初始化好的字串缓冲区，长度至少为 nSize 个字符
nSizeLong，装载到 lpReturnedString 缓冲区的最大字符数
注解
如 lpKeyName 参数为零，那么 lpReturnedString 缓冲区会载入指定小节内所有设置项的一个列表。每个项都用一个 NULL 字符分隔，最后那个项用两个 NULL 字符中止

Top
GetShortPathName
GetShortPathName

VB 声明
Declare Function GetShortPathName Lib "kernel32" Alias "GetShortPathName" (ByVal lpszLongPath As String, ByVal lpszShortPath As String, ByVal cchBuffer As Long) As Long
说明
获取指定文件的短路径名
返回值
Long，装载到 lpszShortPath 缓冲区的字符数量。如 lpszShortPath 的长度不足，不能容下文件名，就返回需要的缓冲区长度
参数表
参数类型及说明
lpszLongPathString，指定欲获取短路径名的那个文件的名字。可以是个完整路径，或者由当前目录决定
lpszShortPathString，指定一个缓冲区，用于装载文件的短路径和文件名
cchBufferLong，lpszShortPath 缓冲区长度

Top
GetSystemDirectory
GetSystemDirectory

VB 声明
Declare Function GetSystemDirectory Lib "kernel32" Alias "GetSystemDirectoryA" (ByVal lpBuffer As String, ByVal nSize As Long) As Long
说明
这个函数能取得 Windows 系统目录（System 目录）的完整路径名。在这个目录中，包含了所有必要的系统文件。根据微软的标准，其他定制控件和一些共享组件也可放到这个目录。通常应避免在这个目录里创建文件。在网络环境中，往往需要管理员权限才可对这个目录进行写操作
返回值
Long，装载到 lpBuffer 缓冲区的字符数量。如 lpBuffer 不够大，不能容下文件名，则返回要求的缓冲区长度
参数表
参数类型及说明
lpBufferString，用于装载系统目录路径名的一个字串缓冲区。它应事先初始化成 nSize+1 个字符的长度。通常至少要为这个缓冲区分配 MAX_PATH 个字符的长度
nSizeLong，lpBuffer 字串的最大长度

Top
GetTempFileName
GetTempFileName

VB 声明
Declare Function GetTempFileName Lib "kernel32" Alias "GetTempFileNameA" (ByVal lpszPath As String, ByVal lpPrefixString As String, ByVal wUnique As Long, ByVal lpTempFileName As String) As Long
说明
这个函数包含了一个临时文件的名字，它可由应用程序使用
返回值
Long，最终用于生成文件名的 wUnique 数字的值。如 wUnique 参数不为零，这就是参数的值。零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpszPathString，临时文件使用的目录。通常用 GetTempPath 函数获得
lpPrefixStringString，要使用的文件名前缀。头三个字符作为文件名前缀使用
wUniqueLong，追加到前缀字串后面的数字。如果为 0，则这个函数会用一个随机数字生成文件。随后，它会检查是否存在同名的文件。如果存在，函数会增加这个数字，并继续尝试，直到生成一个独一无二的名字为止。文件在驱动器上会以长度为 0 字节的形式保存。如果不为零，就不会创建文件，而且函数不会核实它是否一个独一无二的文件名
lpTempFileNameString，用于装载新建临时文件名的缓冲区，这个缓冲区的长度至少应为 MAX_PATH 个字符

注解
函数使用的文件名肯定采用 ANSI 字符集。临时文件不会被 windows 自动删除

Top
GetTempPath
GetTempPath

VB 声明
Declare Function GetTempPath Lib "kernel32" Alias "GetTempPathA" (ByVal nBufferLength As Long, ByVal lpBuffer As String) As Long
说明
获取为临时文件指定的路径
返回值
Long，装载到 lpBuffer 的字符数。如当前缓冲区的长度不够，不能容下整个路径，则返回 lpBuffer 需要的长度。零表示失败。会设置 GetLastError
参数表
参数类型及说明
nBufferLengthLong，lpBuffer 字串的长度
lpBufferString，用于装载临时文件路径的一个预初始化字串
注解
临时路径是由 TMP 环境变量指定的一个路径。如 TMP 不存在，则是由 TEMP 环境变量指定的路径。如果这两个环境变量都不存在，就是当前目录

Top
GetVolumeInformation
GetVolumeInformation

VB 声明
Declare Function GetVolumeInformation Lib "kernel32" Alias "GetVolumeInformationA" (ByVal lpRootPathName As String, ByVal lpVolumeNameBuffer As String, ByVal nVolumeNameSize As Long, lpVolumeSerialNumber As Long, lpMaximumComponentLength As Long, lpFileSystemFlags As Long, ByVal lpFileSystemNameBuffer As String, ByVal nFileSystemNameSize As Long) As Long
说明
获取与一个磁盘卷有关的信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpRootPathNameString，欲获取信息的那个卷的根路径
lpVolumeNameBufferString，用于装载卷名（卷标）的一个字串
nVolumeNameSizeLong，lpVolumeNameBuffer 字串的长度
lpVolumeSerialNumberLong，用于装载磁盘卷序列号的变量
lpMaximumComponentLengthLong，指定一个变量，用于装载文件名每一部分的长度。例如，

在“c:\component1\component2.ext”的情况下，它就代表 component1 或 component2 名称的长度
lpFileSystemFlagsLong，用于装载一个或多个二进制位标志的变量。对这些标志位的解释如下：
FS_CASE_IS_PRESERVED 文件名的大小写记录于文件系统
FS_CASE_SENSITIVE 文件名要区分大小写
FS_UNICODE_STORED_ON_DISK 文件名保存为 Unicode 格式
FS_PERSISTANT_ACLS 文件系统支持文件的访问控制列表（ACL）安全机制
FS_FILE_COMPRESSION 文件系统支持逐文件的进行文件压缩
FS_VOL_IS_COMPRESSED 整个磁盘卷都是压缩的
lpFileSystemNameBufferString，指定一个缓冲区，用于装载文件系统的名称（如 FAT，NTFS 以及其他）
nFileSystemNameSizeLong，lpFileSystemNameBuffer 字串的长度

Top
GetWindowsDirectory
GetWindowsDirectory

VB 声明
Declare Function GetWindowsDirectory Lib "kernel32" Alias "GetWindowsDirectoryA" (ByVal lpBuffer As String, ByVal nSize As Long) As Long
说明
这个函数能获取 Windows 目录的完整路径名。在这个目录里，保存了大多数 windows 应用程序文件及初始化文件
返回值
Long，复制到 lpBuffer 的一个字串的长度。如 lpBuffer 不够大，不能容下整个字串，就会返回 lpBuffer 要求的长度。零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpBufferString，指定一个字串缓冲区，用于装载 Windows 目录名。除非是根目录，否则目录中不会有一个中止用的“\”字符
nSizeLong，lpBuffer 字串的最大长度

Top
hread
hread

VB 声明
Declare Function hread Lib "kernel32" Alias "_hread" (ByVal hFile As Long, lpBuffer As Any, ByVal lBytes As Long) As Long
说明
参考 lread
注解
这个函数在 win16 中用于读取大于 64KB 的数据块。但 win32 的文件 I/O 函数并不受这个

64KB 的限制

Top
hwrite
hwrite

VB 声明
Declare Function hwrite Lib "kernel32" Alias "_hwrite" (ByVal hFile As Long, ByVal lpBuffer As String, ByVal lBytes As Long) As Long
说明
参考 lwrite 函数
注解
这个函数在 win16 中用于写入大于 64KB 的数据块。但 win32 的文件 I/O 函数并不受这个 64KB 的限制

Top
lclose
lclose

VB 声明
Declare Function lclose Lib "kernel32" Alias "_lclose" (ByVal hFile As Long) As Long
说明
关闭指定的文件，请参考 CloseHandle 函数，了解进一步的情况

Top
lcreat
lcreat

VB 声明
Declare Function lcreat Lib "kernel32" Alias "_lcreat" (ByVal lpPathName As String, ByVal iAttribute As Long) As Long
说明
创建一个文件。如文件已经存在，就会将其缩短成零长度，并将其打开，以便读写
返回值
Long，如执行成功，返回打开文件的句柄。如果出错，则返回 HFILE_ERROR.
参数表
参数类型及说明
lpPathNameString，欲创建的文件的名字
iAttributeLong，下述值之一
0——文件能够读写
1——创建只读文件
2——创建隐藏文件
3——创建系统文件
注解

该函数会打开已由其他应用程序打开的文件，所以使用它时要小心。win32 的 CreateFile 函数已取代了这个函数。这个函数与 vb 的 open 语句作用相同

Top
llseek
llseek

VB 声明
Declare Function llseek Lib "kernel32" Alias "_llseek" (ByVal hFile As Long, ByVal lOffset As Long, ByVal iOrigin As Long) As Long
说明
设置文件中进行读写的当前位置。该函数与 vb 的 seek 语句类似。如果用 vb 的 open 命令打开了一个文件，那么不要再对这个文件使用 llseek 函数
返回值
Long，返回一个新位置，设置成从文件起始处算起的一个偏移量。HFILE_ERROR 表示函数执行出错。会设置 GetLastError
参数表
参数类型及说明
hFileLong，系统文件句柄
lOffsetLong，字节偏移量
iOriginLong，下述常数之一
FILE_BEGINlOffset 将新位置指定成从文件起始处的一个偏移距离
FILE_CURRENTlOffset 将新位置指定成从当前位置开始的一个偏移距离
FILE_ENDlOffset 将新位置指定成从文件结尾开始的的一个偏移距离
注解
参考 SetFilePointer 函数，认识能对较大文件进行处理的一个近似函数.

Top
LockFile
LockFile

VB 声明
Declare Function LockFile Lib "kernel32" Alias "LockFile" (ByVal hFile As Long, ByVal dwFileOffsetLow As Long, ByVal dwFileOffsetHigh As Long, ByVal nNumberOfBytesToLockLow As Long, ByVal nNumberOfBytesToLockHigh As Long) As Long
说明
在 windows 中，文件可用共享模式打开——在这种情况下，多个进程可同时访问该文件。利用这个函数，要对文件进行读写的一个应用程序可将文件的某一部分锁定起来，使其不能由其他应用程序访问。这样便避免了同时读写时发生的冲突
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hFileLong，欲锁定文件的句柄

dwFileOffsetLowLong，指定欲锁定区域起始处的低 32 位地址
dwFileOffsetHighLong，指定欲锁定区域起始处的高 32 位地址
nNumberOfBytesToLockLowLong，锁定区域包含字符数量的低 32 位值
nNumberOfBytesToLockHighLong，锁定区域包含字符数量的高 32 位值
注解
锁定的区域不能进行重叠操作。由不同的操作系统决定，可能要求先运行 share.exe 才能保证该函数正常工作

Top
LockFileEx
LockFileEx

VB 声明
Declare Function LockFileEx Lib "kernel32" Alias "LockFileEx" (ByVal hFile As Long, ByVal dwFlags As Long, ByVal dwReserved As Long, ByVal nNumberOfBytesToLockLow As Long, ByVal nNumberOfBytesToLockHigh As Long, lpOverlapped As OVERLAPPED) As Long
说明
与 LockFile 相似，只是它提供了更多的功能
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hFileLong，欲锁定文件的句柄
dwFlagsLong，指定下述一个或两个常数
LOCKFILE_FAIL_IMMEDIATELY 指出如锁定失败，函数应返回一个错误。否则，应用程序线程就会暂时挂起，并一直等待，直到能进行锁定为止
LOCKFILE_EXCLUSIVE_LOCK 指出锁定区域不可由另一个线程或进程读写。否则这个区域就只能防范“写”——其他进程仍然能够读取锁定区域的内容
dwReservedLong，未使用，设为零
nNumberOfBytesToLockLowLong，锁定区域包含字符数的低 32 位
nNumberOfBytesToLockHighLong，锁定区域包含字符数的高 32 位
lpOverlappedOVERLAPPED，包含了文件中相对于锁定区域起始处的偏移量
注解
锁定区域不可重叠操作（即多个进程同时操作）

Top
lopen
lopen

VB 声明
Declare Function lopen Lib "kernel32" Alias "_lopen" (ByVal lpPathName As String, ByVal iReadWrite As Long) As Long
说明
以二进制模式打开指定的文件

返回值
Long，如执行成功，返回打开文件的句柄。HFILE_ERROR 表示出错。会设置 GetLastError
参数表
参数类型及说明
lpPathNameString，欲打开文件的名字
iReadWriteLong，访问模式和共享模式常数的一个组合，如下所示：
1、访问模式
READ 打开文件，读取其中的内容
READ_WRITE 打开文件，对其进行读写
WRITE 打开文件，在其中写入内容
2、共享模式（参考 OpenFile 函数的标志常数表）
OF_SHARE_COMPAT ， OF_SHARE_DENY_NONE ， OF_SHARE_DENY_READ ， OF_SHARE_DENY_WRITE， OF_SHARE_EXCLUSIVE
注解
CreateFile 函数在 win32 下提供了更多的功能

Top
lread
lread

VB 声明
Declare Function lread Lib "kernel32" Alias "_lread" (ByVal hFile As Long, lpBuffer As Any, ByVal wBytes As Long) As Long
说明
将文件中的数据读入内存缓冲区
返回值
Long，返回实际读入的字节数。HFILE_ERROR 意味着函数执行出错。如这个数字小于 wBytes，则表明早已抵达了文件的末尾。会设置 GetLastError
参数表
参数类型及说明
hFileLong，文件句柄
lpBufferAny，指定一个内存块的指针，数据将读入这个内存块
wBytesLong，要读入的字节数

Top
lwrite
lwrite

VB 声明
Declare Function lwrite Lib "kernel32" Alias "_lwrite" (ByVal hFile As Long, ByVal lpBuffer As String, ByVal wBytes As Long) As Long
说明
将数据从内存缓冲区写入一个文件
返回值

Long，写入的字节数。HFILE_ERROR 意味着函数执行出错。如这个数字小于 wBytes，则表明早已抵达了文件的末尾。会设置 GetLastError
参数表
参数类型及说明
hFileLong，文件句柄
lpBufferString，指定一个内存块的指针，把这个内存块的数据写入文件
wBytesLong，要写入的字节数

Top
LZClose
LZClose

VB 声明
Declare Sub LZClose Lib "lz32.dll" Alias "LZClose" (ByVal hfFile As Long)
说明
关闭由 LZOpenFile 或 LZInit 函数打开的一个文件
参数表
参数类型及说明
hfFileLong，欲关闭的句柄。这是由 LZOpenFile 或 LZInit 函数返回的一个句柄，不是普通的系统文件句柄

Top
LZCopy
LZCopy

VB 声明
Declare Function LZCopy Lib "lz32.dll" Alias "LZCopy" (ByVal hfSource As Long, ByVal hfDest As Long) As Long
说明
复制一个文件。如源文件已压缩，则会在复制期间解压。文件必须是用微软公司的 compress.exe 或等效工具压缩的
返回值
Long，如执行成功，返回目标文件的大小，以字节为单位。如执行出错，会返回小于零的一个常数，如下表
LZERROR_BADINHANDLE 源文件无效
LZERROR_BADOUTHANDLE 目标文件无效
LZERROR_GLOBALLOC 内部解压缓冲区的内存容量不足
LZERROR_GLOBLOCK 内部解压缓冲区的句柄无效
LZERROR_READ 无效的源文件格式
LZERROR_UNKNOWNALG 解压 DLL 不能识别源文件采用的压缩算法
LZERROR_WRITE 在磁盘上写入输出文件时出错，通常是由于磁盘空间不足造成的
参数表
参数类型及说明
hfSourceLong，指定源文件句柄。这是由 LZOpenFile 或 LZInit 函数返回的一个句柄

hfDestLong，指定目标文件句柄。这是由 LZOpenFile 或 LZInit 函数返回的一个句柄

Top
LZInit
LZInit

VB 声明
Declare Function LZInit Lib "lz32.dll" Alias "LZInit" (ByVal hfSrc As Long) As Long
说明
这个函数用于初始化内部缓冲区。对一个给出打开文件句柄的一个文件进行解压时，将用到这个缓冲区
返回值
Long，由 lz32.dll 库使用的、那个文件的一个特殊句柄。这个文件句柄兼容于 LZCopy，CopyLZFiles， LZRead 和 LZSeek 函数。如果出错，该函数会返回下表列出的出错代码之一。注意完成后一定用 LZClose 关闭这个句柄
LZERROR_BADINHANDLE 源文件无效
LZERROR_BADOUTHANDLE 目标文件无效
LZERROR_GLOBALLOC 内部解压缓冲区的内存容量不足
LZERROR_GLOBLOCK 内部解压缓冲区的句柄无效
LZERROR_READ 无效的源文件格式
LZERROR_UNKNOWNALG 解压 DLL 不能识别源文件采用的压缩算法
LZERROR_WRITE 在磁盘上写入输出文件时出错，通常是由于磁盘空间不足造成的
参数表
参数类型及说明
hfSrcLong，文件的句柄
注解
最多只能同时打开 16 个压缩文件句柄

Top
LZOpenFile
LZOpenFile

VB 声明
Declare Function LZOpenFile Lib "lz32.dll" Alias "LZOpenFileA" (ByVal lpszFile As String, lpOf As OFSTRUCT, ByVal style As Long) As Long
说明
该函数能执行大量不同的文件处理，而且兼容于压缩文件
返回值
Long，如函数执行成功，且样式（style）参数不为 OF_READ，就返回常规的文件句柄，具体请参考 OpenFile 函数的说明。如样式参数为 OF_READ，而且文件是压缩的，就会返回一个特殊的文件句柄，以便由 LZCopy， LZRead 和 LZSeek 函数使用。如出错，返回如下表所示的一个常数：
LZERROR_BADINHANDLE 源文件无效
LZERROR_BADOUTHANDLE 目标文件无效

LZERROR_GLOBALLOC 内部解压缓冲区的内存容量不足
LZERROR_GLOBLOCK 内部解压缓冲区的句柄无效
LZERROR_READ 无效的源文件格式
LZERROR_UNKNOWNALG 解压 DLL 不能识别源文件采用的压缩算法
LZERROR_WRITE 在磁盘上写入输出文件时出错，通常是由于磁盘空间不足造成的
参数表
参数类型及说明
lpszFileString，欲打开的文件名
lpOfOFSTRUCT，该结构填充的数据包括与本次处理的文件和结果有关的信息
styleLong，处理方式标志常数的一种组合。参考 OpenFile 函数的标志常数表
注解
参考 OpenFile 函数

Top
LZRead
LZRead

VB 声明
Declare Function LZRead Lib "lz32.dll" Alias "LZRead" (ByVal hfFile As Long, ByVal lpvBuf As String, ByVal cbread As Long) As Long
说明
将数据从文件读入内存缓冲区。如 hfFile 是一个压缩文件的句柄，同时那个压缩文件是由 LZOpenFile 或 LZInit 函数打开的，这个函数就会在读入数据的同时对文件进行解压处理
返回值
Long，实际读入的字节数。如这个数字小于 cbread，表明早已抵达了文件的末尾。如出错，返回下表列出的常数之一
LZERROR_BADINHANDLE 源文件无效
LZERROR_BADOUTHANDLE 目标文件无效
LZERROR_GLOBALLOC 内部解压缓冲区的内存容量不足
LZERROR_GLOBLOCK 内部解压缓冲区的句柄无效
LZERROR_READ 无效的源文件格式
LZERROR_UNKNOWNALG 解压 DLL 不能识别源文件采用的压缩算法
LZERROR_WRITE 在磁盘上写入输出文件时出错，通常是由于磁盘空间不足造成的
参数表
参数类型及说明
hfFileLong，源文件的特殊句柄。这个句柄是由 LZOpenFile 或 LZInit 函数提供的
lpvBufString，一个内存块的指针，数据将读入这个内存块
cbreadLong，指定 lpvBuf 缓冲区的长度

Top
LZSeek
LZSeek

VB 声明

Declare Function LZSeek Lib "lz32.dll" Alias "LZSeek" (ByVal hfFile As Long, ByVal lOffset As Long, ByVal nOrigin As Long) As Long
说明
设置一个文件中进行读写的当前位置。如 hfFile 是一个压缩文件的句柄，同时那个压缩文件是由 LZOpenFile 或 LZInit 函数打开的，这个函数就会根据文件的解压版本进行查找
返回值
Long，返回一个新位置，采用从文件起始处计算的字节偏移量。如出错，返回下表列出的常数之一
LZERROR_BADINHANDLE 源文件无效
LZERROR_BADOUTHANDLE 目标文件无效
LZERROR_GLOBALLOC 内部解压缓冲区的内存容量不足
LZERROR_GLOBLOCK 内部解压缓冲区的句柄无效
LZERROR_READ 无效的源文件格式
LZERROR_UNKNOWNALG 解压 DLL 不能识别源文件采用的压缩算法
LZERROR_WRITE 在磁盘上写入输出文件时出错，通常是由于磁盘空间不足造成的
参数表
参数类型及说明
hfFileLong，源文件的特殊句柄。这个句柄是由 LZOpenFile 或 LZInit 函数提供的
lOffsetLong，以字节数表示的偏移量
nOriginLong，下述值之一
0——lOffset 将新位置指定成从文件的起始处计算偏移
1——lOffset 将新位置指定成从当前位置开始计算偏移
2——lOffset 将新位置指定成从文件的结尾处计算偏移

Top
MapViewOfFile
MapViewOfFile, MapViewOfFileEx

VB 声明
Declare Function MapViewOfFile& Lib "kernel32" (ByVal hFileMappingObject As Long, ByVal dwDesiredAccess As Long, ByVal dwFileOffsetHigh As Long, ByVal dwFileOffsetLow As Long, ByVal dwNumberOfBytesToMap As Long)
Declare Function MapViewOfFileEx& Lib "kernel32" (ByVal hFileMappingObject As Long, ByVal dwDesiredAccess As Long, ByVal dwFileOffsetHigh As Long, ByVal dwFileOffsetLow As Long, ByVal dwNumberOfBytesToMap As Long, lpBaseAddress As Any)
说明
将一个文件映射对象映射到当前应用程序的地址空间。MapViewOfFileEx 允许我们指定一个基本地址来进行映射
返回值
Long，文件映射在内存中的起始地址。零表示出错。会设置 GetLastError
参数表
参数类型及说明
hFileMappingObjectLong，文件映射对象的句柄
dwDesiredAccessLong，下述常数之一：

FILE_MAP_WRITE 映射可读可写。文件映射对象必须通过 PAGE_READWRITE 访问创建
FILE_MAP_READ 映射只读。文件映射对象必须通过 PAGE_READ 或 PAGE_READWRITE 访问创建
FILE_MAP_ALL_ACCESS 与 FILE_MAP_WRITE 相同
FILE_MAP_COPY 映射时保留写操作的副本。文件映射对象必须用 PAGE_WRITECOPY 访问在 win95 下创建
dwFileOffsetHighLong，文件中映射起点的高 32 位地址
dwFileOffsetLowLong，文件中映射起点的低 32 位地址
dwNumberOfBytesToMapLong，文件中要映射的字节数。用零映射整个文件映射对象
lpBaseAddressLong，指定映射文件映射对象的地址。如这个地址处没有足够的内存空间，那么对 MapViewOfFileEx 的调用会失效。零表示允许 windows 寻找一个地址
注解
dwFileOffsetLow 和 dwFileOffsetHigh 必须反映一个偏移距离，它由系统的内存分配精度决定。例如，假设系统的内存精度是 64KB（即最小分配单位是 64KB），则这些值必须是 64KB 的整数倍。大多数应用程序都简单的用零从文件的起始处开始映射。 lpBaseAddress 也必须是内存分配精度的整数倍
其他
声明中的参数类型为 Any，而参数表中都是 Long，我也不明白。但关于这个函数的英文资料的确是这样的。

Top
MoveFile
MoveFile, MoveFileEx

VB 声明
Declare Function MoveFile& Lib "kernel32" Alias "MoveFileA" (ByVal lpExistingFileName As String, ByVal lpNewFileName As String)
Declare Function MoveFileEx& Lib "kernel32" Alias "MoveFileExA" (ByVal lpExistingFileName As String, ByVal lpNewFileName As String, ByVal dwFlags As Long)
说明
移动文件。如 dwFlags 设为零，则 MoveFile 完全等价于 MoveFileEx
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpExistingFileNameString，欲移动的文件名
lpNewFileNameString，新文件名
dwFlagsLong，一个或多个下述常数
MOVEFILE_REPLACE_EXISTING 如目标文件存在，则将其替换
MOVEFILE_COPY_ALLOWED 如移动到一个不同的卷，则复制文件并删除原来的文件
MOVEFILE_DELAY_UNTIL_REBOOT 移动操作在系统下次重新启动时正式进行。这样便可在 Windows NT 中改换系统文件
注解
这两个函数通常不能将文件从一个卷移动到另一个卷。但如设置了

MOVEFILE_COPY_ALLOWED 标记，MoveFileEx 可以做到这一点

Top
OpenFile
OpenFile

VB 声明
Declare Function OpenFile Lib "kernel32" Alias "OpenFile" (ByVal lpFileName As String, lpReOpenBuff As OFSTRUCT, ByVal wStyle As Long) As Long
说明
这个函数能执行大量不同的文件操作。和这个函数相比，请优先考虑 win32 的 CreateFile 函数（它能打开命名管道和控制 Unicode 文件名，同时不受 128 个字符的路径名称的限制）
返回值
Long，如执行成功，返回文件句柄。注意文件句柄可能是无效的；例如，假设指定了 OF_EXIST 标志，文件在函数返回前会关闭，但它打开时的句柄却永远不会返回。如果出错，函数会返回 HFILE_ERROR；此时，由 lpReOpenBuff 指定的 OFSTRUCT 结构的 nErrCode 会设置成发生的错误。表 OpenFile-2（OFSTRUCT 出错代码）对这些错误进行了总结。会设置 GetLastError
参数表
参数类型及说明
lpFileNameString，欲打开文件的名字
lpReOpenBuffOFSTRUCT，该结构填充的数据包括与文件和操作结果有关的信息
wStyleLong，参考表 OpenFile-1（OpenFile 函数的标志常数表）总结的标志常数的组合，它决定了要采取的操作方式

表 OpenFile-1（OpenFile 函数的标志常数表）
wStyle 常数说明
OF_CREATE 创建指定的文件。如已经存在，则将其缩减为零长度
OF_DELETE 删除指定的文件
OF_EXIST 通过尝试打开文件的做法，判断一个文件是否存在。如文件存在，则将其关闭。此时，函数会返回文件打开时使用的句柄，但这个句柄是无效的。如指定的文件不存在，则返回一个负数
OF_PARSE 填写 lpReOpenBuff 结构的内容，但不执行其他任何操作
OF_PROMPT 如文件不存在，则显示一个消息框，在其中列出重试和取消按钮
OF_READ 以只读方式打开文件
OF_READWRITE 以可读、可写的方式打开文件
OF_REOPEN 打开 lpReOpenBuff 结构内指定的文件，而不是用 lpFileName 参数
OF_SEARCH 强迫 windows 查找文件——即使指定了特定的路径
OF_SHARE_COMPAT 文件可由多个应用程序打开多次
OF_SHARE_DENY_NONE 可打开文件，以便由其他程序读写
OF_SHARE_DENY_READ 禁止其他程序读写文件内容
OF_SHARE_DENY_WRITE 其他程序可以读文件，但不能写文件
OF_SHARE_EXCLUSIVE 其他任何一个程序都不能再打开这个文件
OF_WRITE 文件以只写模式打开

表 OpenFile-2（OFSTRUCT 出错代码）
十六进制值说明十六进制值说明
1 函数无效 2 文件未找到
3 路径未找到 4 无可用文件句柄
5 拒绝访问 6 句柄无效
7DOS 内存冲突 8 无足够内存完成操作
9 无效块 A 非法环境
B 无效格式 C 无效访问
D 无效数据
F 无效驱动器 10 当前目录无效
11 设备有异 12 没有更多的文件
13 写保护错 14 非法单位
15 驱动器未准备好 16 无效命令
17CRC 校验错 18 无效长度
19 搜索错误 1A 磁盘不兼容 MS-DOS
1B 扇区未找到 1C 缺纸
1D 写错误 1E 读错误
1F 驱动器常规错误 20 共享违例
21 文件锁定违例 22 不正确的磁盘
23 无可用的文件控制块 24 共享缓冲区溢出
32 不支持的设备 33 远程设备不可用
34 重名错误 35 网络路径错误
36 网络忙 37 非法设备
38 命令太多 39 网卡硬件错误
3A 网络响应错误 3B 其他网络错误
3C 远程适配器错误 3D 打印队列满
3E 后台打印缓冲区满 3F 打印取消
40 删除的网络名 41 拒绝网络访问
42 无效设备类型 43 无效网络名
44 名字太多 45 会话太多
46 共享暂停 47 请求未接受
48 重定向暂停 50 文件退出
51 文件控制块重复 52 不能创建
53 中断 24 错误 54 缺少结构
55 已经分配 56 密码无效
57 参数无效 58 网络写错误

Top
OpenFileMapping
OpenFileMapping

VB 声明
Declare Function OpenFileMapping Lib "kernel32" Alias "OpenFileMappingA" (ByVal dwDesiredAccess As Long, ByVal bInheritHandle As Long, ByVal lpName As String) As Long

说明
打开一个现成的文件映射对象
返回值
Long，指定文件映射对象的句柄。零表示出错。会设置 GetLastError
参数表
参数类型及说明
dwDesiredAccessLong，带有前缀 FILE_MAP_???的一个常数。参考 MapViewOfFile 函数的 dwDesiredAccess 参数的说明
bInheritHandleLong，如这个函数返回的句柄能由当前进程启动的新进程继承，则这个参数为 TRUE
lpNameString，指定要打开的文件映射对象名称

Top
QueryDosDevice
QueryDosDevice

VB 声明
Declare Function QueryDosDevice Lib "kernel32" Alias "QueryDosDeviceA" (ByVal lpDeviceName As String, ByVal lpTargetPath As String, ByVal ucchMax As Long) As Long
说明
在 Windows NT 中，DOS 设备名会映射成 NT 系统设备名。该函数可判断当前的设备映射情况
返回值
Long，零表示出错。如执行成功，返回保存到 lpTargetPath 的字符数。会设置 GetLastError
参数表
参数类型及说明
lpDeviceNameString，如果是 vbNullString，那么 lpTargetPath 会载入当前映射的 MS-DOS 名称的一个列表。如果是个 MS-DOS 名，则 lpTargetPath 会载入一个设备映射列表（第一个名字是活动映射，后续的名字是以前尚未删掉的映射）
lpTargetPathString，名称列表，具体取决于 lpDeviceName 参数。这些名字用 NULL 字符分隔。列表最后用两个连续的 NULL 字符中止
ucchMaxLong，lpTargetPath 缓冲区的大小
注解
可用 DefineDosDevice 函数将映射变成 DOS 设备名
适用平台
Windows NT

Top
ReadFile
ReadFile

VB 声明
Declare Function ReadFile Lib "kernel32" Alias "ReadFile" (ByVal hFile As Long, lpBuffer As Any, ByVal nNumberOfBytesToRead As Long, lpNumberOfBytesRead As Long, lpOverlapped

As OVERLAPPED) As Long
说明
从文件中读出数据。与 lread 函数相比，这个函数要明显灵活的多。该函数能够操作通信设备、管道、套接字以及邮槽
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError。如启动的是一次异步读操作，则函数会返回零值，并将 ERROR_IO_PENDING 设置成 GetLastError 的结果。如结果不是零值，但读入的字节数小于 nNumberOfBytesToRead 参数指定的值，表明早已抵达了文件的结尾
参数表
参数类型及说明
hFileLong，文件的句柄
lpBufferAny，用于保存读入数据的一个缓冲区
nNumberOfBytesToReadLong，要读入的字符数
lpNumberOfBytesReadLong，从文件中实际读入的字符数
lpOverlappedOVERLAPPED，如文件打开时指定了 FILE_FLAG_OVERLAPPED，那么必须用这个参数引用一个特殊的结构。那个结构定义了一次异步读取操作。否则，应将这个参数设为 NULL（将函数声明成 ByVal As Long，并传递零值）
注解
并非每种操作系统都支持对每种设备进行异步操作。Windows 95 不支持对一个磁盘文件进行异步读操作（重复读）

Top
ReadFileEx
ReadFileEx

VB 声明
Declare Function ReadFileEx Lib "kernel32" Alias "ReadFileEx" (ByVal hFile As Long, lpBuffer As Any, ByVal nNumberOfBytesToRead As Long, lpOverlapped As OVERLAPPED, ByVal lpCompletionRoutine As Long) As Long
说明
与 ReadFile 相似，只是它只能用于异步读操作，并包含了一个完整的回调
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hFileLong，文件的句柄
lpBufferAny，指定容纳读入数据的一个缓冲区。除非读操作执行完毕，否则不要访问这个缓冲区
nNumberOfBytesToReadLong，要读入的字节数
lpOverlappedOVERLAPPED，定义了一个异步操作的结构。使用这个函数时，结构中的 hEvent 字段会被忽略
lpCompletionRoutineLong，回调函数的返回值

Top

RegCloseKey
RegCloseKey

VB 声明
Declare Function RegCloseKey Lib "advapi32.dll" Alias "RegCloseKey" (ByVal hKey As Long) As Long
说明
关闭系统注册表中的一个项（或键）
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，要关闭的项

Top
RegConnectRegistry
RegConnectRegistry

VB 声明
Declare Function RegConnectRegistry Lib "advapi32.dll" Alias "RegConnectRegistryA" (ByVal lpMachineName As String, ByVal hKey As Long, phkResult As Long) As Long
说明
访问远程系统的部分注册表
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
lpMachineNameString，欲连接的系统。采用“\\计算机名”的形式
hKeyLong，HKEY_LOCAL_MACHINE 或 HKEY_USERS
phkResultLong，用于装载指定项句柄的一个变量

Top
RegCreateKey
RegCreateKey

VB 声明
Declare Function RegCreateKey Lib "advapi32.dll" Alias "RegCreateKeyA" (ByVal hKey As Long, ByVal lpSubKey As String, phkResult As Long) As Long
说明
在指定的项下创建一个新项。如指定的项已经存在，那么函数会打开现有的项
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明

hKeyLong，要打开项的句柄，或者一个标准项名
lpSubKeyString，欲创建的新子项。可同时创建多个项，只需用反斜杠将它们分隔开即可。例如 level1\level2\newkey
phkResultLong，指定一个变量，用于装载新子项的句柄

Top
RegCreateKeyEx
RegCreateKeyEx

VB 声明
Declare Function RegCreateKeyEx Lib "advapi32.dll" Alias "RegCreateKeyExA" (ByVal hKey As Long, ByVal lpSubKey As String, ByVal Reserved As Long, ByVal lpClass As String, ByVal dwOptions As Long, ByVal samDesired As Long, lpSecurityAttributes As SECURITY_ATTRIBUTES, phkResult As Long, lpdwDisposition As Long) As Long
说明
在指定项下创建新项的更复杂的方式。在 Win32 环境中建议使用这个函数。如指定的项已经存在，则函数会打开现有的项
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，一个打开项的句柄，或者一个标准项名
lpSubKeyString，欲创建的新子项的名字
ReservedLong，设为零
lpClassString，项的类名
dwOptionsLong，下述常数为零：REG_OPTION_VOLATILE——这个项不正式保存下来，系统重新启动后会消失
samDesiredLong，带有前缀 KEY_??的一个或多个常数。它们组合起来描述了允许对这个项进行哪些操作
lpSecurityAttributesSECURITY_ATTRIBUTES，对这个项的安全特性进行描述的一个结构（用 ByVal As Long 传递空值）。不适用于 windows 95
phkResultLong，指定用于装载新子项句柄的一个变量
lpdwDispositionLong，用于装载下列某个常数的一个变量：
REG_CREATED_NEW_KEY——新建的一个子项
REG_OPENED_EXISTING_KEY——打开一个现有的项
注解
REG_OPTION_VOLATILE 不适用于 windows 95

Top
RegDeleteKey
RegDeleteKey

VB 声明
Declare Function RegDeleteKey Lib "advapi32.dll" Alias "RegDeleteKeyA" (ByVal hKey As

Long, ByVal lpSubKey As String) As Long
说明
删除现有项下方一个指定的子项
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，一个已打开项的句柄，或者标准项名之一
lpSubKeyString，要删除项的名字。这个项的所有子项也会删除

Top
RegDeleteValue
RegDeleteValue

VB 声明
Declare Function RegDeleteValue Lib "advapi32.dll" Alias "RegDeleteValueA" (ByVal hKey As Long, ByVal lpValueName As String) As Long
说明
删除指定项下方的一个值
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，一个已打开项的句柄，或标准项名之一
lpValueNameString，要删除的值名。可设为 vbNullString 或一个空串，表示删除那个项的默认值

Top
RegEnumKey
RegEnumKey

VB 声明
Declare Function RegEnumKey Lib "advapi32.dll" Alias "RegEnumKeyA" (ByVal hKey As Long, ByVal dwIndex As Long, ByVal lpName As String, ByVal cbName As Long) As Long
说明
枚举指定项的子项。在 Win32 环境中应使用 RegEnumKeyEx
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，一个已打开项的句柄，或者指定一个标准项名
dwIndexLong，欲获取的子项的索引。第一个子项的索引编号为零
lpNameString，用于装载指定索引处项名的一个缓冲区
cbNameLong，lpName 缓冲区的长度

注解
用 RegQueryInfoKey 判断容纳最长那个项所需的缓冲区长度

Top
RegEnumKeyEx
RegEnumKeyEx

VB 声明
Declare Function RegEnumKeyEx Lib "advapi32.dll" Alias "RegEnumKeyExA" (ByVal hKey As Long, ByVal dwIndex As Long, ByVal lpName As String, lpcbName As Long, lpReserved As Long, ByVal lpClass As String, lpcbClass As Long, lpftLastWriteTime As FILETIME) As Long
说明
枚举指定项下方的子项
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，一个已打开项的句柄，或者指定一个标准项名
dwIndexLong，欲获取的子项的索引。第一个子项的索引编号为零
lpNameString，用于装载指定索引处项名的一个缓冲区
lpcbNameLong，指定一个变量，用于装载 lpName 缓冲区的实际长度（包括空字符）。一旦返回，它会设为实际装载到 lpName 缓冲区的字符数量
lpReservedLong，未用，设为零
lpClassString，项使用的类名。可以为 vbNullString
lpcbClassLong，用于装载 lpClass 缓冲区长度的一个变量。一旦返回，它会设为实际装载到缓冲区的字符数量
lpftLastWriteTimeFILETIME，枚举子项上一次修改的时间

Top
RegEnumValue
RegEnumValue

VB 声明
Declare Function RegEnumValue Lib "advapi32.dll" Alias "RegEnumValueA" (ByVal hKey As Long, ByVal dwIndex As Long, ByVal lpValueName As String, lpcbValueName As Long, lpReserved As Long, lpType As Long, lpData As Byte, lpcbData As Long) As Long
说明
枚举指定项的值
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，一个已打开项的句柄，或者指定一个标准项名
dwIndexLong，欲获取值的索引。注意第一个值的索引编号为零

lpValueNameString，用于装载位于指定索引处值名的一个缓冲区
lpcbValueNameLong，用于装载 lpValueName 缓冲区长度的一个变量。一旦返回，它会设为实际载入缓冲区的字符数量
lpReservedLong，未用；设为零
lpTypeLong，用于装载值的类型代码的变量
lpDataByte，用于装载值数据的一个缓冲区
lpcbDataLong，用于装载 lpData 缓冲区长度的一个变量。一旦返回，它会设为实际载入缓冲区的字符数量

Top
RegFlushKey
RegFlushKey

VB 声明
Declare Function RegFlushKey Lib "advapi32.dll" Alias "RegFlushKey" (ByVal hKey As Long) As Long
说明
将对项和它的子项作出的改动实际写入磁盘
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，欲刷新的一个项的句柄，或指定一个标准项名
注解
有些操作系统会将对注册表的修改延迟写入磁盘，以便保持系统的高性能。这个函数的作用就是确定将数据实际写入磁盘。但通常，应尽量避免使用这个函数，因为它可能严重影响一个应用程序的性能

Top
RegGetKeySecurity
RegGetKeySecurity

VB 声明
Declare Function RegGetKeySecurity Lib "advapi32.dll" Alias "RegGetKeySecurity" (ByVal hKey As Long, ByVal SecurityInformation As Long, pSecurityDescriptor As SECURITY_DESCRIPTOR, lpcbSecurityDescriptor As Long) As Long
说明
获取与一个注册表项有关的安全信息
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，指定一个项的句柄，或指定一个标准项名
SecurityInformationLong，对请求信息进行描述的一个标记

pSecurityDescriptorSECURITY_DESCRIPTOR，这个结构用于装载目标项的安全信息
lpcbSecurityDescriptorLong，用于装载 pSecurityDescriptor 缓冲区长度的一个变量。一旦返回，它会设为实际装载到缓冲区的字节数量
适用平台
Windows NT

Top
RegLoadKey
RegLoadKey

VB 声明
Declare Function RegLoadKey Lib "advapi32.dll" Alias "RegLoadKeyA" (ByVal hKey As Long, ByVal lpSubKey As String, ByVal lpFile As String) As Long
说明
从以前用 RegSaveKey 函数创建的一个文件里装载注册表信息
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，HKEY_LOCAL_MACHINE、HKEY_USERS 或者用 RegConnectRegistry 创建的一个子项
lpSubKeyString，要创建的新子项的名字
lpFileString，包含了注册信息的那个文件的名字

Top
RegNotifyChangeKeyValue
RegNotifyChangeKeyValue

VB 声明
Declare Function RegNotifyChangeKeyValue Lib "advapi32.dll" Alias "RegNotifyChangeKeyValue" (ByVal hKey As Long, ByVal bWatchSubtree As Long, ByVal dwNotifyFilter As Long, ByVal hEvent As Long, ByVal fAsynchronus As Long) As Long
说明
注册表项或它的任何一个子项发生变化时，用这个函数提供一种通知机制
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，要监视的一个项的句柄，或者指定一个标准项名
bWatchSubtreeLong，TRUE（非零）表示监视子项以及指定的项
dwNotifyFilterLong，下述常数的一个或多个
REG_NOTIFY_CHANGE_NAME 侦测注册表项名称的变化，以及侦测注册表的创建和删除事件
REG_NOTIFY_CHANGE_ATTRIBUTES 侦测属性的变化

REG_NOTIFY_CHANGE_LAST_SET 侦测上一次修改时间的变化
REG_NOTIFY_CHANGE_SECURITY 侦测对安全特性的改动
hEventLong，一个事件的句柄。如 fAsynchronus 为 False，则这里的设置会被忽略
fAsynchronusLong，如果为零，那么除非侦测到一个变化，否则函数不会返回。否则这个函数会立即返回，而且在发生变化时触发由 hEvent 参数指定的一个事件
适用平台
Windows NT

Top
RegOpenKey
RegOpenKey

VB 声明
Declare Function RegOpenKey Lib "advapi32.dll" Alias "RegOpenKeyA" (ByVal hKey As Long, ByVal lpSubKey As String, phkResult As Long) As Long
说明
打开一个现有的注册表项
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，一个已打开项的句柄，或指定一个标准项名
lpSubKeyString，要打开的项名
phkResultLong，指定一个变量，用于装载（保存）打开注册表项的一个句柄
注解
在 NT 环境下，这个函数会使用默认的安全参数

Top
RegOpenKeyEx
RegOpenKeyEx

VB 声明
Declare Function RegOpenKeyEx Lib "advapi32.dll" Alias "RegOpenKeyExA" (ByVal hKey As Long, ByVal lpSubKey As String, ByVal ulOptions As Long, ByVal samDesired As Long, phkResult As Long) As Long
说明
打开一个现有的项。在 win32 下推荐使用这个函数
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，一个已打开项的句柄，或指定一个标准项名
lpSubKeyString，欲打开注册表项的名字
ulOptionsLong，未用，设为零

samDesiredLong，带有前缀 KEY_??的一个或多个常数。它们的组合描述了允许对这个项进行哪些操作
phkResultLong，用于装载打开项的名字的一个变量

Top
RegQueryInfoKey
RegQueryInfoKey

VB 声明
Declare Function RegQueryInfoKey Lib "advapi32.dll" Alias "RegQueryInfoKeyA" (ByVal hKey As Long, ByVal lpClass As String, lpcbClass As Long, lpReserved As Long, lpcSubKeys As Long, lpcbMaxSubKeyLen As Long, lpcbMaxClassLen As Long, lpcValues As Long, lpcbMaxValueNameLen As Long, lpcbMaxValueLen As Long, lpcbSecurityDescriptor As Long, lpftLastWriteTime As FILETIME) As Long
说明
获取与一个项有关的信息
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码。如一个缓冲区的长度不够，不能容下返回的数据，则函数会返回 ERROR_MORE_DATA
参数表
参数类型及说明
hKeyLong，一个已打开项的句柄，或指定一个标准项名
lpClassString，指定一个字串，用于装载这个注册表项的类名
lpcbClassLong，指定一个变量，用于装载 lpClass 缓冲区的长度。一旦返回，它会设为实际装载到缓冲区的字节数量
lpReservedLong，未用，设为零
lpcSubKeysLong，用于装载（保存）这个项的子项数量的一个变量
lpcbMaxSubKeyLenLong，指定一个变量，用于装载这个项最长一个子项的长度。注意这个长度不包括空中止字符
lpcbMaxClassLenLong，指定一个变量，用于装载这个项之子项的最长一个类名的长度。注意这个长度不包括空中止字符
lpcValuesLong，用于装载这个项的设置值数量的一个变量
lpcbMaxValueNameLenLong，指定一个变量，用于装载这个项之子项的最长一个值名的长度。注意这个长度不包括空中止字符
lpcbMaxValueLenLong，指定一个变量，用于装载容下这个项最长一个值数据所需的缓冲区长度
lpcbSecurityDescriptorLong，装载值安全描述符长度的一个变量
lpftLastWriteTimeFILETIME，指定一个结构，用于容纳该项的上一次修改时间

Top
RegQueryValue
RegQueryValue

VB 声明

Declare Function RegQueryValue Lib "advapi32.dll" Alias "RegQueryValueA" (ByVal hKey As Long, ByVal lpSubKey As String, ByVal lpValue As String, lpcbValue As Long) As Long
说明
取得指定项或子项的默认（未命名）值
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，一个已打开项的句柄，或者指定一个标准项名
lpSubKeyString，要获取一个值的子项。可设为 vbNullString，表示获取 hKey 的值
lpValueString，用于容纳指定项值的一个字串
lpcbValueLong，指定一个变量，用于装载 lpValue 缓冲区的长度。一旦返回，它会设为实际载入缓冲区的字节数量
注解
win32 应用程序应该使用 RegQueryValueEx。lpValue 被定义成一个字串，以维持同 win16 的兼容性（在 win16 中，值全都是字串）

Top
RegQueryValueEx
RegQueryValueEx

VB 声明
Declare Function RegQueryValueEx Lib "advapi32.dll" Alias "RegQueryValueExA" (ByVal hKey As Long, ByVal lpValueName As String, ByVal lpReserved As Long, lpType As Long, lpData As Any, lpcbData As Long) As Long
说明
获取一个项的设置值
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，一个已打开项的句柄，或者指定一个标准项名
lpValueNameString，要获取值的名字
lpReservedLong，未用，设为零
lpTypeLong，用于装载取回数据类型的一个变量
lpDataAny，用于装载指定值的一个缓冲区
lpcbDataLong，用于装载 lpData 缓冲区长度的一个变量。一旦返回，它会设为实际装载到缓冲区的字节数

Top
RegReplaceKey
RegReplaceKey

VB 声明

Declare Function RegReplaceKey Lib "advapi32.dll" Alias "RegReplaceKeyA" (ByVal hKey As Long, ByVal lpSubKey As String, ByVal lpNewFile As String, ByVal lpOldFile As String) As Long
说明
用一个磁盘文件保存的信息替换注册表信息；并创建一个备份，在其中包含当前注册表信息
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，一个已打开项的句柄，或指定一个标准项名
lpSubKeyString，要替换的子项名称。它必须直接位于 HKEY_LOCAL_MACHINE 或 HKEY_USERS 控制项的下方
lpNewFileString，包含了注册表信息的一个文件的名字。这个文件是用 RegSaveKey 函数创建的
lpOldFileString，对当前注册表信息进行备份的一个文件的名字

Top
RegRestoreKey
RegRestoreKey

VB 声明
Declare Function RegRestoreKey Lib "advapi32.dll" Alias "RegRestoreKeyA" (ByVal hKey As Long, ByVal lpFile As String, ByVal dwFlags As Long) As Long
说明
从一个磁盘文件恢复注册表信息
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，一个已打开项的句柄，或者指定一个标准项名
lpFileString，要从中恢复注册表信息的一个文件的名字
dwFlagsLong，0 表示进行常规恢复。REG_WHOLE_HIVE_VOLATILE 表示临时恢复信息（系统重新启动时不保存下来）。在这种情况下，hKey 必须引用 HKEY_LOCAL_MACHINE 或 HKEY_USERS

Top
RegSaveKey
RegSaveKey

VB 声明
Declare Function RegSaveKey Lib "advapi32.dll" Alias "RegSaveKeyA" (ByVal hKey As Long, ByVal lpFile As String, lpSecurityAttributes As SECURITY_ATTRIBUTES) As Long
说明
将一个项以及它的所有子项都保存到一个磁盘文件

返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，一个已打开项的句柄，或指定一个标准项名
lpFileString，要在其中保存注册表信息的一个磁盘文件的名字
lpSecurityAttributesSECURITY_ATTRIBUTES，为保存的信息提供的安全信息。可设为NULL，表示采用默认的安全信息（变成 ByVal As Long，并传递零值）

Top
RegSetKeySecurity
RegSetKeySecurity

VB 声明
Declare Function RegSetKeySecurity Lib "advapi32.dll" Alias "RegSetKeySecurity" (ByVal hKey As Long, ByVal SecurityInformation As Long, pSecurityDescriptor As SECURITY_DESCRIPTOR) As Long
说明
设置指定项的安全特性
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，指定一个项的句柄，或指定一个标准项名
SecurityInformationLong，对要保存的信息进行描述的标志
pSecurityDescriptorSECURITY_DESCRIPTOR，这个结构包含了注册表项新的安全特性设置
适用平台
Windows NT

Top
RegSetValue
RegSetValue

VB 声明
Declare Function RegSetValue Lib "advapi32.dll" Alias "RegSetValueA" (ByVal hKey As Long, ByVal lpSubKey As String, ByVal dwType As Long, ByVal lpData As String, ByVal cbData As Long) As Long
说明
设置指定项或子项的默认值
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，一个已打开项的句柄，或指定一个标准项名

lpSubKeyString，欲对它的值进行设置的一个子项的名字。如指定 vbNullString，表示设置 hKey 的默认值。如指定的子项不存在，则会创建它
dwTypeLong，必须是 REG_SZ
lpDataString，新值
cbDataLong，指定 lpData 的长度，不包括空中止字符

Top
RegSetValueEx
RegSetValueEx

VB 声明
Declare Function RegSetValueEx Lib "advapi32.dll" Alias "RegSetValueExA" (ByVal hKey As Long, ByVal lpValueName As String, ByVal Reserved As Long, ByVal dwType As Long, lpData As Any, ByVal cbData As Long) As Long
说明
设置指定项的值
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，一个已打开项的句柄，或指定一个标准项名
lpValueNameString，要设置值的名字
ReservedLong，未用，设为零
dwTypeLong，要设置的数量类型
lpDataAny，包含数据的缓冲区中的第一个字节
cbDataLong，lpData 缓冲区的长度

Top
RegUnLoadKey
RegUnLoadKey

VB 声明
Declare Function RegUnLoadKey Lib "advapi32.dll" Alias "RegUnLoadKeyA" (ByVal hKey As Long, ByVal lpSubKey As String) As Long
说明
卸载指定的项以及它的所有子项
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，HKEY_LOCAL_MACHINE、HKEY_USERS 或者用 RegConnectRegistry 打开的一个子项
lpSubKeyString，要卸载的子项的名字。必须是早先用 RegLoadKey 函数载入的

Top
RemoveDirectory
RemoveDirectory

VB 声明
Declare Function RemoveDirectory Lib "kernel32" Alias "RemoveDirectoryA" (ByVal lpPathName As String) As Long
说明
删除指定目录
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpPathNameString，要删除的那个目录的名字
注解
在调用这个函数前，目录必须为空

Top
SearchPath
SearchPath

VB 声明
Declare Function SearchPath Lib "kernel32" Alias "SearchPathA" (ByVal lpPath As String, ByVal lpFileName As String, ByVal lpExtension As String, ByVal nBufferLength As Long, ByVal lpBuffer As String, ByVal lpFilePart As String) As Long
说明
查找指定文件
返回值
Long，装载到 lpBuffer 缓冲区的字符数。如缓冲区长度不足，则返回缓冲区必要的长度。零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpPathString，欲搜索的路径。如果为 vbNullString，则采用 windows 搜索路径。参考 OpenFile 函数的 OFSTRUCT 结构中对 OF_SEARCH 标志搜索顺序的介绍
lpFileNameString，要查找的文件名
lpExtensionString，文件扩展名。必须用一个句点符号起头。如文件没有扩展名，或者 lpFileName 包括了扩展名，则设为 vbNullString
nBufferLengthLong，lpBuffer 字串的长度
lpBufferString，用于装载文件名的一个字串
lpFilePartString，指定一个长整数变量，用于装载缓冲文件名部分的地址。在 vb 中不是特别有用
注解
参考 GetFullPathName 函数

Top
SetCurrentDirectory
SetCurrentDirectory

VB 声明
Declare Function SetCurrentDirectory Lib "kernel32" Alias "SetCurrentDirectoryA" (ByVal lpPathName As String) As Long
说明
设置当前目录
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpPathNameString，新当前目录的路径

Top
SetEndOfFile
SetEndOfFile

VB 声明
Declare Function SetEndOfFile Lib "kernel32" Alias "SetEndOfFile" (ByVal hFile As Long) As Long
说明
针对一个打开的文件，将当前文件位置设为文件末尾
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hFileLong，指定一个文件句柄。文件的当前位置设为文件尾，文件会根据需要缩短
注解
如一个文件正作为打开文件映射对象的基准使用，则不要对其应用这个函数

Top
SetFileAttributes
SetFileAttributes

VB 声明
Declare Function SetFileAttributes Lib "kernel32" Alias "SetFileAttributesA" (ByVal lpFileName As String, ByVal dwFileAttributes As Long) As Long
说明
设置文件属性
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表

参数类型及说明
lpFileNameString，要设置其属性的文件名
dwFileAttributesLong，带有 FILE_ATTRIBUTE_??前缀的一个或多个常数

Top
SetFilePointer
SetFilePointer

VB 声明
Declare Function SetFilePointer Lib "kernel32" Alias "SetFilePointer" (ByVal hFile As Long, ByVal lDistanceToMove As Long, lpDistanceToMoveHigh As Long, ByVal dwMoveMethod As Long) As Long
说明
在一个文件中设置当前的读写位置
返回值
Long，返回一个新位置，它采用从文件起始处开始算起的一个字节偏移量。HFILE_ERROR 意味着出错。会设置 GetLastError
参数表
参数类型及说明
hFileLong，系统文件句柄
lDistanceToMoveLong，字节偏移量
lpDistanceToMoveHighLong，指定一个长整数变量，其中包含了要使用的一个高双字偏移。可设为零（将声明变为 ByVal），表示只使用 lDistanceToMove
原文： A long variable containing a high double word offset to use. May be zero (change declaration to ByVal) to use only lDistanceToMove.
dwMoveMethodLong，下述常数之一
FILE_BEGINlOffset 将新位置设为从文件起始处开始算的起的一个偏移
FILE_CURRENTlOffset 将新位置设为从当前位置开始计算的一个偏移
FILE_ENDlOffset 将新位置设为从文件尾开始计算的一个偏移
注解
这个函数与 vb 的 seek 语句类似。不要将函数用于通过 vb 的 open 命令打开的文件。利用这个函数，可以处理那些长度大于 2 ^ 64 字节的大型文件

Top
SetFileTime
SetFileTime

VB 声明
Declare Function SetFileTime Lib "kernel32" Alias "SetFileTime" (ByVal hFile As Long, lpCreationTime As FILETIME, lpLastAccessTime As FILETIME, lpLastWriteTime As FILETIME) As Long
说明
设置文件的创建、访问及上次修改时间
返回值

Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hFileLong，系统文件句柄
lpCreationTimeFILETIME，文件的创建时间
lpLastAccessTimeFILETIME，文件上一次访问的时间
lpLastWriteTimeFILETIME，文件最近一次修改的时间

Top
SetHandleCount
SetHandleCount

VB 声明
Declare Function SetHandleCount Lib "kernel32" Alias "SetHandleCount" (ByVal wNumber As Long) As Long
说明
这个函数不必在 win32 下使用；即使使用，也不会有任何效果
This function is not necessary under Win32 and has no effect.

Top
SetVolumeLabel
SetVolumeLabel

VB 声明
Declare Function SetVolumeLabel Lib "kernel32" Alias "SetVolumeLabelA" (ByVal lpRootPathName As String, ByVal lpVolumeName As String) As Long
说明
设置一个磁盘的卷标（Label）
返回值
Long，TRUE（非零）表示成功，否则返回零。会设置 GetLastError
参数表
参数类型及说明
lpRootPathNameString，磁盘卷的根路径
lpVolumeNameString，指定新卷标。用 vbNullString 指示删除当前卷名

Top
SystemTimeToFileTime
SystemTimeToFileTime

VB 声明
Declare Function SystemTimeToFileTime Lib "kernel32" Alias "SystemTimeToFileTime" (lpSystemTime As SYSTEMTIME, lpFileTime As FILETIME) As Long
说明
根据一个 FILETIME 结构的内容，载入一个 SYSTEMTIME 结构

返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpSystemTimeSYSTEMTIME，包含了系统时间信息的一个结构
lpFileTimeFILETIME，用于装载文件时间的一个结构

Top
UnlockFile
UnlockFile

VB 声明
Declare Function UnlockFile Lib "kernel32" Alias "UnlockFile" (ByVal hFile As Long, ByVal dwFileOffsetLow As Long, ByVal dwFileOffsetHigh As Long, ByVal nNumberOfBytesToUnlockLow As Long, ByVal nNumberOfBytesToUnlockHigh As Long) As Long
说明
解除对一个文件的锁定
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hFileLong，要解锁的文件的句柄
dwFileOffsetLowLong，要解锁文件区域起始位置的低 32 位地址
dwFileOffsetHighLong，要解锁文件区域起始位置的高 32 位地址
nNumberOfBytesToUnlockLowLong，锁定区域中字符数量的低 32 位值
nNumberOfBytesToUnlockHighLong，锁定区域中字符数量的高 32 位值
注解
解锁的文件区域必须与以前锁定时设置的区域完全相符。文件关闭前，应用程序应确定已解除了对任何区域的锁定。参考 LockFile 了解进一步的情况

Top
UnlockFileEx
UnlockFileEx

VB 声明
Declare Function UnlockFileEx Lib "kernel32" Alias "UnlockFileEx" (ByVal hFile As Long, ByVal dwReserved As Long, ByVal nNumberOfBytesToUnlockLow As Long, ByVal nNumberOfBytesToUnlockHigh As Long, lpOverlapped As OVERLAPPED) As Long
说明
解除对一个文件的锁定
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表

参数类型及说明
hFileLong，要解锁的文件的句柄
dwReservedLong，未用；设为零
nNumberOfBytesToUnlockLowLong，锁定区域中字符数量的低 32 位值
nNumberOfBytesToUnlockHighLong，锁定区域中字符数量的高 32 位值
lpOverlappedOVERLAPPED，包含了文件中相对于解锁区域起始处的偏移量
注解
解锁的文件区域必须与以前锁定时设置的区域完全相符。文件关闭前，应用程序应确定已解除了对任何区域的锁定。参考 LockFileEx 了解进一步的情况

Top
UnmapViewOfFile
UnmapViewOfFile

VB 声明
Declare Function UnmapViewOfFile& Lib "kernel32" (ByVal lpBaseAddress As Long)
说明
在当前应用程序的内存地址空间解除对一个文件映射对象的映射
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpBaseAddressLong，指定要解除映射的一个文件映射的基准地址。这个地址是早先用 MapViewOfFile 函数获得的
注解
在 vb 的 api 文本查看器里复制的声明如下，请注意参数类型不同
Declare Function UnmapViewOfFile Lib "kernel32" Alias "UnmapViewOfFile" (lpBaseAddress As Any) As Long

Top
VerFindFile
VerFindFile

VB 声明
Declare Function VerFindFile Lib "version.dll" Alias "VerFindFileA" (ByVal uFlags As Long, ByVal szFileName As String, ByVal szWinDir As String, ByVal szAppDir As String, ByVal szCurDir As String, lpuCurDirLen As Long, ByVal szDestDir As String, lpuDestDirLen As Long) As Long
说明
用这个函数决定一个文件应安装到哪里
返回值
Long，下述值之一：
VFF_CURNEDEST 指出文件现有版本不应在由 szDestDir 参数指定的目录中，那个目录是由函数建议安装新版本的地方

VFF_FILEINUSE 指出现有文件当时正在使用，而且不要在此时删除
VFF_BUFFTOOSMALL 指出 szDestDir 或 szCurDir 缓冲区的一个或两个都太小，不足以容下目录名
参数表
参数类型及说明
uFlagsLong，目前只定义了 VFFF_ISSHAREDFILE，它指出文件可由多个应用程序共享。如指定了这个标志，该函数会建议将文件安装到 windows 或系统目录。如这个参数为零，则函数会建议将文件安装到应用程序目录
szFileNameString，要安装的文件名。注意这个字串不应包括文件的路径
szWinDirString，设为 windows 目录。目录名称是用 GetWindowsDirectory 函数取得的
szAppDirString，应用程序以及所有相关文件的安装目录的完整路径名称
szCurDirString，指定一个字串缓冲区，用于容纳包含了文件现有版本的目录。如文件版本不存在，则在缓冲区中载入源文件的目录。注意必须为这个缓冲区至少分配 MAX_PATH 个字符的空间
lpuCurDirLenLong，szCurDir 缓冲区的长度。这个函数会设为实际装载到缓冲区的字符数量
szDestDirString，指定一个缓冲区，用于装载应在其中安装新文件的一个目录名。注意至少要为这个缓冲区分配 MAX_PATH 个字符的空间
lpuDestDirLenLong，szDestDir 缓冲区的长度。这个变量会设为实际装载到缓冲区的字符数量

Top
VerInstallFile
VerInstallFile

VB 声明
Declare Function VerInstallFile Lib "version.dll" Alias " VerInstallFileA" (ByVal uFlags As Long, ByVal szSrcFileName As String, ByVal szDestFileName As String, ByVal szSrcDir As String, ByVal szDestDir As String, ByVal szCurDir As String, ByVal szTmpFile As String, lpuTmpFileLen As Long) As Long
说明
用这个函数安装一个文件。它利用由 VerFindFile 函数提供的信息决定将文件安装到哪里。这个函数首先会比较两个文件的版本标记。如源文件是最新和兼容的版本，则将源文件复制成目标目录的一个临时文件——如文件处于压缩状态，则同时将其解压。随后，将文件的现有版本删除掉，再对临时文件进行重名处理，使符合目标文件名
返回值
Long，返回一个整数，其中包含了 VerInstallFile 结果常数表里列出的一个或多个常数的组合
参数表
参数类型及说明
uFlagsLong，下述常数值的一个组合：
VIFF_FORCEINSTALL 在不进行版本检查的情况下强制安装源文件
VIFF_DONTDELETEOLD 如文件的现有版本不在目标目录，则不将其删除；如果它在目标目录，就用新文件将其改写（覆盖）
szSrcFileNameString，指定要安装文件的名字。注意其中不应包含文件的路径名
szDestFileNameString，指定文件安装好后应得到的一个正式名称。这个名称与 szSrcFileName

通常都是相同的
szSrcDirString，指定源目录。新版文件将从这里复制到目标目录
szDestDirString，指定目标目录。新版文件将从源目录复制到这里。通常为这个参数使用由 VerFindFile 函数返回的 szDestDir 缓冲区
szCurDirString，包含了文件当前版本的一个目录。通常将由 VerFindFile 函数返回的 szCurDir 缓冲区用于这个参数。如字串为空，则表明系统中不存在文件文件的早期版本
szTmpFileString，用于装载源文件一个临时副本名称的缓冲区。注意必须至少为其分配 MAX_PATH 个字符的空间
lpuTmpFileLenLong，szTmpFile 缓冲区的长度。这个变量会设为装载到缓冲区的实际字符数，其中包括中止用的 NULL 字符。如指定了 VIFF_FORCEINSTALL，且 szTmpFile 不为零，则临时文件会被更名为由 szSrcFileName 参数指定的名字

Top
VerLanguageName
VerLanguageName

VB 声明
Declare Function VerLanguageName Lib "kernel32" Alias "VerLanguageNameA" (ByVal wLang As Long, ByVal szLang As String, ByVal nSize As Long) As Long
说明
这个函数能根据 16 位语言代码获取一种语言的名称。利用版本资源中的语言代码，可以判断出一个文件编写时采用的语言格式。表格“win32 支持的语言代码”对 win32 支持的各种语言代码进行了总结
返回值
Long，装载到 szLang 缓冲区的字符数量。如缓冲区的容量不够，不能容下完整的名称，则函数返回需要的缓冲区大小。零表示出错
参数表
参数类型及说明
wLangLong，语言 ID
szLangString，用于装载指定语言文本名称的一个缓冲区。这个缓冲区至少应预初始化成 nSize+1 个字节的长度
nSizeLong，szLang 缓冲区的大小

Top
VerQueryValue
VerQueryValue

VB 声明
Declare Function VerQueryValue& Lib "version.dll" Alias "VerQueryValueA" (pBlock As Byte, ByVal lpSubBlock As String, lplpBuffer As Long, puLen As Long)
说明
这个函数用于从版本资源中获取信息。调用这个函数前，必须先用 GetFileVersionInfo 函数获取版本资源信息。这个函数会检查资源信息，并将需要的数据复制到一个缓冲区里
返回值

Long，TRUE（非零）表示成功，如请求的信息不存在，或 pBlock 不属于有效版本信息，那就返回一个零
参数表
参数类型及说明
pBlockByte，指定一个内存块第一个字节的地址。这个内存块包含了由 GetFileVersionInfo 函数取回的版本数据信息
lpSubBlockString，下述值之一：
"\"获取文件的 VS_FIXEDFILEINFO 结构
"\VarFileInfo\Translation"获取文件的翻译表
"\StringFileInfo\...."获取文件的字串信息。参考注解
lplpBufferLong，指定一个 Long 变量的地址，该变量用于装载一个缓冲区的地址。请求的版本信息最终会装载到那个缓冲区里
puLenLong，指定由 lplpBuffer 参数引用的数据值的长度，以字节为单位
注解
如 lplpBuffer 参数为"\StringFileInfo\....",缓冲区里就会载入一个整数数组。每一对整数都代表一种语言和代码页，它们描绘了可用的字串信息。通过用下面这三个部分指定一个字串，从而获得 StringFileInfo 字串数据："\StringFileInfo\languagecodepage\stringname"，其中 languagecodepage（语言代码页）是采用字串形式的一个 8 字符十六进制数字。如翻译表中的语言代码页条目是&H04090000，那么这个字串就应该是"04090000"。stringname（字串名）指定的是一个字串名。这个参数的一个例子如下：
"\StringFileInfo\04090000\CompanyName"
其他
从 vb 的 api 文本查看器复制的声明如下：
Declare Function VerQueryValue Lib "version.dll" Alias "VerQueryValue" (pBlock As Any, ByVal lpSubBlock As String, ByVal lplpBuffer As Long, puLen As Long) As Long

Top
WriteFile
WriteFile

VB 声明
Declare Function WriteFile Lib "kernel32" Alias "WriteFile" (ByVal hFile As Long, lpBuffer As Any, ByVal nNumberOfBytesToWrite As Long, lpNumberOfBytesWritten As Long, lpOverlapped As OVERLAPPED) As Long
说明
将数据写入一个文件。该函数比 lwrite 函数要灵活的多。也可将这个函数应用于对通信设备、管道、套接字以及邮槽的处理
返回值
Long，TRUE（非零）表示成功，否则返回零。会设置 GetLastError
参数表
参数类型及说明
hFileLong，一个文件的句柄
lpBufferAny，要写入的一个数据缓冲区
nNumberOfBytesToWriteLong，要写入数据的字节数量。如写入零字节，表示什么都不写入，

但会更新文件的“上一次修改时间”。针对位于远程系统的命名管道，限制在 65535 个字节以内
lpNumberOfBytesWrittenLong，实际写入文件的字节数量
lpOverlappedOVERLAPPED，倘若在指定 FILE_FLAG_OVERLAPPED 的前提下打开文件，这个参数就必须引用一个特殊的结构。那个结构定义了一次异步写操作。否则，该参数应置为空（将声明变为 ByVal As Long，并传递零值）
注解
并不是每种操作系统都支持在任何类型的设备上进行异步操作。windows 95 不支持对磁盘文件的重叠读取操作

Top
WriteFileEx
WriteFileEx

VB 声明
Declare Function WriteFileEx Lib "kernel32" Alias "WriteFileEx" (ByVal hFile As Long, lpBuffer As Any, ByVal nNumberOfBytesToWrite As Long, lpOverlapped As OVERLAPPED, ByVal lpCompletionRoutine As Long) As Long
说明
与 WriteFile 类似，只是它只能用于异步写操作，并包括了一个完整的回调
返回值
Long，非零表示成功，否则返回零。会设置 GetLastError
参数表
参数类型及说明
hFileLong，文件的句柄
lpBufferAny，指定一个缓冲区，其中包含了要写入的数据。除非写操作完成，否则不要访问这个缓冲区
nNumberOfBytesToWriteLong，要写入数据的字节量
lpOverlappedOVERLAPPED，定义了一次异步写操作的结构。使用这个函数时，结构中的 hEvent 字段会被忽略
lpCompletionRoutineLong，回调函数的值
注解
并不是每种操作系统都支持在任何类型的设备上进行异步操作。windows 95 不支持对磁盘文件的重叠读取操作

Top
WritePrivateProfileSection
WritePrivateProfileSection

VB 声明
Declare Function WritePrivateProfileSection Lib "kernel32" Alias "WritePrivateProfileSectionA" (ByVal lpAppName As String, ByVal lpString As String, ByVal lpFileName As String) As Long
说明
为一个初始化文件（.ini）中指定的小节设置所有项名和值

返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpAppNameString，要设置的小节。这个字串不区分大小写
lpStringString，项和值字串的一个列表。每个字串都用一个 NULL 字符分隔，最后一个字串后面用两个 NULL 表示中止。如 lpAppName 指定的小节不存在，则用那个名字新建一个小节，并将其追加到初始化文件的最后。如果存在，则当前的所有项名和值都会被这个缓冲区中指定的数据取代
lpFileNameString，初始化文件的名字。如指定了一个完整路径，而且文件不存在，就会产生错误。如只指定了文件名，而且文件不存在，就在当前的 windows 目录中创建它

Top
WritePrivateProfileString
WritePrivateProfileString

VB 声明
Declare Function WritePrivateProfileString& Lib "kernel32" Alias "WritePrivateProfileStringA" (ByVal lpApplicationName As String, ByVal lpKeyName As Any, ByVal lpString As String, ByVal lpFileName As String)
说明
在初始化文件指定小节内设置一个字串
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpApplicationNameString，要在其中写入新字串的小节名称。这个字串不区分大小写
lpKeyNameAny，要设置的项名或条目名。这个字串不区分大小写。用 vbNullString 可删除这个小节的所有设置项
lpStringString，指定为这个项写入的字串值。用 vbNullString 表示删除这个项现有的字串
lpFileNameString，初始化文件的名字。如果没有指定完整路径名，则 windows 会在 windows 目录查找文件。如果文件没有找到，则函数会创建它
其他
在 vb 的 api 文本查看器里复制的声明如下：
Declare Function WritePrivateProfileString Lib "kernel32" Alias "WritePrivateProfileStringA" (ByVal lpApplicationName As String, ByVal lpKeyName As Any, ByVal lpString As Any, ByVal lpFileName As String) As Long

Top
WriteProfileSection
WriteProfileSection

VB 声明
Declare Function WriteProfileSection Lib "kernel32" Alias "WriteProfileSectionA" (ByVal

lpAppName As String, ByVal lpString As String) As Long
说明
为 Win.ini 初始化文件中一个指定的小节设置所有项名和值
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpAppNameString，要设置的小节。这个字串不区分大小写
lpStringString，项和值字串的一个列表。每个字串都用一个 NULL 字符分隔，最后一个字串后面用两个 NULL 表示中止。如 lpAppName 指定的小节不存在，则用那个名字新建一个小节，并将其追加到初始化文件的最后。如果存在，则当前的所有项名和值都会被这个缓冲区中指定的数据取代
注解
注意对 Win.ini 文件的改动可能影响其他应用程序。如修改了正由其他应用程序使用的小节，一定要向所有窗口都发送一条 WM_WININICHANGE 消息

Top
WriteProfileString
WriteProfileString

VB 声明
Declare Function WriteProfileString Lib "kernel32" Alias "WriteProfileStringA" (ByVal lpszSection As String, ByVal lpszKeyName As String, ByVal lpszString As String) As Long
说明
在 Win.ini 初始化文件指定小节内设置一个字串
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpszSectionString，指定要在其中写入新串的小节。如尚不存在，会创建这个小节。这个字串不区分大小写
lpszKeyNameString，要设置的项名或条目名。这个字串不区分大小写。用 vbNullString 可删除这个小节的所有设置项
lpszStringString，指定为这个项写入的字串值。用 vbNullString 表示删除这个项现有的字串
注解
注意对 Win.ini 文件的改动可能影响其他应用程序。如修改了正由其他应用程序使用的小节，一定要向所有窗口都发送一条 WM_WININICHANGE 消息

Top
网络函数
网络函数，共一页。第一页
WNetAddConnection 创建同一个网络资源的永久性连接
WNetAddConnection2 创建同一个网络资源的连接
WNetAddConnection3 创建同一个网络资源的连接

WNetCancelConnection 结束一个网络连接
WNetCancelConnection2 结束一个网络连接
WNetCloseEnum 结束一次枚举操作
WNetConnectionDialog 启动一个标准对话框，以便建立同网络资源的连接
WNetDisconnectDialog 启动一个标准对话框，以便断开同网络资源的连接
WNetEnumResource 枚举网络资源
WNetGetConnection 获取本地或已连接的一个资源的网络名称
WNetGetLastError 获取网络错误的扩展错误信息
WNetGetUniversalName 获取网络中一个文件的远程名称以及/或者 UNC（统一命名规范）名称
WNetGetUser 获取一个网络资源用以连接的名字
WNetOpenEnum 启动对网络资源进行枚举的过程
完
WNetAddConnection
WNetAddConnection

VB 声明
Declare Function WNetAddConnection Lib "mpr.dll" Alias "WNetAddConnectionA" (ByVal lpszNetPath As String, ByVal lpszPassword As String, ByVal lpszLocalName As String) As Long
说明
创建同一个网络资源的永久性连接
返回值
Long，零表示成功。会设置 GetLastError。如 GetLastError 是 ERROR_EXTENDED_ERROR，则可用 WNetGetLastError 取得额外的错误信息
参数表
参数类型及说明
lpszNetPathString，要连接的网络名
lpszPasswordString，可选的一个密码。如为 vbNullString，表示采用当前用户的默认密码。如为一个空字串，则不用任何密码
lpszLocalNameString，资源的本地名称。（例如，F: 和 LPT1:）

Top
WNetAddConnection2
WNetAddConnection2

VB 声明
Declare Function WNetAddConnection2 Lib "mpr.dll" Alias "WNetAddConnection2A" (lpNetResource As NETRESOURCE, ByVal lpPassword As String, ByVal lpUserName As String, ByVal dwFlags As Long) As Long
说明
创建同一个网络资源的连接
返回值
Long，零表示成功。会设置 GetLastError。如 GetLastError 是 ERROR_EXTENDED_ERROR，则可用 WNetGetLastError 取得额外的错误信息

参数表
参数类型及说明
lpNetResourceNETRESOURCE，在这个结构中设置了下述字段，对要连接的网络资源进行了定义：dwType， lpLocalName （可为 vbNullString）， lpRemoteName， lpProvider （设为 vbNullString 表示用默认提供者）。该结构的其他所有变量都会被忽略
lpPasswordString，可选的一个密码。如为 vbNullString，表示采用当前用户的默认密码。如为一个空字串，则不用任何密码
lpUserNameString，用于连接的用户名。如为 vbNullString，表示使用当前用户
dwFlagsLong，设为零；或指定常数 CONNECT_UPDATE_PROFILE，表示创建永久性连接

Top
WNetAddConnection3
WNetAddConnection3

VB 声明
Declare Function WNetAddConnection3& Lib "mpr.dll" Alias "WNetAddConnection3A" (ByVal hwnd As Long, lpNetResource As NETRESOURCE, ByVal lpPassword As String, ByVal lpUserName As String, ByVal dwFlags As Long)
说明
创建同一个网络资源的连接。这个函数与 WNetAddConnection2 类似，只是它允许我们为这个函数显示的对话框指定一个物主窗口
返回值
Long，
参数表
参数类型及说明
hwndLong，指定一个窗口句柄，用作本函数创建的对话框的父窗口
lpNetResourceNETRESOURCE，在这个结构中设置了下述字段，对要连接的网络资源进行了定义：dwType， lpLocalName （可为 vbNullString）， lpRemoteName， lpProvider （设为 vbNullString 表示用默认提供者）。该结构的其他所有变量都会被忽略
lpPasswordString，可选的一个密码。如为 vbNullString，表示采用当前用户的默认密码。如为一个空字串，则不用任何密码
lpUserNameString，用于连接的用户名。如为 vbNullString，表示使用当前用户
dwFlagsLong，设为零；或指定常数 CONNECT_UPDATE_PROFILE，表示创建永久性连接

Top
WNetCancelConnection
WNetCancelConnection

VB 声明
Declare Function WNetCancelConnection Lib "mpr.dll" Alias "WNetCancelConnectionA" (ByVal lpszName As String, ByVal bForce As Long) As Long
说明
结束一个网络连接
返回值

Long，零表示成功。会设置 GetLastError。如 GetLastError 是 ERROR_EXTENDED_ERROR，则可用 WNetGetLastError 取得额外的错误信息
参数表
参数类型及说明
lpszNameString，已连接资源的远程名称或本地名称
bForceLong，如为 TRUE，表示断开连接（即使连接的资源上正有打开的文件或作业）

Top
WNetCancelConnection2
WNetCancelConnection2

VB 声明
Declare Function WNetCancelConnection2 Lib "mpr.dll" Alias "WNetCancelConnection2A" (ByVal lpName As String, ByVal dwFlags As Long, ByVal fForce As Long) As Long
说明
结束一个网络连接
返回值
Long，零表示成功。会设置 GetLastError。如 GetLastError 是 ERROR_EXTENDED_ERROR，则可用 WNetGetLastError 取得额外的错误信息
参数表
参数类型及说明
lpszNameString，已连接资源的远程名称或本地名称
dwFlagsLong，设为零或 CONNECT_UPDATE_PROFILE。如为零，而且建立的是永久性连接，则在 windows 下次重新启动时仍会重新连接
fForceLong，如为 TRUE，表示强制断开连接（即使连接的资源上正有打开的文件或作业）

Top
WNetCloseEnum
WNetCloseEnum

VB 声明
Declare Function WNetCloseEnum Lib "mpr.dll" Alias "WNetCloseEnum" (ByVal hEnum As Long) As Long
说明
结束一次枚举操作
返回值
Long，零表示成功。会设置 GetLastError。如 GetLastError 是 ERROR_EXTENDED_ERROR，则可用 WNetGetLastError 取得额外的错误信息
参数表
参数类型及说明
hEnumLong，由 WNetOpenEnum 函数返回的一个枚举句柄

Top
WNetConnectionDialog

WNetConnectionDialog

VB 声明
Declare Function WNetConnectionDialog Lib "mpr.dll" Alias "WNetConnectionDialog" (ByVal hwnd As Long, ByVal dwType As Long) As Long
说明
启动一个标准对话框，以便建立同网络资源的连接
返回值
Long，零表示成功。如用户取消了操作，则返回-1。会设置 GetLastError。如 GetLastError 是 ERROR_EXTENDED_ERROR，则可用 WNetGetLastError 取得额外的错误信息
参数表
参数类型及说明
hwndLong，指定要成为对话框父窗口的一个窗口的句柄
dwTypeLong，设成 RESOURCETYPE_DISK，浏览磁盘资源

Top
WNetDisconnectDialog
WNetDisconnectDialog

VB 声明
Declare Function WNetDisconnectDialog Lib "mpr.dll" Alias "WNetDisconnectDialog" (ByVal hwnd As Long, ByVal dwType As Long) As Long
说明
启动一个标准对话框，以便断开同网络资源的连接
返回值
Long，零表示成功。如用户取消了操作，则返回-1。会设置 GetLastError。如 GetLastError 是 ERROR_EXTENDED_ERROR，则可用 WNetGetLastError 取得额外的错误信息
参数表
参数类型及说明
hwndLong，指定要成为对话框父窗口的一个窗口的句柄
dwTypeLong，设成 RESOURCETYPE_DISK 或 RESOURCETYPE_PRINT，决定要断开的是磁盘还是打印机资源

Top
WNetEnumResource
WNetEnumResource

VB 声明
Declare Function WNetEnumResource Lib "mpr.dll" Alias "WNetEnumResourceA" (ByVal hEnum As Long, lpcCount As Long, lpBuffer As Any, lpBufferSize As Long) As Long
说明
枚举网络资源
返回值
Long，零表示成功。ERROR_NO_MORE_ITEMS 表示不剩下可以枚举的条目。

ERROR_MORE_DATA 表示条目不能装入 lpBuffer。会设置 GetLastError。如 GetLastError 是 ERROR_EXTENDED_ERROR，则可用 WNetGetLastError 取得额外的错误信息
参数表
参数类型及说明
hEnumLong，从 WNetOpenEnum 函数返回的一个句柄
lpcCountLong，最初设为要枚举的最大资源数量；或设为-1，表示枚举尽可能多的资源。一旦返回，就会设为实际枚举的资源数量
lpBufferAny，通常是一个字节缓冲区的首字节。该缓冲区装载了枚举信息（可按引用声明为 Byte）
lpBufferSizeLong，以字节为单位指定 lpBuffer 数组的长度。如缓冲区不够大，则设为需要的缓冲区长度
注解
枚举网络条目时，最好用 vb 一次枚举一个资源。尽量不要使用这个函数同时枚举许多网络资源的功能

Top
WNetGetConnection
WNetGetConnection

VB 声明
Declare Function WNetGetConnection Lib "mpr.dll" Alias "WNetGetConnectionA" (ByVal lpszLocalName As String, ByVal lpszRemoteName As String, cbRemoteName As Long) As Long
说明
获取本地或已连接的一个资源的网络名称
返回值
Long，零表示成功。会设置 GetLastError。如 GetLastError 是 ERROR_EXTENDED_ERROR，则可用 WNetGetLastError 取得额外的错误信息
参数表
参数类型及说明
lpszLocalNameString，本地设备的名字
lpszRemoteNameString，指定一个字串缓冲区，用于装载设备的资源名称
cbRemoteNameLong，lpszRemoteName 缓冲区的字符数量。如缓冲区不够大，则设为需要的缓冲区长度

Top
WNetGetLastError
WNetGetLastError

VB 声明
Declare Function WNetGetLastError Lib "mpr.dll" Alias "WNetGetLastErrorA" (lpError As Long, ByVal lpErrorBuf As String, ByVal nErrorBufSize As Long, ByVal lpNameBuf As String, ByVal nNameBufSize As Long) As Long
说明
获取网络错误的扩展错误信息

返回值
Long，零表示成功。ERROR_INVALID_ADDRESS 表示缓冲区无效
参数表
参数类型及说明
lpErrorLong，指定一个变量，用于装载网络错误代码。具体的代码由网络供应商决定
lpErrorBufString，指定一个字串缓冲区，用于装载网络错误的说明
nErrorBufSizeLong，lpErrorBuf 缓冲区包含的字符数量
lpNameBufString，用于装载网络供应商名字的字串缓冲区
nNameBufSizeLong，lpNameBuf 缓冲区的字符数量

Top
WNetGetUniversalName
WNetGetUniversalName

VB 声明
Declare Function WNetGetUniversalName Lib "mpr" Alias "WNetGetUniversalNameA" (ByVal lpLocalPath As String, ByVal dwInfoLevel As Long, lpBuffer As Any, lpBufferSize As Long) As Long
说明
获取网络中一个文件的远程名称以及/或者 UNC（统一命名规范）名称。例如，假设一个已连接的远程驱动器是\\othersystem\CDrive，它对应的本地驱动器是 F:，而且在它的子目录 temp 中包含了文件 xyz.doc。那么运算结果如下：LocalPath xyz.doc 或 f:\temp\xyz.doc（或者文件的任何相对路径名）
UNC 名称： \\othersystem\CDrive\temp\xyz.doc
连接名称： \\othersystem\CDrive
剩余名称： \temp\xyz.doc
它们分别对应于由这个函数装载的 REMOTE_NAME_INFO 结构的字段，对该结构的定义如下：

```
Type REMOTE_NAME_INFO
    pUniversalName As Long
    pConnectionName As Long
    pRemainingPath As Long
End Type
```

返回值
Long，零表示成功。会设置 GetLastError。如 GetLastError 是 ERROR_EXTENDED_ERROR，则可用 WNetGetLastError 取得额外的错误信息
参数表
参数类型及说明
lpLocalPathString，磁盘文件的名字
dwInfoLevelLong，下述常数之一：
UNIVERSAL_NAME_INFO_LEVEL 只设置 pUniversalName 字段
REMOTE_NAME_INFO_LEVEL 设置 REMOTE_NAME_INFO 结构中的所有三个字段
lpBufferAny，指定用于装载 UNC 信息的一个缓冲区。缓冲区起点与一个 REMOTE_NAME_INFO 结构对应

lpBufferSizeLong，以字节为单位指定 lpBuffer 缓冲区的长度。如缓冲区不够大，则设为需要的缓冲区长度

Top
WNetGetUser
WNetGetUser

VB 声明
Declare Function WNetGetUser Lib "mpr.dll" Alias "WNetGetUserA" (ByVal lpName As String, ByVal lpUserName As String, lpnLength As Long) As Long
说明
获取一个网络资源用以连接的名字
返回值
Long，零表示成功。会设置 GetLastError。如 GetLastError 是 ERROR_EXTENDED_ERROR，则可用 WNetGetLastError 取得额外的错误信息
参数表
参数类型及说明
lpNameString，指定已连接资源的远程名称或本地名称。用 vbNullString 获取当前用户的名字
lpUserNameString，用于装载用户名的一个字串缓冲区
lpnLengthLong，lpUserName 缓冲区的长度。如缓冲区不够大，则自动设为需要的缓冲区长度

Top
WNetOpenEnum
WNetOpenEnum

VB 声明
Declare Function WNetOpenEnum Lib "mpr.dll" Alias "WNetOpenEnumA" (ByVal dwScope As Long, ByVal dwType As Long, ByVal dwUsage As Long, lpNetResource As NETRESOURCE, lphEnum As Long) As Long
说明
启动对网络资源进行枚举的过程。这个函数会返回由 WNetEnumResource 函数用于枚举资源所用的一个句柄
返回值
Long，零表示成功。会设置 GetLastError。如 GetLastError 是 ERROR_EXTENDED_ERROR，则可用 WNetGetLastError 取得额外的错误信息
参数表
参数类型及说明
dwScopeLong，指定要枚举的资源范围。可设为下述常数之一：
RESOURCE_CONNECTED 枚举已连接的资源（忽略 dwUsage）
RESOURCE_GLOBALNET 枚举所有资源
RESOURCE_REMEMBERED 只枚举永久性连接
dwTypeLong，下述常数之一

RESOURCE_ANY 枚举所有类型的网络资源
RESOURCE_DISK 枚举磁盘资源
RESOURCE_PRINT 枚举打印资源
dwUsageLong，可设为零，表示枚举所有资源；或设为下述常数的一个或两个：
RESOURCEUSAGE_CONNECTABLE 只枚举那些能够连接的资源
RESOURCEUSAGE_CONTAINER 只枚举包含了其他资源的资源
lpNetResourceNETRESOURCE，这个结构指定了一个容器资源。该函数会枚举包含于这里指定的某个指定资源内的资源。如设为 NULL（把声明变成 ByVal As Long），那么函数会枚举顶级网络资源。倘若在 dwScope 参数里没有指定 RESOURCE_GLOBALNET，那么必须为 NULL
lphEnumLong，指定一个变量，用于装载一个枚举句柄。该句柄由 WNetEnumResource 函数使用。必须用 WNetCloseEnum 函数将其清除

Top
菜单函数
菜单函数：共三页。第一页，第二页，第三页
AppendMenu 在指定的菜单里添加一个菜单项
CheckMenuItem 复选或撤消复选指定的菜单条目
CheckMenuRadioItem 指定一个菜单条目被复选成“单选”项目
CreateMenu 创建新菜单
CreatePopupMenu 创建一个空的弹出式菜单
DeleteMenu 删除指定的菜单条目
DestroyMenu 删除指定的菜单
DrawMenuBar 为指定的窗口重画菜单
EnableMenuItem 允许或禁止指定的菜单条目
GetMenu 取得窗口中一个菜单的句柄
GetMenuCheckMarkDimensions 返回一个菜单复选符的大小
GetMenuContextHelpId 取得一个菜单的帮助场景 ID
GetMenuDefaultItem 判断菜单中的哪个条目是默认条目
GetMenuItemCount 返回菜单中条目（菜单项）的数量
GetMenuItemID 返回位于菜单中指定位置处的条目的菜单 ID
GetMenuItemInfo 取得（接收）与一个菜单条目有关的特定信息
AppendMenu
AppendMenu

VB 声明
Declare Function AppendMenu Lib "user32" Alias "AppendMenuA" (ByVal hMenu As Long, ByVal wFlags As Long, ByVal wIDNewItem As Long, ByVal lpNewItem As Any) As Long
说明
在指定的菜单里添加一个菜单项
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明

hMenuLong，菜单句柄
wFlagsLong，参考 ModifyMenu 函数中的菜单常数标志定义表，其中列出了允许使用的所有常数
wIDNewItemLong，指定菜单条目的新命令 ID。如果在 wFlags 参数中指定了 MF_POPUP 字段，那么这应该是指向一个弹出式菜单的句柄
lpNewItemString（相应的 vb 声明见注解），如果在 wFlags 参数中指定了 MF_STRING 标志，这就代表在菜单中设置的字串。如设置了 MF_BITMAP 标志，这就代表一个 Long 型变量，其中包含了一个位图句柄。如设置了 MF_OWNERDRAW，这个值就会包括在 DRAWITEMSTRUCT 和 MEASUREITEMSTRUCT 结构中，在条目需要重画的时候由 windows 发送出去
注解
Declare Function AppendMenu& Lib "user32" Alias "AppendMenuA" (ByVal hMenu As Long, ByVal wFlags As Long, ByVal wIDNewItem As Long, ByVal lpNewItem As String)

Top
CheckMenuItem
CheckMenuItem

VB 声明
Declare Function CheckMenuItem Lib "user32" Alias "CheckMenuItem" (ByVal hMenu As Long, ByVal wIDCheckItem As Long, ByVal wCheck As Long) As Long
说明
复选或撤消复选指定的菜单条目
返回值
Long，如条目的前一个状态是“复选”，就返回 MF_CHECKED，如果是“未复选”，就返回 MF_UNCHECKED。如指定的菜单条目不存在就返回-1
参数表
参数类型及说明
hMenuLong，菜单句柄
wIDCheckItemLong，欲复选或撤消复选的菜单条目的标识符。如果在 wCheck 中指定了 MF_BYCOMMAND 标志，这个参数就用于指定要改变的菜单条目的命令 ID。如果设置了 MF_BYPOSITION 标志，这个参数就用于指定条目在菜单中的位置（第一个条目的位置是 0）
wCheckLong，参考 ModifyMenu 函数中的菜单常数标志定义表，其中列出了允许使用的所有常数。针对这个函数，只能指定下述常数：MF_BYCOMMAND，MF_BYPOSITION，MF_CHECKED 以及 MF_UNCHECKED
注解
在 vb 里使用：由这个函数做出的改动可以正常发挥作用，但不会由 vb 菜单的 checked 属性反映出来

Top
CheckMenuRadioItem
CheckMenuRadioItem

VB 声明

Declare Function CheckMenuRadioItem Lib "user32" Alias "CheckMenuRadioItem" (ByVal hMenu As Long, ByVal un1 As Long, ByVal un2 As Long, ByVal un3 As Long, ByVal un4 As Long) As Long
说明
指定一个菜单条目被复选成“单选”项目。与单选钮相似，一个特定的组中只能有一个项目被选中。这个组别既可按位置定义，也可按菜单 ID 定义。复选的项目会显示一个圆形的样式复选符号（●），而不是一个标准的复选符号（√）
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hMenuLong，菜单句柄
un1Long，组内第一个位置或菜单 ID
un2Long，组内最后一个位置或菜单 ID
un3Long，欲复选的位置或菜单 ID
un4Long，下述标志之一 ：如 un1，un2，un3 引用菜单条目的位置（第一个肯定在位置 0 处），就设为 MF_BYPOSITION；如它们引用的是菜单 ID，则设为 MF_BYCOMMAND
注解
在 vb 里使用：由这个函数做出的改动可以正常发挥作用，但不会由 vb 菜单的 checked 属性反映出来

Top
CreateMenu
CreateMenu

VB 声明
Declare Function CreateMenu Lib "user32" Alias "CreateMenu" () As Long
说明
创建新菜单
返回值
Long，如成功则返回新的顶级菜单的句柄；零意味着错误
注解
最开始创建时，菜单是空的。可用菜单 api 函数插入菜单条目。一旦菜单不再需要，记住用 DestroyMenu 将其删除

Top
CreatePopupMenu
CreatePopupMenu

VB 声明
Declare Function CreatePopupMenu Lib "user32" Alias "CreatePopupMenu" () As Long
说明
创建一个空的弹出式菜单。可用 AppendMenu 或 InsertMenu 函数在窗口中添加条目，或者为一个现成的菜单添加弹出式菜单，并在新建的菜单中添加条目

返回值
Long，如成功，返回一个菜单句柄；零意味着错误
注解
并不推荐用这个函数来创建备用的 vb 菜单，除非是为 TrackPopupMenu 函数生成菜单。这个窗口中使用的命令 ID 必须与现有 vb 菜单控件的 ID 相符。或者用一个子类处理控件进行管理

Top
DeleteMenu
DeleteMenu

VB 声明
Declare Function DeleteMenu Lib "user32" Alias "DeleteMenu" (ByVal hMenu As Long, ByVal nPosition As Long, ByVal wFlags As Long) As Long
说明
删除指定的菜单条目（在 vb 里使用：强烈建议用 vb 菜单的 visible 属性从菜单中删除条目。如使用这个函数，会造成指定菜单其他菜单条目的 visible 属性错误的影响菜单条目）
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hMenuLong，菜单句柄
nPositionLong，欲删除菜单条目的标识符。如在 wFlags 中设置了 MF_BYCOMMAND 标志，这个参数就代表要改变的菜单条目的命令 ID。如设置了 MF_BYPOSITION 标志，这个参数就代表条目在菜单中的位置（头一个条目肯定是零）
wFlagsLong，MF_BYPOSITION 或 MF_BYCOMMAND，具体由 nPosition 参数决定
注解
如条目连接了一个弹出式菜单，就会清除弹出式菜单。用 RemoveMenu 函数清除一个弹出式菜单条目，同时不影响整个弹出式菜单

Top
DestroyMenu
DestroyMenu

VB 声明
Declare Function DestroyMenu Lib "user32" Alias "DestroyMenu" (ByVal hMenu As Long) As Long
说明
删除指定的菜单。如菜单属于另一个菜单的一部分，或直接分配给一个窗口，那么菜单会在窗口清除后被自动删除
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明

hMenuLong，欲删除菜单的句柄
注解
这个函数通常用于 CreateMenu 和 CreatePopupMenu 函数创建的菜单

Top
DrawMenuBar
DrawMenuBar

VB 声明
Declare Function DrawMenuBar Lib "user32" Alias "DrawMenuBar" (ByVal hwnd As Long) As Long
说明
为指定的窗口重画菜单。用 api 函数改变一个窗口菜单的内容时，就要用到这个函数
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，要重画菜单栏的窗口的句柄
注解
在 vb 里很少有必要用到这个函数，因为不应该用 api 函数来改变一个窗口的顶级菜单栏

Top
EnableMenuItem
EnableMenuItem

VB 声明
Declare Function EnableMenuItem Lib "user32" Alias "EnableMenuItem" (ByVal hMenu As Long, ByVal wIDEnableItem As Long, ByVal wEnable As Long) As Long
说明
允许或禁止指定的菜单条目（在 vb 里使用：由这个函数做出的改动可以正常发挥作用，但不会由 vb 菜单的 enabled 属性反映出来）
返回值
Long，
参数表
参数类型及说明
hMenuLong，菜单句柄
wIDEnableItemLong，欲允许或禁止的一个菜单条目的标识符。如果在 wEnable 参数中设置了 MF_BYCOMMAND 标志，这个参数就代表欲改变菜单条目的命令 ID。如设置的是 MF_BYPOSITION，则这个参数代表菜单条目在菜单中的位置（第一个条目肯定是零）
wEnableLong，参考 ModifyMenu 函数中的菜单常数标志定义表，其中列出了允许使用的所有常数。对于这个函数，只能指定下述常数：MF_BYCOMMAND，MF_BYPOSITION，MF_ENABLED，MF_DISABLED 以及 MF_GRAYED
注解
如指定的菜单条目依附了一个弹出式菜单，那么整个弹出式菜单都会受到影响

Top
GetMenu
GetMenu

VB 声明
Declare Function GetMenu Lib "user32" Alias "GetMenu" (ByVal hwnd As Long) As Long
说明
取得窗口中一个菜单的句柄
返回值
Long，依附于指定窗口的一个菜单的句柄（如果有菜单）；否则返回零
参数表
参数类型及说明
hwndLong，窗口句柄。对于 vb，这应该是一个窗体句柄。注意可能不是子窗口的句柄

Top
GetMenuCheckMarkDimensions
GetMenuCheckMarkDimensions

VB 声明
Declare Function GetMenuCheckMarkDimensions Lib "user32" Alias "GetMenuCheckMarkDimensions" () As Long
说明
返回一个菜单复选符的大小。参考 SetMenuItemBitmaps 以进一步了解如何使用这个函数
返回值
Long，高字（高 16 位）指定菜单复选符的高度，以像素为单位表示；低字代表宽度

Top
GetMenuContextHelpId
GetMenuContextHelpId

VB 声明
Declare Function GetMenuContextHelpId Lib "user32" Alias "GetMenuContextHelpId" (ByVal hMenu As Long) As Long
说明
取得用 SetMenuContextHelpId 函数分配给一个菜单的帮助场景 ID
返回值
Long，如果存在，就返回帮助场景 ID；否则返回零
参数表
参数类型及说明
hMenuLong，菜单句柄

Top
GetMenuDefaultItem

GetMenuDefaultItem

VB 声明
Declare Function GetMenuDefaultItem Lib "user32" Alias "GetMenuDefaultItem" (ByVal hMenu As Long, ByVal fByPos As Long, ByVal gmdiFlags As Long) As Long
说明
允许我们判断菜单中的哪个条目是默认条目。这个条目相应的转换成双击菜单时执行的操作
返回值
Long，默认菜单条目的位置或标识符。如未发现默认条目，就返回-1
参数表
参数类型及说明
hMenuLong，菜单的句柄
fByPosLong，如果为 TRUE，表示接收条目在菜单中的位置；FALSE 表示接收它的菜单 ID
gmdiFlagsLong，下述标志之一：
GMDI_GOINTOPOPUPS 如默认条目是弹出菜单，这个函数就会在其中搜索一个默认的菜单条目
GMDI_USEDISABLED 指明自己在搜索过程中不想跳过被禁止的菜单条目

Top
GetMenuItemCount
GetMenuItemCount

VB 声明
Declare Function GetMenuItemCount Lib "user32" Alias "GetMenuItemCount" (ByVal hMenu As Long) As Long
说明
返回菜单中条目（菜单项）的数量
返回值
Long，菜单中的条目数量；-1 意味着出错。会设置 GetLastError
参数表
参数类型及说明
hMenuLong，目标菜单的句柄

Top
GetMenuItemID
GetMenuItemID

VB 声明
Declare Function GetMenuItemID Lib "user32" Alias "GetMenuItemID" (ByVal hMenu As Long, ByVal nPos As Long) As Long
说明
返回位于菜单中指定位置处的条目的菜单 ID
返回值
Long，指定条目的菜单 ID。如条目属于一个弹出式菜单，就返回-1；如指定的条目属于一

个分隔符（比如一条分隔线）则返回 0
参数表
参数类型及说明
hMenuLong，菜单的句柄
nPosLong，条目在菜单中的位置。第一个条目的编号是 0

Top
GetMenuItemInfo
GetMenuItemInfo

VB 声明
Declare Function GetMenuItemInfo Lib "user32" Alias "GetMenuItemInfoA" (ByVal hMenu As Long, ByVal un As Long, ByVal b As Boolean, lpMenuItemInfo As MENUITEMINFO) As Long
说明
用一个 MENUITEMINFO 结构取得（接收）与一个菜单条目有关的特定信息
返回值
Long，TRUE（非零）表示成功，否则返回零。会设置 GetLastError
参数表
参数类型及说明
hMenuLong，菜单的句柄
unLong，菜单条目的菜单 ID 或位置
bBoolean，如 un 指定的是条目位置，就为 TRUE；如指定的是一个菜单 ID，则为 FALSE
lpMenuItemInfoMENUITEMINFO，这个结构用于装载请求的信息

Top
GetMenuItemRect
GetMenuItemRect

VB 声明
Declare Function GetMenuItemRect Lib "user32" Alias "GetMenuItemRect" (ByVal hWnd As Long, ByVal hMenu As Long, ByVal uItem As Long, lprcItem As RECT) As Long
说明
在一个矩形中装载指定菜单条目的屏幕坐标信息
返回值
Long，TRUE（非零）表示成功，否则返回零。会设置 GetLastError
参数表
参数类型及说明
hWndLong，包含指定菜单或弹出式菜单的一个窗口的句柄
hMenuLong，菜单的句柄
uItemLong，欲检查的菜单条目的位置或菜单 ID
lprcItemRECT，在这个结构中装载菜单条目的位置及大小（采用屏幕坐标表示）

Top
GetMenuState

GetMenuState

VB 声明
Declare Function GetMenuState Lib "user32" Alias "GetMenuState" (ByVal hMenu As Long, ByVal wID As Long, ByVal wFlags As Long) As Long
说明
取得与指定菜单条目状态有关的信息
返回值
Long，在 api32.txt 文件的常数定义的一系列标志的组合，请看下表。如条目是个弹出式菜单，那么结构的最低字节就包含了状态标志，而第二个字节包含条目在弹出式菜单中的数量
MF_HILITE 菜单条目加亮显示（处于选定状态）
MF_CHECKED 菜单条目处于复选状态
MF_DISABLED 菜单条目处于禁止状态
MF_GRAYED 菜单条目以灰色显示，处于禁用状态
MF_MENUBARBREAK 为这个条目指定一条分隔线。参考 ModifyMenu 函数
MF_MENUBREAK 为这个条目指定一个菜单分隔标志。参考 ModifyMenu 函数
MF_SEPARATOR 菜单条目是一个分隔符
参数表
参数类型及说明
hMenu 菜单句柄
wID 欲检查的菜单条目的标识符。如果在 wFlags 参数中设置了 MF_BYCOMMAND 标志，这个参数就用于指定要改变的菜单条目的命令 ID。如果设置的是 MF_BYPOSITION 标志，这个参数就用于指定条目在菜单中的位置（第一个条目的位置为 0）
wFlags 常数 MF_BYCOMMAND 或 MF_BYPOSITION，取决于 wID 参数的设置

Top
GetMenuString
GetMenuString

VB 声明
Declare Function GetMenuString Lib "user32" Alias "GetMenuStringA" (ByVal hMenu As Long, ByVal wIDItem As Long, ByVal lpString As String, ByVal nMaxCount As Long, ByVal wFlag As Long) As Long
说明
取得指定菜单条目的字串
返回值
Long，在 lpString 中返回的字串的长度（不包括空中止字符）。零意味着出错
参数表
参数类型及说明
hMenuLong，菜单句柄
wIDItemLong，欲接收的菜单条目的标识符。如果在 wFlags 参数中设置了 MF_BYCOMMAND 标志，这个参数就用于指定要改变的菜单条目的命令 ID。如果设置的是 MF_BYPOSITION 标志，这个参数就用于指定条目在菜单中的位置（第一个条目的位置为 0）

lpStringString，指定一个预先定义好的字串缓冲区，以便为菜单条目装载字串
nMaxCountLong，载入 lpString 缓冲区中的最大字符数量+1
wFlagLong，常数 MF_BYCOMMAND 或 MF_BYPOSITION，取决于 wID 参数的设置

Top
GetSubMenu
GetSubMenu

VB 声明
Declare Function GetSubMenu Lib "user32" Alias "GetSubMenu" (ByVal hMenu As Long, ByVal nPos As Long) As Long
说明
取得一个弹出式菜单的句柄，它位于菜单中指定的位置
返回值
Long，位于指定位置的弹出式菜单的句柄（如果有的话）；否则返回零
参数表
参数类型及说明
hMenuLong，菜单的句柄
nPosLong，条目在菜单中的位置。第一个条目的编号为 0

Top
GetSystemMenu
GetSystemMenu

VB 声明
Declare Function GetSystemMenu Lib "user32" Alias "GetSystemMenu" (ByVal hwnd As Long, ByVal bRevert As Long) As Long
说明
取得指定窗口的系统菜单的句柄。在 vb 环境，“系统菜单”的正式名称为“控制菜单”，即单击窗口左上角的控制框时出现的菜单
返回值
Long，如执行成功，返回系统菜单的句柄；零意味着出错。如 bRevert 设为 TRUE，也会返回零（简单的恢复原始的系统菜单）
参数表
参数类型及说明
hwndLong，窗口的句柄
bRevertLong，如设为 TRUE，表示接收原始的系统菜单
注解
在 vb 里使用：系统菜单会向窗口发送一条 WM_SYSCOMMAND 消息，而不是 WM_COMMAND 消息

Top
HiliteMenuItem
HiliteMenuItem

VB 声明
Declare Function HiliteMenuItem Lib "user32" Alias "HiliteMenuItem" (ByVal hwnd As Long, ByVal hMenu As Long, ByVal wIDHiliteItem As Long, ByVal wHilite As Long) As Long
说明
控制顶级菜单条目的加亮显示状态
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hwndLong，拥有顶级菜单的一个窗口的句柄
hMenuLong，hwnd 窗口的顶级菜单的句柄
wIDHiliteItemLong，欲加亮或撤消加亮的菜单条目的标识符。倘若在 wHilite 参数中设置了 MF_BYCOMMAND 标志，这个参数就用于指定要改变的菜单条目的命令 ID。如果设置的是 MF_BYPOSITION 标志，这个参数就用于指定条目在菜单中的位置（第一个条目的位置为 0）
wHiliteLong，一系列常数标志的组合。其中包括 MF_BYCOMMAND 或 MF_BYPOSITION，指出要改变的条目。也包括 MF_HILITE，用于设置加亮状态；或者 MF_UNHILITE，用于撤消加亮状态

Top
InsertMenu
InsertMenu

VB 声明
Declare Function InsertMenu Lib "user32" Alias "InsertMenuA" (ByVal hMenu As Long, ByVal nPosition As Long, ByVal wFlags As Long, ByVal wIDNewItem As Long, ByVal lpNewItem As Any) As Long
说明
在菜单的指定位置处插入一个菜单条目，并根据需要将其他条目向下移动
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hMenuLong，菜单的句柄
nPositionLong，定义了新条目插入点的一个现有菜单条目的标志符。如果在 wFlags 中指定了 MF_BYCOMMAND 标志，这个参数就代表欲改变的菜单条目的命令 ID。如设置的是 MF_BYPOSITION 标志，这个参数就代表菜单条目在菜单中的位置，第一个条目的位置为零
wFlagsLong，一系列常数标志的组合。参考 ModifyMenu
wIDNewItemLong，指定菜单条目的新菜单 ID。如果在 wFlags 中指定了 MF_POPUP 标志，就应该指定弹出式菜单的一个句柄
lpNewItem 如果在 wFlags 参数中设置了 MF_STRING 标志，就代表要设置到菜单中的字串（String）。如设置的是 MF_BITMAP 标志，就代表一个 Long 型变量，其中包含了一个位图

句柄
注解
在 vb 里使用：这个函数做出的许多改变都可以正常发挥作用，但却不能由 vb 菜单对象反映出来。添加的命令 ID 必须能由 vb 菜单系统识别

Top
InsertMenuItem
InsertMenuItem

VB 声明
Declare Function InsertMenuItem Lib "user32" Alias "InsertMenuItemA" (ByVal hMenu As Long, ByVal un As Long, ByVal bool As Boolean, ByVal lpcMenuItemInfo As MENUITEMINFO) As Long
说明
用一个 MENUITEMINFO 结构指定的特征插入一个新菜单条目
返回值
Long，TRUE（非零）表示成功，否则返回零。会设置 GetLastError
参数表
参数类型及说明
hMenuLong，菜单的句柄
unLong，菜单条目的菜单 ID。新条目会插入由这个参数指定的项目之前
boolBoolean，如 un 指定的是条目的位置，就为 TRUE，如指定的是菜单 ID，就为 FALSE
lpcMenuItemInfoMENUITEMINFO，用于设置指定菜单条目的特征
注解

Top
IsMenu
IsMenu

VB 声明
Declare Function IsMenu Lib "user32" Alias "IsMenu" (ByVal hMenu As Long) As Long
说明
判断指定的句柄是否为一个菜单的句柄
返回值
Long，如句柄是菜单句柄，就返回 TRUE；否则返回零
参数表
参数类型及说明
hMenuLong，欲测试的菜单的句柄

Top
LoadMenu
LoadMenu

VB 声明
Declare Function LoadMenu Lib "user32" Alias "LoadMenuA" (ByVal hInstance As Long, ByVal lpString As String) As Long
说明
从指定的模块或应用程序实例中载入一个菜单
返回值
Long，已装载的菜单的句柄；零意味着出错。会设置 GetLastError
参数表
参数类型及说明
hInstanceLong，一个动态链接库的模块句柄。或指定了特定可执行文件的一个实例句柄，那个可执行文件中包含了菜单资源
lpStringString，作为字串使用时，指定欲载入的菜单资源的名字；作为 Long 值使用时，指定欲载入的菜单 ID。
注解
在 vb 里使用：由于 vb 不能控制那些不与现有 vb 菜单兼容的菜单，所以不建议使用这个函数

Top
LoadMenuIndirect
LoadMenuIndirect

VB 声明
Declare Function LoadMenuIndirect Lib "user32" Alias "LoadMenuIndirectA" (ByVal lpMenuTemplate As Long) As Long
说明
在包含了 MENUITEMTEMPLATE 数据结构的一个内存块的基础上，载入一个菜单
返回值
Long，已载入的菜单句柄，零意味着出错。会设置 GetLastError
参数表
参数类型及说明
lpMenuTemplateLong，指向一个 MENUITEMTEMPLATEHEADER 结构的指针。在内存中，该结构的后面跟随有一个或多个 MENUITEMTEMPLATE 数据结构。在 windows 95 及 windows nt 4.0 中，可能是指向一个 MENUEX_HEADER_TEMPLATE 结构的指针；在那个结构后跟随有一个或多个 MENUEX_HEADER_ITEM 结构，用于指定扩展菜单

Top
MenuItemFromPoint
MenuItemFromPoint

VB 声明
Declare Function MenuItemFromPoint Lib "user32" Alias "MenuItemFromPoint" (ByVal hWnd As Long, ByVal hMenu As Long, ByVal ptScreen As POINTAPI) As Long
说明
判断哪个菜单条目包含了屏幕上一个指定的点

返回值
Long，包含了指定点的条目的位置。如果没有菜单条目包含了指定的点，就返回-1
参数表
参数类型及说明
hWndLong，包含了指定菜单的那个窗口的句柄
hMenuLong，菜单句柄
ptScreenPOINTAPI，点的位置。如 hMenu 是一个顶级菜单条，这个点就用 hWnd 窗口的窗口坐标表示。否则，它采用窗口的客户区坐标表示

Top
ModifyMenu
ModifyMenu,ModifyMenuBynum

VB 声明
Declare Function ModifyMenu& Lib "user32" Alias "ModifyMenuA" (ByVal hMenu As Long, ByVal nPosition As Long, ByVal wFlags As Long, ByVal wIDNewItem As Long, ByVal lpString As String)
Declare Function ModifyMenuBynum& Lib "user32" Alias "ModifyMenuA" (ByVal hMenu As Long, ByVal nPosition As Long, ByVal wFlags As Long, ByVal wIDNewItem As Long, ByVal lpString As Long)
说明
改变菜单条目。在 vb 里这个函数做出的许多改变都会有效的执行，但不能由 vb 菜单对象反映出来
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hMenuLong，菜单句柄
nPositionLong，欲改变的菜单条目的标识符。如在 wFlags 参数中指定了 MF_BYCOMMAND，这个参数就代表欲改变的菜单条目的命令 ID。如设置的是 MF_BYPOSITION，这个参数就代表菜单条目在菜单中的位置（第一个条目的位置为零）
wFlagsLong，一系列常数标志的组合。详见菜单常数标志表
wIDNewItemLong，指定菜单条目的新命令 ID。如在 wFlags 参数中指定了 MF_POPUP 标志，就应是一个弹出式菜单的句柄
lpStringString 或 Long，如在 wFlags 参数中指定了 MF_STRING 标志，就代表欲设置到菜单的字串。如设置的是 MF_BITMAP，就代表一个 Long 变量，其中包含了一个位图句柄。如设置的是 MF_OWNERDRAW，那么这个值就会包括到 DRAWITEMSTRUCT 和 MEASUREITEMSTRUCT 结构中，并由 windows 在条目需要重画的时候发出
注解
标志的下述组合形式是不允许的：MF_BYCOMMAND 和 MF_BYPOSITION；MF_CHECKED 和 MF_UNCHECKED；MF_MENUBARBREAK 和 MF_MENUBREAK；MF_DISABLED，MF_ENABLED 和 MF_GRAYED；MF_BITMAP，MF_STRING，MF_OWNERDRAW 和 MF_SEPARATOR
菜单常数标志表

MF_BITMAP 菜单条目是一幅位图。一旦设入菜单，这幅位图就绝对不能删除。所以不应该使用由 vb 的 image 属性返回的值
MF_BYCOMMAND 菜单条目由菜单的命令 ID 指定
MF_BYPOSITION 菜单条目由条目在菜单中的位置决定。零代表菜单中的第一个条目
MF_CHECKED 检查指定的菜单条目。不能与 vb 的 checked 属性兼容
MF_DISABLED 禁止指定的菜单条目。不与 vb 的 enabled 属性兼容
MF_ENABLED 允许指定的菜单条目。不与 vb 的 enabled 属性兼容
MF_GRAYED 禁止指定的菜单条目，并用浅灰色描述它。不与 vb 的 enabled 属性兼容
MF_MENUBARBREAK 在弹出式菜单中，将指定的条目放置于一个新列，并用一条垂直线分隔不同的列
MF_MENUBREAK 在弹出式菜单中，将指定的条目放置于一个新列。在顶级菜单中，将条目放置到一个新行
MF_OWNERDRAW 创建一个物主绘图菜单（由您设计的程序负责描绘每个菜单条目）
MF_POPUP 将一个弹出式菜单置于指定的条目。可用于创建子菜单及弹出式菜单
MF_SEPARATOR 在指定的条目处显示一条分隔线
MF_STRING 在指定的条目处放置一个字串。不与 vb 的 caption 属性兼容
MF_UNCHECKED 检查指定的条目。不能与 vb 的 checked 属性兼容

Top
RemoveMenu
RemoveMenu

VB 声明
Declare Function RemoveMenu Lib "user32" Alias "RemoveMenu" (ByVal hMenu As Long, ByVal nPosition As Long, ByVal wFlags As Long) As Long
说明
删除指定的菜单条目。如删除的条目属于一个弹出式菜单，那么这个函数不会同时删除弹出式菜单。首先应该用 GetSubMenu 函数取得弹出式菜单的句柄，再在以后将其删除
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hMenuLong，菜单的句柄
nPositionLong，欲改变的菜单条目的标识符。如在 wFlags 参数中指定了 MF_BYCOMMAND，这个参数就代表欲改变的菜单条目的命令 ID。如设置的是 MF_BYPOSITION，这个参数就代表菜单条目在菜单中的位置（第一个条目的位置为零）
wFlagsLong，常数 MF_BYCOMMAND 或 MF_BYPOSITION，取决于 nPosition 参数
注解
强烈建议大家使用 vb 菜单的 visible 属性从菜单中删除条目，而不要用这个函数，否则会造成指定菜单中其他菜单条目的 visible 属性对错误的菜单条目产生影响

Top
SetMenu
SetMenu

VB 声明
Declare Function SetMenu Lib "user32" Alias "SetMenu" (ByVal hwnd As Long, ByVal hMenu As Long) As Long
说明
设置窗口菜单
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，窗口的句柄
hMenuLong，窗口的新菜单的句柄
注解
不建议在 vb 里使用这个函数。如坚持使用，务必留意新菜单中的命令 ID 并不兼容于原始的 vb 窗口。只有窗体窗口才应通过这个函数指定。窗口的前一个菜单不会由这个函数删除

Top
SetMenuContextHelpId
SetMenuContextHelpId

VB 声明
Declare Function SetMenuContextHelpId Lib "user32" Alias "SetMenuContextHelpId" (ByVal hMenu As Long, ByVal dw As Long) As Long
说明
设置用 SetMenuContextHelpId 函数分配给一个菜单的帮助场景 ID
返回值
Long，TRUE（非零）表示成功，否则返回零
参数表
参数类型及说明
hMenuLong，菜单的句柄
dwLong，要设置的帮助场景 ID
注解
用这个函数设置的场景 ID 并不对应于用 vb 菜单栏设计程序分配的场景 ID。vb 在内部保存有它自己的帮助场景 ID。vb 的帮助场景 ID 应是我们优先考虑的方案。尽可能不要用这个函数

Top
SetMenuDefaultItem
SetMenuDefaultItem

VB 声明
Declare Function SetMenuDefaultItem Lib "user32" Alias "SetMenuDefaultItem" (ByVal hMenu As Long, ByVal uItem As Long, ByVal fByPos As Long) As Long
说明

将一个菜单条目设为默认条目。这个条目会转换成双击菜单时执行的操作
返回值
Long，TRUE（非零）表示成功，否则返回零。会设置 GetLastError
参数表
参数类型及说明
hMenuLong，菜单句柄
uItemLong，欲设为默认菜单条目的一个条目的位置或菜单 ID。如果为-1，表示清除当前的默认条目
fByPosLong，如 uItem 是条目的位置，就为 TRUE；如果是菜单 ID，就为 FALSE

Top
SetMenuItemBitmaps
SetMenuItemBitmaps

VB 声明
Declare Function SetMenuItemBitmaps Lib "user32" Alias "SetMenuItemBitmaps" (ByVal hMenu As Long, ByVal nPosition As Long, ByVal wFlags As Long, ByVal hBitmapUnchecked As Long, ByVal hBitmapChecked As Long) As Long
说明
设置一幅特定位图，令其在指定的菜单条目中使用，代替标准的复选符号（√）。位图的大小必须与菜单复选符号的正确大小相符，这个正确大小可以由 GetMenuCheckMarkDimensions 函数获得
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hMenuLong，菜单句柄
nPositionLong，欲设置位图的一个菜单条目的标识符。如在 wFlags 参数中指定了 MF_BYCOMMAND，这个参数就代表欲改变的菜单条目的命令 ID。如设置的是 MF_BYPOSITION，这个参数就代表菜单条目在菜单中的位置（第一个条目的位置为零）
wFlagsLong，常数 MF_BYCOMMAND 或 MF_BYPOSITION，取决于 nPosition 参数
hBitmapUncheckedLong，撤消复选时为菜单条目显示的一幅位图的句柄。如果为零，表示不在未复选状态下显示任何标志
hBitmapCheckedLong，复选时为菜单条目显示的一幅位图的句柄。可设为零，表示复选时不显示任何标志。如两个位图句柄的值都是零，则为这个条目恢复使用默认复选位图
注解
使用的位图可能由多个条目共享。一旦不再需要，位图必须由应用程序清除，因为 windows 不能自动对它进行清除

Top
SetMenuItemInfo
SetMenuItemInfo

VB 声明

Declare Function SetMenuItemInfo Lib "user32" Alias "SetMenuItemInfoA" (ByVal hMenu As Long, ByVal un As Long, ByVal bool As Boolean, lpcMenuItemInfo As MENUITEMINFO) As Long
说明
为一个菜单条目设置指定的信息，具体信息保存于 MENUITEMINFO 结构中
返回值
Long，TRUE（非零）表示成功，否则返回零。会设置 GetLastError
参数表
参数类型及说明
hMenuLong，菜单句柄
unLong，菜单条目的菜单 ID 或位置
boolBoolean，如 un 指定了条目的位置，就为 TRUE；如指定的是菜单 ID，就为 FALSE
lpcMenuItemInfoMENUITEMINFO，用于设置目标菜单条目的特征

Top
TrackPopupMenu
TrackPopupMenu, TrackPopupMenuBynum

VB 声明
Declare Function TrackPopupMenu& Lib "user32" (ByVal hMenu As Long, ByVal wFlags As Long, ByVal x As Long, ByVal y As Long, ByVal nReserved As Long, ByVal hwnd As Long, lprc As Rect)
Declare Function TrackPopupMenuBynum& Lib "user32" Alias "TrackPopupMenu" (ByVal hMenu As Long, ByVal wFlags As Long, ByVal x As Long, ByVal y As Long, ByVal nReserved As Long, ByVal hwnd As Long, ByVal lprc As Long)
说明
在屏幕的任意地方显示一个弹出式菜单
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hMenuLong，弹出式菜单的句柄
wFlagsLong，位置标志和鼠标追踪标志的组合，见下表
位置标志说明
TPM_CENTERALIGN 菜单在指定位置水平居中
TPM_LEFTALIGN 菜单的左侧置于水平 x 坐标处
TPM_RIGHTALIGN 菜单的右侧置于水平 x 坐标处
TPM_LEFTBUTTON 鼠标左键标准运作方式
TPM_RIGHTBUTTON 用鼠标右键进行菜单追踪
x,yLong，这个点指定了弹出式菜单在屏幕坐标系统中的位置
nReservedLong，未使用，设为零
hwndLong，用于接收弹出式菜单命令的窗口的句柄。应该使用窗体的窗口句柄——窗体中有一个菜单能象弹出式菜单那样接收相同的命令 ID 集
lprcRect，用屏幕坐标定义的一个矩形，如用户在这个矩形的范围内单击，则弹出式菜单不

会关闭。如单击弹出式菜单之外的任何一个地方，则会关闭菜单。可以设为 NULL
注解
用这个函数创建的菜单，菜单中的命令 ID 并不与 vb 期望的那些相符

Top
TrackPopupMenuEx
TrackPopupMenuEx

VB 声明
Declare Function TrackPopupMenuEx Lib "user32" Alias "TrackPopupMenuEx" (ByVal hMenu As Long, ByVal un As Long, ByVal n1 As Long, ByVal n2 As Long, ByVal hWnd As Long, lpTPMParams As TPMPARAMS) As Long
说明
与 TrackPopupMenu 相似，只是它提供了额外的功能
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hMenuLong，弹出式菜单的句柄
unLong，定位标志和鼠标追踪标志的组合。参考 TrackPopupMenu，另外还包括两个标志：TPM_HORIZONTAL 或 TPM_VERTICAL。参考 lpTPMParams 参数的说明
n1,n2Long，定义了弹出式菜单位置的一个 x,y 点（n1,n2），用屏幕坐标表示
hWndLong，用于接收弹出式菜单命令的窗口的句柄。应该使用特定窗体的窗口句柄，该窗体有一个菜单能够与弹出式菜单一样接收相同的命令 ID 集
lpTPMParamsTPMPARAMS，指向一个 TPMPARAMS 结构的指针。这个结构包含了一个矩形，规定了不能由这个弹出式菜单覆盖的区域。如果在 un 参数中指定了 TPM_HORIZONTAL 标志，windows 就会试着设置水平位置，将弹出式菜单垂直移到这个矩形的外部。如指定了 TPM_VERTICAL，那么 windows 会试着水平移动弹出式菜单的位置

Top
文本和字体函数
文本和字体函数，共三页。第一页，第二页，第三页
AddFontResource 在 Windows 系统中添加一种字体资源
CreateFont 用指定的属性创建一种逻辑字体
CreateFontIndirect 用指定的属性创建一种逻辑字体
CreateScalableFontResource 为一种 TureType 字体创建一个资源文件，以便能用 API 函数 AddFontResource 将其加入 Windows 系统
DrawText 将文本描绘到指定的矩形中
DrawTextEx 与 DrawText 相似，只是加入了更多的功能
EnumFontFamilies 列举指定设备可用的字体
EnumFontFamiliesEx 列举指定设备可用的字体
EnumFonts 列举指定设备可用的字体
ExtTextOut 经过扩展的文本描绘函数。也请参考 SetTextAlign 函数
GetAspectRatioFilterEx 用 SetMapperFlags 要求 Windows 只选择与设备当前纵横比相符的光

栅字体时，本函数可判断纵横比大小
GetCharABCWidths 判断 TureType 字体中一个或多个字符的 A-B-C 大小
GetCharABCWidthsFloat 查询一种字体中一个或多个字符的 A-B-C 尺寸
GetCharacterPlacement 该函数用于了解如何用一个给定的字符显示一个字串
GetCharWidth 调查字体中一个或多个字符的宽度
GetFontData 接收一种可缩放字体文件的数据
EnumFonts
EnumFonts

VB 声明
Declare Function EnumFonts Lib "gdi32" Alias "EnumFontsA" (ByVal hDC As Long, ByVal lpsz As String, ByVal lpFontEnumProc As Long, ByVal lParam As Long) As Long
说明
列举指定设备可用的字体
注解
该函数使用的参数与 EnumFontFamilies 函数是一样的，工作原理也大致相同。只是 EnumFontFamilies 会利用 ENUMLOGFONT 和 NEWTEXTMETRIC 结构向回调函数传递附加的信息，而不是使用 LOGFONT 和 TEXTMETRIC 结构。请参考 EnumFontFamilies 函数，那里有更详细的解释

Top
ExtTextOut
ExtTextOut

VB 声明
Declare Function ExtTextOut Lib "gdi32" Alias "ExtTextOutA" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long, ByVal wOptions As Long, lpRect As Rect, ByVal lpString As String, ByVal nCount As Long, lpDx As Long) As Long
说明
经过扩展的文本描绘函数。也请参考 SetTextAlign 函数
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
x,yLong，以逻辑坐标表示的一个点，指定了绘图起点
wOptionsLong，下述标志常数的任意组合
ETO_CLIPPED 将文本剪切出指定的矩形
ETO_GLPYH_INDEXlpString 是一个字样索引表。参考对 GetCharacterPlacement 函数的说明。只适用于 Win95
ETO_OPAQUE 在正式描绘文本前，用当前的背景色填充矩形
lpRectRect，指定一个矩形，用于对文本进行格式化处理。可指定长整数 0，在不用矩形区域的前提下描绘文本
lpStringString，欲描绘的字串

nCountLong，字串中要显示出来的字符数
lpDxLong，如果不是零，这个参数就代表指向一个 Long 值数组的指针。该数组对每一对字符的间距进行了说明（采用逻辑单位）。其中第一个条目是第一和第二个字符的间距；第二个条目是第二和第三个字符的间距；以此类推。如果为零，函数就使用字体的默认间距设置

Top
GetAspectRatioFilterEx
GetAspectRatioFilterEx

VB 声明
Declare Function GetAspectRatioFilterEx Lib "gdi32" Alias "GetAspectRatioFilterEx" (ByVal hdc As Long, lpAspectRatio As SIZE) As Long
说明
我们可用 SetMapperFlags 函数要求 Windows 只选择与设备当前纵横比相符的光栅字体。这个函数可判断那种特殊选择过程中使用的纵横比是多大
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，一个设备场景的句柄
lpAspectRatioSIZE，用于装载纵横比的一个结构

Top
GetCharABCWidths
GetCharABCWidths

VB 声明
Declare Function GetCharABCWidths Lib "gdi32" Alias "GetCharABCWidthsA" (ByVal hdc As Long, ByVal uFirstChar As Long, ByVal uLastChar As Long, lpabc As ABC) As Long
说明
判断 TureType 字体中一个或多个字符的 A-B-C 大小
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
uFirstCharLong，欲调查 A-B-C 尺寸的第一个字符的 ASCII 值
uLastCharLong，欲调查 A-B-C 尺寸的最后一个字符的 ASCII 值
lpabcABC，在 ABC 结构数组中的第一个条目。这个数组填充了指定的字符大小设置。该数组的长度必须足够大，足以容下要求的所有字符
注解
对于非 TureType 字体用 GetCharWidth 函数

Top

GetCharABCWidthsFloat
GetCharABCWidthsFloat

VB 声明
Declare Function GetCharABCWidthsFloat Lib "gdi32" Alias "GetCharABCWidthsFloatA" (ByVal hdc As Long, ByVal iFirstChar As Long, ByVal iLastChar As Long, lpABCF As ABCFLOAT) As Long
说明
查询一种字体中一个或多个字符的 A-B-C 尺寸
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
iFirstCharLong，欲调查 A-B-C 尺寸的第一个字符的 ASCII 值
iLastCharLong，欲调查 A-B-C 尺寸的最后一个字符的 ASCII 值
lpABCFABCFLOAT，在 ABCFLOAT 结构数组中的第一个条目。这个数组填充了指定的字符大小设置。该数组的长度必须足够大，足以容下要求的所有字符
注解
和 GetCharABCWidths 不同，这个函数适用于任何字符。ABC 值是以浮点数的形式返回的，而且可能不是整数——具体取决于用这个函数处理非 TureType 字体时采用的转换方式
适用平台
Windows NT

Top
GetCharacterPlacement
GetCharacterPlacement

VB 声明
Declare Function GetCharacterPlacement Lib "gdi32" Alias " GetCharacterPlacementA" (ByVal hdc As Long, ByVal lpsz As String, ByVal n1 As Long, ByVal n2 As Long, lpGcpResults As GCP_RESULTS, ByVal dw As Long) As Long
说明
该函数用于了解如何用一个给定的字符显示一个字串
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
lpszString，欲分析的字串
n1Long，字串的长度
n2Long，若在 dw 参数中指定了 GCP_MAXEXTENT 常数，那么一旦显示的字串超出了由该参数指定的宽度（用逻辑单位），函数就会停止处理字串
lpGcpResultsGCP_RESULTS，在这个结构中装载为这个字串计算出来的信息

dwLong，下述常数的一个或多个：
GCP_CLASSINlpGcpResults 结构中的 lpClass 数组包含了字串中各字符的分类信息
GCP_DIACRITIC 在计算时将发音符和“废”字符考虑在内
GCP_DISPLAYZWG 显示某些字符集中使用的不可见字符，根据它们在一个词中的位置修改字符
GCP_GLPYPHSHAPE 允许对字样（字面）进行特殊处理。根据 GetFontLanguageInfo 函数的结果使用
GCP_JUSTIFY 调整字样位置，对字串进行对齐处理，使其与 n2 参数指定的范围相符
GCP_JUSTIFYINlpGcpResults 结构中的 lpDX 参数包含了计算过程中使用的对齐粗细设置
GCP_LIGATE 如当前字体支持，就用连字技术将字符合并成单独一个字符
GCP_MAXEXTENT 请参考对 n2 参数的说明
GCP_USERKERNING 计算字符位置时，使用字距表（如果有的话）可用其他标志对希伯来和阿拉伯字体进行特殊处理。这类语言按照从右到左的顺序显示文字，而且具体显示的字样由字符在一个词中的位置决定

Top
GetCharWidth
GetCharWidth, GetCharWidth32, GetCharWidthFloat

VB 声明
Declare Function GetCharWidth& Lib "gdi32" Alias "GetCharWidthA" (ByVal hDC As Long, ByVal iFirstChar As Long, ByVal iLastChar As Long, lpBuffer As Long)
Declare Function GetCharWidth32& Lib "gdi32" Alias "GetCharWidth32A" (ByVal hdc As Long, ByVal iFirstChar As Long, ByVal iLastChar As Long, lpBuffer As Long)
Declare Function GetCharWidthFloat& Lib "gdi32" Alias "GetCharWidthFloatA" (ByVal hdc As Long, ByVal iFirstChar As Long, ByVal iLastChar As Long, pxBuffer As Single)
说明
调查字体中一个或多个字符的宽度。在 Win32 环境中，请使用 GetCharWidth32 函数。用 GetCharWidthFloat 则可获得小数宽度
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hDCLong，设备场景的句柄
iFirstCharLong，要调查宽度的第一个字符的 ASCII 值
iLastCharLong，要调查宽度的最后一个字符的 ASCII 值
lpBufferLong，指定 Long 值数组的第一个条目。该数组容纳了字体的字符宽度设置
pxBufferSingle，指定 Single 值数组的第一个条目。该数组容纳了字体的字符宽度设置
注解
对于 TureType 字体，GetCharABCWidths 可获得更详细的信息

Top
GetFontData
GetFontData

VB 声明
Declare Function GetFontData Lib "gdi32" Alias "GetFontDataA" (ByVal hdc As Long, ByVal dwTable As Long, ByVal dwOffset As Long, lpvBuffer As Any, ByVal cbData As Long) As Long
说明
接收一种可缩放字体文件的数据。随后，可用这些数据将字体信息嵌入一个文档。如果需要在文档使用一种特殊字体，同时这种字体在大多数系统中都不常见，而且程序员希望文档无论如何都要显示这种字体，那么这种技术就相当有用了
在 VB 里使用
未经测试
Use with VB:Untested

Top
GetFontLanguageInfo
GetFontLanguageInfo

VB 声明
Declare Function GetFontLanguageInfo Lib "gdi32" Alias "GetFontLanguageInfo" (ByVal hdc As Long) As Long
说明
返回目前选入指定设备场景中的字体的信息
返回值
Long，如返回零，表示是简单字体；GCP_ERROR 表示出错。否则，返回下述一个或多个标志：
GCP_DBCS 双字节字符集
GCP_DIACRITIC 字体包含了发音字符
FLI_GLYPHS 字体包含了通常不会显示出来的字样
GCP_GLYPHSHAPE 字体包含了特殊字符，用于字样显示由除字符值以外的其他因素决定的场合。例如显示字样由一个字符在单词中位置决定，或者显示单个“连字”，指出这是两个字符值的组合
GCP_KASHIDA 在阿拉伯字体中使用
GCP_LIGATE 字体包含了连字字样
GCP_USERKERNING 字体包含了字距表
GCP_REORDER 字体必须记录下来，以便正确显示。随同希伯来和阿拉伯字体使用
参数表
参数类型及说明
hdcLong，设备场景的句柄
注解
从这个函数返回的值可用于设置 GetCharacterPlacement 函数的标志

Top
GetGlyphOutline
GetGlyphOutline

VB 声明
Declare Function GetGlyphOutline Lib "gdi32" Alias "GetGlyphOutlineA" (ByVal hdc As Long, ByVal uChar As Long, ByVal fuFormat As Long, lpgm As GLYPHMETRICS, ByVal cbBuffer As Long, lpBuffer As Any, lpmat2 As MAT2) As Long
说明
取得 TureType 字体中构成一个字符的曲线信息。主要用于将文本转换成曲线，以及用于字体的特殊处理（比如造字程序。无论如何，都要求程序员掌握高深的字体技术）。请参考由微软发布的 TureType 字体规格书，其中对这个字体进行了更详细的解释

Top
GetKerningPairs
GetKerningPairs

VB 声明
Declare Function GetKerningPairs Lib "gdi32" Alias "GetKerningPairsA" (ByVal hdc As Long, ByVal cPairs As Long, lpkrnpair As KERNINGPAIR) As Long
说明
取得指定字体的字距信息
返回值
Long，返回的字距对数量，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
cPairsLong，KERNINGPAIR 结构在数组中的数量，由 lpkrnpair 参数指定。如果将这个参数与 lpkrnpair 设置成零，可判断出字距表的大小
lpkrnpairKERNINGPAIR，指定 KERNINGPAIR 结构数组中的第一个条目
注解
参考 KERNINGPAIR
返回以后，结构会针对数组中的每个条目象下面这样设置字段：
wFirst 指定一个双字符序列的第一个字符；wSecond 指定第二个字符。iKernAmount 字段指定指定这两个字符的字间距。
例如，假设第一个字符是"f"，第二个是"i"。那么在这两个字符一个紧接一个显示出来时，字间距就是添加到默认字符间距上的一个逻辑距离。这个值通常为负，因为系统通常会令两个字符靠得更近

Top
GetOutlineTextMetrics
GetOutlineTextMetrics

VB 声明
Declare Function GetOutlineTextMetrics Lib "gdi32" Alias "GetOutlineTextMetricsA" (ByVal hdc As Long, ByVal cbData As Long, lpotm As OUTLINETEXTMETRIC) As Long
说明
接收与 TureType 字体内部特征有关的详细信息。请参考微软公司发布的 TureType 字体规格

文件，它提供了这个函数的进一步信息

Top
GetRasterizerCaps
GetRasterizerCaps

VB 声明
Declare Function GetRasterizerCaps Lib "gdi32" Alias "GetRasterizerCaps" (lpraststat As RASTERIZER_STATUS, ByVal cb As Long) As Long
说明
了解系统是否有能力支持可缩放的字体。利用得到的结果，可判断那种系统中是否允许使用 TureType 字体
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpraststatRASTERIZER_STATUS，这个结构用于装载光栅信息
cbLong，欲复制到结构中的字符数
注解
结构目前的大小是 6 个字符，并包含在 RASTERIZER_STATUS 结构的第一个整数中

Top
GetTabbedTextExtent
GetTabbedTextExtent

VB 声明
Declare Function GetTabbedTextExtent Lib "user32" Alias "GetTabbedTextExtentA" (ByVal hdc As Long, ByVal lpString As String, ByVal nCount As Long, ByVal nTabPositions As Long, lpnTabStopPositions As Long) As Long
说明
判断一个字串占据的范围，同时考虑制表站扩充的因素。也请参考 TabbedTextOut 函数
返回值
Long，低 16 位包含了文本宽度，采用设备场景的逻辑坐标表示。高 16 位则包含了文本高度。零意味着出错
参数表
参数类型及说明
hdcLong，设备场景的句柄
lpStringString，欲计算的字串
nCountLong，字串中的字符数
nTabPositionsLong，lpnTabStopPositions 数组中的制表站数量。如果是零，则 lpnTabStopPositions 也应是 NULL（需另行创建一个函数声明，将这个参数声明成 ByVal nTabPositions&）。在这种情况下，制表站会根据当前字体的平均字符宽度，设置成默认的 8 字符间距。如 nTabPositions 是 1，那么制表站间距就会以 lpnTabStopPositions 数组的第一个条目为准

lpnTabStopPositionsLong，指定制表站位置数组的第一个条目。这种位置是按升序用设备坐标指定的
注解
进行这种计算的时候，剪切区不会考虑在内

Top
GetTextAlign
GetTextAlign

VB 声明
Declare Function GetTextAlign Lib "gdi32" Alias "GetTextAlign" (ByVal hdc As Long) As Long
说明
接收一个设备场景当前的文本对齐标志
返回值
Long，当前的文本对齐标志。GDI_ERROR 表示失败。会设置 GetLastError。文本的对齐方法由几个常数的组合决定。其中每个常数都来自下述不同的组别。参考下面总结的文本对齐标志
水平对齐标志 TA_CENTER 文本在约束矩形内居中显示
TA_LEFT 文本在约束矩形内左对齐（默认设置）
TA_RIGHT 文本在约束矩形内右对齐
垂直对齐标志定义文本输出函数的 Y 参数的含义
TA_BASELINEY 参数指定字体基线的位置
TA_BOTTOMY 参数指定约束矩形底边的位置
TA_TOPY 参数指定约束矩形顶边的位置（默认设置）
当前位置 TA_NOUPDATECP 文本输出函数不使用设备场景当前的绘图位置
TA_UPDATECP 文本输出函数使用设备场景当前的绘图位置。完成绘图后，输出函数会对当前位置进行更新。文本输出函数的 X 和 Y 参数会被忽略——绘图会以当前位置为起点
其他 TA_RTLREADING 文本输出从右到左进行。仅在 Windows95 下适用于希伯来和阿拉伯字体
参数表
参数类型及说明
hdcLong，设备场景的句柄
注解
参考对 SetTextAlign 函数的说明，进一步了解文本对齐标志的情况

Top
GetTextCharacterExtra
GetTextCharacterExtra

VB 声明
Declare Function GetTextCharacterExtra Lib "gdi32" Alias "GetTextCharacterExtraA" (ByVal hdc As Long) As Long
说明
判断额外字符间距的当前值。请参考 SetTextCharacterExtra 函数，了解进一步的情况

返回值
Long，返回 Windows 描绘文本时于字符间添加的额外空间大小
参数表
参数类型及说明
hdcLong，设备场景的句柄

Top
GetTextCharset
GetTextCharset

VB 声明
Declare Function GetTextCharset Lib "gdi32" Alias "GetTextCharset" (ByVal hdc As Long) As Long
说明
接收当前选入指定设备场景的字体的字符集标识符
返回值
Long，字符集标识符。DEFAULT_CHARSET 代表默认字符集；-1 表示出错
参数表
参数类型及说明
hdcLong，设备场景的句柄

Top
GetTextCharsetInfo
GetTextCharsetInfo

VB 声明
Declare Function GetTextCharsetInfo Lib "gdi32" Alias "GetTextCharsetInfo" (ByVal hdc As Long, lpSig As FONTSIGNATURE, ByVal dwFlags As Long) As Long
说明
获取与当前选定字体的字符集有关的详细信息
返回值
Long，字符集标识符。DEFAULT_CHARSET 代表默认字符集；-1 表示出错
参数表
参数类型及说明
hdcLong，设备场景的句柄
lpSigFONTSIGNATURE，用于装载字符集信息的结构。有关 Unicode 字体签名（signature）的资料可在 Unicode 规格文件中找到
dwFlagsLong，已保留，设为零

Top
GetTextColor
GetTextColor

VB 声明

Declare Function GetTextColor Lib "gdi32" Alias "GetTextColor" (ByVal hdc As Long) As Long
说明
判断当前字体颜色。通常也称为“前景色”
返回值
Long，文字的当前 RGB 颜色设置。如果出错，会返回 CLR_INVALID。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
注解
在 VB 里用 ForeColor 属性设置图片控件和窗体

Top
GetTextExtentExPoint
GetTextExtentExPoint

VB 声明
Declare Function GetTextExtentExPoint Lib "gdi32" Alias "GetTextExtentExPointA" (ByVal hdc As Long, ByVal lpszStr As String, ByVal cchString As Long, ByVal nMaxExtent As Long, lpnFit As Long, alpDx As Long, lpSize As SIZE) As Long
说明
判断要填入指定区域的字符数量。也用一个数组装载每个字符的范围信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
lpszStrString，准备量度其范围的字串
cchStringLong，lpszStr 字串的长度
nMaxExtentLong，采用逻辑单位表示的水平范围
lpnFitLong，在其中保存欲填充到指定区域的字符数量。可以为 NULL（用一个别名化的声明来设置 ByVal As Long）——此时会忽略 nMaxExtent 设置
AsLong，cchString 数组的第一个条目。每个条目都要保存从字串起点到这个字符的距离（采用逻辑单位）。如果不需要这方面的信息，也可设为 NULL（用别名声明设置 ByVal As Long）
lpSizeSIZE，这个结构用于装载字串范围的高度和宽度信息
注解
可用这个函数计算自动换行输出时的字符位置

Top
GetTextExtentPoint
GetTextExtentPoint, GetTextExtentPoint32

VB 声明
Declare Function GetTextExtentPoint& Lib "gdi32" Alias "GetTextExtentPointA" (ByVal hdc As Long, ByVal lpszString As String, ByVal cbString As Long, lpSize As SIZE)

Declare Function GetTextExtentPoint32& Lib "gdi32" Alias "GetTextExtentPoint32A" (ByVal hdc As Long, ByVal lpsz As String, ByVal cbString As Long, lpSize As SIZE)
说明
判断一个字串的大小（范围）。在 Win32 环境中，最好使用 GetTextExtentPoint32，它提供了更精确的计算结果
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
lpszStringString，欲度量其范围（extent）的一个字串
cbStringLong，lpszString 字串的长度
lpSizeSIZE，这个结构用于装载字串范围的高度和宽度信息
注解
这个函数不会将剪切区考虑在内，但却考虑到了由 SetTextCharacterExtra 函数设置的任何额外空间（间距）

Top
GetTextFace
GetTextFace

VB 声明
Declare Function GetTextFace Lib "gdi32" Alias "GetTextFaceA" (ByVal hdc As Long, ByVal nCount As Long, ByVal lpFacename As String) As Long
说明
获取一种字体的字样名
返回值
Long，缓冲区中载入的字节数量。零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
nCountLong，lpFacename 字串的大小
lpFacenameString，指定一个字串缓冲区，用于装载当前选定字体的字样名称。这个缓冲区事先必须初始化成至少 nCount+1 个字符的长度
注解
类似于读取 VB 的 FontName 属性

Top
GetTextMetrics
GetTextMetrics

VB 声明
Declare Function GetTextMetrics Lib "gdi32" Alias "GetTextMetricsA" (ByVal hdc As Long, lpMetrics As TEXTMETRIC) As Long

说明
获取与选入一种设备场景的物理字体有关的信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
lpMetricsTEXTMETRIC，用于填充物理字体属性信息的一个结构

Top
GrayString
GrayString, GrayStringByString

VB 声明
Declare Function GrayString& Lib "user32" Alias "GrayStringA" (ByVal hDC As Long, ByVal hBrush As Long, ByVal lpOutputFunc As Long, ByVal lpData As Long, ByVal nCount As Long, ByVal X As Long, ByVal Y As Long, ByVal nWidth As Long, ByVal nHeight As Long)
Declare Function GrayStringByString& Lib "user32" Alias "GrayStringA" (ByVal hDC As Long, ByVal hBrush As Long, ByVal lpOutputFunc As Long, ByVal lpData As String, ByVal nCount As Long, ByVal X As Long, ByVal Y As Long, ByVal nWidth As
Long, ByVal nHeight As Long)
说明
描绘一个以灰色显示的字串。通常由 Windows 用于标识禁止状态。这个函数使用的是当前字体，但会忽略背景及文本颜色
返回值
Long，非零表示成功。如果 TextOut 函数或回调控件绘图事件返回零，那么该函数也会返回零
参数表
参数类型及说明
hDCLong，设备场景的句柄
hBrushLong，用于填充灰色的一个刷子。如果为零就用当前刷子
lpOutputFuncLong，指向一个函数的指针，该函数用于输出文本。通常设为零，表示使用 TextOut 函数。否则可将该参数设成一个进程地址。具体地址可用一个标准函数的 AddressOf 运算符返回，或者用一个回调控件的相关属性返回
lpDataLong，如果是一个字串（将 lpOutputFunc 设为 NULL），这就表示要以灰色显示的一个字串。否则就指定一个 Long 型变量，将其传递给回调函数
nCountLong，欲显示的字符数量。如果设为零，而且 lpData 是个字串，那么就用于计算字串的长度
X,YLong，将字串封闭（约束）起来的一个矩形的 X，Y 坐标
nWidthLong，将字串封闭起来的一个矩形的宽度，采用设备坐标表示。如设为零，就根据字串计算出具体的值
nHeightLong，将字串封闭起来的一个矩形的高度，采用设备坐标表示。如设为零，就根据字串计算出具体的值

Top
PolyTextOut
PolyTextOut

VB 声明
Declare Function PolyTextOut Lib "gdi32" Alias "PolyTextOutA" (ByVal hdc As Long, pptxt As POLYTEXT, cStrings As Long) As Long
说明
描绘一系列字串
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，欲在其中绘图的设备场景
pptxtPOLYTEXT，指定 POLYTEXT 结构数组中的第一个条目。该结构对要描绘字串的位置及内容进行了说明
cStringsLong，pptxt 数组中的条目数量

Top
RemoveFontResource
RemoveFontResource

VB 声明
Declare Function RemoveFontResource Lib "gdi32" Alias "RemoveFontResourceA" (ByVal lpFileName As String) As Long
说明
从 Windows 系统中删除一种字体资源。如删除的字体目前正由其他应用程序使用，则并不将其立即删除
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpFileNameString，一个字体资源文件的文件名
注解
删除一种字体资源后，注意必须调用一下 API 函数：
di% = SendMessageBynum(HWND_BROADCAST, WM_FONTCHANGE, X, Y)
其中，HWND_BROADCAST 和 WM_FONTCHANGE 都是来自 API32.TXT 文件的常数。它的作用是通知所有 Windows 应用程序字体列表已发生了变化。
注意磁盘上的字体文件本身并不会由这个函数删除

Top
SetMapperFlags
SetMapperFlags

VB 声明
Declare Function SetMapperFlags Lib "gdi32" Alias "SetMapperFlags" (ByVal hdc As Long, ByVal dwFlag As Long) As Long
说明
Windows 对字体进行映射时，可用该函数选择与目标设备的纵横比相符的光栅字体。请参考对 GetAspectRatioFilterEx 函数的解释，了解进一步的情况
返回值
Long，字体映射标志的前一个值。GDI_ERROR 表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
dwFlagLong，用 ASPECT_FILTERING 常数请求 GDI 选择与设备纵横比相符的字体

Top
SetTextAlign
SetTextAlign

VB 声明
Declare Function SetTextAlign Lib "gdi32" Alias "SetTextAlign" (ByVal hdc As Long, ByVal wFlags As Long) As Long
说明
设置文本对齐方式，并指定在文本输出过程中使用设备场景的当前位置
返回值
Long，前一个文本对齐标志，GDI_ERROR 表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
wFlagsLong，参考 GetTextAlign 函数的返回值列表
在 VB 里使用
针对自己修改的任何 VB 窗体或控件，注意确定恢复其原始的对齐排列状态。可用 GetTextAlign 函数了解目前的对齐方式是什么

Top
SetTextCharacterExtra
SetTextCharacterExtra

VB 声明
Declare Function SetTextCharacterExtra Lib "gdi32" Alias "SetTextCharacterExtraA" (ByVal hdc As Long, ByVal nCharExtra As Long) As Long
说明
描绘文本的时候，指定要在字符间插入的额外间距
返回值
Long，这个设备场景的前一个额外间距设置
参数表

参数类型及说明
hdcLong，设备场景的句柄
nCharExtraLong，要在字符间插入的额外空间，采用设备场景的逻辑坐标系统
在 VB 里使用
如改变了这个设置，注意恢复 VB 窗体或控件原来的字符间距设置

Top
SetTextColor
SetTextColor

VB 声明
Declare Function SetTextColor Lib "gdi32" Alias "SetTextColor" (ByVal hdc As Long, ByVal crColor As Long) As Long
说明
设置当前文本颜色。这种颜色也称为“前景色”
返回值
Long，文本色的前一个 RGB 颜色设定。CLR_INVALID 表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
crColorLong，新的文本色
在 VB 里使用
如改变了这个设置，注意恢复 VB 窗体或控件原始的文本颜色

Top
SetTextJustification
SetTextJustification

VB 声明
Declare Function SetTextJustification Lib "gdi32" Alias "SetTextJustification" (ByVal hdc As Long, ByVal nBreakExtra As Long, ByVal nBreakCount As Long) As Long
说明
通过指定一个文本行应占据的额外空间，可用这个函数对文本进行两端对齐处理
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
nBreakExtraLong，描绘时欲添加到字串的额外空间大小
nBreakCountLong，用于分散额外空间的分隔字符的数量
在 VB 里使用
如使用了这个函数，要确定针对 VB 窗体或控件清除错误条件
注解
额外空间由行内各个分隔字符分摊。这里的“分隔字符”是由特定的字体定义的，通常都是

空格字符。可用 GetTextMetrics 函数了解一种字体采用的分隔字符是什么。对文本进行两端对齐排列的时候，通常需要采取的操作步骤如下：
1、用 GetTextExtentPoint32 这个 API 函数计算字串占据的显示范围
2、决定为了使一个行两端对齐，需要加入多少额外的空间（采用逻辑坐标）。这个空间（或距离）通常等于右页边距减去文本的水平“范围”
3、计算一行文本中采用多少个间隔字符（通常是空格）
4、将额外空间以及间隔字符的数量作为参数，调用 SetTextJustification 函数
5、调用文本绘图（显示）函数
这个函数在内部维持着一种错误条件，用于纠正对齐过程中出现的误差。这样一来，我们就可以区分出行内不同部分间的额外间距（如行内使用了多种字体）。具体的方法是将行分割成几个段，然后为每一段都调用这个函数。对于一个新行，必须清除这个错误条件，方法是向 nBreakExtra 和 nBreakCount 参数传递零值，然后调用这个函数

Top
TabbedTextOut
TabbedTextOut

VB 声明
Declare Function TabbedTextOut Lib "user32" Alias "TabbedTextOutA" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long, ByVal lpString As String, ByVal nCount As Long, ByVal nTabPositions As Long, lpnTabStopPositions As Long, ByVal nTabOrigin As Long) As Long
说明
支持制表站的一个文本描绘函数。也请参考 SetTextAlign 函数
返回值
Long，返回字串的显示“范围”。其中，结果值的高 16 位代表高度，低 16 位代表宽度
参数表
参数类型及说明
hdcLong，设备场景的句柄
x,yLong，用逻辑坐标设置的一个点，指定字体的描绘（显示）起点
lpStringString，欲描绘的字串
nCountLong，字串中要正式描绘出来的字符数
nTabPositionsLong，lpnTabStopPositions 数组中的制表站数量。如果是零，lpnTabStopPositions 也应该是 NULL（需要另行创建一个声明，将参数指定成 ByVal nTabPositions& ）——在这种情况下，制表站会根据当前字体的平均字符宽度设置成默认的 8 字符间距。如 nTabPositions 为 1，那么制表站间距就会根据 lpnTabStopPositions 数组的第一个条目设置
lpnTabStopPositionsLong，指定制表站位置数组中的头一个条目。这些位置用设备坐标按升序指定。如果为负数，表示文本应该右对齐制表站，而不是默认的左对齐（仅适用于 Win95）
nTabOriginLong，指定制表站起点。如为同一行多次调用该函数，而又希望维持相同的制表起点，这个参数就显得非常重要

Top
TextOut
TextOut

VB 声明
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
说明
文本绘图函数。也请参考 SetTextAlign
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
x,yLong，绘图的起点，采用逻辑坐标
lpStringString，欲描绘的字串
nCountLong，字串中要描绘的字符数量
注解
在一个路径中，如绘图背景模式是“不透明”（opaque）——请参考 SetBkMode 函数，那么创建的轮廓将由字符单元格减去字符构成。如背景模式为透明，轮廓就由字符的字样本身构成

Top
硬件与系统函数
硬件与系统函数：共七页。第一页，第二页，第三页，第四页，第五页，第六页，第七页
ActivateKeyboardLayout 激活一个新的键盘布局。键盘布局定义了按键在一种物理性键盘上的位置与含义
Beep 用于生成简单的声音
CharToOem 将一个字串从 ANSI 字符集转换到 OEM 字符集
ClipCursor 将指针限制到指定区域
ConvertDefaultLocale 将一个特殊的地方标识符转换成真实的地方 ID
CreateCaret 根据指定的信息创建一个插入符（光标），并将它选定为指定窗口的默认插入符
DestroyCaret 清除（破坏）一个插入符
EnumCalendarInfo 枚举在指定“地方”环境中可用的日历信息
EnumDateFormats 列举指定的“当地”设置中可用的长、短日期格式
EnumSystemCodePages 枚举系统中已安装或支持的代码页
EnumSystemLocales 枚举系统已经安装或提供支持的“地方”设置
EnumTimeFormats 枚举一个指定的地方适用的时间格式
ExitWindowsEx 退出 windows，并用特定的选项重新启动
ExpandEnvironmentStrings 扩充环境字串
FreeEnvironmentStrings 翻译指定的环境字串块
GetACP 判断目前正在生效的 ANSI 代码页
ActivateKeyboardLayout
ActivateKeyboardLayout

VB 声明
Declare Function ActivateKeyboardLayout Lib "user32" Alias "ActivateKeyboardLayout" (ByVal HKL As Long, ByVal flags As Long) As Long

说明
激活一个新的键盘布局。键盘布局定义了按键在一种物理性键盘上的位置与含义
返回值
Long，如执行成功，返回前一个键盘布局的句柄；零表示失败。会设置 GetLastError
参数表
参数类型及说明
HKLLong，指定一个键盘布局的句柄。这个布局是随同 LoadKeyboardLayout 或 GetKeyboardLayoutList 函数载入的。也可用 HKL_NEXT 常数激活下一个已装载布局；或用 HKL_PREV 载入前一个布局
flagsLong，将指定的键盘移至内部键盘布局列表的起始处

Top
Beep
Beep

VB 声明
Declare Function Beep Lib "kernel32" Alias "Beep" (ByVal dwFreq As Long, ByVal dwDuration As Long) As Long
说明
用于生成简单的声音
返回值
Long，TRUE（非零）表示成功，否则返回零。会设置 GetLastError
参数表
参数类型及说明
dwFreqLong，声音频率（从 37Hz 到 32767Hz）。在 windows95 中忽略
dwDurationLong，声音的持续时间，以毫秒为单位。如为-1，表示一直播放声音，直到再次调用该函数为止。在 windows95 中会被忽略
注解
在 windows95 中，这个函数简单的播放默认系统响铃

Top
CharToOem
CharToOem, CharToOemBuff

VB 声明
Declare Function CharToOem& Lib "user32" Alias "CharToOemA" (ByVal lpszSrc As String, ByVal lpszDst As String)
Declare Function CharToOemBuff& Lib "user32" Alias "CharToOemBuffA" (ByVal lpszSrc As String, ByVal lpszDst As String, ByVal cchDstLength As Long)
说明
将一个字串从 ANSI 字符集转换到 OEM 字符集。CharToOemBuff 允许我们指定字串中需转换的字符数量
返回值
Long，肯定是 TRUE

参数表
参数类型及说明
lpszSrcString，欲转换的字串
lpszDstString，用于包含转换结果的 OEM 字串。注意事先将字串初始化成合适的长度。可将相同的字串传递给这两个参数，执行本地转换（即在同一个字串中转换）
cchDstLengthLong，在字串 lpszSrc 中想转换的字符数量
注解
如用一个 Win32 类型库访问宽字符函数 CharToOemW，则 lpszSrc 是一个 Unicode 字串，且 lpszDst 参数绝对不能与 lpszSrc 相同

Top
ClipCursor
ClipCursor, ClipCursorBynum

VB 声明
Declare Function ClipCursor& Lib "user32" (lpRect As RECT)
Declare Function ClipCursorBynum& Lib "user32" Alias "ClipCursor" (ByVal lpRect As Long)
说明
将指针限制到指定区域。ClipCursorBynum 是一个别名，允许我们清除以前设置的指针剪切区域
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpRectRECT，指定一个矩形，用像素屏幕坐标系统表示。鼠标指针必须在这个区域内运动。如使用函数的 ClipCursorBynum 形式，则可将参数设为 Long 值，用它传递一个 0，禁止指针剪切，恢复常规运作状态
注解
指针剪切后，按 Ctrl+Alt+Del 可简单的清除剪切区域

Top
ConvertDefaultLocale
ConvertDefaultLocale

VB 声明
Declare Function ConvertDefaultLocale Lib "KERNEL32" Alias "ConvertDefaultLocale" (ByVal Locale As Long) As Long
说明
将一个特殊的地方标识符转换成真实的地方 ID
返回值
Long，如执行成功，返回实际的地方 ID。如失败则返回传递给 Locale 参数的值
参数表
参数类型及说明
LocaleLong，如设为常数 LOCALE_SYSTEM_DEFAULT 和 LOCALE_USER_DEFAULT，可

分别接收默认的系统或用户地方设置。如设为零，则取得语言方面的地方设置。主语言 ID 用于接收采用了默认子语言的地方信息
注解
用 LOCALE_SYSTEM_DEFAULT 和 LOCALE_USER_DEFAULT 这两个值调用函数时，得到的结果与 GetSystemDefaultLCID 和 GetUserDefaultLCID 函数是一样的

Top
CreateCaret
CreateCaret

VB 声明
Declare Function CreateCaret Lib "user32" Alias "CreateCaret" (ByVal hwnd As Long, ByVal hBitmap As Long, ByVal nWidth As Long, ByVal nHeight As Long) As Long
说明
根据指定的信息创建一个插入符（光标），并将它选定为指定窗口的默认插入符。插入符可以是一根短线、一个方块或者一幅位图。通常用插入符指示文字在文字框中的插入位置
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，拥有插入符的那个窗口的句柄
hBitmapLong，用作插入符的一幅位图的句柄。可以是 0 或 1；在这种情况下，插入符可通过 nWidth 和 nHeight 参数创建。如设为 1，则新插入符以灰色显示；而不是传统的黑色
nWidthLong，采用逻辑单位的插入符的宽度
nHeightLong，采用逻辑单位的插入符的高度
注解
如创建一个插入符，会同时清除原先的插入符；效果等同于 DestroyCaret 函数。在 vb 的 LostFocus 事件期间，不要试图用 DestroyCaret 函数清除一个插入符。这是由于 vb 的 LostFocus 事件不会接收 DestroyCaret 直到另一个窗口已经有焦点。因此，倘若在那个时候调用 DestroyCaret，会破坏其他窗口的插入符。如果准备自己管理插入符，可以（而且应该）在 WM_KILLFOCUS 消息期间清除插入符
在 vb 里使用
可以使用。但在应用程序切换时，标准的 vb 文本控件不能处理 GotFocus 和 LostFocus 事件。因此，很难知道何时为一个控件设置插入符。此外，vb 假设插入符为一根短竖线，并据此定义它的位置，所以其他形式的插入符可能无法正确定位

Top
DestroyCaret
DestroyCaret

VB 声明
Declare Function DestroyCaret Lib "user32" Alias "DestroyCaret" () As Long
说明
清除（破坏）一个插入符

返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
注解
参考 CreateCaret 函数的注解

Top
EnumCalendarInfo
EnumCalendarInfo

VB 声明
Declare Function EnumCalendarInfo Lib "kernel32" Alias "EnumCalendarInfoA" (ByVal lpCalInfoEnumProc As Long, ByVal Locale As Long, ByVal Calendar As Long, ByVal CalType As Long) As Long
说明
枚举在指定“地方”环境中可用的日历信息
返回值
Long，TRUE（非零）表示成功，零表示失败。会将 GetLastError 设置设为下述某个常数：ERROR_BADDB，ERROR_INVALID_FLAGS，ERROR_INVALID_PARAMETER
参数表
参数类型及说明
lpCalInfoEnumProcLong，指向为每个日历都调用的一个函数的指针。用 Addressof 运算符得到位于一个标准模块中的函数的地址
LocaleLong，用于决定具体枚举日历的“地方”设置
CalendarLong，如设为常数 ENUM_ALL_CALANDARS，表示枚举所有日历；如设为 1，枚举本地罗马日历，2 枚举英国罗马日历，3 枚举日本日历，4 枚举中国日历，5 枚举朝鲜日历
CalTypeLong，CAL_函数之一，指示欲枚举的信息类型

Top
EnumDateFormats
EnumDateFormats

VB 声明
Declare Function EnumDateFormats Lib "KERNEL32" Alias "EnumDateFormats" (ByVal lpDateFmtEnumProc As Long, ByVal Locale As Long, ByVal dwFlags As Long) As Long
说明
列举指定的“当地”设置中可用的长、短日期格式
返回值
Long，TRUE（非零）表示成功，零表示失败。会将 GetLastError 设置设为下述某个常数：ERROR_BADDB，ERROR_INVALID_FLAGS，ERROR_INVALID_PARAMETER
参数表
参数类型及说明
lpDateFmtEnumProcLong，指向为每种日期格式都调用的一个函数的指针。用 Addressof 运算符得到位于一个标准模块中的函数的地址
LocaleLong，用于决定具体枚举格式的“地方”的设置

dwFlagsLong，DATE_SHORTDATE 或 DATE_LONGDATE 常数之一，分别用于枚举短或长日期格式

Top
EnumSystemCodePages
EnumSystemCodePages

VB 声明
Declare Function EnumSystemCodePages Lib "KERNEL32" Alias "EnumSystemCodePages" (ByVal lpCodePageEnumProc As Long, ByVal dwFlags As Long) As Long
说明
枚举系统中已安装或支持的代码页
返回值
Long，TRUE（非零）表示成功，零表示失败。会将 GetLastError 设置设为下述某个常数：ERROR_BADDB，ERROR_INVALID_FLAGS，ERROR_INVALID_PARAMETER
参数表
参数类型及说明
lpCodePageEnumProcLong，指向为每个代码页都调用的一个函数的指针。用 Addressof 运算符得到位于一个标准模块中的函数的地址
dwFlagsLong，常数 CP_INSTALLED 表示枚举已安装的所有代码页，CP_SUPPORTED 表示枚举所有支持的代码页

Top
EnumSystemLocales
EnumSystemLocales

VB 声明
Declare Function EnumSystemLocales Lib "KERNEL32" Alias "EnumSystemLocales" (ByVal lpLocaleEnumProc As Long, ByVal dwFlags As Long) As Long
说明
枚举系统已经安装或提供支持的“地方”设置
返回值
Long，TRUE（非零）表示成功，零表示失败。会将 GetLastError 设置设为下述某个常数：ERROR_BADDB，ERROR_INVALID_FLAGS，ERROR_INVALID_PARAMETER
参数表
参数类型及说明
lpLocaleEnumProcLong，指向为每个“地方”都调用的一个函数的指针。用 Addressof 运算符得到位于一个标准模块中的函数的地址
dwFlagsLong，常数 LCID_INSTALLED 表示枚举已安装的所有地方；LCID_SUPPORTED 则枚举所有支持的地方

Top
EnumTimeFormats
EnumTimeFormats

VB 声明
Declare Function EnumTimeFormats Lib "KERNEL32" Alias "EnumTimeFormats" (ByVal lpTimeFmtEnumProc As Long, ByVal Locale As Long, ByVal dwFlags As Long) As Long
说明
枚举一个指定的地方适用的时间格式
返回值
Long，TRUE（非零）表示成功，零表示失败。会将 GetLastError 设置设为下述某个常数：ERROR_BADDB，ERROR_INVALID_FLAGS，ERROR_INVALID_PARAMETER
参数表
参数类型及说明
lpTimeFmtEnumProcLong，指向为每种时间格式都调用的一个函数的指针。用 Addressof 运算符得到位于一个标准模块中的函数的地址
LocaleLong，用于决定具体枚举格式的“地方”设置
dwFlagsLong，未用，设为零

Top
ExitWindowsEx
ExitWindowsEx

VB 声明
Declare Function ExitWindowsEx Lib "user32" Alias "ExitWindowsEx" (ByVal uFlags As Long, ByVal dwReserved As Long) As Long
说明
退出 windows，并用特定的选项重新启动
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
uFlagsLong，指定下述一个或多个标志（用 OR 运算符合并到一起）
EWX_FORCE 强迫中止没有响应的进程
EWX_LOGOFF 中止进程，然后注销
EWX_SHUTDOWN 关掉系统电源（如果可能的话，ATX 电源就可以）
EWX_REBOOT 重新引导系统
EWX_SHUTDOWN 关闭系统
dwReservedLong，保留，设为零
注解
这个函数调用后会立刻返回，系统关闭过程是在后台进行的。注意先中止自己的应用程序，使关闭过程更显平顺。当然，您的进程必须有足够的优先权，否则也不能执行这种操作

Top
ExpandEnvironmentStrings
ExpandEnvironmentStrings

VB 声明
Declare Function ExpandEnvironmentStrings Lib "kernel32" Alias "ExpandEnvironmentStringsA" (ByVal lpSrc As String, ByVal lpDst As String, ByVal nSize As Long) As Long
说明
扩充环境字串。具体操作过程与命令行处理的所为差不多。也就是说，将由百分号封闭起来的环境变量名转换成那个变量的内容。比如，“%path%”会扩充成完整路径。在 vb 里经常用于为新进程创建一个环境块
返回值
Long，lpDst 要求的缓冲区的大小。如 nSize 小于这个数字（也就是说，缓冲区太小，以至不能全容下扩充过后的字串），那么 lpDst 不会被载入。可利用这个结果改变字串的大小。零表示遇到错误。会设置 GetLastError
参数表
参数类型及说明
lpSrcString，欲扩充的字串
lpDstString，扩充过后的字串
nSizeLong，lpDst 的长度。注意预先对 lpDst 进行初始化，使其与这个长度相符
示例
```
Dim s$, dl&
Dim y As String * 5
s$ = "%PATH%"
dl& = ExpandEnvironmentStrings(s$, y, 499)
Print y
```

Top
FreeEnvironmentStrings
FreeEnvironmentStrings

VB 声明
Declare Function FreeEnvironmentStrings& Lib "kernel32" Alias "FreeEnvironmentStringsA" (ByVal lpsz As Long)
说明
翻译指定的环境字串块
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpszLong，指向一个内存块的句柄；那个内存块是以前通过 GetEnvironmentStrings 函数获得的
注解
开头的声明来自于我的资料，而在 vb 自带的 api 文本查看器中的声明为：Declare Function FreeEnvironmentStrings Lib "kernel32" Alias "FreeEnvironmentStringsA" (ByVal lpsz As String) As Long，lpsz 的类型为 String，不知是谁的错了

Top
GetACP
GetACP

VB 声明
Declare Function GetACP Lib "kernel32" Alias "GetACP" () As Long
说明
判断目前正在生效的 ANSI 代码页
返回值
Long，目前活动 ANSI 代码页的标识符。针对一种特定的语言，可能存在多个这样的代码页。可能的代码页包括下面这些：
874 泰语 932 日语
936 中文（简体）949 朝鲜语
950 中文（台湾和香港繁体）1200Unicode
1250 东欧语言 1251 西里尔语
1252 美国和西欧语言 1253 希腊语
1254 土耳其语 1255 希伯来语
1256 阿拉伯语 1257 波罗的语
注解
不要混淆 ANSI 代码页与 OEM 代码页的概念！ANSI 代码页为不同版本的 windows 定义标准的 ANSI 8 位字符集。而 OEM 代码页指定基础 DOS 代码页，由系统及键盘使用

Top
GetAsyncKeyState
GetAsyncKeyState

VB 声明
Declare Function GetAsyncKeyState Lib "user32" Alias "GetAsyncKeyState" (ByVal vKey As Long) As Integer
说明
判断函数调用时指定虚拟键的状态
返回值
Long，自对 GetAsyncKeyState 函数的上一次调用以来，如键已被按过，则位 0 设为 1；否则设为 0。如键目前处于按下状态，则位 15 设为 1；如抬起，则为 0。微软的 win32 手册指出：倘若输入焦点从属于与调用函数的输入线程不同的另一个输入线程，则返回值为 0（例如，一旦另一个程序拥有焦点，则它应返回零）。证据显示，函数实际是在整个系统的范围内工作的
参数表
参数类型及说明
vKeyLong，欲测试的虚拟键的键码
注解
如指定了 VK_LBUTTON 或 VK_RBUTTON，按钮的状态就会根据实际的按钮报告——无论是否曾用 SwapMouseButton 函数对鼠标的位置进行了交换。win32 提供了额外的一些虚拟键码，比如 VK_LSHIFT 和 VK_RSHIFT，以便在两个完全一样的键中区分出左右（也包括

Ctrl 和 Alt）

Top
GetCaretBlinkTime
GetCaretBlinkTime

VB 声明
Declare Function GetCaretBlinkTime Lib "user32" Alias "GetCaretBlinkTime" () As Long
说明
判断插入符光标的闪烁频率
返回值
Long，插入符连续两次闪烁间隔的时间，以毫秒为单位。零表示函数调用失败。会设置 GetLastError

Top
GetCaretPos
GetCaretPos

VB 声明
Declare Function GetCaretPos Lib "user32" Alias "GetCaretPos" (lpPoint As POINTAPI) As Long
说明
判断插入符的当前位置
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpPointPOINTAPI，这个结构会随同插入符在窗口客户坐标系统中的位置载入；那个窗口是插入符的父窗口

Top
GetClipCursor
GetClipCursor

VB 声明
Declare Function GetClipCursor Lib "user32" Alias "GetClipCursor" (lprc As RECT) As Long
说明
取得一个矩形，用于描述目前为鼠标指针规定的剪切区域；该区域是由 SetClipCursor 函数定义的
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lprcRECT，在屏幕坐标系统中随同当前剪切矩形载入的一个矩形。倘若没有活动的剪切，这个矩形会反映出整个显示屏幕的大小

Top
GetCommandLine
GetCommandLine

VB 声明
Declare Function GetCommandLine Lib "kernel32" Alias "GetCommandLineA" () As String
说明
获得指向当前命令行缓冲区的一个指针
返回值
Long，命令行缓冲区在内存中的地址
注解
Visual Basic Command 函数更易获取参数，但它未提供可执行的名称。使用这个函数时，要求进行内存复制操作

Top
GetComputerName
GetComputerName

VB 声明
Declare Function GetComputerName Lib "kernel32" Alias "GetComputerNameA" (ByVal lpBuffer As String, nSize As Long) As Long
说明
取得这台计算机的名称
返回值
Long，TRUE（非零）表示成功，否则返回零。会设置 GetLastError
参数表
参数类型及说明
lpBufferString，随同计算机名载入的字串缓冲区
nSizeLong，缓冲区的长度。这个变量随同返回计算机名的实际长度载入
注解
注意 nSize 参数并不是按值传递的。参考 api32.txt，了解 MAX_COMPUTER_NAME 常数的值
示例
```
Dim s$
s$ = String$(MAX_COMPUTERNAME_LENGTH+1,0)
Dim dl&
Dim sz&
sz& = MAX_COMPUTERNAME_LENGTH+1
dl& = GetComputerName(s$, sz)
```
其他
也许你会发现，MAX_COMPUTERNAME_LENGTH 常数在 vb 自带的 api 文本查看器中找不到。的确，我也没有找到。但我有一个工具：Listapi，这个常数在它那里可以找到

Top
GetCPInfo
GetCPInfo

VB 声明
Declare Function GetCPInfo Lib "kernel32" Alias "GetCPInfo" (ByVal CodePage As Long, lpCPInfo As CPINFO) As Long
说明
取得与指定代码页有关的信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
CodePageLong，欲为其载入信息的代码页的标识符。可能是一个 ANSI 或 OEM 代码页
lpCPInfoCPINFO，用于容纳代码页信息的一个结构

Top
GetCurrencyFormat
GetCurrencyFormat, GetCurrencyFormatBynum

VB 声明
Declare Function GetCurrencyFormat& Lib "kernel32" Alias "GetCurrencyFormatA" (ByVal Locale As Long, ByVal dwFlags As Long, ByVal lpValue As String, lpFormat As CURRENCYFMT, ByVal lpCurrencyStr As String, ByVal cchCurrency As Long)
Declare Function GetCurrencyFormatBynum& Lib "kernel32" Alias "GetCurrencyFormatA" (ByVal Locale As Long, ByVal dwFlags As Long, ByVal lpValue As String, ByVal lpFormat As Long, ByVal lpCurrencyStr As String, ByVal cchCurrency As Long)
说明
针对指定的“地方”设置，根据货币格式格式化一个数字
返回值
Long，格式化过后的字串的长度。零表示出错。会将 GetLastError 设置为下述值之一：ERROR_INSUFFICIENT_BUFFER，ERROR_INVALID_FLAGS，ERROR_INVALID_PARAMETER
参数表
参数类型及说明
LocaleLong，用于决定格式的地方 ID。lpFormat 参数中指定的任何信息（倘若不是 NULL）都优先于特定于地方的信息
dwFlagsLong，如指定了 lpFormat，那么该参数应该为零。否则，可设为 LOCALE_NOUSEROVERRIDE，强制使用系统地方参数——即使它们已由用户取代
lpValueString，指定欲格式化的数字。可以只有数位、一个前缀“-”号以及一个小数点
lpFormatCURRENCYFMT，可设为 NULL，使用特定于不同地方的值（用 GetCurrencyFormatBynum，则可通过 ByVal As Long 形式传递这个参数）。否则，可引用一个 CURRENCYFMT 结构，其中包含所有必要的字段，可填入需要用到的信息
lpCurrencyStrString，指定一个缓冲区，用于容纳格式化过后的字串。注意事先对字串进行

初始化
cchCurrencyLong，lpCurrencyStr 缓冲区的长度。如为零，表示函数会返回需要缓冲区的大小
注解
在 vb 里，如使用一个别名，其中的 lpFormat 设为 NULL，则可以正常使用。CURRENCYFMT 结构的正确预初始化非常具有挑战性

Top
GetCursor
GetCursor

VB 声明
Declare Function GetCursor Lib "user32" Alias "GetCursor" () As Long
说明
获取目前选择的鼠标指针的句柄
返回值
Long，目前使用的指针的句柄。倘若不存在指针，则返回零
注解
这个函数返回的是当前线程的指针——不能获取其他应用程序的指针

Top
GetCursorPos
GetCursorPos

VB 声明
Declare Function GetCursorPos Lib "user32" Alias "GetCursorPos" (lpPoint As POINTAPI) As Long
说明
获取鼠标指针的当前位置
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpPointPOINTAPI，随同指针在屏幕像素坐标中的位置载入的一个结构

Top
GetDateFormat
GetDateFormat

VB 声明
Declare Function GetDateFormat Lib "kernel32" Alias "GetDateFormatA" (ByVal Locale As Long, ByVal dwFlags As Long, lpDate As SYSTEMTIME, ByVal lpFormat As String, ByVal lpDateStr As String, ByVal cchDate As Long) As Long
说明

针对指定的“当地”格式，对一个系统日期进行格式化
返回值
Long，格式化过后的字串的长度。零表示出错，会将 GetLastError 设置为下述值之一：ERROR_INSUFFICIENT_BUFFER ， ERROR_INVALID_FLAGS ， ERROR_INVALID_PARAMETER
参数表
参数类型及说明
LocaleLong，用于决定格式的地方 ID。lpFormat 参数中指定的任何信息（倘若不是 NULL）都优先于特定于地方的信息
dwFlagsLong， 如指定了 lpFormat， 那么该参数应该为零。否则，可设为 LOCALE_NOUSEROVERRIDE，强制使用系统地方参数——即使它们已由用户取代。用 DATE_SHORTDATE 或 DATE_LONGDATE 选择不同的日期格式
lpDateSYSTEMTIME，包含了一个系统日期的结构
lpFormatString，可设为 NULL，使用特定于不同地方的值（用 vbNullString 传递一个 NULL）。否则包含一个日期格式字串。对 d,dd,ddd,dddd,m,mm,mmm,mmmm,y,yy,yyyy 这样的代码，它们的用法与在 vb 格式命令中的用法是相同的。注意用 gg 指定一个“纪元”
lpDateStrString，指定一个缓冲区，用于容纳格式化过后的字串。注意事先对字串进行初始化
cchDateLong，lpDateStr 缓冲区的长度。如为零，表示函数会返回需要缓冲区的大小

Top
GetDoubleClickTime
GetDoubleClickTime

VB 声明
Declare Function GetDoubleClickTime Lib "user32" Alias "GetDoubleClickTime" () As Long
说明
判断连续两次鼠标单击之间会被处理成双击事件的间隔时间
返回值
Long，以毫秒表示的双击时间

Top
GetEnvironmentStrings
GetEnvironmentStrings

VB 声明
Declare Function GetEnvironmentStrings& Lib "kernel32" Alias "GetEnvironmentStringsA" ()
说明
为包含了当前环境字串设置的一个内存块分配和返回一个句柄。这个内存块包含了所有环境字串。各字串间用一个 NULL 分隔；连续两个 NULL 标志着列表的结尾
返回值
Long，指向包含了环境字串的一个内存块的地址。零表示失败。会设置 GetLastError
注解
注意一定要用 FreeEnvironmentStrings 函数释放这个内存块

其他
请看从 vb 的 api 文本查看器复制的声明：Declare Function GetEnvironmentStrings Lib "kernel32" Alias "GetEnvironmentStringsA" () As String，与前面的声明返回值不同

Top
GetEnvironmentVariable
GetEnvironmentVariable

VB 声明
Declare Function GetEnvironmentVariable Lib "kernel32" Alias "GetEnvironmentVariableA" (ByVal lpName As String, ByVal lpBuffer As String, ByVal nSize As Long) As Long
说明
取得一个环境变量的值
返回值
Long，载入的环境变量的长度。如指定的环境字串不存在，就返回零。如 lpBuffer 的长度不足以全部容下字串，则返回字串的全长。随后可用这个长度分配一个足够大的缓冲区
参数表
参数类型及说明
lpNameString，欲读入的环境字串的名称
lpBufferString，随同字串装载的一个缓冲区。注意预先将其初始化到合适的长度
nSizeLong，lpBuffer 的长度

Top
GetInputState
GetInputState

VB 声明
Declare Function GetInputState Lib "user32" Alias "GetInputState" () As Long
说明
判断是否存在任何待决（等待处理）的鼠标或键盘事件
返回值
Long，非零表示成功，零表示失败
注解
在 win32 下，这个函数只返回当前输入线程的状态

Top
GetKBCodePage
GetKBCodePage

VB 声明
Declare Function GetKBCodePage Lib "user32" Alias "GetKBCodePage" () As Long
说明
由 GetOEMCP 取代，两者功能完全相同

Top
GetKeyboardLayout
GetKeyboardLayout

VB 声明
Declare Function GetKeyboardLayout Lib "user32" Alias "GetKeyboardLayout" (ByVal dwLayout As Long) As Long
说明
取得一个句柄，描述指定应用程序的键盘布局
返回值
Long，键盘布局的句柄
参数表
参数类型及说明
dwLayoutLong，欲检查的线程的标识符

Top
GetKeyboardLayoutList
GetKeyboardLayoutList

VB 声明
Declare Function GetKeyboardLayoutList Lib "user32" Alias "GetKeyboardLayoutList" (ByVal nBuff As Long, lpList As Long) As Long
说明
获得系统适用的所有键盘布局的一个列表
返回值
Long，装载到内存的键盘布局的数量
参数表
参数类型及说明
nBuffLong，lpList 数组中的条目数量。如设为零，表示获取可用键盘布局的数量
lpListLong，指定一个数组，它的元素数量至少应有 nBuff 规定的元素那么多。这个数组会随同句柄载入可用的键盘布局

Top
GetKeyboardLayoutName
GetKeyboardLayoutName

VB 声明
Declare Function GetKeyboardLayoutName Lib "user32" Alias "GetKeyboardLayoutNameA" (ByVal pwszKLID As String) As Long
说明
取得当前活动键盘布局的名称
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表

参数类型及说明
pwszKLIDString，长度为 KL_NAMELENGTH 个字符的字串
注解
在 NT 中，键盘布局与特定的应用程序有关。而在 windows95 中，它取决于特定的线程

Top
GetKeyboardState
GetKeyboardState

VB 声明
Declare Function GetKeyboardState Lib "user32" Alias "GetKeyboardState" (pbKeyState As Byte) As Long
说明
取得键盘上每个虚拟键当前的状态
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pbKeyStateLong，总共含 256 个条目的字节数组中的第一个项目。每个字节都会附带载入与虚拟键对应的状态。如开关键打开，则位 0 设为 1（开关键包括 CapsLock， NumLock，ScrollLock）；如某个键当时按下，则位 7 为 1；如已经抬起，则为 0
注解
虚拟键码常数 VK_? 作为数组的索引使用。这个函数相应于取得按键状态于一瞬间的“快照”——键按下或松开以后，数组不会自动更新。在 win32 中注意用一个字节数组避免由于 vb 向 Unicode 的内部转换而导致错误

Top
GetKeyboardType
GetKeyboardType

VB 声明
Declare Function GetKeyboardType Lib "user32" Alias "GetKeyboardType" (ByVal nTypeFlag As Long) As Long
说明
了解与正在使用的键盘有关的信息
返回值
Long，零表示出错。否则返回下述值之一
nTypeFlag=01——PC 或兼容的 83 键键盘；2——Olivetti102 键键盘；3——AT 或兼容 84 键键盘；4——增强型（IBM）101 或 102 键键盘；5——Nokia1050 键盘；6——Nokia9140 键盘；7——日文键盘
nTypeFlag=1 任何值，由厂商决定
nTypeFlag=21——10 个功能键（即 F? 键）；2——12 或 18 个功能键；3——10 个功能键；4——12 个功能键；5——10 个功能键；6——24 个功能键；7——由厂商决定
参数表

参数类型及说明
nTypeFlagLong，可设为下述值之一
0——返回键盘类型
1——返回键盘子类型
2——返回键盘上的功能键数量

Top
GetKeyNameText
GetKeyNameText

VB 声明
Declare Function GetKeyNameText Lib "user32" Alias "GetKeyNameTextA" (ByVal lParam As Long, ByVal lpBuffer As String, ByVal nSize As Long) As Long
说明
在给出扫描码的前提下，判断键名
返回值
Long，lpBuffer 中载入的键名的实际长度
参数表
参数类型及说明
lParamLong，位 0 到 5=0；位 16 到 23=按键的扫描码；位 24=增强型键盘上的扩展位；位 25=如设为 1，表示忽略左右 Shift 和 Ctrl 键的区别
lpBufferString，字串预先初始化成至少 nSize+1 字节，以便随同键名载入
nSizeLong，字串的最大长度

Top
GetKeyState
GetKeyState

VB 声明
Declare Function GetKeyState Lib "user32" Alias "GetKeyState" (ByVal nVirtKey As Long) As Integer
说明
针对已处理过的按键，在最近一次输入信息时，判断指定虚拟键的状态
返回值
Integer，如开关键打开，则位 0 设为 1（开关键包括 CapsLock， NumLock，ScrollLock）；如某个键当时正处于按下状态，则位 15 为 1；如已经抬起，则为 0
参数表
参数类型及说明
nVirtKeyLong，欲测试的虚拟键键码。对字母、数字字符（A-Z、a-z、0-9），用它们实际的 ASCII 值

Top
GetLastError
GetLastError

VB 声明
Declare Function GetLastError Lib "kernel32" Alias "GetLastError" () As Long
说明
针对之前调用的 api 函数，用这个函数取得扩展错误信息（在 vb 里使用：在 vb 中，用 Err 对象的 GetLastError 属性获取 GetLastError 的值。这样做是必要的，因为在 api 调用返回以及 vb 调用继续执行期间，vb 有时会重设 GetLastError 的值）
返回值
Long，由 api 函数决定。请参考 api32.txt 文件，其中列出了一系列错误常数；都以 ERROR_ 前缀起头。常用的错误代码见下表
ERROR_INVALID_HANDLE 无效的句柄作为一个参数传递
ERROR_CALL_NOT_IMPLEMENTED 在 win 95 下调用专为 win nt 设计的 win32 api 函数
ERROR_INVALID_PARAMETER 函数中有个参数不正确
注解
GetLastError 返回的值通过在 api 函数中调用 SetLastError 或 SetLastErrorEx 设置。函数并无必要设置上一次错误信息，所以即使一次 GetLastError 调用返回的是零值，也不能担保函数已成功执行。只有在函数调用返回一个错误结果时，这个函数指出的错误结果才是有效的。通常，只有在函数返回一个错误结果，而且已知函数会设置 GetLastError 变量的前提下，才应访问 GetLastError；这时能保证获得有效的结果。SetLastError 函数主要在对 api 函数进行模拟的 dll 函数中使用，所以对 vb 应用程序来说是没有意义的

Top
GetLocaleInfo
GetLocaleInfo

VB 声明
Declare Function GetLocaleInfo Lib "kernel32" Alias "GetLocaleInfoA" (ByVal Locale As Long, ByVal LCType As Long, ByVal lpLCData As String, ByVal cchData As Long) As Long
说明
取得与指定“地方”有关的信息
返回值
Long，装载到缓冲区的字符数，或者 cchData 要求的缓冲区长度。零表示出错。会将 GetLastError 设为下述值之一：ERROR_INSUFFICIENT_BUFFER，ERROR_INVALID_FLAGS，ERROR_INVALID_PARAMETER
参数表
参数类型及说明
LocaleLong，欲为其获得信息的地方 ID
LCTypeLong，要取回的信息类型。参考 api32.txt 文件中带 LOCALE_ 前缀的常数。用 OR 运算符合并 LOCALE_NOUSEROVERRIDE，从而强制使用系统默认信息——即使当前用户已修改了相关设置
lpLCDataString，指定一个缓冲区，用于装载要求的信息。注意预先将字串格式化成合适的长度
cchDataLong，lpLCData 缓冲区的长度；如设为零，表示获取必要的缓冲区长度

Top
GetLocalTime
GetLocalTime

VB 声明
Declare Sub GetLocalTime Lib "kernel32" Alias "GetLocalTime" (lpSystemTime As SYSTEMTIME)
说明
在 lpSystemTime 结构中装载本地日期和时间
参数表
参数类型及说明
lpSystemTimeSYSTEMTIME，用于装载本地时间的结构

Top
GetNumberFormat
GetNumberFormat,GetNumberFormatBynum

VB 声明
Declare Function GetNumberFormat& Lib "kernel32" Alias "GetNumberFormatA" (ByVal Locale As Long, ByVal dwFlags As Long, ByVal lpValue As String, lpFormat As NUMBERFMT, ByVal lpNumberStr As String, ByVal cchNumber As Long)
Declare Function GetNumberFormatBynum& Lib "kernel32" Alias "GetNumberFormatA" (ByVal Locale As Long, ByVal dwFlags As Long, ByVal lpValue As String, ByVal lpFormat As Long, ByVal lpNumberStr As String, ByVal cchNumber As Long)
说明
针对指定的“地方”，按特定的格式格式化一个数字
返回值
Long，格式化过后的字串的长度。零表示出错。会将 GetLastError 设为下述值之一：ERROR_INSUFFICIENT_BUFFER，ERROR_INVALID_FLAGS，ERROR_INVALID_PARAMETER
参数表
参数类型及说明
LocaleLong，决定了具体格式的地方 ID。lpFormat 参数（如果不为 NULL）指定的任何信息都优先于各“地方”不同的特定信息
dwFlagsLong，如指定了 lpFormat，这个参数应为零。否则，可设为 LOCALE_NOUSEROVERRIDE，强制使用系统地方参数——无论用户是否已作出相应的修改
lpValueString，欲格式化的数字。数字可以只有数位、加在前面的一个“-”号以及一个小数点
lpFormatNUMBERFMT，可设为 NULL，表示使用本地特有的值（倘若用 GetNumberFormatBynum，则可将这个参数以 ByVal As Long 的形式传递）。否则，可引用一个 NUMBERFMT 结构，其中的各个字段载入欲使用的格式信息
lpNumberStrString，指定一个缓冲区，用于装载格式化过后的字串。注意先初始化成合适的长度

cchNumberLong，lpNumberStr 缓冲区的长度。如为零，函数会返回缓冲区必要的长度

Top
GetOEMCP
GetOEMCP

VB 声明
Declare Function GetOEMCP Lib "kernel32" Alias "GetOEMCP" () As Long
说明
判断在 OEM 和 ANSI 字符集间转换的 windows 代码页
返回值
Long，目前处于活动状态的 OEM 代码页的标识符。针对一种特定的语言，可能存在多个代码页。以下是可用代码页列表
437 默认：美国 708-720 阿拉伯代码页 737 希腊
775 波罗的 850 国际 852Slavic
855 西里尔语 857 土耳其语 860 葡萄牙语
861 冰岛语 862 希伯来语 863 加拿大法语
864 阿拉伯语 865 挪威/丹麦语 866 俄语
874 泰语 932 日语 936 中文（简体）
949 朝鲜语 950 中文（台、港繁体）1361 朝鲜语

Top
GetQueueStatus
GetQueueStatus

VB 声明
Declare Function GetQueueStatus Lib "user32" Alias "GetQueueStatus" (ByVal fuFlags As Long) As Long
说明
判断应用程序消息队列中待决（等待处理）的消息类型
返回值
Long，高字是一个 16 位的旗标字，包含了待决的消息。其中的各个位是由为 fuFlags 参数定义的同样的常数决定的。低字是一个对应的旗标字。其中各个位指出自上次调用这个函数以来，或自消息上一次处理以来，新加入的待处理消息
参数表
参数类型及说明
fuFlagsLong，一个标志（旗标）字，指定要检查的消息。标志位是由下述常数定义的
QS_KEYWM_CHAR 消息（会造成 vb KeyPressed 事件）
QS_MOUSE 任何鼠标消息
QS_MOUSEMOVEMouseMove 消息或事件
QS_MOUSEBUTTON 鼠标按钮消息或相关事件
QS_PAINT 等待处理的 Paint 消息
QS_POSTMESSAGE 投递的其他消息
QS_SENDMESSAGE 从另一个应用程序中发出的消息

QS_TIMER 计时器消息
QS_HOTKEY 队列中的一条 Hotkey 消息
注解
在 vb 里这个函数不特别有用（Use with VB:Not particularly useful.）

Top
GetSysColor
GetSysColor

VB 声明
Declare Function GetSysColor Lib "user32" Alias "GetSysColor" (ByVal nIndex As Long) As Long
说明
判断指定 windows 显示对象的颜色
返回值
Long，指定对象的 RGB 颜色
参数表
参数类型及说明
nIndexLong，一个常数，指出特定的 windows 显示对象，如下表
Windows 对象常数表
常数定义 Windows 对象常数定义 Windows 对象
COLOR_ACTIVEBORDER 活动窗口的边框 COLOR_ACTIVECAPTION 活动窗口的标题
COLOR_APPWORKSPACEMDI 桌面的背景 COLOR_BACKGROUNDwindows 桌面
COLOR_BTNFACE 按钮 COLOR_BTNHIGHLIGHT 按钮的 3D 加亮区
COLOR_BTNSHADOW 按钮的 3D 阴影 COLOR_BTNTEXT 按钮文字
COLOR_CAPTIONTEXT 窗口标题中的文字 COLOR_GRAYTEXT 灰色文字；如使用了抖动技术则为零
COLOR_HIGHLIGHT 选定的项目背景 COLOR_HIGHLIGHTTEXT 选定的项目文字
COLOR_INACTIVEBORDER 不活动窗口的边框 COLOR_INACTIVECAPTION 不活动窗口的标题
COLOR_INACTIVECAPTIONTEXT 不活动窗口的文字 COLOR_MENU 菜单
COLOR_MENUTEXT 菜单正文 COLOR_SCROLLBAR 滚动条
COLOR_WINDOW 窗口背景 COLOR_WINDOWFRAME 窗框
COLOR_WINDOWTEXT 窗口正文 COLOR_3DDKSHADOW3D 深阴影 *
COLOR_3DFACE3D 阴影化对象的正面颜色 *COLOR_3DHILIGHT3D 加亮颜色（win95）
COLOR_3DLIGHT3D 阴影化对象的浅色 *COLOR_INFOBK 工具提示的背景色 *
COLOR_INFOTEXT 工具提示的文本色 *
*：带 * 号的常数未获 NT 3.51 的支持

Top
GetSystemDefaultLangID
GetSystemDefaultLangID

VB 声明

Declare Function GetSystemDefaultLangID Lib "kernel32" Alias "GetSystemDefaultLangID" () As Integer
说明
取得系统的默认语言 ID
返回值
Integer，系统的默认语言 ID

Top
GetSystemDefaultLCID
GetSystemDefaultLCID

VB 声明
Declare Function GetSystemDefaultLCID Lib "kernel32" Alias "GetSystemDefaultLCID" () As Long
说明
取得当前的默认系统“地方”
返回值
Long，默认的系统地方 ID

Top
GetSystemInfo
GetSystemInfo

VB 声明
Declare Sub GetSystemInfo Lib "kernel32" Alias "GetSystemInfo" (lpSystemInfo As SYSTEM_INFO)
说明
在一个 SYSTEM_INFO 结构中载入与底层硬件平台有关的信息
参数表
参数类型及说明
lpSystemInfoSYSTEM_INFO，指定一个结构，用于装载适当的系统信息

Top
GetSystemMetrics
GetSystemMetrics

VB 声明
Declare Function GetSystemMetrics Lib "user32" Alias "GetSystemMetrics" (ByVal nIndex As Long) As Long
说明
返回与 windows 环境有关的信息
返回值
Long，取决于具体的常数索引
参数表

参数类型及说明
nIndexLong，常数，指定欲获取的信息；如下表所示
nIndex 常数设置
常数定义取得信息
SM_ARRANGE 设置 windows 如何排列最小化窗口的一个标志。参考 api32.txt 中的 ARW 常数
SM_CLEANBOOT 指定启动模式。0=普通模式；1=带网络支持的安全模式
SM_CMETRICS 可用系统环境的数量
SM_CMOUSEBUTTON 鼠标按钮（按键）的数量。如没有鼠标，就为零
SM_CXBORDER, SM_CYBORDER 尺寸不可变边框的大小
SM_CXCURSOR, SM_CYCURSOR 标准指针大小
SM_CXDLGFRAME, SM_CYDLGFRAME 对话框边框的大小
SM_CXDOUBLECLK, SM_CYDOUBLECLK 双击区域的大小（参考注解）
SM_CXFRAME, SM_CYFRAME 尺寸可变边框的大小（在 win95 和 nt 4.0 中使用 SM_C?FIXEDFRAME）
SM_CXFULLSCREEN, SM_CYFULLSCREEN 最大化窗口客户区的大小
SM_CXHSCROLL, SM_CYHSCROLL 水平滚动条上的箭头大小
SM_CXHTHUMB, SM_CYHTHUMB 滚动块在水平滚动条上的大小
SM_CXICON, SM_CYICON 标准图标的大小
SM_CXICONSPACING, SM_CYICONSPACING 桌面图标之间的间隔距离。在 win95 和 nt 4.0 中是指大图标的间距
SM_CXMAXIMIZED, SM_CYMAXIMIZED 最大化窗口的默认尺寸
SM_CXMAXTRACK, SM_CYMAXTRACK 改变窗口大小时，最大的轨迹宽度
SM_CXMENUCHECK, SM_CYMENUCHECK 菜单复选号位图的大小
SM_CXMENUSIZE, SM_CYMENUSIZE 菜单栏上的按钮大小
SM_CXMIN, SM_CYMIN 窗口的最小尺寸
SM_CXMINIMIZED, SM_CYMINIMIZED 最小化的窗口必须填充进去的一个矩形小于或等于 SM_C?ICONSPACING
SM_CXMINTRACK, SM_CYMINTRACK 窗口的最小轨迹宽度
SM_CXSCREEN, SM_CYSCREEN 屏幕大小
SM_CXSIZE, SM_CYSIZE 标题栏位图的大小
SM_CXSIZEFRAME, SM_CYSIZEFRAME 具有 WS_THICKFRAME 样式的窗口的大小
SM_CXSMICON, SM_CYSMICON 小图标的大小
SM_CXSMSIZE, SM_CYSMSIZE 小标题按钮的大小
SM_CXVSCROLL, SM_CYVSCROLL 垂直滚动条中的箭头按钮的大小
SM_CYCAPTION 窗口标题的高度
SM_CYKANJIWINDOWKanji 窗口的大小（Height of Kanji window）
SM_CYMENU 菜单高度
SM_CYSMCAPTION 小标题的高度
SM_CYVTHUMB 垂直滚动条上滚动块的高度
SM_DBCSENABLED 如支持双字节则为 TRUE
SM_DEBUG 如 windows 的调试版正在运行，则为 TRUE
SM_MENUDROPALIGNMENT 如弹出式菜单对齐菜单栏项目的左侧，则为零
SM_MIDEASTENABLED 允许了希伯来和阿拉伯语

SM_MOUSEPRESENT 如安装了鼠标则为 TRUE
SM_MOUSEWHEELPRESENT 如安装了带轮鼠标则为 TRUE；只适用于 nt 4.0
SM_NETWORK 如安装了网络，则设置位 0。其他位保留未用
SM_PENWINDOWS 如装载了支持笔窗口的 DLL，则表示笔窗口的句柄
SM_SECURE 如安装了安全（保密）机制，则为 TRUE
SM_SHOWSOUNDS 强制视觉提示播放声音
SM_SLOWMACHINE 系统速度太慢，但仍在运行中（System is too slow for effective use but is being run anyway）
SM_SWAPBUTTON 如左右鼠标键已经交换，则为 TRUE
注解
双击区域指定屏幕上一个特定的显示区域，只有在这个区域内连续进行两次鼠标单击，才有可能被当作双击事件处理
其他
常 数 SM_ARRANGE, SM_CLEANBOOT, SM_CMETRICS, SM_C?MAXIMIZED, SM_C?MAXTRACK, SM_C?SIZEFRAME, SM_C?SMICON, SM_C?SMSIZE, SM_CYSMCAPTION, SM_SECURE, SM_SHOWSOUNDS, and SM_SLOWMACHINE 未获 NT 3.51 及更早版本的支持

Top
GetSystemPowerStatus
GetSystemPowerStatus

VB 声明
Declare Function GetSystemPowerStatus Lib "kernel32" Alias "GetSystemPowerStatus" (lpSystemPowerStatus As SYSTEM_POWER_STATUS) As Long
说明
获得与当前系统电源状态有关的信息。对便携式计算机来说，这些信息特别有用。在那些地方，可用这个函数了解有关电源和电池组的情况
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
lpSystemPowerStatusSYSTEM_POWER_STATUS，用于装载电源状态信息的结构
注解
结果的准确度取决于系统实际的电源管理能力

Top
GetSystemTime
GetSystemTime

VB 声明
Declare Sub GetSystemTime Lib "kernel32" Alias "GetSystemTime" (lpSystemTime As SYSTEMTIME)
说明

在一个 SYSTEMTIME 中载入当前系统时间，这个时间采用的是“协同世界时间”（即 UTC，也叫做 GMT）格式
参数表
参数类型及说明
lpSystemTimeSYSTEMTIME，随同当前时间载入的结构

Top
GetSystemTimeAdjustment
GetSystemTimeAdjustment

VB 声明
Declare Function GetSystemTimeAdjustment Lib "kernel32" Alias "GetSystemTimeAdjustment" (lpTimeAdjustment As Long, lpTimeIncrement As Long, lpTimeAdjustmentDisabled As Boolean) As Long
说明
Win32 可使内部系统时钟与一个外部的时钟信号源同步，方法是定时添加一个校准值。这个函数指定的所有时间都以 100ns（0.1ms）为单位递增
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
lpTimeAdjustmentLong，添加到内部系统时钟的时间量
lpTimeIncrementLong，校准间隔时间。等于中断时钟时间
lpTimeAdjustmentDisabledBoolean，如时间调校功能关闭，则为 TRUE
示例
```
Dim lpTimeAdjustment&, lpTimeIncrement&, lpTimeAdjustmentDisabled&
GetSystemTimeAdjustment lpTimeAdjustment, lpTimeIncrement, lpTimeAdjustmentDisabled
Print "Time adjustment: " & lpTimeAdjustment & "Increment: " & lpTimeIncrement & "Adjustment Disabled: " & lpTimeAdjustmentDisabled
```

Top
GetThreadLocale
GetThreadLocale

VB 声明
Declare Function GetThreadLocale Lib "KERNEL32" Alias "GetThreadLocale" () As Long
说明
取得当前线程的地方 ID
返回值
Long，地方标识符
注解
在 vb 下，除非从一个多线程 EXE 服务器调用，否则当前线程就是当前应用程序

Top

GetTickCount
GetTickCount

VB 声明
Declare Function GetTickCount Lib "kernel32" Alias "GetTickCount" () As Long
说明
用于获取自 windows 启动以来经历的时间长度（毫秒）
返回值
Long，以毫秒为单位的 windows 运行时间

Top
GetTimeFormat
GetTimeFormat

VB 声明
Declare Function GetTimeFormat Lib "kernel32" Alias "GetTimeFormatA" (ByVal Locale As Long, ByVal dwFlags As Long, lpTime As SYSTEMTIME, ByVal lpFormat As String, ByVal lpTimeStr As String, ByVal cchTime As Long) As Long
说明
针对当前指定的“地方”，按特定的格式格式化一个系统时间
返回值
Long，格式化过后的字串的长度。零表示出错。会将 GetLastError 设置为下述值之一：ERROR_INSUFFICIENT_BUFFER，ERROR_INVALID_FLAGS，ERROR_INVALID_PARAMETER
参数表
参数类型及说明
LocaleLong，决定了具体格式的地方 ID。lpFormat 参数中指定的任何信息（倘若不是 NULL）都要优先于各地方不同的特别信息
dwFlagsLong，如指定了 lpFormat，那么该参数应该为零。否则，可设为 LOCALE_NOUSEROVERRIDE，表示强制使用系统地方参数——即使它们已由用户更改。用 DATE_SHORTDATE 或 DATE_LONGDATE 选择不同的日期格式
lpTimeSYSTEMTIME，用于包容系统时间的一个结构
lpFormatString，可设为 NULL，使用特定于不同地方的值（用 vbNullString 传递一个 NULL）。否则包含一个时间格式字串。对 h,hh,hhh,hhhh,m,mm,s,ss 这样的代码来说，它们的用法与在 vb 格式命令中的用法是相同的。t 和 tt 用于指定一个时间段标志（A 或 AM，P 或 PM）
lpTimeStrString，指定一个缓冲区，用于容纳格式化过后的字串。注意事先对字串的长度进行正确的预初始化
cchTimeLong，lpTimeStr 缓冲区的长度。如为零，表示函数会返回需要缓冲区的大小
注解
对于 dwFlags 参数的解释可能翻译有错误。原文为：
If lpFormat is specified, this should be zero. Otherwise, may be set to LOCALE_NOUSEROVERRIDE to force the system locale parameters to be used even if they have been overridden by the user. Use the self-explanatory constants TIME_NOMINUITESORSECONDS, TIME_NOSECONDS, or

TIME_FORCE24HOURFORMAT to choose between date formats. Constant TIME_NOMARKER removes the AM or PM marker.

Top
GetTimeZoneInformation
GetTimeZoneInformation

VB 声明
Declare Function GetTimeZoneInformation Lib "kernel32" Alias "GetTimeZoneInformation" (lpTimeZoneInformation As TIME_ZONE_INFORMATION) As Long
说明
在一个 TIME_ZONE_INFORMATION 结构中载入与系统时区设置有关的信息
返回值
Long，下述常数之一：
TIME_ZONE_ID_INVALID 函数执行失败，会设置 GetLastError
TIME_ZONE_ID_UNKNOWN 时区未知（可能仍然指定了 bias 值）
TIME_ZONE_ID_STANDARD 标准时间有效
TIME_ZONE_ID_DAYLIGHT 夏令时有效
参数表
参数类型及说明
lpTimeZoneInformationTIME_ZONE_INFORMATION，用于载入时区信息的结构
注解
在 lpTimeZoneInformation 结构中为本地时间添加 bias 信息，从而获得系统时间。TIME_ZONE_INFORMATION 结构内部的 DaylightName 和 StandardName 字串肯定采用 Unicode 格式

Top
GetUserDefaultLangID
GetUserDefaultLangID

VB 声明
Declare Function GetUserDefaultLangID Lib "kernel32" Alias "GetUserDefaultLangID" () As Integer
说明
为当前用户取得默认语言 ID
返回值
Long，当前用户的语言 ID

Top
GetUserDefaultLCID
GetUserDefaultLCID

VB 声明
Declare Function GetUserDefaultLCID Lib "kernel32" Alias "GetUserDefaultLCID" () As Long

说明
取得当前用户的默认“地方”设置
返回值
Long，针对当前用户的默认地方 ID

Top
GetUserName
GetUserName

VB 声明
Declare Function GetUserName Lib "advapi32.dll" Alias "GetUserNameA" (ByVal lpBuffer As String, nSize As Long) As Long
说明
取得当前用户的名字
返回值
Long，TRUE（非零）表示成功，否则返回零。会设置 GetLastError
参数表
参数类型及说明
lpBufferString，一个字串缓冲区，预先初始化成由 nSize 指定的长度。它将用于容纳用户名
nSizeLong，初始化成 lpBuffer 的长度。返回以后，它会包含载入 lpBuffer 的字符数量
示例
```
Dim s$, cnt&, dl&
cnt& = 199
s$ = String$(200,0)
dl& = GetUserName(s$, cnt)
Debug.Print Left$(s$, cnt); cnt
```

Top
GetVersion
GetVersion

VB 声明
Declare Function GetVersion Lib "kernel32" Alias "GetVersion" () As Long
说明
判断当前运行的 Windows 和 DOS 版本
返回值
Long，低 16 位包含了 windows 版本；低字节包含了主版本号（3 代表 windows 3.10，4 代表 nt 4.0）；高字节包含了两个数位的辅助版本号（10 代表 windows 3.10，95 代表 windows 95）。高 16 位则包含了平台信息。针对 windows NT，高位设为 0；针对 windows for workgroups 上运行的 Win32s，则高位为 1
注解
在 win32 下，最好换用 GetVersionEx 函数。在 win32 下，高字不会返回 DOS 版本

Top

GetVersionEx
GetVersionEx

VB 声明
Declare Function GetVersionEx Lib "kernel32" Alias "GetVersionExA" (ByVal lpVersionInformation As OSVERSIONINFO) As Long
说明
在一个 OSVERSIONINFO 结构中载入与平台和操作系统有关的版本信息
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
lpVersionInformationOSVERSIONINFO，用于装载版本信息的结构。在正式调用函数之前，必须先将这个结构的 dwOSVersionInfoSize 字段设为结构的大小（148）

Top
HideCaret
HideCaret

VB 声明
Declare Function HideCaret Lib "user32" Alias "HideCaret" (ByVal hwnd As Long) As Long
说明
在指定的窗口隐藏插入符（光标）
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，包含了插入符的窗口的句柄。可能是零；在这种情况下，只有包容了插入符的那个窗口由活动任务拥有时，插入符才会被隐藏
注解
针对插入符的显示，windows 维护着一个内部计数器；类似于 ShowCursor 函数使用的那个。所以对 HideCaret 和 ShowCaret 的调用必须进行一番权衡

Top
IsValidCodePage
IsValidCodePage

VB 声明
Declare Function IsValidCodePage Lib "kernel32" Alias "IsValidCodePage" (ByVal CodePage As Long) As Long
说明
判断一个代码页是否有效
返回值
Long，如代码页有效，就返回 TRUE（非零）；否则返回零

参数表
参数类型及说明
CodePageLong，一个代码页的标识符。参考 GetACP 和 GetOEMCP 函数

Top
IsValidLocale
IsValidLocale

VB 声明
Declare Function IsValidLocale Lib "KERNEL32" Alias "IsValidLocale" (ByVal Locale As Long, ByVal dwFlags As Long) As Long
说明
判断地方标识符是否有效
返回值
Long，根据测试条件，如指定的地方有效，就返回 TRUE（非零）；否则返回零
参数表
参数类型及说明
LocaleLong，要检查的地方标识符
dwFlagsLong，下述常数之一：
LCID_SUPPORTED 检查“地方”是否能由系统支持
LCID_INSTALLED 检查“地方”是否已获支持并安装

Top
keybd_event
keybd_event

VB 声明
Declare Sub keybd_event Lib "user32" Alias "keybd_event" (ByVal bVk As Byte, ByVal bScan As Byte, ByVal dwFlags As Long, ByVal dwExtraInfo As Long)
说明
这个函数模拟了键盘行动
参数表
参数类型及说明
bVkByte，欲模拟的虚拟键码
bScanByte，键的 OEM 扫描码
dwFlagsLong，零；或设为下述两个标志之一
KEYEVENTF_EXTENDEDKEY 指出是一个扩展键，而且在前面冠以 0xE0 代码
KEYEVENTF_KEYUP 模拟松开一个键
dwExtraInfoLong，通常不用的一个值。api 函数 GetMessageExtraInfo 可取得这个值。允许使用的值取决于特定的驱动程序
注解
这个函数支持屏幕捕获（截图）。在 win95 和 nt4.0 下这个函数的行为不同

Top

LoadKeyboardLayout
LoadKeyboardLayout

VB 声明
Declare Function LoadKeyboardLayout Lib "user32" Alias "LoadKeyboardLayoutA" (ByVal pwszKLID As String, ByVal flags As Long) As Long
说明
载入一个键盘布局
返回值
Long，键盘布局的句柄。零表示出错
参数表
参数类型及说明
pwszKLIDString，一个 8 字符字串，用于描述键盘布局的名称。参考注解
flagsLong，下述常数的任何一种组合
KLF_ACTIVATE 载入和激活指定的布局
KLF_NOTELLSHELL 禁止一个外壳挂钩进程（a shell hook procedure）接收到 HSHELL_LANGUAGE 通告。如准备载入一系列键盘布局，就需要考虑设置这个标志，从而改善性能（不要为最后一个载入的布局设置该标志）
KLF_REORDER 将指定的活动布局移至内部键盘布局列表的起始处
KLF_REPLACELANG 如指定语言的键盘布局已经存在，则用这个将其替换。仅适用于 win95
KLF_SUBSTITUTE_OK 在注册表中使用替换信息，为这个语言载入一个由用户指定的替换键盘布局（如果存在的话），而不是载入当前这个布局
KLF_UNLOADPREVIOUS 如 KLF_ACTIVATE 已经指定并成功，则卸载前一个布局
注解
键盘布局的名称采用“ddddnnnn”的形式。其中，nnnn 代表一个语言 ID 的字串形式，而 dddd 代表一个设备代码的字串形式。标准的美国键盘名称是“00000409”
其他
键盘布局在 win95 中取决于特定的线程；在 windows nt 中，则在整个系统的范围内有效

Top
MapVirtualKey
MapVirtualKey

VB 声明
Declare Function MapVirtualKey Lib "user32" Alias "MapVirtualKeyA" (ByVal wCode As Long, ByVal wMapType As Long) As Long
说明
根据指定的映射类型，执行不同的扫描码和字符转换
返回值
Long，取决于 wMapType 参数
参数表
参数类型及说明
wCodeLong，欲转换的源字符或扫描码
wMapTypeLong，控制映射类型，如下所示

0—— wCode 是个虚拟键码。函数返回相应的扫描码
1—— wCode 是个扫描码。函数返回相应的虚拟键码
2—— wCode 是个虚拟键码。函数返回相应的 ASCII 值（未加 Shift 组合键）

Top
MapVirtualKeyEx
MapVirtualKeyEx

VB 声明
Declare Function MapVirtualKeyEx Lib "user32" Alias "MapVirtualKeyExA" (ByVal uCode As Long, ByVal uMapType As Long, ByVal dwhkl As Long) As Long
说明
根据指定的映射类型，执行不同的扫描码和字符转换
返回值
Long，取决于 uMapType 参数
参数表
参数类型及说明
uCodeLong，欲转换的源字符或代码
uMapTypeLong，控制映射类型，如下所示
0—— uCode 是个虚拟键码。函数返回相应的扫描码
1—— uCode 是个扫描码。函数返回相应的虚拟键码
2—— uCode 是个虚拟键码。函数返回相应的 ASCII 值（未加 Shift 组合键）。针对死键，高位设为 1。如果出错，返回 NULL
dwhklLong，键盘布局的句柄
注解
利用这个函数，可在扫描码及附加的虚拟键码间转换。这些虚拟键码包括 VK_LSHIFT 和 VK_RSHIFT 等。这样一来，便可为表面上两个完全一样的键（也包括 Ctrl 和 Alt）区分左键和右键

Top
MessageBeep
MessageBeep

VB 声明
Declare Function MessageBeep Lib "user32" Alias "MessageBeep" (ByVal wType As Long) As Long
说明
播放一个系统声音。系统声音的分配方案是在控制面板里决定的
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
wTypeLong，下述值之一
0xffffffff 标准响铃

MB_ICONASTERISK 系统星号声（System asterisk sound）
MB_ICONEXCLAMATION 系统惊叹声
MB_ICONHAND 系统指针声（System hand sound）
MB_ICONQUESTION 系统提问声

Top
mouse_event
mouse_event

VB 声明
Declare Sub mouse_event Lib "user32" Alias "mouse_event" (ByVal dwFlags As Long, ByVal dx As Long, ByVal dy As Long, ByVal cButtons As Long, ByVal dwExtraInfo As Long)
说明
模拟一次鼠标事件
参数表
参数类型及说明
dwFlagsLong，下述标志的一个组合
MOUSEEVENTF_ABSOLUTEdx 和 dy 指定鼠标坐标系统中的一个绝对位置。在鼠标坐标系统中，屏幕在水平和垂直方向上均匀分割成 65535×65535 个单元
MOUSEEVENTF_MOVE 移动鼠标
MOUSEEVENTF_LEFTDOWN 模拟鼠标左键按下
MOUSEEVENTF_LEFTUP 模拟鼠标左键抬起
MOUSEEVENTF_RIGHTDOWN 模拟鼠标右键按下
MOUSEEVENTF_RIGHTUP 模拟鼠标右键按下
MOUSEEVENTF_MIDDLEDOWN 模拟鼠标中键按下
MOUSEEVENTF_MIDDLEUP 模拟鼠标中键按下
dxLong，根据是否指定了 MOUSEEVENTF_ABSOLUTE 标志，指定水平方向的绝对位置或相对运动
dyLong，根据是否指定了 MOUSEEVENTF_ABSOLUTE 标志，指定垂直方向的绝对位置或相对运动
cButtonsLong，未使用
dwExtraInfoLong，通常未用的一个值。用 GetMessageExtraInfo 函数可取得这个值。可用的值取决于特定的驱动程序
注解
进行相对运动的时候，由 SystemParametersInfo 函数规定的系统鼠标轨迹速度会应用于鼠标运行的速度

Top
OemKeyScan
OemKeyScan

VB 声明
Declare Function OemKeyScan Lib "user32" Alias "OemKeyScan" (ByVal wOemChar As Long) As Long

说明
判断 OEM 字符集中的一个 ASCII 字符的扫描码和 Shift 键状态
返回值
Long，低字包含了扫描码。高字包含了下述标志：位 0 标志着 Shift 已经按下；位 1 标志着 Ctrl 键按下；位 2 标志着 Alt 键按下。如两个字都为-1，表明字符未在 OEM 字符集中得到定义
参数表
参数类型及说明
wOemCharLong，欲转换的字符的 ASCII 值
注解
这个函数只能转换那些单击键就能生成的字符。倘若字符要通过“Alt+3 数位”（比如 Alt+255）才能出现，则不能用这个函数转换
原文：This function only translates keys that can be typed with a single keystroke. Keys that are entered using the ALT + 3 digit entry cannot be converted with this function.

Top
OemToChar
OemToChar, OemToCharBuff

VB 声明
Declare Function OemToChar& Lib "user32" Alias "OemToCharA" (ByVal lpszSrc As String, ByVal lpszDst As String)
Declare Function OemToCharBuff& Lib "user32" Alias "OemToCharBuffA" (ByVal lpszSrc As String, ByVal lpszDst As String, ByVal cchDstLength As Long)
说明
将 OEM 字符集的一个字串转换到 ANSI 字符集。OemToCharBuff 允许我们指定字串中欲转换的字符数量
返回值
Long，肯定是 TRUE
参数表
参数类型及说明
lpszSrcString，欲转换的字串
lpszDstString，用于容纳 ANSI 转换结果的一个字串。注意先将字串初始化成合适的长度。可将相同的字串传递给两个参数，以便在同一个字串中进行转换
cchDstLengthLong，字串中要转换的字符数量
注解
如使用一个 Win32 类型库访问宽字符函数 OemToCharW，那么 lpszSrc 是个 Unicode 字串，而且 lpszDst 参数绝对不能与 lpszSrc 相同

Top
SetCaretBlinkTime
SetCaretBlinkTime

VB 声明

Declare Function SetCaretBlinkTime Lib "user32" Alias "SetCaretBlinkTime" (ByVal wMSeconds As Long) As Long
说明
指定插入符（光标）的闪烁频率
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
wMSecondsLong，插入符新的闪烁间隔时间，以毫秒为单位
注解
插入符或插入标志是一种共享资源，所以闪烁时间设定会对所有应用程序的插入符产生影响。可用 GetCaretBlinkTime 函数获得最初的闪烁时间设置。以后在适当的时候，可用这个设置恢复最开始的值。参考 CreateCaret 函数的注解

Top
SetCaretPos
SetCaretPos

VB 声明
Declare Function SetCaretPos Lib "user32" Alias "SetCaretPos" (ByVal x As Long, ByVal y As Long) As Long
说明
指定插入符的位置
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
x,yLong，插入符在客户区坐标系统中的 X，Y 位置
注解
插入符是一种共享资源。在更改插入符的位置之前，程序员应确定插入符位于由当前任务拥有的一个窗口内。参考 CreateCaret 函数的注解

Top
SetComputerName
SetComputerName

VB 声明
Declare Function SetComputerName Lib "kernel32" Alias "SetComputerNameA" (ByVal lpComputerName As String) As Long
说明
设置新的计算机名
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表

参数类型及说明
lpComputerNameString，新的计算机名称。最多可有 MAX_COMPUTERNAME_LENGTH 个字符
注解
windows95 会将任何非法字符自动转换到标准的字符集里。windows nt 则会报告出错

Top
SetCursor
SetCursor

VB 声明
Declare Function SetCursor Lib "user32" Alias "SetCursor" (ByVal hCursor As Long) As Long
说明
将指定的鼠标指针设为当前指针
返回值
Long，前一个指针的值
参数表
参数类型及说明
hCursorLong，要设为当前指针的一个指针的句柄。如设为零，表示不显示任何指针
注解
在 vb 里这个函数不能很好的工作，因为 vb 习惯在不同的时间将指针变回原来的样子

Top
SetCursorPos
SetCursorPos

VB 声明
Declare Function SetCursorPos Lib "user32" Alias "SetCursorPos" (ByVal x As Long, ByVal y As Long) As Long
说明
设置指针的位置
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
x,y 鼠标指针在屏幕像素坐标系统中的 X，Y 位置

Top
SetDoubleClickTime
SetDoubleClickTime

VB 声明
Declare Function SetDoubleClickTime Lib "user32" Alias "SetDoubleClickTime" (ByVal wCount As Long) As Long

说明
设置连续两次鼠标单击之间能使系统认为是双击事件的间隔时间
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
wCountLong，新的 DoubleClick 间隔时间，以毫秒为单位
注解
双击时间设定的变化会影响整个系统

Top
SetEnvironmentVariable
SetEnvironmentVariable

VB 声明
Declare Function SetEnvironmentVariable Lib "kernel32" Alias "SetEnvironmentVariableA" (ByVal lpName As String, ByVal lpValue As String) As Long
说明
将一个环境变量设为指定的值
返回值
Long，TRUE（非零）表示成功，否则返回零。会设置 GetLastError
参数表
参数类型及说明
lpNameString，欲设置的环境变量的名称。如具有这个名称的一个环境变量尚不存在，则函数会自动创建它
lpValueString，欲为变量设置的新值。如为 NULL，表示清除一个现成值（用 vbNullString 常数向这个函数传递一个 NULL）

Top
SetKeyboardState
SetKeyboardState

VB 声明
Declare Function SetKeyboardState Lib "user32" Alias "SetKeyboardState" (lppbKeyState As Byte) As Long
说明
设置每个虚拟键当前在键盘上的状态
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lppbKeyStateByte，固定为 256 字符长度的一个字串。windows 内部键盘状态表中的每个字符都会根据这张表中对应的虚拟键设置。每个键的状态与 GetKeyboardState 函数的结果是相同的

注解
可用这个函数设置 CapsLock， NumLock 和 ScrollLock 键的状态

Top
SetLocaleInfo
SetLocaleInfo

VB 声明
Declare Function SetLocaleInfo Lib "kernel32" Alias "SetLocaleInfoA" (ByVal Locale As Long, ByVal LCType As Long, ByVal lpLCData As String) As Long
说明
改变用户“地方”设置信息
返回值
Long，TRUE（非零）表示成功，否则返回零。会将 GetLastError 设置为下述值之一：ERROR_INVALID_ACCESS，ERROR_INVALID_FLAGS，ERROR_INVALID_PARAMETER
参数表
参数类型及说明
LocaleLong，要为其改变信息的地方 ID
LCTypeLong，欲改变的信息类型。参考 api32.txt，检视那些带 LOCALE_ 前缀的常数
lpLCDataString，这个地方信息项目的新设置
注解
这个函数不会改变系统地方设置

Top
SetLocalTime
SetLocalTime

VB 声明
Declare Function SetLocalTime Lib "kernel32" Alias "SetLocalTime" (lpSystemTime As SYSTEMTIME) As Long
说明
设置当前地方时间
返回值
Long，TRUE（非零）表示成功，否则返回零。会设置 GetLastError
参数表
参数类型及说明
lpSystemTimeSYSTEMTIME，这个结构指定了新的地方时间。该结构中的 wDayOfWeek 条目会被忽略

Top
SetSysColors
SetSysColors

VB 声明

Declare Function SetSysColors Lib "user32" Alias "SetSysColors" (ByVal nChanges As Long, lpSysColor As Long, lpColorValues As Long) As Long
说明
设置指定窗口显示对象的颜色
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
nChangesLong，欲改变的对象的数量
lpSysColorLong，按引用传递。这是一个整数数组（总共包含 nChanges 个元素）的第一个元素。每个条目都包含了一个常数，指定一个 windows 显示对象。参考 GetSysColor 函数
lpColorValuesLong，按引用传递。这是 RGB 值数组的第一个元素；该数组用于设置 lpSysColor 数组中的对象颜色

Top
SetSystemCursor
SetSystemCursor

VB 声明
Declare Function SetSystemCursor Lib "user32" Alias "SetSystemCursor" (ByVal hcur As Long, ByVal id As Long) As Long
说明
改变任何一个标准系统指针
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hcurLong，新指针
idLong，以 OCR_ 前缀起头的一个常数，用于指定标准系统指针
注解
不要破坏由 hcur 指定的指针——在必要的时候，它会由系统自行清除

Top
SetSystemTime
SetSystemTime

VB 声明
Declare Function SetSystemTime Lib "kernel32" Alias "SetSystemTime" (lpSystemTime As SYSTEMTIME) As Long
说明
设置当前系统时间
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表

参数类型及说明
lpSystemTimeSYSTEMTIME，这个结构指定了新的地方时间。其中的 wDayOfWeek 条目会被忽略

Top
SetSystemTimeAdjustment
SetSystemTimeAdjustment

VB 声明
Declare Function SetSystemTimeAdjustment Lib "kernel32" Alias "SetSystemTimeAdjustment" (ByVal dwTimeAdjustment As Long, ByVal bTimeAdjustmentDisabled As Boolean) As Long
说明
Win32 可使内部系统时钟与一个外部的时钟信号源同步，方法是定时添加一个校准值。这个函数指定的所有时间都以 100ns（0.1ms）为单位递增
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
dwTimeAdjustmentLong，每次时钟中断时添加到内部系统时钟的时间量
bTimeAdjustmentDisabledBoolean，TRUE 用于禁止时间调校

Top
SetThreadLocale
SetThreadLocale

VB 声明
Declare Function SetThreadLocale Lib "kernel32" Alias "SetThreadLocale" (ByVal Locale As Long) As Long
说明
为当前线程设置地方
返回值
Long，TRUE（非零）表示成功，否则返回零。会设置 GetLastError
参数表
参数类型及说明
LocaleLong，这个线程使用的地方 ID
注解
适用平台——Windows NT

Top
SetTimeZoneInformation
SetTimeZoneInformation

VB 声明
Declare Function SetTimeZoneInformation Lib "kernel32" Alias "SetTimeZoneInformation"

(lpTimeZoneInformation As TIME_ZONE_INFORMATION) As Long
说明
设置系统时区信息
返回值
Long，TRUE（非零）表示成功，否则返回零。会设置 GetLastError
参数表
参数类型及说明
lpTimeZoneInformationTIME_ZONE_INFORMATION，要在其中设置当前时区信息的一个结构
注解
参考 GetTimeZoneInformation

Top
ShowCaret
ShowCaret

VB 声明
Declare Function ShowCaret Lib "user32" Alias "ShowCaret" (ByVal hwnd As Long) As Long
说明
在指定的窗口里显示插入符（光标）
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，包含了插入符的窗口的句柄。可以为零；此时，只有在插入符包含于由活动任务拥有的一个窗口时，它才会显示出来
注解
参考 HideCaret 函数

Top
ShowCursor
ShowCursor

VB 声明
Declare Function ShowCursor Lib "user32" Alias "ShowCursor" (ByVal bShow As Long) As Long
说明
控制鼠标指针的可视性
返回值
Long，显示计数（参考注解）
参数表
参数类型及说明
bShowLong，TRUE（非零）显示指针，FALSE 隐藏
注解

windows 维持着一个内部显示计数；倘若 bShow 为 TRUE，那么每调用一次这个函数，计数就会递增 1；反之，如 bShow 为 FALSE，则计数递减 1。只有在这个计数大于或等于 0 的情况下，指针才会显示出来

Top
SwapMouseButton
SwapMouseButton

VB 声明
Declare Function SwapMouseButton Lib "user32" Alias "SwapMouseButton" (ByVal bSwap As Long) As Long
说明
决定是否互换鼠标左右键的功能
返回值
Long，TRUE（非零）表示鼠标按钮的功能在调用这个函数之前已经互换；否则返回零
参数表
参数类型及说明
bSwapLong，倘若为 TRUE（非零），则互换两个鼠标按钮的功能。FALSE 则恢复正常状态
注解
鼠标是一种共享资源，所以这个函数会对系统中的所有应用程序造成影响

Top
SystemParametersInfo
SystemParametersInfo, SystemParametersInfoByval

VB 声明
Declare Function SystemParametersInfo& Lib "user32" Alias "SystemParametersInfoA" (ByVal uAction As Long, ByVal uParam As Long, lpvParam As Any, ByVal fuWinIni As Long)
Declare Function SystemParametersInfoByVal& Lib "user32" Alias "SystemParametersInfoA" (ByVal uAction As Long, ByVal uParam As Long, ByVal lpvParam As Any, ByVal fuWinIni As Long)
说明
允许获取和设置数量众多的 windows 系统参数
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
uActionLong，指定要设置的参数。参考 uAction 常数表
uParamLong，参考 uAction 常数表
lpvParamAny，按引用调用的 Integer、Long 和数据结构。对于 String 数据，请用 SystemParametersInfoByval 函数。具体用法参考 uAction 常数表
fuWinIniLong，取决于不同的参数及操作系统，随同这个函数设置的用户配置参数保存在 win.ini 或注册表里，或同时保存在这两个地方。这个参数规定了在设置系统参数的时候，是否应更新用户设置参数。可以是零（禁止更新），或下述任何一个常数：

SPIF_UPDATEINIFILE 更新 win.ini 和（或）注册表中的用户配置文件
SPIF_SENDWININICHANGE 倘若也设置了 SPIF_UPDATEINIFILE，将一条 WM_WININICHANGE 消息发给所有应用程序。否则没有作用。这调消息告诉应用程序已经改变了用户配置设置
注解
在调用这个函数之前，特别要注意将 lpvParam 参数定义成正确的数据类型

Top
SystemTimeToTzSpecificLocalTime
SystemTimeToTzSpecificLocalTime

VB 声明
Declare Function SystemTimeToTzSpecificLocalTime Lib "kernel32" Alias "SystemTimeToTzSpecificLocalTime" (lpTimeZoneInformation As TIME_ZONE_INFORMATION, lpUniversalTime As SYSTEMTIME, lpLocalTime As SYSTEMTIME) As Long
说明
将系统时间转换成地方时间
返回值
Long，TRUE（非零）表示成功，否则返回零。会设置 GetLastError
参数表
参数类型及说明
lpTimeZoneInformationTIME_ZONE_INFORMATION，包含了时区信息的结构
lpUniversalTimeSYSTEMTIME，包含系统时间的结构
lpLocalTimeSYSTEMTIME，用于装载地方时间的结构
注解
适用平台：Windows NT

Top
ToAscii
ToAscii, ToAsciiEx

VB 声明
Declare Function ToAscii& Lib "user32" (ByVal uVirtKey As Long, ByVal uScanCode As Long, lpbKeyState As Byte, lpwTransKey As Integer, ByVal fuState As Long)
Declare Function ToAsciiEx& Lib "user32" (ByVal uVirtKey As Long, ByVal uScanCode As Long, lpKeyState As Byte, lpwTransKey As Integer, ByVal fuState As Long, ByVal dwhkl As Long)
说明
根据当前的扫描码和键盘信息，将一个虚拟键转换成 ASCII 字符
返回值
Long，负值表明按键是“死”的——不能自己将自己转换成一个字符（重音键[accent keys]就是一个例子）。在给定当前键盘状态的前提下，如按键不能被转换（翻译），则返回 0。如单个字符已载入 lpwTransKey，则返回 1。如 lpwTransKey 里已载入了两个字符（需要把它

分隔到两个字节里），那么返回值是 2。在当前字符集里，倘若单独一个字符不能表达键盘支持的死键或重音按键组合，就可能得到 2 的返回值
参数表
参数类型及说明
uVirtKeyLong，欲转换的虚拟键
uScanCodeLong，键的扫描码。如键处于抬起状态，会设置高位（设为 1）；如按下，则清除高位（设为 0）
lpbKeyStateByte，描述了键盘状态的一个 256 字符数组的第一个条目。参考 GetKeyboardState 函数，了解关于这个数组更多的情况
lpwTransKeyInteger，用于装载转换过后的字符的一个整数变量。可用 chr()函数将这个值转换成一个字串
fuStateLong，如一个菜单处于活动状态，则设为 1
dwhklLong，欲用于转换的一个键盘布局的句柄
注解
NumLock 键的状态会被忽略，因为虚拟键码包括了哪个信息
在微软的 win32 手册里，对 ToAsciiEx 函数的建议是将它的 lpwTransKey 参数设为 Long，而不要设为 Integer。这里的函数声明根据实际的 C 语言头，它将参数定义成一个 16 位的字（既 vb 的整数）

Top
ToUnicode
ToUnicode

VB 声明
Declare Function ToUnicode Lib "user32" Alias "ToUnicode" (ByVal wVirtKey As Long, ByVal wScanCode As Long, lpKeyState As Byte, ByVal pwszBuff As String, ByVal cchBuff As Long, ByVal wFlags As Long) As Long
说明
根据当前的扫描码和键盘信息，将一个虚拟键转换成 Unicode 字符
返回值
Long，值-1 表明按键是“死”的——不能自己将自己转换成一个字符（重音键[accent keys]就是一个例子）。在给定当前键盘状态的前提下，如按键不能被转换（翻译），则返回 0。如单个字符已载入 pwszBuff，则返回 1。如 pwszBuff 里已载入了两个或更多的字符，那么返回值是 2。在当前字符集里，倘若单独一个字符不能表达键盘支持的死键或重音按键组合，就可能得到 2 的返回值
参数表
参数类型及说明
wVirtKeyLong，欲转换的虚拟键
wScanCodeLong，键的扫描码。如键处于抬起状态，会设置高位；如按下，则清除高位
lpKeyStateByte，描述了键盘状态的一个 256 字符数组的第一个条目。参考 GetKeyboardState 函数，了解关于这个数组更多的情况
pwszBuffString，用于装载 Unicode 字符的一个字串缓冲区。注意事先对这个字串进行正确的初始化
cchBuffLong，pwszBuff 字串缓冲区的长度

wFlagsLong，如一个菜单处于活动状态，则设为 1
注解
适用平台：Windows NT

Top
UnloadKeyboardLayout
UnloadKeyboardLayout

VB 声明
Declare Function UnloadKeyboardLayout Lib "user32" Alias "UnloadKeyboardLayout" (ByVal HKL As Long) As Long
说明
卸载指定的键盘布局。参考 LoadKeyboardLayout 函数，了解更多信息
返回值
Long，如执行成功，返回键盘布局的句柄；如出错，则返回零。会设置 GetLastError
参数表
参数类型及说明
HKLLong，欲卸载的键盘布局的句柄
注解
在 windows95 下，这个函数永远都不能卸载默认的系统键盘布局

Top
VkKeyScan
VkKeyScan, VkKeyScanEx

VB 声明
Declare Function VkKeyScan% Lib "user32" Alias "VkKeyScanA" (ByVal cChar As Byte)
Declare Function VkKeyScanEx% Lib "user32" Alias "VkKeyScanExA" (ByVal ch As Byte, ByVal dwhkl As Long)
说明
针对 Windows 字符集中一个 ASCII 字符，判断虚拟键码和 Shift 键的状态
返回值
Long，低字包含了虚拟键码。高字则包含了下述标志：
位 0 指出 Shift 键已经按下；位 1 指出 Ctrl 键已经按下；位 2 指出 Alt 键已经按下
参数表
参数类型及说明
chByte，欲转换的字符的 ASCII 值
dwhklLong，转换时作为依据的键盘布局
注解
数字小键盘的转换会被忽略。这个函数可用于获取发送 WM_KEYDOWN 和 WM_KEYUP 消息时应使用的正确参数

Top
控件与消息函数

控件与消息函数：共六页。第一页，第二页，第三页，第四页，第五页，第六页
AdjustWindowRect 给定一种窗口样式，计算获得目标客户区矩形所需的窗口大小
AnyPopup 判断屏幕上是否存在任何弹出式窗口
ArrangeIconicWindows 排列一个父窗口的最小化子窗口
AttachThreadInput 连接线程输入函数
BeginDeferWindowPos 启动构建一系列新窗口位置的过程
BringWindowToTop 将指定的窗口带至窗口列表顶部
CascadeWindows 以层叠方式排列窗口
ChildWindowFromPoint 返回父窗口中包含了指定点的第一个子窗口的句柄
ClientToScreen 判断窗口内以客户区坐标表示的一个点的屏幕坐标
CloseWindow 最小化指定的窗口
CopyRect 矩形内容复制
DeferWindowPos 该函数为特定的窗口指定一个新窗口位置
DestroyWindow 清除指定的窗口以及它的所有子窗口
DrawAnimatedRects 描绘一系列动态矩形
EnableWindow 指定的窗口里允许或禁止所有鼠标及键盘输入
EndDeferWindowPos 同时更新 DeferWindowPos 调用时指定的所有窗口的位置及状态
AdjustWindowRect
AdjustWindowRect,AdjustWindowRectEx

VB 声明
Declare Function AdjustWindowRect Lib "user32" Alias "AdjustWindowRect" (lpRect As RECT, ByVal dwStyle As Long, ByVal bMenu As Long) As Long
Declare Function AdjustWindowRectEx Lib "user32" Alias "AdjustWindowRectEx" (lpRect As RECT, ByVal dsStyle As Long, ByVal bMenu As Long, ByVal dwEsStyle As Long) As Long
说明
在给定一种窗口样式的前提下，计算获得目标客户区矩形所需的窗口大小
返回值
Long，如执行成功，则返回非零值；如失败，返回零值。会设置 GetLastError
参数表
参数类型及说明
lpRectRECT，最初包含要求的客户区。由函数设为目标窗口矩形大小
dwStyleLong，窗口样式
bMenuLong，如窗口有菜单，则设为 TRUE（非零）
dwEsStyleLong，扩展窗口样式（只适用于 AdjustWindowRectEx）
注解
在调用本函数前，先用 GetWindowLong 取得一个窗体的样式。如菜单占用两行以上的空间，则函数不能正确计算大小。如程序使用了多行标题，则应使用 GetSystemMetrics

Top
AnyPopup
AnyPopup

VB 声明

Declare Function AnyPopup Lib "user32" Alias "AnyPopup" () As Long
说明
判断屏幕上是否存在任何弹出式窗口
返回值
Long，如存在弹出式菜单，则返回 TRUE（非零）
注解
对该函数来说，弹出式菜单包含所有可见的包容顶级窗口，无论弹出式还是重叠窗口

Top
ArrangeIconicWindows
ArrangeIconicWindows

VB 声明
Declare Function ArrangeIconicWindows Lib "user32" Alias "ArrangeIconicWindows" (ByVal hwnd As Long) As Long
说明
排列一个父窗口的最小化子窗口（在 vb 里使用：用于在桌面排列图标，用 GetDesktopWindow 函数获得桌面窗口的一个句柄）
返回值
Long，图标行的高度；如失败，则返回零。会设置 GetLastError
参数表
参数类型及说明
hwndLong，父窗口的句柄
注解
也可将该函数用于包含了图标化子窗口的的定制控件

Top
AttachThreadInput
AttachThreadInput

VB 声明
Declare Function AttachThreadInput Lib "user32" Alias "AttachThreadInput" (ByVal idAttach As Long, ByVal idAttachTo As Long, ByVal fAttach As Long) As Long
说明
通常，系统内的每个线程都有自己的输入队列。本函数（既“连接线程输入函数”）允许线程和进程共享输入队列。连接了线程后，输入焦点、窗口激活、鼠标捕获、键盘状态以及输入队列状态都会进入共享状态
返回值
Long，非零表示成功，零表示失败，会设置会 GetLastError
参数表
参数类型及说明
idAttachLong，欲连接线程的标识符（ID）
idAttachToLong，与 idAttach 线程连接的另一个线程的标识符
fAttachLong，TRUE（非零）连接，FALSE 撤消连接

注解
调用这个函数时，会重设键盘状态

Top
BeginDeferWindowPos
BeginDeferWindowPos

VB 声明
Declare Function BeginDeferWindowPos Lib "user32" Alias "BeginDeferWindowPos" (ByVal nNumWindows As Long) As Long
说明
启动构建一系列新窗口位置的过程（以便同时更新）。该函数会向一个内部结果返回一个句柄，这个结构容纳了与窗口位置有关的信息。随后，该结构会由对 DeferWindowPos 函数的调用填充。准备好更新所有窗口位置以后，对 EndDeferWindowPos 的一个调用可同时更新结构内所有窗口的位置
返回值
Long，内部结构的句柄。零表示失败
参数表
参数类型及说明
nNumWindowsLong，在结构中欲为其分配空间的初始窗口数量。在每次 DeferWindowPos 调用期间结构的大小会根据情况自动调节

Top
BringWindowToTop
BringWindowToTop

VB 声明
Declare Function BringWindowToTop Lib "user32" Alias "BringWindowToTop" (ByVal hwnd As Long) As Long
说明
将指定的窗口带至窗口列表顶部。倘若它部分或全部隐藏于其他窗口下面，则将隐藏的部分完全显示出来。该函数也对弹出式窗口、顶级窗口以及 MDI 子窗口产生作用
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，欲带至顶部的那个窗口的句柄
注解
这个函数也许能随同子窗口使用。函数对一个特定的输入线程来说是“本地的”——换言之，倘若某窗口并非前台应用程序的一部分，那么一旦随同该窗口调用本函数，仍会将窗口带至它自己那个应用程序的窗口列表顶部。但是，不会同时使那个应用成为前台应用程序。这意味着在调用了本函数后，窗口仍会保持隐藏状态

Top

CascadeWindows
CascadeWindows,CascadeWindowsBynum

VB 声明
Declare Function CascadeWindows% Lib "user32" (ByVal hwndParent As Long, ByVal wHow As Long, lpRect As RECT, ByVal cKids As Long, lpKids As Long)
Declare Function CascadeWindowsBynum% Lib "user32" Alias "CascadeWindows" (ByVal hwndParent As Long, ByVal wHow As Long, ByVal lpRect As Long, ByVal cKids As Long, ByVal lpKids As Long)
说明
以层叠方式排列窗口（在 vb 里使用：位于顶部或被所有的窗口没有问题。原文：No problem for top level windows or owned windows.）
返回值
Integer，排列成功的窗口数量，零表示失败
参数表
参数类型及说明
hwndParentLong，指定一个父窗口；准备对它的子窗口进行排列。用 GetDesktopWindow 函数获得顶级窗口的句柄
wHowLong，MDITILE_SKIPDISABLED——不排列已被禁用的 MDI 子窗口
lpRectRECT，指定一个矩形，矩形区域中的窗口才会层叠处理。可设为 NULL，表示使用整个客户区
cKidsLong，在 lpKids 数组中指定的子窗口数量
lpKidsLong，子窗口列表中准备排列的第一个元素。如传递 NULL（注意将参数定义成 ByVal）。Long，则表示排列所有的子窗口（原文：Long--First element in list of child windows to arrange. Pass NULL (be sure to define parameter as ByVal - Long, to arrange all child windows.）
注解
在正式的 win32 文档里，对这个函数的说明是不正确的。这儿的参数建立在实际的 win32 C 头文件基础上。函数不能对诸如控件的子窗口产生——只对顶级窗口及 MDI 子有用。注意在 MDI 窗体的情况下，指定的父窗口应是 MDIClient 窗口的句柄，不应是 MDI 窗体本身的窗口句柄。可用 api 函数 GetParent 获得正确的句柄

Top
ChildWindowFromPoint
ChildWindowFromPoint,ChildWindowFromPointEx

VB 声明
Declare Function ChildWindowFromPoint Lib "user32" Alias "ChildWindowFromPoint" (ByVal hWnd As Long, ByVal xPoint As Long, ByVal yPoint As Long) As Long
Declare Function ChildWindowFromPointEx Lib "user32" Alias "ChildWindowFromPointEx" (ByVal hWnd As Long, ByVal pt As POINTAPI, ByVal un As Long) As Long
说明
返回父窗口中包含了指定点的第一个子窗口的句柄
返回值

Long，发现包含了指定点的第一个子窗口的句柄。如未发现任何窗口，则返回 hWnd（父窗口的句柄）。如指定点位于父窗口外部，则返回零
参数表
参数类型及说明
hWndLong，父窗口的句柄
xPointLong，点的 X 坐标，以像素为单位
yPointLong，点的 Y 坐标，以像素为单位
ptPOINTAPI，点的坐标，以像素为单位
unLong，（只适用于 ChildWindowFromPointEx）控制对窗口的搜索。参见下表
CWP_ALL 测试所有窗口
CWP_SKIPINVISIBLE 忽略不可见窗口
CWP_SKIPDISABLED 忽略已屏蔽的窗口
CWP_SKIPTRANSPARENT 忽略透明窗口

Top
ClientToScreen
ClientToScreen

VB 声明
Declare Function ClientToScreen Lib "user32" Alias "ClientToScreen" (ByVal hwnd As Long, lpPoint As POINTAPI) As Long
说明
判断窗口内以客户区坐标表示的一个点的屏幕坐标
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hwndLong，判断客户区坐标时那个窗口的句柄
lpPointPOINTAPI，用 hwnd 窗口的客户区坐标表示的点，这个参数会包含屏幕坐标系统中相同的点

Top
CloseWindow
CloseWindow

VB 声明
Declare Function CloseWindow Lib "user32" Alias "CloseWindow" (ByVal hwnd As Long) As Long
说明
最小化指定的窗口。窗口不会从内存中清除
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明

hwndLong，欲最小化的那个窗口的句柄

Top
CopyRect
CopyRect

VB 声明
Declare Function CopyRect Lib "user32" Alias "CopyRect" (lpDestRect As RECT, lpSourceRect As RECT) As Long
说明
将矩形的 lpSourceRect 内容复制给矩形 lpDestRect
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpDestRectRECT，目标矩形结构
lpSourceRectRECT，源矩形

Top
DeferWindowPos
DeferWindowPos

VB 声明
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
说明
该函数为特定的窗口指定一个新窗口位置，并将其输入由 BeginDeferWindowPos 创建的结构，以便在 EndDeferWindowPos 函数执行期间更新
返回值
Long，返回一个新句柄，它指向的结构包含了位置更新信息。这个句柄应在对 DeferWindowPos 的后续调用以及对 EndDeferWindowPos 的结束调用中用到。如出错，则返回零值
参数表
参数类型及说明
hWinPosInfoLong，由 BeginDeferWindowPos 为后续对 DeferWindowPos 的调用返回的句柄
hwndLong，欲定位的窗口
hWndInsertAfterLong，窗口句柄。在窗口列表中，窗口 hwnd 会排列于这个窗口后面。也可用下述值之一：
HWND_BOTTOM 将窗口置于窗口列表底部
HWND_TOP 将窗口置于 Z 序列顶部；Z 序列是窗口针对分级结构中一个给定级别显示的顺序
HWND_TOPMOST 将窗口置于列表顶部，位于任何最顶级窗口的前面（请参考 WS_EX_TOPMOST 样式位）

HWND_NOTOPMOST 将窗口置于列表顶部，位于任何最顶级窗口的后面
xLong，窗口新的 x 坐标。如 hwnd 是个子窗口，那么 x 用父窗口的客户区坐标表示
yLong，窗口新的 y 坐标。如 hwnd 是个子窗口，那么 y 用父窗口的客户区坐标表示
cxLong，指定新窗口宽度
cyLong，指定新窗口高度
wFlagsLong，包含了旗标的一个整数，如下所示：
SWP_DRAWFRAME 围绕窗口画一个框
SWP_HIDEWINDOW 隐藏窗口
SWP_NOACTIVATE 不激活窗口
SWP_NOMOVE 保持当前位置（x 和 y 设定将被忽略）
SWP_NOREDRAW 窗口不自动重画
SWP_NOSIZE 保持当前大小（cx 和 cy 会被忽略）
SWP_NOZORDER 保持在窗口列表的当前位置（hWndInsertAfter 会被忽略）
SWP_SHOWWINDOW 显示窗口
SWP_NOOWNERZORDER 不改变 Z 序列的所有者
SWP_NOSENDCHANGING 窗口不发出 WM_WINDOWPOSCHANGING 消息
注解
请参考对 SetWindowPos 函数的注解。同时参考 BeginDeferWindowPos 和 EndDeferWindowPos

Top
DestroyWindow
DestroyWindow

VB 声明
Declare Function DestroyWindow Lib "user32" Alias "DestroyWindow" (ByVal hwnd As Long) As Long
说明
破坏（即清除）指定的窗口以及它的所有子窗口（在 vb 里使用：用处不大。原文：it is unlikely to be of much use.）
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，欲清除的窗口的句柄

Top
DrawAnimatedRects
DrawAnimatedRects

VB 声明
Declare Function DrawAnimatedRects Lib "user32" Alias "DrawAnimatedRects" (ByVal hwnd As Long, ByVal idAni As Long, lprcFrom As Rect, lprcTo As Rect) As Long
说明
在 lprcFrom 和 lprcTo 之间描绘一系列动态矩形

返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hwndLong，要在其中描绘矩形的窗口。如设为零，表示以桌面为背景
idAniLong，windows 95 要设为 0
lprcFromRect，原始矩形
lprcToRect，目标矩形
注解
我的理解——这个函数用于在窗口缩放时产生动画效果

Top
EnableWindow
EnableWindow

VB 声明
Declare Function EnableWindow Lib "user32" Alias "EnableWindow" (ByVal hwnd As Long, ByVal fEnable As Long) As Long
说明
在指定的窗口里允许或禁止所有鼠标及键盘输入（在 vb 里使用：在 vb 窗体和控件中使用 Enabled 属性）
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，窗口句柄
fEnableLong，非零允许窗口，零禁止

Top
EndDeferWindowPos
EndDeferWindowPos

VB 声明
Declare Function EndDeferWindowPos Lib "user32" Alias "EndDeferWindowPos" (ByVal hWinPosInfo As Long) As Long
说明
同时更新 DeferWindowPos 调用时指定的所有窗口的位置及状态
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hWinPosInfoLong，由对 DeferWindowPos 最近一次调用返回的结构句柄

Top

EnumChildWindows
EnumChildWindows

VB 声明
Declare Function EnumChildWindows Lib "user32" Alias "EnumChildWindows" (ByVal hWndParent As Long, ByVal lpEnumFunc As Long, ByVal lParam As Long) As Long
说明
为指定的父窗口枚举子窗口
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hWndParentLong，欲枚举子窗口的父窗口的句柄
lpEnumFuncLong，为每个子窗口调用的函数的指针。用 AddressOf 运算符获得函数在一个标准模块中的地址
lParamLong，在枚举期间，传递给 dwcbkd32.ocx 定制控件之 EnumWindows 事件的值。这个值的含义是由程序员规定的。（原文：Value that is passed to the EnumWindows event of the dwcbkd32.ocx custom control during enumeration. The meaning of this value is defined by the programmer.）
注解
在 vb4 下要求 dwcbkd32.ocx 定制控件。子窗口下属的子窗口也可由这个函数枚举

Top
EnumThreadWindows
EnumThreadWindows

VB 声明
Declare Function EnumThreadWindows Lib "user32" Alias "EnumThreadWindows" (ByVal dwThreadId As Long, ByVal lpfn As Long, ByVal lParam As Long) As Long
说明
枚举与指定任务相关的窗口
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
dwThreadIdLong，某线程的标识符，它的窗口将被枚举
lpfnLong，指向一个函数的指针，要求为每个子窗口都调用这个函数。用 AddressOf 运算符获得函数在标准模式下的地址
lParamLong，在枚举期间，传递给 dwcbkd32d.ocx 定制控件之 EnumWindows 事件的值。这个值的含义是由程序员规定的
注解
子窗口下属的其他子窗口也可由这个函数枚举

Top

EnumWindows
EnumWindows

VB 声明
Declare Function EnumWindows& Lib "user32" (ByVal lpEnumFunc As Long, ByVal lParam As Long)
说明
枚举窗口列表中的所有父窗口（顶级和被所有窗口）
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
lpEnumFuncLong，指向为每个子窗口都调用的一个函数的指针。用 AddressOf 运算符获得函数在标准模式下的地址
lParamLong，在枚举期间，传递给 dwcbkd32.ocx 定制控件之 EnumWindows 事件的值。这个值的含义是由程序员规定的
注解
我的理解——在随 vb5 同时提供的 api32.txt 文件中，找不到这个函数

Top
EqualRect
EqualRect

VB 声明
Declare Function EqualRect Lib "user32" Alias "EqualRect" (lpRect1 As RECT, lpRect2 As RECT) As Long
说明
判断两个矩形结构是否相同
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpRect1RECT，要比较的矩形
lpRect2RECT，要比较的矩形

Top
FindWindow
FindWindow

VB 声明
Declare Function FindWindow Lib "user32" Alias "FindWindowA" (ByVal lpClassName As String, ByVal lpWindowName As String) As Long
说明
寻找窗口列表中第一个符合指定条件的顶级窗口（在 vb 里使用：FindWindow 最常见的一个

用途是获得 ThunderRTMain 类的隐藏窗口的句柄；该类是所有运行中 vb 执行程序的一部分。获得句柄后，可用 api 函数 GetWindowText 取得这个窗口的名称；该名也是应用程序的标题）
返回值
Long，找到窗口的句柄。如未找到相符窗口，则返回零。会设置 GetLastError
参数表
参数类型及说明
lpClassNameString，指向包含了窗口类名的空中止（C 语言）字串的指针；或设为零，表示接收任何类
lpWindowNameString，指向包含了窗口文本（或标签）的空中止（C 语言）字串的指针；或设为零，表示接收任何窗口标题
注解
很少要求同时按类与窗口名搜索。为向自己不准备参数传递一个零，最简便的办法是传递 vbNullString 常数
示例

```
Dim hw&, cnt&
Dim rttitle As String * 256
hw& = FindWindow("ThunderRT5Main", vbNullString) ' ThunderRTMain under VB4
cnt = GetWindowText(hw&, rttitle, 255)
MsgBox Left$(rttitle, cnt), 0, "RTMain title"
```

Top
FindWindowEx
FindWindowEx

VB 声明

```
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
```

说明
在窗口列表中寻找与指定条件相符的第一个子窗口
返回值
Long，找到的窗口的句柄。如未找到相符窗口，则返回零。会设置 GetLastError
参数表
参数类型及说明
hWnd1Long，在其中查找子的父窗口。如设为零，表示使用桌面窗口（通常说的顶级窗口都被认为是桌面的子窗口，所以也会对它们进行查找）
hWnd2Long，从这个窗口后开始查找。这样便可利用对 FindWindowEx 的多次调用找到符合条件的所有子窗口。如设为零，表示从第一个子窗口开始搜索
lpsz1String，欲搜索的类名。零表示忽略
lpsz2String，欲搜索的类名。零表示忽略

Top
FlashWindow
FlashWindow

VB 声明
Declare Function FlashWindow Lib "user32" Alias "FlashWindow" (ByVal hwnd As Long, ByVal bInvert As Long) As Long
说明
闪烁显示指定窗口。这意味着窗口的标题和说明文字会发生变化，似乎从活动切换到非活动状态、或反向切换。通常对不活动的窗口应用这个函数，引起用户的注意
返回值
Long，如窗口在调用前处于活动状态，则返回 TRUE（非零）
参数表
参数类型及说明
hwndLong，要闪烁显示的窗口的句柄
bInvertLong，TRUE（非零）表示切换窗口标题；FALSE 返回最初状态
注解
该函数通常与一个计数器组合使用，生成连续的闪烁效果。在 windows nt 及 windows for workgroup 中，bInvert 参数会被忽略。但在 windows 95 中不会忽略

Top
GetActiveWindow
GetActiveWindow

VB 声明
Declare Function GetActiveWindow Lib "user32" Alias "GetActiveWindow" () As Long
说明
获得活动窗口的句柄
返回值
Long，活动窗口的句柄，如没有窗口处于活动状态，则返回零值

Top
GetCapture
GetCapture

VB 声明
Declare Function GetCapture Lib "user32" Alias "GetCapture" () As Long
说明
获得一个窗口的句柄，这个窗口位于当前输入线程，且拥有鼠标捕获（鼠标活动由它接收）
返回值
Long，拥有捕获的窗口的句柄。倘若当前线程中没有窗口拥有捕获，则返回零值

Top
GetClassInfo
GetClassInfo,GetClassInfoEx

VB 声明
Declare Function GetClassInfo& Lib "user32" Alias "GetClassInfoA" (ByVal hInstance As Long,

ByVal lpClassName As String, lpWndClass As WNDCLASS)
Declare Function GetClassInfoEx& Lib "user32" Alias "GetClassInfoExA" (ByVal hInstance As Long, ByVal lpClassName As String, lpWndClassEx As WNDCLASSEX)
说明
取得 WNDCLASS 结构（或 WNDCLASSEX 结构）的一个副本，结构中包含了与指定类有关的信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hInstanceLong，指向拥有类的那个实例的一个句柄。如设为 NULL，则获得与标准 windows 类有关的信息
lpClassNameString，欲查找的类名。也可能是个 Long 值，其中的低字是包含类名的一个全局（共用）原子
lpWndClassWNDCLASS，（GetClassInfo）用于包含结果信息的结构
lpWndClassExWNDCLASSEX，（GetClassInfoEx）用于包含结果信息的结构
注解
如为这个函数使用了 WNDCLASSEX 参数，请务必设置其中的 cbSize 字段

Top
GetClassLong
GetClassLong

VB 声明
Declare Function GetClassLong Lib "user32" Alias "GetClassLongA" (ByVal hwnd As Long, ByVal nIndex As Long) As Long
说明
取得窗口类的一个 Long 变量条目
返回值
Long，由 nIndex 决定。零表示出错。会设置 GetLastError
参数表
参数类型及说明
hwnd 要为其获得类信息的窗口的句柄
nIndex 欲取得的信息，可能是下述任何一个常数：（正数表示一个字节偏移，用于取得在额外字节中为这个类分配的类信息）
GCL_CBCLSEXTRA 这个类结构中分配的额外字节数
GCL_CBWNDEXTRA 窗口结构里为这个类中每个窗口分配的额外字节数
GCL_HBRBACKGROUND 描绘这个类每个窗口的背景时，使用的默认刷子的句柄
GCL_HCURSOR 指向这个类窗口默认光标的句柄
GCL_HICON 这个类中窗口默认图标的句柄
GCL_HICONSM 这个类的小图标
GCL_HMODULE 这个类的模块的句柄
GCL_MENUNAME 为类菜单取得名称或资源 ID
GCL_STYLE 这个类的样式

GCL_WNDPROC 取得类窗口函数（该类窗口的默认窗口函数）的地址

Top
GetClassName
GetClassName

VB 声明
Declare Function GetClassName Lib "user32" Alias "GetClassNameA" (ByVal hwnd As Long, ByVal lpClassName As String, ByVal nMaxCount As Long) As Long
说明
为指定的窗口取得类名
返回值
Long，以字节数表示的类名长度；排除最后的空中止字符。零表示出错。会设置 GetLastError
参数表
参数类型及说明
hwndLong，欲获得类名的那个窗口的句柄
lpClassNameString，随同类名载入的缓冲区。预先至少必须分配 nMaxCount+1 个字符
nMaxCountLong，由 lpClassName 提供的缓冲区长度

Top
GetClassWord
GetClassWord

VB 声明
Declare Function GetClassWord Lib "user32" Alias "GetClassWord" (ByVal hwnd As Long, ByVal nIndex As Long) As Long
说明
为窗口类取得一个整数变量
返回值
Long，由 nIndex 决定。零表示出错。会设置 GetLastError
参数表
参数类型及说明
hwndLong，欲获得类信息的那个窗口的句柄
nIndexLong，类信息的正偏移量；这些信息是该类的额外字节中分配的
注解
我的迷惑：在某参考书上是这样声明的：Declare Function GetClassWord% Lib "user32" (ByVal hwnd As Long, ByVal nIndex As Long)，即函数的返回值为 Integer，而从 vb 自带的 api 查看器中得到的声明表明返回值是个 Long 类型

Top
GetClientRect
GetClientRect

VB 声明

Declare Function GetClientRect Lib "user32" Alias "GetClientRect" (ByVal hwnd As Long, lpRect As RECT) As Long
说明
返回指定窗口客户区矩形的大小
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，欲计算大小的目标窗口
lpRectRECT，指定一个矩形，用客户区域的大小载入（以像素为单位）
注解
lpRect 的左侧及顶部区域肯定会被这个函数设为零

Top
GetDesktopWindow
GetDesktopWindow

VB 声明
Declare Function GetDesktopWindow Lib "user32" Alias "GetDesktopWindow" () As Long
说明
获得代表整个屏幕的一个窗口（桌面窗口）句柄
返回值
Long，桌面窗口的句柄
注解
所有桌面图标都在这个窗口里拒绝。它也用于各类屏幕保护程序

Top
GetFocus
GetFocus

VB 声明
Declare Function GetFocus Lib "user32" Alias "GetFocus" () As Long
说明
获得拥有输入焦点的窗口的句柄
返回值
Long，拥有焦点的那个窗口的句柄。如没有窗口拥有输入焦点，则返回零

Top
GetForegroundWindow
GetForegroundWindow

VB 声明
Declare Function GetForegroundWindow Lib "user32" Alias "GetForegroundWindow" () As Long
说明

获得前台窗口的句柄。这里的“前台窗口”是指前台应用程序的活动窗口
返回值
Long，前台窗口的句柄
注解
windows nt 支持多个桌面，它们相互间是独立的。每个桌面都有自己的前台窗口

Top
GetLastActivePopup
GetLastActivePopup

VB 声明
Declare Function GetLastActivePopup Lib "user32" Alias "GetLastActivePopup" (ByVal hwndOwnder As Long) As Long
说明
获得在一个给定父窗口中最近激活过的弹出式窗口的句柄（在 vb 里使用：vb 应用程序不用弹出式窗口，所以这个函数并非特别有用）
返回值
Long，指向最近用过的弹出式窗口的句柄。如未发现弹出窗口。则返回 hwndOwnder
参数表
参数类型及说明
hwndOwnderLong，父窗口

Top
GetParent
GetParent

VB 声明
Declare Function GetParent Lib "user32" Alias "GetParent" (ByVal hwnd As Long) As Long
说明
判断指定窗口的父窗口
返回值
Long，父窗口的句柄。如窗口没有父，或遇到错误，则返回零。会设置 GetLastError
参数表
参数类型及说明
hwndLong，欲测试的窗口的句柄

Top
GetTopWindow
GetTopWindow

VB 声明
Declare Function GetTopWindow Lib "user32" Alias "GetTopWindow" (ByVal hwnd As Long) As Long
说明

搜索内部窗口列表，寻找隶属于指定窗口的头一个窗口的句柄
返回值
Long，位于指定窗口内部的顶级子窗口的句柄
参数表
参数类型及说明
ByValLong，想在其中搜寻顶级子的窗口。零表示寻找位于桌面的顶级窗口
注解
Z 序列中的顶级窗口也是内部窗口列表的第一个窗口

Top
GetUpdateRect
GetUpdateRect

VB 声明
Declare Function GetUpdateRect Lib "user32" Alias "GetUpdateRect" (ByVal hwnd As Long, lpRect As RECT, ByVal bErase As Long) As Long
说明
获得一个矩形，它描叙了指定窗口中需要更新的那一部分（在 vb 里使用：到一个 vb 窗体或控件里发生 paint 事件的时候，更新区域已被清除了。所以这个函数对于 vb 来说是没有意义的。然而，可用一个子类模块拦截一个窗体或控件的 WM_PAINT 消息，在 vb 自行清除之前了解更新区域在哪里）
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hwndLong，欲在其中调查更新区域的那个窗口
lpRectRECT，随同更新坐标载入的矩形
bEraseLong，设置 TRUE（非零），清除更新区域
注解
如窗口类样式拥有 CS_OWNDC 集，且窗口映射模式不是 MM_TEXT，那么更新矩形会用逻辑坐标表示

Top
GetWindow
GetWindow

VB 声明
Declare Function GetWindow Lib "user32" Alias "GetWindow" (ByVal hwnd As Long, ByVal wCmd As Long) As Long
说明
获得一个窗口的句柄，该窗口与某源窗口有特定的关系
返回值
Long，由 wCmd 决定的一个窗口的句柄。如没有找到相符窗口，或者遇到错误，则返回零值。会设置 GetLastError

参数表
参数类型及说明
hwndLong，源窗口
wCmdLong，指定结果窗口与源窗口的关系，它们建立在下述常数基础上：
GW_CHILD 寻找源窗口的第一个子窗口
GW_HWNDFIRST 为一个源子窗口寻找第一个兄弟（同级）窗口，或寻找第一个顶级窗口
GW_HWNDLAST 为一个源子窗口寻找最后一个兄弟（同级）窗口，或寻找最后一个顶级窗口
GW_HWNDNEXT 为源窗口寻找下一个兄弟窗口
GW_HWNDPREV 为源窗口寻找前一个兄弟窗口
GW_OWNER 寻找窗口的所有者
注解
兄弟或同级是指在整个分级结构中位于同一级别的窗口。如某个窗口有五个子窗口，那五个窗口就是兄弟窗口。尽管 GetWindow 可用于枚举窗口，但倘若要在枚举过程中重新定位、创建和清除窗口，那么 EnumWindows 和 EnumChildWindows 更为可靠

Top
GetWindowContextHelpId
GetWindowContextHelpId

VB 声明
Declare Function GetWindowContextHelpId Lib "user32" Alias "GetWindowContextHelpId" (ByVal hWnd As Long) As Long
说明
取得与窗口关联在一起的帮助场景 ID
返回值
Long，帮助场景 ID（如果有的话），否则返回零
参数表
参数类型及说明
hWndLong，欲获取帮助场景的那个窗口的句柄
注解
在 vb 环境中，这个函数不能取得一个 vb 窗体或控件的帮助场景的 ID 集

Top
GetWindowLong
GetWindowLong

VB 声明
Declare Function GetWindowLong Lib "user32" Alias "GetWindowLongA" (ByVal hwnd As Long, ByVal nIndex As Long) As Long
说明
从指定窗口的结构中取得信息
返回值
Long，由 nIndex 决定。零表示出错。会设置 GetLastError

参数表
参数类型及说明
hwndLong，欲为其获取信息的窗口的句柄
nIndexLong，欲取回的信息，可以是下述任何一个常数：
GWL_EXSTYLE 扩展窗口样式
GWL_STYLE 窗口样式
GWL_WNDPROC 该窗口的窗口函数的地址
GWL_HINSTANCE 拥有窗口的实例的句柄
GWL_HWNDPARENT 该窗口之父的句柄。不要用 SetWindowWord 来改变这个值
GWL_ID 对话框中一个子窗口的标识符
GWL_USERDATA 含义由应用程序规定
DWL_DLGPROC 这个窗口的对话框函数地址
DWL_MSGRESULT 在对话框函数中处理的一条消息返回的值
DWL_USER 含义由应用程序规定

Top
GetWindowPlacement
GetWindowPlacement

VB 声明
Declare Function GetWindowPlacement Lib "user32" Alias "GetWindowPlacement" (ByVal hwnd As Long, lpwndpl As WINDOWPLACEMENT) As Long
说明
获得指定窗口的状态及位置信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，欲获取信息的那个窗口的句柄
lpwndplWINDOWPLACEMENT，包含的窗口位置和状态信息的结构
注解
在调用这个函数之前，请务必设置 WINDOWPLACEMENT 结构的长度字段

Top
GetWindowRect
GetWindowRect

VB 声明
Declare Function GetWindowRect Lib "user32" Alias "GetWindowRect" (ByVal hwnd As Long, lpRect As RECT) As Long
说明
获得整个窗口的范围矩形，窗口的边框、标题栏、滚动条及菜单等都在这个矩形内
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError

参数表
参数类型及说明
hwndLong，想获得范围矩形的那个窗口的句柄
lpRectRECT，屏幕坐标中随同窗口装载的矩形
注解
如将它与通过 GetDesktopWindow 获得的句柄联合使用，可获得对整个可视显示区域（桌面）进行说明的矩形

Top
GetWindowText
GetWindowText

VB 声明
Declare Function GetWindowText Lib "user32" Alias "GetWindowTextA" (ByVal hwnd As Long, ByVal lpString As String, ByVal cch As Long) As Long
说明
取得一个窗体的标题（caption）文字，或者一个控件的内容（在 vb 里使用：使用 vb 窗体或控件的 caption 或 text 属性）
返回值
Long，复制到 lpString 的字串长度；不包括空中止字符。会设置 GetLastError
参数表
参数类型及说明
hwndLong，欲获取文字的那个窗口的句柄
lpStringString，预定义的一个缓冲区，至少有 cch+1 个字符大小；随同窗口文字载入
cchLong，lpString 缓冲区的长度
注解
不能用它从另一个应用程序的编辑控件中获取文字

Top
GetWindowTextLength
GetWindowTextLength

VB 声明
Declare Function GetWindowTextLength Lib "user32" Alias "GetWindowTextLengthA" (ByVal hwnd As Long) As Long
说明
调查窗口标题文字或控件内容的长短（在 vb 里使用：直接使用 vb 窗体或控件的 caption 或 text 属性）
返回值
Long，字串长度，不包括空中止字符
参数表
参数类型及说明
hwndLong，想调查文字长度的窗口的句柄

Top
GetWindowWord
GetWindowWord

VB 声明
Declare Function GetWindowWord Lib "user32" Alias "GetWindowWord" (ByVal hwnd As Long, ByVal nIndex As Long) As Integer
说明
获得指定窗口结构的信息
返回值
Long，由 nIndex 决定
参数表
参数类型及说明
hwndLong，想获取信息的那个窗口的句柄
nIndexLong，正偏移值，指出在窗口额外字节分配的空间中取得的信息

Top
InflateRect
InflateRect

VB 声明
Declare Function InflateRect Lib "user32" Alias "InflateRect" (lpRect As RECT, ByVal x As Long, ByVal y As Long) As Long
说明
这个函数用于增大或减小一个矩形的大小。x 加在右侧区域，并从左侧区域减去；如 x 为正，则能增大矩形的宽度；如 x 为负，则能减小它。y 对顶部与底部区域产生的影响是是类似的
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpRectRECT，欲修改的矩形
xLong，用这个数字修改宽度
yLong，用这个数字修改高度
注解
注意宽度和高度的实际改变量是 x 及 y 参数值的两倍

Top
IntersectRect
IntersectRect

VB 声明
Declare Function IntersectRect Lib "user32" Alias "IntersectRect" (lpDestRect As RECT, lpSrc1Rect As RECT, lpSrc2Rect As RECT) As Long
说明

这个函数在 lpDestRect 里载入一个矩形，它是 lpSrc1Rect 与 lpSrc2Rect 两个矩形的交集。如两个源矩形根本未发生重叠，则 lpDestRect 会被设为一个空矩形
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpDestRectRECT，目标矩形，用于包含 lpSrc1Rect 与 lpSrc2Rect 两个矩形的重合部分
lpSrc1RectRECT，第一个源矩形
lpSrc2RectRECT，第二个源矩形

Top
InvalidateRect
InvalidateRect,InvalidateRectBynum

VB 声明
Declare Function InvalidateRect& Lib "user32" (ByVal hwnd As Long, lpRect As RECT, ByVal bErase As Long)
Declare Function InvalidateRectBynum& Lib "user32" Alias "InvalidateRect" (ByVal hwnd As Long, ByVal lpRect As Long, ByVal bErase As Long)
说明
这个函数屏蔽一个窗口客户区的全部或部分区域。这会导致窗口在事件期间部分重画
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hwndLong，待屏蔽窗口的句柄
lpRectRECT，用于描述待屏蔽矩形部分的一个矩形结构。可用 InvalidateRectBynum 函数，同时将 lpRect 设为零（Long 数据类型），从而屏蔽（或禁用）整个窗口
bEraseLong，TRUE（非零）导致指定的区域在重画前先删除
注解
一旦系统有些更新屏幕的闲置时间可用，windows 就会重画窗口

Top
IsChild
IsChild

VB 声明
Declare Function IsChild Lib "user32" Alias "IsChild" (ByVal hWndParent As Long, ByVal hwnd As Long) As Long
说明
判断一个窗口是否为另一窗口的子或隶属窗口
返回值
Long，非零表示成功，零表示失败
参数表

参数类型及说明
hWndParentLong，父窗口的句柄
hwndLong，欲测试的窗口的句柄

Top
IsIconic
IsIconic

VB 声明
Declare Function IsIconic Lib "user32" Alias "IsIconic" (ByVal hwnd As Long) As Long
说明
判断窗口是否已最小化
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hwndLong，待检测窗口的句柄

Top
IsRectEmpty
IsRectEmpty

VB 声明
Declare Function IsRectEmpty Lib "user32" Alias "IsRectEmpty" (lpRect As RECT) As Long
说明
判断一个矩形是否为空
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpRectRECT，要检查的矩形

Top
IsWindow
IsWindow

VB 声明
Declare Function IsWindow Lib "user32" Alias "IsWindow" (ByVal hwnd As Long) As Long
说明
判断一个窗口句柄是否有效
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明

hwndLong，待检查窗口的句柄
注解
如在一个程序变量里容纳了窗口句柄，为了解它是否仍然有效，就可考虑使用这个函数

Top
IsWindowEnabled
IsWindowEnabled

VB 声明
Declare Function IsWindowEnabled Lib "user32" Alias "IsWindowEnabled" (ByVal hwnd As Long) As Long
说明
判断窗口是否处于活动状态（在 vb 里使用：针对 vb 窗体和控件，请用 enabled 属性）
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hwndLong，待检测窗口的句柄

Top
IsWindowUnicode
IsWindowUnicode

VB 声明
Declare Function IsWindowUnicode Lib "user32" Alias "IsWindowUnicode" (ByVal hwnd As Long) As Long
说明
判断一个窗口是否为 Unicode 窗口。这意味着窗口为所有基于文本的消息都接收 Unicode 文字
返回值
Long，如是个 Unicode 窗口则返回非零值，如是个 ANSI 窗口则返回零
参数表
参数类型及说明
hwndLong，窗口句柄
注解
在 vb 里无须担心这方面的问题，windows 会自动转换引用了文本的消息，以满足窗口接收消息的需要

Top
IsWindowVisible
IsWindowVisible

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

As Long
说明
判断窗口是否可见
返回值
Long，如窗口可见则返回 TRUE（非零）
参数表
参数类型及说明
hwndLong，要测试的那个窗口的句柄
注解
窗口可能处于可见状态，同时仍被位于它顶部的其他可见窗口（排在内部窗口列表的前面）隐藏起来

Top
IsZoomed
IsZoomed

VB 声明
Declare Function IsZoomed Lib "user32" Alias "IsZoomed" (ByVal hwnd As Long) As Long
说明
判断窗口是否最大化
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hwndLong，欲测试的窗口的句柄

Top
LockWindowUpdate
LockWindowUpdate

VB 声明
Declare Function LockWindowUpdate Lib "user32" Alias "LockWindowUpdate" (ByVal hwndLock As Long) As Long
说明
锁定指定窗口，禁止它更新。同时只能有一个窗口处于锁定状态
返回值
Long，非零表示成功，零表示失败（比如另外已有一个窗口锁定）
参数表
参数类型及说明
hwndLockLong，欲锁定窗口的句柄。如设为零，则对窗口解锁
注解
windows 会跟踪锁定窗口的区域，并会在窗口解锁后重画它们。可用 GetDCEx 获得一个特殊的设备场景，令其与锁定窗口协同工作，从而描绘一个加锁的窗口。这种技术的一个应用场合是创建跟踪矩形（比如用于改变窗口大小的矩形）

Top
MapWindowPoints
MapWindowPoints

VB 声明
Declare Function MapWindowPoints& Lib "user32" (ByVal hwndFrom As Long, ByVal hwndTo As Long, lppt As POINTAPI, ByVal cPoints As Long)
说明
将一个窗口客户区坐标的点转换到另一窗口的客户区坐标系统（在 vb 里使用：无论向函数传递单独一个点，还是传递数组中的第一个 POINTAPI 结构，都要特别谨慎。数组中的条目数量至少等于由 cPoints 参数指定的数量）
返回值
Long，低字代表映射过程中添加给每个点的水平偏移，高字则代表垂直偏移
参数表
参数类型及说明
hwndFromLong，定义源坐标的窗口。用零或桌面窗口句柄指定屏幕坐标
hwndToLong，定义目标坐标的窗口。用零或桌面窗口句柄指定屏幕坐标
lpptPOINTAPI，点结构中待转换的第一个条目。注意 RECT 结构在内存中组织成两个连续的 POINTAPI 结构。这样就可为该函数创建一个别名，并使用 RECT 结构；而不是 POINTAPI 结构。如这样做时，注意将 cPoints 的值加倍
cPointsLong，欲转换的点数
注解
在 vb 自带的 api 查看器中复制的声明为：Declare Function MapWindowPoints Lib "user32" Alias "MapWindowPoints" (ByVal hwndFrom As Long, ByVal hwndTo As Long, lppt As Any, ByVal cPoints As Long) As Long，请注意：lppt As Any

Top
MoveWindow
MoveWindow

VB 声明
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
说明
改变指定窗口的位置和大小。顶级窗口可能受最大或最小尺寸的限制，那些尺寸优先于这里设置的参数
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hwndLong，欲移动窗口的句柄
xLong，窗口新的左侧位置

yLong，窗口新的顶部位置
nWidthLong，窗口的新宽度
nHeightLong，窗口的高宽度
bRepaintLong，如窗口此时应重画，则设为 TRUE（非零）。FALSE（零）则表明应用程序会自己决定是否重画窗口

Top
OffsetRect
OffsetRect

VB 声明
Declare Function OffsetRect Lib "user32" Alias "OffsetRect" (lpRect As RECT, ByVal x As Long, ByVal y As Long) As Long
说明
该函数通过应用一个指定的偏移，从而让矩形移动起来。x 会添加到右侧和左侧区域。y 添加到顶部和底部区域。偏移方向则取决于参数是正数还是负数，以及采用的是什么坐标系统
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpRectRECT，欲移动的矩形
xLong，水平偏移量
yLong，垂直偏移量

Top
OpenIcon
OpenIcon

VB 声明
Declare Function OpenIcon Lib "user32" Alias "OpenIcon" (ByVal hwnd As Long) As Long
说明
恢复一个最小化的程序，并将其激活
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，欲恢复的窗口
注解
针对 vb 窗体，应使用 vb 的 WindowState 属性

Top
PtInRect
PtInRect

VB 声明
Declare Function PtInRect Lib "user32" Alias "PtInRect" (lpRect As RECT, pt As POINTAPI) As Long
说明
这个函数判断指定的点是否位于矩形 lpRect 内部
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpRectRECT，欲检查的矩形
ptPOINTAPI，欲判断的点
注解
如点位于矩形四边之内，或矩形的顶部或左侧边线上，则认为它在矩形内部。如位于矩形的右侧或底部边线，则不认为它在矩形内部

Top
RedrawWindow
RedrawWindow

VB 声明
Declare Function RedrawWindow Lib "user32" Alias "RedrawWindow" (ByVal hwnd As Long, lprcUpdate As RECT, ByVal hrgnUpdate As Long, ByVal fuRedraw As Long) As Long
说明
根据 fuRedraw 旗标的设置，重画全部或部分窗口
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，要重画的窗口的句柄。零表示更新桌面窗口
lprcUpdateRECT，窗口中需要重画的一个矩形区域
hrgnUpdateLong，一个“区”的句柄，这个区描述了要重画的窗口区域。“区”：Region
fuRedrawLong，规定具体重画操作的旗标。下列常数可组合使用，从而进行复杂的重画行动
RDW_ERASE 重画前，先清除重画区域的背景。也必须指定 RDW_INVALIDATE
RDW_FRAME 如非客户区包含在重画区域中，则对非客户区进行更新。也必须指定 RDW_INVALIDATE
RDW_INTERNALPAINT 即使窗口并非无效，也向其投递一条 WM_PAINT 消息
RDW_INVALIDATE 禁用（屏蔽）重画区域
RDW_NOERASE 禁止删除重画区域的背景
RDW_NOFRAME 禁止非客户区域重画（如果它是重画区域的一部分）。也必须指定 RDW_VALIDATE
RDW_NOINTERNALPAINT 禁止内部生成或由这个函数生成的任何待决 WM_PAINT 消息。针对无效区域，仍会生成 WM_PAINT 消息
RDW_VALIDATE 检验重画区域
RDW_ERASENOW 立即删除指定的重画区域

RDW_UPDATENOW 立即更新指定的重画区域
RDW_ALLCHILDREN 重画操作包括子窗口（前提是它们存在于重画区域）
RDW_NOCHILDREN 重画操作排除子窗口（前提是它们存在于重画区域）
注解
如针对桌面窗口应用这个函数，则应用程序必须用 RDW_ERASE 旗标重画桌面

Top
ReleaseCapture
ReleaseCapture

VB 声明
Declare Function ReleaseCapture Lib "user32" Alias "ReleaseCapture" () As Long
说明
为当前的应用程序释放鼠标捕获
返回值
Long，TRUE（非零）表示成功，零表示失败
注解
我的理解：与 SetCapture 函数一起使用，用于判断鼠标离开（mouseleave）事件

Top
ScreenToClient
ScreenToClient

VB 声明
Declare Function ScreenToClient Lib "user32" Alias "ScreenToClient" (ByVal hwnd As Long, lpPoint As POINTAPI) As Long
说明
判断屏幕上一个指定点的客户区坐标
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hwndLong，一个窗口的句柄，该窗口定义了要使用的客户区坐标系统
lpPointPOINTAPI，屏幕坐标系统中包含了屏幕点的结构。这个函数会随同相应的客户区坐标（由 hwnd 决定）载入结构

Top
ScrollWindow
ScrollWindow

VB 声明
Declare Function ScrollWindow Lib "user32" Alias "ScrollWindow" (ByVal hWnd As Long, ByVal XAmount As Long, ByVal YAmount As Long, lpRect As RECT, lpClipRect As RECT) As Long

说明
滚动窗口客户区的全部或一部分
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hWndLong，待滚动窗口的句柄
XAmountLong，水平滚动的距离。正值向右滚动，负值向左滚动
YAmountLong，垂直滚动的距离。正值向下滚动，负值向上滚动
lpRectRECT，用客户区坐标表示的一个矩形，它定义了客户区要滚动的一个部分。如设为 NULL，则滚动整个客户区。在 NULL 的情况下，子窗口和控件的位置也会随同任何无效区域移动。否则，子窗口和无效区域不会一起移动。因此，在滚动之前，如指定了 lpRect，一个明智的做法是先调用 UpdateWindow 函数
lpClipRectRECT，指定剪切区域。只有这个矩形的区域才可能滚动。该矩形优先于 lpRect。可设为 NULL

Top
ScrollWindowEx
ScrollWindowEx

VB 声明
Declare Function ScrollWindowEx Lib "user32" Alias "ScrollWindowEx" (ByVal hwnd As Long, ByVal dx As Long, ByVal dy As Long, lprcScroll As RECT, lprcClip As RECT, ByVal hrgnUpdate As Long, lprcUpdate As RECT, ByVal fuScroll As Long) As Long
说明
根据附加的选项，滚动窗口客户区的全部或部分
返回值
Long，常数值 SIMPLEREGION，COMPLEXREGION，或 NULLREGION，它们描述了无效区域的类型
参数表
参数类型及说明
hwndLong，欲滚动的窗口的句柄
dxLong，水平滚动的距离。正值向右滚动，负值向左滚动
dyLong，垂直滚动的距离。正值向下滚动，负值向上滚动
lprcScrollRECT，用客户区坐标表示的一个矩形，它定义了客户区要滚动的一个部分。如设为零，则滚动整个客户区
lprcClipRECT，指定一个剪切矩形。只有这个矩形的内容才可能滚动。该矩形优先于 lpRect。可能为零，表示不进行剪切处理（原文：Clipping rectangle. Only the area within this rectangle may be scrolled. This rectangle takes priority over lpRect. May be zero, in which case no clipping takes place.）
hrgnUpdateLong，滚动过程中随同无效区域载入的一个“区”。可能是零
lprcUpdateRECT，随同一个矩形载入的矩形结构，该矩形定义了滚动过程中无效的区域。可能是零
fuScrollLong，对滚动进行控制的旗标。可以是下述任何常数的组合

SW_ERASE 清除新无效区域的背景
SW_INVALDATE 使滚动时未覆盖的区域无效
SW_SCROLLCHILDREN 滚动区域内的子窗口进行等量移动。为避免得到无效的结果，在使用这个函数的时候，请确定子窗口或控件要么完全在滚动区域中，要么完全在滚动区域外

Top
SetActiveWindow
SetActiveWindow

VB 声明
Declare Function SetActiveWindow Lib "user32" Alias "SetActiveWindow" (ByVal hwnd As Long) As Long
说明
激活指定的窗口
返回值
Long，前一个活动窗口的句柄
参数表
参数类型及说明
hwndLong，待激活窗口的句柄
注解
在 vb 里使用这个函数要小心，它不会改变输入焦点，所以焦点可能设向一个不活动窗口，最好换用 SetFocusAPI 函数来激活窗口。如指定的窗口不从属于当前输入线程，则没有任何效果

Top
SetCapture
SetCapture

VB 声明
Declare Function SetCapture Lib "user32" Alias "SetCapture" (ByVal hwnd As Long) As Long
说明
将鼠标捕获设置到指定的窗口。在鼠标按钮按下的时候，这个窗口会为当前应用程序或整个系统接收所有鼠标输入
返回值
Long，之前拥有鼠标捕获的窗口的句柄
参数表
参数类型及说明
hwndLong，要接收所有鼠标输入的窗口的句柄
注解
我的理解：与 ReleaseCapture 函数一起使用，用于判断鼠标离开（mouseleave）事件

Top
SetClassLong
SetClassLong

VB 声明
Declare Function SetClassLong Lib "user32" Alias "SetClassLongA" (ByVal hwnd As Long, ByVal nIndex As Long, ByVal dwNewLong As) As Long
说明
为窗口类设置一个 Long 变量条目
返回值
Long，由 nIndex 指定的的类信息的前一个值。零表示出错。会设置 GetLastError
参数表
参数类型及说明
hwndLong，欲为其设置类信息的那个窗口的句柄
nIndexLong，参考 GetClassLong 函数的 nIndex 参数说明
dwNewLongLong，类信息的新值，具体取决于 nIndex
注解
使用这个函数一定要小心。记住，这里的变动会影响指定类的所有窗口，但除非窗口重画，否则那个变动不会显露出来

Top
SetClassWord
SetClassWord

VB 声明
Declare Function SetClassWord Lib "user32" Alias "SetClassWord" (ByVal hwnd As Long, ByVal nIndex As Long, ByVal wNewWord As Long) As Long
说明
为窗口类设置一个条目
返回值
Long，由 nIndex 指定的类信息的前一个值。零表示出错。会设置 GetLastError
参数表
参数类型及说明
hwndLong，欲为其获得类信息的那个窗口的句柄
nIndexLong，信息结构已分配额外空间的正偏移量
wNewWordLong，由 nIndex 指定的类信息的新值
注解
使用这个函数一定要小心。记住，这里的变动会影响指定类的所有窗口，但除非窗口重画，否则那个变动不会显露出来

Top
SetFocusAPI
SetFocusAPI

VB 声明
Declare Function SetFocusAPI& Lib "user32" Alias "SetFocus" (ByVal hwnd As Long)
说明

将输入焦点设到指定的窗口。如有必要，会激活窗口
返回值
Long，前一个拥有焦点的窗口的句柄
参数表
参数类型及说明
hwndLong，准备接收焦点的窗口的句柄
注解
在 vb 里对窗体和控件最好使用 SetFocus 方法。如指定的窗口不属于当前输入线程，则该函数是没有效果的。它用 SetFocusAPI 这个别名避免与 vb 的 SetFocus 方法发生冲突

Top
SetForegroundWindow
SetForegroundWindow

VB 声明
Declare Function SetForegroundWindow Lib "user32" Alias "SetForegroundWindow" (ByVal hwnd As Long) As Long
说明
将窗口设为系统的前台窗口。这个函数可用于改变用户目前正在操作的应用程序
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，带到前台的窗口
注解
不应随便使用它，因为一旦程序突然从后台进入前台，可能会使用户产生迷惑

Top
SetParent
SetParent

VB 声明
Declare Function SetParent Lib "user32" Alias "SetParent" (ByVal hWndChild As Long, ByVal hWndNewParent As Long) As Long
说明
指定一个窗口的新父（在 vb 里使用：利用这个函数，vb 可以多种形式支持子窗口。例如，可将控件从一个容器移至窗体中的另一个。用这个函数在窗体间移动控件是相当冒险的，但却不失为一个有效的办法。如真的这样做，请在关闭任何一个窗体之前，注意用 SetParent 将控件的父设回原来的那个）
返回值
Long，前一个父窗口的句柄
参数表
参数类型及说明
hWndChildLong，子窗口的句柄

hWndNewParentLong，hWndChild 的新父
注解
可用这个函数在运行期将 vb 控件置入容器控件内部（比如将一个按钮设成图象或窗体控件的子窗口），或者将控件从一个容器控件移至另一个。控件移至另一个父后，它的位置将由新父的坐标系统决定。这样一来，有必要重新规定控件的位置，使其能在目标位置显示出来

Top
SetRect
SetRect

VB 声明
Declare Function SetRect Lib "user32" Alias "SetRect" (lpRect As RECT, ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long) As Long
说明
设置指定矩形的内容
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpRectRECT，欲设置的矩形
X1Long，左侧区域（Left）的值
Y1Long，顶部区域（Top）的值
X2Long，右侧区域（Right）的值
Y2Long，底部区域（Bottom）的值

Top
SetRectEmpty
SetRectEmpty

VB 声明
Declare Function SetRectEmpty Lib "user32" Alias "SetRectEmpty" (lpRect As RECT) As Long
说明
将矩形 lpRect 设为一个空矩形（所有字段都为空）
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpRectRECT，欲设为空的矩形

Top
SetWindowContextHelpId
SetWindowContextHelpId

VB 声明

Declare Function SetWindowContextHelpId Lib "user32" Alias "SetWindowContextHelpId" (ByVal hWnd As Long, ByVal dw As Long) As Long
说明
为指定的窗口设置帮助场景（上下文）ID
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hWndLong，窗口句柄
dwLong，新的帮助场景 ID

Top
SetWindowLong
SetWindowLong

VB 声明
Declare Function SetWindowLong Lib "user32" Alias "SetWindowLongA" (ByVal hwnd As Long, ByVal nIndex As Long, ByVal dwNewLong As Long) As Long
说明
在窗口结构中为指定的窗口设置信息
返回值
Long，指定数据的前一个值
参数表
参数类型及说明
hwndLong，欲为其取得信息的窗口的句柄
nIndexLong，请参考 GetWindowLong 函数的 nIndex 参数的说明
dwNewLongLong，由 nIndex 指定的窗口信息的新值

Top
SetWindowPlacement
SetWindowPlacement

VB 声明
Declare Function SetWindowPlacement Lib "user32" Alias "SetWindowPlacement" (ByVal hwnd As Long, lpwndpl As WINDOWPLACEMENT) As Long
说明
设置窗口状态和位置信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，欲设置位置信息的窗口的句柄
lpwndplWINDOWPLACEMENT，这个结构包含了窗口的位置与状态信息

Top
SetWindowPos
SetWindowPos

VB 声明
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
说明
这个函数能为窗口指定一个新位置和状态。它也可改变窗口在内部窗口列表中的位置。该函数与 DeferWindowPos 函数相似，只是它的作用是立即表现出来的（在 vb 里使用：针对 vb 窗体，如它们在 win32 下屏蔽或最小化，则需重设最顶部状态。如有必要，请用一个子类处理模块来重设最顶部状态
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，欲定位的窗口
hWndInsertAfterLong，窗口句柄。在窗口列表中，窗口 hwnd 会置于这个窗口句柄的后面。也可能选用下述值之一：
HWND_BOTTOM 将窗口置于窗口列表底部
HWND_TOP 将窗口置于 Z 序列的顶部；Z 序列代表在分级结构中，窗口针对一个给定级别的窗口显示的顺序
HWND_TOPMOST 将窗口置于列表顶部，并位于任何最顶部窗口的前面
HWND_NOTOPMOST 将窗口置于列表顶部，并位于任何最顶部窗口的后面
xLong，窗口新的 x 坐标。如 hwnd 是一个子窗口，则 x 用父窗口的客户区坐标表示
yLong，窗口新的 y 坐标。如 hwnd 是一个子窗口，则 y 用父窗口的客户区坐标表示
cxLong，指定新的窗口宽度
cyLong，指定新的窗口高度
wFlagsLong，包含了旗标的一个整数
SWP_DRAWFRAME 围绕窗口画一个框
SWP_HIDEWINDOW 隐藏窗口
SWP_NOACTIVATE 不激活窗口
SWP_NOMOVE 保持当前位置（x 和 y 设定将被忽略）
SWP_NOREDRAW 窗口不自动重画
SWP_NOSIZE 保持当前大小（cx 和 cy 会被忽略）
SWP_NOZORDER 保持窗口在列表的当前位置（hWndInsertAfter 将被忽略）
SWP_SHOWWINDOW 显示窗口
SWP_FRAMECHANGED 强迫一条 WM_NCCALCSIZE 消息进入窗口，即使窗口的大小没有改变
注解
窗口成为最顶级窗口后，它下属的所有窗口也会进入最顶级。一旦将其设为非最顶级，则它的所有下属和物主窗口也会转为非最顶级。Z 序列用垂直于屏幕的一根假想 Z 轴量化这种从顶部到底部排列的窗口顺序

Top
SetWindowText
SetWindowText

VB 声明
Declare Function SetWindowText Lib "user32" Alias "SetWindowTextA" (ByVal hwnd As Long, ByVal lpString As String) As Long
说明
设置窗口的标题文字或控件的内容（在 vb 里使用：针对 vb 窗体，应使用 caption 或 text 属性）
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，要设置文字的窗口的句柄
lpStringString，要设到 hwnd 窗口中的文字

Top
SetWindowWord
SetWindowWord

VB 声明
Declare Function SetWindowWord Lib "user32" Alias "SetWindowWord" (ByVal hwnd As Long, ByVal nIndex As Long, ByVal wNewWord As Long) As Long
说明
在窗口结构中为指定的窗口设置信息（在 vb 里使用：使用时要谨慎）
返回值
Long，指定数据的前一个值
参数表
参数类型及说明
hwndLong，欲为其设置信息的那个窗口的句柄
nIndexLong，参考对 GetWindowWord 函数的 nIndex 参数的说明
wNewWordLong，由 nIndex 指定的窗口信息的新值

Top
ShowOwnedPopups
ShowOwnedPopups

VB 声明
Declare Function ShowOwnedPopups Lib "user32" Alias "ShowOwnedPopups" (ByVal hwnd As Long, ByVal fShow As Long) As Long
说明
显示或隐藏由指定窗口所有的全部弹出式窗口（在 vb 里使用：并不特别有用，因为 vb 不用

弹出式窗口）
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，父窗口的句柄
fShowLong，TRUE（非零）表示显示由 hwnd 所有的所有弹出式窗口；FALSE（零）则隐藏它们

Top
ShowWindow
ShowWindow

VB 声明
Declare Function ShowWindow Lib "user32" Alias "ShowWindow" (ByVal hwnd As Long, ByVal nCmdShow As Long) As Long
说明
控制窗口的可见性（在 vb 里使用：针对 vb 窗体及控件，请使用对应的 vb 属性）
返回值
Long，如窗口之前是可见的，则返回 TRUE（非零），否则返回 FALSE（零）
参数表
参数类型及说明
hwndLong，窗口句柄，要向这个窗口应用由 nCmdShow 指定的命令
nCmdShowLong，为窗口指定可视性方面的一个命令。请用下述任何一个常数
SW_HIDE 隐藏窗口，活动状态给令一个窗口
SW_MINIMIZE 最小化窗口，活动状态给令一个窗口
SW_RESTORE 用原来的大小和位置显示一个窗口，同时令其进入活动状态
SW_SHOW 用当前的大小和位置显示一个窗口，同时令其进入活动状态
SW_SHOWMAXIMIZED 最大化窗口，并将其激活
SW_SHOWMINIMIZED 最小化窗口，并将其激活
SW_SHOWMINNOACTIVE 最小化一个窗口，同时不改变活动窗口
SW_SHOWNA 用当前的大小和位置显示一个窗口，不改变活动窗口
SW_SHOWNOACTIVATE 用最近的大小和位置显示一个窗口，同时不改变活动窗口
SW_SHOWNORMAL 与 SW_RESTORE 相同

Top
ShowWindowAsync
ShowWindowAsync

VB 声明
Declare Function ShowWindowAsync Lib "user32" Alias "ShowWindowAsync" (ByVal hWnd As Long, ByVal nCmdShow As Long) As Long
说明
与 ShowWindow 相似，只是这时的 ShowWindow 命令会投递到指定的窗口，然后进行异步

处理。这样一来，就可控制从属于另一个进程的窗口的可视情况。同时无须担心另一个进程挂起的时候，自己的应用程序也会牵连其中
返回值
Long，如窗口之前是可见的，则返回 TRUE（非零），否则返回 FALSE（零）
参数表
参数类型及说明
hWndLong，欲接收 ShowWindow 命令的窗口
nCmdShowLong，与 ShowWindow 相同

Top
SubtractRect
SubtractRect

VB 声明
Declare Function SubtractRect Lib "user32" Alias "SubtractRect" (lprcDst As RECT, lprcSrc1 As RECT, lprcSrc2 As RECT) As Long
说明
这个函数会装载矩形 lprcDst，它是在矩形 lprcSrc1 中减去 lprcSrc2 得到的结果
返回值
Long，TRUE（非零）表示成功，零表示失败
参数表
参数类型及说明
lprcDstRECT，准备包容 lprcSrc1 减 lprcSrc2 结果的目标矩形
lprcSrc1RECT，第一个源矩形
lprcSrc2RECT，第二个源矩形
注解
lprcSrc1 与 lprcSrc2 这两个矩形必须在水平或垂直方向完全相交。换句话说，倘若在 lprcSrc1 中拿掉与 lprcSrc2 相交的 lprcDst 部分，结果必须是个矩形

Top
TileWindows
TileWindows

VB 声明
Declare Function TileWindows% Lib "user32" (ByVal hwndParent As Long, ByVal wHow As Long, lpRect As RECT, ByVal cKids As Long, lpKids As Long)
Declare Function TileWindowsBynum% Lib "user32" Alias "TileWindows" (ByVal hwndParent As Long, ByVal wHow As Long, ByVal lpRect As Long, ByVal cKids As Long, ByVal lpKids As Long)
说明
以平铺顺序排列窗口（在 vb 里使用：对顶级窗口和 MDI 子窗口有效）
返回值
Integer，成功排列的窗口数量，零表示失败
参数表

参数类型及说明
hwndParentLong，欲对其子窗口进行排列的父窗口。可用 GetDesktopWindow 函数获得顶级窗口（桌面）的句柄
wHowLong，MDITILE_HORIZONTAL 或 MDITILE_VERTICAL，用于设置平铺方向（水平或垂直）
lpRectLong，要在其中平铺窗口的矩形，可设为 NULL，表示用整个客户区域
cKidsLong，lpKids 数组中指定的子窗口数量
lpKidsLong，欲排列子窗口列表的第一个元素。如传递 NULL（务必将参数定义成 ByVal Long），可排列所有子窗口
注解
这个函数不能对诸如控件的子窗口产生作用——只对对顶级窗口和 MDI 子窗口有用。在 MDI 窗口的情况下，指定的父窗口应是 MDIClient 窗口的句柄，不应是 MDI 窗体本身的窗口句柄。可用 GetParent 获得正确的句柄

Top
UnionRect
UnionRect

VB 声明
Declare Function UnionRect Lib "user32" Alias "UnionRect" (lpDestRect As RECT, lpSrc1Rect As RECT, lpSrc2Rect As RECT) As Long
说明
这个函数会装载一个 lpDestRect 目标矩形，它是 lpSrc1Rect 和 lpSrc2Rect 联合起来的结果。目标矩形的所有点都同时位于两个源矩形里；也即是它们的一个交集
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpDestRectRECT，用于容纳 lpSrc1Rect 和 lpSrc2Rect 联合运算结果的目标矩形
lpSrc1RectRECT，第一个源矩形
lpSrc2RectRECT，第二个源矩形

Top
UpdateWindow
UpdateWindow

VB 声明
Declare Function UpdateWindow Lib "user32" Alias "UpdateWindow" (ByVal hwnd As Long) As Long
说明
强制立即更新窗口，窗口中以前屏蔽的所有区域都会重画（在 vb 里使用：如 vb 窗体或控件的任何部分需要更新，可考虑直接使用 refresh 方法
返回值
Long，非零表示成功，零表示失败

参数表
参数类型及说明
hwndLong，欲更新窗口的句柄

Top
ValidateRect
ValidateRect

VB 声明
Declare Function ValidateRect& Lib "user32" (ByVal hwnd As Long, lpRect As RECT)
Declare Function ValidateRectBynum& Lib "user32" Alias "ValidateRect" (ByVal hwnd As Long, ByVal lpRect As Long)
说明
校验窗口的全部或部分客户区。这样便可告之 windows 指定的区域不需要重画
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hwndLong，欲检验的窗口句柄
lpRectRECT，指定一个矩形结构，用于描叙欲校验的矩形部分。可使用 ValidateRectBynum，同时将 lpRect 设为零（Long 数据类型），从而对整个窗口进行校验

Top
WindowFromPoint
WindowFromPoint

VB 声明
Declare Function WindowFromPoint Lib "user32" Alias "WindowFromPoint" (ByVal xPoint As Long, ByVal yPoint As Long) As Long
说明
返回包含了指定点的窗口的句柄。忽略屏蔽、隐藏以及透明窗口
返回值
Long，包含了指定点的窗口的句柄。如指定的点处没有窗口存在，则返回零
参数表
参数类型及说明
xPointLong，x 点值
yPointLong，y 点值

Top
位图、图标和光栅运算函数
位图、图标和光栅运算函数，共三页。第一页，第二页，第三页
BitBlt 将一幅位图从一个设备场景复制到另一个
CopyIcon 制作指定图标或鼠标指针的一个副本。这个副本从属于发出调用的应用程序
CopyImage 复制位图、图标或指针，同时在复制过程中进行一些转换工作

CreateBitmap 按照规定的格式创建一幅与设备有关位图
CreateBitmapIndirect 创建一幅与设备有关位图
CreateCompatibleBitmap 创建一幅与设备有关位图，它与指定的设备场景兼容
CreateCursor 创建一个鼠标指针
CreateDIBitmap 根据一幅与设备无关的位图创建一幅与设备有关的位图
CreateDIBSection 创建一个 DIBSection
CreateIcon 创建一个图标
CreateIconIndirect 创建一个图标
DestroyCursor 清除指定的鼠标指针，并释放它占用的所有系统资源
DestroyIcon 清除图标
DrawIcon 在指定的位置画一个图标
DrawIconEx 描绘一个图标或鼠标指针。与 DrawIcon 相比，这个函数提供了更多的功能
ExtractAssociatedIcon 判断一个可执行程序或 DLL 中是否存在图标，或是否有图标与系统注册表中指定的文件存在关联并提取之
BitBlt
BitBlt

VB 声明
Declare Function BitBlt Lib "gdi32" Alias "BitBlt" (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
说明
将一幅位图从一个设备场景复制到另一个。源和目标 DC 相互间必须兼容
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hDestDCLong，目标设备场景
x,yLong，对目标 DC 中目标矩形左上角位置进行描述的那个点。用目标 DC 的逻辑坐标表示
nWidth,nHeightLong，欲传输图象的宽度和高度
hSrcDCLong，源设备场景。如光栅运算未指定源，则应设为 0
xSrc,ySrcLong，对源 DC 中源矩形左上角位置进行描述的那个点。用源 DC 的逻辑坐标表示
dwRopLong，传输过程要执行的光栅运算
注解
在 NT 环境下，如在一次世界传输中要求在源设备场景中进行剪切或旋转处理，这个函数的执行会失败
如目标和源 DC 的映射关系要求矩形中像素的大小必须在传输过程中改变，那么这个函数会根据需要自动伸缩、旋转、折叠、或切断，以便完成最终的传输过程

Top
CopyIcon
CopyIcon

VB 声明
Declare Function CopyIcon Lib "user32" Alias "CopyIcon" (ByVal hIcon As Long) As Long
说明
制作指定图标或鼠标指针的一个副本。这个副本从属于发出调用的应用程序
返回值
Long，新图标的句柄，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hIconLong，欲复制的图标或指针的句柄
注解
由于这个函数在 win32 中可复制鼠标指针，所以 win32 未提供对 win16 函数 CopyCursor 的支持
从一个 DLL 或其他应用程序载入一个图标后，只有 DLL 或应用程序保持载入状态，那个图标才可使用。这个函数允许我们制作图标从属于自己应用程序的一个副本。一旦不再需要，必须用 DestroyIcon 函数将其清除，以释放占用的系统资源

Top
CopyImage
CopyImage

VB 声明
Declare Function CopyImage Lib "user32" Alias "CopyImage" (ByVal handle As Long, ByVal un1 As Long, ByVal n1 As Long, ByVal n2 As Long, ByVal un2 As Long) As Long
说明
复制位图、图标或指针，同时在复制过程中进行一些转换工作
返回值
Long，执行成功则返回新图象的句柄，零表示失败。会设置 GetLastError
参数表
参数类型及说明
handleLong，欲复制的图象的句柄
un1Long，下述常数之一：MAGE_BITMAP， IMAGE_CURSOR 或 IMAGE_ICON
n1Long，副本以像素表示的宽度
n2Long，副本以像素表示的高度
un2Long，下述常数任意组合：
LR_DELETEORG 删除原来的图象
LR_COPYRETURNORG 忽略 n1 和 n2 设置
LR_MONOCHROME 创建一个单色副本
LR_COPYFROMRESOURCE 在原始资源的基础上创建一个副本，原始图象即是从那个资源中载入的。假设我们想为一个 32×32 的图标制作一个 64×64 的副本。如果不设这个标志，CopyImage 会直接放大原来的图标。而使用这个标志后，CopyImage 首先检查资源文件中是否存在这个图标的一个 64×64 版本，如果存在，就直接载入品质更好的图象
注解
这个函数通常在希望复制已选入其他设备场景的一幅位图时使用——例如，复制已成为 ImageList 控件一部分的某幅位图。选定的位图将不能使用，因为一次只能将位图选入一个

设备场景

Top
CreateBitmap
CreateBitmap

VB 声明
Declare Function CreateBitmap Lib "gdi32" Alias "CreateBitmap" (ByVal nWidth As Long, ByVal nHeight As Long, ByVal nPlanes As Long, ByVal nBitCount As Long, lpBits As Any) As Long
说明
按照规定的格式创建一幅与设备有关位图
返回值
Long，执行成功返回位图的句柄，零表示失败
参数表
参数类型及说明
nWidthLong，位图宽度，以像素为单位
nHeightLong，位图高度，以像素为单位
nPlanesLong，色层数量
nBitCountLong，每像素的位数
lpBitsAny，指向欲载入位图的数据的指针。可设为零，表示不对位图进行初始化（用 ByVal 传递一个零值）。这个数据的格式必须与设备的要求相符。扫描线必须对齐 16 位字边界
注解
一旦不再需要，记住用 DeleteObject 函数释放位图占用的内存和资源
可用这个函数创建单色位图（1 层，每像素一位）。对于彩色位图，则应使用 CreateCompatibleBitmap。这个函数可以胜任工作；但要注意，用它创建的位图在使用时会稍慢一些，因为 Windows 每次使用的时候都必须检查它的位图格式
如果 nWidth 和 nHeight 为零，返回的位图就是一个 1×1 的单色位图

Top
CreateBitmapIndirect
CreateBitmapIndirect

VB 声明
Declare Function CreateBitmapIndirect Lib "gdi32" Alias "CreateBitmapIndirect" (lpBitmap As BITMAP) As Long
说明
创建一幅与设备有关位图
返回值
Long，执行成功返回位图句柄，零表示失败
参数表
参数类型及说明
lpBitmapBITMAP，对一幅逻辑位图进行描述的结构。这个结构中的字段与 CreateBitmap 函数的参数大致对应

注解
如使用由 GetObject 函数获得的 BITMAP 结构，注意 GetObject 不会取回位图数据——只有位图的大小和配置信息
如果 nWidth 和 nHeight 为零，返回的位图就是一个 1×1 的单色位图。也请参考对 CreateBitmap 函数的注解

Top
CreateCompatibleBitmap
CreateCompatibleBitmap

VB 声明
Declare Function CreateCompatibleBitmap Lib "gdi32" Alias "CreateCompatibleBitmap" (ByVal hdc As Long, ByVal nWidth As Long, ByVal nHeight As Long) As Long
说明
创建一幅与设备有关位图，它与指定的设备场景兼容
返回值
Long，执行成功返回位图句柄，零表示失败
参数表
参数类型及说明
hdcLong，设备场景的句柄
nWidthLong，位图宽度，以像素为单位
nHeightLong，位图高度，以像素为单位
注解
内存设备场景即与彩色位图兼容，也与单色位图兼容。这个函数的作用是创建一幅与当前选入 hdc 中的场景兼容。对一个内存场景来说，默认的位图是单色的。倘若内存设备场景有一个 DIBSection 选入其中，这个函数就会返回 DIBSection 的一个句柄。如 hdc 是一幅设备位图，那么结果生成的位图就肯定兼容于设备（也就是说，彩色设备生成的肯定是彩色位图）
如果 nWidth 和 nHeight 为零，返回的位图就是一个 1×1 的单色位图
一旦位图不再需要，一定用 DeleteObject 函数释放它占用的内存及资源

Top
CreateCursor
CreateCursor

VB 声明
Declare Function CreateCursor Lib "user32" Alias "CreateCursor" (ByVal hInstance As Long, ByVal nXhotspot As Long, ByVal nYhotspot As Long, ByVal nWidth As Long, ByVal nHeight As Long, lpANDbitPlane As Any, lpXORbitPlane As Any) As Long
说明
创建一个鼠标指针
返回值
Long，执行成功返回指针的句柄，零表示失败。会设置 GetLastError
参数表
参数类型及说明

hInstanceLong，准备拥有指针的应用程序的实例的句柄。可用 GetWindowWord 函数获得拥有一个窗体或控件的一个实例的句柄
nXhotspot,nYhotspotLong，鼠标指针图象中代表准确指针位置的 X，Y 坐标
nWidthLong，指针图象的宽度。可用 GetSystemMetrics 函数判断一个特定设备的正确编号。VGA 的编号是 32
nHeightLong，指针图象的高度。可用 GetSystemMetrics 函数判断一个特定设备的正确编号。VGA 的编号是 32
lpANDbitPlaneAny，指向 AND 位图数据的指针
lpXORbitPlaneAny，指向 XOR 位图数据的指针
注解
一旦不再需要，注意用 DestroyCursor 函数释放鼠标指针占用的内存及资源

Top
CreateDIBitmap
CreateDIBitmap

VB 声明
Declare Function CreateDIBitmap Lib "gdi32" Alias "CreateDIBitmap" (ByVal hdc As Long, lpInfoHeader As BITMAPINFOHEADER, ByVal dwUsage As Long, lpInitBits As Any, lpInitInfo As BITMAPINFO, ByVal wUsage As Long) As Long
说明
根据一幅与设备无关的位图创建一幅与设备有关的位图
返回值
Long，执行成功返回位图句柄，零表示失败
参数表
参数类型及说明
hdcLong，一个设备场景的句柄，该设备场景定义了要创建的与设备有关位图的配置信息
lpInfoHeaderBITMAPINFOHEADER，对 DIB（与设备无关位图）的格式进行描述的一个结构
dwUsageLong，如不应对位图数据进行初始化，那么设为零。如设为 CBM_INIT，表示根据 lpInitBits 和 lpInitInfo 参数对位图进行初始化
lpInitBitsAny，指向 DIB 格式中的位图数据的一个指针，格式是由 lpInitInfo 指定的
lpInitInfoBITMAPINFO，这个结构对 lpInitBits DIB 的格式及颜色进行了说明
wUsageLong，下述常数之一：
DIB_PAL_COLORS——颜色表包含对当前选定的调色板的索引
DIB_RGB_COLORS——颜色表包含了 RGB 颜色
注解
一旦不再需要，记住用 DeleteObject 函数释放位图占用的内存及资源

Top
CreateDIBSection
CreateDIBSection

VB 声明

Declare Function CreateDIBSection Lib "gdi32" Alias "CreateDIBSection" (ByVal hDC As Long, pBitmapInfo As BITMAPINFO, ByVal un As Long, ByVal lplpVoid As Long, ByVal handle As Long, ByVal dw As Long) As Long
说明
创建一个 DIBSection。这是一个 GDI 对象，可象一幅与设备有关位图那样使用。但是，它在内部作为一幅与设备无关位图保存
返回值
Long，执行成功返回 DIBSection 位图的句柄，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hDCLong，一个设备场景的句柄。如 dw 设为 DIB_PAL_COLORS，那么 DIB 颜色表就会用来自逻辑调色板的颜色进行初始化
pBitmapInfoBITMAPINFO，这个结构初始化成欲创建的那幅位图的配置数据
unLong，下述常数之一：
DIB_PAL_COLORSBITMAPINFO 包含了一个 16 位调色板索引的数组
DIB_RGB_COLORSBITMAPINFO 包含了一个颜色表，其中保存有 32 位颜色（RGBQUAD）
lplpVoidLong，用于载入 DIBSection 数据区的内存地址
handleLong，指向一个文件映射对象的可选句柄，位图将在其中创建。如设为零，Windows 会自动分配内存
dwLong，如指定了句柄，就用这个参数指定位图数据在文件映射对象中的偏移量
注解
一旦不再需要，记住用 DeleteObject 函数删除 DIBSection 位图
如 Windows 分配了一个内存缓冲区，那么对象删除以后，缓冲区也会自动删除。如指定了一个文件映射对象，则不会自动将其清除
在直接访问 DIB 内存之前，首先必须保证 Windows 已完成了绘图（记住，Windows 可能对绘图操作进行了排列处理）。通过调用 gdiFlush 函数，可确保完成所有未决的绘图操作

Top
CreateIcon
CreateIcon

VB 声明
Declare Function CreateIcon Lib "user32" Alias "CreateIcon" (ByVal hInstance As Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal nPlanes As Byte, ByVal nBitsPixel As Byte, lpANDbits As Byte, lpXORbits As Byte) As Long
说明
创建一个图标
返回值
Long，执行成功返回图标的句柄，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hInstanceLong，准备拥有图标的应用程序的实例的句柄。可用 GetWindowWord 函数获得拥有一个窗体或控件的一个实例的句柄
nWidthLong，图标图象的宽度。可用 GetSystemMetrics 函数判断一个特定设备的正确编号。

VGA 的编号是 32
nHeightLong，图标图象的高度。可用 GetSystemMetrics 函数判断一个特定设备的正确编号。
VGA 的编号是 32
nPlanesByte，lpXORbits 数据数组中的色层数量
nBitsPixelByte，lpXORbits 数据数组中每像素的位数
lpANDbitsByte，指向 AND 位图数据的指针
lpXORbitsByte，指向 XOR 位图数据的指针
注解
一旦不再需要，注意用 DestroyIcon 函数释放鼠标指针占用的内存及资源

Top
CreateIconIndirect
CreateIconIndirect

VB 声明
Declare Function CreateIconIndirect Lib "user32" Alias "CreateIconIndirect" (piconinfo As ICONINFO) As Long
说明
创建一个图标
返回值
Long，执行成功返回图标的句柄，零表示失败。会设置 GetLastError
参数表
参数类型及说明
piconinfoICONINFO，这个结构包含的信息与欲创建的图标有关。图标是随同两幅参考位图创建出来的。这个函数允许我们在给定 XOR 和 AND 位图的前提下创建图标，而不是给出实际的掩模数组
注解
一旦不再需要，注意用 DestroyIcon 函数释放鼠标指针占用的内存及资源

Top
DestroyCursor
DestroyCursor

VB 声明
Declare Function DestroyCursor Lib "user32" Alias "DestroyCursor" (ByVal hCursor As Long) As Long
说明
清除指定的鼠标指针，并释放它占用的所有系统资源。不要用这个函数清除随同 LoadCursor 函数载入的系统指针资源
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hCursorLong，欲清除的指针对象的句柄

Top
DestroyIcon
DestroyIcon

VB 声明
Declare Function DestroyIcon Lib "user32" Alias "DestroyIcon" (ByVal hIcon As Long) As Long
说明
清除图标
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hIconLong，图标句柄
注解
不要用这个函数清除随同 LoadIcon 函数载入的系统固有图标

Top
DrawIcon
DrawIcon

VB 声明
Declare Function DrawIcon Lib "user32" Alias "DrawIcon" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long, ByVal hIcon As Long) As Long
说明
在指定的位置画一个图标
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景
x,yLong，想描绘图标的位置（逻辑坐标）
hIconLong，欲描绘图标的句柄

Top
DrawIconEx
DrawIconEx

VB 声明
Declare Function DrawIconEx Lib "user32" Alias "DrawIconEx" (ByVal hdc As Long, ByVal xLeft As Long, ByVal yTop As Long, ByVal hIcon As Long, ByVal cxWidth As Long, ByVal cyWidth As Long, ByVal istepIfAniCur As Long, ByVal hbrFlickerFreeDraw As Long, ByVal diFlags As Long) As Long
说明

描绘一个图标或鼠标指针。与 DrawIcon 相比，这个函数提供了更多的功能
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，要在其中画图的一个设备场景的句柄
xLeft,yTopLong，图标左上角的位置，用逻辑坐标表示
hIconLong，要描绘的图标的句柄
cxWidth,cyWidthLong，希望的图标宽度和高度。图标会自动缩放，与指定的值相符
istepIfAniCurLong，如果 hIcon 是个动画指针，那么该参数指定描绘动画中的哪幅图象。注意 Win32 不能区分图标和指针
hbrFlickerFreeDrawLong，如设为一个刷子句柄，那么函数会将图标画入一幅内存位图，并用背景色填充。随后，将图象直接复制到指定的位置。这样做可绘图时减少闪烁（因为画图过程中重现）
diFlagsLong，下述常数之一：
DI_COMPAT 描绘标准的系统指针，而不是指定的图象
DI_DEFAULTSIZE 忽略 cxWidth 和 cyWidth 设置，并采用原始的图标大小
DI_IMAGE 绘图时使用图标的 XOR 部分（即图标没有透明区域）
DI_MASK 绘图时使用图标的 MASK 部分（如单独使用，可获得图标的掩模）
DI_NORMAL 用常规方式绘图（合并 DI_IMAGE 和 DI_MASK）
注解
应检查 Windows95 是否与指定的标志及参数兼容。Win32 用户手册宣称函数与 Windows 95 是兼容的，但在实际运用中发现它有一定的限制

Top
ExtractAssociatedIcon
ExtractAssociatedIcon

VB 声明
Declare Function ExtractAssociatedIcon Lib "shell32.dll" Alias "ExtractAssociateIconA" (ByVal hInst As Long, ByVal lpIconPath As String, lpiIcon As Long) As Long
说明
这个函数可判断一个可执行程序或 DLL 中是否存在图标，或是否有图标与系统注册表中指定的文件存在关联。随后，它允许我们提取出那些图标
返回值
Long，如果找到任何图标，就返回图标的句柄；否则返回零
参数表
参数类型及说明
hInstLong，当前应用程序的实例句柄。也可用 GetWindowWord 函数取得拥有一个窗体或控件的示例的句柄
lpIconPathString，指定一个文件名，准备从该文件中提取图标。如果文件并非执行程序或 DLL 本身，但通过系统注册表与一个可执行文件关联，就用这个字串装载可执行程序的名字
lpiIconLong，在其中装载图标在可执行文件中的资源标识符

注解
注意至少要把 lpIconPath 字串定义成 MAX_PATH 个字符的长度

Top
ExtractIcon
ExtractIcon

VB 声明
Declare Function ExtractIcon Lib "shell32.dll" Alias "ExtractIconA" (ByVal hInst As Long, ByVal lpszExeFileName As String, ByVal nIconIndex As Long) As Long
说明
判断一个可执行文件或 DLL 中是否有图标存在，并将其提取出来
返回值
Long，如成功，返回指向图标的句柄；如文件中不存在图标，则返回零。如果 nIconIndex 设为-1，就返回文件中的图标总数
参数表
参数类型及说明
hInstLong，当前应用程序的实例句柄。也可用 GetWindowWord 函数取得拥有一个窗体或控件的实例的句柄
lpszExeFileNameString，在其中提取图标的那个程序的全名
nIconIndexLong，欲获取的图标的索引。如果为-1，表示取得文件中的图标总数

Top
GetBitmapBits
GetBitmapBits

VB 声明
Declare Function GetBitmapBits Lib "gdi32" Alias "GetBitmapBits" (ByVal hBitmap As Long, ByVal dwCount As Long, lpBits As Any) As Long
说明
将来自位图的二进制位复制到一个缓冲区
返回值
Long，如执行成功，返回位图中的字节数量；零表示失败。会设置 GetLastError
参数表
参数类型及说明
hBitmapLong，位图的句柄
dwCountLong，欲复制的字节数。如设为零，表示取得位图中的字节数
lpBitsAny，指向容纳位图位的一个缓冲区的指针。注意事先将缓冲区至少初始化成 dwCount 个字节
注解
虽然这个函数能正常工作，但强烈建议使用与设备无关的位图（GetDIBits）

Top
GetBitmapDimensionEx

GetBitmapDimensionEx

VB 声明
Declare Function GetBitmapDimensionEx Lib "gdi32" Alias "GetBitmapDimensionEx" (ByVal hBitmap As Long, lpDimension As SIZE) As Long
说明
取得一幅位图的宽度和高度。它们是由 SetBitmapDimensionEx 函数设置的。Windows 不会使用位图的大小
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hBitmapLong，位图句柄
lpDimensionSIZE，这个结构用于容纳由 SetBitmapDimensionEx 函数设置的位图的大小。这个大小以 1mm 的十分之一为单位
注解
参考 SetBitmapDimensionEx 函数

Top
GetDIBColorTable
GetDIBColorTable

VB 声明
Declare Function GetDIBColorTable Lib "gdi32" Alias "GetDIBColorTable" (ByVal hDC As Long, ByVal un1 As Long, ByVal un2 As Long, pRGBQuad As RGBQUAD) As Long
说明
从选入设备场景的 DIBSection 中取得颜色表信息
返回值
Long，取回的颜色条目数量，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hDCLong，已选入了一个 DIBSection 对象的设备场景
un1Long，颜色表中欲取回的第一个条目的索引
un2Long，欲取回的条目数量
pRGBQuadRGBQUAD，这个结构数组用于装载颜色表信息的第一个条目
注解
如 DIBSection 为每个像素使用了 8 个以上的二进制位，则不会再用颜色表

Top
GetDIBits
GetDIBits

VB 声明
Declare Function GetDIBits Lib "gdi32" Alias "GetDIBits" (ByVal aHDC As Long, ByVal

hBitmap As Long, ByVal nStartScan As Long, ByVal nNumScans As Long, lpBits As Any, lpBI As BITMAPINFO, ByVal wUsage As Long) As Long
说明
将来自一幅位图的二进制位复制到一幅与设备无关的位图里
返回值
Long，非零表示成功，零表示失败。在 Windows 95 中，返回值是返回的扫描线数量
参数表
参数类型及说明
aHDCLong，定义了与设备有关位图 hBitmap 的配置信息的一个设备场景的句柄
hBitmapLong，源位图的句柄。绝对不能将这幅位图选入设备场景
nStartScanLong，欲复制到 DIB 中的第一条扫描线的编号
nNumScansLong，欲复制的扫描线数量
lpBitsAny，指向一个缓冲区的指针。这个缓冲区将用于装载采用 DIB 格式的信息，但不取回数据（用 ByVal 传递零值）
lpBIBITMAPINFO，对 lpBits DIB 的格式及颜色进行说明的一个结构。在 BITMAPINFOHEADER 结构中，从 biSize 到 biCompression 之间的所有字段都必须初始化
wUsageLong，下述常数之一：
DIB_PAL_COLORS 在颜色表中装载一个 16 位所以数组，它们与当前选定的调色板有关
DIB_RGB_COLORS 在颜色表中装载 RGB 颜色
注解
起始扫描线与起点有关。除非将 BITMAPINFOHEADER 结构的 biHeight 字段设为负值，否则起点就位于左下角

Top
GetIconInfo
GetIconInfo

VB 声明
Declare Function GetIconInfo Lib "user32" Alias "GetIconInfo" (ByVal hIcon As Long, piconinfo As ICONINFO) As Long
说明
取得与图标有关的信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hIconLong，图标句柄
piconinfoICONINFO，这个结构装载的信息与包含了 XOR 及 AND 位图的图标有关
注解
函数返回时，由 ICONINFO 结构载入的位图必须由应用程序删去

Top
GetStretchBltMode
GetStretchBltMode

VB 声明
Declare Function GetStretchBltMode Lib "gdi32" Alias "GetStretchBltMode" (ByVal hdc As Long) As Long
说明
判断 StretchBlt 和 StretchDIBits 函数采用的伸缩模式。伸缩模式决定了 Windows 如何控制在伸缩过程中剔除的扫描线
返回值
Long，取得当前的伸缩模式。零表示出错
参数表
参数类型及说明
hdcLong，设备场景的句柄
注解
参考 SetStretchBltMode 函数

Top
LoadBitmap
LoadBitmap, LoadBitmapBynum

VB 声明
Declare Function LoadBitmap& Lib "user32" Alias "LoadBitmapA" (ByVal hInstance As Long, ByVal lpBitmapName As String)
Declare Function LoadBitmapBynum& Lib "user32" Alias "LoadBitmapA" (ByVal hInstance As Long, ByVal lpBitmapName As Long)
说明
从指定的模块或应用程序实例中载入一幅位图。LoadBitmapBynum 是 LoadBitmap 函数的类型安全声明
返回值
Long，已载入的位图的句柄，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hInstanceLong，一个 DLL 的模块句柄；或者一个实例句柄，它指定了包含位图的可执行文件
lpBitmapNameString，作为一个字串，指定欲载入的位图资源。作为一个长整数，指定欲载入的资源 ID。或者一个常数，代表某幅固有系统位图。如装载一幅固有系统位图，hInstance 参数应设为 0。常数在 api32.txt 文件中用前缀 OBM_ 标志。OBM_REDUCE 引用的是标题栏右侧显示的下箭头位图，用于最小化一个窗口。参考 api32.txt，其中有系统固有位图的完整列表
注解
一旦系统位图不再需要，必须用 DeleteObject 函数删除由这个函数取得的位图

Top
LoadCursor
LoadCursor, LoadCursorBynum

VB 声明
Declare Function LoadCursor& Lib "user32" Alias "LoadCursorA" (ByVal hInstance As Long, ByVal lpCursorName As String)
Declare Function LoadCursorBynum& Lib "user32" Alias "LoadCursorA" (ByVal hInstance As Long, ByVal lpCursorName As Long)
说明
从指定的模块或应用程序实例中载入一个鼠标指针。LoadCursorBynum 是 LoadCursor 函数的类型安全声明
返回值
Long，执行成功则返回已载入的指针的句柄；零表示失败。在 Windows 95 和 Win16 环境中，这个函数只能载入标准尺寸的图标
参数表
参数类型及说明
hInstanceLong，一个 DLL 的模块句柄；或者一个实例句柄，指定包含了鼠标指针的可执行程序
lpCursorNameString，作为一个字串，指定欲载入的指针资源。作为一个长整数值，指定欲载入的资源 ID；或者设置一个常数，代表某幅固有系统指针。如装载的是一个固有系统指针，注意 hInstance 参数应设为零。在 api32.txt 文件中以前缀 IDC_ 作为标志
注解
不要清除固有系统指针或从属于其他应用程序的指针。注意 lpCursorName 引用的是一个指针资源。因为假如名字引用的是一个有效的资源。即使它并非指针资源，该函数也会返回一个非零的句柄

Top
LoadCursorFromFile
LoadCursorFromFile

VB 声明
Declare Function LoadCursorFromFile Lib "user32" Alias "LoadCursorFromFileA" (ByVal lpFileName As String) As Long
说明
在一个指针文件或一个动画指针文件（扩展名分别是.cur 和.ani）的基础上创建一个指针
返回值
Long，执行成功则返回指向指针的一个句柄，零表示失败。如果失败，会将 GetLastError 设置为常数 ERROR_FILE_NOT_FOUND
参数表
参数类型及说明
lpFileNameString，包含指针的那个文件的名字
注解
如果将 lpFileName 定义成 Long 型数值（ByVal As Long），就可以（理论上）传递带 OCR_ 前缀的常数，以便获取一个固有系统指针。这一做法在 Windows NT 3.5 下不适用

Top

LoadIcon
LoadIcon, LoadIconBynum

VB 声明
Declare Function LoadIcon& Lib "user32" Alias "LoadIconA" (ByVal hInstance As Long, ByVal lpIconName As String)
Declare Function LoadIconBynum& Lib "user32" Alias "LoadIconA" (ByVal hInstance As Long, ByVal lpIconName As Long)
说明
从指定的模块或应用程序实例中载入一个图标。其中，LoadIconBynum 是 LoadIcon 函数的类型安全声明
返回值
Long，执行成功则返回已载入的图标的句柄；零表示失败。会设置 GetLastError
参数表
参数类型及说明
hInstanceLong，一个 DLL 的模块句柄；或者一个实例句柄，指定包含了图标的可执行程序
lpIconNameString，作为一个字串，指定欲载入的图标资源。作为一个长整数值，指定欲载入的资源 ID；或者设置一个常数，代表某幅固有系统图标。如装载的是一个固有系统图标，注意 hInstance 参数应设为零。在 api32.txt 文件中以前缀 IDI_ 作为标志

注解
在 Windows 95 和 Win16 环境中，这个函数只能载入标准尺寸的图标。用 GetSystemMetrics 可以获得 SM_CXICON 和 SM_CYICON 标准图标设置。用 LoadImage 则可载入非标准尺寸的图标
文件中可能包含了同一个图标资源的多种版本。这个函数会自动挑选与当前显示模式最相配的一种

Top
LoadImage
LoadImage, LoadImageBynum

VB 声明
Declare Function LoadImage& Lib "user32" Alias "LoadImageA" (ByVal hInst As Long, ByVal lpsz As String, ByVal un1 As Long, ByVal n1 As Long, ByVal n2 As Long, ByVal un2 As Long)
Declare Function LoadImageBynum& Lib "user32" Alias "LoadImageA" (ByVal hInst As Long, ByVal lpsz As Long, ByVal un1 As Long, ByVal n1 As Long, ByVal n2 As Long, ByVal un2 As Long)
说明
载入一个位图、图标或指针
返回值
Long，执行成功则返回对象的一个句柄；零表示失败
参数表
参数类型及说明
hInstLong，要从其中载入图象的 DLL 或应用程序模块或实例句柄。零表示装载一幅固有图

象
lpszString，欲载入图象的名字。如指定了 hInst，就用这个参数指定资源或资源的标志符（标志符是一个长整数）。如 hInst 为空，而且已指定了 LR_LOADFROMFILE，那么这个参数代表文件名（位图、图标或指针文件）。如果是个 Long 型值，这个参数就代表固有位图、图标或指针的编号
un1Long，下述常数之一，指定了欲载入的图象类型：IMAGE_BITMAP， IMAGE_CURSOR，IMAGE_ICON
n1,n2Long，要求的图象宽度和高度。图象会根据情况自动伸缩。如设为零，表示用图象的默认大小
un2Long，下述常数的任意组合，它们都在 api32.txt 文件中得到了定义：
LR_DEFAULTCOLOR 以常规方式载入图象
LR_LOADREALSIZE 不对图象进行缩放处理。忽略 n1 和 n2 的设置
LR_CREATEDIBSECTION 如果指定了 IMAGE_BITMAP，就返回 DIBSection 的句柄，而不是位图的句柄
LR_DEFAULTSIZE 如果 n1 和 n2 为零，就使用由系统定义的图象默认大小，而不是图象本身定义的大小
LR_LOADFROMFILE 如 hInst 为零，lpsz 就代表要载入适当类型的一个文件的名字，仅适用于 Win95
LR_LOADMAP3DCOLORS 将图象中的深灰、灰、以及浅灰像素都替换成 COLOR_3DSHADOW，COLOR_3DFACE 以及 COLOR_3DLIGHT 的当前设置
LR_LOADTRANSPARENT 与图象中第一个像素相符的所有像素都由系统替换
LR_MONOCHROME 将图象转换成单色
LR_SHARED 将图象作为一个共享资源载入。在 NT 4.0 中装载固有资源时要用到这个设置

Top
MaskBlt
MaskBlt

VB 声明
Declare Function MaskBlt Lib "gdi32" Alias "MaskBlt" (ByVal hdcDest As Long, ByVal nXDest As Long, ByVal nYDest As Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal hdcSrc As Long, ByVal nXSrc As Long, ByVal nYSrc As Long, ByVal hbmMask As Long, ByVal xMask As Long, ByVal yMask As Long, ByVal dwRop As Long) As Long
说明
执行复杂的图象传输，同时进行掩模（MASK）处理
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcDestLong，目标设备场景
nXDest,nYDestLong，目标图象左上角的 x,y 坐标
nWidth,nHeightLong，图象在目标设备场景中的宽度和高度
hdcSrcLong，源设备场景
nXSrc,nYSrcLong，源图象的左上角 x,y 坐标

hbmMaskLong，作为掩模使用的一幅单色位图的句柄。如果 dwRop 代码包括一个源，那么这幅位图必须与源尺寸相同，否则必须与目标尺寸相符
xMask,yMaskLong，单色掩模位图的 x,y 偏移。这样便允许我们创建一幅使用了多个掩模的大型位图
dwRopLong，一种特殊的光栅运算，在传输过程中使用
适用平台
Windows NT
注解
dwRop 代码是一种非标准的光栅运算代码，由两个普通的光栅运算代码组成：一个前景代码以及一个背景代码
在掩模位图设为 1 的每个像素处，都在传输过程中应用前景转换处理。如对应的掩蔽位图像素为零，则应用背景转换。如果未指定掩模位图，那么这个函数就执行与 BitBlt 相同的操作
如果对源应用了旋转或剪切处理，则函数调用会失败
注意可用 GetDeviceCaps 判断这个函数是否得到了一个特定设备场景的支持

Top
PatBlt
PatBlt

VB 声明
Declare Function PatBlt Lib "gdi32" Alias "PatBlt" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal dwRop As Long) As Long
说明
在当前选定的刷子的基础上，用一个图案填充指定的设备场景
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，欲描绘的一个设备场景的句柄
x,yLong，对目标 DC 中目标矩形左上角位置进行定义的一个点，用逻辑坐标表示
nWidth,nHeightLong，目标矩形的宽度和高度，用逻辑坐标表示
dwRopLong，传输过程中欲进行的光栅运算。对一个源进行引用的光栅运算也许不能在这个函数中使用
注解
可用 GetDeviceCaps 判断这个函数是否得到了一个特定设备场景的支持

Top
PlgBlt
PlgBlt

VB 声明
Declare Function PlgBlt Lib "gdi32" Alias "PlgBlt" (ByVal hdcDest As Long, lpPoint As POINTAPI, ByVal hdcSrc As Long, ByVal nXSrc As Long, ByVal nYSrc As Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal hbmMask As Long, ByVal xMask As Long, ByVal

yMask As Long) As Long
说明
复制一幅位图，同时将其转换成一个平行四边形。利用它可对位图进行旋转处理
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcDestLong，图象使用的目标设备场景
lpPointPOINTAPI，POINTAPI 结构数组中使用的第一个条目。第一个点对应于一个平行四边形左上角位置；第二个点代表右下角位置；第三个点代表左下角位置；第四个点是在前三个点的基础上导出的
hdcSrcLong，图象的源设备场景
nXSrc,nYSrcLong，源图象左上角的 x,y 坐标，采用逻辑坐标系统表示
nWidth,nHeightLong，源图象大小，用逻辑坐标表示
hbmMaskLong，一个可选的句柄，指向一个单色掩模。如设定了这个参数，那么只有与掩模值 1 对应的二进制位才会传输到目的地
xMask,yMaskLong，掩模位图欲使用区域左上角的 x,y 坐标
适用平台
Windows NT
注解
如果对源图象应用了旋转或剪切处理，这个函数的执行就会失败。可用 GetDeviceCaps 判断这个函数是否得到了一个特定设备场景的支持

Top
sample
例程
图形菜单（5K）这个例程有完整的中文注释！演示如何实现图形的菜单条目，用到了几个菜单的 api 函数，本站均有介绍
屏蔽文本框菜单（2.08K）用 api 函数拦截消息，从而屏蔽掉文本框菜单，参考文档中的屏蔽文本框菜单一文
奇形怪状的窗体（6.19K）圆角矩形的窗体中有一个椭圆形的洞。参考奇形怪状的窗体一文
系统菜单（8.89K）在系统菜单里加上自己的菜单项，并可以恢复到原来的菜单
提取图标（9.27K）从 DLL 和 EXE 文件中提取图标，并显示在图片框上
API 函数示例（2.6K）用于列表框
SetBitmapBits
SetBitmapBits

VB 声明
Declare Function SetBitmapBits Lib "gdi32" Alias "SetBitmapBits" (ByVal hBitmap As Long, ByVal dwCount As Long, lpBits As Any) As Long
说明
将来自缓冲区的二进制位复制到一幅位图
返回值
Long，执行成功则返回字节数量，零表示失败

参数表
参数类型及说明
hBitmapLong，位图的句柄
dwCountLong，欲复制的字节数量
lpBitsAny，指向一个缓冲区的指针。这个缓冲区包含了为位图正确格式化的位图位
注解
在 Win32 中，应使用与设备无关位图

Top
SetBitmapDimensionEx
SetBitmapDimensionEx

VB 声明
Declare Function SetBitmapDimensionEx Lib "gdi32" Alias "SetBitmapDimensionEx" (ByVal hbm As Long, ByVal nX As Long, ByVal nY As Long, lpSize As SIZE) As Long
说明
设置一幅位图的宽度。以一毫米的十分之一为单位。Windows 不使用这种信息，但也许能通过 GetBitmapDimensionEx 函数获取
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hbmLong，一幅位图的句柄
nX,nYLong，位图的建议大小，以 0.1mm 为单位
lpSizeSIZE，用于载入前一个位图大小的结构

Top
SetDIBits
SetDIBits

VB 声明
Declare Function SetDIBits Lib "gdi32" Alias "SetDIBits" (ByVal hdc As Long, ByVal hBitmap As Long, ByVal nStartScan As Long, ByVal nNumScans As Long, lpBits As Any, lpBI As BITMAPINFO, ByVal wUsage As Long) As Long
说明
将来自与设备无关位图的二进制位复制到一幅与设备有关的位图里
返回值
Long，执行成功则返回扫描线的数量，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，指向一个设备场景的句柄，那个设备场景定义了与设备有关位图（hBitmap）的配置
hBitmapLong，目标位图的一个句柄。这幅位图绝对不能选入一个设备场景
nStartScanLong，lpBits 数组中第一条扫描线的编号。如 lpBI 之 BITMAPINFOHEADER 部分

的 biHeight 字段是正数，那么这条扫描线就会从位图的底部开始计算；如果是负数，就从顶部开始计算
nNumScansLong，欲复制的扫描线数量
Any，指向一个缓冲区的指针。这个缓冲区包含了以 DIB 格式描述的位图数据；这种格式是由 lpBI 指定的
lpBIBITMAPINFO，对 lpBits DIB 的格式和颜色进行描述的一个结构
wUsageLong，下述常数之一
DIB_PAL_COLORS 颜色表是一个整数数组，其中包含了与目前选入 hdc 设备场景的调色板相关的索引
DIB_RGB_COLORS 颜色表包含了 RG 颜色
注解
用 GetDeviceCaps 判断设备是否支持这个函数

Top
SetDIBitsToDevice
SetDIBitsToDevice

VB 声明
Declare Function SetDIBitsToDevice Lib "gdi32" Alias "SetDIBitsToDevice" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long, ByVal dx As Long, ByVal dy As Long, ByVal SrcX As Long, ByVal SrcY As Long, ByVal Scan As Long, ByVal NumScans As Long, Bits As Any, BitsInfo As BITMAPINFO, ByVal wUsage As Long) As Long
说明
将一幅与设备无关位图的全部或部分数据直接复制到一个设备。这个函数在设备中定义了一个目标矩形，以便接收位图数据。它也在 DIB 中定义了一个源矩形，以便从中提取数据
返回值
Long，执行成功则返回扫描线的数量，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，一个设备场景的句柄。该场景用于接收位图数据
x,yLong，用逻辑坐标表示的目标矩形的起点
dx,dyLong，用目标矩形的设备单位表示的宽度及高度
SrcX,SrcYLong，用设备坐标表示的源矩形在 DIB 中的起点
ScanLong，Bits 数组中第一条扫描线的编号。如 BitsInfo 之 BITMAPINFOHEADER 部分的 biHeight 字段是正数，那么这条扫描线就会从位图的底部开始计算；如果是负数，就从顶部开始计算
NumScansLong，欲复制的扫描线数量
BitsAny，指向一个缓冲区的指针。这个缓冲区包含了以 DIB 格式描述的位图数据；这种格式是由 BitsInfo 指定的
BitsInfoBITMAPINFO，对 Bits DIB 的格式和颜色进行描述的一个结构
wUsageLong，下述常数之一
DIB_PAL_COLORS 颜色表是一个整数数组，其中包含了与目前选入 hdc 设备场景的调色板相关的索引
DIB_RGB_COLORS 颜色表包含了 RG 颜色

注解
用 GetDeviceCaps 判断设备是否支持这个函数

Top
SetStretchBltMode
SetStretchBltMode

VB 声明
Declare Function SetStretchBltMode Lib "gdi32" Alias "SetStretchBltMode" (ByVal hdc As Long, ByVal nStretchMode As Long) As Long
说明
指定 StretchBlt 和 StretchDIBits 函数的伸缩模式。这种伸缩模式定义了 Windows 如何对伸缩过程中剔除的扫描线进行控制。对于 VB 窗体和控件，倘若在 API 绘图过程中使用这个函数，建议恢复原来的 StretchBlt 模式
返回值
Long，上一次伸缩模式的值，零表示失败
参数表
参数类型及说明
hdcLong，设备场景的句柄
nStretchModeLong，新伸缩模式建立在下述任何一个常数的基础上，它们均在 API32.TXT 文件中得到了定义：
STRETCH_ANDSCANS 默认设置。剔除的线段与剩下的线段进行 AND 运算。这个模式通常应用于采用了白色背景的单色位图
STRETCH_DELETESCANS 剔除的线段被简单的清除。这个模式通常用于彩色位图
STRETCH_ORSCANS 剔除的线段与剩下的线段进行 OR 运算。这个模式通常应用于采用了白色背景的单色位图
STRETCH_HALFTONE 目标位图上的像素块被设为源位图上大致近似的块。这个模式要明显慢于其他模式
注解
如果要对伸缩模式有一个更深刻的印象，可想象一下对白色图象中的一条白色细线进行压缩。压缩过程中，像素会从图象中删去。为避免线段消失，在删除它们之前，有必要先对线段中的像素与邻近像素进行 AND 运算。为达到这个目的，应考虑选用 STRETCH_ANDSCANS 伸缩模式

Top
StretchBlt
StretchBlt

VB 声明
Declare Function StretchBlt Lib "gdi32" Alias "StretchBlt" (ByVal hdc 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 nSrcWidth As Long, ByVal nSrcHeight As Long, ByVal dwRop As Long) As Long
说明

将一幅位图从一个设备场景复制到另一个。源和目标 DC 相互间必须兼容。这个函数会在设备场景中定义一个目标矩形，并在位图中定义一个源图象。源矩形会根据需要进行伸缩，以便与目标矩形的大小相符
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，目标设备场景
x,yLong，目标矩形左上角的 x,y 坐标，以逻辑坐标表示
nWidth,nHeightLong，目标矩形的宽度和高度，以逻辑坐标表示
hSrcDCLong，源设备场景。如光栅运算未指定一个源，则这个参数应为零
xSrc,ySrcLong，用源 DC 的逻辑坐标表示的源矩形左上角位置
nSrcWidth,nSrcHeightLong，分别指定用逻辑单位（以源 DC 为基础）传输的一幅图象的宽度和高度。如其中有一个参数的符号（指正负号）与对应的目标参数不符，位图就会在对应的轴上作镜像转换处理
dwRopLong，传输过程中进行的光栅运算。如刷子属于光栅运算的一部分，就使用选入目标 DC 的刷子
注解
可用 GetDeviceCaps 函数判断特定的设备场景是否支持此函数
不可选择对源位图进行剪切或旋转处理，源位图也不能是一个图元文件设备场景

Top
StretchDIBits
StretchDIBits

VB 声明
Declare Function StretchDIBits Lib "gdi32" Alias "StretchDIBits" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long, ByVal dx As Long, ByVal dy As Long, ByVal SrcX As Long, ByVal SrcY As Long, ByVal wSrcWidth As Long, ByVal wSrcHeight As Long, lpBits As Any, lpBitsInfo As BITMAPINFO, ByVal wUsage As Long, ByVal dwRop As Long) As Long
说明
将一幅与设备无关位图的全部或部分数据直接复制到指定的设备场景。这个函数在设备场景中定义了一个目标矩形，用于接收位图数据。它也在 DIB 中定义了一个源矩形，以便从中提取数据。根据设备场景的 StretchBlt 模式（由 SetStretchBltMode 函数决定），源矩形会根据需要调整，以便符合目标矩形的要求
返回值
Long，如函数执行成功，返回欲复制的扫描线的数量；如返回常数 GDI_ERROR，表示出错
参数表
参数类型及说明
hdcLong，一个设备场景的句柄。该场景用于接收位图数据
x,yLong，用逻辑坐标表示的目标矩形的起点
dx,dyLong，目标矩形的宽度及高度，以逻辑坐标表示
SrcX,SrcYLong，用设备坐标表示的源矩形在 DIB 中的起点
wSrcWidth,wSrcHeightLong，源矩形的宽度与高度，用设备坐标表示。如其中有一个参数的

符号（指正负号）与对应的目标参数不符，位图就会在对应的轴上作镜像转换
lpBitsAny，指向一个缓冲区的指针。这个缓冲区包含了以 DIB 格式描述的位图数据；这种格式是由 lpBitsInfo 指定的
lpBitsInfoBITMAPINFO，对 lpBits DIB 的格式和颜色进行描述的一个结构
wUsageLong，下述常数之一
DIB_PAL_COLORS 颜色表是一个整数数组，其中包含了与目前选入 hdc 设备场景的调色板相关的索引
DIB_RGB_COLORS 颜色表包含了 RG 颜色
dwRopLong，欲进行的光栅运算

Top
绘图函数
绘图函数：共七页。第一页，第二页，第三页，第四页，第五页，第六页，第七页
AbortPath 抛弃选入指定设备场景中的所有路径。也取消目前正在进行的任何路径的创建工作
AngleArc 用一个连接弧画一条线
Arc 画一个圆弧
BeginPath 启动一个路径分支
CancelDC 取消另一个线程里的长时间绘图操作
Chord 画一个弦
CloseEnhMetaFile 关闭指定的增强型图元文件设备场景，并将新建的图元文件返回一个句柄
CloseFigure 描绘到一个路径时，关闭当前打开的图形
CloseMetaFile 关闭指定的图元文件设备场景，并向新建的图元文件返回一个句柄
CopyEnhMetaFile 制作指定增强型图元文件的一个副本（拷贝）
CopyMetaFile 制作指定（标准）图元文件的一个副本
CreateBrushIndirect 在一个 LOGBRUSH 数据结构的基础上创建一个刷子
CreateDIBPatternBrush 用一幅与设备无关的位图创建一个刷子，以便指定刷子样式（图案）
CreateEnhMetaFile 创建一个增强型的图元文件设备场景
CreateHatchBrush 创建带有阴影图案的一个刷子
CreateMetaFile 创建一个图元文件设备场景
AbortPath
AbortPath

VB 声明
Declare Function AbortPath Lib "gdi32" Alias "AbortPath" (ByVal hdc As Long) As Long
说明
抛弃选入指定设备场景中的所有路径。也取消目前正在进行的任何路径的创建工作
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景

Top

AngleArc
AngleArc

VB 声明
Declare Function AngleArc Lib "gdi32" Alias "AngleArc" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long, ByVal dwRadius As Long, ByVal eStartAngle As Double, ByVal eSweepAngle As Double) As Long
说明
用一个连接弧画一条线，参考注解
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hdcLong，要在其中作图的设备场景
x,yLong，对弧进行描述的一个圆的中心点坐标
dwRadiusLong，圆的半径
eStartAngleDouble，线同圆连接时的角度（以度数为单位）
eSweepAngleDouble，弧在圆上占据的范围（以度数为单位）
注解
注意 eStartAngle 和 eSweepAngle 参数是以度数为单位指定的，而且应该是单精度数（Single）而不是双精度。相应的函数声明为：Declare Function AngleArc& Lib "gdi32" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long, ByVal dwRadius As Long, ByVal eStartAngle As Single, ByVal eSweepAngle As Single)。
我的理解：本文开头的函数声明复制于 vb 的 api 文本查看器，此处的声明来自于我的参考资料，也不知谁对谁错。参数表的说明，按 vb 的 api 文本查看器中复制来的声明中的数据类型。请使用者注意

Top
Arc
Arc,ArcTo

VB 声明
Declare Function Arc& Lib "gdi32" (ByVal hdc As Long, ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long, ByVal X3 As Long, ByVal Y3 As Long, ByVal X4 As Long, ByVal Y4 As Long)
Declare Function ArcTo& Lib "gdi32" (ByVal hdc As Long, ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long, ByVal X3 As Long, ByVal Y3 As Long, ByVal X4 As Long, ByVal Y4 As Long)
说明
象注解中那样画一个圆弧。（X1,Y1）和（X2,Y2）定义了椭圆的一个范围（约束）矩形。从矩形中心点到点（X3,Y3）的一条线段与椭圆的交点标志着圆弧的起点。而到（X4,Y4）的一条线与椭圆的交点则标志着圆弧的终点。ArcTo 函数会将当前画笔位置设为弧的终点，而 Arc 函数则不会对当前的画笔位置造成影响
返回值

Long，非零表示成功，零表示失败
参数表
参数类型及说明
hdcLong，一个显示场景的句柄
X1,Y1Long，指定围绕椭圆的一个矩形的左上角位置
X2,Y2Long，指定围绕椭圆的一个矩形的右下角位置
X3,Y3Long，指定圆弧起点
X4,Y4Long，指定圆弧终点
注解
在 win16 和 win95 中，约束矩形的宽度和高度必须在 3——32766 间。绘图方向肯定是逆时针方向。
在 win nt 中：绘图方向由 SetArcDirection 函数决定。默认为逆时针方向

Top
BeginPath
BeginPath

VB 声明
Declare Function BeginPath Lib "gdi32" Alias "BeginPath" (ByVal hdc As Long) As Long
说明
启动一个路径分支。在这个命令后执行的 GDI 绘图命令会自动成为路径的一部分。对线段的连接会结合到一起。设备场景中任何现成的路径都会被清除。参考下表，其中列出的函数都可记录到路径中
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，欲在其中记录的设备场景

合法的路径函数
函数 Windows NTWindows 95 函数 Windows NTWindows 95
AngleArcYesNoArcYesNo
ArcToYesNoChordYesNo
EllipseYesNoExtTextOutYesYes
LineToYesYesMoveToExYesYes
PieYesNoPolyBezierYesYes
PolyBezierToYesYesPolyDrawYesNo
PolygonYesYesPolylineYesYes
PolylineToYesYesPolyPolygonYesYes
PolyPolylineYesYesRectangleYesNo
RoundRectYesNoTextOutYesYes

Top

CancelDC
CancelDC

VB 声明
Declare Function CancelDC Lib "gdi32" Alias "CancelDC" (ByVal hdc As Long) As Long
说明
取消另一个线程里的长时间绘图操作
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hdcLong，要在其中取消绘图操作的设备场景
注解
绘图操作有时可能会相当花时间。在多线程应用程序中，这个命令允许一个线程终止另一个在线程里的绘图操作

Top
Chord
Chord

VB 声明
Declare Function Chord Lib "gdi32" Alias "Chord" (ByVal As Long, ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long, ByVal X3 As Long, ByVal Y3 As Long, ByVal X4 As Long, ByVal Y4 As Long) As Long
说明
象注解中那样画一个弦。(X1,Y1)和(X2,Y2)定义了椭圆的一个范围(约束)矩形。点(X3,Y3)和点（X4,Y4）定义了一条线段。该线段与椭圆之间的区域就是“弦”
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，一个显示场景的句柄
X1,Y1Long，指定围绕椭圆的一个矩形的左上角位置
X2,Y2Long，指定围绕椭圆的一个矩形的右下角位置
X3,Y3Long，指定与椭圆相交的一条线的一个点
X4,Y4Long，指定与椭圆相交的一条线的另一个点
注解
在 win16 和 win95 中，约束矩形的宽度和高度必须在 3——32766 个单位之间。请参考 Arc 获得更详细的图例

Top
CloseEnhMetaFile
CloseEnhMetaFile

VB 声明
Declare Function CloseEnhMetaFile Lib "gdi32" Alias "CloseEnhMetaFile" (ByVal hdc As) As Long
说明
关闭指定的增强型图元文件设备场景，并将新建的图元文件返回一个句柄
返回值
Long，指向增强型图元文件的一个句柄。也许能用 PlayEnhMetaFile 函数回放图元文件。零表示出错（原文：A handle to the enhanced metafile. The PlayEnhMetaFile function may be used to play the metafile. Zero on error.）
参数表
参数类型及说明
hdcLong，由 CreateEnhMetaFile 函数返回的一个图元文件设备场景

Top
CloseFigure
CloseFigure

VB 声明
Declare Function CloseFigure Lib "gdi32" Alias "CloseFigure" (ByVal hdc As Long) As Long
说明
描绘到一个路径时，关闭当前打开的图形
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，包含了一个打开的 Path 分支的设备场景
注解
如准备在一个路径里描绘一系列线段，就会有一个打开的图形。调用这个函数的时候，windows 会在当前位置和图形的起始处（通常是最后一个 MoveToEx 操作设置画笔位置的地方）画一条线。图中的这条线和第一条线会连接到一起。注意倘若自己描绘这条线，图形仍会处于打开状态——即使起点与终点相同。这样便造成了与几何画笔的差异。利用 CloseFigure，线段会连接到一起——否则它们会用笔尖显示出来。一旦关闭了图形，在路径中画的下一条线就会从一幅新图形开始。打开的图形会被那些用于填充路径的函数自动关闭

Top
CloseMetaFile
CloseMetaFile

VB 声明
Declare Function CloseMetaFile Lib "gdi32" Alias "CloseMetaFile" (ByVal hMF As Long) As Long
说明
关闭指定的图元文件设备场景，并向新建的图元文件返回一个句柄
返回值

Long，指向图元文件的一个句柄。也许能用 PlayMetaFile 回放图元文件。零表示出错
参数表
参数类型及说明
hMFLong，由 CreateMetaFile 函数返回的一个图元文件设备场景
注解
图元文件不再需要后，一定要用 DeleteMetaFile 函数删除图元文件，并释放它的资源

Top
CopyEnhMetaFile
CopyEnhMetaFile

VB 声明
Declare Function CopyEnhMetaFile Lib "gdi32" Alias "CopyEnhMetaFileA" (ByVal hemfSrc As Long, ByVal lpszFile As String) As Long
说明
制作指定增强型图元文件的一个副本（拷贝）
返回值
Long，如执行成功，返回副本的句柄。零表示失败。会设置 GetLastError
参数表
参数类型及说明
hemfSrcLong，欲复制的增强型图元文件的句柄
lpszFileString，副本的文件名（要在磁盘上创建一个新的图元文件）。可用 vbNullString 向这个参数传递一个 NULL 值，以便在内存中创建副本

Top
CopyMetaFile
CopyMetaFile

VB 声明
Declare Function CopyMetaFile Lib "gdi32" Alias "CopyMetaFileA" (ByVal hMF As Long, ByVal lpFileName As String) As Long
说明
制作指定（标准）图元文件的一个副本
返回值
Long，如执行成功，则返回副本的句柄。零表示失败。会设置 GetLastError
参数表
参数类型及说明
hMFLong，指向欲复制的图元文件的句柄
lpFileNameString，新图元文件的文件名。可用 vbNullString 向这个参数传递一个 NULL 值，以便在内存中创建副本

Top
CreateBrushIndirect
CreateBrushIndirect

VB 声明
Declare Function CreateBrushIndirect Lib "gdi32" Alias "CreateBrushIndirect" (lpLogBrush As LOGBRUSH) As Long
说明
在一个 LOGBRUSH 数据结构的基础上创建一个刷子
返回值
Long，如执行成功，返回指向新刷子的一个句柄。零表示失败
参数表
参数类型及说明
lpLogBrushLOGBRUSH
注解
如不再需要，请用 DeleteObject 函数删除刷子。也请参考 CreateBrush 函数，它的参数与 LOGBRUSH 结构的字段是对应的

Top
CreateDIBPatternBrush
CreateDIBPatternBrush,CreateDIBPatternBrushPt

VB 声明
Declare Function CreateDIBPatternBrush& Lib "gdi32" (ByVal hPackedDIB As Long, ByVal wUsage As Long)
Declare Function CreateDIBPatternBrushPt& Lib "gdi32" (lpPackedDIB As Any, ByVal wUsage As Long)
说明
用一幅与设备无关的位图创建一个刷子，以便指定刷子样式（图案）
返回值
Long，如执行成功，返回指向刷子的一个句柄。零表示失败
参数表
参数类型及说明
hPackedDIB,lpPackedDIBLong，hPackedDIB 是指向一个内存块的全局内存句柄。那个内存块包含了一个 BITMAPINFO 结构，后面跟随一幅与设备无关的位图。lpPackedDIB 是具有相同配置的一个内存块的地址。如指定了单色 DIB，DIB 颜色就会忽略，而换用文本和背景颜色
wUsageLong，下述常数之一：
DIB_PAL_COLORSDIB 颜色表，包含了当前逻辑调色板的索引
DIB_RGB_COLORSDIB 颜色表，包含了 32 位的 RGB 色值
注解
编制 win32 应用程序的时候，最好使用 CreateDIBPatternBrushPt

Top
CreateEnhMetaFile
CreateEnhMetaFile

VB 声明
Declare Function CreateEnhMetaFile Lib "gdi32" Alias "CreateEnhMetaFileA" (ByVal hdcRef As Long, ByVal lpFileName As String, lpRect As RECT, ByVal lpDescription As String) As Long
说明
创建一个增强型的图元文件设备场景。绘图操作也许在这个设备场景中执行。调用 CloseEnhMetaFile 函数关闭了这个设备场景后，会创建一个图元文件句柄，在其中包含记录下来的绘图命令序列。随后，可在任何设备场景中回放这些命令
返回值
Long，一个增强型图元文件设备场景。零表示函数执行出错。不要将这个设备场景与图元文件句柄混淆起来。图元文件设备场景用于描绘图元文件——这与 GDI 绘图函数作为参数使用的其他任何设备场景是一样的。调用 CloseEnhMetaFile 函数的时候，会获得实际的图元文件句柄
参数表
参数类型及说明
hdcRefLong，一个参考设备场景。函数会用该设备场景在图元文件中保存与创建图元文件的那个设备的分辨率有关的信息。如设为零，表示将整个显示器（屏幕）作为参考设备使用
lpFileNameString，这个图元文件的磁盘文件名。文件应有一个.EMF 扩展名。可用 vbNullString 传递一个 NULL，从而创建内存图元文件
lpRectRECT，一个约束矩形，用于描述图元文件的大小和位置（以 0.01 毫米为单位）。可用它精确定义图元文件的物理尺寸
lpDescriptionString，对图元文件的一段说明。包括创建应用程序的名字、一个 NULL 字符、对图元文件的一段说明以及两个 NULL 字符。如："My app" & chr$(0) & "my metafile" & chr$(0) & chr$(0)。如果不愿意包含一段说明，也可设为 vbNullString
注解
与标准图元文件相比，增强型图元文件的一个优点在于它们包括了对图元文件实际大小和位置进行描述的信息，这些信息与它最开始创建时的情况相符。windows 和绘图程序可读取这种信息，在任何设备上实际重现图元文件

Top
CreateHatchBrush
CreateHatchBrush

VB 声明
Declare Function CreateHatchBrush Lib "gdi32" Alias "CreateHatchBrush" (ByVal nIndex As Long, ByVal crColor As Long) As Long
说明
创建带有阴影图案的一个刷子（阴影图案见注解）
返回值
Long，如执行成功，返回指向新刷子的一个句柄。否则返回零。注意在不需要时，用 DeleteObject 清除刷子
参数表
参数类型及说明
nIndexLong，象下图那样指定一种阴影类型
crColorLong，指定刷子的 RGB 前景色

注解

Top
CreateMetaFile
CreateMetaFile

VB 声明
Declare Function CreateMetaFile Lib "gdi32" Alias "CreateMetaFileA" (ByVal lpString As String) As Long
说明
创建一个图元文件设备场景。绘图操作也许能在这个设备中执行。调用 CloseMetaFile 函数关闭了这个设备场景后，会创建一个图元文件句柄，其中包含了记录下来的绘图命令序列。随后，可在任何设备场景中回放这些命令
返回值
Long，指向图元文件设备场景的一个句柄。零表示出错。注意不要把这个设备场景与图元文件句柄混淆起来。图元文件设备场景用于描绘图元文件——它的用法与 GDI 绘图函数作为参数使用的其他任何设备场景都是一样的。以后调用了 CloseMetaFile 函数以后，会得到实际的图元文件句柄
参数表
参数类型及说明
lpStringString，欲用来容纳图元文件的文件名。用 vbNullString 可传递一个零值，从而创建一个内存句柄
注解
尽管这个函数能创建基于磁盘的图元文件，但这些图元文件并非通常在 windows 下使用的标准“可放置”图元文件格式

Top
CreatePatternBrush
CreatePatternBrush

VB 声明
Declare Function CreatePatternBrush Lib "gdi32" Alias "CreatePatternBrush" (ByVal hBitmap As Long) As Long
说明
用指定了刷子图案的一幅位图创建一个刷子
返回值
Long，如执行成功，则返回新刷子的一个句柄；否则返回零
参数表
参数类型及说明
hBitmapLong，指向一幅位图的句柄。如指定了单色位图，文本和背景色就会在图案中使用
注解
一旦刷子不再需要，记得用 DeleteObject 函数将其删除。不要在这个函数里使用作为 DIB 分区创建的位图

Top
CreatePen
CreatePen

VB 声明
Declare Function CreatePen Lib "gdi32" Alias "CreatePen" (ByVal nPenStyle As Long, ByVal nWidth As Long, ByVal crColor As Long) As Long
说明
用指定的样式、宽度和颜色创建一个画笔
返回值
Long，如函数执行成功，就返回指向新画笔的一个句柄；否则返回零
参数表
参数类型及说明
nPenStyleLong，指定画笔样式，可以是下述常数之一
PS_SOLID 画笔画出的是实线
PS_DASH 画笔画出的是虚线（nWidth 必须是 1）
PS_DOT 画笔画出的是点线（nWidth 必须是 1）
PS_DASHDOT 画笔画出的是点划线（nWidth 必须是 1）
PS_DASHDOTDOT 画笔画出的是点-点-划线（nWidth 必须是 1）
PS_NULL 画笔不能画图
PS_INSIDEFRAME 画笔在由椭圆、矩形、圆角矩形、饼图以及弦等生成的封闭对象框中画图。如指定的准确 RGB 颜色不存在，就进行抖动处理
nWidthLong，以逻辑单位表示的画笔的宽度
crColorLong，画笔的 RGB 颜色
注解
一旦不再需要画笔，记得用 DeleteObject 函数将其删除

Top
CreatePenIndirect
CreatePenIndirect

VB 声明
Declare Function CreatePenIndirect Lib "gdi32" Alias "CreatePenIndirect" (lpLogPen As LOGPEN) As Long
说明
根据指定的 LOGPEN 结构创建一个画笔
返回值
Long，如执行成功，返回指向新画笔的一个句柄；否则返回零
参数表
参数类型及说明
lpLogPenLOGPEN，逻辑画笔结构。这个结构与 CreatePen 函数的参数非常接近
注解
一旦不再需要画笔，记得用 DeleteObject 函数将其删除

Top
CreateSolidBrush
CreateSolidBrush

VB 声明
Declare Function CreateSolidBrush Lib "gdi32" Alias "CreateSolidBrush" (ByVal crColor As Long) As Long
说明
用纯色创建一个刷子
返回值
Long，如执行成功，返回新刷子的一个句柄；否则返回零
参数表
参数类型及说明
crColorLong，数字的 RGB 彩色
注解
一旦刷子不再需要，就用 DeleteObject 函数将其删除

Top
DeleteEnhMetaFile
DeleteEnhMetaFile

VB 声明
Declare Function DeleteEnhMetaFile Lib "gdi32" Alias "DeleteEnhMetaFile" (ByVal hemf As Long) As Long
说明
删除指定的增强型图元文件
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hemfLong，增强型图元文件的句柄
注解
如图元文件是建立在一个磁盘文件基础上的，那么文件本身就不会由函数删除。这样有可能由 GetEnhMetaFile 函数再次将其打开

Top
DeleteMetaFile
DeleteMetaFile

VB 声明
Declare Function DeleteMetaFile Lib "gdi32" Alias "DeleteMetaFile" (ByVal hMF As Long) As Long
说明

删除指定的图元文件
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hMFLong，图元文件的句柄
注解
如图元文件是建立在一个磁盘文件基础上的，那么文件本身就不会由函数删除。这样有可能造成由 GetEnhMetaFile 函数再次将其打开。然而，这个图元文件并不采用标准可放置图元文件格式

Top
DeleteObject
DeleteObject

VB 声明
Declare Function DeleteObject Lib "gdi32" Alias "DeleteObject" (ByVal hObject As Long) As Long
说明
用这个函数删除 GDI 对象，比如画笔、刷子、字体、位图、区域以及调色板等等。对象使用的所有系统资源都会被释放
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hObjectLong，一个 GDI 对象的句柄
注解
不要删除一个已选入设备场景的画笔、刷子或位图。如删除以位图为基础的阴影（图案）刷子，位图不会由这个函数删除——只有刷子被删掉

Top
DrawEdge
DrawEdge

VB 声明
Declare Function DrawEdge Lib "user32" Alias "DrawEdge" (ByVal hdc As Long, qrc As RECT, ByVal edge As Long, ByVal grfFlags As Long) As Long
说明
用指定的样式描绘一个矩形的边框
返回值
Long，TRUE（非零）表示成功，否则返回零。会设置 GetLastError。（在 vb 里使用：推荐使用。利用这个函数，我们没有必要再使用许多 3D 边框和面板。所以就资源和内存的占用率来说，这个函数的效率要高得多。它可在一定程度上提升性能）
参数表

参数类型及说明
hdcLong，要在其中绘图的设备场景
qrcRECT，要为其描绘边框的矩形
edgeLong，带有前缀 BDR_的两个常数的组合。一个指定内部边框是上凸还是下凹；另一个则指定外部边框。有时能换用带 EDGE_前缀的常数。
grfFlagsLong，带有 BF_前缀的常数的组合
注解
由于这是一个 GDI 函数，所以矩形坐标是逻辑坐标

Top
DrawEscape
DrawEscape

VB 声明
Declare Function DrawEscape Lib "gdi32" Alias "DrawEscape" (ByVal hdc As Long, ByVal nEscape As Long, ByVal cbInput As Long, ByVal lpszInData As String) As Long
说明
换码（Escape）函数将数据直接发至显示设备驱动程序（在 vb 里使用：能够使用。但由于 Escape 对设备有较强的依赖性，所以除非万不得以，尽量不要用它）
返回值
Long，非零表示成功，零表示错误（QUERYESCSUPPORT 换码例外；返回 TRUE 表示支持特定的换码；否则返回零）
参数表
参数类型及说明
hdcLong，显示设备的设备场景
nEscapeLong，指定欲执行的换码的一个常数
cbInputLong，输入缓冲区的长度
lpszInDataString，输入字串或缓冲区
注解
可用 QUERYESCSUPPORT 换码判断一个特定的换码是否得到当前显示驱动程序的支持

Top
DrawFocusRect
DrawFocusRect

VB 声明
Declare Function DrawFocusRect Lib "user32" Alias "DrawFocusRect" (ByVal hdc As Long, lpRect As RECT) As Long
说明
画一个焦点矩形。这个矩形是在标志焦点的样式中通过异或运算完成的（焦点通常用一个点线表示）。如用同样的参数再次调用这个函数，就表示删除焦点矩形
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表

参数类型及说明
hdcLong，设备场景的句柄
lpRectRECT，要在逻辑坐标中描绘的矩形

Top
DrawFrameControl
DrawFrameControl

VB 声明
Declare Function DrawFrameControl Lib "user32" Alias "DrawFrameControl" (ByVal hDC As Long, lpRect As RECT, ByVal un1 As Long, ByVal un2 As Long) As Long
说明
这个函数用于描绘一个标准控件。例如，可描绘一个按钮或滚动条的帧（原文：This function draws a standard control. For example, you can draw the frame of a button or scrollbar.）
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hDCLong，要在其中作画的设备场景
lpRectRECT，指定帧的位置及大小的一个矩形
un1Long，指定帧类型的一个常数。这些常数包括 DFC_BUTTON，DFC_CAPTION，DFC_MENU，以及 DFC_SCROLL
un2Long，一个常数，指定欲描绘的帧的状态。由带有前缀 DFCS_的一个常数构成

Top
DrawState
DrawState

VB 声明
Declare Function DrawState Lib "user32" Alias "DrawStateA" (ByVal hDC As Long, ByVal hBrush As Long, ByVal lpDrawStateProc As Long, ByVal lParam As Long, ByVal wParam As Long, ByVal n1 As Long, ByVal n2 As Long, ByVal n3 As Long, ByVal n4 As Long, ByVal un As Long) As Long
说明
这个函数可为一幅图象或绘图操作应用各式各样的效果
返回值
Long，TRUE（非零）表示成功，FALSE 表示失败
参数表
参数类型及说明
hDCLong，要在其中绘图的设备场景
hBrushLong，如状态为 DSS_MONO（在 un 参数中设定），则指定一个刷子句柄
lpDrawStateProcLong，指向一个函数地址的指针。如图象类型为 DST_COMPLEX，必须设置这个参数。对于 DST_TEXT，则可设可不设
lParamLong，由图象的类型决定

wParamLong，由图象的类型决定
n1Long，图象的水平位置
n2Long，图象的垂直位置
n3Long，图象的宽度。如图象类型为 DST_COMPLEX，必须设置这个参数。而对于其他类型，则可以设为零。如为零，表示该参数在图象的基础上计算
n4Long，图象的高度。如图象类型为 DST_COMPLEX，必须设置这个参数。而对于其他类型，则可以设为零。如为零，表示该参数在图象的基础上计算
unLong，图象类型和状态的一个组合。参见下表
图象类型
DST_BITMAPlParam 中的句柄
DST_COMPLEX 绘图在由 lpDrawStateProc 参数指定的回调函数期间执行。lParam 和 wParam 会传递给回调事件
DST_ICONlParam 包括图标句柄
DST_TEXTlParam 代表文字的地址（可使用一个字串别名），wParam 代表字串的长度
DST_PREFIXTEXT 与 DST_TEXT 类似，只是 & 字符指出为下各字符加上下划线
图象状态常数
DSS_NORMAL 普通图象
DSS_UNION 图象进行抖动处理
DSS_DISABLED 图象具有浮雕效果
DSS_MONO 用 hBrush 描绘图象
DSS_RIGHT 手册未正式说明——经实验证明没有什么作用（原文：Undocumented-experimentation seems to show no effect.）
注解
windows95 用它获得我们应用于图象的一些视觉效果；例如，可使位图或其他图象在视觉上进入禁用或抖动状态。对于位图和图标，它在描绘位图或图标的时候应用一种效果。对于文本，既可让函数画出文本，也可在一个回调函数中执行自己的绘图操作。对于复杂的（用户自定义）图象，则必须用一个回调函数。在回调函数执行过程中，用自己的代码将自己希望的任何东西画入设备场景。在这之后，利用 DrawState 函数应用希望的效果

Top
Ellipse
Ellipse

VB 声明
Declare Function Ellipse Lib "gdi32" Alias "Ellipse" (ByVal hdc As Long, ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long) As Long
说明
描绘一个椭圆，由指定的矩形围绕。椭圆用当前选择的画笔描绘，并用当前选择的刷子填充
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
X1，Y1Long，约束矩形采用逻辑坐标的左上角位置

X2，Y2Long，约束矩形采用逻辑坐标的右下角位置

Top
EndPath
EndPath

VB 声明
Declare Function EndPath Lib "gdi32" Alias "EndPath" (ByVal hdc As Long) As Long
说明
停止定义一个路径。如执行成功，BeginPath 函数调用和这个函数之间发生的所有绘图操作都会正式成为指定设备场景的路径
返回值
Long，非零表示成功，零表示失败。会将 GetLastError 设置为下述值之一：ERROR_CAN_NOT_COMPLETE 或 ERROR_INVALID_PARAMETER
参数表
参数类型及说明
hdcLong，设备场景

Top
EnumEnhMetaFile
EnumEnhMetaFile

VB 声明
Declare Function EnumEnhMetaFile Lib "gdi32" Alias "EnumEnhMetaFile" (ByVal hdc As Long, ByVal hemf As Long, ByVal lpEnhMetaFunc As Long, lpData As Any, lpRect As RECT) As Long
说明
针对一个增强型图元文件，列举其中单独的图元文件记录。每条图元文件记录都由单个 GDI 命令组成。可伴随 PlayEnhMetaFileRecord 函数使用，选择性的回放图元文件的一部分
返回值
Long，如图元文件的所有记录都列举出来，就返回 TRUE（非零）；否则返回零
参数表
参数类型及说明
hdcLong，用于输出的设备场景的句柄。只有在用回调函数进行绘图操作的时候，才要求设置这个参数
hemfLong，欲列举的一个增强型图元文件的句柄
lpEnhMetaFuncLong，指向为每个图元文件调用的一个函数的指针
lpData 用户自定义的值
lpRectRECT，定义图元文件边框的一个矩形

Top
EnumMetaFile
EnumMetaFile

VB 声明

Declare Function EnumMetaFile Lib "gdi32" Alias "EnumMetaFile" (ByVal hDC As Long, ByVal hMetafile As Long, ByVal lpMFEnumProc As Long, ByVal lParam As Long) As Long
说明
为一个标准的 windows 图元文件枚举单独的图元文件记录。每条图元文件记录都包含了单个 GDI 绘图命令。可与 PlayMetaFileRecord 函数联合使用，选择性的回放图元文件的某一部分
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hDCLong，用于输出的设备场景的句柄。只有在用回调函数进行绘图操作的时候，才要求设置这个参数
hMetafileLong，欲列举的一个标准图元文件的句柄
lpMFEnumProcLong，指向为每个图元文件调用的一个函数的指针
lParamLong，用户自定义的值

Top
EnumObjects
EnumObjects

VB 声明
Declare Function EnumObjects Lib "gdi32" Alias "EnumObjects" (ByVal hDC As Long, ByVal n As Long, ByVal lpGOBJEnumProc As Long, lpVoid As Any) As Long
说明
枚举可随同指定设备场景使用的画笔和刷子
返回值
Long，如函数要枚举的对象太多，就返回-1。否则由用户自己定义
参数表
参数类型及说明
hDCLong，设备场景的句柄
nLong，欲枚举的对象的类型。请查找带 OBJ_前缀的常数，这样可得到一个对象列表。win32 手册建议只使用 OBJ_PEN 和 OBJ_BRUSH 两个常数
lpGOBJEnumProcLong，指向为每个 GDI 对象调用的指针
lpVoid 枚举过程中传递给回调函数的值

Top
ExtCreatePen
ExtCreatePen

VB 声明
Declare Function ExtCreatePen Lib "gdi32" Alias "ExtCreatePen" (ByVal dwPenStyle As Long, ByVal dwWidth As Long, lplb As LOGBRUSH, ByVal dwStyleCount As Long, lpStyle As Long) As Long
说明

创建一个扩展画笔（装饰或几何）
返回值
Long，如执行成功，返回一个指向扩展画笔的句柄。零表示执行出错。一旦不再需要，记得用 DeleteObject 将画笔删除
参数表
参数类型及说明
dwPenStyleLong，画笔样式来自下述常数组的任何一个常数的组合（OR 运算）：
PS_COSMETIC or PS_GEOMETRIC 画笔的类型
PS_ALTERNATE, PS_SOLID, PS_DASH, PS_DOT, PS_DASHDOT, PS_DASHDOTDOT, PS_NULL, PS_USERSTYLE, PS_INSIDEFRAME 画笔的样式
PS_ENDCAP_???画笔的笔尖
PS_JOIN_???在图形中连接线段或在路径中连接直线的方式
dwWidthLong，指定线宽。几何画笔的线宽肯定是 1
lplbLOGBRUSH，lbColor 代表画笔颜色。对于装饰画笔，lbStyle 为 PS_SOLID；对于几何画笔，lbStyle 则代表实际的样式。针对几何画笔，必须设置其他所有字体
dwStyleCountLong，如指定了 PS_USERSTYLE，则代表 lpStyle 数组中的条目数量
lpStyleLong，指定 PS_USERSTYLE 的“线段/空白”对（原文：Line/space pairs for PS_USERSTYLE）

Top
ExtFloodFill
ExtFloodFill

VB 声明
Declare Function ExtFloodFill Lib "gdi32" Alias "ExtFloodFill" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long, ByVal crColor As Long, ByVal wFillType As Long) As Long
说明
在指定的设备场景里，用当前选择的刷子填充一个区域
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
x,yLong，开始填充的一个点，采用逻辑坐标表示
crColorLong，要使用的边界颜色
wFillTypeLong，欲执行的填充类型，由下述任何一个常数决定
FLOODFILLBORDER 等同于 FloodFill 函数的功能
FLOODFILLSURFACE 从指定的点向外填充，只到找到了 crColor 颜色（在边框采用了多种颜色时使用）
注解
如指定了 FLOODFILLBORDER，那么 x,y 点绝对不能为 crColor 颜色。如指定了 FLOODFILLSURFACE，那么 x,y 点必须是 crColor 颜色。这个函数只能在光栅设备中使用。可用 GetDeviceCaps 函数判断设备是否支持这个函数
提示

一旦指定了 FLOODFILLBORDER，务必保证初始点的颜色没有 crColor。如果使用的是 FLOODFILLSURFACE，务必保证初始点有颜色 crColor（这是函数执行失败最常见的两个原因）。注意保证初始点位于剪切区内

Top
FillPath
FillPath

VB 声明
Declare Function FillPath Lib "gdi32" Alias "FillPath" (ByVal hdc As Long) As Long
说明
关闭路径中任何打开的图形，并用当前刷子填充
返回值
Long，非零表示成功，零表示失败。会将 GetLastError 设置为下述值之一：ERROR_CAN_NOT_COMPLETE，ERROR_INVALID_PARAMETER，ERROR_NOT_ENOUGH_MEMORY
参数表
hdcLong，欲在其中操作的设备场景
注解
函数执行完毕后，选定的路径会自行清除

Top
FillRect
FillRect

VB 声明
Declare Function FillRect Lib "user32" Alias "FillRect" (ByVal hdc As Long, lpRect As RECT, ByVal hBrush As Long) As Long
说明
用指定的刷子填充一个矩形
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
lpRectRECT，对填充区域进行描述的一个矩形，采用逻辑坐标
hBrushLong，欲使用的刷子的句柄
注解
矩形的右边和底边不会描绘

Top
FlattenPath
FlattenPath

VB 声明
Declare Function FlattenPath Lib "gdi32" Alias "FlattenPath" (ByVal hdc As Long) As Long
说明
将一个路径中的所有曲线都转换成线段
返回值
Long，非零表示成功，零表示失败。会将 GetLastError 设置为下述值之一：ERROR_CAN_NOT_COMPLETE，ERROR_INVALID_PARAMETER，ERROR_NOT_ENOUGH_MEMORY
参数表
参数类型及说明
hdcLong，包含了路径的设备场景

Top
FloodFill
FloodFill

VB 声明
Declare Function FloodFill Lib "gdi32" Alias "FloodFill" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long, ByVal crColor As Long) As Long
说明
用当前选定的刷子在指定的设备场景中填充一个区域。区域是由颜色 crColor 定义的
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
x,yLong，开始填充的那个点，用逻辑坐标表示
crColorLong，欲使用的边界颜色。由这个颜色包围的表面会被填充
注解
点 x,y 绝对不能有颜色 crColor，而且必须在剪切区域内。这个函数只对光栅设备有效，请参考 ExtFloodFill 的注解

Top
FrameRect
FrameRect

VB 声明
Declare Function FrameRect Lib "user32" Alias "FrameRect" (ByVal hdc As Long, lpRect As RECT, ByVal hBrush As Long) As Long
说明
用指定的刷子围绕一个矩形画一个边框（组成一个帧），边框的宽度是一个逻辑单位
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表

参数类型及说明
hdcLong，设备场景的句柄
lpRectRECT，对要描绘的边框进行描述的一个矩形。这等效于将画笔设成一个单位的宽度，然后用矩形函数画出一个矩形
hBrushLong，欲使用的刷子的句柄
注解
lpRect 的顶部边距值必须小于底部边距值。lpRect 的左侧边距值必须小于右侧边距值

Top
GdiComment
GdiComment

VB 声明
Declare Function GdiComment Lib "gdi32" Alias "GdiComment" (ByVal hdc As Long, ByVal cbSize As Long, lpData As Byte) As Long
说明
为指定的增强型图元文件设备场景添加一条注释信息
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hdcLong，目标增强型图元文件设备场景的句柄
cbSizeLong，欲嵌入图元文件的数据长度
lpDataByte，一个注释结构或一个缓冲区的 Long 内存地址，其中包含了欲添加的注释正文
注解
尽管可在图元文件中嵌入任何专用或私有（Private）信息，但只有几种全局数据格式能够嵌入。如将缓冲区看作一个 32 位 Long 型值的数组，则全局注释的值就是下面这个样子：
第一个条目是常数 GDICOMMENT_IDENTIFIER
第二个条目如下所示：
首先是一个 GDICOMMENT_WINDOWS_METAFILE——在增强型图元文件中嵌入一个标准图元文件。它的后面跟随下述值之一：
□ 标准图元文件的版本号
□ 一个校验和（checksum）值：所有图元文件数据的总和——包括这个值——必须是零
□ 零
□ 后面跟随的窗口图元文件的大小
GDICOMMENT_BEGINGROUP——标志一组绘图命令在增强型图元文件在中的起始处。它的后面跟随：
□ 四个 Long 值。定义一个 RECT 结构。结构中包含了绘图命令的约束矩形
□ 可选的 Unicode 字串的长度。字串中包含对命令组的说明文字。如不想提供说明，可设为零
GDICOMMENT_ENDGROUP——标志增强型图元文件中的一组绘图命令的结尾
GDICOMMENT_MULTIFORMATS——以不同的格式嵌入一幅处理过的图象。例如，可利用这个注释在一个增强型图元文件中嵌入一个封装式 PostScript 图象。回放这条记录的时候，windows 会重画它能描绘的第一组格式。它的后面跟随：

□ 四个 Long 值。定义一个 RECT 结构。结构中包含了绘图命令的约束矩形
□ 包括在注释中的格式数量
□ 一系列 EMRFORMAT 结构，每种格式使用一个

Top
GdiFlush
GdiFlush

VB 声明
Declare Function GdiFlush Lib "gdi32" Alias "GdiFlush" () As Long
说明
执行任何未决的绘图操作
返回值
Long，如所有未决的绘图操作都成功完成，就返回 TRUE（非零）。如任何一个操作失败，就返回零值
注解
通过成批合并绘图操作命令，win32 图形子系统（GDI）可改善绘图的性能。如调用一系列绘图命令，他们都返回布尔值（TRUE 表示成功，零表示失败），就可将他们置于一个内部 GDI 队列里。此时，函数可以立即返回。随后，GDI 子系统会执行这些待决的绘图命令。可考虑一种最常见的情况。在这种情况下，系统安装了一块显示卡。卡上自带图形处理器或加速器。画图的时候，GDI 只需将图形命令简单的发送给显示卡，另其完成实际的操作。如果必须等待每个绘图命令都完成并返回，系统和应用程序的性能就会受到显示卡绘图速度的极大限制。所以在这个时候，GDI 将绘图命令置于一个名为“批”（Batch）的队列里。这样一来，系统和应用程序就能继续运行，同时仍然让显示卡进行绘图操作
GdiFlush 命令指示应用程序进入等待状态，直到所有待决的绘图操作完成为止。如执行的是一个特殊的 GDI 绘图命令，它不会返回一个布尔值，那么也会面临这种情况。例如，GetPixel 函数需要读取一个像素值。但除非所有待决的绘图完成，否则该函数不能可靠的完成工作

Top
GdiGetBatchLimit
GdiGetBatchLimit

VB 声明
Declare Function GdiGetBatchLimit Lib "gdi32" Alias "GdiGetBatchLimit" () As Long
说明
判断有多少个 GDI 绘图命令位于队列中
返回值
Long，待决绘图命令的最大数量，零表示出错。会设置 GetLastError
注解
参考对 GdiFlush 的注解

Top
GdiSetBatchLimit
GdiSetBatchLimit

VB 声明
Declare Function GdiSetBatchLimit Lib "gdi32" Alias "GdiSetBatchLimit" (ByVal dwLimit As Long) As Long
说明
指定有多少个 GDI 绘图命令能够进入队列
返回值
Long，如执行成功，返回前一个限制；零表示出错。会设置 GetLastError
参数表
参数类型及说明
dwLimitLong，可排队的绘图操作最大数量；0 意味着恢复默认值；1 表示禁止绘图命令排队
注解
参考对 GdiFlush 的注解

Top
GetArcDirection
GetArcDirection

VB 声明
Declare Function GetArcDirection Lib "gdi32" Alias "GetArcDirection" (ByVal hdc As Long) As Long
说明
画圆弧的时候，判断当前采用的绘图方向
返回值
Long，常数 AD_COUNTERCLOCKWISE（逆时针）或 AD_CLOCKWISE（顺时针），零表示出错
参数表
参数类型及说明
hdcLong，要查询的设备场景

Top
GetBkColor
GetBkColor

VB 声明
Declare Function GetBkColor Lib "gdi32" Alias "GetBkColor" (ByVal hdc As Long) As Long
说明
取得指定设备场景当前的背景颜色
返回值
Long，当前背景色的 RGB 颜色值
参数表
参数类型及说明
hdcLong，欲查询背景颜色的一个设备场景

Top
GetBkMode
GetBkMode

VB 声明
Declare Function GetBkMode Lib "gdi32" Alias "GetBkMode" (ByVal hdc As Long) As Long
说明
针对指定的设备场景，取得当前的背景填充模式
返回值
Long，下述常数之一，零意味着出错
OPAQUE 将文本、阴影刷以及虚线画笔线段的背景设为当前的背景色
TRANSPARENT 不修改文本、阴影刷以及虚线画笔线段的背景
参数表
参数类型及说明
hdcLong，设备场景的句柄

Top
GetBrushOrgEx
GetBrushOrgEx

VB 声明
Declare Function GetBrushOrgEx Lib "gdi32" Alias "GetBrushOrgEx" (ByVal hDC As Long, lpPoint As POINTAPI) As Long
说明
判断指定设备场景中当前选定刷子起点
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hDCLong，设备场景的句柄
lpPointPOINTAPI，这个结构用来装载当前刷子的起点

Top
GetCurrentObject
GetCurrentObject

VB 声明
Declare Function GetCurrentObject Lib "gdi32" Alias "GetCurrentObject" (ByVal hdc As Long, ByVal uObjectType As Long) As Long
说明
用于获得指定类型的当前选定对象
返回值
Long，当前选定对象的句柄，零表示错误

参数表
参数类型及说明
hdcLong，欲查询的设备场景
uObjectTypeLong，对象类型。可以是下述常数之一：OBJ_PEN，OBJ_BRUSH，OBJ_PAL，OBJ_FONT，或 OBJ_BITMAP

Top
GetCurrentPositionEx
GetCurrentPositionEx

VB 声明
Declare Function GetCurrentPositionEx Lib "gdi32" Alias "GetCurrentPositionEx" (ByVal hdc As Long, lpPoint As POINTAPI) As Long
说明
在指定的设备场景中取得当前的画笔位置
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hdcLong，设备场景的句柄
lpPointPOINTAPI，用于装载当前位置

Top
GetEnhMetaFile
GetEnhMetaFile

VB 声明
Declare Function GetEnhMetaFile Lib "gdi32" Alias "GetEnhMetaFileA" (ByVal lpszMetaFile As String) As Long
说明
取得磁盘文件中包含的一个增强型图元文件的图元文件句柄（原文：Retrieves a metafile handle to an enhanced metafile contained in a disk file）
返回值
Long，指向图元文件的句柄。零表示出错
参数表
参数类型及说明
lpszMetaFileString，包含了增强型图元文件的一个磁盘文件的名字

Top
GetEnhMetaFileBits
GetEnhMetaFileBits

VB 声明
Declare Function GetEnhMetaFileBits Lib "gdi32" Alias "GetEnhMetaFileBits" (ByVal hemf As

Long, ByVal cbBuffer As Long, lpbBuffer As Byte) As Long
说明
将指定的增强型图元文件复制到一个内存缓冲区里。这个函数可用于取得一个图元文件的原始数据，以便将其保存到磁盘文件中
返回值
Long，如 lpbBuffer 为零，则必须设置缓冲区的长度（用 ByVal As Long 在这种情况下传递一个 NULL 参数）。如执行成功，返回缓冲区实际载入的字节数；零表示执行失败
参数表
参数类型及说明
hemfLong，指向一个增强型图元文件的句柄
cbBufferLong，lpbBuffer 缓冲区的长度
lpbBufferByte，长度为 cbBuffer 的一个缓冲区的头一个字节。可用一个别名，将这个参数定义成 ByVal As Long，以便传递一个内存地址
注解
调用了这个函数后，图元文件句柄 hemf 仍然有效

Top
GetEnhMetaFileDescription
GetEnhMetaFileDescription

VB 声明
Declare Function GetEnhMetaFileDescription Lib "gdi32" Alias "GetEnhMetaFileDescriptionA" (ByVal hemf As Long, ByVal cchBuffer As Long, ByVal lpszDescription As String) As Long
说明
返回对一个增强型图元文件的说明
返回值
Long，如 lpszDescription 为 NULL（使用 vbNullString），则必须设置缓冲区的长度。如函数执行成功，返回实际载入缓冲区的字节数；如不存在说明信息则返回零
参数表
参数类型及说明
hemfLong，目标增强型图元文件的句柄
cchBufferLong，lpszDescription 缓冲区的长度
lpszDescriptionString，指定一个预先初始化好的字串缓冲区，准备随同图元文件说明载入。参考 CreateEnhMetaFile 函数，了解增强型图元文件说明字串的具体格式

Top
GetEnhMetaFileHeader
GetEnhMetaFileHeader

VB 声明
Declare Function GetEnhMetaFileHeader Lib "gdi32" Alias "GetEnhMetaFileHeader" (ByVal hemf As Long, ByVal cbBuffer As Long, lpemh As ENHMETAHEADER) As Long
说明
取得增强型图元文件的图元文件头

返回值
Long，若 lpemh 为零（用 ByVal As Long 在这种情况下传递一个 NULL 参数）则必须设置缓冲区的长度。如执行成功，返回缓冲区中实际载入的字节数；零表示失败
参数表
参数类型及说明
hemfLong，指向一个增强型图元文件的句柄
cbBufferLong，ENHMETAHEADER 结构的大小
lpemhENHMETAHEADER

Top
GetEnhMetaFilePaletteEntries
GetEnhMetaFilePaletteEntries

VB 声明
Declare Function GetEnhMetaFilePaletteEntries Lib "gdi32" Alias "GetEnhMetaFilePaletteEntries" (ByVal hemf As Long, ByVal cEntries As Long, lppe As PALETTEENTRY) As Long
说明
取得增强型图元文件的全部或部分调色板
返回值
Long，如 lppe 为零（用 ByVal As Long 在这种情况下传递一个 NULL 参数）则必须设置缓冲区的长度。如执行成功，返回缓冲区实际载入的条目数量；如图元文件中不存在调色板就返回一个零值。如返回 GDI_ERROR，表示函数执行出错
参数表
参数类型及说明
hemfLong，增强型图元文件的句柄
cEntriesLong，欲取回的条目数量
lppePALETTEENTRY，一个 PALETTEENTRY 结构的数组，用于装载增强型图元文件的调色板条目。注意至少要包括 cEntries 个结构

Top
GetMetaFile
GetMetaFile

VB 声明
Declare Function GetMetaFile Lib "gdi32" Alias "GetMetaFileA" (ByVal lpFileName As String) As Long
说明
取得包含在一个磁盘文件中的图元文件的图元文件句柄
返回值
Long，指向已载入的图元文件的一个句柄，零表示错误
参数表
参数类型及说明
lpFileNameString，包含了一个图元文件的磁盘文件的名称

注解
这个函数在 vb 里没有什么用。因为 windows 图元文件通常保存为可放置的图元文件格式，而该函数不能识别这种格式

Top
GetMetaFileBitsEx
GetMetaFileBitsEx

VB 声明
Declare Function GetMetaFileBitsEx Lib "gdi32" Alias "GetMetaFileBitsEx" (ByVal hMF As Long, ByVal nSize As Long, lpvData As Any) As Long
说明
将指定的图元文件复制到一个内存缓冲区。这个函数可用于取得一个图元文件的原始数据，以便转存到磁盘文件
返回值
Long，如 lpvData 为零（在这种情况下用 ByVal As Long 传递一个 NULL 参数），那么必须指定缓冲区的长度。如执行成功，返回实际载入缓冲区的字节数，零表示出错
参数表
参数类型及说明
hMFLong，标准 windows 图元文件的句柄
nSizeLong，lpvData 缓冲区的长度
lpvData 任何类型，一个字节数组中的第一个条目，该数组用于装载图元文件数据。用 ByVal As Long 可指定空值或缓冲区的内存地址

Top
GetMiterLimit
GetMiterLimit

VB 声明
Declare Function GetMiterLimit& Lib "gdi32" (ByVal hdc As Long, peLimit As Single)
说明
取得设备场景的斜率限制（Miter）设置——斜率限制是指斜角长度与线宽间的比率
返回值
Long，
参数表
参数类型及说明
hdcLong，设备场景的句柄
peLimitSingle，随同当前斜率设置载入一个 Single 值
注解
开头的声明来自于我的资料，而在 vb 的 api 文本查看器里的声明如下：Declare Function GetMiterLimit Lib "gdi32" Alias "GetMiterLimit" (ByVal hdc As Long, peLimit As Double) As Long，参数 peLimit 的类型不同，不知是谁错了。

Top

GetNearestColor
GetNearestColor

VB 声明
Declare Function GetNearestColor Lib "gdi32" Alias "GetNearestColor" (ByVal hdc As Long, ByVal crColor As Long) As Long
说明
根据设备的显示能力，取得与指定颜色最接近的一种纯色
返回值
Long，取得与指定颜色最接近的一种颜色，这种颜色可由设备场景实际显示出来（给定当前系统调色板）。如返回 CLR_INVALID，表示函数执行出错。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
crColorLong，欲测试的 RGB 颜色

Top
GetObjectAPI
GetObjectAPI

VB 声明
Declare Function GetObjectAPI& Lib "gdi32" Alias "GetObjectA" (ByVal hObject As Long, ByVal nCount As Long, lpObject As Any)
说明
取得对指定对象进行说明的一个结构。windows 手册建议用 GetObject 这个名字来引用该函数。GetObjectAPI 在 vb 中用于避免与 GetObject 关键字混淆
返回值
Long，如 lpObject 设为零（用 ByVal As Long 在这种情况下传递一个 NULL 参数），则必须设置缓冲区的长度。如执行成功，返回载入结构内部的实际字节数；如失败，返回零值
参数表
参数类型及说明
hObjectLong，画笔、刷子、字体、位图或调色板等对象的句柄
nCountLong，欲取回的字节数。通常是由 lpObject 定义的那个结构的长度
lpObject 任何类型，用于容纳对象数据的结构。针对画笔，通常是一个 LOGPEN 结构；针对扩展画笔，通常是 EXTLOGPEN；针对字体是 LOGBRUSH；针对位图是 BITMAP；针对 DIBSection 位图是 DIBSECTION；针对调色板，应指向一个整型变量，代表调色板中的条目数量

Top
GetObjectType
GetObjectType

VB 声明
Declare Function GetObjectType Lib "gdi32" Alias "GetObjectType" (ByVal hgdiobj As Long) As

Long
说明
判断由指定句柄引用的 GDI 对象的类型
返回值
Long，带有 OBJ_前缀的一个常数，标志着一种特定的对象。零表示错误
参数表
参数类型及说明
hgdiobjLong，一个 GDI 对象句柄

Top
GetPath
GetPath

VB 声明
Declare Function GetPath Lib "gdi32" Alias "GetPath" (ByVal hdc As Long, lpPoint As POINTAPI, lpTypes As Byte, ByVal nSize As Long) As Long
说明
取得对当前路径进行定义的一系列数据
返回值
Long，载入数组的点数（如 nSize 设为零，则返回要求的条目数量）。如数组空间不够大，不足以容下所有的点，就返回 -1。会将 GetLastError 设置为下述值之一：ERROR_CAN_NOT_COMPLETE，ERROR_INVALID_PARAMETER，ERROR_BUFFER_OVERFLOW
参数表
参数类型及说明
hdcLong，包含了路径的设备场景
lpPointPOINTAPI，一个 POINTAPI 结构数组中的第一个元素。这个数组为路径中的每个段（segment）都要载入坐标数据。具体的信息是采用逻辑坐标提供的
lpTypesByte，一个字节数组中的第一个元素；这个数组定义了与每个坐标对应的操作类型。其中包括：
PT_MOVETO 坐标是一个新子路径的起始处
PT_LINETO 坐标是来自前一个坐标的一条线的终点
PT_BEZIERTO 肯定以三点一组的形式出现。头两个点是控制点，第三个是贝塞尔（Bezier）曲线的终点。PT_LINETO 和 PT_BEZIERTO 也许能与 PT_CLOSEFIGURE 联合使用。在这种情况下，它代表一幅图象的最后一个点。将这个点与子路径的第一个连接起来后，路径就会封闭
nSizeLong，lpPoint 和 lpTypes 数组的大小。如设为零，表示取得要求的数组大小
注解
尽管路径信息是在设备坐标的内部保存的，这个函数的所有坐标都是用逻辑坐标返回的。具体坐标取决于当前的坐标系统及转换设置。可用 FlattenPath 函数强迫路径中的所有点都成为 PT_MOVETO 和 PT_LINETO 类型

Top
GetPixel

GetPixel

VB 声明
Declare Function GetPixel Lib "gdi32" Alias "GetPixel" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long) As Long
说明
在指定的设备场景中取得一个像素的 RGB 值
返回值
Long，指定点的 RGB 颜色。如指定的点位于设备场景的剪切区之外，则返回 CLR_INVALID
参数表
参数类型及说明
hdcLong，一个设备场景的句柄
x,yLong，逻辑坐标中要检查的点
注解
用 GetDeviceCaps 判断设备是否支持本函数

Top
GetPolyFillMode
GetPolyFillMode

VB 声明
Declare Function GetPolyFillMode Lib "gdi32" Alias "GetPolyFillMode" (ByVal hdc As Long) As Long
说明
针对指定的设备场景，获得多边形填充模式。关于填充模式见注解
返回值
Long，常数 ALTERNATE 或 WINDING。零表示失败
参数表
参数类型及说明
hdcLong，设备场景的句柄
注解
填充模式 1——ALTERNATE：为判断一个点是否位于填充区，windows 会从这个点到图形外部画一条假想的线。每与一条线相交，计数器就会增 1。如最后一个记数是奇数，则填充这个点；如果是偶数，则保留原样不变
填充模式 2——WINDING：为判断一个点是否位于填充区，windows 会从这个点到图形外部画一条假想的线。windows 会跟踪画出每个顶点（线段）的方向。这条假想的线每次穿过一个顶点时，而且顶点的 Y 方向为正，则减一个记数。如结果记数不是零，就表明该点位于填充区域

Top
GetROP2
GetROP2

VB 声明

Declare Function GetROP2 Lib "gdi32" Alias "GetROP2" (ByVal hdc As Long) As Long
说明
针对指定的设备场景，取得当前的绘图模式。这样可定义绘图操作如何与正在显示的图象合并起来
返回值
Long，请参考绘图模式常数表
参数表
参数类型及说明
hdcLong，设备场景的句柄
注解
这个函数只对光栅设备有效

绘图模式常数表
常数 DrawMode 像素值
R2_BLACKvbBlackness 黑色
R2_WHITEvbWhitness 白色
R2_NOPvbNop 不变
R2_NOTvbInvert 当前显示颜色的反色
R2_COPYPENvbCopyPen 画笔颜色
R2_NOTCOPYPENvbNotCopyPenR2_COPYPEN 的反色
R2_MERGEPENNOTvbMergePenNot 显示颜色的反色与画笔颜色进行 OR 运算
R2_MASKPENNOTvbMaskPenNot 显示颜色的反色与画笔颜色进行 AND 运算
R2_MERGENOTPENvbMergeNotPen 画笔颜色的反色与显示颜色进行 OR 运算
R2_MASKNOTPENvbMaskNotPen 画笔颜色的反色与显示颜色进行 AND 运算
R2_MERGEPENvbMergePen 画笔颜色与显示颜色进行 OR 运算
R2_NOTMERGEPENvbNotMergePenR2_MERGEPEN 的反色
R2_MASKPENvbMaskPen 显示颜色与画笔颜色进行 AND 运算
R2_NOTMASKPENvbNotMaskPenR2_MASKPEN 的反色
R2_XORPENvbXorPen 显示颜色与画笔颜色进行异或运算
R2_NOTXORPENvbNotXorPenR2_XORPEN 的反色

Top
GetStockObject
GetStockObject

VB 声明
Declare Function GetStockObject Lib "gdi32" Alias "GetStockObject" (ByVal nIndex As Long) As Long
说明
取得一个固有对象（Stock）。这是可由任何应用程序使用的 windows 标准对象之一
返回值
Long，指向指定对象的一个句柄。零表示出错
参数表
参数类型及说明

nIndexLong，下述表格中定义的任何常数之一
BLACK_BRUSH 黑色刷子 DKGRAY_BRUSH 黑灰色刷子
GRAY_BRUSH 灰色刷子 HOLLOW_BRUSH 凹刷子
LTGRAY_BRUSH 浅灰色刷子 NULL_BRUSH 空刷子
WHITE_BRUSH 白色刷子 BLACK_PEN 黑色画笔
NULL_PEN 空画笔 WHITE_PEN 白色画笔
ANSI_FIXED_FONT 采用 windows（ANSI）字符集的等宽字体 ANSI_VAR_FONT 采用 windows（ANSI）字符集的不等宽字体
DEVICE_DEFAULT_FONT 设备使用的默认字体（NT）DEFAULT_GUI_FONT 用户界面的默认字体，包括菜单和对话框字体（Windows 95）
OEM_FIXED_FONTOEM 字符集的固有字体 SYSTEM_FONT 屏幕系统字体。这是用于菜单、对话框等等的默认不等宽字体
SYSTEM_FIXED_FONT 屏幕系统字体。这是用于菜单、对话框等等的默认等宽字体（在 windows 3.0 之前使用）DEFAULT_PALETTE 默认调色板
注解
固有刷子的起点可能不会改变。不应用 DeleteObject 函数删除这些对象。不要对那些不具备 CS_HREDRAW 和 CS_VREDRAW 类样式的窗口使用 DK_GRAY_BRUSH，GRAY_BRUSH 和 LTGRAY_BRUSH 刷子

Top
GetSysColorBrush
GetSysColorBrush

VB 声明
Declare Function GetSysColorBrush Lib "user32" Alias "GetSysColorBrush" (ByVal nIndex As Long) As Long
说明
为任何一种标准系统颜色取得一个刷子
返回值
Long，针对一种系统颜色的一个固有刷子的句柄。零表示出错
参数表
参数类型及说明
nIndexLong，系统颜色索引，也即带有 COLOR_前缀的某个常数。参考 GetSysColor
注解
不要用 DeleteObject 函数删除这些刷子。它们是由系统拥有的固有对象。不要将这些刷子指定成一种窗口类的默认刷子

Top
GetWinMetaFileBits
GetWinMetaFileBits

VB 声明
Declare Function GetWinMetaFileBits Lib "gdi32" Alias "GetWinMetaFileBits" (ByVal hemf As Long, ByVal cbBuffer As Long, lpbBuffer As Byte, ByVal fnMapMode As Long, ByVal hdcRef

As Long) As Long
说明
通过在一个缓冲区中填充用于标准图元文件的数据，将一个增强型图元文件转换成标准 windows 图元文件
返回值
Long，以字节数表示的图元文件长度。如 lpbBuffer 为 NULL（在这种情况下用一个别名指定 ByVal As Long，从而传递一个 NULL 值）——返回字节数组的长度。零表示出错（原文: The size in bytes of the metafile. If lpbBuffer is null (use an alias with the parameter specified ByVal as Long to pass null to this function)-returns the required size of the byte array. Zero on error.）
参数表
参数类型及说明
hemfLong，欲转换的增强型图元文件的句柄。函数调用完毕后，该句柄仍然保持有效
cbBufferLong，目标缓冲区的长度
lpbBufferByte，作为目标缓冲区使用的一个字节数组的第一个字节。这个数组的长度至少为 cbBuffer 个字节
fnMapModeLong，转换时采用的映射（对应）模型。通常用 MM_ANISOTROPIC 创建一个可扩展的图元文件
hdcRefLong，一个参考设备场景，用于决定新图元文件采用的参考分辨率
注解
有些增强型图元文件命令没有对应的标准图元文件命令。这些命令会转换成最接近的命令，或者丢弃。结果生成的图元文件已指定了窗口的显示范围。窗口的起点是 0，0

Top
InvertRect
InvertRect

VB 声明
Declare Function InvertRect Lib "user32" Alias "InvertRect" (ByVal hdc As Long, lpRect As RECT) As Long
说明
通过反转每个像素的值，从而反转一个设备场景中指定的矩形
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
lpRectRECT，要反转的矩形，用逻辑坐标指定
注解
反转是一种光栅操作

Top
LineDDA
LineDDA

VB 声明
Declare Function LineDDA Lib "gdi32" Alias "LineDDA" (ByVal n1 As Long, ByVal n2 As Long, ByVal n3 As Long, ByVal n4 As Long, ByVal lpLineDDAProc As Long, ByVal lParam As Long) As Long
说明
枚举指定线段中的所有点
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
n1,n2Long，线段的 x,y 起点坐标
n3,n4Long，线段的 x,y 终点坐标
lpLineDDAProcLong，vb5 中的一个函数地址
lParamLong，枚举过程中传递给回调函数的用户自定义值
注解
通常用这个函数执行自定义的线段作图——例如，可将一条线中的其他每个像素都设成不同的颜色。在 MM_TEXT 模式下，每个点都对应于设备中的一个像素——在这种模式下，也可用这个函数进行线段中的击中测试。线段中的最后一个点不会枚举出来

Top
LineTo
LineTo

VB 声明
Declare Function LineTo Lib "gdi32" Alias "LineTo" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long) As Long
说明
用当前画笔画一条线，从当前位置连到一个指定的点。这个函数调用完毕，当前位置变成 x,y 点
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hdcLong，设备场景的句柄
x,yLong，线段终点位置，采用逻辑坐标表示。这个点不会实际画出来；它不属于线段的一部分
注解
如重复调用这个函数和一个几何画笔，从而创建一系列线段，那么除非在一个路径的场景中调用，否则不会认为这些线段已结合到一起

Top
MoveToEx
MoveToEx

VB 声明
Declare Function MoveToEx Lib "gdi32" Alias "MoveToEx" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long, lpPoint As POINTAPI) As Long
说明
为指定的设备场景指定一个新的当前画笔位置。前一个位置保存在 lpPoint 中
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hdcLong，指向一个设备场景的句柄
x,yLong，采用逻辑坐标表示的新画笔位置
lpPointPOINTAPI，用于保存前一个画笔位置。可以为 NULL（将参数改为 ByVal As Long，以传递一个空参数）
注解
在一个路径分支中描绘的时候，这个函数会创建一个新的子路径

Top
PaintDesktop
PaintDesktop

VB 声明
Declare Function PaintDesktop Lib "user32" Alias "PaintDesktop" (ByVal hdc As Long) As Long
说明
在指定的设备场景中描绘桌面墙纸图案
返回值
Long，TRUE（非零）表示成功，否则返回零
参数表
参数类型及说明
hdcLong，要在其中填充的设备场景

Top
PathToRegion
PathToRegion

VB 声明
Declare Function PathToRegion Lib "gdi32" Alias "PathToRegion" (ByVal hdc As Long) As Long
说明
将当前选定的路径转换到一个区域里
返回值
Long，新区域的句柄。零表示错误。会将 GetLastError 设为下述值之一：ERROR_CAN_NOT_COMPLETE，ERROR_INVALID_PARAMETER，ERROR_NOT_ENOUGH_MEMORY
参数表

参数类型及说明
hdcLong，包含了欲转换的路径的设备场景
注解
函数执行完毕后，路径会自动清除

Top
Pie
Pie

VB 声明
Declare Function Pie Lib "gdi32" Alias "Pie" (ByVal hdc As Long, ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long, ByVal X3 As Long, ByVal Y3 As Long, ByVal X4 As Long, ByVal Y4 As Long) As Long
说明
象注解里那样画一个饼图，X1，Y1，X2，Y2 指定椭圆的一个约束矩形。从矩形的中心分别向 X3，Y3 和 X4，Y4 画一条线，这两条线与椭圆的交点定义了饼图占据椭圆的面积
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，指定一个显示场景的句柄
X1,Y1Long，指定椭圆约束矩形的左上角位置
X2,Y2Long，指定椭圆约束矩形的右下角位置
X3,Y3Long，指定饼图的一个斜边
X4,Y4Long，指定饼图的另一个斜边
注解
在 win95 和 win16 中，约束矩形的宽度和高度必须在 3-32766 之间
参考 Arc 函数

Top
PlayEnhMetaFile
PlayEnhMetaFile

VB 声明
Declare Function PlayEnhMetaFile Lib "gdi32" Alias "PlayEnhMetaFile" (ByVal hdc As Long, ByVal hemf As Long, lpRect As RECT) As Long
说明
在指定的设备场景中画一个增强型图元文件。与标准图元文件不同，完成回放后，增强型图元文件会恢复设备场景以前的状态
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，目标设备场景

hemfLong，欲描绘的增强型图元文件的句柄
lpRectRECT，一个约束矩形，定义了在哪里描绘图元文件

Top
PlayEnhMetaFileRecord
PlayEnhMetaFileRecord

VB 声明
Declare Function PlayEnhMetaFileRecord Lib "gdi32" Alias "PlayEnhMetaFileRecord" (ByVal hdc As Long, lpHandletable As HANDLETABLE, lpEnhMetaRecord As ENHMETARECORD, ByVal nHandles As Long) As Long
说明
回放单独一条增强型图元文件记录。可与 EnumEnhMetaFile 函数联合使用，只回放选定的图元文件记录。这个函数的参数设置与那些由 EnumMetaFile 回调函数返回的值是相似的
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hdcLong，欲在其中绘图的设备场景
lpHandletableHANDLETABLE，使用的句柄的一个数组（An array of handles used.）
lpEnhMetaRecordENHMETARECORD，一个特定的结构（或指向结构的指针），包含了增强型图元文件记录
nHandlesLong，句柄表中的句柄数量

Top
PlayMetaFile
PlayMetaFile

VB 声明
Declare Function PlayMetaFile Lib "gdi32" Alias "PlayMetaFile" (ByVal hdc As Long, ByVal hMF As Long) As Long
说明
在指定的设备场景中回放一个图元文件。图元文件中记录的 GDI 操作会针对设备场景而执行
返回值
Long，
参数表
参数类型及说明
hdcLong，要在其中回放图元文件的一个设备场景的句柄
hMFLong，欲回放的一个图元文件的句柄
注解
在 vb 里，图元文件有能力改变一个设备场景的对象及映射模式。注意在调用这个函数之前，必须保存一个 vb 窗体或图片控件设备场景的状态

Top
PlayMetaFileRecord
PlayMetaFileRecord

VB 声明
Declare Function PlayMetaFileRecord Lib "gdi32" Alias "PlayMetaFileRecord" (ByVal hdc As Long, lpHandletable As HANDLETABLE, lpMetaRecord As METARECORD, ByVal nHandles As Long) As Long
说明
回放来自图元文件的单条记录（每条记录都包含了单个 GDI 绘图命令）。可与 EnumMetaFile 函数联合使用，只回放那些选定的图元文件记录。这个函数的参数与那些由 EnumMetaFile 回调函数返回的值相似
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hdcLong，欲在其中回放图元文件记录 GDI 命令的一个设备场景的句柄
lpHandletableHANDLETABLE，一个整数数组的第一个条目，该数组用于容纳图元文件使用的 GDI 对象的句柄
lpMetaRecordMETARECORD，指定单条图元文件记录
nHandlesLong，图元文件句柄表格中的句柄数目

Top
PolyBezier
PolyBezier, PolyBezierTo

VB 声明
Declare Function PolyBezier& Lib "gdi32" (ByVal hdc As Long, lppt As POINTAPI, ByVal cPoints As Long)
Declare Function PolyBezierTo& Lib "gdi32" (ByVal hdc As Long, lppt As POINTAPI, ByVal cCount As Long)
说明
描绘一条或多条贝塞尔（Bezier）曲线。PolyBezierTo 用于将当前画笔位置设为前一条曲线的终点
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hdcLong，要在其中绘图的设备场景
lpptPOINTAPI，指定一个 POINTAPI 结构数组。其中的第一个结构指定了起点。剩下的点三个一组——包括两个控件点和一个终点
原文：An array of POINTAPI structures. The first structure specifies the starting point. The remaining points are in groups of three, consisting of two control points and an end point.
cPointsLong，lppt 数组的总点数

Top
PolyDraw
PolyDraw

VB 声明
Declare Function PolyDraw Lib "gdi32" Alias "PolyDraw" (ByVal hdc As Long, lppt As POINTAPI, lpbTypes As Byte, ByVal cCount As Long) As Long
说明
描绘一条复杂的曲线，由线段及贝塞尔曲线组成
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hdcLong，用于绘图的设备场景
lpptPOINTAPI，POINTAPI 结构数组的第一个元素。这个数组用于为描绘的每一段都载入坐标数据。这些信息是用逻辑坐标提供的
lpbTypesByte，一个字节数组的第一个元素。这个数组定义了与每个坐标对应的操作类型。其中包括：
PT_MOVETO 坐标是一幅新打开图形的起点
PT_LINETO 坐标是来自前一个坐标的一条线的终点
PT_BEZIERTO 以三点一组的形式出现。头两个点是控制点，第三个是贝塞尔曲线的终点。
PT_LINETO 和 PT_BEZIERTO 也许能与 PT_CLOSEFIGURE 联合使用。在这种情况下，它代表一幅图形的最后一个点。将这个点与图形的第一个点连接起来后，图形就会封闭
cCountLong，lppt 和 lpbTypes 数组的大小，设为零表示取得需要的数组大小
原文：The size of the lpPoint and lpTypes array. Set to zero to retrieve the required array size.
注解
当前的画笔位置设为最后一条线段或 lppt 数组中的曲线的终点

Top
Polygon
Polygon

VB 声明
Declare Function Polygon Lib "gdi32" Alias "Polygon" (ByVal hdc As Long, lpPoint As POINTAPI, ByVal nCount As Long) As Long
说明
描绘一个多边形，由两点或三点的任意系列构成。windows 会将最后一个点与第一个点连接起来，从而封闭多边形。多边形的边框用当前选定的画笔描绘，多边形用当前选定的刷子填充
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明

hdcLong，用于描绘的设备场景
lpPointPOINTAPI，在 nCount POINTAPI 结构数组中的第一个 POINTAPI 结构
nCountLong，多边形的总点数（顶点数）
注解
GetPolyFillMode 和 SetPolyFillMode 函数决定了如何在多边形内部填充

Top
Polyline
Polyline, PolyLineTo

VB 声明
Declare Function Polyline Lib "gdi32" Alias "Polyline" (ByVal hdc As Long, lpPoint As POINTAPI, ByVal nCount As Long) As Long
Declare Function PolylineTo Lib "gdi32" Alias "PolylineTo" (ByVal hdc As Long, lppt As POINTAPI, ByVal cCount As Long) As Long
说明
用当前画笔描绘一系列线段。使用 PolylineTo 函数时，当前位置会设为最后一条线段的终点。它不会由 Polyline 函数改动
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hdcLong，要在其中绘图的设备场景
lpPointPOINTAPI，nCount POINTAPI 结构数组中的第一个 POINTAPI 结构
nCountLong，lpPoint 数组中的点数。会从第一个点到第二个点画一条线，以次类推

Top
PolyPolygon
PolyPolygon

VB 声明
Declare Function PolyPolygon Lib "gdi32" Alias "PolyPolygon" (ByVal hdc As Long, lpPoint As POINTAPI, lpPolyCounts As Long, ByVal nCount As Long) As Long
说明
用当前选定画笔描绘两个或多个多边形。根据由 SetPolyFillMode 函数指定的多边形填充模式，用当前选定的刷子填充它们。每个多边形都必须是封闭的
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，要在其中绘图的设备场景
lpPointPOINTAPI，nCount POINTAPI 结构数组中的第一个 POINTAPI 结构
lpPolyCountsLong，在 Long 值数组中的第一个条目。每个条目都包含了一定数量的点，用于构成一个封闭的多边形。lpPoint 数组由一系列封闭的多边形构成，每个在 lpPolyCounts

数组中都有一个条目
nCountLong，要描绘的多边形总数（就是 lpPolyCounts 数组的大小）。至少为 2

Top
PolyPolyline
PolyPolyline

VB 声明
Declare Function PolyPolyline Lib "gdi32" Alias "PolyPolyline" (ByVal hdc As Long, lppt As POINTAPI, lpdwPolyPoints As Long, ByVal cCount As Long) As Long
说明
用当前选定画笔描绘两个或多个多边形
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hdcLong，要在其中绘图的设备场景
lpPointPOINTAPI，nCount POINTAPI 结构数组中的第一个 POINTAPI 结构
lpdwPolyPointsLong，在 Long 值数组中的第一个条目。每个条目都包含了构成一个多边形的点数。lpPoint 数组由一系列多边形构成，每个在 lpdwPolyPoints 数组中都有一个条目
cCountLong，要描绘的多边形总数（就是 lpdwPolyPoints 数组的大小）
注解
这个函数几乎与 PolyPolygon 函数完全一致，只是多边形不会填充，也不需要封闭

Top
Rectangle
Rectangle

VB 声明
Declare Function Rectangle Lib "gdi32" Alias "Rectangle" (ByVal hdc As Long, ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long) As Long
说明
用当前选定的画笔描绘矩形，并用当前选定的刷子进行填充
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
X1,Y1Long，指定矩形左上角位置
X2,Y2Long，指定矩形右下角位置

Top
RoundRect
RoundRect

VB 声明
Declare Function RoundRect Lib "gdi32" Alias "RoundRect" (ByVal hdc As Long, ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long, ByVal X3 As Long, ByVal Y3 As Long) As Long
说明
用当前选定的画笔画一个圆角矩形，并用当前选定的刷子在其中填充。X3 和 Y3 定义了用于生成圆角的椭圆
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，用于绘图的设备场景
X1,Y1Long，对矩形左上角位置进行说明的 X，Y 坐标
X2,Y2Long，对矩形右下角位置进行说明的 X，Y 坐标
X3Long，用于生成圆角效果的一个椭圆的宽度。取值范围从零（表示不加圆角），一直到矩形的宽度（全圆）
Y3Long，用于生成圆角效果的一个椭圆的高度。取值范围从零（表示不加圆角），一直到矩形的高度（全圆）

Top
SelectClipPath
SelectClipPath

VB 声明
Declare Function SelectClipPath Lib "gdi32" Alias "SelectClipPath" (ByVal hdc As Long, ByVal iMode As Long) As Long
说明
将设备场景当前的路径合并到剪切区域里
返回值
Long，非零表示成功，零表示失败。会将 GetLastError 设置为下述值之一：ERROR_CAN_NOT_COMPLETE，ERROR_INVALID_PARAMETER，ERROR_NOT_ENOUGH_MEMORY
参数表
参数类型及说明
hdcLong，包含了欲合并到剪切区域的路径的一个设备场景的句柄
iModeLong，决定如何将路径与当前剪切区域合并到一起。可选下述常数之一：
RGN_AND 新的剪切区域只包含了在路径和当前剪切区域都存在的点
RGN_COPY 新剪切区域设为路径
RGN_DIFF 新剪切区域只包含了当前剪切区域的点，但这些点不在路径中
RGN_OR 新的剪切区域包含了在路径或当前剪切区域中的点
RGN_XOR 新的剪切区域只包含了存在于路径或当前剪切区域中的点，但两地共有的点不包括在内

Top
SelectObject
SelectObject

VB 声明
Declare Function SelectObject Lib "gdi32" Alias "SelectObject" (ByVal hdc As Long, ByVal hObject As Long) As Long
说明
每个设备场景都可能有选入其中的图形对象。其中包括位图、刷子、字体、画笔以及区域等等。一次选入设备场景的只能有一个对象。选定的对象会在设备场景的绘图操作中使用。例如，当前选定的画笔决定了在设备场景中描绘的线段颜色及样式
返回值
Long，与以前选入设备场景的相同 hObject 类型的一个对象的句柄，零表示出错。如选定的对象是一个区域（Region），结果就是下列常数之一：SIMPLEREGION， COMPLEXREGION 或 NULLREGION 对区域进行描述， GDI_ERROR 表示出错
参数表
参数类型及说明
hdcLong，一个设备场景的句柄
hObjectLong，一个画笔、位图、刷子、字体或区域的句柄
注解
返回值通常用于获得选入 DC 的对象的原始值。绘图操作完成后，原始的对象通常选回设备场景。在清除一个设备场景前，务必注意恢复原始的对象

Top
SetArcDirection
SetArcDirection

VB 声明
Declare Function SetArcDirection Lib "gdi32" Alias "SetArcDirection" (ByVal hdc As Long, ByVal ArcDirection As Long) As Long
说明
设置圆弧的描绘方向
返回值
Long，如执行成功，返回原始的圆弧方向；零意味着出错
参数表
参数类型及说明
hdcLong，要设置的设备场景
ArcDirectionLong，AD_CLOCKWISE （顺时针）或 AD_COUNTERCLOCKWISE （逆时针）
注解
可应用于下列函数：Arc， ArcTo， Chord， Ellipse， Pie， Rectangle 和 RoundRect

Top
SetBkColor

SetBkColor

VB 声明
Declare Function SetBkColor Lib "gdi32" Alias "SetBkColor" (ByVal hdc As Long, ByVal crColor As Long) As Long
说明
为指定的设备场景设置背景颜色。背景颜色用于填充阴影刷子、虚线画笔以及字符（如背景模式为 OPAQUE）中的空隙。也在位图颜色转换期间使用。参考 SetBkMode
返回值
Long，前一个背景色，CLR_INVALID 表示出错
参数表
参数类型及说明
hdcLong，设备场景的句柄
crColorLong，新背景颜色的 RGB 颜色值
注解
背景实际是设备能够显示的最接近于 crColor 的颜色

Top
SetBkMode
SetBkMode

VB 声明
Declare Function SetBkMode Lib "gdi32" Alias "SetBkMode" (ByVal hdc As Long, ByVal nBkMode As Long) As Long
说明
指定阴影刷子、虚线画笔以及字符中的空隙的填充方式
返回值
Long，前一个背景模式的值
参数表
参数类型及说明
hdcLong，设备场景的句柄
nBkModeLong，下述常数之一：
OPAQUE 用当前的背景色填充虚线画笔、阴影刷子以及字符的空隙
TRANSPARENT 透明处理，即不作上述填充
注解
背景模式不会影响用扩展画笔描绘的线条

Top
SetBrushOrgEx
SetBrushOrgEx

VB 声明
Declare Function SetBrushOrgEx Lib "gdi32" Alias "SetBrushOrgEx" (ByVal hdc As Long, ByVal nXOrg As Long, ByVal nYOrg As Long, lppt As POINTAPI) As Long

说明
为指定的设备场景设置当前选定刷子的起点
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
nXOrg,nYOrgLong，刷子的新起点，采用设备坐标表示。取值范围在 0-7 之间（不需要更大的值，因为 windows95 下最大的刷子的尺寸是 8×8；而这个函数在 NT 下是不必要的）
lpptPOINTAPI，用于装载前一个刷子的起点
注解
Windows NT 会自动设置刷子的起点，所以不应在 NT 下使用这个函数
在 vb 里使用
注意完成以后一定要将设备场景的刷子起点设为 0，0。既可明确指定坐标，也可用 RestoreDC 函数恢复恢复原始的 DC

Top
SetEnhMetaFileBits
SetEnhMetaFileBits

VB 声明
Declare Function SetEnhMetaFileBits Lib "gdi32" Alias "SetEnhMetaFileBits" (ByVal cbBuffer As Long, lpData As Byte) As Long
说明
用指定内存缓冲区内包含的数据创建一个增强型图元文件。在从磁盘读入原始的图元文件数据后（最开始是由 GetEnhMetaFileBits 函数获得的），通常再用这个函数创建一个图元文件
返回值
Long，如执行成功，返回一个增强型图元文件句柄；否则返回零
参数表
参数类型及说明
cbBufferLong，lpData 数组的长度
lpDataByte，一个字节数组的头一个条目，这个数组内包含了图元文件数据

Top
SetMetaFileBitsEx
SetMetaFileBitsEx

VB 声明
Declare Function SetMetaFileBitsEx Lib "gdi32" Alias "SetMetaFileBitsEx" (ByVal nSize As Long, lpData As Byte) As Long
说明
用包含在指定内存缓冲区内的数据结构创建一个图元文件。在从磁盘读入原始的图元文件数据后（最开始是由 GetMetaFileBitsEx 函数获得的），通常再用这个函数创建一个图元文件
返回值

Long，如执行成功，返回一个标准图元文件的句柄；零意味着失败
参数表
参数类型及说明
nSizeLong，lpData 数组的长度
lpDataByte，一个字节数组的头一个条目，这个数组内包含了图元文件数据

Top
SetMiterLimit
SetMiterLimit

VB 声明
Declare Function SetMiterLimit& Lib "gdi32" (ByVal hdc As Long, ByVal eNewLimit As Single, peOldLimit As Single)
说明
设置设备场景当前的斜率限制
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，欲设置的设备场景
eNewLimitSingle，新的斜率限制
peOldLimitSingle，用于容纳原始斜率限制的一个单精度值
注解
斜率限制是指斜角长度与线宽之间的比率
其他
请看 vb 的 api 文本查看器里的声明：Function SetMiterLimit Lib "gdi32" Alias "SetMiterLimit" (ByVal hdc As Long, ByVal eNewLimit As Double, peOldLimit As Double) As Long

Top
SetPixel
SetPixel

VB 声明
Declare Function SetPixel Lib "gdi32" Alias "SetPixel" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long, ByVal crColor As Long) As Long
说明
在指定的设备场景中设置一个像素的 RGB 值
返回值
Long，指定点的实际 RGB 颜色。如设备不支持指定的准确颜色，则返回的值会与 crColor 有所不同。如指定的点不能设置，则会返回-1（例如，指定的点可能位于设备场景剪切区外面）。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄

x,yLong，要设置的点，用逻辑坐标表示
crColorLong，指定像素的新 RGB 颜色
注解
可用 GetDeviceCaps 判断一个设备是否支持这个函数

Top
SetPixelV
SetPixelV

VB 声明
Declare Function SetPixelV Lib "gdi32" Alias "SetPixelV" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long, ByVal crColor As Long) As Long
说明
在指定的设备场景中设置一个像素的 RGB 值
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，一个设备场景的句柄
x,yLong，要设置的点，用逻辑坐标表示
crColorLong，指定像素的新 RGB 颜色值
注解
这个函数比 SetPixel 快一些，但不会返回设置的实际颜色。可用 GetDeviceCaps 判断设备是否支持这个函数

Top
SetPolyFillMode
SetPolyFillMode

VB 声明
Declare Function SetPolyFillMode Lib "gdi32" Alias "SetPolyFillMode" (ByVal hdc As Long, ByVal nPolyFillMode As Long) As Long
说明
设置多边形的填充模式。参考 GetPolyFillMode 函数的注解
返回值
Long，如执行成功，返回前一种多边形填充模式。零表示出错
参数表
参数类型及说明
hdcLong，设备场景的句柄
nPolyFillModeLong，下述常数之一：
ALTERNATE 交替填充
WINDING 根据绘图方向填充

Top

SetROP2
SetROP2

VB 声明
Declare Function SetROP2 Lib "gdi32" Alias "SetROP2" (ByVal hdc As Long, ByVal nDrawMode As Long) As Long
说明
设置指定设备场景的绘图模式。与 vb 的 DrawMode 属性完全一致
返回值
Long，如执行成功，返回前一个绘图模式；零表示出错
参数表
参数类型及说明
hdcLong，设备场景的句柄
nDrawModeLong，设备场景的新绘图模式。参考 GetROP2 函数的绘图模式常数表
注解
在 vb 里，为一个 vb 窗体或图片控件使用设备场景的时候，这个函数会设置 DrawMode 属性

Top
SetWinMetaFileBits
SetWinMetaFileBits

VB 声明
Declare Function SetWinMetaFileBits Lib "gdi32" Alias "SetWinMetaFileBits" (ByVal cbBuffer As Long, lpbBuffer As Byte, ByVal hdcRef As Long, lpmfp As METAFILEPICT) As Long
说明
将一个标准 Windows 图元文件转换成增强型图元文件
返回值
Long，如执行成功，返回一个增强型图元文件（位于内存中）的句柄；零意味着出错
参数表
参数类型及说明
cbBufferLong，lpbBuffer 数组的长度
lpbBufferByte，一个字节数组的头一个条目，这个数组包含了标准图元文件数据。数据是用 GetMetaFileBitsEx 或 GetWinMetaFileBits 函数获得的
hdcRefLong，用于决定原始格式及图元文件分辨率的一个参考设备场景。可以为零，表示采用显示器分辨率
lpmfpMETAFILEPICT，定义图元文件附加参考信息的一个结构。可设为 NULL（用一个别名传递 NULL 值，将参数定义成 ByVal As Long）；此时，会假定使用当前显示器的 MM_ANISOTROPIC 映射模式

Top
StrokeAndFillPath
StrokeAndFillPath

VB 声明

Declare Function StrokeAndFillPath Lib "gdi32" Alias "StrokeAndFillPath" (ByVal hdc As Long) As Long
说明
针对指定的设备场景，关闭路径上打开的所有区域。用当前画笔描绘路径的一个轮廓，并用当前刷子填充路径
返回值
Long，TRUE（非零）表示成功，否则返回零。会将 GetLastError 设置为下述值：ERROR_CAN_NOT_COMPLETE，ERROR_INVALID_PARAMETER，ERROR_NOT_ENOUGH_MEMORY
参数表
参数类型及说明
hdcLong，包含了要描绘和填充的那个路径的一个设备场景
注解
函数执行完后记住清除路径

Top
StrokePath
StrokePath

VB 声明
Declare Function StrokePath Lib "gdi32" Alias "StrokePath" (ByVal hdc As Long) As Long
说明
用当前画笔描绘一个路径的轮廓。打开的图形不会被这个函数关闭
返回值
Long，TRUE（非零）表示成功，否则返回零。会将 GetLastError 设置为下述值：ERROR_CAN_NOT_COMPLETE，ERROR_INVALID_PARAMETER，ERROR_NOT_ENOUGH_MEMORY
参数表
参数类型及说明
hdcLong，包含了要描绘和填充的那个路径的一个设备场景

Top
UnrealizeObject
UnrealizeObject

VB 声明
Declare Function UnrealizeObject Lib "gdi32" Alias "UnrealizeObject" (ByVal hObject As Long) As Long
说明
将一个刷子对象选入设备场景之前，如刷子的起点准备用 SetBrushOrgEx 修改，则必须先调用本函数
返回值
Long，TRUE（非零）表示成功，否则返回零。会设置 GetLastError
参数表

参数类型及说明
hObjectLong，刷子或逻辑调色板的句柄
注解
Windows NT 会自动跟踪刷子的起点，所以如果对象是一个刷子，那么这个函数不会产生任何影响。在那个时候，它会返回 TRUE

Top
WidenPath
WidenPath

VB 声明
Declare Function WidenPath Lib "gdi32" Alias "WidenPath" (ByVal hdc As Long) As Long
说明
根据选定画笔的宽度，重新定义当前选定的路径。例如，假设路径描述了一个封闭的矩形。面积为 10×10 像素，而且用一个 3 像素宽的画笔描绘。此时，加宽后的路径将由一个 12×12 的矩形构成
返回值
Long，TRUE（非零）表示成功，否则返回零。会将 GetLastError 设置为下述值：ERROR_CAN_NOT_COMPLETE，ERROR_INVALID_PARAMETER，ERROR_NOT_ENOUGH_MEMORY
参数表
参数类型及说明
hdcLong，包含路径的一个设备的句柄
注解
所有贝塞尔曲线都会由这个函数转换成线段

Top
打印函数
打印函数，共五页。第一页，第二页，第三页，第四页，第五页
AbortDoc 取消一份文档的打印
AbortPrinter 删除与一台打印机关联在一起的缓冲文件
AddForm 为打印机的表单列表添加一个新表单
AddJob 用于获取一个有效的路径名，以便用它为作业创建一个后台打印文件。它也会为作业分配一个作业编号
AddMonitor 为系统添加一个打印机监视器
AddPort 启动“添加端口”对话框，允许用户在系统可用端口列表中加入一个新端口
AddPrinter 在系统中添加一台新打印机
AddPrinterConnection 连接指定的打印机
AddPrinterDriver 为指定的系统添加一个打印驱动程序
AddPrintProcessor 为指定的系统添加一个打印处理器
AddPrintProvidor 为系统添加一个打印供应商
AdvancedDocumentProperties 启动打印机文档设置对话框
ClosePrinter 关闭一个打开的打印机对象
ConfigurePort 针对指定的端口，启动一个端口配置对话框

ConnectToPrinterDlg 启动连接打印机对话框，用它同访问网络的打印机连接
DeleteForm 从打印机可用表单列表中删除一个表单
AbortDoc
AbortDoc

VB 声明
Declare Function AbortDoc Lib "gdi32" Alias "AbortDoc" (ByVal hdc As Long) As Long
说明
取消一份文档的打印。自上次调用 StartDoc 函数以来的所有输出都会被取消。如对打印机进行了配置，令其在正式打印文档之前先在打印缓冲区内对文档进行排队，那么文档的任何一部分都不会打印；否则，就可能出现文档打印到一半被取消的情况
返回值
Long，大于零表示成功，SP_ERROR 表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
在 VB 里使用
如将这个函数用于由打印机对象的 hDC 属性指定的打印机设备场景，那么它可以正常发挥作用。然而，倘若之后调用了 EndDoc 方法，却有可能得到一条打印机出错消息。当大家结合 API 打印函数与 VB 打印机方法的时候，强烈建议对打印机的错误进行跟踪捕获；或干脆避免这种结合

Top
AbortPrinter
AbortPrinter

VB 声明
Declare Function AbortPrinter Lib "winspool.drv" Alias "AbortPrinter" (ByVal hPrinter As Long) As Long
说明
删除与一台打印机关联在一起的缓冲文件
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，一个已打开的打印机对象的句柄（用 OpenPrinter 获得）
注解
如打印机没有使用后台打印缓冲文件，那么该函数将无法发挥作用。例如，后台打印程序可将数据直接发给打印机

Top
AddForm
AddForm

VB 声明
Declare Function AddForm& Lib "spoolss.dll" Alias "AddFormA" (ByVal hPrinter As Long, ByVal Level As Long, pForm As FORM_INFO_1)
说明
为打印机的表单列表添加一个新表单。“表单”描述了一个页面大小及布局，提供了一种与设备无关的机制，可实现 Windows NT 下的纸张尺寸的标准化
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，一个已打开的打印机对象的句柄（用 OpenPrinter 获得）
LevelLong，设为 1
pFormFORM_INFO_1，对表单进行描述的一个结构
适用平台
Windows NT
其他
在 VB 的 API 文本查看器里复制的声明如下：
Declare Function AddForm Lib "winspool.drv" Alias "AddFormA" (ByVal hPrinter As Long, ByVal Level As Long, pForm As Byte) As Long

Top
AddJob
AddJob

VB 声明
Declare Function AddJob Lib "winspool.drv" Alias "AddJobA" (ByVal hPrinter As Long, ByVal Level As Long, pData As Byte, ByVal cdBuf As Long, pcbNeeded As Long) As Long
说明
用于获取一个有效的路径名，以便用它为作业创建一个后台打印文件。它也会为作业分配一个作业编号
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，一个已打开的打印机对象的句柄（用 OpenPrinter 获得）
LevelLong，设为 1
pDataByte，缓冲区会引用一个 ADDJOB_INFO_1 结构
cdBufLong，pData 缓冲区中的字符数量
pcbNeededLong，指向一个 Long 型变量的指针，该变量用于保存请求的缓冲区长度，或者实际读入的字节数量
注解
调用这个函数以后，可创建指定的文件，向其中写入数据，然后用 API 函数 ScheduleJob 令其将数据发给打印机

Top
AddMonitor
AddMonitor

VB 声明
Declare Function AddMonitor Lib "winspool.drv" Alias "AddMonitorA" (ByVal pName As String, ByVal Level As Long, pMonitors As Byte) As Long
说明
为系统添加一个打印机监视器
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，欲在其中安装监视器的一个服务器的名字。对于本地（本机）监视器，请设置成 vbNullString
LevelLong，设为 2
pMonitorsByte，指定一个结构中的第一个字节。那个结构又包含了一个 MONITOR_INFO_2 结构

Top
AddPort
AddPort

VB 声明
Declare Function AddPort Lib "winspool.drv" Alias "AddPortA" (ByVal pName As String, ByVal hwnd As Long, ByVal pMonitorName As String) As Long
说明
启动“添加端口”对话框，允许用户在系统可用端口列表中加入一个新端口
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，欲在其中安装端口的一个服务器的名字。对本地（本机）端口，请设置成 vbNullString
hwndLong，指定 AddPort 对话框的父窗口的句柄
pMonitorNameString，用于指定端口的一个监视器的名称

Top
AddPrinter
AddPrinter

VB 声明
Declare Function AddPrinter Lib "winspool.drv" Alias "AddPrinterA" (ByVal pName As String，ByVal Level As Long， pPrinter As Any) As Long

说明
在系统中添加一台新打印机
返回值
Long，如执行成功，返回一台新打印机的句柄；零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，欲在其中安装打印机的一个服务器的名字。对本地打印机，设为 vbNullString
LevelLong，设为 2
pPrinterAny，指定一个缓冲区的第一个条目。该缓冲区包含了一个 PRINTER_INFO_2 结构。结构中的下述字段会设为有效值：pPrinterName，pPortName，pDriverName，pPrintProcessor 和 pDataType。也可象 PRINTER_INFO_2 那样设置 pPrinter 字段。也可以设置下述字段：Attributes， DefaultPriority， pComment， pDevMode， pLocation， pParameters， Priority，pSecurityDescriptor， pSepFile， pShareName， StartTime 和 UntilTime。而其他字段都应置空
注解
在 NT 下，调用者必须有足够的权限对指定服务器上的打印机进行配置

Top
AddPrinterConnection
AddPrinterConnection

VB 声明
Declare Function AddPrinterConnection Lib "winspool.drv" Alias "AddPrinterConnectionA" (ByVal pName As String) As Long
说明
连接指定的打印机
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，网络上的一台打印机的名字

Top
AddPrinterDriver
AddPrinterDriver

VB 声明
Declare Function AddPrinterDriver Lib "winspool.drv" Alias "AddPrinterDriverA" (ByVal pName As String, ByVal Level As Long, pDriverInfo As Any) As Long
说明
为指定的系统添加一个打印驱动程序
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表

参数类型及说明
pNameString，指定要在其中安装驱动程序的一台服务器的名字。对于本地系统，设为 vbNullString
LevelLong，2 或 3（2 仅适用于 NT 3.51）
pDriverInfoAny，指定一个缓冲区，其中包含了一个 DRIVER_INFO_2 或 DRIVER_INFO_3 结构，它们指定了要添加的驱动程序
注解
在调用这个函数之前，所有驱动程序文件都必须位于适当的目录

Top
AddPrintProcessor
AddPrintProcessor

VB 声明
Declare Function AddPrintProcessor Lib "winspool.drv" Alias "AddPrintProcessorA" (ByVal pName As String, ByVal pEnvironment As String, ByVal pPathName As String, ByVal pPrintProcessorName As String) As Long
说明
为指定的系统添加一个打印处理器
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，指定要在其中安装驱动程序的一台服务器的名字。对于本地系统，设为 vbNullString
pEnvironmentString，要在其中添加打印处理器的一个环境（如“Windows NT x86”）。对于当前（本地）系统环境，则设为 vbNullString
pPathNameString，包含了打印管理器的一个文件的名字。文件必须位于打印处理器目录中
pPrintProcessorNameString，打印处理器的名字

Top
AddPrintProvidor
AddPrintProvidor

VB 声明
Declare Function AddPrintProvidor Lib "winspool.drv" Alias "AddPrintProvidorA" (ByVal pName As String, ByVal Level As Long, pProvidorInfo As Byte) As Long
说明
为系统添加一个打印供应商
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，指出要在其中安装打印供应商的一台服务器的名字。对于本地系统，设为

vbNullString
LevelLong，设为 1
pProvidorInfoByte，包含了一个 PROVIDOR_INFO_1 结构的缓冲区

Top
AdvancedDocumentProperties
AdvancedDocumentProperties

VB 声明
Declare Function AdvancedDocumentProperties Lib "winspool.drv" Alias "AdvancedDocumentPropertiesA" (ByVal hwnd As Long, ByVal hPrinter As Long, ByVal pDeviceName As String, pDevModeOutput As DEVMODE, pDevModeInput As DEVMODE) As Long
说明
启动打印机文档设置对话框。这个函数几乎完全等价于调用 DocumentProperties 函数，同时将 fMode 设为 DM_IN_PROMPT。请参考对 DocumentProperties 函数的说明，了解这个函数的详细情况
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
注解
将 pDevModeOutput 设为 0 后可得到要求的 DEVMODE 结构的大小

Top
ClosePrinter
ClosePrinter

VB 声明
Declare Function ClosePrinter Lib "winspool.drv" Alias "ClosePrinter" (ByVal hPrinter As Long) As Long
说明
关闭一个打开的打印机对象
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，一个打开的打印机对象的句柄

Top
ConfigurePort
ConfigurePort

VB 声明
Declare Function ConfigurePort Lib "winspool.drv" Alias "ConfigurePortA" (ByVal pName As String, ByVal hwnd As Long, ByVal pPortName As String) As Long

说明
针对指定的端口，启动一个端口配置对话框
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，欲对其端口进行配置的一台服务器的名字。对于本地系统，请设为 vbNullString
hwndLong，对话框父窗口的句柄
pPortNameString，端口名

Top
ConnectToPrinterDlg
ConnectToPrinterDlg

VB 声明
Declare Function ConnectToPrinterDlg Lib "winspool.drv" Alias "ConnectToPrinterDlg" (ByVal hwnd As Long, ByVal flags As Long) As Long
说明
启动连接打印机对话框，用它同访问网络的打印机连接
返回值
Long，已连接或选择的打印机的句柄，零意味着失败或用户取消了操作
参数表
参数类型及说明
hwndLong，对话框的父窗口句柄
flagsLong，保留，设为零

Top
DeleteForm
DeleteForm

VB 声明
Declare Function DeleteForm Lib "winspool.drv" Alias "DeleteFormA" (ByVal hPrinter As Long, ByVal pFormName As String) As Long
说明
从打印机可用表单列表中删除一个表单
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，一个打开的打印机的名字（用 OpenPrinter 获得）
pFormNameString，欲删除的表单的名字
适用平台
Windows NT
注解

不能删除内建表单。请参考 AddForm 函数了解进一步的情况

Top
DeleteMonitor
DeleteMonitor

VB 声明
Declare Function DeleteMonitor Lib "winspool.drv" Alias "DeleteMonitorA" (ByVal pName As String, ByVal pEnvironment As String, ByVal pMonitorName As String) As Long
说明
删除指定的打印监视器
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，监视器所在的服务器的名字。用 vbNullString 指定本地系统
pEnvironmentString，欲在其中删除监视器的环境。用 vbNullString 指定当前系统
pMonitorNameString，欲删除监视器的名字
注解
参考 AddMonitor 函数，了解进一步的信息

Top
DeletePort
DeletePort

VB 声明
Declare Function DeletePort Lib "winspool.drv" Alias "DeletePortA" (ByVal pName As String, ByVal hwnd As Long, ByVal pPortName As String) As Long
说明
启动“删除端口”对话框，允许用户从当前系统删除一个端口
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，欲在其中删除端口的服务器的名字。vbNullString 表示使用本地端口
hwndLong，DeletePort 对话框的父窗口的句柄
pPortNameString，欲删除的端口的名字
注解
如打印机已连接到端口，则函数执行会失败

Top
DeletePrinter
DeletePrinter

VB 声明
Declare Function DeletePrinter Lib "winspool.drv" Alias "DeletePrinter" (ByVal hPrinter As Long) As Long
说明
将指定的打印机标志为从系统中删除
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，已打开的打印机对象的句柄
注解
除非所有待决的打印作业都已完成，否则打印机不会实际删除

Top
DeletePrinterConnection
DeletePrinterConnection

VB 声明
Declare Function DeletePrinterConnection Lib "winspool.drv" Alias "DeletePrinterConnectionA" (ByVal pName As String) As Long
说明
删除与指定打印机的连接
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，要删除的打印机连接

Top
DeletePrinterDriver
DeletePrinterDriver

VB 声明
Declare Function DeletePrinterDriver Lib "winspool.drv" Alias "DeletePrinterDriverA" (ByVal pName As String, ByVal pEnvironment As String, ByVal pDriverName As String) As Long
说明
从系统删除一个打印机驱动程序
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，服务器的名字。针对本地驱动程序设为 vbNullString
pEnvironmentString，指定欲在其中删除驱动程序的一个环境（如“Windows NT x86”）。对于当前（本地）系统环境，则设为 vbNullString

pDriverNameString，要删除的驱动程序的名字
注解
驱动程序文件不会从系统物理性的删除，只是不能继续使用

Top
DeletePrintProcessor
DeletePrintProcessor

VB 声明
Declare Function DeletePrintProcessor Lib "winspool.drv" Alias "DeletePrintProcessorA" (ByVal pName As String, ByVal pEnvironment As String, ByVal pPrintProcessorName As String) As Long
说明
从指定系统删除一个打印处理器
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，服务器的名字。针对本地处理器设为 vbNullString
pEnvironmentString，指定欲删除打印处理器的一个环境（如“Windows NT x86”）。对于当前（本地）系统环境，则设为 vbNullString
pPrintProcessorNameString，要删除的打印处理器的名字

Top
DeletePrintProvidor
DeletePrintProvidor

VB 声明
Declare Function DeletePrintProvidor Lib "winspool.drv" Alias "DeletePrintProvidorA" (ByVal pName As String, ByVal pEnvironment As String, ByVal pPrintProvidorName As String) As Long
说明
从系统中删除一个打印供应商
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，服务器的名字。针对本地服务器设为 vbNullString
pEnvironmentString，指定欲删除打印供应商的一个环境（如“Windows NT x86”）。对于当前（本地）系统环境，则设为 vbNullString
pPrintProvidorNameString，要删除的打印供应商的名字

Top
DeviceCapabilities
DeviceCapabilities

VB 声明

Declare Function DeviceCapabilities Lib "winspool.drv" Alias "DeviceCapabilitiesA" (ByVal lpDeviceName As String, ByVal lpPort As String, ByVal iIndex As Long, ByVal lpOutput As String, lpDevMode As DEVMODE) As Long

说明

利用这个函数可获得与一个设备的能力有关的信息

返回值

Long，由 iIndex 参数的值决定，请参考设备能力常数表。如函数执行失败，或打印机的驱动程序不支持这个函数，那么函数就会返回-1

参数表

参数类型及说明

lpDeviceNameString，设备名

lpPortString，指定连接了指定设备的那个端口

iIndexLong，欲测试的能力。请参考设备能力常数表，其中列出了可选的值

lpOutputString，指定一个缓冲区的地址，能力数据会装载到这个缓冲区中。在设备能力常数表中，针对每个 fwCapabilities 值的缓冲区的内容都进行了总结。这个表格同时总结了应将参数设为 vbNullString 的一些情况

lpDevModeDEVMODE，一个 DEVMODE 结构的地址，或者为零。如指定了那个结构，函数会根据这个结构的设置来接收信息。如果为零，函数就会根据打印机驱动程序的默认值接收信息

注解

使用 lpOutput 时要注意：在许多时候，这个函数会返回一系列名称的列表。例如，假设将 fwCapabilities 标志设为 DC_PAPERNAMES，那么就会得到一系列支持的纸张尺寸的名字。在这种情况下，lpOutput 缓冲区应该是一个 String 变量，而且根据设备能力常数表的总结预先初始化成合适的长度。函数会在缓冲区中载入所有名称，而且每个名称在字串中都占用固定的空间。所以，我们完全能用 Mid 函数提取出每一个条目。

某些情况下，lpOutput 需要指向一个数值数组的指针

设备能力常数表

fwCapabilities 说明

DC_BINADJUST 返回来自 API32.TXT 的某个常数。它应带有 DCBA_ 前缀，用于指定当前纸张源的正确纸张方向。仅适用于 Win95

DC_BINNAMES 如 lpOutput 为零，就返回由打印机支持的纸匣数量。否则，lpOutput 应指向一个缓冲区（长度至少为 24×纸匣数）。每 24 个字节都会保存一个纸匣的 NULL 中止名称

DC_BINS 如 lpOutput 为零，就返回由打印机支持的纸匣数量。否则，lpOutput 应指向一个整数数组（长度至少为纸匣数量）。这些值对应于为 DEVMODE 结构定义的 DMBIN_??? 常数

DC_COPIES 返回打印机能够打印的最大副本数量

DC_DATATYPE_PRODUCED 接收由打印机支持的一系列数据类型。这些类型可作为由 StartDoc 函数使用的 DOCINFO 结构的输出数据类型提供。如这个函数返回-1，那么支持的唯一数据类型就是 RAW 。仅适用于 Win95

DC_DRIVER 返回打印机驱动程序的版本号

DC_DUPLEX 如打印机有双面打印功能，就返回 1；否则返回 0
DC_EMF_COMPLIANT 如打印机能直接支持增强型图元文件，就返回 TRUE。仅适用于 Win95
DC_ENUMRESOLUTIONS 如 lpOutput 为零，就返回由打印机支持的分辨率数量。否则，lpOutput 应该是一个指向 Long 型数组的指针。该数组至少应包含（2×分辨率数量）个条目。每对条目都反映出水平和垂直分辨率（以每英寸的点数——dpi——为单位）
DC_EXTRA 返回与具体设备有关的特殊字节，它们要为这个设备追加到 DEVMODE 结构后面
DC_FIELDS 针对设备默认的 DEVMODE 数据结构，返回 dmFields 字段的值
DC_FILEDEPENDENCIES 如 lpOutput 为零，就返回打印机驱动程序要求的文件数量。否则，lpOutput 应指向一个至少有（64×文件数）个字节的缓冲区。每 64 个字节都会保存一个请求文件的 NULL 中止名称
DC_MAXEXTENT 返回一个 Long 型值，其中包含打印机支持的最大纸张长度和宽度。其中，低字（16 位）包含的是宽度数据。它们是由 dmPaperWidth 和 dmPaperLength 这两个 DEVMODE 字段的最大值
DC_MINEXTENT 返回一个 Long 型值，其中包含打印机支持的最小纸张长度和宽度。其中，低字（16 位）包含的是宽度数据。它们是由 dmPaperWidth 和 dmPaperLength 这两个 DEVMODE 字段的最大值
DC_ORIENTATION 返回横向模式和纵向模式间的旋转度数。如果是零，表示驱动程序不支持横向打印模式。对于激光打印机，90 度是最常见的一个设置；而对于点阵式打印机，一般都是 270 度
DC_PAPERNAMES 如 lpOutput 为零，就返回由打印机支持的纸张尺寸数量。否则，lpOutput 就应指向一个缓冲区（长度至少为 64×纸张尺寸种数）。每 64 个字节都会装载一种支持的纸张尺寸的空中止名称
DC_PAPERS 如 lpOutput 为零，就返回由打印机支持的纸张尺寸数量。否则，lpOutput 就应指向一个整数数组（长度至少为纸张的尺寸种数）。值对应于为 DEVMODE 结构定义的 DMPAPER_??? 常数
DC_SIZE 返回打印机 DEVMODE 数据结构的 dmSize 字段
DC_TRUETYPE 下述常数之一：
DCTT_BITMAP 设备能将 TrueType 字体当作图形打印
DCTT_DOWNLOAD 设备能下载 TrueType 字体
DCTT_OUTLINE 设备能下载轮廓型 TrueType 字体
DCTT_SUBDEV 设备能取代与对应的 TrueType 字体兼容的内建字体
DC_VERSION 返回设备驱动程序的规格版本号

Top
DocumentProperties
DocumentProperties

VB 声明
Declare Function DocumentProperties& Lib "winspool.dll" Alias "DocumentPropertiesA" (ByVal hwnd As Long, ByVal hPrinter As Long, ByVal pDeviceName As String, ByVal pDevModeOutput As Long, ByVal pDevModeInput As Long, ByVal fMode As Long)
说明

这是一个灵活的打印机配置控制函数。该函数定义了两个 DEVMODE 结构，可在创建一个设备场景时为单个应用程序改变打印机设置。甚至能在文档打印期间改变打印机设置
返回值
Long，由 fMode 字段的值决定。如下所示：
若 fMode 为零，这个函数就返回 DEVMODE 结构的尺寸。注意这个结构可能比类型定义文件 API32.TXT 中规定的尺寸大
若 fMode 设置了 DM_IN_PROMPT 标志，那么打印机设置对话框就会出现。在这种情况下，返回值将是常数 IDOK 或 IDCANCEL——具体由用户关闭对话框时按下的按钮决定
在其他任何情况下，该函数执行成功后会返回 IDOK。而在任何情况下，如函数执行失败，都会返回一个负数
参数表
参数类型及说明
hwndLong，对话框父窗口的句柄。这通常是当前的活动窗体
hPrinterLong，一个已打开的打印机对象的句柄
pDeviceNameString，打印机的名字
pDevModeOutputLong，指向一个 DEVMODE 数据结构的指针。请参考 DocumentProperties 运行模式表。注意这个指针必须引用一个足够大的缓冲区，它能同时容下专用打印机驱动程序数据，以及标准的 DEVMODE 结构
pDevModeInputLong，指向一个 DEVMODE 数据结构的指针。请参考 DocumentProperties 运行模式表
fModeLong，决定这个函数运作模式的一个标志。请参考 DocumentProperties 运行模式表

DocumentProperties 运行模式表
常数标志运行模式
无不使用 pDevModeInput。pDevModeOutput 可能为零。函数会返回由这两个参数引用的 DEVMODE 结构需要的大小
DM_IN_BUFFERpDevModeInput 缓冲区应载入打印机驱动程序的新位置。在调用这个函数判断应使用结构中的哪些字段前，应设置结构的 dmFields 字段
DM_IN_PROMPT 显示出打印机设置对话框，以便用户指定输出时采用的打印机设置。如指定了 DM_IN_BUFFER，那么在显示对话框前，输入缓冲区中指定的任何字段都会与当前的打印机 DEVMODE 结构合并起来
DM_OUT_BUFFER 令打印机设置信息输出到由 pDevModeOutput 参数指定的缓冲区。这些设置由两个输入标志决定，而且由此反映了原始的输入结构、当前的打印机设置以及用户在打印机设置对话框中作出的任何修改。如未指定这个标志，lpdmOutput 参数就可以设为零

Top
EndDocAPI
EndDocAPI

VB 声明
Declare Function EndDocAPI& Lib "gdi32" Alias "EndDoc" (ByVal hDC As Long)
说明
结束一个成功的打印作业
返回值

Long，大于零表示成功，小于或等于零表示失败。会设置 GetLastError
参数表
参数类型及说明
hDCLong，设备场景的句柄

Top
EndDocPrinter
EndDocPrinter

VB 声明
Declare Function EndDocPrinter Lib "winspool.drv" Alias "EndDocPrinter" (ByVal hPrinter As Long) As Long
说明
在后台打印程序的级别指定一个文档的结束
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，一个已打开的打印机的句柄（用用 OpenPrinter 获得）
注解
在应用程序的级别并非特别有用。后台打印程序用它标识打印作业中一个文档的结束

Top
EndPage
EndPage

VB 声明
Declare Function EndPage Lib "gdi32" Alias "EndPage" (ByVal hdc As Long) As Long
说明
用这个函数完成一个页面的打印，并准备设备场景，以便打印下一个页
返回值
Long，大于零表示成功，小于或等于零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
在 VB 里使用
使用 VB 的打印机对象时，应该使用 NewPage 方法来代替这个函数
注解
这个函数的执行可能相当耗时——具体取决于正在进行的绘图操作的复杂程度、使用什么操作系统以及打印机与本机连接还是与远程主机连接。为了让用户能随时取消打印，在执行期间，这个函数会定时调用由 API 调用 SetAbortProc 指定的取消例程

Top
EndPagePrinter

EndPagePrinter

VB 声明
Declare Function EndPagePrinter Lib "winspool.drv" Alias "EndPagePrinter" (ByVal hPrinter As Long) As Long
说明
指定一个页在打印作业中的结尾
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，一个已打开的打印机对象的句柄（用 OpenPrinter 获得）
注解
在应用程序的级别并非特别有用，后台打印程序用它在实际打印机数据中标识一个页的结尾

Top
EnumForms
EnumForms

VB 声明
Declare Function EnumForms Lib "winspool.drv" Alias "EnumFormsA" (ByVal hPrinter As Long, ByVal Level As Long, pForm As Byte, ByVal cbBuf As Long, pcbNeeded As Long, pcReturned As Long) As Long
说明
枚举一台打印机可用的表单
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，一个已打开的打印机对象的句柄（用 OpenPrinter 获得）
LevelLong，设为 1
pFormByte，一个包含 FORM_INFO_1 结构的缓冲区
cbBufLong，pForm 缓冲区中的字符数量
pcbNeededLong，指向一个 Long 型变量的指针，该变量用于保存请求的缓冲区长度，或者实际读入的字节数量
pcReturnedLong，载入缓冲区的结构数量（用于那些能返回多个结构的函数）
适用平台
Windows NT
注解
参考 AddForm 函数，了解进一步的情况

Top
EnumJobs
EnumJobs

VB 声明
Declare Function EnumJobs Lib "winspool.drv" Alias "EnumJobsA" (ByVal hPrinter As Long, ByVal FirstJob As Long, ByVal NoJobs As Long, ByVal Level As Long, pJob As Byte, ByVal cdBuf As Long, pcbNeeded As Long, pcReturned As Long) As Long
说明
枚举打印队列中的作业
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，一个已打开的打印机对象的句柄（用 OpenPrinter 获得）
FirstJobLong，作业列表中要枚举的第一个作业的索引（注意编号从 0 开始）
NoJobsLong，要枚举的作业数量
LevelLong，1 或 2
pJobByte，包含 JOB_INFO_1 或 JOB_INFO_2 结构的缓冲区
cbBufLong，pJob 缓冲区中的字符数量
pcbNeededLong，指向一个 Long 型变量的指针，该变量用于保存请求的缓冲区长度，或者实际读入的字节数量
pcReturnedLong，载入缓冲区的结构数量（用于那些能返回多个结构的函数）
适用平台

Top
EnumMonitors
EnumMonitors

VB 声明
Declare Function EnumMonitors Lib "winspool.drv" Alias "EnumMonitorsA" (ByVal pName As String, ByVal Level As Long, pMonitors As Byte, ByVal cbBuf As Long, pcbNeeded As Long, pcReturned As Long) As Long
说明
枚举可用的打印监视器
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，服务器的名字。用 vbNullString 指定本地系统
LevelLong，设为 1
pMonitorsByte，包含 MONINTOR_INFO_1 结构的缓冲区
cbBufLong，pMonitors 缓冲区中的字符数量
pcbNeededLong，指向一个 Long 型变量的指针，该变量用于保存请求的缓冲区长度，或者实际读入的字节数量
pcReturnedLong，载入缓冲区的结构数量（用于那些能返回多个结构的函数）
注解

MONINTOR_INFO_1 结构有一个字段包含了打印监视器的名字

Top
EnumPorts
EnumPorts

VB 声明
Declare Function EnumPorts Lib "winspool.drv" Alias "EnumPortsA" (ByVal pName As String, ByVal Level As Long, ByVal lpbPorts As Long, ByVal cbBuf As Long, pcbNeeded As Long, pcReturned As Long) As Long
说明
枚举一个系统可用的端口
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，指定服务器的名字。用 vbNullString 指定本地系统
LevelLong，1 或 2（1 用于 NT 3.51），分别指定 PORT_INFO_1 或 PORT_INFO_2
lpbPortsLong，包含 PORT_INFO_1 或 PORT_INFO_2 结构的缓冲区
cbBufLong，lpbPorts 缓冲区中的字符数量
pcbNeededLong，指向一个 Long 型变量的指针，该变量用于保存请求的缓冲区长度，或者实际读入的字节数量
pcReturnedLong，载入缓冲区的结构数量（用于那些能返回多个结构的函数）
注解
参考 AddPort 函数，了解进一步的情况

Top
EnumPrinterDrivers
EnumPrinterDrivers

VB 声明
Declare Function EnumPrinterDrivers Lib "winspool.drv" Alias "EnumPrinterDriversA" (ByVal pName As String, ByVal pEnvironment As String, ByVal Level As Long, pDriverInfo As Byte, ByVal cdBuf As Long, pcbNeeded As Long, pcRetruned As Long) As Long
说明
枚举指定系统中已安装的打印机驱动程序
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，指定服务器的名字。用 vbNullString 指定本地系统
pEnvironmentString，欲在其中对驱动程序进行枚举的环境（如：Windows NT x86）。如设为 vbNullString，表示使用当前（本地）系统环境
LevelLong，1，2 或 3（3 仅适用于 Windows 95 和 NT 4.0）

pDriverInfoByte，包含 DRIVER_INFO_1， DRIVER_INFO_2 或 DRIVER_INFO_3 结构的缓冲区
cbBufLong，pDriverInfo 缓冲区中的字符数量
pcbNeededLong，指向一个 Long 型变量的指针，该变量用于保存请求的缓冲区长度，或者实际读入的字节数量
pcReturnedLong，载入缓冲区的结构数量（用于那些能返回多个结构的函数）

Top
EnumPrinters
EnumPrinters

VB 声明
Declare Function EnumPrinters Lib "winspool.drv" Alias "EnumPrintersA" (ByVal flags As Long, ByVal name As String, ByVal Level As Long, pPrinterEnum As Byte, ByVal cdBuf As Long, pcbNeeded As Long, pcReturned As Long) As Long
说明
枚举系统中安装的打印机
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
flagsLong，一个或多个下述标志
PRINTER_ENUM_LOCAL 枚举本地打印机（包括 Windows 95 中的网络打印机）。名字会被忽略
PRINTER_ENUM_NAME 枚举由 name 参数指定的打印机。其中的名字可以是一个供应商、域或服务器。如 name 为 NULL，则枚举出可用的打印机
PRINTER_ENUM_SHARE 枚举共享打印机（必须同其他常数组合使用）
PRINTER_ENUM_CONNECTIONS 枚举网络连接列表中的打印机（即使目前没有连接——仅适用于 NT）
PRINTER_ENUM_NETWORK 枚举通过网络连接的打印机。级别（Level）必须为 1。仅适用于 NT
PRINTER_ENUM_REMOTE 枚举通过网络连接的打印机和打印服务器。级别必须为 1。仅适用于 NT
nameString，vbNullString 表示枚举同本机连接的打印机。否则由标志和级别决定
LevelLong，1，2，4 或 5（4 仅适用于 NT；5 仅适用于 Win95 和 NT 4.0），指定欲枚举的结构的类型。如果是 1，则 name 参数由标志设置决定。如果是 2 或 5，那么 name 就代表欲对其打印机进行枚举的服务器的名字；或者为 vbNullString。如果是 4，那么只有 PRINTER_ENUM_LOCAL 和 PRINTER_ENUM_CONNECTIONS 才有效。名字必须是 vbNullString
pPrinterEnumByte，包含 PRINTER_ENUM_x 结构的缓冲区，其中的 x 代表级别（Level）
cbBufLong，pPrinterEnum 缓冲区中的字符数量
pcbNeededLong，指向一个 Long 型变量的指针，该变量用于保存请求的缓冲区长度，或者实际读入的字节数量
pcReturnedLong，载入缓冲区的结构数量（用于那些能返回多个结构的函数）

注解
第 4 和第 5 级将它们的结构建立在系统注册表的基础上，而且比第 2 级快得多。后者要求每台打印机都处于打开状态
请参考微软 Win32 手册，了解这个函数进一步的情况

Top
EnumPrintProcessorDatatypes
EnumPrintProcessorDatatypes

VB 声明
Declare Function EnumPrintProcessorDatatypes Lib "winspool.drv" Alias "EnumPrintProcessorDatatypesA" (ByVal pName As String, ByVal pPrintProcessorName As String, ByVal Level As Long, pDatatypes As Byte, ByVal cdBuf As Long, pcbNeeded As Long, pcRetruned As Long) As Long
说明
枚举由一个打印处理器支持的数据类型
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，指定服务器的名字。用 vbNullString 表示使用本地系统
pPrintProcessorNameString，欲对其数据类型进行枚举的打印处理器的名字
LevelLong，设为 1
pDatatypesByte，包含 DATATYPES_INFO_1 结构的缓冲区
cbBufLong，pDatatypes 缓冲区中的字符数量
pcbNeededLong，指向一个 Long 型变量的指针，该变量用于保存请求的缓冲区长度，或者实际读入的字节数量
pcReturnedLong，载入缓冲区的结构数量（用于那些能返回多个结构的函数）
注解
DATATYPES_INFO_1 结构包含了单个字段，其中保存了数据类型的名称

Top
EnumPrintProcessors
EnumPrintProcessors

VB 声明
Declare Function EnumPrintProcessors Lib "winspool.drv" Alias "EnumPrintProcessorsA" (ByVal pName As String, ByVal pEnvironment As String, ByVal Level As Long, pPrintProcessorInfo As Byte, ByVal cdBuf As Long, pcbNeeded As Long, pcReturned As Long) As Long
说明
枚举系统中可用的打印处理器
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表

参数类型及说明
pNameString，指定服务器的名字。用 vbNullString 表示使用本地系统
pEnvironmentString，欲枚举的打印处理器的环境（如： Windows NT x86）。如设为 vbNullString，表示使用当前（本地）系统环境
LevelLong，设为 1
pPrintProcessorInfoByte，包含 PRINTPROCESSOR_INFO_1 结构的缓冲区
cbBufLong，pPrintProcessorInfo 缓冲区中的字符数量
pcbNeededLong，指向一个 Long 型变量的指针，该变量用于保存请求的缓冲区长度，或者实际读入的字节数量
pcReturnedLong，载入缓冲区的结构数量（用于那些能返回多个结构的函数）

Top
Escape
Escape

VB 声明
Declare Function Escape Lib "gdi32" Alias "Escape" (ByVal hdc As Long, ByVal nEscape As Long, ByVal nCount As Long, ByVal lpInData As String, lpOutData As Any) As Long
说明
一个灵活的设备控制函数
返回值
Long，对于 QUERYESCSUPPORT，如支持指定的换码，则返回 TRUE（非零）；否则返回零。对于 PASSTHROUGH，大于零值表示成功；如指定的换码不支持，则返回零；如果出错，则返回负值
参数表
参数类型及说明
hdcLong，设备场景的句柄
nEscapeLong，换码数量，由 API32.TXT 文件中的一个常数定义。这决定了具体的运作方式。请参考注解
nCountLong，lpInData 缓冲区的大小，用字节数表示
lpInDataString，由换码类型决定。对于 QUERYESCSUPPORT，这代表指向一个整数变量的指针，那个变量包含了要测试的换码值。对于 PASSTHROUGH，这代表指向一个数据块的指针，那个数据块包含于要发送数据的头 16 位字节数量中。数据块剩余的部分包含了要发送给打印机的实际数据缓冲区
lpOutDataAny，指定一个输出缓冲区，它的具体使用由换码决定。它不由 QUERYESCSUPPORT 或 PASSTHROUGH 使用，而且应设为 NULL（ByVal 0&）
注解
只有两个换码在 Win32 环境中经常用到。请用 QUERYESCSUPPORT 换码判断一个换码是否得到了驱动程序的支持。用 PASSTHROUGH 换码将原始数据直接发给一台打印机。其他换码在 Win32 仍然得到了支持，但目的只是为了与 Win16 保持兼容

Top
FindClosePrinterChangeNotification
FindClosePrinterChangeNotification

VB 声明
Declare Function FindClosePrinterChangeNotification Lib "winspool.drv" Alias "FindClosePrinterChangeNotification" (ByVal hChange As Long) As Long
说明
关闭用 FindFirstPrinterChangeNotification 函数获取的一个打印机通告对象
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hChangeLong，欲关闭的打印机通告对象句柄
适用平台
Windows NT

Top
FindFirstPrinterChangeNotification
FindFirstPrinterChangeNotification

VB 声明
Declare Function FindFirstPrinterChangeNotification& Lib "winspool.dll" (ByVal hPrinter As Long, ByVal fdwFlags As Long, ByVal fdwOptions As Long, pPrinterNotifyOptions As Byte)
说明
创建一个新的改变通告对象，以便我们注意打印机状态的各种变化
返回值
Long，执行成功则返回改变通告对象的句柄。INVALID_HANDLE_VALUE 表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，一个已打开的打印机对象的句柄（用 OpenPrinter 获得）
fdwFlagsLong，来自 API32.TXT 文件的、带有 PRINTER_CHANGE_??前缀的某个常数，它们对要观察的对象进行了描述。如 pPrinterNotifyOptions 不为零，那么可将这个参数设为零
fdwOptionsLong，保留，设为零
pPrinterNotifyOptionsByte，指定一个缓冲区，其中包含了一个 PRINTER_NOTIFY_OPTIONS 结构。而这个结构又包含了指向一个或多个 PRINTER_NOTIFY_OPTIONS_TYPE 结构的指针。可将这个参数设为零（将声明方式改为 ByVal As Long 并传递零值），以便用 fdwFlags 字段指定想观察的变化
适用平台
Windows NT
其他
以下的声明是从 VB 的 API 文本查看器里复制的：
Declare Function FindFirstPrinterChangeNotification Lib "winspool.drv" Alias "FindFirstPrinterChangeNotification" (ByVal hPrinter As Long, ByVal fdwFlags As Long, ByVal fdwOptions As Long, ByVal pPrinterNotifyOptions As String) As Long

Top
FindNextPrinterChangeNotification
FindNextPrinterChangeNotification

VB 声明
Declare Function FindNextPrinterChangeNotification& Lib "winspool.dll" (ByVal hChange As Long, pdwChange As Long, ByVal pvReserved As Long, ByVal ppPrinterNotifyInfo As Long)
说明
用这个函数判断触发一次打印机改变通告信号的原因
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hChangeLong，一个打印机通告改变对象的句柄
pdwChangeLong，指定用于装载特定标志的一个 Long 型值，该标志标志着信号的来源。请在 API32.TXT 文件中寻找以 PRINTER_CHANGE_???前缀开头的常数
pvReservedLong，指定一个 PRINTER_NOTIFY_OPTIONS 结构的地址。如这个结构的 Flags 字段设为 PRINTER_NOTIFY_OPTIONS_REFRESH，那么 ppPrinterNotifyInfo 缓冲区就会载入正在监视的所有事件的状态——并不仅是那些触发了通告信号的事件。结构中所有其他字段会被忽略。可设为 NULL（零），表示只返回与状态改变有关信息
ppPrinterNotifyInfoLong，由系统分配的一个缓冲区的地址。完成后，应该用 FreePrinterNotifyInfo 函数将这个缓冲区删除。缓冲区内包含了一个 PRINTER_NOTIFY_INFO 结构，其后跟随一系列 PRINTER_NOTIFY_INFO_DATA 结构（具体数量由第一个结构决定）
适用平台
Windows NT
其他
在 VB 的 API 文本查看器中复制的声明如下：
Declare Function FindNextPrinterChangeNotification Lib "winspool.drv" Alias "FindNextPrinterChangeNotification" (ByVal hChange As Long, pdwChange As Long, ByVal pvReserved As String, ByVal ppPrinterNotifyInfo As Long) As Long

Top
FreePrinterNotifyInfo
FreePrinterNotifyInfo

VB 声明
Declare Function FreePrinterNotifyInfo Lib "winspool.drv" (ByVal addr As Long) As Long
说明
释放由 FindNextPrinterChangeNotification 函数分配的一个缓冲区
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明

addrLong，指定由 FindNextPrinterChangeNotification 函数分配的一个系统缓冲区的地址
适用平台
Windows NT

Top
GetForm
GetForm

VB 声明
Declare Function GetForm Lib "winspool.drv" Alias "GetFormA" (ByVal hPrinter As Long, ByVal pFormName As String, ByVal Level As Long, pForm As Byte, ByVal cbBuf As Long, pcbNeeded As Long) As Long
说明
取得与指定表单有关的信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，一个已打开的打印机的句柄（用 OpenPrinter 获得）
pFormNameString，想获取信息的一个表单的名字
LevelLong，设为 1
pFormByte，包含 FORM_INFO_1 结构的缓冲区
cbBufLong，pForm 缓冲区中的字符数量
pcbNeededLong，指向一个 Long 型变量的指针，该变量用于保存请求的缓冲区长度，或者实际读入的字节数量
适用平台
Windows NT

Top
GetJob
GetJob

VB 声明
Declare Function GetJob Lib "winspool.drv" Alias "GetJobA" (ByVal hPrinter As Long, ByVal JobId As Long, ByVal Level As Long, pJob As Byte, ByVal cdBuf As Long, pcbNeeded As Long) As Long
说明
获取与指定作业有关的信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，一个已打开的打印机的句柄（用 OpenPrinter 获得）
JobIdLong，作业编号

LevelLong，1 或 2
pJobByte，包含 JOB_INFO_1 或 JOB_INFO_2 结构的缓冲区，结构中包含了与打印作业有关的信息
cbBufLong，pJob 缓冲区中的字符数量
pcbNeededLong，指向一个 Long 型变量的指针，该变量用于保存请求的缓冲区长度，或者实际读入的字节数量

Top
GetPrinter
GetPrinter

VB 声明
Declare Function GetPrinter Lib "winspool.drv" Alias "GetPrinterA" (ByVal hPrinter As Long, ByVal Level As Long, pPrinter As Any, ByVal cbBuf As Long, pcbNeeded As Long) As Long
说明
取得与指定打印机有关的信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，一个已打开的打印机的句柄（用 OpenPrinter 获得）
LevelLong，1，2，3（仅适用于 NT），4（仅适用于 NT），或者 5（仅适用于 Windows 95 和 NT 4.0）
pPrinterAny，包含 PRINTER_INFO_x 结构的缓冲区。x 代表级别
cbBufLong，pPrinterEnum 缓冲区中的字符数量
pcbNeededLong，指向一个 Long 型变量的指针，该变量用于保存请求的缓冲区长度，或者实际读入的字节数量
注解
只有在发出调用的应用程序有足够的权限时，PRINTER_INFO_x 结构中的一些字段才能够被读取。这种权限由系统当前的安全设置决定

Top
GetPrinterData
GetPrinterData

VB 声明
Declare Function GetPrinterData Lib "winspool.drv" Alias "GetPrinterDataA" (ByVal hPrinter As Long, ByVal pValueName As String, pType As Long, pData As Byte, ByVal nSize As Long, pcbNeeded As Long) As Long
说明
为打印机设置注册表配置信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表

参数类型及说明
hPrinterLong，一个已打开的打印机的句柄（用 OpenPrinter 获得）
pValueNameString，欲设置的注册表值的名称
pTypeLong，指定数据类型。使用来自 API32.TXT 的、以 REG_??开头的一个常数
pDataByte，指定一个 Byte 数组以接收数据
nSizeLong，以字节表示的 pData 数组的长度
pcbNeededLong，指向一个 Long 型变量的指针，该变量用于保存请求的缓冲区长度，或者实际读入的字节数量

Top
GetPrinterDriver
GetPrinterDriver

VB 声明
Declare Function GetPrinterDriver Lib "winspool.drv" Alias "GetPrinterDriverA" (ByVal hPrinter As Long, ByVal pEnvironment As String, ByVal Level As Long, pDriverInfo As Byte, ByVal cdBuf As Long, pcbNeeded As Long) As Long
说明
针对指定的打印机，获取与打印机驱动程序有关的信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，一个已打开的打印机的句柄（用 OpenPrinter 获得）
pEnvironmentString，欲获取的驱动程序环境（如：Windows NT x86）。如设为 vbNullString，表示使用当前（本地）系统环境
LevelLong，1，2 或 3（仅适用于 Windows 95 和 NT 4.0）
pDriverInfoByte，载入一个 DRIVER_INFO_x 结构的缓冲区。其中的 x 代表级别（Level）设置
cbBufLong，pDriverInfo 缓冲区中的字符数量
pcbNeededLong，指向一个 Long 型变量的指针，该变量用于保存请求的缓冲区长度，或者实际读入的字节数量

Top
GetPrinterDriverDirectory
GetPrinterDriverDirectory

VB 声明
Declare Function GetPrinterDriverDirectory Lib "winspool.drv" Alias "GetPrinterDriverDirectoryA" (ByVal pName As String, ByVal pEnvironment As String, ByVal Level As Long, pDriverDirectory As Byte, ByVal cdBuf As Long, pcbNeeded As Long) As Long
说明
判断指定系统中包含了打印机驱动程序的目录是什么
返回值

Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，服务器的名字。如设为 vbNullString，表示使用本地系统
pEnvironmentString，欲在其中获取目录的一个环境（如： Windows NT x86）。vbNullString 表示使用当前（本地）系统环境
LevelLong，设为 1
pDriverDirectoryByte，指定一个缓冲区，其中会载入打印机驱动程序目录的完整路径名。可定义成 ByVal As String，以便将字节数组分配给一个字串，从而避免执行 ANSI 到 Unicode 格式的转换
cbBufLong，pDriverDirectory 缓冲区中的字符数量
pcbNeededLong，指向一个 Long 型变量的指针，该变量用于保存请求的缓冲区长度，或者实际读入的字节数量

Top
GetPrintProcessorDirectory
GetPrintProcessorDirectory

VB 声明
Declare Function GetPrintProcessorDirectory Lib "winspool.drv" Alias "GetPrintProcessorDirectoryA" (ByVal pName As String, ByVal pEnvironment As String, ByVal Level As Long, ByVal pPrintProcessorInfo As String, ByVal cdBuf As Long, pcbNeeded As Long) As Long
说明
判断指定系统中包含了打印机处理器驱动程序及文件的目录
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，指定服务器的名字。设置成 vbNullString 则表示使用本地系统
pEnvironmentString，欲在其中获取目录的一个环境（如：Windows NT x86）。用 vbNullString 表示使用当前（本地）系统环境
LevelLong，设为 1
pPrintProcessorInfoString，指定一个缓冲区，其中载入打印机处理器目录的完整路径。可定义成 ByVal As String，以便将字节数组分配给一个字串，从而取消进行 ANSI 到 Unicode 转换的必要
cbBufLong，pPrintProcessorInfo 缓冲区中的字符数量
pcbNeededLong，指向一个 Long 型变量的指针，该变量用于保存请求的缓冲区长度，或者实际读入的字节数量

Top
OpenPrinter
OpenPrinter

VB 声明
Declare Function OpenPrinter Lib "winspool.drv" Alias "OpenPrinterA" (ByVal pPrinterName As String, phPrinter As Long, pDefault As PRINTER_DEFAULTS) As Long
说明
打开指定的打印机，并获取打印机的句柄
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pPrinterNameString，要打开的打印机的名字
phPrinterLong，用于装载打印机的句柄
pDefaultPRINTER_DEFAULTS，这个结构保存要载入的打印机信息

Top
PrinterMessageBox
PrinterMessageBox

VB 声明
Declare Function PrinterMessageBox Lib "winspool.drv" Alias "PrinterMessageBoxA" (ByVal hPrinter As Long, ByVal error As Long, ByVal hwnd As Long, ByVal pText As String, ByVal pCaption As String, ByVal dwType As Long) As Long
说明
在拥有指定打印作业的系统上显示一个打印机出错消息框。如一名用户在远程登录，这种做法便相当有用
返回值
Long，IDOK， IDRETRY 或 IDCANCEL；由用户的输入决定（如消息框在远程系统显示，则肯定是 IDOK）
参数表
参数类型及说明
hPrinterLong，出现错误的打印机的句柄
errorLong，ERROR_OUT_OF_PAPER（缺纸）或 ERROR_NOT_READY（未就绪）
hwndLong，指定消息框的父窗口。可以为 NULL
pTextLong，欲显示的消息正文
pCaptionLong，消息框的标题
dwTypeLong，指定任何一个标准的 MessageBox 标志。建议使用 MB_ICONSTOP 或 MB_RETRYCANCEL 或 MB_SETFOREGROUND
适用平台
Windows NT

Top
PrinterProperties
PrinterProperties

VB 声明

Declare Function PrinterProperties Lib "winspool.drv" Alias "PrinterProperties" (ByVal hwnd As Long, ByVal hPrinter As Long) As Long
说明
启动打印机属性对话框，以便对打印机进行配置
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，对话框的父窗口
hPrinterLong，一个已打开的打印机的句柄
注解
如打印机打开的时候没有使用足够的访问权限，对话框的有些功能也许会禁止使用

Top
ReadPrinter
ReadPrinter

VB 声明
Declare Function ReadPrinter Lib "winspool.drv" Alias "ReadPrinter" (ByVal hPrinter As Long, pBuf As Any, ByVal cdBuf As Long, pNoBytesRead As Long) As Long
说明
从打印机读入数据
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，一个已打开的打印机的句柄（用 OpenPrinter 获得）
pBufAny，指定一个缓冲区或结构，用于装载来自打印机的数据
cdBufLong，欲读入的缓冲区大小或字节数
pNoBytesReadLong，用于装载实际读取字节数的一个变量
注解
为使这个函数正常使用，端口必须是双向的

Top
ResetDC
ResetDC

VB 声明
Declare Function ResetDC Lib "gdi32" Alias "ResetDCA" (ByVal hdc As Long, lpInitData As DEVMODE) As Long
说明
根据提供的 DEVMODE 结构，对一个设备场景进行重设。这样便允许我们在打印期间改变打印机的配置。利用这个函数，可将文档中的某个页改为横向打印。可试着用 DocumentProperties 函数取得一个设备的默认 DEVMODE 结构

返回值
Long，执行成功则返回设备场景的句柄，零表示失败
参数表
参数类型及说明
hdcLong，设备场景的句柄
lpInitDataDEVMODE，指定一个缓冲区的第一个字节。该缓冲区包含了用于那个设备的一个有效 DEVMODE 结构。记住在这个缓冲区中包括设备专用的数据区
注解
这个函数可成功用于由 VB 的 Printer 对象的 hDC 属性返回的设备场景上
注意一定要正确设置 lpdm 的 dmFields 字段
这个函数在 StartPage 和 EndPage 之间会被禁用——即只能在页与页之间调用这个函数，不能在页内调用
驱动程序、设备和输出端口不可以用这个函数更改

Top
ResetPrinter
ResetPrinter

VB 声明
Declare Function ResetPrinter Lib "winspool.drv" Alias "ResetPrinterA" (ByVal hPrinter As Long, pDefault As PRINTER_DEFAULTS) As Long
说明
改变指定打印机的默认数据类型及文档设置
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，欲修改的一台打印机的句柄
pDefaultPRINTER_DEFAULTS，定义了打印机新设置的一个结构。参考 OpenPrinter 函数的说明，了解这个结构进一步的细节。结构中的 DesiredAccess 字段会被忽略
适用平台
Windows NT

Top
ScheduleJob
ScheduleJob

VB 声明
Declare Function ScheduleJob Lib "winspool.drv" Alias "ScheduleJob" (ByVal hPrinter As Long, ByVal JobId As Long) As Long
说明
提交一个要打印的作业
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError

参数表
参数类型及说明
hPrinterLong，一台已打开的打印机句柄
JobIdLong，以前用 AddJob 函数获得的作业编号
注解
参考 AddJob 函数以获得进一步的信息

Top
SetAbortProc
SetAbortProc

VB 声明
Declare Function SetAbortProc Lib "gdi32" Alias "SetAbortProc" (ByVal hDC As Long, ByVal lpAbortProc As Long) As Long
说明
我们可以为 Windows 提供一个特殊的函数，令其在扩展打印操作过程中调用。这个函数叫“取消函数”。其结果告诉 Windows 是继续打印操作，还是立即取消
SetAbortProc 函数的作用是为 windows 指定取消函数的地址。由于 VB 不支持函数地址的概念，所以要使用特定的通用回调定制控件，否则就不能使用这个函数
返回值
Long，如结果大于零，表示执行成功；SP_ERROR 表示出错。会设置 GetLastError
参数表
参数类型及说明
hDCLong，一个设备场景的句柄。
lpAbortProcLong，一个取消函数的地址
在 VB 里使用
如随同 VB 的打印机对象使用这个函数，就可能干扰正常的 VB 打印机制。这个函数的确可以在 VB 环境中使用，但有可能造成打印机出错。因此，在下一次使用 Printer.NewPage 方法的时候，有必要用适当的机制捕获这种错误。如果在自己的程序中为 VB 的 Printer 对象设置了一个取消函数，那么建议您完整测试代码
如果在自己创建的一个设备场景中打印，那么这个函数的使用没有丝毫问题

Top
SetForm
SetForm

VB 声明
Declare Function SetForm& Lib "spoolss.dll" Alias "SetFormA" (ByVal hPrinter As Long, ByVal pFormName As String, ByVal Level As Long, pForm As Byte)
说明
为指定的表单设置信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表

参数类型及说明
hPrinterLong，指定一个打开打印机的句柄（用 OpenPrinter 取得）
pFormNameString，欲设置的表单的名字
LevelLong，设为 1
pFormByte，包含一个有效 FORM_INFO_1 结构的缓冲区
适用平台
Windows NT
注解
请参考 AddForm 函数。

Top
SetJob
SetJob

VB 声明
Declare Function SetJob Lib "winspool.drv" Alias "SetJobA" (ByVal hPrinter As Long, ByVal JobId As Long, ByVal Level As Long, pJob As Byte, ByVal Command As Long) As Long
说明
对一个打印作业的状态进行控制
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，指定一个打开打印机的句柄（用 OpenPrinter 取得）
JobIdLong，要修改的作业的编号
LevelLong，0，1 或 2
pJobByte，指定一个缓冲区。如级别（Level）设为 1 或 2，那该缓冲区就包含了一个 JOB_INFO_1 或 JOB_INFO_2 结构。如级别为 0，缓冲区为 NULL（变成 ByVal As Long，以便传递零值）。如指定了一个结构，则来自那个结构的信息会用于改变打印作业的设置（除 JobId，pPrinterName，pMachineName，pDriverName，Size，Submitte 以及 Time 字段外）
CommandLong，下述常数之一：
JOB_CONTROL_CANCEL 取消作业
JOB_CONTROL_PAUSE 暂停作业
JOB_CONTROL_RESTART 重新启动一个已开始打印的作业
JOB_CONTROL_RESUME 恢复一个暂停的作业

Top
SetPrinter
SetPrinter

VB 声明
Declare Function SetPrinter Lib "winspool.drv" Alias "SetPrinterA" (ByVal hPrinter As Long, ByVal Level As Long, pPrinter As Byte, ByVal Command As Long) As Long
说明

对一台打印机的状态进行控制
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，指定一个已打开的打印机的句柄（用 OpenPrinter 取得）
LevelLong，0，2 或 3（4 或 5 用于 windows95，5 或 6 用于 NT 4.0）。如 Command 不是零，则这个参数必须是零
pPrinterByte，包含一个 PRINTER_INFO_x 的结构的缓冲区，其中的 x 代表级别的设定（Level）。假如级别为零，并且 Command 设为 PRINTER_CONTROL_SET_STATUS，那缓冲区就包含了一个 PRINTER_CONTROL_STATUS 结构。否则，如级别为零，就设为 NULL（要把声明变成 ByVal As Long，以便传递零值）
CommandLong，下述值之一：
零根据 PRINTER_INFO_x 结构改变打印机
PRINTER_CONTROL_PAUSE 暂停打印机
PRINTER_CONTROL_PURGE 删除打印机的所有作业
PRINTER_CONTROL_RESUME 恢复一台暂停的打印机
PRINTER_CONTROL_SET_STATUS 载入打印机的 PRINTER_CONTROL_STATUS 结构（不可在 NT 3.51 下使用）
注解
在 PRINTER_INFO_2 结构的基础上设置打印机状态时，pServerName，AveragePPM，Status 和 cJobs 字段都会被忽略

Top
SetPrinterData
SetPrinterData

VB 声明
Declare Function SetPrinterData Lib "winspool.drv" Alias "SetPrinterDataA" (ByVal hPrinter As Long, ByVal pValueName As String, ByVal dwType As Long, pData As Byte, ByVal cbData As Long) As Long
说明
设置打印机的注册表配置信息
返回值
Long，ERROR_SUCCESS 表示成功，一个错误值表示失败。
参数表
参数类型及说明
hPrinterLong，指定一个已打开的打印机的句柄（用 OpenPrinter 取得）
pValueName String，欲设置的注册表值名
dwTypeLong，定义数据的类型，使用来自 API32.txt 文件的、以 REG_起头的一个常数
pDataByte，指定一个缓冲区的第一个条目，缓冲区中包含了要设置的适当的数据类型
cbDataLong，缓冲区 pData 的长度

Top

StartDoc
StartDoc

VB 声明
Declare Function StartDoc Lib "gdi32" Alias "StartDocA" (ByVal hdc As Long, lpdi As DOCINFO) As Long
说明
开始一个打印作业
返回值
Long，如执行成功，返回文档的作业编号，常数 SP_ERROR 失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
lpdiDOCINFO，对文档定义的一个结构

Top
StartDocPrinter
StartDocPrinter

VB 声明
Declare Function StartDocPrinter Lib "winspool.drv" Alias "StartDocPrinterA" (ByVal hPrinter As Long, ByVal Level As Long, pDocInfo As Byte) As Long
说明
在后台打印的级别启动一个新文档
返回值
Long，非零表示成功，零表示失败。
参数表
参数类型及说明
hPrinterLong，指定一个已打开的打印机的句柄（用 OpenPrinter 取得）
LevelLong，1 或 2（仅用于 win95）
pDocInfoByte，包含一个 DOC_INFO_1 或 DOC_INFO_2 结构的缓冲区
注解
在应用程序的级别并非特别有用。后台打印程序用它标识一个文档的开始

Top
StartPage
StartPage

VB 声明
Declare Function StartPage Lib "gdi32" Alias "StartPage" (ByVal hdc As Long) As Long
说明
打印一个新页前要先调用这个函数
返回值
Long，非零表示成功，零表示失败

参数表
参数类型及说明
hdcLong，一个设备场景的句柄
在 VB 里使用
操作 VB 的打印机对象时，注意不要使用这个函数

Top
StartPagePrinter
StartPagePrinter

VB 声明
Declare Function StartPagePrinter Lib "winspool.drv" Alias "StartPagePrinter" (ByVal hPrinter As Long) As Long
说明
在打印作业中指定一个新页的开始
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，指定一个已打开的打印机的句柄（用 OpenPrinter 取得）
注解
在应用程序的级别并非特别有用。后台打印程序用它在实际的打印机数据中标识一个文档的开始

Top
WritePrinter
WritePrinter

VB 声明
Declare Function WritePrinter Lib "winspool.drv" Alias "WritePrinter" (ByVal hPrinter As Long, pBuf As Any, ByVal cdBuf As Long, pcWritten As Long) As Long
说明
将发送目录中的数据写入打印机
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，指定一个已打开的打印机的句柄（用 OpenPrinter 取得）
pBufAny，包含了要写入打印机的数据的一个缓冲区或结构
cdBufLong，pBuf 缓冲区的长度
pcWrittenLong，指定一个 Long 型变量，用于装载实际写入的字节数

Top
设备场景函数

设备场景函数，共五页。第一页，第二页，第三页，第四页，第五页
CombineRgn 将两个区域组合为一个新区域
CombineTransform 驱动世界转换。它相当于依顺序进行两次转换
CreateCompatibleDC 创建一个与特定设备场景一致的内存设备场景
CreateDC 为专门设备创建设备场景
CreateEllipticRgn 创建一个椭圆
CreateEllipticRgnIndirect 创建一个内切于特定矩形的椭圆区域
CreateIC 为专用设备创建一个信息场景
CreatePolygonRgn 创建一个由一系列点围成的区域
CreatePolyPolygonRgn 创建由多个多边形构成的区域。每个多边形都应是封闭的
CreateRectRgn 创建一个矩形区域
CreateRectRgnIndirect 创建一个矩形区域
CreateRoundRectRgn 创建一个圆角矩形
DeleteDC 删除专用设备场景或信息场景，释放所有相关窗口资源
DPtoLP 将点阵从设备坐标转换到专用设备场景逻辑坐标
EqualRgn 确定两个区域是否相等
ExcludeClipRect 从专用设备场景的剪裁区中去掉一个矩形区。矩形内不能进行绘图
AddFontResource
AddFontResource

VB 声明
Declare Function AddFontResource Lib "gdi32" Alias "AddFontResourceA" (ByVal lpFileName As String) As Long
说明
在 Windows 系统中添加一种字体资源。添加完毕后，该字体即可由任何 Windows 应用程序调用
返回值
Long，添加的字体数量，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpFileNameString，字体资源文件的文件名。可以是.FON， .FNT， .TTF 或 .FOT 文件
注解
添加了一种资源后必须调用下述 API 函数：
di% = SendMessageBynum(HWND_BROADCAST, WM_FONTCHANGE, x, y)
其中，HWND_BROADCAST、WM_FONTCHANGE 使用来自 API32.TXT 文件的值。这样便可告诉所有 Windows 应用程序字体列表已发生了变化
示例
Call AddFontResource("myfont.ttf")
Call SendMessage(HWND_BROADCAST, WM_FONTCHANGE, 0, 0)

Top
CombineRgn
CombineRgn

VB 声明
Declare Function CombineRgn Lib "gdi32" Alias "CombineRgn" (ByVal hDestRgn As Long, ByVal hSrcRgn1 As Long, ByVal hSrcRgn2 As Long, ByVal nCombineMode As Long) As Long
说明
将两个区域组合为一个新区域
返回值
Long，下列常数之一：
COMPLEXREGION：区域有互相交叠的边界
SIMPLEREGION：区域边界没有互相交叠
NULLREGION：区域为空
ERRORAPI：不能创建组合区域
参数表
参数类型及说明
hDestRgnLong，包含组合结果的区域句柄
hSrcRgn1Long，源区域 1
hSrcRgn2Long，源区域 2
nCombineModeLong，组合两区域的方法。可设为下述常数
RGN_ANDhDestRgn 被设置为两个源区域的交集
RGN_COPYhDestRgn 被设置为 hSrcRgn1 的拷贝
RGN_DIFFhDestRgn 被设置为 hSrcRgn1 中与 hSrcRgn2 不相交的部分
RGN_ORhDestRgn 被设置为两个区域的并集
RGN_XORhDestRgn 被设置为除两个源区域 OR 之外的部分

Top
CombineTransform
CombineTransform

VB 声明
Declare Function CombineTransform Lib "gdi32" Alias "CombineTransform" (lpxformResult As xform, lpxform1 As xform, lpxform2 As xform) As Long
说明
驱动世界转换。它相当于依顺序进行两次转换
返回值
Long，执行成功为 TRUE（非零），失败则为零
参数表
参数类型及说明
lpxformResultxform，保存转换结果的结构
lpxform1xform，按顺序的第一个结构
xformxform，按顺序的第二个结构
适用平台
Windows NT

Top
CreateCompatibleDC

CreateCompatibleDC

VB 声明
Declare Function CreateCompatibleDC Lib "gdi32" Alias "CreateCompatibleDC" (ByVal hdc As Long) As Long
说明
创建一个与特定设备场景一致的内存设备场景
返回值
Long，新设备场景句柄，若出错则为零
参数表
参数类型及说明
hdcLong，设备场景句柄。新的设备场景将与它一致。也可能为 0 以创建一个与屏幕一致的设备场景
注解
在绘制之前，先要为该设备场景选定一个位图。不再需要时，该设备场景可用 DeleteDC 函数删除。删除前，其所有对象应回复初始状态

Top
CreateDC
CreateDC, CreateDCBynum

VB 声明
Declare Function CreateDC& Lib "gdi32" Alias "CreateDCA" (ByVal lpDriverName As String, ByVal lpDeviceName As String, ByVal lpOutput As String, lpInitData As DEVMODE)
Declare Function CreateDCBynum& Lib "gdi32" Alias "CreateDCA" (ByVal lpDriverName As String, ByVal lpDeviceName As String, ByVal lpOutput As String, lpInitData As Long)
说明
为专门设备创建设备场景
返回值
Long，新设备场景句柄，若出错则为零
参数表
参数类型及说明
lpDriverNameString，用 vbNullString 传递 null 值给该参数，除非：1、用 DISPLAY，是获取整个屏幕的设备场景；2、用 WINSPOOL，则是访问打印驱动
lpDeviceNameString，所用专门设备的名称。该名由打印管理器分配显示
lpOutputString，用 vbNullString 传递 null 值给该参数
lpInitDataDEVMODE，这个结构保存初始值。用 CreateDCBynum 传递 0（NULL）值则适用默认设置
注解
在绘制之前，先要为该设备场景选定一个位图。不再需要时，该设备场景可用 DeleteDC 函数删除。删除前，其所有对象应回复初始状态。若有设备初始设置可用 DocumentProperties API 函数载入 DEVMODE 结构。使用屏幕设备场景（DISPLAY）时要小心，因为它会干扰其他应用程序的外观
示例：靠近屏幕左上角画一个矩形

dc& = CreateDCBynum("DISPLAY", vbNullString, vbNullString, 0)
dl& = Rectangle(dc&, 5, 5, 100, 100)

Top
CreateEllipticRgn
CreateEllipticRgn

VB 声明
Declare Function CreateEllipticRgn Lib "gdi32" Alias "CreateEllipticRgn" (ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long) As Long
说明
创建一个椭圆，该椭圆与 X1，Y1 和 X2，Y2 坐标点确定的矩形内切
返回值
Long，执行成功则为区域句柄，失败则为零
参数表
参数类型及说明
X1,Y1Long，矩形左上角 X，Y 坐标
X2,Y2Long，矩形右下角 X，Y 坐标
注解
不用时一定要用 DeleteObject 函数删除区域。用 Ellipse API 函数绘出的椭圆与该椭圆区域不完全相同，因为本函数的绘图计算不包括矩形的下边和右边

Top
CreateEllipticRgnIndirect
CreateEllipticRgnIndirect

VB 声明
Declare Function CreateEllipticRgnIndirect Lib "gdi32" Alias "CreateEllipticRgnIndirect" (lpRect As Rect) As Long
说明
创建一个内切于特定矩形的椭圆区域
返回值
Long，执行成功则返回区域句柄，失败则为零
参数表
参数类型及说明
lpRectLong，定义要创建的椭圆区域尺寸的矩形
注解
不用时一定要用 DeleteObject 函数删除该区域

Top
CreateFont
CreateFont

VB 声明

Declare Function CreateFont Lib "gdi32" Alias "CreateFontA" (ByVal H As Long, ByVal W As Long, ByVal E As Long, ByVal O As Long, ByVal W As Long, ByVal I As Long, ByVal u As Long, ByVal S As Long, ByVal C As Long, ByVal OP As Long, ByVal CP As Long, ByVal Q As Long, ByVal PAF As Long, ByVal F As String) As Long
说明
用指定的属性创建一种逻辑字体
返回值
Long，执行成功则返回逻辑字体的句柄，零表示失败。会设置 GetLastError
参数表
参数类型及说明
HLong，IfHeight
WLong，IfWidth
ELong，IfEscapement
OLong，IfOrientation
WLong，IfWeight
ILong，IfItalic
uLong，IfUnderline
SLong，IfStrikeOut
CLong，IfCharSet
OPLong，IfOutputPrecision
CPLong，IfClipPrecision
QLong，IfQuality
PAFLong，IfPitchAndFamily
FString，IfFaceName
注解
VB 的字体属性在选择字体的时候显得更有效

Top
CreateFontIndirect
CreateFontIndirect

VB 声明
Declare Function CreateFontIndirect Lib "gdi32" Alias "CreateFontIndirectA" (lpLogFont As LOGFONT) As Long
说明
用指定的属性创建一种逻辑字体
返回值
Long，执行成功则返回逻辑字体句柄，零表示失败
参数表
参数类型及说明
lpLogFontLOGFONT，这个结构定义了逻辑字体请求的属性
注解
VB 的字体属性在选择字体的时候显得更有效

Top
CreateIC
CreateIC

VB 声明
Declare Function CreateIC Lib "gdi32" Alias "CreateICA" (ByVal lpDriverName As String, ByVal lpDeviceName As String, ByVal lpOutput As String, lpInitData As DEVMODE) As Long
说明
为专用设备创建一个信息场景。信息场景可用来快速获取某设备的信息而无须创建设备场景这样的系统开销。它可作为参数传递给 GetDeviceCaps 一类的信息函数以替代设备场景参数
返回值
Long，执行成功为信息场景句柄，失败则为零
参数表
参数类型及说明
lpDriverNameString，用 vbNullString 传递 null 值给该参数，除非：1、用 DISPLAY，是获取整个屏幕的设备场景；2、用 WINSPOOL，则是访问打印驱动
lpDeviceNameString，所用专门设备的名称。该名由打印管理器分配显示
lpOutputString，用 vbNullString 传递 null 值给该参数
lpInitDataDEVMODE，这个结构保存初始值
注解
Long，不用时一定要用 DeleteDC 函数删除设备场景。进一步的说明参考 CreateDC 函数
示例：为一个名为“Color Stylus”的打印机取回信息场景
dc& = CreateICBynum("WINSPOOL", "Color Stylus", vbNullString,0)

Top
CreatePolygonRgn
CreatePolygonRgn

VB 声明
Declare Function CreatePolygonRgn Lib "gdi32" Alias "CreatePolygonRgn" (lpPoint As POINTAPI, ByVal nCount As Long, ByVal nPolyFillMode As Long) As Long
说明
创建一个由一系列点围成的区域。windows 在需要时自动将最后点与第一点相连以封闭多边形
返回值
Long，执行成功为创建的区域句柄，失败则为 0
参数表
参数类型及说明
lpPointPOINTAPI，nCount 个 POINTAPI 结构中的第一个 POINTAPI 结构
nCountLong，多边形的点数
nPolyFillModeLong，描述多边形填充模式。可为 ALTERNATE 或 WINDING 常数。参考 SetPolyFillMode 函数对多边形填充模式的解释
注解
不用时一定要用 DeleteObject 函数删除该区域

Top
CreatePolyPolygonRgn
CreatePolyPolygonRgn

VB 声明
Declare Function CreatePolyPolygonRgn Lib "gdi32" Alias "CreatePolyPolygonRgn" (lpPoint As POINTAPI, lpPolyCounts As Long, ByVal nCount As Long, ByVal nPolyFillMode As Long) As Long
说明
创建由多个多边形构成的区域。每个多边形都应是封闭的
返回值
Long，执行成功则为创建区域的句柄，失败则为零
参数表
参数类型及说明
lpPointPOINTAPI，nCount 个 POINTAPI 结构中的第一个 POINTAPI 结构
lpPolyCountsLong，长整数阵列的第一个入口。每个入口包含构成一个封闭多边形的点数。lpPoint 阵列组成了一系列多边形，每个多边形在 lpPolyCounts 中有一个入口
nCountLong，多边形的点数
nPolyFillModeLong，描述多边形填充模式。可为 ALTERNATE 或 WINDING 常数。参考 SetPolyFillMode 函数对多边形填充模式的解释
注解
不用时一定要用 DeleteObject 函数删除该区域

Top
CreateRectRgn
CreateRectRgn

VB 声明
Declare Function CreateRectRgn Lib "gdi32" Alias "CreateRectRgn" (ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long) As Long
说明
创建一个由点 X1，Y1 和 X2，Y2 描述的矩形区域
返回值
Long，执行成功为区域句柄，失败则为零
参数表
参数类型及说明
X1,Y1Long，矩形左上角 X，Y 坐标
X2,Y2Long，矩形右下角 X，Y 坐标
注解
不用时一定要用 DeleteObject 函数删除该区域
这个矩形的下边和右边不包含在区域之内

Top

CreateRectRgnIndirect
CreateRectRgnIndirect

VB 声明
Declare Function CreateRectRgnIndirect Lib "gdi32" Alias "CreateRectRgnIndirect" (lpRect As) As Long
说明
创建一个由 lpRect 确定的矩形区域
返回值
Long，执行成功则为区域句柄，失败则为 0
参数表
参数类型及说明
lpRectRECT，要用来创建区域的矩形
注解
参考 CreateRectRgn 的注解

Top
CreateRoundRectRgn
CreateRoundRectRgn

VB 声明
Declare Function CreateRoundRectRgn Lib "gdi32" Alias "CreateRoundRectRgn" (ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long, ByVal X3 As Long, ByVal Y3 As Long) As Long
说明
创建一个圆角矩形，该矩形由 X1，Y1-X2，Y2 确定，并由 X3，Y3 确定的椭圆描述圆角弧度
返回值
Long，执行成功则为区域句柄，失败则为 0
参数表
参数类型及说明
X1,Y1Long，矩形左上角的 X，Y 坐标
X2,Y2Long，矩形右下角的 X，Y 坐标
X3Long，圆角椭圆的宽。其范围从 0（没有圆角）到矩形宽（全圆）
Y3Long，圆角椭圆的高。其范围从 0（没有圆角）到矩形高（全圆）
注解
不用时一定要用 DeleteObject 函数删除该区域
用该函数创建的区域与用 RoundRect API 函数画的圆角矩形不完全相同，因为本矩形的右边和下边不包括在区域之内

Top
CreateScalableFontResource
CreateScalableFontResource

VB 声明
Declare Function CreateScalableFontResource Lib "gdi32" Alias "CreateScalableFontResourceA" (ByVal fHidden As Long, ByVal lpszResourceFile As String, ByVal lpszFontFile As String, ByVal lpszCurrentPath As String) As Long
说明
为一种 TureType 字体创建一个资源文件，以便能用 API 函数 AddFontResource 将其加入 Windows 系统。字体信息本身并不复制到字体资源文件中；相反，资源文件中包含了欲使用的 TureType 文件的名字
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
fHiddenLong，如果是零，表示创建一个普通的字体资源；如果是 1，表示创建一个只读字体资源，它只能在文档中嵌入使用
lpszResourceFileString，欲创建的资源文件的名字。普通文件使用.FOT 扩展名，只读文件使用.FOR 扩展名
lpszFontFileString，TureType 字体文件文件的文件名。如果其中包含了一个路径，就到指定的路径寻找字体文件，同时不使用 lpszCurrentPath 参数指定的位置。而且在调用 AddFontResource 函数之前，会将字体复制到 Windows 的 SYSTEM 目录
lpszCurrentPathString，由 lpszFontFile 参数决定

Top
DeleteDC
DeleteDC

VB 声明
Declare Function DeleteDC Lib "gdi32" Alias "DeleteDC" (ByVal hdc As Long) As Long
说明
删除专用设备场景或信息场景，释放所有相关窗口资源。不要将它用于 GetDC 函数取回的设备场景
返回值
Long，执行成功则为非零，失败则为零
参数表
参数类型及说明
hdcLong，将要删除的设备场景
注解
若有对象被选入设备场景，则在调用本函数前应将它们选出。为此，可将初始对象回选入 DC，也可用 SaveDC， RestoreDC 函数对回复 DC 为其创建时的状态
在 vb 里使用
不要将它用于由 vb hdc 属性获取的设备场景句柄

Top
DPtoLP
DPtoLP

VB 声明
Declare Function DPtoLP Lib "gdi32" Alias "DPtoLP" (ByVal hdc As Long, lpPoint As POINTAPI, ByVal nCount As Long) As Long
说明
将点阵从设备坐标转换到专用设备场景逻辑坐标
返回值
Long，执行成功为非零值，失败则为零
参数表
参数类型及说明
hdcLong，确定逻辑坐标系统的设备场景句柄
lpPointPOINTAPI，包含有设备坐标点的一个或多个 POINTAPI 结构的第一入口。每个入口都将被转换为逻辑坐标（若为世界转换则转换为世界坐标）
nCountLong，lpPoint 阵列中的入口数

Top
DrawText
DrawText

VB 声明
Declare Function DrawText Lib "user32" Alias "DrawTextA" (ByVal hdc As Long, ByVal lpStr As String, ByVal nCount As Long, lpRect As RECT, ByVal wFormat As Long) As Long
说明
将文本描绘到指定的矩形中
返回值
Long，描绘文字的高度
参数表
参数类型及说明
hdcLong，欲在其中显示文字的一个设备场景的句柄
lpStrString，欲描绘的文本字串
nCountLong，欲描绘的字符数量。如果要描绘整个字串（直到空中止符），则可将这个参数设为-1
lpRectRECT，指定用于绘图的一个格式化矩形（采用逻辑坐标）
wFormatLong，一个标志位数组，决定了以何种形式执行绘图。参考下面总结的常数类型列表
标志常数说明
DT_BOTTOM 必须同时指定 DT_SINGLE。指示文本对齐格式化矩形的底边
DT_CALCRECT 象下面这样计算格式化矩形：多行绘图时矩形的底边根据需要进行延展，以便容下所有文字；单行绘图时，延展矩形的右侧。不描绘文字。由 lpRect 参数指定的矩形会载入计算出来的值
DT_CENTER 文本垂直居中
DT_EXPANDTABS 描绘文字的时候，对制表站进行扩展。默认的制表站间距是 8 个字符。但是，可用 DT_TABSTOP 标志改变这项设定
DT_EXTERNALLEADING 计算文本行高度的时候，使用当前字体的外部间距属性（the

external leading attribute）
DT_LEFT 文本左对齐
DT_NOCLIP 描绘文字时不剪切到指定的矩形
DT_NOPREFIX 通常，函数认为 & 字符表示应为下一个字符加上下划线。该标志禁止这种行为
DT_RIGHT 文本右对齐
DT_SINGLELINE 只画单行
DT_TABSTOP 指定新的制表站间距，采用这个整数的高 8 位
DT_TOP 必须同时指定 DT_SINGLE。指示文本对齐格式化矩形的顶部
DT_VCENTER 必须同时指定 DT_SINGLE。指示文本对齐格式化矩形的中部
DT_WORDBREAK 进行自动换行。如用 SetTextAlign 函数设置了 TA_UPDATECP 标志，这里的设置则无效

Top
DrawTextEx
DrawTextEx

VB 声明
Declare Function DrawTextEx Lib "user32" Alias "DrawTextExA" (ByVal hDC As Long, ByVal lpsz As String, ByVal n As Long, lpRect As RECT, ByVal un As Long, lpDrawTextParams As DRAWTEXTPARAMS) As Long
说明
与 DrawText 相似，只是加入了更多的功能
返回值
Long，描绘文字的高度
参数表
参数类型及说明
hDCLong，要在其中绘图的一个设备场景的句柄
lpszString，欲描绘的文本字串
nLong，欲描绘的字符数量。如果要描绘整个字串（直到空中止符），则可将这个参数设为-1
lpRectRECT，指定用于绘图的一个格式化矩形（采用逻辑坐标）
unLong，一个标志位。决定了以何种形式执行绘图。参考 DrawText 的 wFormat 参数和下表。其中下表列出的是新增的常数
标志常数说明
DT_EDITCONTROL 对一个多行编辑控件进行模拟。不显示部分可见的行
DT_ENDELLIPSES 倘若字串不能在矩形里全部容下，就在末尾显示省略号
DT_PATHELLIPSES 如字串包含了 \ 字符，就用省略号替换字串内容，使其能在矩形中全部容下。例如，一个很长的路径名可能换成这样显示——c:\windows\...\doc\readme.txt
DT_MODIFYSTRING 如指定了 DT_ENDELLIPSES 或 DT_PATHELLIPSES，就会对字串进行修改，使其与实际显示的字串相符
DT_RTLREADING 如选入设备场景的字体属于希伯来或阿拉伯语系，就从右到左描绘文字
lpDrawTextParamsDRAWTEXTPARAMS，这个结构包含了附加的绘图参数

Top

EnumFontFamilies
EnumFontFamilies

VB 声明
Declare Function EnumFontFamilies Lib "gdi32" Alias "EnumFontFamiliesA" (ByVal hdc As Long, ByVal lpszFamily As String, ByVal lpEnumFontFamProc As Long, ByVal lParam As Long) As Long
说明
列举指定设备可用的字体
返回值
Long，由回调函数返回的前一个值
参数表
参数类型及说明
hdcLong，设备场景的句柄
lpszFamilyString，欲枚举的字体家族。如指定 vbNullString，可枚举出每种可用字体家族中的一种字体
lpEnumFontFamProcLong，欲调用的函数地址。这个地址是用 AddressOf 运算符为来自一个标准模块的函数进行操作，或者利用某个回调控件得到
lParamLong，指定希望传递给回调函数的一个用户自定义值
注解
这个函数取代了 API 函数 EnumFonts，因为它能对 TureType 字体样式说明进行控制
只有实际存在的字体才会列举出来，那些可由 GDI 合成的字体不会列出

Top
EnumFontFamiliesEx
EnumFontFamiliesEx

VB 声明
Declare Function EnumFontFamiliesEx Lib "gdi32" Alias "EnumFontFamiliesExA" (ByVal hdc As Long, lpLogFont As LOGFONT, ByVal lpEnumFontProc As Long, ByVal lParam As Long, ByVal dw As Long) As Long
说明
根据一个 LOGFONT 结构提供的信息，列举指定设备可用的字体
返回值
Long，由回调函数返回的前一个值
参数表
参数类型及说明
hdcLong，设备场景的句柄
lpLogFontLOGFONT，这个结构指定了欲枚举的字体。此时用到的字段包括：lfCharSet，lfFaceName 和 lfPitchAndFamily。其他所有字段都会忽略
lpEnumFontFamProcLong，欲调用的函数地址。这个地址是用 AddressOf 运算符为来自一个标准模块的函数进行操作，或者利用某个回调控件得到
lParamLong，指定希望传递给回调函数的一个用户自定义值
dwLong，保留，设为零

注解
参见 EnumFontFamilies 函数的注解

Top
EqualRgn
EqualRgn

VB 声明
Declare Function EqualRgn Lib "gdi32" Alias "EqualRgn" (ByVal hSrcRgn1 As Long, ByVal hSrcRgn2 As Long) As Long
说明
确定两个区域是否相等
返回值
Long，若两区域相等为非零值。若不等为 0。若有一个区域无效则返回 ERRORAPI
参数表
参数类型及说明
hSrcRgn1Long，一个区域的句柄
hSrcRgn2Long，区域句柄

Top
ExcludeClipRect
ExcludeClipRect

VB 声明
Declare Function ExcludeClipRect Lib "gdi32" Alias "ExcludeClipRect" (ByVal hdc As Long, ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long) As Long
说明
从专用设备场景的剪裁区中去掉一个由点 X1，Y1 和 X2，Y2 确定的矩形区。矩形内不能进行绘图
返回值
Long，返回以下常数之一以描述所得剪裁区：
COMPLEXREGION：区域边界互相交叠
SIMPLEREGION：区域边界没有互相交叠
NULLREGION：区域为空
ERRORAPI：发生了错误
参数表
参数类型及说明
hdcLong，要修改的设备场景
X1,Y1Long，矩形（逻辑坐标）左上角的 X，Y 坐标
X2,Y2Long，矩形（逻辑坐标）右下角的 X，Y 坐标，矩形的右边和下边不会从剪裁区中移走

Top
ExcludeUpdateRgn

ExcludeUpdateRgn

VB 声明
Declare Function ExcludeUpdateRgn Lib "user32" Alias "ExcludeUpdateRgn" (ByVal hdc As Long, ByVal hwnd As Long) As Long
说明
从专用设备场景剪裁区去掉指定窗口的刷新区域。这防止对无效区绘图（该无效区将推迟一点再刷新）
返回值
Long，见 ExcludeClipRect 函数返回值
参数表
参数类型及说明
hdcLong，设备场景，其剪裁区将被修改以排除 hwnd 窗口的刷新区
hwndLong，窗口句柄
在 vb 中使用
设您要完成一个较长或复杂的绘图操作，该函数可允许您避免画到现在没有遮盖而随后即将画的区域。这可提高性能并消除闪烁效果

Top
ExtCreateRegion
ExtCreateRegion

VB 声明
Declare Function ExtCreateRegion Lib "gdi32" Alias "ExtCreateRegion" (lpXform As xform, ByVal nCount As Long, lpRgnData As RGNDATA) As Long
说明
根据世界转换修改区域
返回值
Long，执行成功为已转换区域的句柄，失败则为 0
参数表
参数类型及说明
lpXformxform，将用于由 lpRgnData 指定区域的转换
nCountLong，lpRgnData 数据结构或缓冲区的字节数
lpRgnDataRGNDATA，用 GetRegionData 定义区域时载入的一个结构
注解
在 win95 下，只有缩放和平移转换才能用该函数。它不支持旋转和剪切

Top
ExtSelectClipRgn
ExtSelectClipRgn

VB 声明
Declare Function ExtSelectClipRgn Lib "gdi32" Alias "ExtSelectClipRgn" (ByVal hdc As Long, ByVal hRgn As Long, ByVal fnMode As Long) As Long

说明
将指定区域组合到设备场景的当前剪裁区
返回值
Long，返回下列常数之一，以描述所得剪裁区：
COMPLEXREGION：区域边界互相交叠
SIMPLEREGION：区域边界没有互相交叠
NULLREGION：区域为空
ERRORAPI：发生了错误
参数表
参数类型及说明
hdcLong，剪裁区将被修改的设备场景的句柄
hRgnLong，源区域句柄。若 fnMode 为 RGN_COPY，它可为 NULL
fnModeLong，下列常数之一：
RGN_AND 新剪裁区包括即属 hRgn 又属当前剪裁区的部分
RGN_COPY 新剪裁区为 hRgn 区域
RGN_DIFF 新剪裁区为当前剪裁区除掉其在 hRgn 区内的部分
RGN_OR 新剪裁区包括属 hRgn 或当前属当前剪裁区的部分
RGN_XOR 新剪裁区包括属 hRgn 或当前属当前剪裁区的部分，但要除去同属两者的部分
注解
本函数对 hRgn 区域没有影响，执行后可毁去（destroy）该区域

Top
FillRgn
FillRgn

VB 声明
Declare Function FillRgn Lib "gdi32" Alias "FillRgn" (ByVal hdc As Long, ByVal hRgn As Long, ByVal hBrush As Long) As Long
说明
用指定刷子填充指定区域
返回值
Long，执行成功为非零值，失败则为 0
参数表
参数类型及说明
hdcLong，设备场景句柄
hRgnLong，以数据设备坐标填充的区域句柄
hBrushLong，要用的刷子的句柄

Top
FrameRgn
FrameRgn

VB 声明
Declare Function FrameRgn Lib "gdi32" Alias "FrameRgn" (ByVal hdc As Long, ByVal hRgn As

Long, ByVal hBrush As Long, ByVal nWidth As Long, ByVal nHeight As Long) As Long
说明
用指定刷子围绕指定区域画一个外框
返回值
Long，执行成功返回非零值，失败为 0
参数表
参数类型及说明
hdcLong，设备场景句柄
hRgnLong，将数据设备坐标填充的区域句柄
hBrushLong，将用的刷子句柄
nWidthLong，垂直边框宽度（以设备单元为单位）
nHeightLong，水平边框高度（以设备单元为单位）

Top
GetBoundsRect
GetBoundsRect

VB 声明
Declare Function GetBoundsRect Lib "gdi32" Alias "GetBoundsRect" (ByVal hdc As Long, lprcBounds As RECT, ByVal flags As Long) As Long
说明
获取指定设备场景的边界矩形。每个设备场景都有一个边界矩形，程序员可用它来堆放表示当前图象边界的信息
返回值
Long，出错为 0，否则为下列常数之一：
DCB_SET：边界矩形非空
DCB_RESET：边界矩形为空
DCB_ENABLE：边界矩形正被堆放
DCB_DISABLE：边界矩形当前没有被堆放
参数表
参数类型及说明
hdcLong，边界矩形对应的设备场景
lprcBoundsRECT，装载设备场景 hdc 的当前边界矩形
flagsLong，可设为常数 DCB_RESET 以清除边界矩形，否则设为零
注解
详情参考 SetBoundsRect

Top
GetClipBox
GetClipBox

VB 声明
Declare Function GetClipBox Lib "gdi32" Alias "GetClipBox" (ByVal hdc As Long, lpRect As RECT) As Long

说明
获取完全包含指定设备场景剪裁区的最小矩形
返回值
Long，返回下列常数之一，以描述所得剪裁区：
COMPLEXREGION：区域边界互相交叠
SIMPLEREGION：区域边界没有互相交叠
NULLREGION：区域为空
ERRORAPI：发生了错误
参数表
参数类型及说明
hdcLong，设备场景句柄
lpRectRECT，装载包含设备场景剪裁区的矩形

Top
GetClipRgn
GetClipRgn

VB 声明
Declare Function GetClipRgn Lib "gdi32" Alias "GetClipRgn" (ByVal hdc As Long, ByVal hRgn As Long) As Long
说明
获取设备场景当前剪裁区
返回值
Long，下列常数之一：
1——执行成功，剪裁区存在
0——执行成功，没有定义剪裁区
-1——出错
参数表
参数类型及说明
hdcLong，想要访问的设备场景剪裁区
hRgnLong，已存在的区域句柄。该区域将被设为设备场景当前剪裁区
注解
hRgn 装载剪裁区拷贝。它后面的改变不影响剪裁区。该函数只能载入由应用程序设置的剪裁区，那些由 BeginPaint 一类函数设置的内部剪裁区不能被函数访问

Top
GetDC
GetDC

VB 声明
Declare Function GetDC Lib "user32" Alias "GetDC" (ByVal hwnd As Long) As Long
说明
获取指定窗口的设备场景
返回值

Long，指定窗口的设备场景句柄，出错则为 0
参数表
参数类型及说明
hwndLong，将获取其设备场景的窗口的句柄。若为 0，则要获取整个屏幕的 DC
注解
若窗口所属类具有 CS_OWNDC， CS_CLASSDC 或 CS_PARENTDC 样式，则获取的设备场景属窗口或类专有。vb 的窗体和图片框控件也是这种情况，它用该函数取得的结果和控件的 hdc 属性相同（在 autoredraw 为 FALSE 时）。您无须考虑取回的窗体或图片框控件设备场景的默认状态，特别是绘图对象。另外，默认状态随着窗体和控件 autoredraw 属性的设置而不同。在设备场景释放前您必须回复其状态为初始值。对于没有 CS_OWNDC，CS_CLASSDC 或 CS_PARENTDC 样式的窗口的设备场景，可从通用 windows 缓存中获取，其状态为默认值。缓存中可用设备场景数量是有限的，因此只要可能就释放设备场景
用本函数获取的设备场景一定要用 ReleaseDC 函数释放，不能用 DeleteDC

Top
GetDCEx
GetDCEx

VB 声明
Declare Function GetDCEx Lib "user32" Alias "GetDCEx" (ByVal hwnd As Long, ByVal hrgnclip As Long, ByVal fdwOptions As Long) As Long
说明
为指定窗口获取设备场景。相比 GetDC，本函数提供了更多的选项
返回值
Long，执行成功为指定窗口设备场景句柄。出错则为 0
参数表
参数类型及说明
hwndLong，窗口句柄
hrgnclipLong，窗口剪裁区
fdwOptionsLong，标志字。根据下列常数设置各位：
DCX_CACHE 不管窗口类的样式，从 windows 缓存获取设备场景
DCX_CLIPCHILREN 所有可见的子窗口区都要从 DC 的剪裁区中排除
DCX_CLIPSIBLINGS 窗口 hWnd 上的所有可见兄弟窗口都要从 DC 的剪裁区中排除
DCX_EXCLUDERGN 从 DC 剪裁区中排除由 hrgnclip 指定的区域
DCX_EXCLUDEUPDATE 从设备场景剪裁区中排除刷新区域
DCX_INTERSECTRGN 由 hrgnclip 指定的区域与设备场景剪裁区相交
DCX_INTERSECTUPDATE 指定区域与设备场景刷新区域相交
DCX_LOCKWINDOWUPDATE 该标志为允许向窗口绘图，即使它由于 LockWindowUpdate 的调用被锁住
DCX_NORESETATTRS 设备场景释放后不被重置为默认状态
DCX_PARENTCLIP 放弃 CS_PARENTDC 类样式设置。DC 的起点设为 hWnd 窗口的左上角
DCX_WINDOWA device context is returned for the entire window rectangle rather than just the client area of the window
DCX_VALIDATECombine with DCX_INTERSECTUPDATE, validates the clipping region

注解
若窗口所属类具有 CS_OWNDC， CS_CLASSDC 或 CS_PARENTDC 样式，则获取的设备场景属窗口或类专有。这时，设备场景状态不能从初值修改。vb 的窗体和控件通常是这种情况。否则，置 DCX_CACHE 位以从通用 windows 缓冲区恢复设备场景。若不置该位，则函数返回 0。DC 的状态位默认设置。从缓存获取的设备场景用过后要用 ReleaseDC 函数释放以防止系统死锁，因为 windows 只有 5 个缓存 DC 可用
其他情况参见 GetDC 函数注解

Top
GetDCOrgEx
GetDCOrgEx

VB 声明
Declare Function GetDCOrgEx Lib "gdi32" Alias "GetDCOrgEx" (ByVal hdc As Long, lpPoint As POINTAPI) As Long
说明
获取指定设备场景起点位置（以屏幕坐标表示）。例如，设 DC 起点为窗口客户区左上角，则该函数返回值为该角在屏幕中的位置（以屏幕像素坐标表示）
返回值
Long，执行成功为非零值，否则为零
参数表
参数类型及说明
hdcLong，设备场景句柄
lpPointPOINTAPI，装载设备场景起点的屏幕坐标
示例：在窗体模块中使用时，该代码装载窗体客户区左上角坐标（以屏幕坐标表示）。它也是窗体 hDC 属性的起点
```
Dim dl&
Dim pt As POINTAPI
dl& = GetDCOrgEx(hDC, pt)
```

Top
GetDeviceCaps
GetDeviceCaps

VB 声明
Declare Function GetDeviceCaps Lib "gdi32" Alias "GetDeviceCaps" (ByVal hdc As Long, ByVal nIndex As Long) As Long
说明
根据指定设备场景代表的设备的功能返回信息
返回值
Long，参见 GetDeviceCaps 索引表
参数表
参数类型及说明
hdcLong，要查询其设备的信息的设备场景

nIndexLong，根据 GetDeviceCaps 索引表所示常数确定返回信息的类型

Top
GetGraphicsMode
GetGraphicsMode

VB 声明
Declare Function GetGraphicsMode Lib "gdi32" Alias "GetGraphicsMode" (ByVal hdc As Long) As Long
说明
确定是否允许增强图形模式（世界转换）
返回值
Long，下列常数之一：
GM_COMPATIBLE：图形模式兼容 win16
GM_ADVANCED：允许世界转换
参数表
参数类型及说明
hdcLong，其模式将被测试的设备场景
注解
只有 Windows NT 支持世界转换

Top
GetMapMode
GetMapMode

VB 声明
Declare Function GetMapMode Lib "gdi32" Alias "GetMapMode" (ByVal hdc As Long) As Long
说明
为特定设备场景调入映象模式
返回值
Long，当前设备场景的映象模式。关于映象模式的说明参见 SetMapMode 函数
参数表
参数类型及说明
hdcLong，其模式将被测试的设备场景

Top
GetRegionData
GetRegionData

VB 声明
Declare Function GetRegionData Lib "gdi32" Alias "GetRegionDataA" (ByVal hRgn As Long, ByVal dwCount As Long, lpRgnData As RgnData) As Long
说明
装入描述一个区域信息的 RgnData 结构或缓冲区

返回值
Long，如果结构足够大以装入区域的数据，返回 1；出错时返回 0。如果 lpRgnData 不够大，不能装入区域数据，则返回需要的结构大小
参数表
参数类型及说明
hRgnLong，包含信息的区域的句柄
dwCountLong，RgnData 结构的大小
lpRgnDataRgnData，这个结构用以装入区域信息
注解
RgnData 是一个描述区域的定长结构。Buffer 是存放区域数据的缓冲区。缓冲区实际需要的大小取决于区域的复杂程度（显然，1 字节是永远不够的）。有两个选择：
1、将 RgnData 重定义为永远不会用到的一个大尺寸。这是需要的，因为 vb 不允许动态重定义结构的大小
2、分配一个字节数组并用它来代替 RgnData 结构。这要求将 As RgnData 换为 As Byte 来改变函数的 API 声明，并且传送字节数组的第一个元素
如果以后要访问 RGNDATAHEADER 结构的元素，需要用一个内存拷贝例程将数据从缓冲区拷贝到一个特别定义的 RGNDATAHEADER 结构中

Top
GetRgnBox
GetRgnBox

VB 声明
Declare Function GetRgnBox Lib "gdi32" Alias "GetRgnBox" (ByVal hRgn As Long, lpRect As RECT) As Long
说明
获取完全包含指定区域的最小矩形
返回值
Long，下列常数之一，以描述当前剪裁区：
COMPLEXREGION：该区域有互相交叠的边界
SIMPLEREGION：该区域边界没有互相交叠
NULLREGION：区域为空
ERRORAPI：发生了错误
参数表
参数类型及说明
hRgnLong，区域句柄
lpRectRECT，矩形结构，装载完全包含指定区域的矩形

Top
GetUpdateRgn
GetUpdateRgn

VB 声明
Declare Function GetUpdateRgn Lib "user32" Alias "GetUpdateRgn" (ByVal hwnd As Long,

ByVal hRgn As Long, ByVal fErase As Long) As Long
说明
确定指定窗口的刷新区域。该区域当前无效，需要刷新
返回值
Long，下列常数之一，以描述当前剪裁区：
COMPLEXREGION：该区域有互相交叠的边界
SIMPLEREGION：该区域边界没有互相交叠
NULLREGION：区域为空
ERRORAPI：发生了错误
参数表
参数类型及说明
hwndLong，将确定刷新区域的窗口的句柄
hRgnLong，装载 hwnd 窗口刷新区域的区域句柄
fEraseLong，为非零值表示窗口背景应擦除，客户区外的窗口部分是被重画
原文：TRUE (nonzero) to specify that the window background should be erased and parts of the window outside of the client area should be redrawn.

Top
GetViewportExtEx
GetViewportExtEx

VB 声明
Declare Function GetViewportExtEx Lib "gdi32" Alias "GetViewportExtEx" (ByVal hdc As Long, lpSize As SIZE) As Long
说明
获取设备场景视口（viewport）范围
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hdcLong，设备场景句柄
lpSizeSIZE，装载 DC 视口水平和垂直范围（以设备单元表示）的结构

Top
GetViewportOrgEx
GetViewportOrgEx

VB 声明
Declare Function GetViewportOrgEx Lib "gdi32" Alias "GetViewportOrgEx" (ByVal hdc As Long, lpPoint As POINTAPI) As Long
说明
获取设备场景视口起点
返回值
Long，执行成功为非零值，失败为 0

参数表
参数类型及说明
hdcLong，设备场景句柄
lpPointPOINTAPI，装载视口起点的结构

Top
GetWindowDC
GetWindowDC

VB 声明
Declare Function GetWindowDC Lib "user32" Alias "GetWindowDC" (ByVal hwnd As Long) As Long
说明
获取整个窗口（包括边框、滚动条、标题栏、菜单等）的设备场景
返回值
Long，执行成功为窗口设备场景，失败则为 0
参数表
参数类型及说明
hwndLong，将获取其设备场景的窗口
注解
不推荐在 vb 里使用这个函数。用完后一定要用 ReleaseDC 函数释放场景

Top
GetWindowExtEx
GetWindowExtEx

VB 声明
Declare Function GetWindowExtEx Lib "gdi32" Alias "GetWindowExtEx" (ByVal hdc As Long, lpSize As SIZE) As Long
说明
获取指定设备场景的窗口范围
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hdcLong，设备场景句柄
lpSizeSIZE，装载设备场景逻辑窗口水平和垂直范围（以逻辑单元表示）的结构

Top
GetWindowOrgEx
GetWindowOrgEx

VB 声明
Declare Function GetWindowOrgEx Lib "gdi32" Alias "GetWindowOrgEx" (ByVal hdc As Long,

lpPoint As POINTAPI) As Long
说明
获取指定设备场景的逻辑窗口的起点
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hdcLong，设备场景句柄
lpPointPOINTAPI，装载逻辑窗口起点的结构

Top
GetWindowRgn
GetWindowRgn

VB 声明
Declare Function GetWindowRgn Lib "user32" Alias "GetWindowRgn" (ByVal hWnd As Long, ByVal hRgn As Long) As Long
说明
根据以前 SetWindowRgn 函数的定义获取窗口区域
返回值
Long，下列常数之一，以描述当前剪裁区：
COMPLEXREGION：该区域有互相交叠的边界
SIMPLEREGION：该区域边界没有互相交叠
NULLREGION：区域为空
ERRORAPI：发生了错误
参数表
参数类型及说明
hWndLong，要获取区域的窗口
hRgnLong，区域句柄，若已设置有一个区域，则装载当前窗口区域的拷贝
注解
参考 SetWindowRgn 函数

Top
GetWorldTransform
GetWorldTransform

VB 声明
Declare Function GetWorldTransform Lib "gdi32" Alias "GetWorldTransform" (ByVal hdc As Long, lpXform As xform) As Long
说明
如果有世界转换，为设备场景获取当前世界转换
返回值
Long，执行成功为非零值，失败为 0
参数表

参数类型及说明
hdcLong，设备场景句柄
lpXformxform，装载设备场景当前世界转换的结构
适用平台
Windows NT

Top
IntersectClipRect
IntersectClipRect

VB 声明
Declare Function IntersectClipRect Lib "gdi32" Alias "IntersectClipRect" (ByVal hdc As Long, ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long) As Long
说明
为指定设备定义一个新的剪裁区，该区为当前剪裁区与由点 X1，Y1 和 X2，Y2 定义的矩形的交集
返回值
Long，下列常数之一，以描述当前剪裁区：
COMPLEXREGION：该区域有互相交叠的边界
SIMPLEREGION：该区域边界没有互相交叠
NULLREGION：区域为空
ERRORAPI：发生了错误
参数表
参数类型及说明
hdcLong，设备场景
X1,Y1Long，矩形左上角 X，Y 坐标
X2,Y2Long，矩形右下角 X，Y 坐标

Top
InvalidateRgn
InvalidateRgn

VB 声明
Declare Function InvalidateRgn Lib "user32" Alias "InvalidateRgn" (ByVal hwnd As Long, ByVal hRgn As Long, ByVal bErase As Long) As Long
说明
使窗口指定区域不活动，并将它加入窗口刷新区，使之可随后被重画
返回值
Long，TRUE
参数表
参数类型及说明
hwndLong，窗口句柄
hRgnLong，不活动区域句柄，该区域以窗口客户坐标定义。若该句柄为 0，则将使整个窗口客户区不活动

bEraseLong，为非零值则表示要在刷新前擦除
注解
若 bErase 为 TRUE，则刷新前整个刷新区都将被擦除而不只是定义的区域

Top
InvertRgn
InvertRgn

VB 声明
Declare Function InvertRgn Lib "gdi32" Alias "InvertRgn" (ByVal hdc As Long, ByVal hRgn As Long) As Long
说明
通过颠倒每个像素值反转设备场景指定区域
返回值
Long，执行成功为非零值，失败为零
参数表
参数类型及说明
hdcLong，设备场景句柄
hRgnLong，将反转的设备区域

Top
LPtoDP
LPtoDP

VB 声明
Declare Function LPtoDP Lib "gdi32" Alias "LPtoDP" (ByVal hdc As Long, lpPoint As POINTAPI, ByVal nCount As Long) As Long
说明
将点阵从指定设备场景逻辑坐标转换为设备坐标
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hdcLong，定义了逻辑坐标系统的设备场景句柄
lpPointPOINTAPI，包含逻辑坐标的点的一个或多个 POINTAPI 结构的第一入口，每个入口将被转换为设备坐标
nCountLong，lpPoint 阵列中的入口数
注解
若存在世界转换，lpPoint 阵列中的点则是以世界坐标表示的

Top
ModifyWorldTransform
ModifyWorldTransform

VB 声明
Declare Function ModifyWorldTransform Lib "gdi32" Alias "ModifyWorldTransform" (ByVal hdc As Long, lpXform As xform, ByVal iMode As Long) As Long
说明
根据指定的模式修改世界转换
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hdcLong，设备场景，其世界坐标将被修改
lpXformxform，这个结构内含根据 iMode 参数附加的转换
iModeLong，下列常数之一：
MWT_IDENTITY 设置世界转换为默认的特性转换（用该特性转换时，世界转换无效），忽略 lpXform 参数
MWT_LEFTMULTIPLYlpXform 转换被当前转换乘，乘积设为新的世界转换
MWT_RIGHTMULTIPLY 当前转换被 lpXform 转换乘，乘积设为新的世界转换
注解
与一般的乘法不同，矩阵乘法的结果与矩阵顺序有关。在 MWT_LEFTMULTIPLY 模式，lpXform 为左边的操作数。请参考高等代数相关书籍。
调用本函数前先应调用 SetGraphicsMode 函数设置 GM_ADVANCED 图形模式
适用平台
Windows NT

Top
OffsetClipRgn
OffsetClipRgn

VB 声明
Declare Function OffsetClipRgn Lib "gdi32" Alias "OffsetClipRgn" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long) As Long
说明
按指定量平移设备场景剪裁区
返回值
Long，下列常数之一，以描述当前剪裁区：
COMPLEXREGION：该区域有互相交叠的边界
SIMPLEREGION：该区域边界没有互相交叠
NULLREGION：区域为空
ERRORAPI：发生了错误
参数表
参数类型及说明
hdcLong，设备场景
xLong，以逻辑单元表示的水平偏移
yLong，以逻辑单元表示的垂直偏移

Top
OffsetRgn
OffsetRgn

VB 声明
Declare Function OffsetRgn Lib "gdi32" Alias "OffsetRgn" (ByVal hRgn As Long, ByVal x As Long, ByVal y As Long) As Long
说明
按指定偏移量平移指定区域
返回值
Long，下列常数之一，以描述当前剪裁区：
COMPLEXREGION：该区域有互相交叠的边界
SIMPLEREGION：该区域边界没有互相交叠
NULLREGION：区域为空
ERRORAPI：发生了错误
参数表
参数类型及说明
hRgnLong，区域句柄
xLong，以逻辑坐标表示的水平偏移量
yLong，以逻辑坐标表示的垂直偏移量

Top
OffsetViewportOrgEx
OffsetViewportOrgEx

VB 声明
Declare Function OffsetViewportOrgEx Lib "gdi32" Alias "OffsetViewportOrgEx" (ByVal hdc As Long, ByVal nX As Long, ByVal nY As Long, lpPoint As POINTAPI) As Long
说明
平移设备场景视口区域
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hdcLong，设备场景句柄
nX,nYLong，将加到视口起点的水平和垂直偏移量（以设备坐标表示）
lpPointPOINTAPI，装载设备场景视口原先的起点的结构，以设备坐标表示

Top
OffsetWindowOrgEx
OffsetWindowOrgEx

VB 声明
Declare Function OffsetWindowOrgEx Lib "gdi32" Alias "OffsetWindowOrgEx" (ByVal hdc As

Long, ByVal nX As Long, ByVal nY As Long, lpPoint As POINTAPI) As Long
说明
平移指定设备场景窗口起点
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hdcLong，设备场景句柄
nX,nYLong，以逻辑坐标表示的窗口起点
lpPointPOINTAPI，装载 DC 窗口原有的起点的的结构，以逻辑坐标表示

Top
PaintRgn
PaintRgn

VB 声明
Declare Function PaintRgn Lib "gdi32" Alias "PaintRgn" (ByVal hdc As Long, ByVal hRgn As Long) As Long
说明
用当前刷子背景色填充指定区域
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hdcLong，设备场景句柄
hRgnLong，将填充的区域句柄

Top
PtInRegion
PtInRegion

VB 声明
Declare Function PtInRegion Lib "gdi32" Alias "PtInRegion" (ByVal hRgn As Long, ByVal x As Long, ByVal y As Long) As Long
说明
确定点是否在指定区域内
返回值
Long，若点在区域内为非零值，否则为 0
参数表
参数类型及说明
hRgnLong，区域句柄
xLong，以逻辑坐标表示的点的 X 坐标
yLong，以逻辑坐标表示的点的 Y 坐标
在 vb 里使用

在测试复杂图象时非常有用

Top
PtVisible
PtVisible

VB 声明
Declare Function PtVisible Lib "gdi32" Alias "PtVisible" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long) As Long
说明
确定指定点是否可见（即，点是否在设备场景剪裁区内）
返回值
Long，若点在指定设备场景剪裁区内为非零值，否则为 0
参数表
参数类型及说明
hdcLong，设备场景句柄
xLong，要检查的点的 x,y 坐标
y

Top
RectInRegion
RectInRegion

VB 声明
Declare Function RectInRegion Lib "gdi32" Alias "RectInRegion" (ByVal hRgn As Long, lpRect As RECT) As Long
说明
确定矩形是否有部分在指定区域内
返回值
Long，若矩形有部分在指定区域内为非零值，否则为 0
参数表
参数类型及说明
hRgnLong，区域句柄
lpRectRECT，要测试的矩形结构
注解
矩形的底边和右边不包括在 lpRect 内

Top
RectVisible
RectVisible

VB 声明
Declare Function RectVisible Lib "gdi32" Alias "RectVisible" (ByVal hdc As Long, lpRect As RECT) As Long

说明
确定指定矩形是否有部分可见（是否在设备场景剪裁区内）
返回值
Long，矩形有部分在指定设备场景剪裁区内为非零值，否则为零
参数表
参数类型及说明
hdcLong，设备场景句柄
lpRectRECT，用于测试是否可见的矩形（用逻辑坐标）
注解
矩形的底边和右边不包括在 lpRect 内

Top
ReleaseDC
ReleaseDC

VB 声明
Declare Function ReleaseDC Lib "user32" Alias "ReleaseDC" (ByVal hwnd As Long, ByVal hdc As Long) As Long
说明
释放由调用 GetDC 或 GetWindowDC 函数获取的指定设备场景。它对类或私有设备场景无效（但这样的调用不会造成损害）
返回值
Long，执行成功为 1，否则为 0
参数表
参数类型及说明
hwndLong，要释放的设备场景相关的窗口句柄
hdcLong，要释放的设备场景句柄
注解
对那些用 CreateDC 一类的 DC 创建函数生成的设备场景，不要用本函数

Top
RestoreDC
RestoreDC

VB 声明
Declare Function RestoreDC Lib "gdi32" Alias "RestoreDC" (ByVal hdc As Long, ByVal nSavedDC As Long) As Long
说明
从设备场景堆栈恢复一个原先保存的设备场景
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hdcLong，将恢复的设备场景句柄

nSavedDCLong，将恢复的设备场景 ID 号，它由 SaveDC 函数返回。若为-1，则恢复最近的设备场景
注解
若 nSavedDC 所指的不是栈顶的 DC，则栈中 nSavedDC 之上的所有设备场景都要从栈中移出，详情参见 SaveDC 函数

Top
SaveDC
SaveDC

VB 声明
Declare Function SaveDC Lib "gdi32" Alias "SaveDC" (ByVal hdc As Long) As Long
说明
将指定设备场景状态保存到 Windows 设备场景堆栈。DC 的映射模式、窗口和视口缩放、剪裁区、所选对象列表都要保存。本函数通常在要对设备场景作临时改动时使用，DC 属性可用 RestoreDC 函数恢复
返回值
Long，一个代表该设备场景的整数，据此可用 RestoreDC 函数恢复。出错则为 0
参数表
参数类型及说明
hdcLong，要保存的设备场景句柄
注解
设备场景被保存到内部堆栈，这意味着可多次保存一个设备场景状态。例如，您可在进入某个函数时保存设备场景，在其中完成某些 api 调用，再保存一次，作更多的调用后，恢复前一状态，再作其他 api 函数调用，最后退出该函数前恢复原有设备场景状态。可保存的设备场景个数仅受限于内存大小

Top
ScaleViewportExtEx
ScaleViewportExtEx

VB 声明
Declare Function ScaleViewportExtEx Lib "gdi32" Alias "ScaleViewportExtEx" (ByVal hdc As Long, ByVal nXnum As Long, ByVal nXdenom As Long, ByVal nYnum As Long, ByVal nYdenom As Long, lpSize As SIZE) As Long
说明
缩放设备场景视口的范围
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hdcLong，设备场景句柄
nXnum,nYnumLong，当前 X，Y 范围将与本数值相乘
nXdenom,nYdenomLong，当前 X，Y 范围将被本数值除

lpSizeSIZE，装载原有水平和垂直视口范围的结构
注解
本函数仅在 MM_ISOTROPIC 和 MM_ANISOTROPIC 映射模式下有效，在 MM_ISOTROPIC 模式，要在设置视口范围之前设置窗口范围

Top
ScaleWindowExtEx
ScaleWindowExtEx

VB 声明
Declare Function ScaleWindowExtEx Lib "gdi32" Alias "ScaleWindowExtEx" (ByVal hdc As Long, ByVal nXnum As Long, ByVal nXdenom As Long, ByVal nYnum As Long, ByVal nYdenom As Long, lpSize As SIZE) As Long
说明
缩放指定设备场景窗口范围
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hdcLong，设备场景句柄
nXnum,nYnumLong，当前 X，Y 范围将与本数值相乘
nXdenom,nYdenomLong，当前 X，Y 范围将被本数值除
lpSizeSIZE，装载原有水平和垂直视口范围的结构
注解
本函数仅在 MM_ISOTROPIC 和 MM_ANISOTROPIC 映射模式下有效，在 MM_ISOTROPIC 模式，要在设置视口范围之前设置窗口范围

Top
ScrollDC
ScrollDC

VB 声明
Declare Function ScrollDC Lib "user32" Alias "ScrollDC" (ByVal hdc As Long, ByVal dx As Long, ByVal dy As Long, lprcScroll As RECT, lprcClip As RECT, ByVal hrgnUpdate As Long, lprcUpdate As RECT) As Long
说明
在窗口（由设备场景代表）中水平和（或）垂直滚动矩形
返回值
Long，执行成功为非零值，失败为 0，并设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景句柄
dxLong，将滚动的水平单位
dyLong，将滚动的垂直单位

lprcScrollRECT，要滚动的矩形
lprcClipRECT，定义剪裁区的矩形，滚动只能在该区内进行
hrgnUpdateLong，该区域设置为客户坐标系统中不被滚动操作覆盖的区，也可为 0，则表示没有设置刷新区域
lprcUpdateRECT，该矩形为客户坐标系统中不被滚动操作覆盖的区，也可为 0（要求修改 vb 声明使之能接收 lprcUpdate 为长整型参数）
注解
若 hrgnUpdate 和 lprcUpdate 都为 NULL，windows 将不能为该 DC 计算刷新区

Top
SelectClipRgn
SelectClipRgn

VB 声明
Declare Function SelectClipRgn Lib "gdi32" Alias "SelectClipRgn" (ByVal hdc As Long, ByVal hRgn As Long) As Long
说明
为指定设备场景选择新的剪裁区
返回值
Long，下列常数之一，以描述当前剪裁区：
COMPLEXREGION：该区域有互相交叠的边界
SIMPLEREGION：该区域边界没有互相交叠
NULLREGION：区域为空
ERRORAPI：发生了错误（保留原有剪裁区）
参数表
参数类型及说明
hdcLong，将设置新剪裁区的设备场景
hRgnLong，将为设备场景设置剪裁区的句柄，该区域使用设备坐标
注解
该函数拷贝剪裁区；这样同一剪裁区可用于多个设备场景，删除它不会影响每个设备场景的剪裁区。某些打印机文本和图形有不同的坐标系统，这时，应选用精度高的坐标系统——例如，点阵打印机用抖动算法，其文本精度比图形精度高，则应选用文本坐标系统
若使用 WM_PAINT 消息的子类，处理该消息使您用本函数增加的剪裁区不能超过窗口的不活动区

Top
SetBoundsRect
SetBoundsRect

VB 声明
Declare Function SetBoundsRect Lib "gdi32" Alias "SetBoundsRect" (ByVal hdc As Long, lprcBounds As RECT, ByVal flags As Long) As Long
说明
设置指定设备场景的边界矩形。每个设备场景都有一个边界矩形，编程者可用来堆放表示当

前图象边界的信息。当允许边界矩形堆放时，每次 windows 向设备场景绘图，绘图位置都与边界矩形比较，若有必要就要扩展边界矩形以包括绘图位置。这提供了一种方法可确定设备场景的哪些部分已画了
返回值
Long，调用本函数前的边界设置：
DCB_SET：边界矩形不为空
DCB_RESET：边界矩形为空
DCB_ENABLE：边界矩形已被堆放
DCB_DISABLE：边界矩形没被堆放
参数表
参数类型及说明
hdcLong，边界矩形对应的设备场景
lprcBoundsRECT，用于设置边界矩形的矩形
flagsLong，标志集合，可用下列常数组合而成
DCB_ACCUMULATE 新的边界矩形由当前边界矩形和 lprcBounds 联合而成
DCB_DISABLE 关闭边界堆放，该值为默认值
DCB_ENABLE 打开边界堆放
DCB_RESET 清除原有边界矩形，用 DCB_ACCUMULATE 设置新的边界矩形为 lprcBounds

Top
SetGraphicsMode
SetGraphicsMode

VB 声明
Declare Function SetGraphicsMode Lib "gdi32" Alias "SetGraphicsMode" (ByVal hdc As Long, ByVal iMode As Long) As Long
说明
允许或禁止增强图形模式，以提供某些支持（包括世界转换）
返回值
Long，执行成功为原来的图形模式，失败为 0
参数表
参数类型及说明
hdcLong，将为其设置模式的设备场景
iModeLong，下列值之一：
GM_COMPATIBLE：设置 win16 兼容模式
GM_ADVANCED：允许增强图形模式，包括世界转换
注解
只有 Windows NT 支持世界转换
若您已进入增强图形模式并已设置有世界转换，则您在将图形模式改回 GM_COMPATIBLE 之前，必须用 SetWorldTransform 或 ModifyWorldTransform 函数将世界转换改回默认的特性转换。关于特性转换更多的信息参考 ModifyWorldTransform

Top
SetMapMode

SetMapMode

VB 声明
Declare Function SetMapMode Lib "gdi32" Alias "SetMapMode" (ByVal hdc As Long, ByVal nMapMode As Long) As Long
说明
设置指定设备场景的映射模式
返回值
Long，执行成功为设备场景原来的映射模式，失败为 0
参数表
参数类型及说明
hdcLong，设备场景句柄
nMapModeLong，下列常数之一：
MM_ANISOTROPIC 视口和窗口范围可完全任意
MM_HIENGLISH 逻辑单元为 0.001 inch，起点在左下角
MM_HIMETRIC 逻辑单元为 0.01 millimeter，起点在左下角
MM_ISOTROPIC 视口和窗口范围任意，只是 x 和 y 逻辑单元尺寸要相同
MM_LOENGLISH 逻辑单元为 0.01 inch，起点在左下角
MM_HIMETRIC 逻辑单元为 0.1 millimeter，起点在左下角
MM_TEXT 逻辑单元为一个像素
MM_TWIPS 逻辑单元为 1 twip (1/1440 inch)，起点在左下角

Top
SetRectRgn
SetRectRgn

VB 声明
Declare Function SetRectRgn Lib "gdi32" Alias "SetRectRgn" (ByVal hRgn As Long, ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long) As Long
说明
设置区域为 X1，Y1 和 X2，Y2 描述的矩形
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hRgnLong，该区域将被设置为指定矩形
X1,Y1Long，矩形左上角 X，Y 坐标
X2,Y2Long，矩形右下角 X，Y 坐标
注解
本函数与 CreateRectRgn 相同，只是它是设置一个已存在区域而不是创建一个新区域
矩形的底和右边不包括在区域内

Top
SetViewportExtEx

SetViewportExtEx

VB 声明
Declare Function SetViewportExtEx Lib "gdi32" Alias "SetViewportExtEx" (ByVal hdc As Long, ByVal nX As Long, ByVal nY As Long, lpSize As SIZE) As Long
说明
设置设备场景视口范围
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hdcLong，设备场景句柄
nX,nYLong，视口水平和垂直范围
lpSizeSIZE，装载 DC 视口原来的水平和垂直范围（以设备单元表示）的结构
注解
本函数只在 MM_ISOTROPIC 和 MM_ANISOTROPIC 映射模式下有效。在 MM_ISOTROPIC 模式下，设置视口范围必须在窗口范围之前

Top
SetViewportOrgEx
SetViewportOrgEx

VB 声明
Declare Function SetViewportOrgEx Lib "gdi32" Alias "SetViewportOrgEx" (ByVal hdc As Long, ByVal nX As Long, ByVal nY As Long, lpPoint As POINTAPI) As Long
说明
设置设备场景视口起点
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hdcLong，设备场景句柄
nX,nYLong，以设备坐标表示的视口起点
lpPointPOINTAPI，装载 DC 视口原来的起点的结构

Top
SetWindowExtEx
SetWindowExtEx

VB 声明
Declare Function SetWindowExtEx Lib "gdi32" Alias "SetWindowExtEx" (ByVal hdc As Long, ByVal nX As Long, ByVal nY As Long, lpSize As SIZE) As Long
说明
设置指定设备场景窗口范围

返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hdcLong，设备场景句柄
nX,nYLong，窗口水平和垂直范围
lpSizeSIZE，装载设备场景原来的水平和垂直窗口范围（以逻辑单元表示）的结构
注解
本函数只在 MM_ISOTROPIC 和 MM_ANISOTROPIC 映射模式下有效。在 MM_ISOTROPIC 模式下，设置视口范围必须在窗口范围之前

Top
SetWindowOrgEx
SetWindowOrgEx

VB 声明
Declare Function SetWindowOrgEx Lib "gdi32" Alias "SetWindowOrgEx" (ByVal hdc As Long, ByVal nX As Long, ByVal nY As Long, lpPoint As POINTAPI) As Long
说明
设置指定设备场景窗口起点
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hdcLong，设备场景句柄
nX,nYLong，以逻辑坐标表示的窗口起点
lpPointPOINTAPI，装载原来窗口起点的结构

Top
SetWindowRgn
SetWindowRgn

VB 声明
Declare Function SetWindowRgn Lib "user32" Alias "SetWindowRgn" (ByVal hWnd As Long, ByVal hRgn As Long, ByVal bRedraw As Boolean) As Long
说明
这是那些很难有人注意到的对编程者来说是个巨大的宝藏的隐含的 API 函数中的一个。本函数允许您改变窗口的区域。
通常所有窗口都是矩形的——窗口一旦存在就含有一个矩形区域。本函数允许您放弃该区域。这意味着您可以创建圆的、星形的窗口，也可以将它分为两个或许多部分——实际上可以是任何形状
返回值
Long，执行成功为非零值，失败为 0
参数表

参数类型及说明
hWndLong，将设置其区域的窗口
hRgnLong，将设置的区域的句柄，一旦设置了该区域，就不能使用或修改该区域句柄，也不要删除它
bRedrawBoolean，若为 TRUE，则立即重画窗口
注解
为区域指定的所有坐标都以窗口坐标（不是客户坐标）表示，它们以整个窗口（包括标题栏和边框）的左上角为起点

Top
SetWorldTransform
SetWorldTransform

VB 声明
Declare Function SetWorldTransform Lib "gdi32" Alias "SetWorldTransform" (ByVal hdc As Long, lpXform As xform) As Long
说明
设置世界转换
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hdcLong，设备场景
lpXformxform，一个包含将用世界转换的结构
注解
适用于 Windows NT
在使用世界转换前必须调用 SetGraphicsMode 函数设置图形模式为 GM_ADVANCED 模式

Top
ValidateRgn
ValidateRgn

VB 声明
Declare Function ValidateRgn Lib "user32" Alias "ValidateRgn" (ByVal hwnd As Long, ByVal hRgn As Long) As Long
说明
激活窗口中指定区域，把它从刷新区移走
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hwndLong，窗口句柄
hRgnLong，定义要激活区的区域句柄，该区域以窗口客户坐标表示

Top
WindowFromDC
WindowFromDC

VB 声明
Declare Function WindowFromDC Lib "user32" Alias "WindowFromDC" (ByVal hdc As Long) As Long
说明
取回与某一设备场景相关的窗口的句柄
返回值
Long，执行成功为设备场景对应的窗口的句柄，失败为 0
参数表
参数类型及说明
hdcLong，设备场景句柄

Top
进程和线程函数
进程与线程函数，共五页。第一页，第二页，第三页，第四页，第五页
CancelWaitableTimer 这个函数用于取消一个可以等待下去的计时器操作
CallNamedPipe 这个函数由一个希望通过管道通信的一个客户进程调用
ConnectNamedPipe 指示一台服务器等待下去，直至客户机同一个命名管道连接
CreateEvent 创建一个事件对象
CreateMailslot 创建一个邮路。返回的句柄由邮路服务器使用（收件人）
CreateMutex 创建一个互斥体（MUTEX）
CreateNamedPipe 创建一个命名管道。返回的句柄由管道的服务器端使用
CreatePipe 创建一个匿名管道
CreateProcess 创建一个新进程（比如执行一个程序）
CreateSemaphore 创建一个新的信号机
CreateWaitableTimer 创建一个可等待的计时器对象
DisconnectNamedPipe 断开一个客户与一个命名管道的连接
DuplicateHandle 在指出一个现有系统对象当前句柄的情况下，为那个对象创建一个新句柄
ExitProcess 中止一个进程
FindCloseChangeNotification 关闭一个改动通知对象
FindExecutable 查找与一个指定文件关联在一起的程序的文件名
CallNamedPipe
CallNamedPipe

VB 声明
Declare Function CallNamedPipe Lib "kernel32" Alias "CallNamedPipeA" (ByVal lpNamedPipeName As String, lpInBuffer As Any, ByVal nInBufferSize As Long, lpOutBuffer As Any, ByVal nOutBufferSize As Long, lpBytesRead As Long, ByVal nTimeOut As Long) As Long
说明
这个函数由一个希望通过管道通信的一个客户进程调用。如有可能，它就同一个管道连接（在必要的情况下等候管道可用）。随后，它对指定的数据进行读写，然后将管道关闭

返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpNamedPipeNameString，欲打开管道的名称
lpInBufferAny，包含了要写入管道的数据的一个内存缓冲区
nInBufferSizeLong，lpInBuffer 缓冲区中的字符数量
lpOutBufferAny，指定一个内存缓冲区，用于装载从管道中读出的数据
nOutBufferSizeLong，指定一个长整数变量，用于装载来自管道的数据
lpBytesReadLong，指定从管道中读出的字节数。会阅读单条消息。如 lpOutBuffer 的容量不够大，不能容下整条消息，则函数会返回 FALSE，而且 GetLastError 会设为 ERROR_MORE_DATA（消息中留下的任何字节都会丢失）
nTimeOutLong，下述常数之一：
NMPWAIT_NOWAIT 如管道不可用，则立即返回一个错误
NMPWAIT_WAIT_FOREVER 永远等候管道可用
NMPWAIT_USE_DEFAULT_WAIT 使用管道的默认超时设置，这个设置是用 CreateNamedPipe 函数指定的

Top
CancelWaitableTimer
CancelWaitableTimer

VB 声明
Declare Function CancelWaitableTimer Lib "kernel32" (ByVal hTimer As Long)
说明
这个函数用于取消一个可以等待下去的计时器操作。计时器保持它当前的状态，而且除非用 SetWaitableTimer 函数明确启动，否则它不会重新启动
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hTimerLong，可等待计时器的句柄
适用平台
Windows NT

Top
ConnectNamedPipe
ConnectNamedPipe

VB 声明
Declare Function ConnectNamedPipe Lib "kernel32" Alias "ConnectNamedPipe" (ByVal hNamedPipe As Long, lpOverlapped As OVERLAPPED) As Long
说明
指示一台服务器等待下去，直至客户机同一个命名管道连接

返回值
Long，如 lpOverlapped 为 NULL，那么：
□ 如管道已连接，就返回 Ture（非零）；如发生错误，或者管道已经连接，就返回零（GetLastError 此时会返回 ERROR_PIPE_CONNECTED）
□ lpOverlapped 有效，就返回零；如管道已经连接，GetLastError 会返回 ERROR_PIPE_CONNECTED；如重叠操作成功完成，就返回 ERROR_IO_PENDING。在这两种情况下，倘若一个客户已关闭了管道，且服务器尚未用 DisconnectNamedPipe 函数同客户断开连接，那么 GetLastError 都会返回 ERROR_NO_DATA
参数表
参数类型及说明
hNamedPipeLong，管道的句柄
lpOverlappedOVERLAPPED，如设为 NULL（传递 ByVal As Long），表示将线程挂起，直到一个客户同管道连接为止。否则就立即返回；此时，如管道尚未连接，客户同管道连接时就会触发 lpOverlapped 结构中的事件对象。随后，可用一个等待函数来监视连接
适用平台
Windows NT
注解
可用这个函数将一个管道换成同另一个客户连接，但首先必须用 DisconnectNamedPipe 函数断开同当前进程的连接

Top
CreateEvent
CreateEvent

VB 声明
Declare Function CreateEvent Lib "kernel32" Alias "CreateEventA" (lpEventAttributes As SECURITY_ATTRIBUTES, ByVal bManualReset As Long, ByVal bInitialState As Long, ByVal lpName As String) As Long
说明
创建一个事件对象
返回值
Long，如执行成功，返回事件对象句柄；零表示出错。会设置 GetLastError。即使返回一个有效的句柄，但同时指出指定的名字已经存在，GetLastError 也会设为 ERROR_ALREADY_EXISTS
参数表
参数类型及说明
lpEventAttributesSECURITY_ATTRIBUTES，指定一个结构，用于设置对象的安全特性。如变成 ByVal As Long，并传递零值，则表明使用对象默认的安全设置
bManualResetLong，如果为 TRUE，表示创建一个人工重设事件；如果为 FALSE，表示创建一个自动重设事件
bInitialStateLong，如事件应内部进入触发状态，则为 TRUE
lpNameString，指定事件对象的名字。用 vbNullString 创建一个未命名事件对象。如已经存在拥有这个名字的一个事件，则现有的命名事件就会打开。这个名字可能不与一个现有互斥体、信号机、可等待计时器或文件映射的名字相符

注解
一旦不再需要，注意一定要用 CloseHandle 关闭事件句柄。如对象的所有句柄都已关闭，对象也会自动删除

Top
CreateMailslot
CreateMailslot

VB 声明
Declare Function CreateMailslot Lib "kernel32" Alias "CreateMailslotA" (ByVal lpName As String, ByVal nMaxMessageSize As Long, ByVal lReadTimeout As Long, lpSecurityAttributes As SECURITY_ATTRIBUTES) As Long
说明
创建一个邮路。返回的句柄由邮路服务器使用（收件人）
返回值
Long，如执行成功，返回邮路的句柄；INVALID_HANDLE_VALUE 表示失败。会设置 GetLastError
参数表
参数类型及说明
lpNameString，指定邮路的名字，采用的形式如下：\\.\邮路\[路径\]邮路名
nMaxMessageSizeLong，指定一个邮路消息的最大长度。零表示无限长。请注意，对于穿越一个网络域到多个邮路的广播消息，最大长度是 400
lReadTimeoutLong，等待指定的数据时，用这个参数指定邮路使用的默认超时设置，以毫秒为单位。零表示不等待。常数 MAILSLOT_WAIT_FOREVER 表示一直等到数据到达
lpSecurityAttributesSECURITY_ATTRIBUTES，指定一个结构，或传递零值（将参数声明为 ByVal As Long，并传递零值），表示使用不允许继承的默认描述符

Top
CreateMutex
CreateMutex

VB 声明
Declare Function CreateMutex Lib "kernel32" Alias "CreateMutexA" (lpMutexAttributes As SECURITY_ATTRIBUTES, ByVal bInitialOwner As Long, ByVal lpName As String) As Long
说明
创建一个互斥体（MUTEX）
返回值
Long，如执行成功，就返回互斥体对象的句柄；零表示出错。会设置 GetLastError。即使返回的是一个有效句柄，但倘若指定的名字已经存在，GetLastError 也会设为 ERROR_ALREADY_EXISTS
参数表
参数类型及说明
lpMutexAttributesSECURITY_ATTRIBUTES，指定一个 SECURITY_ATTRIBUTES 结构，或传递零值（将参数声明为 ByVal As Long，并传递零值），表示使用不允许继承的默认描述符

bInitialOwnerLong，如创建进程希望立即拥有互斥体，则设为 TRUE。一个互斥体同时只能由一个线程拥有
lpNameString，指定互斥体对象的名字。用 vbNullString 创建一个未命名的互斥体对象。如已经存在拥有这个名字的一个事件，则打开现有的已命名互斥体。这个名字可能不与现有的事件、信号机、可等待计时器或文件映射相符
注解
一旦不再需要，注意必须用 CloseHandle 函数将互斥体句柄关闭。从属于它的所有句柄都被关闭后，就会删除对象
进程中止前，一定要释放互斥体，如不慎未采取这个措施，就会将这个互斥体标记为废弃，并自动释放所有权。共享这个互斥体的其他应用程序也许仍然能够用它，但会接收到一个废弃状态信息，指出上一个所有进程未能正常关闭。这种状况是否会造成影响取决于涉及到的具体应用程序

Top
CreateNamedPipe
CreateNamedPipe

VB 声明
Declare Function CreateNamedPipe Lib "kernel32" Alias "CreateNamedPipeA" (ByVal lpName As String, ByVal dwOpenMode As Long, ByVal dwPipeMode As Long, ByVal nMaxInstances As Long, ByVal nOutBufferSize As Long, ByVal nInBufferSize As Long, ByVal nDefaultTimeOut As Long, lpSecurityAttributes As SECURITY_ATTRIBUTES) As Long
说明
创建一个命名管道。返回的句柄由管道的服务器端使用
返回值
Long，如执行成功，返回管道的句柄。INVALID_HANDLE_VALUE 表示失败。会设置 GetLastError
参数表
参数类型及说明
lpNameString，指定管道名，采用的形式是：\\.\管道\管道名。最多可达 256 个字符的长度，而且不用区分大小写。如果存在指定名字的一个管道，则创建那个管道的一个新实例
dwOpenModeLong，下述常数组的一个组合
下述常数之一（对于管道的所有实例都要一样）：
PIPE_ACCESS_DUPLEX 管道是双向的
PIPE_ACCESS_INBOUND 数据从客户端流到服务器端
PIPE_ACCESS_OUTBOUND 数据从服务器端流到客户端
下述常数的任意组合
FILE_FLAG_WRITE_THROUGH 在网络中建立的字节型管道内，强迫数据在每次读写操作的时候通过网络传输。否则传输就可能延迟
FILE_FLAG_OVERLAPPED 允许（但不要求）用这个管道进行异步（重叠式）操作
常数 WRITE_DAC，WRITE_OWNER 和 ACCESS_ SYSTEM_SECURITY 提供了附加的安全选项
dwPipeModeLong，下述常数组的一个组合：
下述常数之一（管道的所有实例都必须指定相同的常数）

PIPE_TYPE_BYTE 数据作为一个连续的字节数据流写入管道
PIPE_TYPE_MESSAGE 数据用数据块（名为“消息”或“报文”）的形式写入管道
下述常数之一：
PIPE_READMODE_PIPE 数据以单独字节的形式从管道中读出
PIPE_READMODE_MESSAGE 数据以名为“消息”的数据块形式从管道中读出（要求指定 PIPE_TYPE_MESSAGE）
下述常数之一：
PIPE_WAIT 同步操作在等待的时候挂起线程
PIPE_NOWAIT（不推荐！）同步操作立即返回。这样可为异步传输提供一种落后的实现方法，已由 Win32 的重叠式传输机制取代了
nMaxInstancesLong，这个管道能够创建的最大实例数量。必须是 1 到常数 PIPE_UNLIMITED_INSTANCES 间的一个值。它对于管道的所有实例来说都应是相同的
nOutBufferSizeLong，建议的输出缓冲区长度；零表示用默认设置
nInBufferSizeLong，建议的输入缓冲区长度；零表示用默认设置
nDefaultTimeOutLong，管道的默认等待超时。对一个管道的所有实例来说都应相同
lpSecurityAttributesSECURITY_ATTRIBUTES，指定一个 SECURITY_ATTRIBUTES 结构，或者传递零值（将参数声明为 ByVal As Long，并传递零值），以便使用不允许继承的一个默认描述符
适用平台
Windows NT

Top
CreatePipe
CreatePipe

VB 声明
Declare Function CreatePipe Lib "kernel32" Alias "CreatePipe" (phReadPipe As Long, phWritePipe As Long, lpPipeAttributes As SECURITY_ATTRIBUTES, ByVal nSize As Long) As Long
说明
创建一个匿名管道
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
phReadPipeLong，指定一个变量，设为管道读入（输出）端的一个句柄
phWritePipeLong，指定一个变量，设为管道写入（输入）端的一个句柄
lpPipeAttributesSECURITY_ATTRIBUTES，指定一个 SECURITY_ATTRIBUTES 结构，或者传递零值（将参数声明为 ByVal As Long，并传递零值），以便使用不允许继承的一个默认描述符
nSizeLong，管道缓冲区的建议大小。零表示用默认值
注解
匿名管道不允许异步操作，所以如在一个管道中写入数据，且缓冲区已满，那么除非另一个进程从管道中读出数据，从而腾出了缓冲区的空间，否则写入函数不会返回

Top
CreateProcess
CreateProcess

VB 声明
Declare Function CreateProcess Lib "kernel32" Alias "CreateProcessA" (ByVal lpApplicationName As String, ByVal lpCommandLine As String, lpProcessAttributes As SECURITY_ATTRIBUTES, lpThreadAttributes As SECURITY_ATTRIBUTES, ByVal bInheritHandles As Long, ByVal dwCreationFlags As Long, lpEnvironment As Any, ByVal lpCurrentDriectory As String, lpStartupInfo As STARTUPINFO, lpProcessInformation As PROCESS_INFORMATION) As Long
说明
创建一个新进程（比如执行一个程序）
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpApplicationNameString，要执行的应用程序的名字。可设为 vbNullString；在这种情况下，应用程序的名字应在 lpCommandLine 参数的起始处出现
lpCommandLineString，要执行的命令行。可用 GetCommandLine 函数取得一个进程使用的命令行。Windows 会尽可能地根据下述搜索顺序来查找执行文件：
（1）包含了父进程执行文件的目录
（2）父进程当前的目录
（3）由 GetSystemDirectory 返回的系统目录
（4）仅适于 windows NT：16 位系统目录
（5）由 GetWindowDirectory 返回的 Windows 目录
（6）由 PATH 环境变量指定的目录
lpProcessAttributesSECURITY_ATTRIBUTES，指定一个 SECURITY_ATTRIBUTES 结构，或传递零值（将参数声明为 ByVal As Long，并传递零值）——表示采用不允许继承的默认描述符。该参数定义了进程的安全特性
lpThreadAttributesSECURITY_ATTRIBUTES，指定一个 SECURITY_ATTRIBUTES 结构，或传递零值（将参数声明为 ByVal As Long，并传递零值）——表示采用不允许继承的默认描述符。该参数定义了进程之主线程的安全特性
bInheritHandlesLong，TRUE 表示允许当前进程中的所有句柄都由新建的子进程继承
dwCreationFlagsLong，来自 API32.TXT 文件的一个或多个下述常数之一，它们都带有前缀 CREATE_。下面这些用于 VB 程序员：
CREATE_SEPARATE_WOW_VDM（仅适用于 NT）启动一个 16 位的 Windows 应用程序时，强迫它在自己的内存空间运行
CREATE_SHARED_WOW_VDM（仅适用于 NT）启动一个 16 位的 Windows 应用程序时，强迫它在共享的 16 位虚拟机（VM）内运行
CREATE_SUSPENDED 立即挂起新进程。除非调用了 ResumeThread 函数函数，否则它不会恢复运行
也可能是下述常数之一，用于指定优先级

IDLE_PRIORITY_CLASS 新进程应该有非常低的优先级——只有在系统空闲的时候才能运行。基本值是 4
HIGH_PRIORITY_CLASS 新进程有非常高的优先级，它优先于大多数应用程序。基本值是 13。注意尽量避免采用这个优先级
NORMAL_PRIORITY_CLASS 标准优先级。如进程位于前台，则基本值是 9；如在后台，则优先值是 7
不要在 VB 中使用 REALTIME_PRIORITY_CLASS
lpEnvironmentAny，指向一个环境块的指针（环境缓冲区的头一个字符，或者环境块的地址）
lpCurrentDriectoryString，新进程的当前目录路径。调用函数的时候，可用 vbNullString 指定当前目录
lpStartupInfoSTARTUPINFO，指定一个 STARTUPINFO 结构，其中包含了创建进程时使用的附加信息
lpProcessInformationPROCESS_INFORMATION，该结构用于容纳新进程的进程和线程标识符。大多数情况下，一旦这个函数返回，父应用程序都会关闭两个句柄。

Top
CreateSemaphore
CreateSemaphore

VB 声明
Declare Function CreateSemaphore Lib "kernel32" Alias "CreateSemaphoreA" (lpSemaphoreAttributes As SECURITY_ATTRIBUTES, ByVal lInitialCount As Long, ByVal lMaximumCount As Long, ByVal lpName As String) As Long
说明
创建一个新的信号机
返回值
Long，如执行成功，返回信号机对象的句柄；零表示出错。会设置 GetLastError。即使返回一个有效的句柄，但倘若它指出同名的一个信号机已经存在，那么 GetLastError 也会返回 ERROR_ALREADY_EXISTS
参数表
参数类型及说明
lpSemaphoreAttributesSECURITY_ATTRIBUTES，指定一个 SECURITY_ATTRIBUTES 结构，或传递零值（将参数声明为 ByVal As Long，并传递零值）——表示采用不允许继承的默认描述符。该参数定义了信号机的安全特性
lInitialCountLong，设置信号机的初始计数。可设置零到 lMaximumCount 之间的一个值
lMaximumCountLong，设置信号机的最大计数
lpNameString，指定信号机对象的名称。用 vbNullString 可创建一个未命名的信号机对象。如果已经存在拥有这个名字的一个信号机，就直接打开现成的信号机。这个名字可能不与一个现有的互斥体、事件、可等待计时器或文件映射的名称相符
注解
一旦不再需要，一定记住用 CloseHandle 关闭信号机的句柄。它的所有句柄都关闭以后，对象自己也会删除
一旦值大于零，信号机就会触发（发出信号）。ReleaseSemaphore 函数的作用是增加信号机的计数。如果成功，就调用信号机上的一个等待函数来减少它的计数

Top
CreateWaitableTimer
CreateWaitableTimer

VB 声明
Declare Function CreateWaitableTimer Lib "kernel32" Alias "CreateWaitableTimerA" (lpSemaphoreAttributes As SECURITY_ATTRIBUTES, ByVal bManualReset As Long, ByVal lpName As String) As Long
说明
创建一个可等待的计时器对象
返回值
Long，如执行成功，返回可等待计时器对象的句柄；零表示出错。会设置 GetLastError。即使返回一个有效的句柄，但倘若它指出同名的一个计时器对象已经存在，那么 GetLastError 也会返回 ERROR_ALREADY_EXISTS
参数表
参数类型及说明
lpSemaphoreAttributesSECURITY_ATTRIBUTES，指定一个结构，用于设置对象的安全特性。如将参数声明为 ByVal As Long，并传递零值，就可使用对象的默认安全设置
bManualResetLong，如果为 TRUE，表示创建一个人工重设计时器；如果为 FALSE，则创建一个自动重设计时器
lpNameString，指定可等待计时器对象的名称。用 vbNullString 可创建一个未命名的计时器对象。如果已经存在拥有这个名字的一个可等待计时器，就直接打开现成的可等待计时器。这个名字可能不与一个现有的互斥体、事件、信号机或文件映射的名称相符
注解
一旦不再需要，一定记住用 CloseHandle 关闭计时器对象的句柄。它的所有句柄都关闭以后，对象自己也会删除

Top
DisconnectNamedPipe
DisconnectNamedPipe

VB 声明
Declare Function DisconnectNamedPipe Lib "kernel32" Alias "DisconnectNamedPipe" (ByVal hNamedPipe As Long) As Long
说明
断开一个客户与一个命名管道的连接
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hNamedPipeLong，管道的句柄
适用平台
Windows NT

注解
如客户尚未在它自己那端关闭管道句柄，下次试图访问管道的时候就会发生错误

Top
DuplicateHandle
DuplicateHandle

VB 声明
Declare Function DuplicateHandle Lib "kernel32" Alias "DuplicateHandle" (ByVal hSourceProcessHandle As Long, ByVal hSourceHandle As Long, ByVal hTargetProcessHandle As Long, lpTargetHandle As Long, ByVal dwDesiredAccess As Long, ByVal bInheritHandle As Long, ByVal dwOptions As Long) As Long
说明
在指出一个现有系统对象当前句柄的情况下，为那个对象创建一个新句柄。当前句柄可能位于一个不同的进程
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hSourceProcessHandleLong，拥有源句柄的那个进程的句柄。如源句柄从属于当前进程，则使用 GetCurrentProcess
hSourceHandleLong，指定对象的现有句柄。
hTargetProcessHandleLong，即将拥有新对象句柄的一个进程的句柄。如源句柄从属于当前进程，则使用 GetCurrentProcess
lpTargetHandleLong，指定用于装载新句柄的一个长整型变量
dwDesiredAccessLong，新句柄要求的安全访问级别。如 dwOptions 已指定了 DUPLICATE_SAME_ACCESS，那么忽略这里的设置。可以进行的访问由对象的类型决定，它们在不同系统对象的访问常数表里进行了总结
bInheritHandleLong，如新句柄可由 hSourceProcessHandle 的子进程继承，则为 TRUE
dwOptionsLong，下列常数的一个或两个：
DUPLICATE_SAME_ACCESS 新句柄拥有与原始句柄相同的安全访问特征
DUPLICATE_CLOSE_SOURCE 原始句柄已经关闭。即使发生错误。它也要关闭
注解
在一个进程中，这个函数可根据位于不同进程内的现有句柄创建一个新句柄。可以从这两个进程中发出对这个函数的调用。进程必须提供 PROCESS_DUP_HANDLE 访问权限，否则函数执行不能成功
句柄可以重复的对象包括控制台、文件（包括通信设备）、文件映射、事件、可等待计时器、互斥体、管道、进程、注册表项、信号机以及线程

Top
ExitProcess
ExitProcess

VB 声明

Declare Sub ExitProcess Lib "kernel32" Alias "ExitProcess" (ByVal uExitCode As Long)
说明
中止一个进程
参数表
参数类型及说明
uExitCodeLong，指定想中断的那个进程的一个退出代码
在 VB 中使用
应尽量避免用该函数来关闭进程。不要在自己的 VB 程序中使用它。此时，应试着向要关闭的那个程序的主窗口投递一条 WM_CLOSE 消息

Top
FindCloseChangeNotification
FindCloseChangeNotification

VB 声明
Declare Function FindCloseChangeNotification Lib "kernel32" Alias "FindCloseChangeNotification" (ByVal hChangeHandle As Long) As Long
说明
关闭一个改动通知对象
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hChangeHandleLong，要关闭的那个文件改动通知对象的句柄

Top
FindExecutable
FindExecutable

VB 声明
Declare Function FindExecutable Lib "shell32.dll" Alias "FindExecutableA" (ByVal lpFile As String, ByVal lpDirectory As String, ByVal lpResult As String) As Long
说明
查找与一个指定文件关联在一起的程序的文件名。可用 Windows 注册表编辑器将文件类型与特定的应用程序关联到一起。比如，扩展名为.TXT 的文本文件通常与 Windows 记事本（Notepad.exe）关联到一起。如在文件管理器中双击含.TXT 扩展名的一个文件，会自动启动记事本，并在其中载入文本文件
返回值
Long，大于 32 表示成功；31 表示不存在文件类型的关联；0 表示系统内存或资源不足；ERROR_FILE_NOT_FOUND 表示指定的文件不存在；ERROR_PATH_NOT_FOUND 表示指定的路径不存在；ERROR_BAD_FORMAT 表示执行格式无效
参数表
参数类型及说明
lpFileString，指定要为其查找相关程序的一个文件名或程序名

lpDirectoryString，要使用的默认目录的完整路径
lpResultString，指定一个字串缓冲区，用于装载可执行程序的名字。注意这个字串预先至少都要初始化成 MAX_PATH 个字符的长度

Top
FindFirstChangeNotification
FindFirstChangeNotification

VB 声明
Declare Function FindFirstChangeNotification Lib "kernel32" Alias "FindFirstChangeNotificationA" (ByVal lpPathName As String, ByVal bWatchSubtree As Long, ByVal dwNotifyFilter As Long) As Long
说明
创建一个文件通知对象。该对象用于监视文件系统发生的变化
返回值
Long，如成功，返回一个改变通知对象的句柄；INVALID_HANDLE_VALUE 表示失败。会设置 GetLastError
参数表
参数类型及说明
lpPathNameString，要监视的目录
bWatchSubtreeLong，如果为 TRUE，表示监视 lpPathName 的所有子目录
dwNotifyFilterLong，带有前缀 FILE_NOTIFY_CHANGE_???前缀的一个或多个常数，它们指定了对象发出信号的条件
注解
用 FindCloseChangeNotification 函数关闭句柄，不要用 CloseHandle 函数

Top
FindNextChangeNotification
FindNextChangeNotification

VB 声明
Declare Function FindNextChangeNotification Lib "kernel32" Alias "FindNextChangeNotification" (ByVal hChangeHandle As Long) As Long
说明
重设一个文件改变通知对象，令其继续监视下一次变化
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hChangeHandleLong，要重设的那个文件改变通知对象的句柄

Top
FreeLibrary
FreeLibrary

VB 声明
Declare Function FreeLibrary Lib "kernel32" Alias "FreeLibrary" (ByVal hLibModule As Long) As Long
说明
释放指定的动态链接库，它们早先是用 LoadLibrary API 函数装载的
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hLibModuleLong，要释放的一个库句柄
在 VB 里使用
只能用这个函数释放那些由应用程序明确装载的 DLL。对 LoadLibrary 的每一次调用都应该有一个对应的 FreeLibrary 调用

Top
GetCurrentProcess
GetCurrentProcess

VB 声明
Declare Function GetCurrentProcess Lib "kernel32" Alias "GetCurrentProcess" () As Long
说明
获取当前进程的一个伪句柄
返回值
Long，当前进程的伪句柄
注解
只要当前进程需要一个进程句柄，就可以使用这个伪句柄。该句柄可以复制，但不可继承。不必调用 CloseHandle 函数来关闭这个句柄

Top
GetCurrentProcessId
GetCurrentProcessId

VB 声明
Declare Function GetCurrentProcessId Lib "kernel32" Alias "GetCurrentProcessId" () As Long
说明
获取当前进程一个唯一的标识符
返回值
Long，当前的进程标识符

Top
GetCurrentThread
GetCurrentThread

VB 声明
Declare Function GetCurrentThread Lib "kernel32" Alias "GetCurrentThread" () As Long
说明
获取当前线程的一个伪句柄
返回值
Long，当前线程的伪句柄
注解
只要当前线程需要使用一个线程句柄，就可以使用这个伪句柄（但在其他任务线程中都无效）。该句柄可以复制，但不可继承。不必调用 CloseHandle 函数来关闭这个句柄

Top
GetCurrentThreadId
GetCurrentThreadId

VB 声明
Declare Function GetCurrentThreadId Lib "kernel32" Alias "GetCurrentThreadId" () As Long
说明
获取当前线程一个唯一的线程标识符
返回值
Long，当前的线程标识符

Top
GetExitCodeProcess
GetExitCodeProcess

VB 声明
Declare Function GetExitCodeProcess Lib "kernel32" Alias "GetExitCodeProcess" (ByVal hProcess As Long, lpExitCode As Long) As Long
说明
获取一个已中断进程的退出代码
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hProcess Long，想获取退出代码的一个进程的句柄
lpExitCodeLong，用于装载进程退出代码的一个长整数变量。如进程尚未中止，则设为常数 STILL_ACTIVE

Top
GetExitCodeThread
GetExitCodeThread

VB 声明
Declare Function GetExitCodeThread Lib "kernel32" Alias "GetExitCodeThread" (ByVal hThread

As Long, lpExitCode As Long) As Long
说明
获取一个已中止线程的退出代码
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hThreadLong，想获取退出代码的一个线程的句柄
lpExitCodeLong，用于装载线程退出代码的一个长整数变量。如线程尚未中断，则设为常数 STILL_ACTIVE

Top
GetHandleInformation
GetHandleInformation

VB 声明
Declare Function GetHandleInformation Lib "kernel32" Alias "GetHandleInformation" (ByVal hObject As Long, lpdwFlags As Long) As Long
说明
获取与一个系统对象句柄有关的信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hObjectLong，要测试的句柄
lpdwFlagsLong，设置了下述一个或多个常数的位标志：
HANDLE_FLAG_INHERIT 对象可由一个子进程继承
HANDLE_FLAG_PROTECT_FROM_CLOSE 句柄不可用 CloseHandle 函数关闭
适用平台
Windows NT

Top
GetMailslotInfo
GetMailslotInfo

VB 声明
Declare Function GetMailslotInfo Lib "kernel32" Alias "GetMailslotInfo" (ByVal hMailslot As Long, lpMaxMessageSize As Long, lpNextSize As Long, lpMessageCount As Long, lpReadTimeout As Long) As Long
说明
获取与一个邮路有关的信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表

参数类型及说明
hMailslotLong，指定一个邮路的句柄
lpMaxMessageSizeLong，指定一个长整数变量，用于装载这个邮路的最大消息长度
lpNextSizeLong，指定一个长整数变量，用于装载下一条消息的长度。如没有消息准备好，则可设为常数 MAILSLOT_NO_MESSAGE
lpMessageCountLong，指定一个长整数变量，用于装载邮路中准备好的消息数量
lpReadTimeoutLong，指定一个长整数变量，用于装载邮路的默认阅读超时

Top
GetModuleFileName
GetModuleFileName

VB 声明
Declare Function GetModuleFileName Lib "kernel32" Alias "GetModuleFileNameA" (ByVal hModule As Long, ByVal lpFileName As String, ByVal nSize As Long) As Long
说明
获取一个已装载模板的完整路径名称
返回值
Long，如执行成功，返回复制到 lpFileName 的实际字符数量；零表示失败。会设置 GetLastError
参数表
参数类型及说明
hModuleLong，一个模块的句柄。可以是一个 DLL 模块，或者是一个应用程序的实例句柄
lpFileNameString，指定一个字串缓冲区，要在其中容纳文件的用 NULL 字符中止的路径名，hModule 模块就是从这个文件装载进来的
nSizeLong，装载到缓冲区 lpFileName 的最大字符数量
注解
在 Windows 95 下，函数会核查应用程序的内部版本号是否为 4.0 或更大的一个数字。如果是，就返回一个长文件名，否则返回短文件名

Top
GetModuleHandle
GetModuleHandle

VB 声明
Declare Function GetModuleHandle Lib "kernel32" Alias "GetModuleHandleA" (ByVal lpModuleName As String) As Long
说明
获取一个应用程序或动态链接库的模块句柄
返回值
Long，如执行成功成功，则返回模块句柄。零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpModuleNameString，指定模块名，这通常是与模块的文件名相同的一个名字。例如，NOTEPAD.EXE 程序的模块文件名就叫作 NOTEPAD

注解
只有在当前进程的场景中，这个句柄才会有效

Top
GetPriorityClass
GetPriorityClass

VB 声明
Declare Function GetPriorityClass Lib "kernel32" Alias "GetPriorityClass" (ByVal hProcess As Long) As Long

说明
　获取特定进程的优先级别
返回值
Long，进程的优先级，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hProcessLong，进程句柄

Top
GetProcessShutdownParameters
GetProcessShutdownParameters

VB 声明
Declare Function GetProcessShutdownParameters Lib "kernel32" Alias "GetProcessShutdownParameters" (lpdwLevel As Long, lpdwFlags As Long) As Long

说明
　调查系统关闭时一个指定的进程相对于其它进程的关闭早迟情况
返回值
Long，非零表示成功，零表示失败。
参数表
参数类型及说明
lpdwLevelLong，从 0 到&H4FF 的一个值。编号越高程序在系统关闭时越早关闭。
lpdwFlagsLong，指定 SHUTDOWN_NORETRY 标志。如关闭一个进程时，会出现一个重试对话框。通过设置这个标志，就可以禁止那个对话框出现，并直接关闭进程
适用平台
　Wnidows NT

Top
GetProcessTimes
GetProcessTimes

VB 声明

Declare Function GetProcessTimes Lib "kernel32" Alias "GetProcessTimes" (ByVal hProcess As Long, lpCreationTime As FILETIME, lpExitTime As FILETIME, lpKernelTime As FILETIME, lpUserTime As FILETIME) As Long

说明
　获取与一个进程的经过时间有关的信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hProcessLong，一个进程句柄
lpCreationTimeFILETIME，指定一个 FILETIME 结构，在其中装载进程的创建时间
lpExitTimeFILETIME，指定一个 FILETIME 结构，在其中装载进程的中止时间
lpKernelTimeFILETIME，指定一个 FILETIME 结构，在其中装载进程花在内核模式上的总时间
lpUserTimeFILETIME，指定一个 FILETIME 结构，在其中装载进程花在用户模式上的总时间
适用平台
Windows NT

Top
GetProcessWorkingSetSize
GetProcessWorkingSetSize

VB 声明
Declare Function GetProcessWorkingSetSize Lib "kernel32" Alias "GetProcessWorkingSetSize" (ByVal hProcess As Long, lpMinimumWorkingSetSize As Long, lpMaximumWorkingSetSize As Long) As Long

说明
　了解一个应用程序在运行过程中实际向它交付了多大容量的内存
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hProcessLong，指定一个进程的句柄
lpMinimumWorkingSetSizeLong，用于装载最小进程容量的一个变量
lpMaximumWorkingSetSizeLong，用于装载最大进程容量的一个变量
适用平台
Windows NT

Top
GetSartupInfo

GetSartupInfo

VB 声明
Declare Sub GetStartupInfo Lib "kernel32" Alias "GetStartupInfoA" (lpStartupInfo As STARTUPINFO)

说明
获取一个进程的启动信息
参数表
参数类型及说明
lpStartupInfoSTARTUPINFO，指定一个 STARTUPINFO 结构，用于最终载入进程的启动信息

Top
GetTheardTimes
GetTheardTimes

VB 声明
Declare Function GetThreadTimes Lib "kernel32" Alias "GetThreadTimes" (ByVal hThread As Long, lpCreationTime As FILETIME, lpExitTime As FILETIME, lpKernelTime As FILETIME, lpUserTime As FILETIME) As Long
说明
　获取与一个线程的经过时间有关的信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hThreadLong，一个线程句柄
lpCreationTimeFILETIME，指定一个 FILETIME 结构，在其中装载线程的创建时间
lpExitTimeFILETIME，指定一个 FILETIME 结构，在其中装载线程的中止时间
lpKernelTimeFILETIME，指定一个 FILETIME 结构，在其中装载线程花在内核模式上的总时间
lpUserTimeFILETIME，指定一个 FILETIME 结构，在其中装载线程花在用户模式上的总时间
适用平台
Windows NT

Top
GetThreadPriority
GetThreadPriority

VB 声明
Declare Function GetThreadPriority Lib "kernel32" Alias "GetThreadPriority" (ByVal hThread As Long) As Long

说明
获取特定线程的优先级别
返回值
Long，返回带有 THREAD_PRIORITY_???前缀的某个函数，它规定了线程的优级。THREAD_PRIORITY_ERROR_RETURN 表示出错
参数表
参数类型及说明
hThreadLong，线程句柄
注解
线程的优先级同进程的优先级类组合在一起就决定了线程的实际优先级

Top
GetWindowThreadProcessId
GetWindowThreadProcessId

VB 声明
Declare Function GetWindowThreadProcessId Lib "user32" Alias "GetWindowThreadProcessId" (ByVal hwnd As Long, lpdwProcessId As Long) As Long

说明
获取与指定窗口关联在一起的一个进程和线程标识符
返回值
Long，拥有窗口的线程的标识符
参数表
参数类型及说明
lpdwProcessIdLong，指定一个变量，用于装载拥有那个窗口的一个进程的标识符
hwndLong，指定窗口句柄

Top
LoadLibrary
LoadLibrary

VB 声明
Declare Function LoadLibrary Lib "kernel32" Alias "LoadLibraryA" (ByVal lpLibFileName As String) As Long
说明
载入指定的动态链接库，并将它映射到当前进程使用的地址空间。一旦载入，即可访问库内保存的资源
返回值
Long，成功则返回库模块的句柄，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpLibFileNameString，指定要载入的动态链接库的名称。采用与 CreateProcess 函数的 lpCommandLine 参数指定的同样的搜索顺序

注解
一旦不需要，用 FreeLibrary 函数释放 DLL

Top
LoadLibraryEx
LoadLibraryEx

VB 声明
Declare Function LoadLibraryEx Lib "kernel32" Alias "LoadLibraryExA" (ByVal lpLibFileName As String, ByVal hFile As Long, ByVal dwFlags As Long) As Long

说明
装载指定的动态链接库，并为当前进程把它映射到地址空间。一旦载入，就可以访问库内保存的资源
返回值
Long，成功则返回库模块的句柄，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpLibFileNameString，指定要载入的动态链接库的名称。采用与 CreateProcess 函数的 lpCommandLine 参数指定的同样的搜索顺序
hFileLong，未用，设为零
dwFlagsLong，指定下述常数的一个或多个
DONT_RESOLVE_DLL_REFERENCES：不对 DLL 进行初始化，仅用于 NT
LOAD_LIBRARY_AS_DATAFILE：不准备 DLL 执行。如装载一个 DLL 只是为了访问它的资源，就可以改善一部分性能
LOAD_WITH_ALTERED_SEARCH_PATH：指定搜索的路径
注解
一旦不需要，用 FreeLibrary 函数释放 DLL

Top
LoadModule
LoadModule

VB 声明
Declare Function LoadModule Lib "kernel32" Alias "LoadModule" (ByVal lpModuleName As String, lpParameterBlock As Any) As Long
说明
　载入一个 windows 应用程序，并在指定的环境中运行
返回值
Long，大于 32 表示成功，请参考 FindExecutable 函数的返回值
参数表
参数类型及说明
lpModuleNameSring，要装载的可执行程序的文件名
lpParameterBlockAny，指定一个结构，用它定义装载新应用程序时使用的参数

Top
MsgWaitForMultipleObjects
MsgWaitForMultipleObjects

VB 声明
Declare Function MsgWaitForMultipleObjects Lib "user32" Alias "MsgWaitForMultipleObjects" (ByVal nCount As Long, pHandles As Long, ByVal fWaitAll As Long, ByVal dwMilliseconds As Long, ByVal dwWakeMask As Long) As Long
说明
等侯单个对象或一系列对象发出信号---标志着规定的超时已经过去，或特定类型的消息已抵达线程的输入队列。如返回条件已经满足，则立即返回
返回值
Long，如 fWaitAll 设为 TRUE，则下述任何一个常数都标志着成功：
WAIT_ABANDONED_0：所有对象都发出消息，而且其中一个或多个属于互斥体（一但拥有它门的进程中止，就会发出信号）。
WAIT_TIMEOUT：对象保持未发信号的状态，但规定的等待超时时间已经超过
WAIT_OBJECT_0：所有的对象都发出信号
WAIT_TO_COMPLETION（仅适用于 WaitForSingleObjectEx），由于一个 I/O 完成操作已准备好执行，从而造成了函数的返回。
返回 WAIT_FALIED 表示函数执行失败。会设置 GetLastError
如 fWaitAll 设为 FALSE，那返回结果与前面说的相似，只是可能还会返回相对于 WAIT_ABANDONED_0 或 WAIT_OBJECT_0 的一个正偏移量，指出哪个对象是被抛弃还是发出信号。
如果是由于 dwWakeMask 指定的，符合特殊标准的一条消息的到达而造成了函数的返回，则返回 WAIT_OBJECT_0 + nCount
参数表
参数类型及说明
nCountLong，指定列表中的句柄数量
pHandlesLong，指定对象句柄组合中的第一个元素
fWaitAllLong，如果为 TRUE，表示除非对象同时发出信号，否则就等待下去。如果为 FALSE，表示任何对象发出信号即可。
dwMillisecondsLong，指定要等待的毫秒数。
dwWakeMaskLong，带有 QS_？？前缀的一个或多个常数，用于标识特定的消息类型。
注解
如果函数是由于对象发出信号而返回，这个函数还会得到一些额外的效果：
信号机：递增信号机计数
互斥体：将互斥体的所有权限赋于发出调用的线程
自动重设事件：将事件发信状态设为 FALSE
自动重设可等待计时器：将计时器状态设为 FALSE

Top
SetPriorityClass
SetPriorityClass

VB 声明
Declare Function SetPriorityClass Lib "kernel32" Alias "SetPriorityClass" (ByVal hProcess As Long, ByVal dwPriorityClass As Long) As Long
说明
　设置一个进程的优先级别
返回值
Long，进程的优先级，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hProcessLong，指定一个进程句柄
dwPriorityClassLong，指定一个新优先级类的一个常数，请参考 CreateProcess 函数

Top
SetProcessShutdownParameters
SetProcessShutdownParameters

VB 声明
Declare Function SetProcessShutdownParameters Lib "kernel32" Alias "SetProcessShutdownParameters" (ByVal dwLevel As Long, ByVal dwFlags As Long) As Long
说明
　在系统关闭期间，为指定进程设置他相对于其它程序的关闭顺序
返回值
Long，非零表示成功，零表示失败。
参数表
参数类型及说明
lpdwLevelLong，从 0 到&H4FF 的一个值。编号越高程序在系统关闭时越早关闭。
lpdwFlagsLong，指定 SHUTDOWN_NORETRY 标志。如关闭一个进程时，会出现一个重试对话框。通过设置这个标志，就可以禁止那个对话框出现，并直接关闭进程
适用平台
　Wnidows NT

Top
SetProcessWorkingSetSize
SetProcessWorkingSetSize

VB 声明
Declare Function SetProcessWorkingSetSize Lib "kernel32" Alias "SetProcessWorkingSetSize" (ByVal hProcess As Long, ByVal dwMinimumWorkingSetSize As Long, ByVal dwMaximumWorkingSetSize As Long) As Long
说明
　设置操作系统实际划分给进程使用的内存容量
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError

参数表
参数类型及说明
hProcessLong，指定一个进程的句柄
lpMinimumWorkingSetSizeLong，用于装载最小进程容量的一个变量
lpMaximumWorkingSetSizeLong，用于装载最大进程容量的一个变量
适用平台
Windows NT

Top
SetThreadPriority
SetThreadPriority

VB 声明
Declare Function SetThreadPriority Lib "kernel32" Alias "SetThreadPriority" (ByVal hThread As Long, ByVal nPriority As Long) As Long
说明
　设定线程的优先级别
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hThreadLong，线程句柄
nPriorityLong，返回带有 THREAD_PRIORITY_???前缀的某个函数，它定义了线程的优级。
注解
线程的优先级同进程的优先级类组合在一起就决定了线程的实际优先级

Top
ShellExecute
ShellExecute

VB 声明
Declare Function ShellExecute Lib "shell32.dll" Alias "ShellExecuteA" (ByVal hwnd As Long, ByVal lpOperation As String, ByVal lpFile As String, ByVal lpParameters As String, ByVal lpDirectory As String, ByVal nShowCmd As Long) As Long
说明
　查找与指定文件关联在一起的程序的文件名
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，指定一个窗口的句柄，有时候，windows 程序有必要在创建自己的主窗口前显示一个消息框
lpOperationString，指定字串“open”来打开 lpFlie 文档，或指定“Print”来打印它
lpFileString，想用关联程序打印或打开一个程序名或文件名

lpParametersString，如 lpszFlie 是可执行文件，则这个字串包含传递给执行程序的参数
lpDirectoryString，想使用的完整路径
nShowCmdLong，定义了如何显示启动程序的常数值。参考 ShowWindow 函数的 nCmdShow 参数

Top
TerminateProcess
TerminateProcess

VB 声明
Declare Function TerminateProcess Lib "kernel32" Alias "TerminateProcess" (ByVal hProcess As Long, ByVal uExitCode As Long) As Long
说明
结束一个进程
在 VB 里使用
可以使用，但尽量不用
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hProcessLong，指定要中断的一个进程的句柄
uExitCodeLong，进程的一个退出代码

Top
WinExec
WinExec

VB 声明
Declare Function WinExec Lib "kernel32" Alias "WinExec" (ByVal lpCmdLine As String, ByVal nCmdShow As Long) As Long
说明
运行指定的程序
返回值
Long，大于 32 表示成功，请参考 FindExecutable 函数
参数表
参数类型及说明
lpCmdLineString，包含要执行的命令行
nCmdShowLong，定义了以怎样的形式启动程序的常数值。参考 ShowWindow 函数的 nCmdShow 参数
注解
请参考对 CreateProcess 函数的说明，了解在目录中查找指定文件的顺序

Top

VB API 大全

所有类别
类别
控件与消息函数共 91 个函数
硬件与系统函数共 98 个函数
设备场景函数共 73 个函数
绘图函数共 105 个函数
位图、图标和光栅运算函数共 39 个函数
菜单函数共 37 个函数
文本和字体函数共 41 个函数
打印函数共 66 个函数
文件处理函数共 118 个函数
进程和线程函数共 40 个函数
Windows 消息函数共 11 个函数
网络函数共 14 个函数
Windows 消息函数
Windows 消息函数，共一页。第一页
BroadcastSystemMessage 将一条系统消息广播给系统中所有的顶级窗口
GetMessagePos 取得消息队列中上一条消息处理完毕时的鼠标指针屏幕位置
GetMessageTime 取得消息队列中上一条消息处理完毕时的时间
PostMessage 将一条消息投递到指定窗口的消息队列
PostThreadMessage 将一条消息投递给应用程序
RegisterWindowMessage 获取分配给一个字串标识符的消息编号
ReplyMessage 答复一个消息
SendMessage 调用一个窗口的窗口函数，将一条消息发给那个窗口
SendMessageCallback 将一条消息发给窗口
SendMessageTimeout 向窗口发送一条消息
SendNotifyMessage 向窗口发送一条消息
完
第一页
Windows 消息函数，共一页。第一页
BroadcastSystemMessage 将一条系统消息广播给系统中所有的顶级窗口
GetMessagePos 取得消息队列中上一条消息处理完毕时的鼠标指针屏幕位置
GetMessageTime 取得消息队列中上一条消息处理完毕时的时间
PostMessage 将一条消息投递到指定窗口的消息队列
PostThreadMessage 将一条消息投递给应用程序
RegisterWindowMessage 获取分配给一个字串标识符的消息编号
ReplyMessage 答复一个消息
SendMessage 调用一个窗口的窗口函数，将一条消息发给那个窗口
SendMessageCallback 将一条消息发给窗口
SendMessageTimeout 向窗口发送一条消息
SendNotifyMessage 向窗口发送一条消息
完

文件处理函数
文件处理函数，共八页。第一页，第二页，第三页，第四页，第五页，第六页，第七页，第八页
CloseHandle 关闭一个内核对象。其中包括文件、文件映射、进程、线程、安全和同步对象等
CompareFileTime 对比两个文件的时间
CopyFile 复制文件
CreateDirectory 创建一个新目录
CreateFile 打开和创建文件、管道、邮槽、通信服务、设备以及控制台
CreateFileMapping 创建一个新的文件映射对象
DeleteFile 删除指定文件
DeviceIoControl 对设备执行指定的操作
DosDateTimeToFileTime 将 DOS 日期和时间值转换成一个 win32 FILETIME 值
FileTimeToDosDateTime 将一个 win32 FILETIME 值转换成 DOS 日期和时间值
FileTimeToLocalFileTime 将一个 FILETIME 结构转换成本地时间
FileTimeToSystemTime 根据一个 FILETIME 结构的内容，装载一个 SYSTEMTIME 结构
FindClose 关闭由 FindFirstFile 函数创建的一个搜索句柄
FindFirstFile 根据文件名查找文件
FindNextFile 根据调用 FindFirstFile 函数时指定的一个文件名查找下一个文件
FlushFileBuffers 针对指定的文件句柄，刷新内部文件缓冲区
第一页
文件处理函数，共八页。第一页，第二页，第三页，第四页，第五页，第六页，第七页，第八页
CloseHandle 关闭一个内核对象。其中包括文件、文件映射、进程、线程、安全和同步对象等
CompareFileTime 对比两个文件的时间
CopyFile 复制文件
CreateDirectory 创建一个新目录
CreateFile 打开和创建文件、管道、邮槽、通信服务、设备以及控制台
CreateFileMapping 创建一个新的文件映射对象
DeleteFile 删除指定文件
DeviceIoControl 对设备执行指定的操作
DosDateTimeToFileTime 将 DOS 日期和时间值转换成一个 win32 FILETIME 值
FileTimeToDosDateTime 将一个 win32 FILETIME 值转换成 DOS 日期和时间值
FileTimeToLocalFileTime 将一个 FILETIME 结构转换成本地时间
FileTimeToSystemTime 根据一个 FILETIME 结构的内容，装载一个 SYSTEMTIME 结构
FindClose 关闭由 FindFirstFile 函数创建的一个搜索句柄
FindFirstFile 根据文件名查找文件
FindNextFile 根据调用 FindFirstFile 函数时指定的一个文件名查找下一个文件
FlushFileBuffers 针对指定的文件句柄，刷新内部文件缓冲区
第二页
文件处理函数，共八页。第一页，第二页，第三页，第四页，第五页，第六页，第七页，第八页
FlushViewOfFile 将写入文件映射缓冲区的所有数据都刷新到磁盘

GetBinaryType 判断文件是否可以执行
GetCompressedFileSize 判断一个压缩文件在磁盘上实际占据的字节数
GetCurrentDirectory 在一个缓冲区中装载当前目录
GetDiskFreeSpace 获取与一个磁盘的组织有关的信息，以及了解剩余空间的容量
GetDiskFreeSpaceEx 获取与一个磁盘的组织以及剩余空间容量有关的信息
GetDriveType 判断一个磁盘驱动器的类型
GetExpandedName 取得一个压缩文件的全名
GetFileAttributes 判断指定文件的属性
GetFileInformationByHandle 这个函数提供了获取文件信息的一种机制
GetFileSize 判断文件长度
GetFileTime 取得指定文件的时间信息
GetFileType 在给出文件句柄的前提下，判断文件类型
GetFileVersionInfo 从支持版本标记的一个模块里获取文件版本信息
GetFileVersionInfoSize 针对包含了版本资源的一个文件，判断容纳文件版本信息需要一个多大的缓冲区
GetFullPathName 获取指定文件的完整路径名
第三页
文件处理函数，共八页。第一页，第二页，第三页，第四页，第五页，第六页，第七页，第八页
GetLogicalDrives 判断系统中存在哪些逻辑驱动器字母
GetLogicalDriveStrings 获取一个字串，其中包含了当前所有逻辑驱动器的根驱动器路径
GetOverlappedResult 判断一个重叠操作当前的状态
GetPrivateProfileInt 为初始化文件（.ini 文件）中指定的条目获取一个整数值
GetPrivateProfileSection 获取指定小节（在.ini 文件中）所有项名和值的一个列表
GetPrivateProfileString 为初始化文件中指定的条目取得字串
GetProfileInt 取得 win.ini 初始化文件中指定条目的一个整数值
GetProfileSection 获取指定小节（在 win.ini 文件中）所有项名和值的一个列表
GetProfileString 为 win.ini 初始化文件中指定的条目取得字串
GetShortPathName 获取指定文件的短路径名
GetSystemDirectory 取得 Windows 系统目录（即 System 目录）的完整路径名
GetTempFileName 这个函数包含了一个临时文件的名字，它可由应用程序使用
GetTempPath 获取为临时文件指定的路径
GetVolumeInformation 获取与一个磁盘卷有关的信息
GetWindowsDirectory 获取 Windows 目录的完整路径名
hread 参考 lread
第四页
文件处理函数，共八页。第一页，第二页，第三页，第四页，第五页，第六页，第七页，第八页
hwrite 参考 lwrite 函数
lclose 关闭指定的文件
lcreat 创建一个文件
llseek 设置文件中进行读写的当前位置
LockFile 锁定文件的某一部分，使其不与其他应用程序共享
LockFileEx 与 LockFile 相似，只是它提供了更多的功能

lopen 以二进制模式打开指定的文件
lread 将文件中的数据读入内存缓冲区
lwrite 将数据从内存缓冲区写入一个文件
LZClose 关闭由 LZOpenFile 或 LZInit 函数打开的一个文件
LZCopy 复制一个文件
LZInit 这个函数用于初始化内部缓冲区
LZOpenFile 该函数能执行大量不同的文件处理，而且兼容于压缩文件
LZRead 将数据从文件读入内存缓冲区
LZSeek 设置一个文件中进行读写的当前位置
MapViewOfFile 将一个文件映射对象映射到当前应用程序的地址空间
第五页
文件处理函数，共八页。第一页，第二页，第三页，第四页，第五页，第六页，第七页，第八页
MoveFile 移动文件
OpenFile 这个函数能执行大量不同的文件操作
OpenFileMapping 打开一个现成的文件映射对象
QueryDosDevice 在 Windows NT 中，DOS 设备名会映射成 NT 系统设备名。该函数可判断当前的设备映射情况
ReadFile 从文件中读出数据
ReadFileEx 与 ReadFile 相似，只是它只能用于异步读操作，并包含了一个完整的回调
RegCloseKey 关闭系统注册表中的一个项（或键）
RegConnectRegistry 访问远程系统的部分注册表
RegCreateKey 在指定的项下创建或打开一个项
RegCreateKeyEx 在指定项下创建新项的更复杂的方式。在 Win32 环境中建议使用这个函数
RegDeleteKey 删除现有项下方一个指定的子项
RegDeleteValue 删除指定项下方的一个值
RegEnumKey 枚举指定项的子项。在 Win32 环境中应使用 RegEnumKeyEx
RegEnumKeyEx 枚举指定项下方的子项
RegEnumValue 枚举指定项的值
RegFlushKey 将对项和它的子项作出的改动实际写入磁盘
第六页
文件处理函数，共八页。第一页，第二页，第三页，第四页，第五页，第六页，第七页，第八页
RegGetKeySecurity 获取与一个注册表项有关的安全信息
RegLoadKey 从以前用 RegSaveKey 函数创建的一个文件里装载注册表信息
RegNotifyChangeKeyValue 注册表项或它的任何一个子项发生变化时，用这个函数提供一种通知机制
RegOpenKey 打开一个现有的注册表项
RegOpenKeyEx 打开一个现有的项。在 win32 下推荐使用这个函数
RegQueryInfoKey 获取与一个项有关的信息
RegQueryValue 取得指定项或子项的默认（未命名）值
RegQueryValueEx 获取一个项的设置值
RegReplaceKey 用一个磁盘文件保存的信息替换注册表信息；并创建一个备份，在其中包含当前注册表信息

RegRestoreKey 从一个磁盘文件恢复注册表信息
RegSaveKey 将一个项以及它的所有子项都保存到一个磁盘文件
RegSetKeySecurity 设置指定项的安全特性
RegSetValue 设置指定项或子项的默认值
RegSetValueEx 设置指定项的值
RegUnLoadKey 卸载指定的项以及它的所有子项
RemoveDirectory 删除指定目录
第七页
文件处理函数，共八页。第一页，第二页，第三页，第四页，第五页，第六页，第七页，第八页
SearchPath 查找指定文件
SetCurrentDirectory 设置当前目录
SetEndOfFile 针对一个打开的文件，将当前文件位置设为文件末尾
SetFileAttributes 设置文件属性
SetFilePointer 在一个文件中设置当前的读写位置
SetFileTime 设置文件的创建、访问及上次修改时间
SetHandleCount 这个函数不必在 win32 下使用；即使使用，也不会有任何效果
SetVolumeLabel 设置一个磁盘的卷标（Label）
SystemTimeToFileTime 根据一个 FILETIME 结构的内容，载入一个 SYSTEMTIME 结构
UnlockFile 解除对一个文件的锁定
UnlockFileEx 解除对一个文件的锁定
UnmapViewOfFile 在当前应用程序的内存地址空间解除对一个文件映射对象的映射
VerFindFile 用这个函数决定一个文件应安装到哪里
VerInstallFile 用这个函数安装一个文件
VerLanguageName 这个函数能根据 16 位语言代码获取一种语言的名称
VerQueryValue 这个函数用于从版本资源中获取信息
第八页
文件处理函数，共八页。第一页，第二页，第三页，第四页，第五页，第六页，第七页，第八页
WriteFile 将数据写入一个文件
WriteFileEx 与 WriteFile 类似，只是它只能用于异步写操作，并包括了一个完整的回调
WritePrivateProfileSection 为一个初始化文件（.ini）中指定的小节设置所有项名和值
WritePrivateProfileString 在初始化文件指定小节内设置一个字串
WriteProfileSection 为 Win.ini 初始化文件中一个指定的小节设置所有项名和值
WriteProfileString 在 Win.ini 初始化文件指定小节内设置一个字串
完
网络函数
网络函数，共一页。第一页
WNetAddConnection 创建同一个网络资源的永久性连接
WNetAddConnection2 创建同一个网络资源的连接
WNetAddConnection3 创建同一个网络资源的连接
WNetCancelConnection 结束一个网络连接
WNetCancelConnection2 结束一个网络连接
WNetCloseEnum 结束一次枚举操作

WNetConnectionDialog 启动一个标准对话框，以便建立同网络资源的连接
WNetDisconnectDialog 启动一个标准对话框，以便断开同网络资源的连接
WNetEnumResource 枚举网络资源
WNetGetConnection 获取本地或已连接的一个资源的网络名称
WNetGetLastError 获取网络错误的扩展错误信息
WNetGetUniversalName 获取网络中一个文件的远程名称以及/或者 UNC（统一命名规范）名称
WNetGetUser 获取一个网络资源用以连接的名字
WNetOpenEnum 启动对网络资源进行枚举的过程
完
第一页
网络函数，共一页。第一页
WNetAddConnection 创建同一个网络资源的永久性连接
WNetAddConnection2 创建同一个网络资源的连接
WNetAddConnection3 创建同一个网络资源的连接
WNetCancelConnection 结束一个网络连接
WNetCancelConnection2 结束一个网络连接
WNetCloseEnum 结束一次枚举操作
WNetConnectionDialog 启动一个标准对话框，以便建立同网络资源的连接
WNetDisconnectDialog 启动一个标准对话框，以便断开同网络资源的连接
WNetEnumResource 枚举网络资源
WNetGetConnection 获取本地或已连接的一个资源的网络名称
WNetGetLastError 获取网络错误的扩展错误信息
WNetGetUniversalName 获取网络中一个文件的远程名称以及/或者 UNC（统一命名规范）名称
WNetGetUser 获取一个网络资源用以连接的名字
WNetOpenEnum 启动对网络资源进行枚举的过程
完
菜单函数
菜单函数：共三页。第一页，第二页，第三页
AppendMenu 在指定的菜单里添加一个菜单项
CheckMenuItem 复选或撤消复选指定的菜单条目
CheckMenuRadioItem 指定一个菜单条目被复选成“单选”项目
CreateMenu 创建新菜单
CreatePopupMenu 创建一个空的弹出式菜单
DeleteMenu 删除指定的菜单条目
DestroyMenu 删除指定的菜单
DrawMenuBar 为指定的窗口重画菜单
EnableMenuItem 允许或禁止指定的菜单条目
GetMenu 取得窗口中一个菜单的句柄
GetMenuCheckMarkDimensions 返回一个菜单复选符的大小
GetMenuContextHelpId 取得一个菜单的帮助场景 ID
GetMenuDefaultItem 判断菜单中的哪个条目是默认条目
GetMenuItemCount 返回菜单中条目（菜单项）的数量

GetMenuItemID 返回位于菜单中指定位置处的条目的菜单 ID
GetMenuItemInfo 取得（接收）与一个菜单条目有关的特定信息
第一页
菜单函数：共三页。第一页，第二页，第三页
AppendMenu 在指定的菜单里添加一个菜单项
CheckMenuItem 复选或撤消复选指定的菜单条目
CheckMenuRadioItem 指定一个菜单条目被复选成“单选”项目
CreateMenu 创建新菜单
CreatePopupMenu 创建一个空的弹出式菜单
DeleteMenu 删除指定的菜单条目
DestroyMenu 删除指定的菜单
DrawMenuBar 为指定的窗口重画菜单
EnableMenuItem 允许或禁止指定的菜单条目
GetMenu 取得窗口中一个菜单的句柄
GetMenuCheckMarkDimensions 返回一个菜单复选符的大小
GetMenuContextHelpId 取得一个菜单的帮助场景 ID
GetMenuDefaultItem 判断菜单中的哪个条目是默认条目
GetMenuItemCount 返回菜单中条目（菜单项）的数量
GetMenuItemID 返回位于菜单中指定位置处的条目的菜单 ID
GetMenuItemInfo 取得（接收）与一个菜单条目有关的特定信息
第二页
菜单函数：共三页。第一页，第二页，第三页
GetMenuItemRect 在一个矩形中装载指定菜单条目的屏幕坐标信息
GetMenuState 取得与指定菜单条目状态有关的信息
GetMenuString 取得指定菜单条目的字串
GetSubMenu 取得一个弹出式菜单的句柄，它位于菜单中指定的位置
GetSystemMenu 取得指定窗口的系统菜单的句柄
HiliteMenuItem 控制顶级菜单条目的加亮显示状态
InsertMenu 在菜单的指定位置处插入一个菜单条目，并根据需要将其他条目向下移动
InsertMenuItem 插入一个新菜单条目
IsMenu 判断指定的句柄是否为一个菜单的句柄
LoadMenu 从指定的模块或应用程序实例中载入一个菜单
LoadMenuIndirect 载入一个菜单
MenuItemFromPoint 判断哪个菜单条目包含了屏幕上一个指定的点
ModifyMenu 改变菜单条目
RemoveMenu 删除指定的菜单条目
SetMenu 设置窗口菜单
SetMenuContextHelpId 设置一个菜单的帮助场景 ID
第三页
菜单函数：共三页。第一页，第二页，第三页
SetMenuDefaultItem 将一个菜单条目设为默认条目
SetMenuItemBitmaps 设置一幅特定位图，令其在指定的菜单条目中使用，代替标准的复选符号（√）
SetMenuItemInfo 为一个菜单条目设置指定的信息

TrackPopupMenu 在屏幕的任意地方显示一个弹出式菜单
TrackPopupMenuEx 与 TrackPopupMenu 相似，只是它提供了额外的功能
完
文本和字体函数
文本和字体函数，共三页。第一页，第二页，第三页
AddFontResource 在 Windows 系统中添加一种字体资源
CreateFont 用指定的属性创建一种逻辑字体
CreateFontIndirect 用指定的属性创建一种逻辑字体
CreateScalableFontResource 为一种 TureType 字体创建一个资源文件，以便能用 API 函数 AddFontResource 将其加入 Windows 系统
DrawText 将文本描绘到指定的矩形中
DrawTextEx 与 DrawText 相似，只是加入了更多的功能
EnumFontFamilies 列举指定设备可用的字体
EnumFontFamiliesEx 列举指定设备可用的字体
EnumFonts 列举指定设备可用的字体
ExtTextOut 经过扩展的文本描绘函数。也请参考 SetTextAlign 函数
GetAspectRatioFilterEx 用 SetMapperFlags 要求 Windows 只选择与设备当前纵横比相符的光栅字体时，本函数可判断纵横比大小
GetCharABCWidths 判断 TureType 字体中一个或多个字符的 A-B-C 大小
GetCharABCWidthsFloat 查询一种字体中一个或多个字符的 A-B-C 尺寸
GetCharacterPlacement 该函数用于了解如何用一个给定的字符显示一个字串
GetCharWidth 调查字体中一个或多个字符的宽度
GetFontData 接收一种可缩放字体文件的数据
第一页
文本和字体函数，共三页。第一页，第二页，第三页
AddFontResource 在 Windows 系统中添加一种字体资源
CreateFont 用指定的属性创建一种逻辑字体
CreateFontIndirect 用指定的属性创建一种逻辑字体
CreateScalableFontResource 为一种 TureType 字体创建一个资源文件，以便能用 API 函数 AddFontResource 将其加入 Windows 系统
DrawText 将文本描绘到指定的矩形中
DrawTextEx 与 DrawText 相似，只是加入了更多的功能
EnumFontFamilies 列举指定设备可用的字体
EnumFontFamiliesEx 列举指定设备可用的字体
EnumFonts 列举指定设备可用的字体
ExtTextOut 经过扩展的文本描绘函数。也请参考 SetTextAlign 函数
GetAspectRatioFilterEx 用 SetMapperFlags 要求 Windows 只选择与设备当前纵横比相符的光栅字体时，本函数可判断纵横比大小
GetCharABCWidths 判断 TureType 字体中一个或多个字符的 A-B-C 大小
GetCharABCWidthsFloat 查询一种字体中一个或多个字符的 A-B-C 尺寸
GetCharacterPlacement 该函数用于了解如何用一个给定的字符显示一个字串
GetCharWidth 调查字体中一个或多个字符的宽度
GetFontData 接收一种可缩放字体文件的数据
第二页

文本和字体函数，共三页。第一页，第二页，第三页
GetFontLanguageInfo 返回目前选入指定设备场景中的字体的信息
GetGlyphOutline 取得 TureType 字体中构成一个字符的曲线信息
GetKerningPairs 取得指定字体的字距信息
GetOutlineTextMetrics 接收与 TureType 字体内部特征有关的详细信息
GetRasterizerCaps 了解系统是否有能力支持可缩放的字体
GetTabbedTextExtent 判断一个字串占据的范围，同时考虑制表站扩充的因素
GetTextAlign 接收一个设备场景当前的文本对齐标志
GetTextCharacterExtra 判断额外字符间距的当前值
GetTextCharset 接收当前选入指定设备场景的字体的字符集标识符
GetTextCharsetInfo 获取与当前选定字体的字符集有关的详细信息
GetTextColor 判断当前字体颜色。通常也称为“前景色”
GetTextExtentExPoint 判断要填入指定区域的字符数量。也用一个数组装载每个字符的范围信息
GetTextExtentPoint 判断一个字串的大小（范围）
GetTextFace 获取一种字体的字样名
GetTextMetrics 获取与选入一种设备场景的物理字体有关的信息
GrayString 描绘一个以灰色显示的字串。通常由 Windows 用于标识禁止状态
第三页
文本和字体函数，共三页。第一页，第二页，第三页
PolyTextOut 描绘一系列字串
RemoveFontResource 从 Windows 系统中删除一种字体资源
SetMapperFlagsWindows 对字体进行映射时，可用该函数选择与目标设备的纵横比相符的光栅字体
SetTextAlign 设置文本对齐方式，并指定在文本输出过程中使用设备场景的当前位置
SetTextCharacterExtra 描绘文本的时候，指定要在字符间插入的额外间距
SetTextColor 设置当前文本颜色。这种颜色也称为“前景色”
SetTextJustification 通过指定一个文本行应占据的额外空间，可用这个函数对文本进行两端对齐处理
TabbedTextOut 支持制表站的一个文本描绘函数
TextOut 文本绘图函数
完
硬件与系统函数
硬件与系统函数：共七页。第一页，第二页，第三页，第四页，第五页，第六页，第七页
ActivateKeyboardLayout 激活一个新的键盘布局。键盘布局定义了按键在一种物理性键盘上的位置与含义
Beep 用于生成简单的声音
CharToOem 将一个字串从 ANSI 字符集转换到 OEM 字符集
ClipCursor 将指针限制到指定区域
ConvertDefaultLocale 将一个特殊的地方标识符转换成真实的地方 ID
CreateCaret 根据指定的信息创建一个插入符（光标），并将它选定为指定窗口的默认插入符
DestroyCaret 清除（破坏）一个插入符
EnumCalendarInfo 枚举在指定“地方”环境中可用的日历信息
EnumDateFormats 列举指定的“当地”设置中可用的长、短日期格式

EnumSystemCodePages 枚举系统中已安装或支持的代码页
EnumSystemLocales 枚举系统已经安装或提供支持的“地方”设置
EnumTimeFormats 枚举一个指定的地方适用的时间格式
ExitWindowsEx 退出 windows，并用特定的选项重新启动
ExpandEnvironmentStrings 扩充环境字串
FreeEnvironmentStrings 翻译指定的环境字串块
GetACP 判断目前正在生效的 ANSI 代码页
第一页
硬件与系统函数：共七页。第一页，第二页，第三页，第四页，第五页，第六页，第七页
ActivateKeyboardLayout 激活一个新的键盘布局。键盘布局定义了按键在一种物理性键盘上的位置与含义
Beep 用于生成简单的声音
CharToOem 将一个字串从 ANSI 字符集转换到 OEM 字符集
ClipCursor 将指针限制到指定区域
ConvertDefaultLocale 将一个特殊的地方标识符转换成真实的地方 ID
CreateCaret 根据指定的信息创建一个插入符（光标），并将它选定为指定窗口的默认插入符
DestroyCaret 清除（破坏）一个插入符
EnumCalendarInfo 枚举在指定“地方”环境中可用的日历信息
EnumDateFormats 列举指定的“当地”设置中可用的长、短日期格式
EnumSystemCodePages 枚举系统中已安装或支持的代码页
EnumSystemLocales 枚举系统已经安装或提供支持的“地方”设置
EnumTimeFormats 枚举一个指定的地方适用的时间格式
ExitWindowsEx 退出 windows，并用特定的选项重新启动
ExpandEnvironmentStrings 扩充环境字串
FreeEnvironmentStrings 翻译指定的环境字串块
GetACP 判断目前正在生效的 ANSI 代码页
第二页
硬件与系统函数：共七页。第一页，第二页，第三页，第四页，第五页，第六页，第七页
GetAsyncKeyState 判断函数调用时指定虚拟键的状态
GetCaretBlinkTime 判断插入符光标的闪烁频率
GetCaretPos 判断插入符的当前位置
GetClipCursor 取得一个矩形，用于描述目前为鼠标指针规定的剪切区域
GetCommandLine 获得指向当前命令行缓冲区的一个指针
GetComputerName 取得这台计算机的名称
GetCPInfo 取得与指定代码页有关的信息
GetCurrencyFormat 针对指定的“地方”设置，根据货币格式格式化一个数字
GetCursor 获取目前选择的鼠标指针的句柄
GetCursorPos 获取鼠标指针的当前位置
GetDateFormat 针对指定的“当地”格式，对一个系统日期进行格式化
GetDoubleClickTime 判断连续两次鼠标单击之间会被处理成双击事件的间隔时间
GetEnvironmentStrings 为包含了当前环境字串设置的一个内存块分配和返回一个句柄
GetEnvironmentVariable 取得一个环境变量的值
GetInputState 判断是否存在任何待决（等待处理）的鼠标或键盘事件
GetKBCodePage 由 GetOEMCP 取代，两者功能完全相同

第三页
硬件与系统函数：共七页。第一页，第二页，第三页，第四页，第五页，第六页，第七页
GetKeyboardLayout 取得一个句柄，描述指定应用程序的键盘布局
GetKeyboardLayoutList 获得系统适用的所有键盘布局的一个列表
GetKeyboardLayoutName 取得当前活动键盘布局的名称
GetKeyboardState 取得键盘上每个虚拟键当前的状态
GetKeyboardType 了解与正在使用的键盘有关的信息
GetKeyNameText 在给出扫描码的前提下，判断键名
GetKeyState 针对已处理过的按键，在最近一次输入信息时，判断指定虚拟键的状态
GetLastError 针对之前调用的 api 函数，用这个函数取得扩展错误信息
GetLocaleInfo 取得与指定“地方”有关的信息
GetLocalTime 取得本地日期和时间
GetNumberFormat 针对指定的“地方”，按特定的格式格式化一个数字
GetOEMCP 判断在 OEM 和 ANSI 字符集间转换的 windows 代码页
GetQueueStatus 判断应用程序消息队列中待决（等待处理）的消息类型
GetSysColor 判断指定 windows 显示对象的颜色
GetSystemDefaultLangID 取得系统的默认语言 ID
GetSystemDefaultLCID 取得当前的默认系统“地方”
第四页
硬件与系统函数：共七页。第一页，第二页，第三页，第四页，第五页，第六页，第七页
GetSystemInfo 取得与底层硬件平台有关的信息
GetSystemMetrics 返回与 windows 环境有关的信息
GetSystemPowerStatus 获得与当前系统电源状态有关的信息
GetSystemTime 取得当前系统时间，这个时间采用的是“协同世界时间”（即 UTC，也叫做 GMT）格式
GetSystemTimeAdjustment 使内部系统时钟与一个外部的时钟信号源同步
GetThreadLocale 取得当前线程的地方 ID
GetTickCount 用于获取自 windows 启动以来经历的时间长度（毫秒）
GetTimeFormat 针对当前指定的“地方”，按特定的格式格式化一个系统时间
GetTimeZoneInformation 取得与系统时区设置有关的信息
GetUserDefaultLangID 为当前用户取得默认语言 ID
GetUserDefaultLCID 取得当前用户的默认“地方”设置
GetUserName 取得当前用户的名字
GetVersion 判断当前运行的 Windows 和 DOS 版本
GetVersionEx 取得与平台和操作系统有关的版本信息
HideCaret 在指定的窗口隐藏插入符（光标）
IsValidCodePage 判断一个代码页是否有效
第五页
硬件与系统函数：共七页。第一页，第二页，第三页，第四页，第五页，第六页，第七页
IsValidLocale 判断地方标识符是否有效
keybd_event 这个函数模拟了键盘行动
LoadKeyboardLayout 载入一个键盘布局
MapVirtualKey 根据指定的映射类型，执行不同的扫描码和字符转换
MapVirtualKeyEx 根据指定的映射类型，执行不同的扫描码和字符转换

MessageBeep 播放一个系统声音。系统声音的分配方案是在控制面板里决定的
mouse_event 模拟一次鼠标事件
OemKeyScan 判断 OEM 字符集中的一个 ASCII 字符的扫描码和 Shift 键状态
OemToChar 将 OEM 字符集的一个字串转换到 ANSI 字符集
SetCaretBlinkTime 指定插入符（光标）的闪烁频率
SetCaretPos 指定插入符的位置
SetComputerName 设置新的计算机名
SetCursor 将指定的鼠标指针设为当前指针
SetCursorPos 设置指针的位置
SetDoubleClickTime 设置连续两次鼠标单击之间能使系统认为是双击事件的间隔时间
SetEnvironmentVariable 将一个环境变量设为指定的值
第六页
硬件与系统函数：共七页。第一页，第二页，第三页，第四页，第五页，第六页，第七页
SetKeyboardState 设置每个虚拟键当前在键盘上的状态
SetLocaleInfo 改变用户“地方”设置信息
SetLocalTime 设置当前地方时间
SetSysColors 设置指定窗口显示对象的颜色
SetSystemCursor 改变任何一个标准系统指针
SetSystemTime 设置当前系统时间
SetSystemTimeAdjustment 定时添加一个校准值使内部系统时钟与一个外部的时钟信号源同步
SetThreadLocale 为当前线程设置地方
SetTimeZoneInformation 设置系统时区信息
ShowCaret 在指定的窗口里显示插入符（光标）
ShowCursor 控制鼠标指针的可视性
SwapMouseButton 决定是否互换鼠标左右键的功能
SystemParametersInfo 获取和设置数量众多的 windows 系统参数
SystemTimeToTzSpecificLocalTime 将系统时间转换成地方时间
ToAscii 根据当前的扫描码和键盘信息，将一个虚拟键转换成 ASCII 字符
ToUnicode 根据当前的扫描码和键盘信息，将一个虚拟键转换成 Unicode 字符
第七页
硬件与系统函数：共七页。第一页，第二页，第三页，第四页，第五页，第六页，第七页
UnloadKeyboardLayout 卸载指定的键盘布局
VkKeyScan 针对 Windows 字符集中一个 ASCII 字符，判断虚拟键码和 Shift 键的状态
完
控件与消息函数
控件与消息函数：共六页。第一页，第二页，第三页，第四页，第五页，第六页
AdjustWindowRect 给定一种窗口样式，计算获得目标客户区矩形所需的窗口大小
AnyPopup 判断屏幕上是否存在任何弹出式窗口
ArrangeIconicWindows 排列一个父窗口的最小化子窗口
AttachThreadInput 连接线程输入函数
BeginDeferWindowPos 启动构建一系列新窗口位置的过程
BringWindowToTop 将指定的窗口带至窗口列表顶部
CascadeWindows 以层叠方式排列窗口

ChildWindowFromPoint 返回父窗口中包含了指定点的第一个子窗口的句柄
ClientToScreen 判断窗口内以客户区坐标表示的一个点的屏幕坐标
CloseWindow 最小化指定的窗口
CopyRect 矩形内容复制
DeferWindowPos 该函数为特定的窗口指定一个新窗口位置
DestroyWindow 清除指定的窗口以及它的所有子窗口
DrawAnimatedRects 描绘一系列动态矩形
EnableWindow 指定的窗口里允许或禁止所有鼠标及键盘输入
EndDeferWindowPos 同时更新 DeferWindowPos 调用时指定的所有窗口的位置及状态

AdjustWindowRect 给定一种窗口样式，计算获得目标客户区矩形所需的窗口大小
AnyPopup 判断屏幕上是否存在任何弹出式窗口
ArrangeIconicWindows 排列一个父窗口的最小化子窗口
AttachThreadInput 连接线程输入函数
BeginDeferWindowPos 启动构建一系列新窗口位置的过程
BringWindowToTop 将指定的窗口带至窗口列表顶部
CascadeWindows 以层叠方式排列窗口
ChildWindowFromPoint 返回父窗口中包含了指定点的第一个子窗口的句柄
ClientToScreen 判断窗口内以客户区坐标表示的一个点的屏幕坐标
CloseWindow 最小化指定的窗口
CopyRect 矩形内容复制
DeferWindowPos 该函数为特定的窗口指定一个新窗口位置
DestroyWindow 清除指定的窗口以及它的所有子窗口
DrawAnimatedRects 描绘一系列动态矩形
EnableWindow 指定的窗口里允许或禁止所有鼠标及键盘输入
EndDeferWindowPos 同时更新 DeferWindowPos 调用时指定的所有窗口的位置及状态

EnumChildWindows 为指定的父窗口枚举子窗口
EnumThreadWindows 枚举与指定任务相关的窗口
EnumWindows 枚举窗口列表中的所有父窗口
EqualRect 判断两个矩形结构是否相同
FindWindow 寻找窗口列表中第一个符合指定条件的顶级窗口
FindWindowEx 在窗口列表中寻找与指定条件相符的第一个子窗口
FlashWindow 闪烁显示指定窗口
GetActiveWindow 获得活动窗口的句柄
GetCapture 获得一个窗口的句柄，这个窗口位于当前输入线程，且拥有鼠标捕获（鼠标活动由它接收）
GetClassInfo 取得 WNDCLASS 结构（或 WNDCLASSEX 结构）的一个副本，结构中包含了与指定类有关的信息
GetClassLong 取得窗口类的一个 Long 变量条目
GetClassName 为指定的窗口取得类名
GetClassWord 为窗口类取得一个整数变量

GetClientRect 返回指定窗口客户区矩形的大小
GetDesktopWindow 获得代表整个屏幕的一个窗口（桌面窗口）句柄
GetFocus 获得拥有输入焦点的窗口的句柄
第三页

GetForegroundWindow 获得前台窗口的句柄
GetLastActivePopup 获得在一个给定父窗口中最近激活过的弹出式窗口的句柄
GetParent 判断指定窗口的父窗口
GetTopWindow 搜索内部窗口列表，寻找隶属于指定窗口的头一个窗口的句柄
GetUpdateRect 获得一个矩形，它描叙了指定窗口中需要更新的那一部分
GetWindow 获得一个窗口的句柄，该窗口与某源窗口有特定的关系
GetWindowContextHelpId 取得与窗口关联在一起的帮助场景 ID
GetWindowLong 从指定窗口的结构中取得信息
GetWindowPlacement 获得指定窗口的状态及位置信息
GetWindowRect 获得整个窗口的范围矩形，窗口的边框、标题栏、滚动条及菜单等都在这个矩形内
GetWindowText 取得一个窗体的标题（caption）文字，或者一个控件的内容
GetWindowTextLength 调查窗口标题文字或控件内容的长短
GetWindowWord 获得指定窗口结构的信息
InflateRect 增大或减小一个矩形的大小
IntersectRect 这个函数在 lpDestRect 里载入一个矩形，它是 lpSrc1Rect 与 lpSrc2Rect 两个矩形的交集
InvalidateRect 屏蔽一个窗口客户区的全部或部分区域
第四页

IsChild 判断一个窗口是否为另一窗口的子或隶属窗口
IsIconic 判断窗口是否已最小化
IsRectEmpty 判断一个矩形是否为空
IsWindow 判断一个窗口句柄是否有效
IsWindowEnabled 判断窗口是否处于活动状态
IsWindowUnicode 判断一个窗口是否为 Unicode 窗口。这意味着窗口为所有基于文本的消息都接收 Unicode 文字
IsWindowVisible 判断窗口是否可见
IsZoomed 判断窗口是否最大化
LockWindowUpdate 锁定指定窗口，禁止它更新
MapWindowPoints 将一个窗口客户区坐标的点转换到另一窗口的客户区坐标系统
MoveWindow 改变指定窗口的位置和大小
OffsetRect 通过应用一个指定的偏移，从而让矩形移动起来
OpenIcon 恢复一个最小化的程序，并将其激活
PtInRect 判断指定的点是否位于矩形内部
RedrawWindow 重画全部或部分窗口
ReleaseCapture 为当前的应用程序释放鼠标捕获
第五页

ScreenToClient 判断屏幕上一个指定点的客户区坐标
ScrollWindow 滚动窗口客户区的全部或一部分
ScrollWindowEx 根据附加的选项，滚动窗口客户区的全部或部分
SetActiveWindow 激活指定的窗口
SetCapture 将鼠标捕获设置到指定的窗口
SetClassLong 为窗口类设置一个 Long 变量条目
SetClassWord 为窗口类设置一个条目
SetFocusAPI 将输入焦点设到指定的窗口。如有必要，会激活窗口
SetForegroundWindow 将窗口设为系统的前台窗口
SetParent 指定一个窗口的新父
SetRect 设置指定矩形的内容
SetRectEmpty 将矩形设为一个空矩形
SetWindowContextHelpId 为指定的窗口设置帮助场景（上下文）ID
SetWindowLong 在窗口结构中为指定的窗口设置信息
SetWindowPlacement 设置窗口状态和位置信息
SetWindowPos 为窗口指定一个新位置和状态
第六页
控件与消息函数：共六页。第一页，第二页，第三页，第四页，第五页，第六页
SetWindowText 设置窗口的标题文字或控件的内容
SetWindowWord 在窗口结构中为指定的窗口设置信息
ShowOwnedPopups 显示或隐藏由指定窗口所有的全部弹出式窗口
ShowWindow 控制窗口的可见性
ShowWindowAsync 与 ShowWindow 相似
SubtractRect 装载矩形 lprcDst，它是在矩形 lprcSrc1 中减去 lprcSrc2 得到的结果
TileWindows 以平铺顺序排列窗口
UnionRect 装载一个 lpDestRect 目标矩形，它是 lpSrc1Rect 和 lpSrc2Rect 联合起来的结果
UpdateWindow 强制立即更新窗口
ValidateRect 校验窗口的全部或部分客户区
WindowFromPoint 返回包含了指定点的窗口的句柄。忽略屏蔽、隐藏以及透明窗口
完
位图、图标和光栅运算函数
位图、图标和光栅运算函数，共三页。第一页，第二页，第三页
BitBlt 将一幅位图从一个设备场景复制到另一个
CopyIcon 制作指定图标或鼠标指针的一个副本。这个副本从属于发出调用的应用程序
CopyImage 复制位图、图标或指针，同时在复制过程中进行一些转换工作
CreateBitmap 按照规定的格式创建一幅与设备有关位图
CreateBitmapIndirect 创建一幅与设备有关位图
CreateCompatibleBitmap 创建一幅与设备有关位图，它与指定的设备场景兼容
CreateCursor 创建一个鼠标指针
CreateDIBitmap 根据一幅与设备无关的位图创建一幅与设备有关的位图
CreateDIBSection 创建一个 DIBSection
CreateIcon 创建一个图标
CreateIconIndirect 创建一个图标
DestroyCursor 清除指定的鼠标指针，并释放它占用的所有系统资源

DestroyIcon 清除图标
DrawIcon 在指定的位置画一个图标
DrawIconEx 描绘一个图标或鼠标指针。与 DrawIcon 相比，这个函数提供了更多的功能
ExtractAssociatedIcon 判断一个可执行程序或 DLL 中是否存在图标，或是否有图标与系统注册表中指定的文件存在关联并提取之

位图、图标和光栅运算函数，共三页。第一页，第二页，第三页
BitBlt 将一幅位图从一个设备场景复制到另一个
CopyIcon 制作指定图标或鼠标指针的一个副本。这个副本从属于发出调用的应用程序
CopyImage 复制位图、图标或指针，同时在复制过程中进行一些转换工作
CreateBitmap 按照规定的格式创建一幅与设备有关位图
CreateBitmapIndirect 创建一幅与设备有关位图
CreateCompatibleBitmap 创建一幅与设备有关位图，它与指定的设备场景兼容
CreateCursor 创建一个鼠标指针
CreateDIBitmap 根据一幅与设备无关的位图创建一幅与设备有关的位图
CreateDIBSection 创建一个 DIBSection
CreateIcon 创建一个图标
CreateIconIndirect 创建一个图标
DestroyCursor 清除指定的鼠标指针，并释放它占用的所有系统资源
DestroyIcon 清除图标
DrawIcon 在指定的位置画一个图标
DrawIconEx 描绘一个图标或鼠标指针。与 DrawIcon 相比，这个函数提供了更多的功能
ExtractAssociatedIcon 判断一个可执行程序或 DLL 中是否存在图标，或是否有图标与系统注册表中指定的文件存在关联并提取之

位图、图标和光栅运算函数，共三页。第一页，第二页，第三页
ExtractIcon 判断一个可执行文件或 DLL 中是否有图标存在，并将其提取出来
GetBitmapBits 将来自位图的二进制位复制到一个缓冲区
GetBitmapDimensionEx 取得一幅位图的宽度和高度
GetDIBColorTable 从选入设备场景的 DIBSection 中取得颜色表信息
GetDIBits 将来自一幅位图的二进制位复制到一幅与设备无关的位图里
GetIconInfo 取得与图标有关的信息
GetStretchBltMode 判断 StretchBlt 和 StretchDIBits 函数采用的伸缩模式
LoadBitmap 从指定的模块或应用程序实例中载入一幅位图
LoadCursor 从指定的模块或应用程序实例中载入一个鼠标指针
LoadCursorFromFile 在一个指针文件或一个动画指针文件的基础上创建一个指针
LoadIcon 从指定的模块或应用程序实例中载入一个图标
LoadImage 载入一个位图、图标或指针
MaskBlt 执行复杂的图象传输，同时进行掩模（MASK）处理
PatBlt 在当前选定的刷子的基础上，用一个图案填充指定的设备场景
PlgBlt 复制一幅位图，同时将其转换成一个平行四边形。利用它可对位图进行旋转处理
SetBitmapBits 将来自缓冲区的二进制位复制到一幅位图

位图、图标和光栅运算函数，共三页。第一页，第二页，第三页

SetBitmapDimensionEx 设置一幅位图的宽度。以一毫米的十分之一为单位
SetDIBColorTable 设置选入设备场景的一个 DIBSection 的颜色表信息
SetDIBits 将来自与设备无关位图的二进制位复制到一幅与设备有关的位图里
SetDIBitsToDevice 将一幅与设备无关位图的全部或部分数据直接复制到一个设备
SetStretchBltMode 指定 StretchBlt 和 StretchDIBits 函数的伸缩模式
StretchBlt 将一幅位图从一个设备场景复制到另一个
StretchDIBits 将一幅与设备无关位图的全部或部分数据直接复制到指定的设备场景
完
绘图函数
绘图函数：共七页。第一页，第二页，第三页，第四页，第五页，第六页，第七页
AbortPath 抛弃选入指定设备场景中的所有路径。也取消目前正在进行的任何路径的创建工作
AngleArc 用一个连接弧画一条线
Arc 画一个圆弧
BeginPath 启动一个路径分支
CancelDC 取消另一个线程里的长时间绘图操作
Chord 画一个弦
CloseEnhMetaFile 关闭指定的增强型图元文件设备场景，并将新建的图元文件返回一个句柄
CloseFigure 描绘到一个路径时，关闭当前打开的图形
CloseMetaFile 关闭指定的图元文件设备场景，并向新建的图元文件返回一个句柄
CopyEnhMetaFile 制作指定增强型图元文件的一个副本（拷贝）
CopyMetaFile 制作指定（标准）图元文件的一个副本
CreateBrushIndirect 在一个 LOGBRUSH 数据结构的基础上创建一个刷子
CreateDIBPatternBrush 用一幅与设备无关的位图创建一个刷子，以便指定刷子样式（图案）
CreateEnhMetaFile 创建一个增强型的图元文件设备场景
CreateHatchBrush 创建带有阴影图案的一个刷子
CreateMetaFile 创建一个图元文件设备场景
第一页
绘图函数：共七页。第一页，第二页，第三页，第四页，第五页，第六页，第七页
AbortPath 抛弃选入指定设备场景中的所有路径。也取消目前正在进行的任何路径的创建工作
AngleArc 用一个连接弧画一条线
Arc 画一个圆弧
BeginPath 启动一个路径分支
CancelDC 取消另一个线程里的长时间绘图操作
Chord 画一个弦
CloseEnhMetaFile 关闭指定的增强型图元文件设备场景，并将新建的图元文件返回一个句柄
CloseFigure 描绘到一个路径时，关闭当前打开的图形
CloseMetaFile 关闭指定的图元文件设备场景，并向新建的图元文件返回一个句柄
CopyEnhMetaFile 制作指定增强型图元文件的一个副本（拷贝）
CopyMetaFile 制作指定（标准）图元文件的一个副本
CreateBrushIndirect 在一个 LOGBRUSH 数据结构的基础上创建一个刷子
CreateDIBPatternBrush 用一幅与设备无关的位图创建一个刷子，以便指定刷子样式（图案）
CreateEnhMetaFile 创建一个增强型的图元文件设备场景

CreateHatchBrush 创建带有阴影图案的一个刷子
CreateMetaFile 创建一个图元文件设备场景
第二页
绘图函数：共七页。第一页，第二页，第三页，第四页，第五页，第六页，第七页
CreatePatternBrush 用指定了刷子图案的一幅位图创建一个刷子
CreatePen 用指定的样式、宽度和颜色创建一个画笔
CreatePenIndirect 根据指定的 LOGPEN 结构创建一个画笔
CreateSolidBrush 用纯色创建一个刷子
DeleteEnhMetaFile 删除指定的增强型图元文件
DeleteMetaFile 删除指定的图元文件
DeleteObject 删除 GDI 对象，对象使用的所有系统资源都会被释放
DrawEdge 用指定的样式描绘一个矩形的边框
DrawEscape 换码（Escape）函数将数据直接发至显示设备驱动程序
DrawFocusRect 画一个焦点矩形
DrawFrameControl 描绘一个标准控件
DrawState 为一幅图象或绘图操作应用各式各样的效果
Ellipse 描绘一个椭圆，由指定的矩形围绕
EndPath 停止定义一个路径
EnumEnhMetaFile 针对一个增强型图元文件，列举其中单独的图元文件记录
EnumMetaFile 为一个标准的 windows 图元文件枚举单独的图元文件记录
第三页
绘图函数：共七页。第一页，第二页，第三页，第四页，第五页，第六页，第七页
EnumObjects 枚举可随同指定设备场景使用的画笔和刷子
ExtCreatePen 创建一个扩展画笔（装饰或几何）
ExtFloodFill 在指定的设备场景里，用当前选择的刷子填充一个区域
FillPath 关闭路径中任何打开的图形，并用当前刷子填充
FillRect 用指定的刷子填充一个矩形
FlattenPath 将一个路径中的所有曲线都转换成线段
FloodFill 用当前选定的刷子在指定的设备场景中填充一个区域
FrameRect 用指定的刷子围绕一个矩形画一个边框
GdiComment 为指定的增强型图元文件设备场景添加一条注释信息
GdiFlush 执行任何未决的绘图操作
GdiGetBatchLimit 判断有多少个 GDI 绘图命令位于队列中
GdiSetBatchLimit 指定有多少个 GDI 绘图命令能够进入队列
GetArcDirection 画圆弧的时候，判断当前采用的绘图方向
GetBkColor 取得指定设备场景当前的背景颜色
GetBkMode 针对指定的设备场景，取得当前的背景填充模式
GetBrushOrgEx 判断指定设备场景中当前选定刷子起点
第四页
绘图函数：共七页。第一页，第二页，第三页，第四页，第五页，第六页，第七页
GetCurrentObject 获得指定类型的当前选定对象
GetCurrentPositionEx 在指定的设备场景中取得当前的画笔位置
GetEnhMetaFile 取得磁盘文件中包含的一个增强型图元文件的图元文件句柄
GetEnhMetaFileBits 将指定的增强型图元文件复制到一个内存缓冲区里

GetEnhMetaFileDescription 返回对一个增强型图元文件的说明
GetEnhMetaFileHeader 取得增强型图元文件的图元文件头
GetEnhMetaFilePaletteEntries 取得增强型图元文件的全部或部分调色板
GetMetaFile 取得包含在一个磁盘文件中的图元文件的图元文件句柄
GetMetaFileBitsEx 将指定的图元文件复制到一个内存缓冲区
GetMiterLimit 取得设备场景的斜率限制（Miter）设置
GetNearestColor 根据设备的显示能力，取得与指定颜色最接近的一种纯色
GetObjectAPI 取得对指定对象进行说明的一个结构
GetObjectType 判断由指定句柄引用的 GDI 对象的类型
GetPath 取得对当前路径进行定义的一系列数据
GetPixel 在指定的设备场景中取得一个像素的 RGB 值
GetPolyFillMode 针对指定的设备场景，获得多边形填充模式
第五页
绘图函数：共七页。第一页，第二页，第三页，第四页，第五页，第六页，第七页
GetROP2 针对指定的设备场景，取得当前的绘图模式
GetStockObject 取得一个固有对象（Stock）
GetSysColorBrush 为任何一种标准系统颜色取得一个刷子
GetWinMetaFileBits 通过在一个缓冲区中填充用于标准图元文件的数据，将一个增强型图元文件转换成标准 windows 图元文件
InvertRect 通过反转每个像素的值，从而反转一个设备场景中指定的矩形
LineDDA 枚举指定线段中的所有点
LineTo 用当前画笔画一条线，从当前位置连到一个指定的点
MoveToEx 为指定的设备场景指定一个新的当前画笔位置
PaintDesktop 在指定的设备场景中描绘桌面墙纸图案
PathToRegion 将当前选定的路径转换到一个区域里
Pie 画一个饼图
PlayEnhMetaFile 在指定的设备场景中画一个增强型图元文件
PlayEnhMetaFileRecord 回放单独一条增强型图元文件记录
PlayMetaFile 在指定的设备场景中回放一个图元文件
PlayMetaFileRecord 回放来自图元文件的单条记录
PolyBezier 描绘一条或多条贝塞尔（Bezier）曲线
第六页
绘图函数：共七页。第一页，第二页，第三页，第四页，第五页，第六页，第七页
PolyDraw 描绘一条复杂的曲线，由线段及贝塞尔曲线组成
Polygon 描绘一个多边形
Polyline 用当前画笔描绘一系列线段
PolyPolygon 用当前选定画笔描绘两个或多个多边形
PolyPolyline 用当前选定画笔描绘两个或多个多边形
Rectangle 用当前选定的画笔描绘矩形，并用当前选定的刷子填充
RoundRect 用当前选定的画笔画一个圆角矩形，并用当前选定的刷子在其中填充
SelectClipPath 将设备场景当前的路径合并到剪切区域里
SelectObject 为当前设备场景选择图形对象
SetArcDirection 设置圆弧的描绘方向
SetBkColor 为指定的设备场景设置背景颜色

SetBkMode 指定阴影刷子、虚线画笔以及字符中的空隙的填充方式
SetBrushOrgEx 为指定的设备场景设置当前选定刷子的起点
SetEnhMetaFileBits 用指定内存缓冲区内包含的数据创建一个增强型图元文件
SetMetaFileBitsEx 用包含在指定内存缓冲区内的数据结构创建一个图元文件
SetMiterLimit 设置设备场景当前的斜率限制
第七页
绘图函数：共七页。第一页，第二页，第三页，第四页，第五页，第六页，第七页
SetPixel 在指定的设备场景中设置一个像素的 RGB 值
SetPixelV 在指定的设备场景中设置一个像素的 RGB 值
SetPolyFillMode 设置多边形的填充模式
SetROP2 设置指定设备场景的绘图模式。与 vb 的 DrawMode 属性完全一致
SetWinMetaFileBits 将一个标准 Windows 图元文件转换成增强型图元文件
StrokeAndFillPath 针对指定的设备场景，关闭路径上打开的所有区域
StrokePath 用当前画笔描绘一个路径的轮廓。打开的图形不会被这个函数关闭
UnrealizeObject 将一个刷子对象选入设备场景之前，如刷子的起点准备用 SetBrushOrgEx 修改，则必须先调用本函数
WidenPath 根据选定画笔的宽度，重新定义当前选定的路径
完
打印函数
打印函数，共五页。第一页，第二页，第三页，第四页，第五页
AbortDoc 取消一份文档的打印
AbortPrinter 删除与一台打印机关联在一起的缓冲文件
AddForm 为打印机的表单列表添加一个新表单
AddJob 用于获取一个有效的路径名，以便用它为作业创建一个后台打印文件。它也会为作业分配一个作业编号
AddMonitor 为系统添加一个打印机监视器
AddPort 启动“添加端口”对话框，允许用户在系统可用端口列表中加入一个新端口
AddPrinter 在系统中添加一台新打印机
AddPrinterConnection 连接指定的打印机
AddPrinterDriver 为指定的系统添加一个打印驱动程序
AddPrintProcessor 为指定的系统添加一个打印处理器
AddPrintProvidor 为系统添加一个打印供应商
AdvancedDocumentProperties 启动打印机文档设置对话框
ClosePrinter 关闭一个打开的打印机对象
ConfigurePort 针对指定的端口，启动一个端口配置对话框
ConnectToPrinterDlg 启动连接打印机对话框，用它同访问网络的打印机连接
DeleteForm 从打印机可用表单列表中删除一个表单
第一页
打印函数，共五页。第一页，第二页，第三页，第四页，第五页
AbortDoc 取消一份文档的打印
AbortPrinter 删除与一台打印机关联在一起的缓冲文件
AddForm 为打印机的表单列表添加一个新表单
AddJob 用于获取一个有效的路径名，以便用它为作业创建一个后台打印文件。它也会为作业分配一个作业编号

AddMonitor 为系统添加一个打印机监视器
AddPort 启动“添加端口”对话框，允许用户在系统可用端口列表中加入一个新端口
AddPrinter 在系统中添加一台新打印机
AddPrinterConnection 连接指定的打印机
AddPrinterDriver 为指定的系统添加一个打印驱动程序
AddPrintProcessor 为指定的系统添加一个打印处理器
AddPrintProvidor 为系统添加一个打印供应商
AdvancedDocumentProperties 启动打印机文档设置对话框
ClosePrinter 关闭一个打开的打印机对象
ConfigurePort 针对指定的端口，启动一个端口配置对话框
ConnectToPrinterDlg 启动连接打印机对话框，用它同访问网络的打印机连接
DeleteForm 从打印机可用表单列表中删除一个表单

DeleteMonitor 删除指定的打印监视器
DeletePort 启动“删除端口”对话框，允许用户从当前系统删除一个端口
DeletePrinter 将指定的打印机标志为从系统中删除
DeletePrinterConnection 删除与指定打印机的连接
DeletePrinterDriver 从系统删除一个打印机驱动程序
DeletePrintProcessor 从指定系统删除一个打印处理器
DeletePrintProvidor 从系统中删除一个打印供应商
DeviceCapabilities 利用这个函数可获得与一个设备的能力有关的信息
DocumentProperties 打印机配置控制函数
EndDocAPI 结束一个成功的打印作业
EndDocPrinter 在后台打印程序的级别指定一个文档的结束
EndPage 用这个函数完成一个页面的打印，并准备设备场景，以便打印下一个页
EndPagePrinter 指定一个页在打印作业中的结尾
EnumForms 枚举一台打印机可用的表单
EnumJobs 枚举打印队列中的作业
EnumMonitors 枚举可用的打印监视器

EnumPorts 枚举一个系统可用的端口
EnumPrinterDrivers 枚举指定系统中已安装的打印机驱动程序
EnumPrinters 枚举系统中安装的打印机
EnumPrintProcessorDatatypes 枚举由一个打印处理器支持的数据类型
EnumPrintProcessors 枚举系统中可用的打印处理器
Escape 设备控制函数
FindClosePrinterChangeNotification 关闭用 FindFirstPrinterChangeNotification 函数获取的一个打印机通告对象
FindFirstPrinterChangeNotification 创建一个新的改变通告对象，以便我们注意打印机状态的各种变化
FindNextPrinterChangeNotification 用这个函数判断触发一次打印机改变通告信号的原因
FreePrinterNotifyInfo 释放由 FindNextPrinterChangeNotification 函数分配的一个缓冲区

GetForm 取得与指定表单有关的信息
GetJob 获取与指定作业有关的信息
GetPrinter 取得与指定打印机有关的信息
GetPrinterData 为打印机设置注册表配置信息
GetPrinterDriver 针对指定的打印机，获取与打印机驱动程序有关的信息
GetPrinterDriverDirectory 判断指定系统中包含了打印机驱动程序的目录是什么
第四页
打印函数，共五页。第一页，第二页，第三页，第四页，第五页
GetPrintProcessorDirectory 判断指定系统中包含了打印机处理器驱动程序及文件的目录
OpenPrinter 打开指定的打印机，并获取打印机的句柄
PrinterMessageBox 在拥有指定打印作业的系统上显示一个打印机出错消息框
PrinterProperties 启动打印机属性对话框，以便对打印机进行配置
ReadPrinter 从打印机读入数据
ResetDC 重设一个设备场景
ResetPrinter 改变指定打印机的默认数据类型及文档设置
ScheduleJob 提交一个要打印的作业
SetAbortProc 为 Windows 指定取消函数的地址
SetForm 为指定的表单设置信息
SetJob 对一个打印作业的状态进行控制
SetPrinter 对一台打印机的状态进行控制
SetPrinterData 设置打印机的注册表配置信息
StartDoc 开始一个打印作业
StartDocPrinter 在后台打印的级别启动一个新文档
StartPage 打印一个新页前要先调用这个函数
第五页
打印函数，共五页。第一页，第二页，第三页，第四页，第五页
StartPagePrinter 在打印作业中指定一个新页的开始
WritePrinter 将发送目录中的数据写入打印机
完
设备场景函数
设备场景函数，共五页。第一页，第二页，第三页，第四页，第五页
CombineRgn 将两个区域组合为一个新区域
CombineTransform 驱动世界转换。它相当于依顺序进行两次转换
CreateCompatibleDC 创建一个与特定设备场景一致的内存设备场景
CreateDC 为专门设备创建设备场景
CreateEllipticRgn 创建一个椭圆
CreateEllipticRgnIndirect 创建一个内切于特定矩形的椭圆区域
CreateIC 为专用设备创建一个信息场景
CreatePolygonRgn 创建一个由一系列点围成的区域
CreatePolyPolygonRgn 创建由多个多边形构成的区域。每个多边形都应是封闭的
CreateRectRgn 创建一个矩形区域
CreateRectRgnIndirect 创建一个矩形区域
CreateRoundRectRgn 创建一个圆角矩形
DeleteDC 删除专用设备场景或信息场景，释放所有相关窗口资源

DPtoLP 将点阵从设备坐标转换到专用设备场景逻辑坐标
EqualRgn 确定两个区域是否相等
ExcludeClipRect 从专用设备场景的剪裁区中去掉一个矩形区。矩形内不能进行绘图
第一页
设备场景函数，共五页。第一页，第二页，第三页，第四页，第五页
CombineRgn 将两个区域组合为一个新区域
CombineTransform 驱动世界转换。它相当于依顺序进行两次转换
CreateCompatibleDC 创建一个与特定设备场景一致的内存设备场景
CreateDC 为专门设备创建设备场景
CreateEllipticRgn 创建一个椭圆
CreateEllipticRgnIndirect 创建一个内切于特定矩形的椭圆区域
CreateIC 为专用设备创建一个信息场景
CreatePolygonRgn 创建一个由一系列点围成的区域
CreatePolyPolygonRgn 创建由多个多边形构成的区域。每个多边形都应是封闭的
CreateRectRgn 创建一个矩形区域
CreateRectRgnIndirect 创建一个矩形区域
CreateRoundRectRgn 创建一个圆角矩形
DeleteDC 删除专用设备场景或信息场景，释放所有相关窗口资源
DPtoLP 将点阵从设备坐标转换到专用设备场景逻辑坐标
EqualRgn 确定两个区域是否相等
ExcludeClipRect 从专用设备场景的剪裁区中去掉一个矩形区。矩形内不能进行绘图
第二页
设备场景函数，共五页。第一页，第二页，第三页，第四页，第五页
ExcludeUpdateRgn 从专用设备场景剪裁区去掉指定窗口的刷新区域
ExtCreateRegion 根据世界转换修改区域
ExtSelectClipRgn 将指定区域组合到设备场景的当前剪裁区
FillRgn 用指定刷子填充指定区域
FrameRgn 用指定刷子围绕指定区域画一个外框
GetBoundsRect 获取指定设备场景的边界矩形
GetClipBox 获取完全包含指定设备场景剪裁区的最小矩形
GetClipRgn 获取设备场景当前剪裁区
GetDC 获取指定窗口的设备场景
GetDCEx 为指定窗口获取设备场景。相比 GetDC，本函数提供了更多的选项
GetDCOrgEx 获取指定设备场景起点位置（以屏幕坐标表示）
GetDeviceCaps 根据指定设备场景代表的设备的功能返回信息
GetGraphicsMode 确定是否允许增强图形模式（世界转换）
GetMapMode 为特定设备场景调入映象模式
GetRegionData 装入描述一个区域信息的 RgnData 结构或缓冲区
GetRgnBox 获取完全包含指定区域的最小矩形
第三页
设备场景函数，共五页。第一页，第二页，第三页，第四页，第五页
GetUpdateRgn 确定指定窗口的刷新区域。该区域当前无效，需要刷新
GetViewportExtEx 获取设备场景视口（viewport）范围
GetViewportOrgEx 获取设备场景视口起点

GetWindowDC 获取整个窗口（包括边框、滚动条、标题栏、菜单等）的设备场景
GetWindowExtEx 获取指定设备场景的窗口范围
GetWindowOrgEx 获取指定设备场景的逻辑窗口的起点
GetWindowRgn 获取窗口区域
GetWorldTransform 如果有世界转换，为设备场景获取当前世界转换
IntersectClipRect 为指定设备定义一个新的剪裁区
InvalidateRgn 使窗口指定区域不活动，并将它加入窗口刷新区，使之可随后被重画
InvertRgn 通过颠倒每个像素值反转设备场景指定区域
LPtoDP 将点阵从指定设备场景逻辑坐标转换为设备坐标
ModifyWorldTransform 根据指定的模式修改世界转换
OffsetClipRgn 按指定量平移设备场景剪裁区
OffsetRgn 按指定偏移量平移指定区域
OffsetViewportOrgEx 平移设备场景视口区域
第四页
设备场景函数，共五页。第一页，第二页，第三页，第四页，第五页
OffsetWindowOrgEx 平移指定设备场景窗口起点
PaintRgn 用当前刷子背景色填充指定区域
PtInRegion 确定点是否在指定区域内
PtVisible 确定指定点是否可见（即，点是否在设备场景剪裁区内）
RectInRegion 确定矩形是否有部分在指定区域内
RectVisible 确定指定矩形是否有部分可见（是否在设备场景剪裁区内）
ReleaseDC 释放由调用 GetDC 或 GetWindowDC 函数获取的指定设备场景
RestoreDC 从设备场景堆栈恢复一个原先保存的设备场景
SaveDC 将指定设备场景状态保存到 Windows 设备场景堆栈
ScaleViewportExtEx 缩放设备场景视口的范围
ScaleWindowExtEx 缩放指定设备场景窗口范围
ScrollDC 在窗口（由设备场景代表）中水平和（或）垂直滚动矩形
SelectClipRgn 为指定设备场景选择新的剪裁区
SetBoundsRect 设置指定设备场景的边界矩形
SetGraphicsMode 允许或禁止增强图形模式，以提供某些支持（包括世界转换）
SetMapMode 设置指定设备场景的映射模式
第五页
设备场景函数，共五页。第一页，第二页，第三页，第四页，第五页
SetRectRgn 设置区域为指定的矩形
SetViewportExtEx 设置设备场景视口范围
SetViewportOrgEx 设置设备场景视口起点
SetWindowExtEx 设置指定设备场景窗口范围
SetWindowOrgEx 设置指定设备场景窗口起点
SetWindowRgn 设置窗口区域
SetWorldTransform 设置世界转换
ValidateRgn 激活窗口中指定区域，把它从刷新区移走
WindowFromDC 取回与某一设备场景相关的窗口的句柄
完
进程和线程函数

进程与线程函数，共五页。第一页，第二页，第三页，第四页，第五页
CancelWaitableTimer 这个函数用于取消一个可以等待下去的计时器操作
CallNamedPipe 这个函数由一个希望通过管道通信的一个客户进程调用
ConnectNamedPipe 指示一台服务器等待下去，直至客户机同一个命名管道连接
CreateEvent 创建一个事件对象
CreateMailslot 创建一个邮路。返回的句柄由邮路服务器使用（收件人）
CreateMutex 创建一个互斥体（MUTEX）
CreateNamedPipe 创建一个命名管道。返回的句柄由管道的服务器端使用
CreatePipe 创建一个匿名管道
CreateProcess 创建一个新进程（比如执行一个程序）
CreateSemaphore 创建一个新的信号机
CreateWaitableTimer 创建一个可等待的计时器对象
DisconnectNamedPipe 断开一个客户与一个命名管道的连接
DuplicateHandle 在指出一个现有系统对象当前句柄的情况下，为那个对象创建一个新句柄
ExitProcess 中止一个进程
FindCloseChangeNotification 关闭一个改动通知对象
FindExecutable 查找与一个指定文件关联在一起的程序的文件名
第一页
进程与线程函数，共五页。第一页，第二页，第三页，第四页，第五页
CancelWaitableTimer 这个函数用于取消一个可以等待下去的计时器操作
CallNamedPipe 这个函数由一个希望通过管道通信的一个客户进程调用
ConnectNamedPipe 指示一台服务器等待下去，直至客户机同一个命名管道连接
CreateEvent 创建一个事件对象
CreateMailslot 创建一个邮路。返回的句柄由邮路服务器使用（收件人）
CreateMutex 创建一个互斥体（MUTEX）
CreateNamedPipe 创建一个命名管道。返回的句柄由管道的服务器端使用
CreatePipe 创建一个匿名管道
CreateProcess 创建一个新进程（比如执行一个程序）
CreateSemaphore 创建一个新的信号机
CreateWaitableTimer 创建一个可等待的计时器对象
DisconnectNamedPipe 断开一个客户与一个命名管道的连接
DuplicateHandle 在指出一个现有系统对象当前句柄的情况下，为那个对象创建一个新句柄
ExitProcess 中止一个进程
FindCloseChangeNotification 关闭一个改动通知对象
FindExecutable 查找与一个指定文件关联在一起的程序的文件名
第二页
进程与线程函数，共五页。第一页，第二页，第三页，第四页，第五页
FindFirstChangeNotification 创建一个文件通知对象。该对象用于监视文件系统发生的变化
FindNextChangeNotification 重设一个文件改变通知对象，令其继续监视下一次变化
FreeLibrary 释放指定的动态链接库
GetCurrentProcess 获取当前进程的一个伪句柄
GetCurrentProcessId 获取当前进程一个唯一的标识符
GetCurrentThread 获取当前线程的一个伪句柄
GetCurrentThreadId 获取当前线程一个唯一的线程标识符

GetExitCodeProces 获取一个已中断进程的退出代码
GetExitCodeThread 获取一个已中止线程的退出代码
GetHandleInformation 获取与一个系统对象句柄有关的信息
GetMailslotInfo 获取与一个邮路有关的信息
GetModuleFileName 获取一个已装载模板的完整路径名称
GetModuleHandle 获取一个应用程序或动态链接库的模块句柄
GetPriorityClass 获取特定进程的优先级别
GetProcessShutdownParameters 调查系统关闭时一个指定的进程相对于其它进程的关闭早迟情况
GetProcessTimes 获取与一个进程的经过时间有关的信息
第三页
进程与线程函数，共五页。第一页，第二页，第三页，第四页，第五页
GetProcessWorkingSetSize 了解一个应用程序在运行过程中实际向它交付了多大容量的内存
GetSartupInfo 获取一个进程的启动信息
GetThreadPriority 获取特定线程的优先级别
GetTheardTimes 获取与一个线程的经过时间有关的信息
GetWindowThreadProcessId 获取与指定窗口关联在一起的一个进程和线程标识符
LoadLibrary 载入指定的动态链接库，并将它映射到当前进程使用的地址空间
LoadLibraryEx 装载指定的动态链接库，并为当前进程把它映射到地址空间
LoadModule 载入一个 Windows 应用程序，并在指定的环境中运行
MsgWaitForMultipleObjects 等侯单个对象或一系列对象发出信号。如返回条件已经满足，则立即返回
SetPriorityClass 设置一个进程的优先级别
SetProcessShutdownParameters 在系统关闭期间，为指定进程设置他相对于其它程序的关闭顺序
SetProcessWorkingSetSize 设置操作系统实际划分给进程使用的内存容量
SetThreadPriority 设定线程的优先级别
ShellExecute 查找与指定文件关联在一起的程序的文件名
TerminateProcess 结束一个进程
WinExec 运行指定的程序
第四页
进程与线程函数，共五页。第一页，第二页，第三页，第四页，第五页
第五页
进程与线程函数，共五页。第一页，第二页，第三页，第四页，第五页
Windows 消息函数
Windows 消息函数，共一页。第一页
BroadcastSystemMessage 将一条系统消息广播给系统中所有的顶级窗口
GetMessagePos 取得消息队列中上一条消息处理完毕时的鼠标指针屏幕位置
GetMessageTime 取得消息队列中上一条消息处理完毕时的时间
PostMessage 将一条消息投递到指定窗口的消息队列
PostThreadMessage 将一条消息投递给应用程序
RegisterWindowMessage 获取分配给一个字串标识符的消息编号
ReplyMessage 答复一个消息
SendMessage 调用一个窗口的窗口函数，将一条消息发给那个窗口

SendMessageCallback 将一条消息发给窗口
SendMessageTimeout 向窗口发送一条消息
SendNotifyMessage 向窗口发送一条消息
完
BroadcastSystemMessage
BroadcastSystemMessage

VB 声明
Declare Function BroadcastSystemMessage Lib "user32" Alias "BroadcastSystemMessage" (ByVal dw As Long, pdw As Long, ByVal un As Long, ByVal wParam As Long, ByVal lParam As Long) As Long
说明
将一条系统消息广播给系统中所有的顶级窗口
返回值
Long，大于零表示成功；-1 表示出错。如设置了 BSF_QUERY，而且至少有一个消息接收者返回零，那么这个函数返回零
参数表
参数类型及说明
dwLong，下述常数的一个或多个
BSF_FLUSHDISK 每次处理完一条消息后，都对磁盘进行刷新（将未存盘的数据存下来
BSF_FORCEIFHUNG 如目标处于挂起状态，则在设定的超时后到期返回
BSF_IGNORECURRENTTASK 发送任务不接收消息
BSF_LPARAMBUFFERlParam 指向一个内存缓冲区
BSF_NOHANG 跳过被挂起的所有进程
BSF_POSTMESSAGE 投递消息。不与 BSF_LPARAMBUFFER 和 BSF_QUERY 兼容
BSF_QUERY 将消息顺序发给进程，只有前一个返回 TRUE 时，才进入下一个进程
pdwLong，下述常数的一个或多个
BSF_ALLCOMPONENTS 消息进入能够接收消息的每一个系统组件
BSF_APPLICATIONS 消息到达应用程序
BSF_INSTALLABLEDRIVERS 消息到达可安装的驱动程序
BSF_NETDRIVERS 消息到达网络驱动程序
BSF_VXDS 消息到达系统设备驱动程序
unLong，消息编号
wParamLong，由消息决定
lParamLong，由消息决定。如指定了 BSF_LPARAMBUFFER，这就是位于调用进程地址空间的一个内存缓冲区的地址，而且缓冲区的第一个 16 位字包含了缓冲区的长度

Top
GetMessagePos
GetMessagePos

VB 声明
Declare Function GetMessagePos Lib "user32" Alias "GetMessagePos" () As Long
说明

取得消息队列中上一条消息处理完毕时的鼠标指针屏幕位置
返回值
Long，X 坐标对应于结果值的低字，Y 坐标对应于高字

Top
GetMessageTime
GetMessageTime

VB 声明
Declare Function GetMessageTime Lib "user32" Alias "GetMessageTime" () As Long
说明
取得消息队列中上一条消息处理完毕时的时间
返回值
Long，返回一个时间，表示为自系统启动以来经历的毫秒数
原文：The time is specified in milliseconds from the time the system was started.

Top
PostMessage
PostMessage, PostMessageBynum, PostMessageBystring

VB 声明
Declare Function PostMessage& Lib "user32" Alias "PostMessageA" (ByVal hwnd As Long, ByVal wMsg As Long, ByVal wParam As Long, lParam As Any)
Declare Function PostMessageByNum& Lib "user32" Alias "PostMessageA" (ByVal hwnd As Long, ByVal wMsg As Long, ByVal wParam As Long, ByVal lParam As Long)
Declare Function PostMessageByString& Lib "user32" Alias "PostMessageA" (ByVal hwnd As Long, ByVal wMsg As Long, ByVal wParam As Long, ByVal lParam As String)
说明
将一条消息投递到指定窗口的消息队列。投递的消息会在 Windows 事件处理过程中得到处理。在那个时候，会随同投递的消息调用指定窗口的窗口函数。特别适合那些不需要立即处理的窗口消息的发送
返回值
Long，如消息投递成功，则返回 TRUE（非零）。会设置 GetLastError
参数表
参数类型及说明
hwndLong，接收消息的那个窗口的句柄。如设为 HWND_BROADCAST，表示投递给系统中的所有顶级窗口。如设为零，表示投递一条线程消息（参考 PostThreadMessage）
wMsgLong，消息标识符
wParamLong，具体由消息决定
lParamAny，具体由消息决定

Top
PostThreadMessage
PostThreadMessage

VB 声明
Declare Function PostThreadMessage Lib "user32" Alias "PostThreadMessageA" (ByVal idThread As Long, ByVal msg As Long, ByVal wParam As Long, ByVal lParam As Long) As Long
说明
将一条消息投递给应用程序。这条消息由应用程序的内部 GetMessage 循环获得，但不会传给一个特定的窗口
返回值
Long，如消息投递成功，则返回 TRUE（非零）。会设置 GetLastError
参数表
参数类型及说明
idThreadLong，用于接收消息的那个线程的标识符
msgLong，消息标识符
wParamLong，具体由消息决定
ByValLong，具体由消息决定

Top
RegisterWindowMessage
RegisterWindowMessage

VB 声明
Declare Function RegisterWindowMessage Lib "user32" Alias "RegisterWindowMessageA" (ByVal lpString As String) As Long
说明
获取分配给一个字串标识符的消息编号
返回值
Long，&C000 到 &FFFF 之间的一个消息编号。零意味着出错
参数表
参数类型及说明
lpStringString，注册消息的名字
注解
如果没有一个子类处理程序的帮助，这个函数就没有什么用

Top
ReplyMessage
ReplyMessage

VB 声明
Declare Function ReplyMessage Lib "user32" Alias "ReplyMessage" (ByVal lReply As Long) As Long
说明
如将消息传送给位于不同进程的一个窗口，通常第一个进程会暂时挂起，直到另一个进程中的窗口函数完成操作为止。在目标进程的窗口函数完成之前，另一个进程可用这个函数向第

一个进程返回一个结果，使之能继续进行
返回值
Long，如准备答复的消息是由另一个进程发来的，则返回 TRUE。如果它是从同一个进程中发出来的，则返回 FALSE（此时，该函数没有任何效果）
参数表
参数类型及说明
lReplyLong，指定发回调用进程的一个结果

Top
SendMessage
SendMessage, SendMessageBynum, SendMessageByString

VB 声明
Declare Function SendMessage& Lib "user32" Alias "SendMessageA" (ByVal hwnd As Long, ByVal wMsg As Long, ByVal wParam As Long, lParam As Any)
Declare Function SendMessageBynum& Lib "user32" Alias "SendMessageA" (ByVal hwnd As Long, ByVal wMsg As Long, ByVal wParam As Long, ByVal lParam As Long)
Declare Function SendMessageByString& Lib "user32" Alias "SendMessageA" (ByVal hwnd As Long, ByVal wMsg As Long, ByVal wParam As Long, ByVal lParam As String)
说明
调用一个窗口的窗口函数，将一条消息发给那个窗口。除非消息处理完毕，否则该函数不会返回。SendMessageBynum， SendMessageByString 是该函数的“类型安全”声明形式
返回值
Long，由具体的消息决定
参数表
参数类型及说明
hwndLong，要接收消息的那个窗口的句柄
wMsgLong，消息的标识符
wParamLong，具体取决于消息
lParamAny，具体取决于消息

Top
SendMessageCallback
SendMessageCallback

VB 声明
Declare Function SendMessageCallback Lib "user32" Alias "SendMessageCallbackA" (ByVal hwnd As Long, ByVal msg As Long, ByVal wParam As Long, ByVal lParam As Long, ByVal lpResultCallBack As Long, ByVal dwData As Long) As Long
说明
将一条消息发给窗口。该函数最大的特定是可以立即返回。目标窗口函数执行完毕后，会用回调函数的形式将结果返回
返回值
Long，TRUE 表示成功，FALSE 表示失败。会设置 GetLastError

参数表
参数类型及说明
hwndLong，要接收消息的那个窗口的句柄
msgLong，消息的标识符
wParamLong，取决于消息
lParamLong，取决于消息
lpResultCallBackLong，指定函数地址。在 vb5 中可用 AddressOf 操作符获得
dwDataLong，用户自定义值
注解
回调函数声明如下：
Public Function WndProc(ByVal hwnd&, ByVal msg&, ByVal wp&, ByVal lp&) As Long
其中，wp 参数是作为 dwData 参数传递的值。lp 参数包含了来自窗口函数的结果

Top
SendMessageTimeout
SendMessageTimeout

VB 声明
Declare Function SendMessageTimeout Lib "user32" Alias "SendMessageTimeoutA" (ByVal hwnd As Long, ByVal msg As Long, ByVal wParam As Long, ByVal lParam As Long, ByVal fuFlags As Long, ByVal uTimeout As Long, lpdwResult As Long) As Long
说明
向窗口发送一条消息。如窗口位于不同的线程中，则利用这个函数可以指定一个超时值，以便在另一个进程挂起的时候防止调用进程也永远挂起
返回值
Long，成功时返回 TRUE，失败时返回 FALSE。会设置 GetLastError
参数表
参数类型及说明
hwndLong，要接收消息的一个窗口的句柄
msgLong，消息的标识符
wParamLong，由消息决定
lParamLong，由消息决定
fuFlagsLong，下述常数的一个或多个
SMTO_ABORTIFHUNG 如目标进程挂起，则函数立即返回
SMTO_BLOCK 除非函数返回，否则调用线程不能处理消息
SMTO_NORMAL 允许调用线程处理消息，同时保持函数继续执行
uTimeoutLong，超时值，采用毫秒为单位
lpdwResultLong，用于装载函数结果的一个变量

Top
SendNotifyMessage
SendNotifyMessage

VB 声明

Declare Function SendNotifyMessage Lib "user32" Alias "SendNotifyMessageA" (ByVal hwnd As Long, ByVal msg As Long, ByVal wParam As Long, ByVal lParam As Long) As Long
说明
向窗口发送一条消息。如目标窗口位于同调用方相同的线程内，则这个函数会表现为 SendMessage 函数。而且除非消息得到处理，否则函数不会返回。如目标窗口从属于一个不同的线程，则函数会立即返回
返回值
Long，TRUE 表示成功，FALSE 表示出错。会设置 GetLastError
参数表
参数类型及说明
hwndLong，用于接收消息的一个窗口的句柄
msgLong，消息的标识符
wParamLong，具体由消息决定
lParamLong，具体由消息决定

Top
文件处理函数
文件处理函数，共八页。第一页，第二页，第三页，第四页，第五页，第六页，第七页，第八页
CloseHandle 关闭一个内核对象。其中包括文件、文件映射、进程、线程、安全和同步对象等
CompareFileTime 对比两个文件的时间
CopyFile 复制文件
CreateDirectory 创建一个新目录
CreateFile 打开和创建文件、管道、邮槽、通信服务、设备以及控制台
CreateFileMapping 创建一个新的文件映射对象
DeleteFile 删除指定文件
DeviceIoControl 对设备执行指定的操作
DosDateTimeToFileTime 将 DOS 日期和时间值转换成一个 win32 FILETIME 值
FileTimeToDosDateTime 将一个 win32 FILETIME 值转换成 DOS 日期和时间值
FileTimeToLocalFileTime 将一个 FILETIME 结构转换成本地时间
FileTimeToSystemTime 根据一个 FILETIME 结构的内容，装载一个 SYSTEMTIME 结构
FindClose 关闭由 FindFirstFile 函数创建的一个搜索句柄
FindFirstFile 根据文件名查找文件
FindNextFile 根据调用 FindFirstFile 函数时指定的一个文件名查找下一个文件
FlushFileBuffers 针对指定的文件句柄，刷新内部文件缓冲区
CloseHandle
CloseHandle

VB 声明
Declare Function CloseHandle Lib "kernel32" Alias "CloseHandle" (ByVal hObject As Long) As Long
说明
关闭一个内核对象。其中包括文件、文件映射、进程、线程、安全和同步对象等。涉及文件

处理时，这个函数通常与 vb 的 close 命令相似。应尽可能的使用 close，因为它支持 vb 的差错控制。注意这个函数使用的文件句柄与 vb 的文件编号是完全不同的
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hObjectLong，欲关闭的一个对象的句柄
注解
除非对内核对象的所有引用都已关闭，否则该对象不会实际删除

Top
CompareFileTime
CompareFileTime

VB 声明
Declare Function CompareFileTime Lib "kernel32" Alias "CompareFileTime" (lpFileTime1 As FILETIME, lpFileTime2 As FILETIME) As Long
说明
根据 FILETIME 结构的信息，对比两个文件的时间
返回值
Long，如两个时间相等，就返回零；如 lpFileTime1 小于 lpFileTime2，返回-1；如 lpFileTime2 小于 lpFileTime1，返回 1
参数表
参数类型及说明
lpFileTime1FILETIME，参考 FILETIME
lpFileTime2

Top
CopyFile
CopyFile

VB 声明
Declare Function CopyFile Lib "kernel32" Alias "CopyFileA" (ByVal lpExistingFileName As String, ByVal lpNewFileName As String, ByVal bFailIfExists As Long) As Long
说明
复制文件。与 vb 的 filecopy 命令相似
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpExistingFileNameString，源文件名
lpNewFileNameString，目标文件名
bFailIfExistsLong，如果设为 TRUE（非零），那么一旦目标文件已经存在，则函数调用会失败。否则目标文件被改写

Top
CreateDirectory
CreateDirectory, CreateDirectoryEx

VB 声明
Declare Function CreateDirectory& Lib "kernel32" Alias "CreateDirectoryA" (ByVal lpNewDirectory As String, lpSecurityAttributes As SECURITY_ATTRIBUTES)
Declare Function CreateDirectoryEx& Lib "kernel32" Alias "CreateDirectoryExA" (ByVal lpTemplateDirectory As String, ByVal lpNewDirectory As String, lpSecurityAttributes As SECURITY_ATTRIBUTES)
说明
创建一个新目录
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpTemplateDirectoryString，指定一个模板目录的名字，从中复制默认属性（比如目录中文件的默认压缩方式）。如设为 vbNullString，则表示不使用模板
lpNewDirectoryString，新目录的名字
lpSecurityAttributesSECURITY_ATTRIBUTES，这个结构定义了目录的安全特性——如果操作系统支持的话

Top
CreateFile
CreateFile

VB 声明
Declare Function CreateFile Lib "kernel32" Alias "CreateFileA" (ByVal lpFileName As String, ByVal dwDesiredAccess As Long, ByVal dwShareMode As Long, lpSecurityAttributes As SECURITY_ATTRIBUTES, ByVal dwCreationDisposition As Long, ByVal dwFlagsAndAttributes As Long, ByVal hTemplateFile As Long) As Long
说明
这是一个全功能的例程，可打开和创建文件、管道、邮槽、通信服务、设备以及控制台
返回值
Long，如执行成功，则返回文件句柄。INVALID_HANDLE_VALUE 表示出错，会设置 GetLastError。即使函数成功，但若文件存在，且指定了 CREATE_ALWAYS 或 OPEN_ALWAYS，GetLastError 也会设为 ERROR_ALREADY_EXISTS
参数表
参数类型及说明
lpFileNameString，要打开的文件的名字
dwDesiredAccessLong，如果为 GENERIC_READ 表示允许对设备进行读访问；如果为 GENERIC_WRITE 表示允许对设备进行写访问（可组合使用）；如果为零，表示只允许获取与一个设备有关的信息

dwShareModeLong，零表示不共享； FILE_SHARE_READ 和/或 FILE_SHARE_WRITE 表示允许对文件进行共享访问
lpSecurityAttributesSECURITY_ATTRIBUTES，指向一个 SECURITY_ATTRIBUTES 结构的指针，定义了文件的安全特性（如果操作系统支持的话）
dwCreationDispositionLong，下述常数之一：
CREATE_NEW 创建文件；如文件存在则会出错
CREATE_ALWAYS 创建文件，会改写前一个文件
OPEN_EXISTING 文件必须已经存在。由设备提出要求
OPEN_ALWAYS 如文件不存在则创建它
TRUNCATE_EXISTING 讲现有文件缩短为零长度
dwFlagsAndAttributesLong，一个或多个下述常数
FILE_ATTRIBUTE_ARCHIVE 标记归档属性
FILE_ATTRIBUTE_COMPRESSED 将文件标记为已压缩，或者标记为文件在目录中的默认压缩方式
FILE_ATTRIBUTE_NORMAL 默认属性
FILE_ATTRIBUTE_HIDDEN 隐藏文件或目录
FILE_ATTRIBUTE_READONLY 文件为只读
FILE_ATTRIBUTE_SYSTEM 文件为系统文件
FILE_FLAG_WRITE_THROUGH 操作系统不得推迟对文件的写操作
FILE_FLAG_OVERLAPPED 允许对文件进行重叠操作
FILE_FLAG_NO_BUFFERING 禁止对文件进行缓冲处理。文件只能写入磁盘卷的扇区块
FILE_FLAG_RANDOM_ACCESS 针对随机访问对文件缓冲进行优化
FILE_FLAG_SEQUENTIAL_SCAN 针对连续访问对文件缓冲进行优化
FILE_FLAG_DELETE_ON_CLOSE 关闭了上一次打开的句柄后，将文件删除。特别适合临时文件
也可在 Windows NT 下组合使用下述常数标记：
SECURITY_ANONYMOUS ， SECURITY_IDENTIFICATION ， SECURITY_IMPERSONATION ， SECURITY_DELEGATION ， SECURITY_CONTEXT_TRACKING， SECURITY_EFFECTIVE_ONLY
hTemplateFileLong，如果不为零，则指定一个文件句柄。新文件将从这个文件中复制扩展属性
注解
打开一个通信端口时（如 COM1），无论如何都要设置成 OPEN_EXISTING
这个函数代替了 lOpen 和 lCreate 函数，应该是我们的首选

Top
CreateFileMapping
CreateFileMapping

VB 声明
Declare Function CreateFileMapping Lib "kernel32" Alias "CreateFileMappingA" (ByVal hFile As Long, lpFileMappigAttributes As SECURITY_ATTRIBUTES, ByVal flProtect As Long, ByVal dwMaximumSizeHigh As Long, ByVal dwMaximumSizeLow As Long, ByVal lpName As String) As Long

说明
创建一个新的文件映射对象
返回值
Long，新建文件映射对象的句柄；零意味着出错。会设置 GetLastError。即使函数成功，但倘若返回的句柄属于一个现成的文件映射对象，那么 GetLastError 也会设置成 ERROR_ALREADY_EXISTS。在这种情况下，文件映射的长度就是现有对象的长度，而不是这个函数指定的尺寸
参数表
参数类型及说明
hFileLong，指定欲在其中创建映射的一个文件句柄。&HFFFFFFFF&表示在内存中创建一个文件映射
lpFileMappigAttributesSECURITY_ATTRIBUTES，指定一个安全对象，在创建文件映射时使用。如果为 NULL（用 ByVal As Long 传递零），表示使用默认安全对象
flProtectLong，下述常数之一：
PAGE_READONLY 以只读方式打开映射
PAGE_READWRITE 以可读、可写方式打开映射
PAGE_WRITECOPY 为写操作留下备份
可组合使用下述一个或多个常数
SEC_COMMIT 为文件映射一个小节中的所有页分配内存
SEC_IMAGE 文件是个可执行文件
SEC_RESERVE 为没有分配实际内存的一个小节保留虚拟内存空间
dwMaximumSizeHighLong，文件映射的最大长度（高 32 位）
dwMaximumSizeLowLong，文件映射的最小长度（低 32 位）。如这个参数和 dwMaximumSizeHigh 都是零，就用磁盘文件的实际长度
lpNameString，指定文件映射对象的名字。如存在这个名字的一个映射，函数就会打开它。用 vbNullString 创建一个无名的文件映射

Top
DeleteFile
DeleteFile

VB 声明
Declare Function DeleteFile Lib "kernel32" Alias "DeleteFileA" (ByVal lpFileName As String) As Long
说明
删除指定文件
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpFileNameString，欲删除文件的名字
注解
与 vb 的 kill 语句相似，在 windows 95 下使用这个函数要小心——即使文件当前正由一个应用程序打开，该函数也会将其删除

Top
DeviceIoControl
DeviceIoControl

VB 声明
Declare Function DeviceIoControl Lib "kernel32" Alias "DeviceIoControl" (ByVal hDevice As Long, ByVal dwIoControlCode As Long, lpInBuffer As Any, ByVal nInBufferSize As Long, lpOutBuffer As Any, ByVal nOutBufferSize As Long, lpBytesReturned As Long, lpOverlapped As OVERLAPPED) As Long
说明
对设备执行指定的操作
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hDeviceLong，设备句柄
dwIoControlCodeLong，带有 FSCTL_ 前缀的常数。参考设备控制选项的部分列表
lpInBufferAny，具体取决于 dwIoControlCode 参数。参考设备控制选项的部分列表
nInBufferSizeLong，输入缓冲区的长度
lpOutBufferAny，具体取决于 dwIoControlCode 参数。参考设备控制选项的部分列表
nOutBufferSizeLong，输出缓冲区的长度
lpBytesReturnedLong，实际装载到输出缓冲区的字节数量
lpOverlappedOVERLAPPED，这个结构用于重叠操作。针对同步操作，请用 ByVal As Long 传递零值
注解
可用于 windows 95 和 windows nt，但并非所有的操作都得到了两种操作系统的同时支持

Top
DosDateTimeToFileTime
DosDateTimeToFileTime

VB 声明
Declare Function DosDateTimeToFileTime Lib "kernel32" Alias "DosDateTimeToFileTime" (ByVal wFatDate As Long, ByVal wFatTime As Long, lpFileTime As FILETIME) As Long
说明
将 DOS 日期和时间值转换成一个 win32 FILETIME 值
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
wFatDateLong，16 位 DOS 日期值
wFatTimeLong，16 位 DOS 时间值
lpFileTimeFILETIME，用于装载 win32 时间的一个结构

Top
FileTimeToDosDateTime
FileTimeToDosDateTime

VB 声明
Declare Function FileTimeToDosDateTime Lib "kernel32" Alias "FileTimeToDosDateTime" (lpFileTime As FILETIME, ByVal lpFatDate As Long, ByVal lpFatTime As Long) As Long
说明
将一个 win32 FILETIME 值转换成 DOS 日期和时间值
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpFileTimeFILETIME，包含了欲转换时间的一个结构
lpFatDateLong，16 位 DOS 日期值
lpFatTimeLong，16 位 DOS 时间值

Top
FileTimeToLocalFileTime
FileTimeToLocalFileTime

VB 声明
Declare Function FileTimeToLocalFileTime Lib "kernel32" Alias "FileTimeToLocalFileTime" (lpFileTime As FILETIME, lpLocalFileTime As FILETIME) As Long
说明
将一个 FILETIME 结构转换成本地时间
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpFileTimeFILETIME，包含了 UTC 时间信息的一个结构
lpLocalFileTimeFILETIME，用于装载转换过后的本地时间的结构

Top
FileTimeToSystemTime
FileTimeToSystemTime

VB 声明
Declare Function FileTimeToSystemTime Lib "kernel32" Alias "FileTimeToSystemTime" (lpFileTime As FILETIME, lpSystemTime As SYSTEMTIME) As Long
说明
根据一个 FILETIME 结构的内容，装载一个 SYSTEMTIME 结构
返回值

Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpFileTimeFILETIME，包含了文件时间的一个结构
lpSystemTimeSYSTEMTIME，用于装载系统时间信息的一个结构

Top
FindClose
FindClose

VB 声明
Declare Function FindClose Lib "kernel32" Alias "FindClose" (ByVal hFindFile As Long) As Long
说明
关闭由 FindFirstFile 函数创建的一个搜索句柄
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hFindFileLong，由 FindFirstFile 函数提供的搜索句柄

Top
FindFirstFile
FindFirstFile

VB 声明
Declare Function FindFirstFile Lib "kernel32" Alias "FindFirstFileA" (ByVal lpFileName As String, lpFindFileData As WIN32_FIND_DATA) As Long
说明
根据文件名查找文件
返回值
Long，如执行成功，返回一个搜索句柄。如果出错，返回一个 INVALID_HANDLE_VALUE 常数，一旦不再需要，应该用 FindClose 函数关闭这个句柄
参数表
参数类型及说明
lpFileNameString，欲搜索的文件名。可包含通配符，并可包含一个路径或相对路径名
lpFindFileDataWIN32_FIND_DATA，这个结构用于装载与找到的文件有关的信息。该结构可用于后续的搜索
注解
由这个函数返回的句柄可以作为一个参数用于 FindNextFile 函数。这样一来，就可以方便的枚举出与 lpFileName 参数指定的文件名相符的所有文件

Top
FindNextFile

FindNextFile

VB 声明
Declare Function FindNextFile Lib "kernel32" Alias "FindNextFileA" (ByVal hFindFile As Long, lpFindFileData As WIN32_FIND_DATA) As Long
说明
根据调用 FindFirstFile 函数时指定的一个文件名查找下一个文件
返回值
Long，非零表示成功，零表示失败。如不再有与指定条件相符的文件，会将 GetLastError 设置成 ERROR_NO_MORE_FILES
参数表
参数类型及说明
hFindFileLong，由 FindFirstFile 函数返回的搜索句柄
lpFindFileDataWIN32_FIND_DATA，这个结构用于装载与找到的文件有关的信息

Top
FlushFileBuffers
FlushFileBuffers

VB 声明
Declare Function FlushFileBuffers Lib "kernel32" Alias "FlushFileBuffers" (ByVal hFile As Long) As Long
说明
针对指定的文件句柄，刷新内部文件缓冲区
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hFileLong，文件句柄

Top
FlushViewOfFile
FlushViewOfFile

VB 声明
Declare Function FlushViewOfFile Lib "kernel32" Alias "FlushViewOfFile" (lpBaseAddress As Any, ByVal dwNumberOfBytesToFlush As Long) As Long
说明
将写入文件映射缓冲区的所有数据都刷新到磁盘
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpBaseAddressAny，包含了刷新基本地址的一个 Long 值（参考注解）

dwNumberOfBytesToFlushLong，欲刷新的字节数
注解
如与远程系统建立了文件映射，那么虽然这个函数可保证数据已在当前系统写入，但不能保证数据实际写入远程系统的磁盘——除非用 FILE_FLAG_WRITE_THROUGH 选项打开文件。该选项的作用是禁止写延迟，所有更新的数据都必须立即写入磁盘
这个函数的另一种声明形式：Declare Function FlushViewOfFile& Lib "kernel32" (ByVal lpBaseAddress As Long, ByVal dwNumberOfBytesToFlush As Long)

Top
GetBinaryType
GetBinaryType

VB 声明
Declare Function GetBinaryType Lib "kernel32" Alias "GetBinaryTypeA" (ByVal lpApplicationName As String, lpBinaryType As Long) As Long
说明
判断文件是否可以执行
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
lpApplicationNameString，欲测试文件的完整路径名
lpBinaryTypeLong，用于装载文件类型的一个变量。这些类型由下述任何一个常数定义：
SCS_32BIT_BINARYwin32 执行程序
SCS_DOS_BINARYDOS 执行程序
SCS_OS216_BINARY16 位 OS/2 执行程序
SCS_PIF_BINARY 用于执行 DOS 程序的一个 pif 文件
SCS_POSIX_BINARY 一个 Posix 应用
SCS_WOW_BINARY16 位 windows 执行程序

Top
GetCompressedFileSize
GetCompressedFileSize

VB 声明
Declare Function GetCompressedFileSize Lib "kernel32" Alias "GetCompressedFileSizeA" (ByVal lpFileName As String, lpFileSizeHigh As Long) As Long
说明
判断一个压缩文件在磁盘上实际占据的字节数
返回值
Long，返回文件长度。&HFFFFFFFF 表示出错。注意如 lpFileSizeHigh 不为 NULL，且结果为&HFFFFFFFF，那么必须调用 GetLastError，判断是否实际发生了一个错误，因为这是一个有效的结果
参数表

参数类型及说明
lpFileNameString，欲测试的文件名
lpFileSizeHighLong，指定一个 Long 值，用于装载一个 64 位文件尺寸的高 32 位。如长度没有超过 2 ^ 32 字节，则可设为 NULL（变成 ByVal）
注解
如磁盘卷已被压缩，可检查这个函数的结果是否与 GetFileSize 函数的结果有异，从而判断文件是否也被压缩（如有异，表明文件已被压缩）

Top
GetCurrentDirectory
GetCurrentDirectory

VB 声明
Declare Function GetCurrentDirectory Lib "kernel32" Alias "GetCurrentDirectory" (ByVal nBufferLength As Long, ByVal lpBuffer As String) As Long
说明
在一个缓冲区中装载当前目录
返回值
Long，装载到 lpBuffer 的字节数。如 nBufferLength 的长度不够，不足以容纳目录，则返回值是必要的缓冲区长度（要求至少这个长度），其中包括空中止字符。零表示失败。会设置 GetLastError
参数表
参数类型及说明
nBufferLengthLong，lpBuffer 缓冲区的长度
lpBufferString，指定一个预定义字串，用于装载当前目录

Top
GetDiskFreeSpace
GetDiskFreeSpace

VB 声明
Declare Function GetDiskFreeSpace Lib "kernel32" Alias "GetDiskFreeSpaceA" (ByVal lpRootPathName As String, lpSectorsPerCluster As Long, lpBytesPerSector As Long, lpNumberOfFreeClusters As Long, lpTtoalNumberOfClusters As Long) As Long
说明
获取与一个磁盘的组织有关的信息，以及了解剩余空间的容量
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpRootPathNameString，不包括卷名的一个磁盘根路径
lpSectorsPerClusterLong，用于装载一个簇内扇区数的变量
lpBytesPerSectorLong，用于装载一个扇区内字节数的变量
lpNumberOfFreeClustersLong，用于装载磁盘上剩余簇数的变量

lpTtoalNumberOfClustersLong，用于装载磁盘上总簇数的变量
注解
在采用 FAT16 格式的 windows95 系统中，如一个驱动器（分区）的容量超过了 2GB，则不应使用这个函数。此时，这个函数能识别的最大分区容量只有 2GB

Top
GetDiskFreeSpaceEx
GetDiskFreeSpaceEx

VB 声明
Declare Function GetDiskFreeSpaceEx Lib "kernel32" Alias "GetDiskFreeSpaceExA" (ByVal lpRootPathName As String, lpFreeBytesAvailableToCaller As LARGE_INTEGER, lpTotalNumberOfBytes As LARGE_INTEGER, lpTotalNumberOfFreeBytes As LARGE_INTEGER) As Long
说明
获取与一个磁盘的组织以及剩余空间容量有关的信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpRootPathNameString，不包括卷名的磁盘根路径名
lpFreeBytesAvailableToCallerLARGE_INTEGER，指定一个变量，用于容纳调用者可用的字节数量
lpTotalNumberOfBytesLARGE_INTEGER，指定一个变量，用于容纳磁盘上的总字节数
lpTotalNumberOfFreeBytesLARGE_INTEGER，指定一个变量，用于容纳磁盘上可用的字节数
适用平台
Windows 95 OSR2，Windows NT 4.0
注解
LARGE_INTEGER 结构与 FILETIME 结构在内部完全一致。正式调用前，用 GetVersionEx 判断函数是否得到了支持。在 Windows 95 OSR2 环境中，OSVERSIONINFO 结构的 dwBuildNumbe 字段会大于 1000

Top
GetDriveType
GetDriveType

VB 声明
Declare Function GetDriveType Lib "kernel32" Alias "GetDriveTypeA" (ByVal nDrive As String) As Long
说明
判断一个磁盘驱动器的类型
返回值
Long，如驱动器不能识别，则返回零。如指定的目录不存在，则返回 1。如执行成功，则用

下述任何一个常数指定驱动器类型： DRIVE_REMOVABLE， DRIVE_FIXED，DRIVE_REMOTE， DRIVE_CDROM 或 DRIVE_RAMDISK
参数表
参数类型及说明
nDriveString，包含了驱动器根目录路径的一个字串

Top
GetExpandedName
GetExpandedName

VB 声明
Declare Function GetExpandedName Lib "lz32.dll" Alias "GetExpandedNameA" (ByVal lpszSource As String, ByVal lpszBuffer As String) As Long
说明
取得一个压缩文件的全名。文件必须是用 COMPRESS.EXE 程序压缩的，而且在压缩时适用/r 选项
返回值
Long，1 表示成功，LZERROR_BADVALUE 表示失败
参数表
参数类型及说明
lpszSourceString，压缩文件的名字
lpszBufferString，指定一个缓冲区，用于装载文件全名
注解
注意事先将 lpszBuffer 字串初始化成一个合适的长度，使其足以容纳文件名

Top
GetFileAttributes
GetFileAttributes

VB 声明
Declare Function GetFileAttributes Lib "kernel32" Alias "GetFileAttributesA" (ByVal lpFileName As String) As Long
说明
判断指定文件的属性
返回值
Long，-1 表示出错。如返回包含了标志的一个 Long 值，则指定文件的属性。其中的标志对应于带有 FILE_ATTRIBUTE_??? 前缀的常数。具体参考 BY_HANDLE_FILE_INFORMATION 结构的 File Attribute Types table 表格
参数表
参数类型及说明
lpFileNameString，指定欲获取属性的一个文件的名字

Top
GetFileInformationByHandle

GetFileInformationByHandle

VB 声明
Declare Function GetFileInformationByHandle Lib "kernel32" Alias "GetFileInformationByHandle" (ByVal hFile As Long, lpFileInformation As BY_HANDLE_FILE_INFORMATION) As Long
说明
这个函数提供了获取文件信息的一种机制——在一个 BY_HANDLE_FILE_INFORMATION 结构中装载与文件有关的信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hFileLong，文件的句柄
lpFileInformationBY_HANDLE_FILE_INFORMATION，用于容纳文件信息的结构

Top
GetFileSize
GetFileSize

VB 声明
Declare Function GetFileSize Lib "kernel32" Alias "GetFileSize" (ByVal hFile As Long, lpFileSizeHigh As Long) As Long
说明
判断文件长度
返回值
Long，返回文件长度。&HFFFFFFFF 表示出错。注意如 lpFileSizeHigh 不为 NULL，且结果为&HFFFFFFFF，那么必须调用 GetLastError，判断是否实际发生了一个错误，因为这是一个有效的结果
参数表
参数类型及说明
hFileLong，文件的句柄
lpFileSizeHighLong，指定一个长整数，用于装载一个 64 位文件长度的头 32 位。如这个长度没有超过 2 ^ 32 字节，则该参数可以设为 NULL（变成 ByVal）

Top
GetFileTime
GetFileTime

VB 声明
Declare Function GetFileTime Lib "kernel32" Alias "GetFileTime" (ByVal hFile As Long, lpCreationTime As FILETIME, lpLastAccessTime As FILETIME, lpLastWriteTime As FILETIME) As Long
说明

取得指定文件的时间信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hFileLong，文件的句柄
lpCreationTimeFILETIME，用于装载文件的创建时间
lpLastAccessTimeFILETIME，用于装载文件上一次访问的时间（FAT 文件系统不支持这一特性）
lpLastWriteTimeFILETIME，用于装载文件上一次修改的时间
注解
如果不需要特定的信息，那么 lpCreationTime，lpLastAccessTime，lpLastWriteTime 都可以设置为零（用 ByVal As Long）。这个函数返回的文件时间采用 UTC 格式

Top
GetFileType
GetFileType

VB 声明
Declare Function GetFileType Lib "kernel32" Alias "GetFileType" (ByVal hFile As Long) As Long
说明
在给出文件句柄的前提下，判断文件类型
返回值
Long，下述常数之一：
FILE_TYPE_UNKNOWN 文件类型未知
FILE_TYPE_DISK 属于磁盘文件
FILE_TYPE_CHAR 文件是一个控制台或打印机
FILE_TYPE_PIPE 文件是个管道
参数表
参数类型及说明
hFileLong，要检查的文件的句柄

Top
GetFileVersionInfo
GetFileVersionInfo

VB 声明
Declare Function GetFileVersionInfo& Lib "version.dll" Alias "GetFileVersionInfoA" (ByVal lptstrFilename As String, ByVal dwHandle As Long, ByVal dwLen As Long, lpData As Byte)
说明
从支持版本标记的一个模块里获取文件版本信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError

参数表
参数类型及说明
lptstrFilenameString，欲从中载入版本信息的一个文件的名字
dwHandleLong，win32 中未用
dwLenLong，由 lpData 参数指定的字节数组或缓冲区的大小。用 GetFileVersionInfoSize 函数判断要求的缓冲区长度有多大
lpDataByte，指定一个字节缓冲区的第一个字节。该缓冲区用于装载文件的版本信息
注解
在 win32 下不用 dwHandle 参数
其他
请看 vb 的 api 文本查看器中的声明：Declare Function GetFileVersionInfo Lib "version.dll" Alias "GetFileVersionInfoA" (ByVal lptstrFilename As String, ByVal dwHandle As Long, ByVal dwLen As Long, lpData As Any) As Long

Top
GetFileVersionInfoSize
GetFileVersionInfoSize

VB 声明
Declare Function GetFileVersionInfoSize Lib "version.dll" Alias "GetFileVersionInfoSizeA" (ByVal lptstrFilename As String, lpdwHandle As Long) As Long
说明
针对包含了版本资源的一个文件，判断容纳文件版本信息需要一个多大的缓冲区
返回值
Long，容纳文件的版本资源所需的缓冲区长度。如文件不包含版本信息，则返回一个 0 值。会设置 GetLastError
参数表
参数类型及说明
lptstrFilenameString，包含了版本资源的一个文件的名字
lpdwHandleLong，在这个变量中载入 0 值
注解
lpdwHandle 参数在 win32 中已经放弃

Top
GetFullPathName
GetFullPathName

VB 声明
Declare Function GetFullPathName& Lib "kernel32" Alias "GetFullPathNameA" (ByVal lpFileName As String, ByVal nBufferLength As Long, ByVal lpBuffer As String, lpFilePart As Long)
说明
获取指定文件的完整路径名
返回值

Long，装载到 lpBuffer 中的字符数量（排除空中止字符）。如缓冲区的长度不足以容下完整的路径，则返回值就是要求的缓冲区大小。零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpFileNameString，指定文件名（长文件名或 8.3 格式的 DOS 文件名）
nBufferLengthLong，lpBuffer 字串的长度
lpBufferString，指定一个预先定义好的字串，用于装载目标文件的驱动器及路径名称。如存在长文件名，那么这个参数保存的就肯定是长文件名
lpFilePartLong，指定一个长整数变量，用于装载文件名起始的地方。参考注解
注解
lpFilePart 参数在 vb 里很难使用。它的问题在于：尽管 windows 在这个 Long 值中装载 lpBuffer 字串中的地址，用它表示路径信息文件名部分的起始处。但非常不幸，由 vb 创建的、传递给 api 的 ANSI 字串缓冲区也会使用这个地址。等这个函数返回的时候，vb 已将返回的（lpBuffer）字串复制回它的内部 Unicode 字串缓冲区，所以 lpFilePart 地址已没有任何意义。因此，我们面临两种选择。首先，可以简单的不使用 lpFilePart 信息（忽略 windows 装载在参数中的值）。其次，可以将 lpBuffer 参数变成一个字节数组（lpFilePart As Byte——将数组的第一个元素作为参数传递）
其他
在 vb 的 api 文本查看器中复制的声明为：Declare Function GetFullPathName Lib "kernel32" Alias "GetFullPathNameA" (ByVal lpFileName As String, ByVal nBufferLength As Long, ByVal lpBuffer As String, ByVal lpFilePart As String) As Long

Top
GetLogicalDrives
GetLogicalDrives

VB 声明
Declare Function GetLogicalDrives Lib "kernel32" Alias "GetLogicalDrives" () As Long
说明
判断系统中存在哪些逻辑驱动器字母
返回值
Long，这个结构中的二进制位标志着存在哪些驱动器。其中，位 0 设为 1 表示驱动器 A:存在于系统中；位 1 设为 1 表示存在 B:驱动器；以次类推

Top
GetLogicalDriveStrings
GetLogicalDriveStrings

VB 声明
Declare Function GetLogicalDriveStrings Lib "kernel32" Alias "GetLogicalDriveStringsA" (ByVal nBufferLength As Long, ByVal lpBuffer As String) As Long
说明
获取一个字串，其中包含了当前所有逻辑驱动器的根驱动器路径
返回值

Long，装载到 lpBuffer 的字符数量（排除空中止字符）。如缓冲区的长度不够，不能容下路径，则返回值就变成要求的缓冲区大小。零表示失败。会设置 GetLastError
参数表
参数类型及说明
nBufferLengthLong，lpBuffer 字串的长度
lpBufferString，用于装载逻辑驱动器名称的字串。每个名字都用一个 NULL 字符分隔，在最后一个名字后面用两个 NULL 表示中止（空中止）

Top
GetOverlappedResult
GetOverlappedResult

VB 声明
Declare Function GetOverlappedResult Lib "kernel32" Alias "GetOverlappedResult" (ByVal hFile As Long, lpOverlapped As OVERLAPPED, lpNumberOfBytesTransferred As Long, ByVal bWait As Long) As Long
说明
判断一个重叠操作当前的状态
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError。如 bWait 为 FALSE，而且异步操作仍在执行，则函数回返回零，而 GetLastError 会设置成 ERROR_IO_INCOMPLETE
参数表
参数类型及说明
hFileLong，指定一个文件、管道或通信设备的句柄
lpOverlappedOVERLAPPED，为欲检查的 I/O 操作指定的一个结构
lpNumberOfBytesTransferredLong，用于容纳传输字节数量的一个变量
bWaitLong，如果为 TRUE，就一直等到异步操作结束才返回。FALSE 表示立即返回

Top
GetPrivateProfileInt
GetPrivateProfileInt

VB 声明
Declare Function GetPrivateProfileInt Lib "kernel32" Alias "GetPrivateProfileIntA" (ByVal lpApplicationName As String, ByVal lpKeyName As String, ByVal nDefault As Long, ByVal lpFileName As String) As Long
说明
为初始化文件中指定的条目获取一个整数值
返回值
Long，找到的条目的值；如指定的条目未找到，就返回默认值。如找到的数字不是一个合法的整数，函数会返回其中合法的一部分。如，对于“xyz=55zz”这个条目，函数返回 55。这个函数也能理解采用标准 C 语言格式的十六进制数字：用 0x 作为一个十六进制数字的前缀——所以 0x55ab 等价于 vb 的&H55AB
参数表

参数类型及说明
lpApplicationNameString，指定在其中查找条目的小节。注意这个字串是不区分大小写的
lpKeyNameString，欲获取的设置项或条目。这个支持不区分大小写
nDefaultLong，指定条目未找到时返回的默认值
lpFileNameString，初始化文件的名字。如果没有指定完整的路径名，windows 就会在 Windows 目录中搜索文件
注解
在 Windows NT 中，有些初始化文件实际是在注册表中。可在注册表的下面这个项处找到这些文件的一个列表： HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\IniFileMapping

Top
GetPrivateProfileSection
GetPrivateProfileSection

VB 声明
Declare Function GetPrivateProfileSection Lib "kernel32" Alias "GetPrivateProfileSectionA" (ByVal lpAppName As String, ByVal lpReturnedString As String, ByVal nSize As Long, ByVal lpFileName As String) As Long
说明
获取指定小节所有项名和值的一个列表
返回值
Long，装载到 lpReturnedString 缓冲区的字符数量。如缓冲区的容量不够大，不能容下所有信息，就返回 nSize-2
参数表
参数类型及说明
lpAppNameString，欲获取的小节。注意这个字串不区分大小写
lpReturnedStringString，项和值字串的列表。每个字串都由一个 NULL 字符分隔，最后一个字串后面用两个 NULL 字符中止
nSizeLong，lpReturnedString 缓冲区的大小。在 windows 系统中最大值为 32767
lpFileNameString，初始化文件的名字。如没有指定完整路径名，windows 就在 Windows 目录中查找文件
注解
参考对 GetPrivateProfileInt 函数的注解

Top
GetPrivateProfileString
GetPrivateProfileString

VB 声明
Declare Function GetPrivateProfileString& Lib "kernel32" Alias "GetPrivateProfileStringA" (ByVal lpApplicationName As String, ByVal lpKeyName As String, ByVal lpDefault As String, ByVal lpReturnedString As String, ByVal nSize As Long, ByVal lpFileName As String)
说明

为初始化文件中指定的条目取得字串
返回值
Long，复制到 lpReturnedString 缓冲区的字节数量，其中不包括那些 NULL 中止字符。如 lpReturnedString 缓冲区不够大，不能容下全部信息，就返回 nSize-1（若 lpApplicationName 或 lpKeyName 为 NULL，则返回 nSize-2）
参数表
参数类型及说明
lpApplicationNameString，欲在其中查找条目的小节名称。这个字串不区分大小写。如设为 vbNullString，就在 lpReturnedString 缓冲区内装载这个 ini 文件所有小节的列表
lpKeyNameString，欲获取的项名或条目名。这个字串不区分大小写。如设为 vbNullString，就在 lpReturnedString 缓冲区内装载指定小节所有项的列表
lpDefaultString，指定的条目没有找到时返回的默认值。可设为空（""）
lpReturnedStringString，指定一个字串缓冲区，长度至少为 nSize
nSizeLong，指定装载到 lpReturnedString 缓冲区的最大字符数量
lpFileNameString，初始化文件的名字。如没有指定一个完整路径名，windows 就在 Windows 目录中查找文件
注解
如 lpKeyName 参数为 vbNullString，那么 lpReturnedString 缓冲区会载入指定小节所有设置项的一个列表。每个项都用一个 NULL 字符分隔，最后一个项用两个 NULL 字符中止。也请参考 GetPrivateProfileInt 函数的注解
其他
在 vb 的 api 文本查看器中复制的声明为：Declare Function GetPrivateProfileString Lib "kernel32" Alias "GetPrivateProfileStringA" (ByVal lpApplicationName As String, ByVal lpKeyName As Any, ByVal lpDefault As String, ByVal lpReturnedString As String, ByVal nSize As Long, ByVal lpFileName As String) As Long

Top
GetProfileInt
GetProfileInt

VB 声明
Declare Function GetProfileInt Lib "kernel32" Alias "GetProfileIntA" (ByVal lpAppName As String, ByVal lpKeyName As String, ByVal nDefault As Long) As Long
说明
取得 win.ini 初始化文件中指定条目的一个整数值
返回值
Long，找到条目的值；如指定的条目未找到，就返回默认值。如找到的数字不是一个合法的整数，函数就会返回其中合法的一部分。例如，对于“xyz=55zz”这个条目，函数会返回 55。这个函数也能理解采用标准 C 语言格式的十六进制数字：用 0x 作为一个十六进制数字的前缀——所以 0x55ab 等价于 vb 的&H55AB
参数表
参数类型及说明
lpAppNameString，欲在其中搜索条目的小节名。这个字串不区分大小写
lpKeyNameString，欲获取的项名或条目名。这个字串不区分大小写

nDefaultLong，指定在条目未找到时返回的默认值
注解
参考对 GetPrivateProfileInt 函数的注解

Top
GetProfileSection
GetProfileSection

VB 声明
Declare Function GetProfileSection Lib "kernel32" Alias "GetProfileSectionA" (ByVal lpAppName As String, ByVal lpReturnedString As String, ByVal nSize As Long) As Long
说明
获取指定小节（在 win.ini 文件中）所有项名和值的一个列表
返回值
Long，装载到 lpReturnedString 缓冲区的字符数量。如缓冲区的长度不足以容下所有信息，则返回 nSize-2
参数表
参数类型及说明
lpAppNameString，欲获取的小节。这个字串不区分大小写
lpReturnedStringString，用于容纳项和值字串列表的一个缓冲区。每个字串都用一个 NULL 分隔，最后一个字串用两个 NULL 字符中止
nSizeLong，lpReturnedString 缓冲区的大小，在 windows 95 中最大为 32767
注解
参考 GetPrivateProfileInt 函数的注解

Top
GetProfileString
GetProfileString

VB 声明
Declare Function GetProfileString Lib "kernel32" Alias "GetProfileStringA" (ByVal lpAppName As String, ByVal lpKeyName As String, ByVal lpDefault As String, ByVal lpReturnedString As String, ByVal nSize As Long) As Long
说明
为 win.ini 初始化文件中指定的条目取得字串
返回值
Long，复制到 lpReturnedString 缓冲区的字节数量，其中不包括那些 NULL 中止字符。如 lpReturnedString 缓冲区不够大，不能容下全部信息，就返回 nSize-1（若 lpAppName 或 lpKeyName 为 NULL，则返回 nSize-2）
参数表
参数类型及说明
lpAppNameString，要在其中查找条目的小节名。这个字串不区分大小写。如果为 vbNullString，则在 lpReturnedString 缓冲区装载这个.ini 文件的所有小节的一个列表
lpKeyNameString，欲获取的项名或条目名。这个字串不区分大小写。如果为 vbNullString，

则在 lpReturnedString 缓冲区装载指定小节内所有项的一个列表
lpDefaultString，指定条目未找到时返回的默认值。可设为空（""）
lpReturnedStringString，指定一个预先初始化好的字串缓冲区，长度至少为 nSize 个字符
nSizeLong，装载到 lpReturnedString 缓冲区的最大字符数
注解
如 lpKeyName 参数为零，那么 lpReturnedString 缓冲区会载入指定小节内所有设置项的一个列表。每个项都用一个 NULL 字符分隔，最后那个项用两个 NULL 字符中止

Top
GetShortPathName
GetShortPathName

VB 声明
Declare Function GetShortPathName Lib "kernel32" Alias "GetShortPathName" (ByVal lpszLongPath As String, ByVal lpszShortPath As String, ByVal cchBuffer As Long) As Long
说明
获取指定文件的短路径名
返回值
Long，装载到 lpszShortPath 缓冲区的字符数量。如 lpszShortPath 的长度不足，不能容下文件名，就返回需要的缓冲区长度
参数表
参数类型及说明
lpszLongPathString，指定欲获取短路径名的那个文件的名字。可以是个完整路径，或者由当前目录决定
lpszShortPathString，指定一个缓冲区，用于装载文件的短路径和文件名
cchBufferLong，lpszShortPath 缓冲区长度

Top
GetSystemDirectory
GetSystemDirectory

VB 声明
Declare Function GetSystemDirectory Lib "kernel32" Alias "GetSystemDirectoryA" (ByVal lpBuffer As String, ByVal nSize As Long) As Long
说明
这个函数能取得 Windows 系统目录（System 目录）的完整路径名。在这个目录中，包含了所有必要的系统文件。根据微软的标准，其他定制控件和一些共享组件也可放到这个目录。通常应避免在这个目录里创建文件。在网络环境中，往往需要管理员权限才可对这个目录进行写操作
返回值
Long，装载到 lpBuffer 缓冲区的字符数量。如 lpBuffer 不够大，不能容下文件名，则返回要求的缓冲区长度
参数表
参数类型及说明

lpBufferString，用于装载系统目录路径名的一个字串缓冲区。它应事先初始化成 nSize+1 个字符的长度。通常至少要为这个缓冲区分配 MAX_PATH 个字符的长度
nSizeLong，lpBuffer 字串的最大长度

Top
GetTempFileName
GetTempFileName

VB 声明
Declare Function GetTempFileName Lib "kernel32" Alias "GetTempFileNameA" (ByVal lpszPath As String, ByVal lpPrefixString As String, ByVal wUnique As Long, ByVal lpTempFileName As String) As Long
说明
这个函数包含了一个临时文件的名字，它可由应用程序使用
返回值
Long，最终用于生成文件名的 wUnique 数字的值。如 wUnique 参数不为零，这就是参数的值。零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpszPathString，临时文件使用的目录。通常用 GetTempPath 函数获得
lpPrefixStringString，要使用的文件名前缀。头三个字符作为文件名前缀使用
wUniqueLong，追加到前缀字串后面的数字。如果为 0，则这个函数会用一个随机数字生成文件。随后，它会检查是否存在同名的文件。如果存在，函数会增加这个数字，并继续尝试，直到生成一个独一无二的名字为止。文件在驱动器上会以长度为 0 字节的形式保存。如果不为零，就不会创建文件，而且函数不会核实它是否一个独一无二的文件名
lpTempFileNameString，用于装载新建临时文件名的缓冲区，这个缓冲区的长度至少应为 MAX_PATH 个字符
注解
函数使用的文件名肯定采用 ANSI 字符集。临时文件不会被 windows 自动删除

Top
GetTempPath
GetTempPath

VB 声明
Declare Function GetTempPath Lib "kernel32" Alias "GetTempPathA" (ByVal nBufferLength As Long, ByVal lpBuffer As String) As Long
说明
获取为临时文件指定的路径
返回值
Long，装载到 lpBuffer 的字符数。如当前缓冲区的长度不够，不能容下整个路径，则返回 lpBuffer 需要的长度。零表示失败。会设置 GetLastError
参数表
参数类型及说明

nBufferLengthLong，lpBuffer 字串的长度
lpBufferString，用于装载临时文件路径的一个预初始化字串
注解
临时路径是由 TMP 环境变量指定的一个路径。如 TMP 不存在，则是由 TEMP 环境变量指定的路径。如果这两个环境变量都不存在，就是当前目录

Top
GetVolumeInformation
GetVolumeInformation

VB 声明
Declare Function GetVolumeInformation Lib "kernel32" Alias "GetVolumeInformationA" (ByVal lpRootPathName As String, ByVal lpVolumeNameBuffer As String, ByVal nVolumeNameSize As Long, lpVolumeSerialNumber As Long, lpMaximumComponentLength As Long, lpFileSystemFlags As Long, ByVal lpFileSystemNameBuffer As String, ByVal nFileSystemNameSize As Long) As Long
说明
获取与一个磁盘卷有关的信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpRootPathNameString，欲获取信息的那个卷的根路径
lpVolumeNameBufferString，用于装载卷名（卷标）的一个字串
nVolumeNameSizeLong，lpVolumeNameBuffer 字串的长度
lpVolumeSerialNumberLong，用于装载磁盘卷序列号的变量
lpMaximumComponentLengthLong，指定一个变量，用于装载文件名每一部分的长度。例如，在“c:\component1\component2.ext”的情况下，它就代表 component1 或 component2 名称的长度
lpFileSystemFlagsLong，用于装载一个或多个二进制位标志的变量。对这些标志位的解释如下：
FS_CASE_IS_PRESERVED 文件名的大小写记录于文件系统
FS_CASE_SENSITIVE 文件名要区分大小写
FS_UNICODE_STORED_ON_DISK 文件名保存为 Unicode 格式
FS_PERSISTANT_ACLS 文件系统支持文件的访问控制列表（ACL）安全机制
FS_FILE_COMPRESSION 文件系统支持逐文件的进行文件压缩
FS_VOL_IS_COMPRESSED 整个磁盘卷都是压缩的
lpFileSystemNameBufferString，指定一个缓冲区，用于装载文件系统的名称（如 FAT，NTFS 以及其他）
nFileSystemNameSizeLong，lpFileSystemNameBuffer 字串的长度

Top
GetWindowsDirectory
GetWindowsDirectory

VB 声明
Declare Function GetWindowsDirectory Lib "kernel32" Alias "GetWindowsDirectoryA" (ByVal lpBuffer As String, ByVal nSize As Long) As Long
说明
这个函数能获取 Windows 目录的完整路径名。在这个目录里，保存了大多数 windows 应用程序文件及初始化文件
返回值
Long，复制到 lpBuffer 的一个字串的长度。如 lpBuffer 不够大，不能容下整个字串，就会返回 lpBuffer 要求的长度。零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpBufferString，指定一个字串缓冲区，用于装载 Windows 目录名。除非是根目录，否则目录中不会有一个中止用的“\”字符
nSizeLong，lpBuffer 字串的最大长度

Top
hread
hread

VB 声明
Declare Function hread Lib "kernel32" Alias "_hread" (ByVal hFile As Long, lpBuffer As Any, ByVal lBytes As Long) As Long
说明
参考 lread
注解
这个函数在 win16 中用于读取大于 64KB 的数据块。但 win32 的文件 I/O 函数并不受这个 64KB 的限制

Top
hwrite
hwrite

VB 声明
Declare Function hwrite Lib "kernel32" Alias "_hwrite" (ByVal hFile As Long, ByVal lpBuffer As String, ByVal lBytes As Long) As Long
说明
参考 lwrite 函数
注解
这个函数在 win16 中用于写入大于 64KB 的数据块。但 win32 的文件 I/O 函数并不受这个 64KB 的限制

Top
lclose

lclose

VB 声明
Declare Function lclose Lib "kernel32" Alias "_lclose" (ByVal hFile As Long) As Long
说明
关闭指定的文件，请参考 CloseHandle 函数，了解进一步的情况

Top
lcreat
lcreat

VB 声明
Declare Function lcreat Lib "kernel32" Alias "_lcreat" (ByVal lpPathName As String, ByVal iAttribute As Long) As Long
说明
创建一个文件。如文件已经存在，就会将其缩短成零长度，并将其打开，以便读写
返回值
Long，如执行成功，返回打开文件的句柄。如果出错，则返回 HFILE_ERROR.
参数表
参数类型及说明
lpPathNameString，欲创建的文件的名字
iAttributeLong，下述值之一
0——文件能够读写
1——创建只读文件
2——创建隐藏文件
3——创建系统文件
注解
该函数会打开已由其他应用程序打开的文件，所以使用它时要小心。win32 的 CreateFile 函数已取代了这个函数。这个函数与 vb 的 open 语句作用相同

Top
llseek
llseek

VB 声明
Declare Function llseek Lib "kernel32" Alias "_llseek" (ByVal hFile As Long, ByVal lOffset As Long, ByVal iOrigin As Long) As Long
说明
设置文件中进行读写的当前位置。该函数与 vb 的 seek 语句类似。如果用 vb 的 open 命令打开了一个文件，那么不要再对这个文件使用 llseek 函数
返回值
Long，返回一个新位置，设置成从文件起始处算起的一个偏移量。HFILE_ERROR 表示函数执行出错。会设置 GetLastError
参数表

参数类型及说明
hFileLong，系统文件句柄
lOffsetLong，字节偏移量
iOriginLong，下述常数之一
FILE_BEGINlOffset 将新位置指定成从文件起始处的一个偏移距离
FILE_CURRENTlOffset 将新位置指定成从当前位置开始的一个偏移距离
FILE_ENDlOffset 将新位置指定成从文件结尾开始的的一个偏移距离
注解
参考 SetFilePointer 函数，认识能对较大文件进行处理的一个近似函数.

Top
LockFile
LockFile

VB 声明
Declare Function LockFile Lib "kernel32" Alias "LockFile" (ByVal hFile As Long, ByVal dwFileOffsetLow As Long, ByVal dwFileOffsetHigh As Long, ByVal nNumberOfBytesToLockLow As Long, ByVal nNumberOfBytesToLockHigh As Long) As Long
说明
在 windows 中，文件可用共享模式打开——在这种情况下，多个进程可同时访问该文件。利用这个函数，要对文件进行读写的一个应用程序可将文件的某一部分锁定起来，使其不能由其他应用程序访问。这样便避免了同时读写时发生的冲突
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hFileLong，欲锁定文件的句柄
dwFileOffsetLowLong，指定欲锁定区域起始处的低 32 位地址
dwFileOffsetHighLong，指定欲锁定区域起始处的高 32 位地址
nNumberOfBytesToLockLowLong，锁定区域包含字符数量的低 32 位值
nNumberOfBytesToLockHighLong，锁定区域包含字符数量的高 32 位值
注解
锁定的区域不能进行重叠操作。由不同的操作系统决定，可能要求先运行 share.exe 才能保证该函数正常工作

Top
LockFileEx
LockFileEx

VB 声明
Declare Function LockFileEx Lib "kernel32" Alias "LockFileEx" (ByVal hFile As Long, ByVal dwFlags As Long, ByVal dwReserved As Long, ByVal nNumberOfBytesToLockLow As Long, ByVal nNumberOfBytesToLockHigh As Long, lpOverlapped As OVERLAPPED) As Long
说明

与 LockFile 相似，只是它提供了更多的功能
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hFileLong，欲锁定文件的句柄
dwFlagsLong，指定下述一个或两个常数
LOCKFILE_FAIL_IMMEDIATELY 指出如锁定失败，函数应返回一个错误。否则，应用程序线程就会暂时挂起，并一直等待，直到能进行锁定为止
LOCKFILE_EXCLUSIVE_LOCK 指出锁定区域不可由另一个线程或进程读写。否则这个区域就只能防范“写”——其他进程仍然能够读取锁定区域的内容
dwReservedLong，未使用，设为零
nNumberOfBytesToLockLowLong，锁定区域包含字符数的低 32 位
nNumberOfBytesToLockHighLong，锁定区域包含字符数的高 32 位
lpOverlappedOVERLAPPED，包含了文件中相对于锁定区域起始处的偏移量
注解
锁定区域不可重叠操作（即多个进程同时操作）

Top
lopen
lopen

VB 声明
Declare Function lopen Lib "kernel32" Alias "_lopen" (ByVal lpPathName As String, ByVal iReadWrite As Long) As Long
说明
以二进制模式打开指定的文件
返回值
Long，如执行成功，返回打开文件的句柄。HFILE_ERROR 表示出错。会设置 GetLastError
参数表
参数类型及说明
lpPathNameString，欲打开文件的名字
iReadWriteLong，访问模式和共享模式常数的一个组合，如下所示：
1、访问模式
READ 打开文件，读取其中的内容
READ_WRITE 打开文件，对其进行读写
WRITE 打开文件，在其中写入内容
2、共享模式（参考 OpenFile 函数的标志常数表）
OF_SHARE_COMPAT ， OF_SHARE_DENY_NONE ， OF_SHARE_DENY_READ ， OF_SHARE_DENY_WRITE， OF_SHARE_EXCLUSIVE
注解
CreateFile 函数在 win32 下提供了更多的功能

Top

lread
lread

VB 声明
Declare Function lread Lib "kernel32" Alias "_lread" (ByVal hFile As Long, lpBuffer As Any, ByVal wBytes As Long) As Long
说明
将文件中的数据读入内存缓冲区
返回值
Long，返回实际读入的字节数。HFILE_ERROR 意味着函数执行出错。如这个数字小于 wBytes，则表明早已抵达了文件的末尾。会设置 GetLastError
参数表
参数类型及说明
hFileLong，文件句柄
lpBufferAny，指定一个内存块的指针，数据将读入这个内存块
wBytesLong，要读入的字节数

Top
lwrite
lwrite

VB 声明
Declare Function lwrite Lib "kernel32" Alias "_lwrite" (ByVal hFile As Long, ByVal lpBuffer As String, ByVal wBytes As Long) As Long
说明
将数据从内存缓冲区写入一个文件
返回值
Long，写入的字节数。HFILE_ERROR 意味着函数执行出错。如这个数字小于 wBytes，则表明早已抵达了文件的末尾。会设置 GetLastError
参数表
参数类型及说明
hFileLong，文件句柄
lpBufferString，指定一个内存块的指针，把这个内存块的数据写入文件
wBytesLong，要写入的字节数

Top
LZClose
LZClose

VB 声明
Declare Sub LZClose Lib "lz32.dll" Alias "LZClose" (ByVal hfFile As Long)
说明
关闭由 LZOpenFile 或 LZInit 函数打开的一个文件
参数表

参数类型及说明
hfFileLong，欲关闭的句柄。这是由 LZOpenFile 或 LZInit 函数返回的一个句柄，不是普通的系统文件句柄

Top
LZCopy
LZCopy

VB 声明
Declare Function LZCopy Lib "lz32.dll" Alias "LZCopy" (ByVal hfSource As Long, ByVal hfDest As Long) As Long
说明
复制一个文件。如源文件已压缩，则会在复制期间解压。文件必须是用微软公司的 compress.exe 或等效工具压缩的
返回值
Long，如执行成功，返回目标文件的大小，以字节为单位。如执行出错，会返回小于零的一个常数，如下表
LZERROR_BADINHANDLE 源文件无效
LZERROR_BADOUTHANDLE 目标文件无效
LZERROR_GLOBALLOC 内部解压缓冲区的内存容量不足
LZERROR_GLOBLOCK 内部解压缓冲区的句柄无效
LZERROR_READ 无效的源文件格式
LZERROR_UNKNOWNALG 解压 DLL 不能识别源文件采用的压缩算法
LZERROR_WRITE 在磁盘上写入输出文件时出错，通常是由于磁盘空间不足造成的
参数表
参数类型及说明
hfSourceLong，指定源文件句柄。这是由 LZOpenFile 或 LZInit 函数返回的一个句柄
hfDestLong，指定目标文件句柄。这是由 LZOpenFile 或 LZInit 函数返回的一个句柄

Top
LZInit
LZInit

VB 声明
Declare Function LZInit Lib "lz32.dll" Alias "LZInit" (ByVal hfSrc As Long) As Long
说明
这个函数用于初始化内部缓冲区。对一个给出打开文件句柄的一个文件进行解压时，将用到这个缓冲区
返回值
Long，由 lz32.dll 库使用的、那个文件的一个特殊句柄。这个文件句柄兼容于 LZCopy，CopyLZFiles， LZRead 和 LZSeek 函数。如果出错，该函数会返回下表列出的出错代码之一。注意完成后一定用 LZClose 关闭这个句柄
LZERROR_BADINHANDLE 源文件无效
LZERROR_BADOUTHANDLE 目标文件无效

LZERROR_GLOBALLOC 内部解压缓冲区的内存容量不足
LZERROR_GLOBLOCK 内部解压缓冲区的句柄无效
LZERROR_READ 无效的源文件格式
LZERROR_UNKNOWNALG 解压 DLL 不能识别源文件采用的压缩算法
LZERROR_WRITE 在磁盘上写入输出文件时出错，通常是由于磁盘空间不足造成的
参数表
参数类型及说明
hfSrcLong，文件的句柄
注解
最多只能同时打开 16 个压缩文件句柄

Top
LZOpenFile
LZOpenFile

VB 声明
Declare Function LZOpenFile Lib "lz32.dll" Alias "LZOpenFileA" (ByVal lpszFile As String, lpOf As OFSTRUCT, ByVal style As Long) As Long
说明
该函数能执行大量不同的文件处理，而且兼容于压缩文件
返回值
Long，如函数执行成功，且样式（style）参数不为 OF_READ，就返回常规的文件句柄，具体请参考 OpenFile 函数的说明。如样式参数为 OF_READ，而且文件是压缩的，就会返回一个特殊的文件句柄，以便由 LZCopy， LZRead 和 LZSeek 函数使用。如出错，返回如下表所示的一个常数：
LZERROR_BADINHANDLE 源文件无效
LZERROR_BADOUTHANDLE 目标文件无效
LZERROR_GLOBALLOC 内部解压缓冲区的内存容量不足
LZERROR_GLOBLOCK 内部解压缓冲区的句柄无效
LZERROR_READ 无效的源文件格式
LZERROR_UNKNOWNALG 解压 DLL 不能识别源文件采用的压缩算法
LZERROR_WRITE 在磁盘上写入输出文件时出错，通常是由于磁盘空间不足造成的
参数表
参数类型及说明
lpszFileString，欲打开的文件名
lpOfOFSTRUCT，该结构填充的数据包括与本次处理的文件和结果有关的信息
styleLong，处理方式标志常数的一种组合。参考 OpenFile 函数的标志常数表
注解
参考 OpenFile 函数

Top
LZRead
LZRead

VB 声明
Declare Function LZRead Lib "lz32.dll" Alias "LZRead" (ByVal hfFile As Long, ByVal lpvBuf As String, ByVal cbread As Long) As Long
说明
将数据从文件读入内存缓冲区。如 hfFile 是一个压缩文件的句柄，同时那个压缩文件是由 LZOpenFile 或 LZInit 函数打开的，这个函数就会在读入数据的同时对文件进行解压处理
返回值
Long，实际读入的字节数。如这个数字小于 cbread，表明早已抵达了文件的末尾。如出错，返回下表列出的常数之一
LZERROR_BADINHANDLE 源文件无效
LZERROR_BADOUTHANDLE 目标文件无效
LZERROR_GLOBALLOC 内部解压缓冲区的内存容量不足
LZERROR_GLOBLOCK 内部解压缓冲区的句柄无效
LZERROR_READ 无效的源文件格式
LZERROR_UNKNOWNALG 解压 DLL 不能识别源文件采用的压缩算法
LZERROR_WRITE 在磁盘上写入输出文件时出错，通常是由于磁盘空间不足造成的
参数表
参数类型及说明
hfFileLong，源文件的特殊句柄。这个句柄是由 LZOpenFile 或 LZInit 函数提供的
lpvBufString，一个内存块的指针，数据将读入这个内存块
cbreadLong，指定 lpvBuf 缓冲区的长度

Top
LZSeek
LZSeek

VB 声明
Declare Function LZSeek Lib "lz32.dll" Alias "LZSeek" (ByVal hfFile As Long, ByVal lOffset As Long, ByVal nOrigin As Long) As Long
说明
设置一个文件中进行读写的当前位置。如 hfFile 是一个压缩文件的句柄，同时那个压缩文件是由 LZOpenFile 或 LZInit 函数打开的，这个函数就会根据文件的解压版本进行查找
返回值
Long，返回一个新位置，采用从文件起始处计算的字节偏移量。如出错，返回下表列出的常数之一
LZERROR_BADINHANDLE 源文件无效
LZERROR_BADOUTHANDLE 目标文件无效
LZERROR_GLOBALLOC 内部解压缓冲区的内存容量不足
LZERROR_GLOBLOCK 内部解压缓冲区的句柄无效
LZERROR_READ 无效的源文件格式
LZERROR_UNKNOWNALG 解压 DLL 不能识别源文件采用的压缩算法
LZERROR_WRITE 在磁盘上写入输出文件时出错，通常是由于磁盘空间不足造成的
参数表
参数类型及说明

hfFileLong，源文件的特殊句柄。这个句柄是由 LZOpenFile 或 LZInit 函数提供的
lOffsetLong，以字节数表示的偏移量
nOriginLong，下述值之一
0——lOffset 将新位置指定成从文件的起始处计算偏移
1——lOffset 将新位置指定成从当前位置开始计算偏移
2——lOffset 将新位置指定成从文件的结尾处计算偏移

Top
MapViewOfFile
MapViewOfFile, MapViewOfFileEx

VB 声明
Declare Function MapViewOfFile& Lib "kernel32" (ByVal hFileMappingObject As Long, ByVal dwDesiredAccess As Long, ByVal dwFileOffsetHigh As Long, ByVal dwFileOffsetLow As Long, ByVal dwNumberOfBytesToMap As Long)
Declare Function MapViewOfFileEx& Lib "kernel32" (ByVal hFileMappingObject As Long, ByVal dwDesiredAccess As Long, ByVal dwFileOffsetHigh As Long, ByVal dwFileOffsetLow As Long, ByVal dwNumberOfBytesToMap As Long, lpBaseAddress As Any)
说明
将一个文件映射对象映射到当前应用程序的地址空间。MapViewOfFileEx 允许我们指定一个基本地址来进行映射
返回值
Long，文件映射在内存中的起始地址。零表示出错。会设置 GetLastError
参数表
参数类型及说明
hFileMappingObjectLong，文件映射对象的句柄
dwDesiredAccessLong，下述常数之一：
FILE_MAP_WRITE 映射可读可写。文件映射对象必须通过 PAGE_READWRITE 访问创建
FILE_MAP_READ 映射只读。文件映射对象必须通过 PAGE_READ 或 PAGE_READWRITE 访问创建
FILE_MAP_ALL_ACCESS 与 FILE_MAP_WRITE 相同
FILE_MAP_COPY 映射时保留写操作的副本。文件映射对象必须用 PAGE_WRITECOPY 访问在 win95 下创建
dwFileOffsetHighLong，文件中映射起点的高 32 位地址
dwFileOffsetLowLong，文件中映射起点的低 32 位地址
dwNumberOfBytesToMapLong，文件中要映射的字节数。用零映射整个文件映射对象
lpBaseAddressLong，指定映射文件映射对象的地址。如这个地址处没有足够的内存空间，那么对 MapViewOfFileEx 的调用会失效。零表示允许 windows 寻找一个地址
注解
dwFileOffsetLow 和 dwFileOffsetHigh 必须反映一个偏移距离，它由系统的内存分配精度决定。例如，假设系统的内存精度是 64KB（即最小分配单位是 64KB），则这些值必须是 64KB 的整数倍。大多数应用程序都简单的用零从文件的起始处开始映射。 lpBaseAddress 也必须是内存分配精度的整数倍
其他

声明中的参数类型为 Any，而参数表中都是 Long，我也不明白。但关于这个函数的英文资料的确是这样的。

Top
MoveFile
MoveFile, MoveFileEx

VB 声明
Declare Function MoveFile& Lib "kernel32" Alias "MoveFileA" (ByVal lpExistingFileName As String, ByVal lpNewFileName As String)
Declare Function MoveFileEx& Lib "kernel32" Alias "MoveFileExA" (ByVal lpExistingFileName As String, ByVal lpNewFileName As String, ByVal dwFlags As Long)
说明
移动文件。如 dwFlags 设为零，则 MoveFile 完全等价于 MoveFileEx
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpExistingFileNameString，欲移动的文件名
lpNewFileNameString，新文件名
dwFlagsLong，一个或多个下述常数
MOVEFILE_REPLACE_EXISTING 如目标文件存在，则将其替换
MOVEFILE_COPY_ALLOWED 如移动到一个不同的卷，则复制文件并删除原来的文件
MOVEFILE_DELAY_UNTIL_REBOOT 移动操作在系统下次重新启动时正式进行。这样便可在 Windows NT 中改换系统文件
注解
这两个函数通常不能将文件从一个卷移动到另一个卷。但如设置了 MOVEFILE_COPY_ALLOWED 标记，MoveFileEx 可以做到这一点

Top
OpenFile
OpenFile

VB 声明
Declare Function OpenFile Lib "kernel32" Alias "OpenFile" (ByVal lpFileName As String, lpReOpenBuff As OFSTRUCT, ByVal wStyle As Long) As Long
说明
这个函数能执行大量不同的文件操作。和这个函数相比，请优先考虑 win32 的 CreateFile 函数（它能打开命名管道和控制 Unicode 文件名，同时不受 128 个字符的路径名称的限制）
返回值
Long，如执行成功，返回文件句柄。注意文件句柄可能是无效的；例如，假设指定了 OF_EXIST 标志，文件在函数返回前会关闭，但它打开时的句柄却永远不会返回。如果出错，函数会返回 HFILE_ERROR；此时，由 lpReOpenBuff 指定的 OFSTRUCT 结构的 nErrCode 会设置成发生的错误。表 OpenFile-2（OFSTRUCT 出错代码）对这些错误进行了总结。会设置

GetLastError
参数表
参数类型及说明
lpFileNameString，欲打开文件的名字
lpReOpenBuffOFSTRUCT，该结构填充的数据包括与文件和操作结果有关的信息
wStyleLong，参考表 OpenFile-1（OpenFile 函数的标志常数表）总结的标志常数的组合，它决定了要采取的操作方式

表 OpenFile-1（OpenFile 函数的标志常数表）
wStyle 常数说明
OF_CREATE 创建指定的文件。如已经存在，则将其缩减为零长度
OF_DELETE 删除指定的文件
OF_EXIST 通过尝试打开文件的做法，判断一个文件是否存在。如文件存在，则将其关闭。此时，函数会返回文件打开时使用的句柄，但这个句柄是无效的。如指定的文件不存在，则返回一个负数
OF_PARSE 填写 lpReOpenBuff 结构的内容，但不执行其他任何操作
OF_PROMPT 如文件不存在，则显示一个消息框，在其中列出重试和取消按钮
OF_READ 以只读方式打开文件
OF_READWRITE 以可读、可写的方式打开文件
OF_REOPEN 打开 lpReOpenBuff 结构内指定的文件，而不是用 lpFileName 参数
OF_SEARCH 强迫 windows 查找文件——即使指定了特定的路径
OF_SHARE_COMPAT 文件可由多个应用程序打开多次
OF_SHARE_DENY_NONE 可打开文件，以便由其他程序读写
OF_SHARE_DENY_READ 禁止其他程序读写文件内容
OF_SHARE_DENY_WRITE 其他程序可以读文件，但不能写文件
OF_SHARE_EXCLUSIVE 其他任何一个程序都不能再打开这个文件
OF_WRITE 文件以只写模式打开
表 OpenFile-2（OFSTRUCT 出错代码）
十六进制值说明十六进制值说明
1 函数无效 2 文件未找到
3 路径未找到 4 无可用文件句柄
5 拒绝访问 6 句柄无效
7DOS 内存冲突 8 无足够内存完成操作
9 无效块 A 非法环境
B 无效格式 C 无效访问
D 无效数据
F 无效驱动器 10 当前目录无效
11 设备有异 12 没有更多的文件
13 写保护错 14 非法单位
15 驱动器未准备好 16 无效命令
17CRC 校验错 18 无效长度
19 搜索错误 1A 磁盘不兼容 MS-DOS
1B 扇区未找到 1C 缺纸
1D 写错误 1E 读错误

1F 驱动器常规错误 20 共享违例
21 文件锁定违例 22 不正确的磁盘
23 无可用的文件控制块 24 共享缓冲区溢出
32 不支持的设备 33 远程设备不可用
34 重名错误 35 网络路径错误
36 网络忙 37 非法设备
38 命令太多 39 网卡硬件错误
3A 网络响应错误 3B 其他网络错误
3C 远程适配器错误 3D 打印队列满
3E 后台打印缓冲区满 3F 打印取消
40 删除的网络名 41 拒绝网络访问
42 无效设备类型 43 无效网络名
44 名字太多 45 会话太多
46 共享暂停 47 请求未接受
48 重定向暂停 50 文件退出
51 文件控制块重复 52 不能创建
53 中断 24 错误 54 缺少结构
55 已经分配 56 密码无效
57 参数无效 58 网络写错误

Top
OpenFileMapping
OpenFileMapping

VB 声明
Declare Function OpenFileMapping Lib "kernel32" Alias "OpenFileMappingA" (ByVal dwDesiredAccess As Long, ByVal bInheritHandle As Long, ByVal lpName As String) As Long
说明
打开一个现成的文件映射对象
返回值
Long，指定文件映射对象的句柄。零表示出错。会设置 GetLastError
参数表
参数类型及说明
dwDesiredAccessLong，带有前缀 FILE_MAP_???的一个常数。参考 MapViewOfFile 函数的 dwDesiredAccess 参数的说明
bInheritHandleLong，如这个函数返回的句柄能由当前进程启动的新进程继承，则这个参数为 TRUE
lpNameString，指定要打开的文件映射对象名称

Top
QueryDosDevice
QueryDosDevice

VB 声明

Declare Function QueryDosDevice Lib "kernel32" Alias "QueryDosDeviceA" (ByVal lpDeviceName As String, ByVal lpTargetPath As String, ByVal ucchMax As Long) As Long
说明
在 Windows NT 中，DOS 设备名会映射成 NT 系统设备名。该函数可判断当前的设备映射情况
返回值
Long，零表示出错。如执行成功，返回保存到 lpTargetPath 的字符数。会设置 GetLastError
参数表
参数类型及说明
lpDeviceNameString，如果是 vbNullString，那么 lpTargetPath 会载入当前映射的 MS-DOS 名称的一个列表。如果是个 MS-DOS 名，则 lpTargetPath 会载入一个设备映射列表（第一个名字是活动映射，后续的名字是以前尚未删掉的映射）
lpTargetPathString，名称列表，具体取决于 lpDeviceName 参数。这些名字用 NULL 字符分隔。列表最后用两个连续的 NULL 字符中止
ucchMaxLong，lpTargetPath 缓冲区的大小
注解
可用 DefineDosDevice 函数将映射变成 DOS 设备名
适用平台
Windows NT

Top
ReadFile
ReadFile

VB 声明
Declare Function ReadFile Lib "kernel32" Alias "ReadFile" (ByVal hFile As Long, lpBuffer As Any, ByVal nNumberOfBytesToRead As Long, lpNumberOfBytesRead As Long, lpOverlapped As OVERLAPPED) As Long
说明
从文件中读出数据。与 lread 函数相比，这个函数要明显灵活的多。该函数能够操作通信设备、管道、套接字以及邮槽
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError。如启动的是一次异步读操作，则函数会返回零值，并将 ERROR_IO_PENDING 设置成 GetLastError 的结果。如结果不是零值，但读入的字节数小于 nNumberOfBytesToRead 参数指定的值，表明早已抵达了文件的结尾
参数表
参数类型及说明
hFileLong，文件的句柄
lpBufferAny，用于保存读入数据的一个缓冲区
nNumberOfBytesToReadLong，要读入的字符数
lpNumberOfBytesReadLong，从文件中实际读入的字符数
lpOverlappedOVERLAPPED，如文件打开时指定了 FILE_FLAG_OVERLAPPED，那么必须用这个参数引用一个特殊的结构。那个结构定义了一次异步读取操作。否则，应将这个参数设为 NULL（将函数声明成 ByVal As Long，并传递零值）

注解
并非每种操作系统都支持对每种设备进行异步操作。Windows 95 不支持对一个磁盘文件进行异步读操作（重复读）

Top
ReadFileEx
ReadFileEx

VB 声明
Declare Function ReadFileEx Lib "kernel32" Alias "ReadFileEx" (ByVal hFile As Long, lpBuffer As Any, ByVal nNumberOfBytesToRead As Long, lpOverlapped As OVERLAPPED, ByVal lpCompletionRoutine As Long) As Long
说明
与 ReadFile 相似，只是它只能用于异步读操作，并包含了一个完整的回调
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hFileLong，文件的句柄
lpBufferAny，指定容纳读入数据的一个缓冲区。除非读操作执行完毕，否则不要访问这个缓冲区
nNumberOfBytesToReadLong，要读入的字节数
lpOverlappedOVERLAPPED，定义了一个异步操作的结构。使用这个函数时，结构中的 hEvent 字段会被忽略
lpCompletionRoutineLong，回调函数的返回值

Top
RegCloseKey
RegCloseKey

VB 声明
Declare Function RegCloseKey Lib "advapi32.dll" Alias "RegCloseKey" (ByVal hKey As Long) As Long
说明
关闭系统注册表中的一个项（或键）
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，要关闭的项

Top
RegConnectRegistry
RegConnectRegistry

VB 声明
Declare Function RegConnectRegistry Lib "advapi32.dll" Alias "RegConnectRegistryA" (ByVal lpMachineName As String, ByVal hKey As Long, phkResult As Long) As Long
说明
访问远程系统的部分注册表
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
lpMachineNameString，欲连接的系统。采用“\\计算机名”的形式
hKeyLong，HKEY_LOCAL_MACHINE 或 HKEY_USERS
phkResultLong，用于装载指定项句柄的一个变量

Top
RegCreateKey
RegCreateKey

VB 声明
Declare Function RegCreateKey Lib "advapi32.dll" Alias "RegCreateKeyA" (ByVal hKey As Long, ByVal lpSubKey As String, phkResult As Long) As Long
说明
在指定的项下创建一个新项。如指定的项已经存在，那么函数会打开现有的项
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，要打开项的句柄，或者一个标准项名
lpSubKeyString，欲创建的新子项。可同时创建多个项，只需用反斜杠将它们分隔开即可。例如 level1\level2\newkey
phkResultLong，指定一个变量，用于装载新子项的句柄

Top
RegCreateKeyEx
RegCreateKeyEx

VB 声明
Declare Function RegCreateKeyEx Lib "advapi32.dll" Alias "RegCreateKeyExA" (ByVal hKey As Long, ByVal lpSubKey As String, ByVal Reserved As Long, ByVal lpClass As String, ByVal dwOptions As Long, ByVal samDesired As Long, lpSecurityAttributes As SECURITY_ATTRIBUTES, phkResult As Long, lpdwDisposition As Long) As Long
说明
在指定项下创建新项的更复杂的方式。在 Win32 环境中建议使用这个函数。如指定的项已经存在，则函数会打开现有的项

返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，一个打开项的句柄，或者一个标准项名
lpSubKeyString，欲创建的新子项的名字
ReservedLong，设为零
lpClassString，项的类名
dwOptionsLong，下述常数为零：REG_OPTION_VOLATILE——这个项不正式保存下来，系统重新启动后会消失
samDesiredLong，带有前缀 KEY_??的一个或多个常数。它们组合起来描述了允许对这个项进行哪些操作
lpSecurityAttributesSECURITY_ATTRIBUTES，对这个项的安全特性进行描述的一个结构（用 ByVal As Long 传递空值）。不适用于 windows 95
phkResultLong，指定用于装载新子项句柄的一个变量
lpdwDispositionLong，用于装载下列某个常数的一个变量：
REG_CREATED_NEW_KEY——新建的一个子项
REG_OPENED_EXISTING_KEY——打开一个现有的项
注解
REG_OPTION_VOLATILE 不适用于 windows 95

Top
RegDeleteKey
RegDeleteKey

VB 声明
Declare Function RegDeleteKey Lib "advapi32.dll" Alias "RegDeleteKeyA" (ByVal hKey As Long, ByVal lpSubKey As String) As Long
说明
删除现有项下方一个指定的子项
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，一个已打开项的句柄，或者标准项名之一
lpSubKeyString，要删除项的名字。这个项的所有子项也会删除

Top
RegDeleteValue
RegDeleteValue

VB 声明
Declare Function RegDeleteValue Lib "advapi32.dll" Alias "RegDeleteValueA" (ByVal hKey As Long, ByVal lpValueName As String) As Long

说明
删除指定项下方的一个值
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，一个已打开项的句柄，或标准项名之一
lpValueNameString，要删除的值名。可设为 vbNullString 或一个空串，表示删除那个项的默认值

Top
RegEnumKey
RegEnumKey

VB 声明
Declare Function RegEnumKey Lib "advapi32.dll" Alias "RegEnumKeyA" (ByVal hKey As Long, ByVal dwIndex As Long, ByVal lpName As String, ByVal cbName As Long) As Long
说明
枚举指定项的子项。在 Win32 环境中应使用 RegEnumKeyEx
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，一个已打开项的句柄，或者指定一个标准项名
dwIndexLong，欲获取的子项的索引。第一个子项的索引编号为零
lpNameString，用于装载指定索引处项名的一个缓冲区
cbNameLong，lpName 缓冲区的长度
注解
用 RegQueryInfoKey 判断容纳最长那个项所需的缓冲区长度

Top
RegEnumKeyEx
RegEnumKeyEx

VB 声明
Declare Function RegEnumKeyEx Lib "advapi32.dll" Alias "RegEnumKeyExA" (ByVal hKey As Long, ByVal dwIndex As Long, ByVal lpName As String, lpcbName As Long, lpReserved As Long, ByVal lpClass As String, lpcbClass As Long, lpftLastWriteTime As FILETIME) As Long
说明
枚举指定项下方的子项
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明

hKeyLong，一个已打开项的句柄，或者指定一个标准项名
dwIndexLong，欲获取的子项的索引。第一个子项的索引编号为零
lpNameString，用于装载指定索引处项名的一个缓冲区
lpcbNameLong，指定一个变量，用于装载 lpName 缓冲区的实际长度（包括空字符）。一旦返回，它会设为实际装载到 lpName 缓冲区的字符数量
lpReservedLong，未用，设为零
lpClassString，项使用的类名。可以为 vbNullString
lpcbClassLong，用于装载 lpClass 缓冲区长度的一个变量。一旦返回，它会设为实际装载到缓冲区的字符数量
lpftLastWriteTimeFILETIME，枚举子项上一次修改的时间

Top
RegEnumValue
RegEnumValue

VB 声明
Declare Function RegEnumValue Lib "advapi32.dll" Alias "RegEnumValueA" (ByVal hKey As Long, ByVal dwIndex As Long, ByVal lpValueName As String, lpcbValueName As Long, lpReserved As Long, lpType As Long, lpData As Byte, lpcbData As Long) As Long
说明
枚举指定项的值
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，一个已打开项的句柄，或者指定一个标准项名
dwIndexLong，欲获取值的索引。注意第一个值的索引编号为零
lpValueNameString，用于装载位于指定索引处值名的一个缓冲区
lpcbValueNameLong，用于装载 lpValueName 缓冲区长度的一个变量。一旦返回，它会设为实际载入缓冲区的字符数量
lpReservedLong，未用；设为零
lpTypeLong，用于装载值的类型代码的变量
lpDataByte，用于装载值数据的一个缓冲区
lpcbDataLong，用于装载 lpData 缓冲区长度的一个变量。一旦返回，它会设为实际载入缓冲区的字符数量

Top
RegFlushKey
RegFlushKey

VB 声明
Declare Function RegFlushKey Lib "advapi32.dll" Alias "RegFlushKey" (ByVal hKey As Long) As Long
说明

将对项和它的子项作出的改动实际写入磁盘
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，欲刷新的一个项的句柄，或指定一个标准项名
注解
有些操作系统会将对注册表的修改延迟写入磁盘，以便保持系统的高性能。这个函数的作用就是确定将数据实际写入磁盘。但通常，应尽量避免使用这个函数，因为它可能严重影响一个应用程序的性能

Top
RegGetKeySecurity
RegGetKeySecurity

VB 声明
Declare Function RegGetKeySecurity Lib "advapi32.dll" Alias "RegGetKeySecurity" (ByVal hKey As Long, ByVal SecurityInformation As Long, pSecurityDescriptor As SECURITY_DESCRIPTOR, lpcbSecurityDescriptor As Long) As Long
说明
获取与一个注册表项有关的安全信息
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，指定一个项的句柄，或指定一个标准项名
SecurityInformationLong，对请求信息进行描述的一个标记
pSecurityDescriptorSECURITY_DESCRIPTOR，这个结构用于装载目标项的安全信息
lpcbSecurityDescriptorLong，用于装载 pSecurityDescriptor 缓冲区长度的一个变量。一旦返回，它会设为实际装载到缓冲区的字节数量
适用平台
Windows NT

Top
RegLoadKey
RegLoadKey

VB 声明
Declare Function RegLoadKey Lib "advapi32.dll" Alias "RegLoadKeyA" (ByVal hKey As Long, ByVal lpSubKey As String, ByVal lpFile As String) As Long
说明
从以前用 RegSaveKey 函数创建的一个文件里装载注册表信息
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码

参数表
参数类型及说明
hKeyLong，HKEY_LOCAL_MACHINE、HKEY_USERS 或者用 RegConnectRegistry 创建的一个子项
lpSubKeyString，要创建的新子项的名字
lpFileString，包含了注册信息的那个文件的名字

Top
RegNotifyChangeKeyValue
RegNotifyChangeKeyValue

VB 声明
Declare Function RegNotifyChangeKeyValue Lib "advapi32.dll" Alias "RegNotifyChangeKeyValue" (ByVal hKey As Long, ByVal bWatchSubtree As Long, ByVal dwNotifyFilter As Long, ByVal hEvent As Long, ByVal fAsynchronus As Long) As Long
说明
注册表项或它的任何一个子项发生变化时，用这个函数提供一种通知机制
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，要监视的一个项的句柄，或者指定一个标准项名
bWatchSubtreeLong，TRUE（非零）表示监视子项以及指定的项
dwNotifyFilterLong，下述常数的一个或多个
REG_NOTIFY_CHANGE_NAME 侦测注册表项名称的变化，以及侦测注册表的创建和删除事件
REG_NOTIFY_CHANGE_ATTRIBUTES 侦测属性的变化
REG_NOTIFY_CHANGE_LAST_SET 侦测上一次修改时间的变化
REG_NOTIFY_CHANGE_SECURITY 侦测对安全特性的改动
hEventLong，一个事件的句柄。如 fAsynchronus 为 False，则这里的设置会被忽略
fAsynchronusLong，如果为零，那么除非侦测到一个变化，否则函数不会返回。否则这个函数会立即返回，而且在发生变化时触发由 hEvent 参数指定的一个事件
适用平台
Windows NT

Top
RegOpenKey
RegOpenKey

VB 声明
Declare Function RegOpenKey Lib "advapi32.dll" Alias "RegOpenKeyA" (ByVal hKey As Long, ByVal lpSubKey As String, phkResult As Long) As Long
说明
打开一个现有的注册表项

返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，一个已打开项的句柄，或指定一个标准项名
lpSubKeyString，要打开的项名
phkResultLong，指定一个变量，用于装载（保存）打开注册表项的一个句柄
注解
在 NT 环境下，这个函数会使用默认的安全参数

Top
RegOpenKeyEx
RegOpenKeyEx

VB 声明
Declare Function RegOpenKeyEx Lib "advapi32.dll" Alias "RegOpenKeyExA" (ByVal hKey As Long, ByVal lpSubKey As String, ByVal ulOptions As Long, ByVal samDesired As Long, phkResult As Long) As Long
说明
打开一个现有的项。在 win32 下推荐使用这个函数
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，一个已打开项的句柄，或指定一个标准项名
lpSubKeyString，欲打开注册表项的名字
ulOptionsLong，未用，设为零
samDesiredLong，带有前缀 KEY_??的一个或多个常数。它们的组合描述了允许对这个项进行哪些操作
phkResultLong，用于装载打开项的名字的一个变量

Top
RegQueryInfoKey
RegQueryInfoKey

VB 声明
Declare Function RegQueryInfoKey Lib "advapi32.dll" Alias "RegQueryInfoKeyA" (ByVal hKey As Long, ByVal lpClass As String, lpcbClass As Long, lpReserved As Long, lpcSubKeys As Long, lpcbMaxSubKeyLen As Long, lpcbMaxClassLen As Long, lpcValues As Long, lpcbMaxValueNameLen As Long, lpcbMaxValueLen As Long, lpcbSecurityDescriptor As Long, lpftLastWriteTime As FILETIME) As Long
说明
获取与一个项有关的信息
返回值

Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码。如一个缓冲区的长度不够，不能容下返回的数据，则函数会返回 ERROR_MORE_DATA
参数表
参数类型及说明
hKeyLong，一个已打开项的句柄，或指定一个标准项名
lpClassString，指定一个字串，用于装载这个注册表项的类名
lpcbClassLong，指定一个变量，用于装载 lpClass 缓冲区的长度。一旦返回，它会设为实际装载到缓冲区的字节数量
lpReservedLong，未用，设为零
lpcSubKeysLong，用于装载（保存）这个项的子项数量的一个变量
lpcbMaxSubKeyLenLong，指定一个变量，用于装载这个项最长一个子项的长度。注意这个长度不包括空中止字符
lpcbMaxClassLenLong，指定一个变量，用于装载这个项之子项的最长一个类名的长度。注意这个长度不包括空中止字符
lpcValuesLong，用于装载这个项的设置值数量的一个变量
lpcbMaxValueNameLenLong，指定一个变量，用于装载这个项之子项的最长一个值名的长度。注意这个长度不包括空中止字符
lpcbMaxValueLenLong，指定一个变量，用于装载容下这个项最长一个值数据所需的缓冲区长度
lpcbSecurityDescriptorLong，装载值安全描述符长度的一个变量
lpftLastWriteTimeFILETIME，指定一个结构，用于容纳该项的上一次修改时间

Top
RegQueryValue
RegQueryValue

VB 声明
Declare Function RegQueryValue Lib "advapi32.dll" Alias "RegQueryValueA" (ByVal hKey As Long, ByVal lpSubKey As String, ByVal lpValue As String, lpcbValue As Long) As Long
说明
取得指定项或子项的默认（未命名）值
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，一个已打开项的句柄，或者指定一个标准项名
lpSubKeyString，要获取一个值的子项。可设为 vbNullString，表示获取 hKey 的值
lpValueString，用于容纳指定项值的一个字串
lpcbValueLong，指定一个变量，用于装载 lpValue 缓冲区的长度。一旦返回，它会设为实际载入缓冲区的字节数量
注解
win32 应用程序应该使用 RegQueryValueEx。lpValue 被定义成一个字串，以维持同 win16 的兼容性（在 win16 中，值全都是字串）

Top
RegQueryValueEx
RegQueryValueEx

VB 声明
Declare Function RegQueryValueEx Lib "advapi32.dll" Alias "RegQueryValueExA" (ByVal hKey As Long, ByVal lpValueName As String, ByVal lpReserved As Long, lpType As Long, lpData As Any, lpcbData As Long) As Long
说明
获取一个项的设置值
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，一个已打开项的句柄，或者指定一个标准项名
lpValueNameString，要获取值的名字
lpReservedLong，未用，设为零
lpTypeLong，用于装载取回数据类型的一个变量
lpDataAny，用于装载指定值的一个缓冲区
lpcbDataLong，用于装载 lpData 缓冲区长度的一个变量。一旦返回，它会设为实际装载到缓冲区的字节数

Top
RegReplaceKey
RegReplaceKey

VB 声明
Declare Function RegReplaceKey Lib "advapi32.dll" Alias "RegReplaceKeyA" (ByVal hKey As Long, ByVal lpSubKey As String, ByVal lpNewFile As String, ByVal lpOldFile As String) As Long
说明
用一个磁盘文件保存的信息替换注册表信息；并创建一个备份，在其中包含当前注册表信息
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，一个已打开项的句柄，或指定一个标准项名
lpSubKeyString，要替换的子项名称。它必须直接位于 HKEY_LOCAL_MACHINE 或 HKEY_USERS 控制项的下方
lpNewFileString，包含了注册表信息的一个文件的名字。这个文件是用 RegSaveKey 函数创建的
lpOldFileString，对当前注册表信息进行备份的一个文件的名字

Top

RegRestoreKey
RegRestoreKey

VB 声明
Declare Function RegRestoreKey Lib "advapi32.dll" Alias "RegRestoreKeyA" (ByVal hKey As Long, ByVal lpFile As String, ByVal dwFlags As Long) As Long
说明
从一个磁盘文件恢复注册表信息
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，一个已打开项的句柄，或者指定一个标准项名
lpFileString，要从中恢复注册表信息的一个文件的名字
dwFlagsLong，0 表示进行常规恢复。REG_WHOLE_HIVE_VOLATILE 表示临时恢复信息（系统重新启动时不保存下来）。在这种情况下，hKey 必须引用 HKEY_LOCAL_MACHINE 或 HKEY_USERS

Top
RegSaveKey
RegSaveKey

VB 声明
Declare Function RegSaveKey Lib "advapi32.dll" Alias "RegSaveKeyA" (ByVal hKey As Long, ByVal lpFile As String, lpSecurityAttributes As SECURITY_ATTRIBUTES) As Long
说明
将一个项以及它的所有子项都保存到一个磁盘文件
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，一个已打开项的句柄，或指定一个标准项名
lpFileString，要在其中保存注册表信息的一个磁盘文件的名字
lpSecurityAttributesSECURITY_ATTRIBUTES，为保存的信息提供的安全信息。可设为 NULL，表示采用默认的安全信息（变成 ByVal As Long，并传递零值）

Top
RegSetKeySecurity
RegSetKeySecurity

VB 声明
Declare Function RegSetKeySecurity Lib "advapi32.dll" Alias "RegSetKeySecurity" (ByVal hKey As Long, ByVal SecurityInformation As Long, pSecurityDescriptor As SECURITY_DESCRIPTOR) As Long

说明
设置指定项的安全特性
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，指定一个项的句柄，或指定一个标准项名
SecurityInformationLong，对要保存的信息进行描述的标志
pSecurityDescriptorSECURITY_DESCRIPTOR，这个结构包含了注册表项新的安全特性设置
适用平台
Windows NT

Top
RegSetValue
RegSetValue

VB 声明
Declare Function RegSetValue Lib "advapi32.dll" Alias "RegSetValueA" (ByVal hKey As Long, ByVal lpSubKey As String, ByVal dwType As Long, ByVal lpData As String, ByVal cbData As Long) As Long
说明
设置指定项或子项的默认值
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，一个已打开项的句柄，或指定一个标准项名
lpSubKeyString，欲对它的值进行设置的一个子项的名字。如指定 vbNullString，表示设置 hKey 的默认值。如指定的子项不存在，则会创建它
dwTypeLong，必须是 REG_SZ
lpDataString，新值
cbDataLong，指定 lpData 的长度，不包括空中止字符

Top
RegSetValueEx
RegSetValueEx

VB 声明
Declare Function RegSetValueEx Lib "advapi32.dll" Alias "RegSetValueExA" (ByVal hKey As Long, ByVal lpValueName As String, ByVal Reserved As Long, ByVal dwType As Long, lpData As Any, ByVal cbData As Long) As Long
说明
设置指定项的值
返回值

Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，一个已打开项的句柄，或指定一个标准项名
lpValueNameString，要设置值的名字
ReservedLong，未用，设为零
dwTypeLong，要设置的数量类型
lpDataAny，包含数据的缓冲区中的第一个字节
cbDataLong，lpData 缓冲区的长度

Top
RegUnLoadKey
RegUnLoadKey

VB 声明
Declare Function RegUnLoadKey Lib "advapi32.dll" Alias "RegUnLoadKeyA" (ByVal hKey As Long, ByVal lpSubKey As String) As Long
说明
卸载指定的项以及它的所有子项
返回值
Long，零（ERROR_SUCCESS）表示成功。其他任何值都代表一个错误代码
参数表
参数类型及说明
hKeyLong，HKEY_LOCAL_MACHINE、HKEY_USERS 或者用 RegConnectRegistry 打开的一个子项
lpSubKeyString，要卸载的子项的名字。必须是早先用 RegLoadKey 函数载入的

Top
RemoveDirectory
RemoveDirectory

VB 声明
Declare Function RemoveDirectory Lib "kernel32" Alias "RemoveDirectoryA" (ByVal lpPathName As String) As Long
说明
删除指定目录
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpPathNameString，要删除的那个目录的名字
注解
在调用这个函数前，目录必须为空

Top
SearchPath
SearchPath

VB 声明
Declare Function SearchPath Lib "kernel32" Alias "SearchPathA" (ByVal lpPath As String, ByVal lpFileName As String, ByVal lpExtension As String, ByVal nBufferLength As Long, ByVal lpBuffer As String, ByVal lpFilePart As String) As Long
说明
查找指定文件
返回值
Long，装载到 lpBuffer 缓冲区的字符数。如缓冲区长度不足，则返回缓冲区必要的长度。零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpPathString，欲搜索的路径。如果为 vbNullString，则采用 windows 搜索路径。参考 OpenFile 函数的 OFSTRUCT 结构中对 OF_SEARCH 标志搜索顺序的介绍
lpFileNameString，要查找的文件名
lpExtensionString，文件扩展名。必须用一个句点符号起头。如文件没有扩展名，或者 lpFileName 包括了扩展名，则设为 vbNullString
nBufferLengthLong，lpBuffer 字串的长度
lpBufferString，用于装载文件名的一个字串
lpFilePartString，指定一个长整数变量，用于装载缓冲文件名部分的地址。在 vb 中不是特别有用
注解
参考 GetFullPathName 函数

Top
SetCurrentDirectory
SetCurrentDirectory

VB 声明
Declare Function SetCurrentDirectory Lib "kernel32" Alias "SetCurrentDirectoryA" (ByVal lpPathName As String) As Long
说明
设置当前目录
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpPathNameString，新当前目录的路径

Top
SetEndOfFile

SetEndOfFile

VB 声明
Declare Function SetEndOfFile Lib "kernel32" Alias "SetEndOfFile" (ByVal hFile As Long) As Long
说明
针对一个打开的文件，将当前文件位置设为文件末尾
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hFileLong，指定一个文件句柄。文件的当前位置设为文件尾，文件会根据需要缩短
注解
如一个文件正作为打开文件映射对象的基准使用，则不要对其应用这个函数

Top
SetFileAttributes
SetFileAttributes

VB 声明
Declare Function SetFileAttributes Lib "kernel32" Alias "SetFileAttributesA" (ByVal lpFileName As String, ByVal dwFileAttributes As Long) As Long
说明
设置文件属性
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpFileNameString，要设置其属性的文件名
dwFileAttributesLong，带有 FILE_ATTRIBUTE_??前缀的一个或多个常数

Top
SetFilePointer
SetFilePointer

VB 声明
Declare Function SetFilePointer Lib "kernel32" Alias "SetFilePointer" (ByVal hFile As Long, ByVal lDistanceToMove As Long, lpDistanceToMoveHigh As Long, ByVal dwMoveMethod As Long) As Long
说明
在一个文件中设置当前的读写位置
返回值
Long，返回一个新位置，它采用从文件起始处开始算起的一个字节偏移量。HFILE_ERROR 意味着出错。会设置 GetLastError

参数表
参数类型及说明
hFileLong，系统文件句柄
lDistanceToMoveLong，字节偏移量
lpDistanceToMoveHighLong，指定一个长整数变量，其中包含了要使用的一个高双字偏移。可设为零（将声明变为 ByVal），表示只使用 lDistanceToMove
原文： A long variable containing a high double word offset to use. May be zero (change declaration to ByVal) to use only lDistanceToMove.
dwMoveMethodLong，下述常数之一
FILE_BEGINlOffset 将新位置设为从文件起始处开始算的起的一个偏移
FILE_CURRENTlOffset 将新位置设为从当前位置开始计算的一个偏移
FILE_ENDlOffset 将新位置设为从文件尾开始计算的一个偏移
注解
这个函数与 vb 的 seek 语句类似。不要将函数用于通过 vb 的 open 命令打开的文件。利用这个函数，可以处理那些长度大于 2 ^ 64 字节的大型文件

Top
SetFileTime
SetFileTime

VB 声明
Declare Function SetFileTime Lib "kernel32" Alias "SetFileTime" (ByVal hFile As Long, lpCreationTime As FILETIME, lpLastAccessTime As FILETIME, lpLastWriteTime As FILETIME) As Long
说明
设置文件的创建、访问及上次修改时间
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hFileLong，系统文件句柄
lpCreationTimeFILETIME，文件的创建时间
lpLastAccessTimeFILETIME，文件上一次访问的时间
lpLastWriteTimeFILETIME，文件最近一次修改的时间

Top
SetHandleCount
SetHandleCount

VB 声明
Declare Function SetHandleCount Lib "kernel32" Alias "SetHandleCount" (ByVal wNumber As Long) As Long
说明
这个函数不必在 win32 下使用；即使使用，也不会有任何效果

This function is not necessary under Win32 and has no effect.

Top
SetVolumeLabel
SetVolumeLabel

VB 声明
Declare Function SetVolumeLabel Lib "kernel32" Alias "SetVolumeLabelA" (ByVal lpRootPathName As String, ByVal lpVolumeName As String) As Long
说明
设置一个磁盘的卷标（Label）
返回值
Long，TRUE（非零）表示成功，否则返回零。会设置 GetLastError
参数表
参数类型及说明
lpRootPathNameString，磁盘卷的根路径
lpVolumeNameString，指定新卷标。用 vbNullString 指示删除当前卷名

Top
SystemTimeToFileTime
SystemTimeToFileTime

VB 声明
Declare Function SystemTimeToFileTime Lib "kernel32" Alias "SystemTimeToFileTime" (lpSystemTime As SYSTEMTIME, lpFileTime As FILETIME) As Long
说明
根据一个 FILETIME 结构的内容，载入一个 SYSTEMTIME 结构
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpSystemTimeSYSTEMTIME，包含了系统时间信息的一个结构
lpFileTimeFILETIME，用于装载文件时间的一个结构

Top
UnlockFile
UnlockFile

VB 声明
Declare Function UnlockFile Lib "kernel32" Alias "UnlockFile" (ByVal hFile As Long, ByVal dwFileOffsetLow As Long, ByVal dwFileOffsetHigh As Long, ByVal nNumberOfBytesToUnlockLow As Long, ByVal nNumberOfBytesToUnlockHigh As Long) As Long
说明

解除对一个文件的锁定
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hFileLong，要解锁的文件的句柄
dwFileOffsetLowLong，要解锁文件区域起始位置的低 32 位地址
dwFileOffsetHighLong，要解锁文件区域起始位置的高 32 位地址
nNumberOfBytesToUnlockLowLong，锁定区域中字符数量的低 32 位值
nNumberOfBytesToUnlockHighLong，锁定区域中字符数量的高 32 位值
注解
解锁的文件区域必须与以前锁定时设置的区域完全相符。文件关闭前，应用程序应确定已解除了对任何区域的锁定。参考 LockFile 了解进一步的情况

Top
UnlockFileEx
UnlockFileEx

VB 声明
Declare Function UnlockFileEx Lib "kernel32" Alias "UnlockFileEx" (ByVal hFile As Long, ByVal dwReserved As Long, ByVal nNumberOfBytesToUnlockLow As Long, ByVal nNumberOfBytesToUnlockHigh As Long, lpOverlapped As OVERLAPPED) As Long
说明
解除对一个文件的锁定
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hFileLong，要解锁的文件的句柄
dwReservedLong，未用；设为零
nNumberOfBytesToUnlockLowLong，锁定区域中字符数量的低 32 位值
nNumberOfBytesToUnlockHighLong，锁定区域中字符数量的高 32 位值
lpOverlappedOVERLAPPED，包含了文件中相对于解锁区域起始处的偏移量
注解
解锁的文件区域必须与以前锁定时设置的区域完全相符。文件关闭前，应用程序应确定已解除了对任何区域的锁定。参考 LockFileEx 了解进一步的情况

Top
UnmapViewOfFile
UnmapViewOfFile

VB 声明
Declare Function UnmapViewOfFile& Lib "kernel32" (ByVal lpBaseAddress As Long)
说明

在当前应用程序的内存地址空间解除对一个文件映射对象的映射
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpBaseAddressLong，指定要解除映射的一个文件映射的基准地址。这个地址是早先用 MapViewOfFile 函数获得的
注解
在 vb 的 api 文本查看器里复制的声明如下，请注意参数类型不同
Declare Function UnmapViewOfFile Lib "kernel32" Alias "UnmapViewOfFile" (lpBaseAddress As Any) As Long

Top
VerFindFile
VerFindFile

VB 声明
Declare Function VerFindFile Lib "version.dll" Alias "VerFindFileA" (ByVal uFlags As Long, ByVal szFileName As String, ByVal szWinDir As String, ByVal szAppDir As String, ByVal szCurDir As String, lpuCurDirLen As Long, ByVal szDestDir As String, lpuDestDirLen As Long) As Long
说明
用这个函数决定一个文件应安装到哪里
返回值
Long，下述值之一：
VFF_CURNEDEST 指出文件现有版本不应在由 szDestDir 参数指定的目录中，那个目录是由函数建议安装新版本的地方
VFF_FILEINUSE 指出现有文件当时正在使用，而且不要在此时删除
VFF_BUFFTOOSMALL 指出 szDestDir 或 szCurDir 缓冲区的一个或两个都太小，不足以容下目录名
参数表
参数类型及说明
uFlagsLong，目前只定义了 VFFF_ISSHAREDFILE，它指出文件可由多个应用程序共享。如指定了这个标志，该函数会建议将文件安装到 windows 或系统目录。如这个参数为零，则函数会建议将文件安装到应用程序目录
szFileNameString，要安装的文件名。注意这个字串不应包括文件的路径
szWinDirString，设为 windows 目录。目录名称是用 GetWindowsDirectory 函数取得的
szAppDirString，应用程序以及所有相关文件的安装目录的完整路径名称
szCurDirString，指定一个字串缓冲区，用于容纳包含了文件现有版本的目录。如文件版本不存在，则在缓冲区中载入源文件的目录。注意必须为这个缓冲区至少分配 MAX_PATH 个字符的空间
lpuCurDirLenLong，szCurDir 缓冲区的长度。这个函数会设为实际装载到缓冲区的字符数量
szDestDirString，指定一个缓冲区，用于装载应在其中安装新文件的一个目录名。注意至少要为这个缓冲区分配 MAX_PATH 个字符的空间

lpuDestDirLenLong，szDestDir 缓冲区的长度。这个变量会设为实际装载到缓冲区的字符数量

Top
VerInstallFile
VerInstallFile

VB 声明
Declare Function VerInstallFile Lib "version.dll" Alias " VerInstallFileA" (ByVal uFlags As Long, ByVal szSrcFileName As String, ByVal szDestFileName As String, ByVal szSrcDir As String, ByVal szDestDir As String, ByVal szCurDir As String, ByVal szTmpFile As String, lpuTmpFileLen As Long) As Long
说明
用这个函数安装一个文件。它利用由 VerFindFile 函数提供的信息决定将文件安装到哪里。这个函数首先会比较两个文件的版本标记。如源文件是最新和兼容的版本，则将源文件复制成目标目录的一个临时文件——如文件处于压缩状态，则同时将其解压。随后，将文件的现有版本删除掉，再对临时文件进行重名处理，使符合目标文件名
返回值
Long，返回一个整数，其中包含了 VerInstallFile 结果常数表里列出的一个或多个常数的组合
参数表
参数类型及说明
uFlagsLong，下述常数值的一个组合：
VIFF_FORCEINSTALL 在不进行版本检查的情况下强制安装源文件
VIFF_DONTDELETEOLD 如文件的现有版本不在目标目录，则不将其删除；如果它在目标目录，就用新文件将其改写（覆盖）
szSrcFileNameString，指定要安装文件的名字。注意其中不应包含文件的路径名
szDestFileNameString，指定文件安装好后应得到的一个正式名称。这个名称与 szSrcFileName 通常都是相同的
szSrcDirString，指定源目录。新版文件将从这里复制到目标目录
szDestDirString，指定目标目录。新版文件将从源目录复制到这里。通常为这个参数使用由 VerFindFile 函数返回的 szDestDir 缓冲区
szCurDirString，包含了文件当前版本的一个目录。通常将由 VerFindFile 函数返回的 szCurDir 缓冲区用于这个参数。如字串为空，则表明系统中不存在文件文件的早期版本
szTmpFileString，用于装载源文件一个临时副本名称的缓冲区。注意必须至少为其分配 MAX_PATH 个字符的空间
lpuTmpFileLenLong，szTmpFile 缓冲区的长度。这个变量会设为装载到缓冲区的实际字符数，其中包括中止用的 NULL 字符。如指定了 VIFF_FORCEINSTALL，且 szTmpFile 不为零，则临时文件会被更名为由 szSrcFileName 参数指定的名字

Top
VerLanguageName
VerLanguageName

VB 声明

Declare Function VerLanguageName Lib "kernel32" Alias "VerLanguageNameA" (ByVal wLang As Long, ByVal szLang As String, ByVal nSize As Long) As Long
说明
这个函数能根据 16 位语言代码获取一种语言的名称。利用版本资源中的语言代码，可以判断出一个文件编写时采用的语言格式。表格“win32 支持的语言代码”对 win32 支持的各种语言代码进行了总结
返回值
Long，装载到 szLang 缓冲区的字符数量。如缓冲区的容量不够，不能容下完整的名称，则函数返回需要的缓冲区大小。零表示出错
参数表
参数类型及说明
wLangLong，语言 ID
szLangString，用于装载指定语言文本名称的一个缓冲区。这个缓冲区至少应预初始化成 nSize+1 个字节的长度
nSizeLong，szLang 缓冲区的大小

Top
VerQueryValue
VerQueryValue

VB 声明
Declare Function VerQueryValue& Lib "version.dll" Alias "VerQueryValueA" (pBlock As Byte, ByVal lpSubBlock As String, lplpBuffer As Long, puLen As Long)
说明
这个函数用于从版本资源中获取信息。调用这个函数前，必须先用 GetFileVersionInfo 函数获取版本资源信息。这个函数会检查资源信息，并将需要的数据复制到一个缓冲区里
返回值
Long，TRUE（非零）表示成功，如请求的信息不存在，或 pBlock 不属于有效版本信息，那就返回一个零
参数表
参数类型及说明
pBlockByte，指定一个内存块第一个字节的地址。这个内存块包含了由 GetFileVersionInfo 函数取回的版本数据信息
lpSubBlockString，下述值之一：
"\"获取文件的 VS_FIXEDFILEINFO 结构
"\VarFileInfo\Translation"获取文件的翻译表
"\StringFileInfo\...."获取文件的字串信息。参考注解
lplpBufferLong，指定一个 Long 变量的地址，该变量用于装载一个缓冲区的地址。请求的版本信息最终会装载到那个缓冲区里
puLenLong，指定由 lplpBuffer 参数引用的数据值的长度，以字节为单位
注解
如 lplpBuffer 参数为"\StringFileInfo\....”，缓冲区里就会载入一个整数数组。每一对整数都代表一种语言和代码页，它们描绘了可用的字串信息。通过用下面这三个部分指定一个字串，从而获得 StringFileInfo 字串数据："\StringFileInfo\languagecodepage\stringname"，其中

languagecodepage（语言代码页）是采用字串形式的一个 8 字符十六进制数字。如翻译表中的语言代码页条目是&H04090000，那么这个字串就应该是"04090000"。stringname（字串名）指定的是一个字串名。这个参数的一个例子如下：
"\StringFileInfo\04090000\CompanyName"
其他
从 vb 的 api 文本查看器复制的声明如下：
Declare Function VerQueryValue Lib "version.dll" Alias "VerQueryValue" (pBlock As Any, ByVal lpSubBlock As String, ByVal lplpBuffer As Long, puLen As Long) As Long

Top
WriteFile
WriteFile

VB 声明
Declare Function WriteFile Lib "kernel32" Alias "WriteFile" (ByVal hFile As Long, lpBuffer As Any, ByVal nNumberOfBytesToWrite As Long, lpNumberOfBytesWritten As Long, lpOverlapped As OVERLAPPED) As Long
说明
将数据写入一个文件。该函数比 lwrite 函数要灵活的多。也可将这个函数应用于对通信设备、管道、套接字以及邮槽的处理
返回值
Long，TRUE（非零）表示成功，否则返回零。会设置 GetLastError
参数表
参数类型及说明
hFileLong，一个文件的句柄
lpBufferAny，要写入的一个数据缓冲区
nNumberOfBytesToWriteLong，要写入数据的字节数量。如写入零字节，表示什么都不写入，但会更新文件的“上一次修改时间”。针对位于远程系统的命名管道，限制在 65535 个字节以内
lpNumberOfBytesWrittenLong，实际写入文件的字节数量
lpOverlappedOVERLAPPED，倘若在指定 FILE_FLAG_OVERLAPPED 的前提下打开文件，这个参数就必须引用一个特殊的结构。那个结构定义了一次异步写操作。否则，该参数应置为空（将声明变为 ByVal As Long，并传递零值）
注解
并不是每种操作系统都支持在任何类型的设备上进行异步操作。windows 95 不支持对磁盘文件的重叠读取操作

Top
WriteFileEx
WriteFileEx

VB 声明
Declare Function WriteFileEx Lib "kernel32" Alias "WriteFileEx" (ByVal hFile As Long, lpBuffer As Any, ByVal nNumberOfBytesToWrite As Long, lpOverlapped As OVERLAPPED, ByVal

lpCompletionRoutine As Long) As Long
说明
与 WriteFile 类似，只是它只能用于异步写操作，并包括了一个完整的回调
返回值
Long，非零表示成功，否则返回零。会设置 GetLastError
参数表
参数类型及说明
hFileLong，文件的句柄
lpBufferAny，指定一个缓冲区，其中包含了要写入的数据。除非写操作完成，否则不要访问这个缓冲区
nNumberOfBytesToWriteLong，要写入数据的字节量
lpOverlappedOVERLAPPED，定义了一次异步写操作的结构。使用这个函数时，结构中的 hEvent 字段会被忽略
lpCompletionRoutineLong，回调函数的值
注解
并不是每种操作系统都支持在任何类型的设备上进行异步操作。windows 95 不支持对磁盘文件的重叠读取操作

Top
WritePrivateProfileSection
WritePrivateProfileSection

VB 声明
Declare Function WritePrivateProfileSection Lib "kernel32" Alias "WritePrivateProfileSectionA" (ByVal lpAppName As String, ByVal lpString As String, ByVal lpFileName As String) As Long
说明
为一个初始化文件（.ini）中指定的小节设置所有项名和值
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpAppNameString，要设置的小节。这个字串不区分大小写
lpStringString，项和值字串的一个列表。每个字串都用一个 NULL 字符分隔，最后一个字串后面用两个 NULL 表示中止。如 lpAppName 指定的小节不存在，则用那个名字新建一个小节，并将其追加到初始化文件的最后。如果存在，则当前的所有项名和值都会被这个缓冲区中指定的数据取代
lpFileNameString，初始化文件的名字。如指定了一个完整路径，而且文件不存在，就会产生错误。如只指定了文件名，而且文件不存在，就在当前的 windows 目录中创建它

Top
WritePrivateProfileString
WritePrivateProfileString

VB 声明

Declare Function WritePrivateProfileString& Lib "kernel32" Alias "WritePrivateProfileStringA" (ByVal lpApplicationName As String, ByVal lpKeyName As Any, ByVal lpString As String, ByVal lpFileName As String)
说明
在初始化文件指定小节内设置一个字串
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpApplicationNameString，要在其中写入新字串的小节名称。这个字串不区分大小写
lpKeyNameAny，要设置的项名或条目名。这个字串不区分大小写。用 vbNullString 可删除这个小节的所有设置项
lpStringString，指定为这个项写入的字串值。用 vbNullString 表示删除这个项现有的字串
lpFileNameString，初始化文件的名字。如果没有指定完整路径名，则 windows 会在 windows 目录查找文件。如果文件没有找到，则函数会创建它
其他
在 vb 的 api 文本查看器里复制的声明如下：
Declare Function WritePrivateProfileString Lib "kernel32" Alias "WritePrivateProfileStringA" (ByVal lpApplicationName As String, ByVal lpKeyName As Any, ByVal lpString As Any, ByVal lpFileName As String) As Long

Top
WriteProfileSection
WriteProfileSection

VB 声明
Declare Function WriteProfileSection Lib "kernel32" Alias "WriteProfileSectionA" (ByVal lpAppName As String, ByVal lpString As String) As Long
说明
为 Win.ini 初始化文件中一个指定的小节设置所有项名和值
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpAppNameString，要设置的小节。这个字串不区分大小写
lpStringString，项和值字串的一个列表。每个字串都用一个 NULL 字符分隔，最后一个字串后面用两个 NULL 表示中止。如 lpAppName 指定的小节不存在，则用那个名字新建一个小节，并将其追加到初始化文件的最后。如果存在，则当前的所有项名和值都会被这个缓冲区中指定的数据取代
注解
注意对 Win.ini 文件的改动可能影响其他应用程序。如修改了正由其他应用程序使用的小节，一定要向所有窗口都发送一条 WM_WININICHANGE 消息

Top

WriteProfileString
WriteProfileString

VB 声明
Declare Function WriteProfileString Lib "kernel32" Alias "WriteProfileStringA" (ByVal lpszSection As String, ByVal lpszKeyName As String, ByVal lpszString As String) As Long
说明
在 Win.ini 初始化文件指定小节内设置一个字串
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpszSectionString，指定要在其中写入新串的小节。如尚不存在，会创建这个小节。这个字串不区分大小写
lpszKeyNameString，要设置的项名或条目名。这个字串不区分大小写。用 vbNullString 可删除这个小节的所有设置项
lpszStringString，指定为这个项写入的字串值。用 vbNullString 表示删除这个项现有的字串
注解
注意对 Win.ini 文件的改动可能影响其他应用程序。如修改了正由其他应用程序使用的小节，一定要向所有窗口都发送一条 WM_WININICHANGE 消息

Top
网络函数
网络函数，共一页。第一页
WNetAddConnection 创建同一个网络资源的永久性连接
WNetAddConnection2 创建同一个网络资源的连接
WNetAddConnection3 创建同一个网络资源的连接
WNetCancelConnection 结束一个网络连接
WNetCancelConnection2 结束一个网络连接
WNetCloseEnum 结束一次枚举操作
WNetConnectionDialog 启动一个标准对话框，以便建立同网络资源的连接
WNetDisconnectDialog 启动一个标准对话框，以便断开同网络资源的连接
WNetEnumResource 枚举网络资源
WNetGetConnection 获取本地或已连接的一个资源的网络名称
WNetGetLastError 获取网络错误的扩展错误信息
WNetGetUniversalName 获取网络中一个文件的远程名称以及/或者 UNC（统一命名规范）名称
WNetGetUser 获取一个网络资源用以连接的名字
WNetOpenEnum 启动对网络资源进行枚举的过程
完
WNetAddConnection
WNetAddConnection

VB 声明

Declare Function WNetAddConnection Lib "mpr.dll" Alias "WNetAddConnectionA" (ByVal lpszNetPath As String, ByVal lpszPassword As String, ByVal lpszLocalName As String) As Long
说明
创建同一个网络资源的永久性连接
返回值
Long，零表示成功。会设置 GetLastError。如 GetLastError 是 ERROR_EXTENDED_ERROR，则可用 WNetGetLastError 取得额外的错误信息
参数表
参数类型及说明
lpszNetPathString，要连接的网络名
lpszPasswordString，可选的一个密码。如为 vbNullString，表示采用当前用户的默认密码。如为一个空字串，则不用任何密码
lpszLocalNameString，资源的本地名称。（例如，F: 和 LPT1:）

Top
WNetAddConnection2
WNetAddConnection2

VB 声明
Declare Function WNetAddConnection2 Lib "mpr.dll" Alias "WNetAddConnection2A" (lpNetResource As NETRESOURCE, ByVal lpPassword As String, ByVal lpUserName As String, ByVal dwFlags As Long) As Long
说明
创建同一个网络资源的连接
返回值
Long，零表示成功。会设置 GetLastError。如 GetLastError 是 ERROR_EXTENDED_ERROR，则可用 WNetGetLastError 取得额外的错误信息
参数表
参数类型及说明
lpNetResourceNETRESOURCE，在这个结构中设置了下述字段，对要连接的网络资源进行了定义：dwType， lpLocalName （可为 vbNullString）， lpRemoteName， lpProvider （设为 vbNullString 表示用默认提供者）。该结构的其他所有变量都会被忽略
lpPasswordString，可选的一个密码。如为 vbNullString，表示采用当前用户的默认密码。如为一个空字串，则不用任何密码
lpUserNameString，用于连接的用户名。如为 vbNullString，表示使用当前用户
dwFlagsLong，设为零；或指定常数 CONNECT_UPDATE_PROFILE，表示创建永久性连接

Top
WNetAddConnection3
WNetAddConnection3

VB 声明
Declare Function WNetAddConnection3& Lib "mpr.dll" Alias "WNetAddConnection3A" (ByVal hwnd As Long, lpNetResource As NETRESOURCE, ByVal lpPassword As String, ByVal

lpUserName As String, ByVal dwFlags As Long)
说明
创建同一个网络资源的连接。这个函数与 WNetAddConnection2 类似，只是它允许我们为这个函数显示的对话框指定一个物主窗口
返回值
Long，
参数表
参数类型及说明
hwndLong，指定一个窗口句柄，用作本函数创建的对话框的父窗口
lpNetResourceNETRESOURCE，在这个结构中设置了下述字段，对要连接的网络资源进行了定义：dwType， lpLocalName （可为 vbNullString）， lpRemoteName， lpProvider （设为 vbNullString 表示用默认提供者）。该结构的其他所有变量都会被忽略
lpPasswordString，可选的一个密码。如为 vbNullString，表示采用当前用户的默认密码。如为一个空字串，则不用任何密码
lpUserNameString，用于连接的用户名。如为 vbNullString，表示使用当前用户
dwFlagsLong，设为零；或指定常数 CONNECT_UPDATE_PROFILE，表示创建永久性连接

Top
WNetCancelConnection
WNetCancelConnection

VB 声明
Declare Function WNetCancelConnection Lib "mpr.dll" Alias "WNetCancelConnectionA" (ByVal lpszName As String, ByVal bForce As Long) As Long
说明
结束一个网络连接
返回值
Long，零表示成功。会设置 GetLastError。如 GetLastError 是 ERROR_EXTENDED_ERROR，则可用 WNetGetLastError 取得额外的错误信息
参数表
参数类型及说明
lpszNameString，已连接资源的远程名称或本地名称
bForceLong，如为 TRUE，表示断开连接（即使连接的资源上正有打开的文件或作业）

Top
WNetCancelConnection2
WNetCancelConnection2

VB 声明
Declare Function WNetCancelConnection2 Lib "mpr.dll" Alias "WNetCancelConnection2A" (ByVal lpName As String, ByVal dwFlags As Long, ByVal fForce As Long) As Long
说明
结束一个网络连接
返回值

Long，零表示成功。会设置 GetLastError。如 GetLastError 是 ERROR_EXTENDED_ERROR，则可用 WNetGetLastError 取得额外的错误信息
参数表
参数类型及说明
lpszNameString，已连接资源的远程名称或本地名称
dwFlagsLong，设为零或 CONNECT_UPDATE_PROFILE。如为零，而且建立的是永久性连接，则在 windows 下次重新启动时仍会重新连接
fForceLong，如为 TRUE，表示强制断开连接（即使连接的资源上正有打开的文件或作业）

Top
WNetCloseEnum
WNetCloseEnum

VB 声明
Declare Function WNetCloseEnum Lib "mpr.dll" Alias "WNetCloseEnum" (ByVal hEnum As Long) As Long
说明
结束一次枚举操作
返回值
Long，零表示成功。会设置 GetLastError。如 GetLastError 是 ERROR_EXTENDED_ERROR，则可用 WNetGetLastError 取得额外的错误信息
参数表
参数类型及说明
hEnumLong，由 WNetOpenEnum 函数返回的一个枚举句柄

Top
WNetConnectionDialog
WNetConnectionDialog

VB 声明
Declare Function WNetConnectionDialog Lib "mpr.dll" Alias "WNetConnectionDialog" (ByVal hwnd As Long, ByVal dwType As Long) As Long
说明
启动一个标准对话框，以便建立同网络资源的连接
返回值
Long，零表示成功。如用户取消了操作，则返回-1。会设置 GetLastError。如 GetLastError 是 ERROR_EXTENDED_ERROR，则可用 WNetGetLastError 取得额外的错误信息
参数表
参数类型及说明
hwndLong，指定要成为对话框父窗口的一个窗口的句柄
dwTypeLong，设成 RESOURCETYPE_DISK，浏览磁盘资源

Top
WNetDisconnectDialog

WNetDisconnectDialog

VB 声明
Declare Function WNetDisconnectDialog Lib "mpr.dll" Alias "WNetDisconnectDialog" (ByVal hwnd As Long, ByVal dwType As Long) As Long
说明
启动一个标准对话框，以便断开同网络资源的连接
返回值
Long，零表示成功。如用户取消了操作，则返回-1。会设置 GetLastError。如 GetLastError 是 ERROR_EXTENDED_ERROR，则可用 WNetGetLastError 取得额外的错误信息
参数表
参数类型及说明
hwndLong，指定要成为对话框父窗口的一个窗口的句柄
dwTypeLong，设成 RESOURCETYPE_DISK 或 RESOURCETYPE_PRINT，决定要断开的是磁盘还是打印机资源

Top
WNetEnumResource
WNetEnumResource

VB 声明
Declare Function WNetEnumResource Lib "mpr.dll" Alias "WNetEnumResourceA" (ByVal hEnum As Long, lpcCount As Long, lpBuffer As Any, lpBufferSize As Long) As Long
说明
枚举网络资源
返回值
Long，零表示成功。ERROR_NO_MORE_ITEMS 表示不剩下可以枚举的条目。ERROR_MORE_DATA 表示条目不能装入 lpBuffer。会设置 GetLastError。如 GetLastError 是 ERROR_EXTENDED_ERROR，则可用 WNetGetLastError 取得额外的错误信息
参数表
参数类型及说明
hEnumLong，从 WNetOpenEnum 函数返回的一个句柄
lpcCountLong，最初设为要枚举的最大资源数量；或设为-1，表示枚举尽可能多的资源。一旦返回，就会设为实际枚举的资源数量
lpBufferAny，通常是一个字节缓冲区的首字节。该缓冲区装载了枚举信息（可按引用声明为 Byte）
lpBufferSizeLong，以字节为单位指定 lpBuffer 数组的长度。如缓冲区不够大，则设为需要的缓冲区长度
注解
枚举网络条目时，最好用 vb 一次枚举一个资源。尽量不要使用这个函数同时枚举许多网络资源的功能

Top
WNetGetConnection

WNetGetConnection

VB 声明
Declare Function WNetGetConnection Lib "mpr.dll" Alias "WNetGetConnectionA" (ByVal lpszLocalName As String, ByVal lpszRemoteName As String, cbRemoteName As Long) As Long
说明
获取本地或已连接的一个资源的网络名称
返回值
Long，零表示成功。会设置 GetLastError。如 GetLastError 是 ERROR_EXTENDED_ERROR，则可用 WNetGetLastError 取得额外的错误信息
参数表
参数类型及说明
lpszLocalNameString，本地设备的名字
lpszRemoteNameString，指定一个字串缓冲区，用于装载设备的资源名称
cbRemoteNameLong，lpszRemoteName 缓冲区的字符数量。如缓冲区不够大，则设为需要的缓冲区长度

Top
WNetGetLastError
WNetGetLastError

VB 声明
Declare Function WNetGetLastError Lib "mpr.dll" Alias "WNetGetLastErrorA" (lpError As Long, ByVal lpErrorBuf As String, ByVal nErrorBufSize As Long, ByVal lpNameBuf As String, ByVal nNameBufSize As Long) As Long
说明
获取网络错误的扩展错误信息
返回值
Long，零表示成功。ERROR_INVALID_ADDRESS 表示缓冲区无效
参数表
参数类型及说明
lpErrorLong，指定一个变量，用于装载网络错误代码。具体的代码由网络供应商决定
lpErrorBufString，指定一个字串缓冲区，用于装载网络错误的说明
nErrorBufSizeLong，lpErrorBuf 缓冲区包含的字符数量
lpNameBufString，用于装载网络供应商名字的字串缓冲区
nNameBufSizeLong，lpNameBuf 缓冲区的字符数量

Top
WNetGetUniversalName
WNetGetUniversalName

VB 声明
Declare Function WNetGetUniversalName Lib "mpr" Alias "WNetGetUniversalNameA" (ByVal lpLocalPath As String, ByVal dwInfoLevel As Long, lpBuffer As Any, lpBufferSize As Long) As

Long
说明
获取网络中一个文件的远程名称以及/或者 UNC（统一命名规范）名称。例如，假设一个已连接的远程驱动器是\\othersystem\CDrive，它对应的本地驱动器是 F:，而且在它的子目录 temp 中包含了文件 xyz.doc。那么运算结果如下：LocalPath xyz.doc 或 f:\temp\xyz.doc（或者文件的任何相对路径名）
UNC 名称： \\othersystem\CDrive\temp\xyz.doc
连接名称： \\othersystem\CDrive
剩余名称： \temp\xyz.doc
它们分别对应于由这个函数装载的 REMOTE_NAME_INFO 结构的字段，对该结构的定义如下：

```
Type REMOTE_NAME_INFO
    pUniversalName As Long
    pConnectionName As Long
    pRemainingPath As Long
End Type
```

返回值
Long，零表示成功。会设置 GetLastError。如 GetLastError 是 ERROR_EXTENDED_ERROR，则可用 WNetGetLastError 取得额外的错误信息
参数表
参数类型及说明
lpLocalPathString，磁盘文件的名字
dwInfoLevelLong，下述常数之一：
UNIVERSAL_NAME_INFO_LEVEL 只设置 pUniversalName 字段
REMOTE_NAME_INFO_LEVEL 设置 REMOTE_NAME_INFO 结构中的所有三个字段
lpBufferAny，指定用于装载 UNC 信息的一个缓冲区。缓冲区起点与一个 REMOTE_NAME_INFO 结构对应
lpBufferSizeLong，以字节为单位指定 lpBuffer 缓冲区的长度。如缓冲区不够大，则设为需要的缓冲区长度

Top
WNetGetUser
WNetGetUser

VB 声明

```
Declare Function WNetGetUser Lib "mpr.dll" Alias "WNetGetUserA" (ByVal lpName As String, ByVal lpUserName As String, lpnLength As Long) As Long
```

说明
获取一个网络资源用以连接的名字
返回值
Long，零表示成功。会设置 GetLastError。如 GetLastError 是 ERROR_EXTENDED_ERROR，则可用 WNetGetLastError 取得额外的错误信息
参数表
参数类型及说明

lpNameString，指定已连接资源的远程名称或本地名称。用 vbNullString 获取当前用户的名字
lpUserNameString，用于装载用户名的一个字串缓冲区
lpnLengthLong，lpUserName 缓冲区的长度。如缓冲区不够大，则自动设为需要的缓冲区长度

Top
WNetOpenEnum
WNetOpenEnum

VB 声明
Declare Function WNetOpenEnum Lib "mpr.dll" Alias "WNetOpenEnumA" (ByVal dwScope As Long, ByVal dwType As Long, ByVal dwUsage As Long, lpNetResource As NETRESOURCE, lphEnum As Long) As Long
说明
启动对网络资源进行枚举的过程。这个函数会返回由 WNetEnumResource 函数用于枚举资源所用的一个句柄
返回值
Long，零表示成功。会设置 GetLastError。如 GetLastError 是 ERROR_EXTENDED_ERROR，则可用 WNetGetLastError 取得额外的错误信息
参数表
参数类型及说明
dwScopeLong，指定要枚举的资源范围。可设为下述常数之一：
RESOURCE_CONNECTED 枚举已连接的资源（忽略 dwUsage）
RESOURCE_GLOBALNET 枚举所有资源
RESOURCE_REMEMBERED 只枚举永久性连接
dwTypeLong，下述常数之一
RESOURCE_ANY 枚举所有类型的网络资源
RESOURCE_DISK 枚举磁盘资源
RESOURCE_PRINT 枚举打印资源
dwUsageLong，可设为零，表示枚举所有资源；或设为下述常数的一个或两个：
RESOURCEUSAGE_CONNECTABLE 只枚举那些能够连接的资源
RESOURCEUSAGE_CONTAINER 只枚举包含了其他资源的资源
lpNetResourceNETRESOURCE，这个结构指定了一个容器资源。该函数会枚举包含于这里指定的某个指定资源内的资源。如设为 NULL（把声明变成 ByVal As Long），那么函数会枚举顶级网络资源。倘若在 dwScope 参数里没有指定 RESOURCE_GLOBALNET，那么必须为 NULL
lphEnumLong，指定一个变量，用于装载一个枚举句柄。该句柄由 WNetEnumResource 函数使用。必须用 WNetCloseEnum 函数将其清除

Top
菜单函数
菜单函数：共三页。第一页，第二页，第三页
AppendMenu 在指定的菜单里添加一个菜单项

CheckMenuItem 复选或撤消复选指定的菜单条目
CheckMenuRadioItem 指定一个菜单条目被复选成“单选”项目
CreateMenu 创建新菜单
CreatePopupMenu 创建一个空的弹出式菜单
DeleteMenu 删除指定的菜单条目
DestroyMenu 删除指定的菜单
DrawMenuBar 为指定的窗口重画菜单
EnableMenuItem 允许或禁止指定的菜单条目
GetMenu 取得窗口中一个菜单的句柄
GetMenuCheckMarkDimensions 返回一个菜单复选符的大小
GetMenuContextHelpId 取得一个菜单的帮助场景 ID
GetMenuDefaultItem 判断菜单中的哪个条目是默认条目
GetMenuItemCount 返回菜单中条目（菜单项）的数量
GetMenuItemID 返回位于菜单中指定位置处的条目的菜单 ID
GetMenuItemInfo 取得（接收）与一个菜单条目有关的特定信息
AppendMenu
AppendMenu

VB 声明
Declare Function AppendMenu Lib "user32" Alias "AppendMenuA" (ByVal hMenu As Long, ByVal wFlags As Long, ByVal wIDNewItem As Long, ByVal lpNewItem As Any) As Long
说明
在指定的菜单里添加一个菜单项
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hMenuLong，菜单句柄
wFlagsLong，参考 ModifyMenu 函数中的菜单常数标志定义表，其中列出了允许使用的所有常数
wIDNewItemLong，指定菜单条目的新命令 ID。如果在 wFlags 参数中指定了 MF_POPUP 字段，那么这应该是指向一个弹出式菜单的句柄
lpNewItemString（相应的 vb 声明见注解），如果在 wFlags 参数中指定了 MF_STRING 标志，这就代表在菜单中设置的字串。如设置了 MF_BITMAP 标志，这就代表一个 Long 型变量，其中包含了一个位图句柄。如设置了 MF_OWNERDRAW，这个值就会包括在 DRAWITEMSTRUCT 和 MEASUREITEMSTRUCT 结构中，在条目需要重画的时候由 windows 发送出去
注解
Declare Function AppendMenu& Lib "user32" Alias "AppendMenuA" (ByVal hMenu As Long, ByVal wFlags As Long, ByVal wIDNewItem As Long, ByVal lpNewItem As String)

Top
CheckMenuItem
CheckMenuItem

VB 声明
Declare Function CheckMenuItem Lib "user32" Alias "CheckMenuItem" (ByVal hMenu As Long, ByVal wIDCheckItem As Long, ByVal wCheck As Long) As Long
说明
复选或撤消复选指定的菜单条目
返回值
Long，如条目的前一个状态是“复选”，就返回 MF_CHECKED，如果是“未复选”，就返回 MF_UNCHECKED。如指定的菜单条目不存在就返回-1
参数表
参数类型及说明
hMenuLong，菜单句柄
wIDCheckItemLong，欲复选或撤消复选的菜单条目的标识符。如果在 wCheck 中指定了 MF_BYCOMMAND 标志，这个参数就用于指定要改变的菜单条目的命令 ID。如果设置了 MF_BYPOSITION 标志，这个参数就用于指定条目在菜单中的位置（第一个条目的位置是 0）
wCheckLong，参考 ModifyMenu 函数中的菜单常数标志定义表，其中列出了允许使用的所有常数。针对这个函数，只能指定下述常数：MF_BYCOMMAND，MF_BYPOSITION，MF_CHECKED 以及 MF_UNCHECKED
注解
在 vb 里使用：由这个函数做出的改动可以正常发挥作用，但不会由 vb 菜单的 checked 属性反映出来

Top
CheckMenuRadioItem
CheckMenuRadioItem

VB 声明
Declare Function CheckMenuRadioItem Lib "user32" Alias "CheckMenuRadioItem" (ByVal hMenu As Long, ByVal un1 As Long, ByVal un2 As Long, ByVal un3 As Long, ByVal un4 As Long) As Long
说明
指定一个菜单条目被复选成“单选”项目。与单选钮相似，一个特定的组中只能有一个项目被选中。这个组别既可按位置定义，也可按菜单 ID 定义。复选的项目会显示一个圆形的样式复选符号（●），而不是一个标准的复选符号（√）
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hMenuLong，菜单句柄
un1Long，组内第一个位置或菜单 ID
un2Long，组内最后一个位置或菜单 ID
un3Long，欲复选的位置或菜单 ID
un4Long，下述标志之一 ：如 un1，un2，un3 引用菜单条目的位置（第一个肯定在位置 0 处），就设为 MF_BYPOSITION；如它们引用的是菜单 ID，则设为 MF_BYCOMMAND

注解
在 vb 里使用：由这个函数做出的改动可以正常发挥作用，但不会由 vb 菜单的 checked 属性反映出来

Top
CreateMenu
CreateMenu

VB 声明
Declare Function CreateMenu Lib "user32" Alias "CreateMenu" () As Long
说明
创建新菜单
返回值
Long，如成功则返回新的顶级菜单的句柄；零意味着错误
注解
最开始创建时，菜单是空的。可用菜单 api 函数插入菜单条目。一旦菜单不再需要，记住用 DestroyMenu 将其删除

Top
CreatePopupMenu
CreatePopupMenu

VB 声明
Declare Function CreatePopupMenu Lib "user32" Alias "CreatePopupMenu" () As Long
说明
创建一个空的弹出式菜单。可用 AppendMenu 或 InsertMenu 函数在窗口中添加条目，或者为一个现成的菜单添加弹出式菜单，并在新建的菜单中添加条目
返回值
Long，如成功，返回一个菜单句柄；零意味着错误
注解
并不推荐用这个函数来创建备用的 vb 菜单，除非是为 TrackPopupMenu 函数生成菜单。这个窗口中使用的命令 ID 必须与现有 vb 菜单控件的 ID 相符。或者用一个子类处理控件进行管理

Top
DeleteMenu
DeleteMenu

VB 声明
Declare Function DeleteMenu Lib "user32" Alias "DeleteMenu" (ByVal hMenu As Long, ByVal nPosition As Long, ByVal wFlags As Long) As Long
说明
删除指定的菜单条目（在 vb 里使用：强烈建议用 vb 菜单的 visible 属性从菜单中删除条目。如使用这个函数，会造成指定菜单其他菜单条目的 visible 属性错误的影响菜单条目）

返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hMenuLong，菜单句柄
nPositionLong，欲删除菜单条目的标识符。如在 wFlags 中设置了 MF_BYCOMMAND 标志，这个参数就代表要改变的菜单条目的命令 ID。如设置了 MF_BYPOSITION 标志，这个参数就代表条目在菜单中的位置（头一个条目肯定是零）
wFlagsLong，MF_BYPOSITION 或 MF_BYCOMMAND，具体由 nPosition 参数决定
注解
如条目连接了一个弹出式菜单，就会清除弹出式菜单。用 RemoveMenu 函数清除一个弹出式菜单条目，同时不影响整个弹出式菜单

Top
DestroyMenu
DestroyMenu

VB 声明
Declare Function DestroyMenu Lib "user32" Alias "DestroyMenu" (ByVal hMenu As Long) As Long
说明
删除指定的菜单。如菜单属于另一个菜单的一部分，或直接分配给一个窗口，那么菜单会在窗口清除后被自动删除
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hMenuLong，欲删除菜单的句柄
注解
这个函数通常用于 CreateMenu 和 CreatePopupMenu 函数创建的菜单

Top
DrawMenuBar
DrawMenuBar

VB 声明
Declare Function DrawMenuBar Lib "user32" Alias "DrawMenuBar" (ByVal hwnd As Long) As Long
说明
为指定的窗口重画菜单。用 api 函数改变一个窗口菜单的内容时，就要用到这个函数
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明

hwndLong，要重画菜单栏的窗口的句柄
注解
在 vb 里很少有必要用到这个函数，因为不应该用 api 函数来改变一个窗口的顶级菜单栏

Top
EnableMenuItem
EnableMenuItem

VB 声明
Declare Function EnableMenuItem Lib "user32" Alias "EnableMenuItem" (ByVal hMenu As Long, ByVal wIDEnableItem As Long, ByVal wEnable As Long) As Long
说明
允许或禁止指定的菜单条目（在 vb 里使用：由这个函数做出的改动可以正常发挥作用，但不会由 vb 菜单的 enabled 属性反映出来）
返回值
Long，
参数表
参数类型及说明
hMenuLong，菜单句柄
wIDEnableItemLong，欲允许或禁止的一个菜单条目的标识符。如果在 wEnable 参数中设置了 MF_BYCOMMAND 标志，这个参数就代表欲改变菜单条目的命令 ID。如设置的是 MF_BYPOSITION，则这个参数代表菜单条目在菜单中的位置（第一个条目肯定是零）
wEnableLong，参考 ModifyMenu 函数中的菜单常数标志定义表，其中列出了允许使用的所有常数。对于这个函数，只能指定下述常数：MF_BYCOMMAND，MF_BYPOSITION，MF_ENABLED，MF_DISABLED 以及 MF_GRAYED
注解
如指定的菜单条目依附了一个弹出式菜单，那么整个弹出式菜单都会受到影响

Top
GetMenu
GetMenu

VB 声明
Declare Function GetMenu Lib "user32" Alias "GetMenu" (ByVal hwnd As Long) As Long
说明
取得窗口中一个菜单的句柄
返回值
Long，依附于指定窗口的一个菜单的句柄（如果有菜单）；否则返回零
参数表
参数类型及说明
hwndLong，窗口句柄。对于 vb，这应该是一个窗体句柄。注意可能不是子窗口的句柄

Top
GetMenuCheckMarkDimensions

GetMenuCheckMarkDimensions

VB 声明
Declare Function GetMenuCheckMarkDimensions Lib "user32" Alias "GetMenuCheckMarkDimensions" () As Long
说明
返回一个菜单复选符的大小。参考 SetMenuItemBitmaps 以进一步了解如何使用这个函数
返回值
Long，高字（高 16 位）指定菜单复选符的高度，以像素为单位表示；低字代表宽度

Top
GetMenuContextHelpId
GetMenuContextHelpId

VB 声明
Declare Function GetMenuContextHelpId Lib "user32" Alias "GetMenuContextHelpId" (ByVal hMenu As Long) As Long
说明
取得用 SetMenuContextHelpId 函数分配给一个菜单的帮助场景 ID
返回值
Long，如果存在，就返回帮助场景 ID；否则返回零
参数表
参数类型及说明
hMenuLong，菜单句柄

Top
GetMenuDefaultItem
GetMenuDefaultItem

VB 声明
Declare Function GetMenuDefaultItem Lib "user32" Alias "GetMenuDefaultItem" (ByVal hMenu As Long, ByVal fByPos As Long, ByVal gmdiFlags As Long) As Long
说明
允许我们判断菜单中的哪个条目是默认条目。这个条目相应的转换成双击菜单时执行的操作
返回值
Long，默认菜单条目的位置或标识符。如未发现默认条目，就返回-1
参数表
参数类型及说明
hMenuLong，菜单的句柄
fByPosLong，如果为 TRUE，表示接收条目在菜单中的位置；FALSE 表示接收它的菜单 ID
gmdiFlagsLong，下述标志之一：
GMDI_GOINTOPOPUPS 如默认条目是弹出菜单，这个函数就会在其中搜索一个默认的菜单条目
GMDI_USEDISABLED 指明自己在搜索过程中不想跳过被禁止的菜单条目

Top
GetMenuItemCount
GetMenuItemCount

VB 声明
Declare Function GetMenuItemCount Lib "user32" Alias "GetMenuItemCount" (ByVal hMenu As Long) As Long
说明
返回菜单中条目（菜单项）的数量
返回值
Long，菜单中的条目数量；-1 意味着出错。会设置 GetLastError
参数表
参数类型及说明
hMenuLong，目标菜单的句柄

Top
GetMenuItemID
GetMenuItemID

VB 声明
Declare Function GetMenuItemID Lib "user32" Alias "GetMenuItemID" (ByVal hMenu As Long, ByVal nPos As Long) As Long
说明
返回位于菜单中指定位置处的条目的菜单 ID
返回值
Long，指定条目的菜单 ID。如条目属于一个弹出式菜单，就返回-1；如指定的条目属于一个分隔符（比如一条分隔线）则返回 0
参数表
参数类型及说明
hMenuLong，菜单的句柄
nPosLong，条目在菜单中的位置。第一个条目的编号是 0

Top
GetMenuItemInfo
GetMenuItemInfo

VB 声明
Declare Function GetMenuItemInfo Lib "user32" Alias "GetMenuItemInfoA" (ByVal hMenu As Long, ByVal un As Long, ByVal b As Boolean, lpMenuItemInfo As MENUITEMINFO) As Long
说明
用一个 MENUITEMINFO 结构取得（接收）与一个菜单条目有关的特定信息
返回值
Long，TRUE（非零）表示成功，否则返回零。会设置 GetLastError

参数表
参数类型及说明
hMenuLong，菜单的句柄
unLong，菜单条目的菜单 ID 或位置
bBoolean，如 un 指定的是条目位置，就为 TRUE；如指定的是一个菜单 ID，则为 FALSE
lpMenuItemInfoMENUITEMINFO，这个结构用于装载请求的信息

Top
GetMenuItemRect
GetMenuItemRect

VB 声明
Declare Function GetMenuItemRect Lib "user32" Alias "GetMenuItemRect" (ByVal hWnd As Long, ByVal hMenu As Long, ByVal uItem As Long, lprcItem As RECT) As Long
说明
在一个矩形中装载指定菜单条目的屏幕坐标信息
返回值
Long，TRUE（非零）表示成功，否则返回零。会设置 GetLastError
参数表
参数类型及说明
hWndLong，包含指定菜单或弹出式菜单的一个窗口的句柄
hMenuLong，菜单的句柄
uItemLong，欲检查的菜单条目的位置或菜单 ID
lprcItemRECT，在这个结构中装载菜单条目的位置及大小（采用屏幕坐标表示）

Top
GetMenuState
GetMenuState

VB 声明
Declare Function GetMenuState Lib "user32" Alias "GetMenuState" (ByVal hMenu As Long, ByVal wID As Long, ByVal wFlags As Long) As Long
说明
取得与指定菜单条目状态有关的信息
返回值
Long，在 api32.txt 文件的常数定义的一系列标志的组合，请看下表。如条目是个弹出式菜单，那么结构的最低字节就包含了状态标志，而第二个字节包含条目在弹出式菜单中的数量
MF_HILITE 菜单条目加亮显示（处于选定状态）
MF_CHECKED 菜单条目处于复选状态
MF_DISABLED 菜单条目处于禁止状态
MF_GRAYED 菜单条目以灰色显示，处于禁用状态
MF_MENUBARBREAK 为这个条目指定一条分隔线。参考 ModifyMenu 函数
MF_MENUBREAK 为这个条目指定一个菜单分隔标志。参考 ModifyMenu 函数
MF_SEPARATOR 菜单条目是一个分隔符

参数表
参数类型及说明
hMenu 菜单句柄
wID 欲检查的菜单条目的标识符。如果在 wFlags 参数中设置了 MF_BYCOMMAND 标志，这个参数就用于指定要改变的菜单条目的命令 ID。如果设置的是 MF_BYPOSITION 标志，这个参数就用于指定条目在菜单中的位置（第一个条目的位置为 0）
wFlags 常数 MF_BYCOMMAND 或 MF_BYPOSITION，取决于 wID 参数的设置

Top
GetMenuString
GetMenuString

VB 声明
Declare Function GetMenuString Lib "user32" Alias "GetMenuStringA" (ByVal hMenu As Long, ByVal wIDItem As Long, ByVal lpString As String, ByVal nMaxCount As Long, ByVal wFlag As Long) As Long
说明
取得指定菜单条目的字串
返回值
Long，在 lpString 中返回的字串的长度（不包括空中止字符）。零意味着出错
参数表
参数类型及说明
hMenuLong，菜单句柄
wIDItemLong，欲接收的菜单条目的标识符。如果在 wFlags 参数中设置了 MF_BYCOMMAND 标志，这个参数就用于指定要改变的菜单条目的命令 ID。如果设置的是 MF_BYPOSITION 标志，这个参数就用于指定条目在菜单中的位置（第一个条目的位置为 0）
lpStringString，指定一个预先定义好的字串缓冲区，以便为菜单条目装载字串
nMaxCountLong，载入 lpString 缓冲区中的最大字符数量+1
wFlagLong，常数 MF_BYCOMMAND 或 MF_BYPOSITION，取决于 wID 参数的设置

Top
GetSubMenu
GetSubMenu

VB 声明
Declare Function GetSubMenu Lib "user32" Alias "GetSubMenu" (ByVal hMenu As Long, ByVal nPos As Long) As Long
说明
取得一个弹出式菜单的句柄，它位于菜单中指定的位置
返回值
Long，位于指定位置的弹出式菜单的句柄（如果有的话）；否则返回零
参数表
参数类型及说明

hMenuLong，菜单的句柄
nPosLong，条目在菜单中的位置。第一个条目的编号为 0

Top
GetSystemMenu
GetSystemMenu

VB 声明
Declare Function GetSystemMenu Lib "user32" Alias "GetSystemMenu" (ByVal hwnd As Long, ByVal bRevert As Long) As Long
说明
取得指定窗口的系统菜单的句柄。在 vb 环境，“系统菜单”的正式名称为“控制菜单”，即单击窗口左上角的控制框时出现的菜单
返回值
Long，如执行成功，返回系统菜单的句柄；零意味着出错。如 bRevert 设为 TRUE，也会返回零（简单的恢复原始的系统菜单）
参数表
参数类型及说明
hwndLong，窗口的句柄
bRevertLong，如设为 TRUE，表示接收原始的系统菜单
注解
在 vb 里使用：系统菜单会向窗口发送一条 WM_SYSCOMMAND 消息，而不是 WM_COMMAND 消息

Top
HiliteMenuItem
HiliteMenuItem

VB 声明
Declare Function HiliteMenuItem Lib "user32" Alias "HiliteMenuItem" (ByVal hwnd As Long, ByVal hMenu As Long, ByVal wIDHiliteItem As Long, ByVal wHilite As Long) As Long
说明
控制顶级菜单条目的加亮显示状态
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hwndLong，拥有顶级菜单的一个窗口的句柄
hMenuLong，hwnd 窗口的顶级菜单的句柄
wIDHiliteItemLong，欲加亮或撤消加亮的菜单条目的标识符。倘若在 wHilite 参数中设置了 MF_BYCOMMAND 标志，这个参数就用于指定要改变的菜单条目的命令 ID。如果设置的是 MF_BYPOSITION 标志，这个参数就用于指定条目在菜单中的位置（第一个条目的位置为 0）
wHiliteLong，一系列常数标志的组合。其中包括 MF_BYCOMMAND 或 MF_BYPOSITION，

指出要改变的条目。也包括 MF_HILITE，用于设置加亮状态；或者 MF_UNHILITE，用于撤消加亮状态

Top
InsertMenu
InsertMenu

VB 声明
Declare Function InsertMenu Lib "user32" Alias "InsertMenuA" (ByVal hMenu As Long, ByVal nPosition As Long, ByVal wFlags As Long, ByVal wIDNewItem As Long, ByVal lpNewItem As Any) As Long
说明
在菜单的指定位置处插入一个菜单条目，并根据需要将其他条目向下移动
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hMenuLong，菜单的句柄
nPositionLong，定义了新条目插入点的一个现有菜单条目的标志符。如果在 wFlags 中指定了 MF_BYCOMMAND 标志，这个参数就代表欲改变的菜单条目的命令 ID。如设置的是 MF_BYPOSITION 标志，这个参数就代表菜单条目在菜单中的位置，第一个条目的位置为零
wFlagsLong，一系列常数标志的组合。参考 ModifyMenu
wIDNewItemLong，指定菜单条目的新菜单 ID。如果在 wFlags 中指定了 MF_POPUP 标志，就应该指定弹出式菜单的一个句柄
lpNewItem 如果在 wFlags 参数中设置了 MF_STRING 标志，就代表要设置到菜单中的字串（String）。如设置的是 MF_BITMAP 标志，就代表一个 Long 型变量，其中包含了一个位图句柄
注解
在 vb 里使用：这个函数做出的许多改变都可以正常发挥作用，但却不能由 vb 菜单对象反映出来。添加的命令 ID 必须能由 vb 菜单系统识别

Top
InsertMenuItem
InsertMenuItem

VB 声明
Declare Function InsertMenuItem Lib "user32" Alias "InsertMenuItemA" (ByVal hMenu As Long, ByVal un As Long, ByVal bool As Boolean, ByVal lpcMenuItemInfo As MENUITEMINFO) As Long
说明
用一个 MENUITEMINFO 结构指定的特征插入一个新菜单条目
返回值
Long，TRUE（非零）表示成功，否则返回零。会设置 GetLastError

参数表
参数类型及说明
hMenuLong，菜单的句柄
unLong，菜单条目的菜单 ID。新条目会插入由这个参数指定的项目之前
boolBoolean，如 un 指定的是条目的位置，就为 TRUE，如指定的是菜单 ID，就为 FALSE
lpcMenuItemInfoMENUITEMINFO，用于设置指定菜单条目的特征
注解

Top
IsMenu
IsMenu

VB 声明
Declare Function IsMenu Lib "user32" Alias "IsMenu" (ByVal hMenu As Long) As Long
说明
判断指定的句柄是否为一个菜单的句柄
返回值
Long，如句柄是菜单句柄，就返回 TRUE；否则返回零
参数表
参数类型及说明
hMenuLong，欲测试的菜单的句柄

Top
LoadMenu
LoadMenu

VB 声明
Declare Function LoadMenu Lib "user32" Alias "LoadMenuA" (ByVal hInstance As Long, ByVal lpString As String) As Long
说明
从指定的模块或应用程序实例中载入一个菜单
返回值
Long，已装载的菜单的句柄；零意味着出错。会设置 GetLastError
参数表
参数类型及说明
hInstanceLong，一个动态链接库的模块句柄。或指定了特定可执行文件的一个实例句柄，那个可执行文件中包含了菜单资源
lpStringString，作为字串使用时，指定欲载入的菜单资源的名字；作为 Long 值使用时，指定欲载入的菜单 ID。
注解
在 vb 里使用：由于 vb 不能控制那些不与现有 vb 菜单兼容的菜单，所以不建议使用这个函数

Top
LoadMenuIndirect
LoadMenuIndirect

VB 声明
Declare Function LoadMenuIndirect Lib "user32" Alias "LoadMenuIndirectA" (ByVal lpMenuTemplate As Long) As Long
说明
在包含了 MENUITEMTEMPLATE 数据结构的一个内存块的基础上，载入一个菜单
返回值
Long，已载入的菜单句柄，零意味着出错。会设置 GetLastError
参数表
参数类型及说明
lpMenuTemplateLong，指向一个 MENUITEMTEMPLATEHEADER 结构的指针。在内存中，该结构的后面跟随有一个或多个 MENUITEMTEMPLATE 数据结构。在 windows 95 及 windows nt 4.0 中，可能是指向一个 MENUEX_HEADER_TEMPLATE 结构的指针；在那个结构后跟随有一个或多个 MENUEX_HEADER_ITEM 结构，用于指定扩展菜单

Top
MenuItemFromPoint
MenuItemFromPoint

VB 声明
Declare Function MenuItemFromPoint Lib "user32" Alias "MenuItemFromPoint" (ByVal hWnd As Long, ByVal hMenu As Long, ByVal ptScreen As POINTAPI) As Long
说明
判断哪个菜单条目包含了屏幕上一个指定的点
返回值
Long，包含了指定点的条目的位置。如果没有菜单条目包含了指定的点，就返回-1
参数表
参数类型及说明
hWndLong，包含了指定菜单的那个窗口的句柄
hMenuLong，菜单句柄
ptScreenPOINTAPI，点的位置。如 hMenu 是一个顶级菜单条，这个点就用 hWnd 窗口的窗口坐标表示。否则，它采用窗口的客户区坐标表示

Top
ModifyMenu
ModifyMenu,ModifyMenuBynum

VB 声明
Declare Function ModifyMenu& Lib "user32" Alias "ModifyMenuA" (ByVal hMenu As Long, ByVal nPosition As Long, ByVal wFlags As Long, ByVal wIDNewItem As Long, ByVal lpString As String)

Declare Function ModifyMenuBynum& Lib "user32" Alias "ModifyMenuA" (ByVal hMenu As Long, ByVal nPosition As Long, ByVal wFlags As Long, ByVal wIDNewItem As Long, ByVal lpString As Long)

说明

改变菜单条目。在 vb 里这个函数做出的许多改变都会有效的执行，但不能由 vb 菜单对象反映出来

返回值

Long，非零表示成功，零表示失败。会设置 GetLastError

参数表

参数类型及说明

hMenuLong，菜单句柄

nPositionLong，欲改变的菜单条目的标识符。如在 wFlags 参数中指定了 MF_BYCOMMAND，这个参数就代表欲改变的菜单条目的命令 ID。如设置的是 MF_BYPOSITION，这个参数就代表菜单条目在菜单中的位置（第一个条目的位置为零）

wFlagsLong，一系列常数标志的组合。详见菜单常数标志表

wIDNewItemLong，指定菜单条目的新命令 ID。如在 wFlags 参数中指定了 MF_POPUP 标志，就应是一个弹出式菜单的句柄

lpStringString 或 Long，如在 wFlags 参数中指定了 MF_STRING 标志，就代表欲设置到菜单的字串。如设置的是 MF_BITMAP，就代表一个 Long 变量，其中包含了一个位图句柄。如设置的是 MF_OWNERDRAW，那么这个值就会包括到 DRAWITEMSTRUCT 和 MEASUREITEMSTRUCT 结构中，并由 windows 在条目需要重画的时候发出

注解

标志的下述组合形式是不允许的：MF_BYCOMMAND 和 MF_BYPOSITION；MF_CHECKED 和 MF_UNCHECKED；MF_MENUBARBREAK 和 MF_MENUBREAK；MF_DISABLED，MF_ENABLED 和 MF_GRAYED；MF_BITMAP，MF_STRING，MF_OWNERDRAW 和 MF_SEPARATOR

菜单常数标志表

MF_BITMAP 菜单条目是一幅位图。一旦设入菜单，这幅位图就绝对不能删除。所以不应该使用由 vb 的 image 属性返回的值

MF_BYCOMMAND 菜单条目由菜单的命令 ID 指定

MF_BYPOSITION 菜单条目由条目在菜单中的位置决定。零代表菜单中的第一个条目

MF_CHECKED 检查指定的菜单条目。不能与 vb 的 checked 属性兼容

MF_DISABLED 禁止指定的菜单条目。不与 vb 的 enabled 属性兼容

MF_ENABLED 允许指定的菜单条目。不与 vb 的 enabled 属性兼容

MF_GRAYED 禁止指定的菜单条目，并用浅灰色描述它。不与 vb 的 enabled 属性兼容

MF_MENUBARBREAK 在弹出式菜单中，将指定的条目放置于一个新列，并用一条垂直线分隔不同的列

MF_MENUBREAK 在弹出式菜单中，将指定的条目放置于一个新列。在顶级菜单中，将条目放置到一个新行

MF_OWNERDRAW 创建一个物主绘图菜单（由您设计的程序负责描绘每个菜单条目）

MF_POPUP 将一个弹出式菜单置于指定的条目。可用于创建子菜单及弹出式菜单

MF_SEPARATOR 在指定的条目处显示一条分隔线

MF_STRING 在指定的条目处放置一个字串。不与 vb 的 caption 属性兼容

MF_UNCHECKED 检查指定的条目。不能与 vb 的 checked 属性兼容

Top
RemoveMenu
RemoveMenu

VB 声明
Declare Function RemoveMenu Lib "user32" Alias "RemoveMenu" (ByVal hMenu As Long, ByVal nPosition As Long, ByVal wFlags As Long) As Long
说明
删除指定的菜单条目。如删除的条目属于一个弹出式菜单，那么这个函数不会同时删除弹出式菜单。首先应该用 GetSubMenu 函数取得弹出式菜单的句柄，再在以后将其删除
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hMenuLong，菜单的句柄
nPositionLong，欲改变的菜单条目的标识符。如在 wFlags 参数中指定了 MF_BYCOMMAND，这个参数就代表欲改变的菜单条目的命令 ID。如设置的是 MF_BYPOSITION，这个参数就代表菜单条目在菜单中的位置（第一个条目的位置为零）
wFlagsLong，常数 MF_BYCOMMAND 或 MF_BYPOSITION，取决于 nPosition 参数
注解
强烈建议大家使用 vb 菜单的 visible 属性从菜单中删除条目，而不要用这个函数，否则会造成指定菜单中其他菜单条目的 visible 属性对错误的菜单条目产生影响

Top
SetMenu
SetMenu

VB 声明
Declare Function SetMenu Lib "user32" Alias "SetMenu" (ByVal hwnd As Long, ByVal hMenu As Long) As Long
说明
设置窗口菜单
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，窗口的句柄
hMenuLong，窗口的新菜单的句柄
注解
不建议在 vb 里使用这个函数。如坚持使用，务必留意新菜单中的命令 ID 并不兼容于原始的 vb 窗口。只有窗体窗口才应通过这个函数指定。窗口的前一个菜单不会由这个函数删除

Top

SetMenuContextHelpId
SetMenuContextHelpId

VB 声明
Declare Function SetMenuContextHelpId Lib "user32" Alias "SetMenuContextHelpId" (ByVal hMenu As Long, ByVal dw As Long) As Long
说明
设置用 SetMenuContextHelpId 函数分配给一个菜单的帮助场景 ID
返回值
Long，TRUE（非零）表示成功，否则返回零
参数表
参数类型及说明
hMenuLong，菜单的句柄
dwLong，要设置的帮助场景 ID
注解
用这个函数设置的场景 ID 并不对应于用 vb 菜单栏设计程序分配的场景 ID。vb 在内部保存有它自己的帮助场景 ID。vb 的帮助场景 ID 应是我们优先考虑的方案。尽可能不要用这个函数

Top
SetMenuDefaultItem
SetMenuDefaultItem

VB 声明
Declare Function SetMenuDefaultItem Lib "user32" Alias "SetMenuDefaultItem" (ByVal hMenu As Long, ByVal uItem As Long, ByVal fByPos As Long) As Long
说明
将一个菜单条目设为默认条目。这个条目会转换成双击菜单时执行的操作
返回值
Long，TRUE（非零）表示成功，否则返回零。会设置 GetLastError
参数表
参数类型及说明
hMenuLong，菜单句柄
uItemLong，欲设为默认菜单条目的一个条目的位置或菜单 ID。如果为-1，表示清除当前的默认条目
fByPosLong，如 uItem 是条目的位置，就为 TRUE；如果是菜单 ID，就为 FALSE

Top
SetMenuItemBitmaps
SetMenuItemBitmaps

VB 声明
Declare Function SetMenuItemBitmaps Lib "user32" Alias "SetMenuItemBitmaps" (ByVal hMenu As Long, ByVal nPosition As Long, ByVal wFlags As Long, ByVal hBitmapUnchecked As

Long, ByVal hBitmapChecked As Long) As Long
说明
设置一幅特定位图，令其在指定的菜单条目中使用，代替标准的复选符号（√）。位图的大小必须与菜单复选符号的正确大小相符，这个正确大小可以由 GetMenuCheckMarkDimensions 函数获得
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hMenuLong，菜单句柄
nPositionLong，欲设置位图的一个菜单条目的标识符。如在 wFlags 参数中指定了 MF_BYCOMMAND，这个参数就代表欲改变的菜单条目的命令 ID。如设置的是 MF_BYPOSITION，这个参数就代表菜单条目在菜单中的位置（第一个条目的位置为零）
wFlagsLong，常数 MF_BYCOMMAND 或 MF_BYPOSITION，取决于 nPosition 参数
hBitmapUncheckedLong，撤消复选时为菜单条目显示的一幅位图的句柄。如果为零，表示不在未复选状态下显示任何标志
hBitmapCheckedLong，复选时为菜单条目显示的一幅位图的句柄。可设为零，表示复选时不显示任何标志。如两个位图句柄的值都是零，则为这个条目恢复使用默认复选位图
注解
使用的位图可能由多个条目共享。一旦不再需要，位图必须由应用程序清除，因为 windows 不能自动对它进行清除

Top
SetMenuItemInfo
SetMenuItemInfo

VB 声明
Declare Function SetMenuItemInfo Lib "user32" Alias "SetMenuItemInfoA" (ByVal hMenu As Long, ByVal un As Long, ByVal bool As Boolean, lpcMenuItemInfo As MENUITEMINFO) As Long
说明
为一个菜单条目设置指定的信息，具体信息保存于 MENUITEMINFO 结构中
返回值
Long，TRUE（非零）表示成功，否则返回零。会设置 GetLastError
参数表
参数类型及说明
hMenuLong，菜单句柄
unLong，菜单条目的菜单 ID 或位置
boolBoolean，如 un 指定了条目的位置，就为 TRUE；如指定的是菜单 ID，就为 FALSE
lpcMenuItemInfoMENUITEMINFO，用于设置目标菜单条目的特征

Top
TrackPopupMenu
TrackPopupMenu, TrackPopupMenuBynum

VB 声明
Declare Function TrackPopupMenu& Lib "user32" (ByVal hMenu As Long, ByVal wFlags As Long, ByVal x As Long, ByVal y As Long, ByVal nReserved As Long, ByVal hwnd As Long, lprc As Rect)
Declare Function TrackPopupMenuBynum& Lib "user32" Alias "TrackPopupMenu" (ByVal hMenu As Long, ByVal wFlags As Long, ByVal x As Long, ByVal y As Long, ByVal nReserved As Long, ByVal hwnd As Long, ByVal lprc As Long)
说明
在屏幕的任意地方显示一个弹出式菜单
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hMenuLong，弹出式菜单的句柄
wFlagsLong，位置标志和鼠标追踪标志的组合，见下表
位置标志说明
TPM_CENTERALIGN 菜单在指定位置水平居中
TPM_LEFTALIGN 菜单的左侧置于水平 x 坐标处
TPM_RIGHTALIGN 菜单的右侧置于水平 x 坐标处
TPM_LEFTBUTTON 鼠标左键标准运作方式
TPM_RIGHTBUTTON 用鼠标右键进行菜单追踪
x,yLong，这个点指定了弹出式菜单在屏幕坐标系统中的位置
nReservedLong，未使用，设为零
hwndLong，用于接收弹出式菜单命令的窗口的句柄。应该使用窗体的窗口句柄——窗体中有一个菜单能象弹出式菜单那样接收相同的命令 ID 集
lprcRect，用屏幕坐标定义的一个矩形，如用户在这个矩形的范围内单击，则弹出式菜单不会关闭。如单击弹出式菜单之外的任何一个地方，则会关闭菜单。可以设为 NULL
注解
用这个函数创建的菜单，菜单中的命令 ID 并不与 vb 期望的那些相符

Top
TrackPopupMenuEx
TrackPopupMenuEx

VB 声明
Declare Function TrackPopupMenuEx Lib "user32" Alias "TrackPopupMenuEx" (ByVal hMenu As Long, ByVal un As Long, ByVal n1 As Long, ByVal n2 As Long, ByVal hWnd As Long, lpTPMParams As TPMPARAMS) As Long
说明
与 TrackPopupMenu 相似，只是它提供了额外的功能
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表

参数类型及说明
hMenuLong，弹出式菜单的句柄
unLong，定位标志和鼠标追踪标志的组合。参考 TrackPopupMenu，另外还包括两个标志：TPM_HORIZONTAL 或 TPM_VERTICAL。参考 lpTPMParams 参数的说明
n1,n2Long，定义了弹出式菜单位置的一个 x,y 点（n1,n2），用屏幕坐标表示
hWndLong，用于接收弹出式菜单命令的窗口的句柄。应该使用特定窗体的窗口句柄，该窗体有一个菜单能够与弹出式菜单一样接收相同的命令 ID 集
lpTPMParamsTPMPARAMS，指向一个 TPMPARAMS 结构的指针。这个结构包含了一个矩形，规定了不能由这个弹出式菜单覆盖的区域。如果在 un 参数中指定了 TPM_HORIZONTAL 标志，windows 就会试着设置水平位置，将弹出式菜单垂直移到这个矩形的外部。如指定了 TPM_VERTICAL，那么 windows 会试着水平移动弹出式菜单的位置

Top
文本和字体函数
文本和字体函数，共三页。第一页，第二页，第三页
AddFontResource 在 Windows 系统中添加一种字体资源
CreateFont 用指定的属性创建一种逻辑字体
CreateFontIndirect 用指定的属性创建一种逻辑字体
CreateScalableFontResource 为一种 TureType 字体创建一个资源文件，以便能用 API 函数 AddFontResource 将其加入 Windows 系统
DrawText 将文本描绘到指定的矩形中
DrawTextEx 与 DrawText 相似，只是加入了更多的功能
EnumFontFamilies 列举指定设备可用的字体
EnumFontFamiliesEx 列举指定设备可用的字体
EnumFonts 列举指定设备可用的字体
ExtTextOut 经过扩展的文本描绘函数。也请参考 SetTextAlign 函数
GetAspectRatioFilterEx 用 SetMapperFlags 要求 Windows 只选择与设备当前纵横比相符的光栅字体时，本函数可判断纵横比大小
GetCharABCWidths 判断 TureType 字体中一个或多个字符的 A-B-C 大小
GetCharABCWidthsFloat 查询一种字体中一个或多个字符的 A-B-C 尺寸
GetCharacterPlacement 该函数用于了解如何用一个给定的字符显示一个字串
GetCharWidth 调查字体中一个或多个字符的宽度
GetFontData 接收一种可缩放字体文件的数据
EnumFonts
EnumFonts

VB 声明
Declare Function EnumFonts Lib "gdi32" Alias "EnumFontsA" (ByVal hDC As Long, ByVal lpsz As String, ByVal lpFontEnumProc As Long, ByVal lParam As Long) As Long
说明
列举指定设备可用的字体
注解
该函数使用的参数与 EnumFontFamilies 函数是一样的，工作原理也大致相同。只是 EnumFontFamilies 会利用 ENUMLOGFONT 和 NEWTEXTMETRIC 结构向回调函数传递附

加的信息，而不是使用 LOGFONT 和 TEXTMETRIC 结构。请参考 EnumFontFamilies 函数，那里有更详细的解释

Top
ExtTextOut
ExtTextOut

VB 声明
Declare Function ExtTextOut Lib "gdi32" Alias "ExtTextOutA" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long, ByVal wOptions As Long, lpRect As Rect, ByVal lpString As String, ByVal nCount As Long, lpDx As Long) As Long
说明
经过扩展的文本描绘函数。也请参考 SetTextAlign 函数
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
x,yLong，以逻辑坐标表示的一个点，指定了绘图起点
wOptionsLong，下述标志常数的任意组合
ETO_CLIPPED 将文本剪切出指定的矩形
ETO_GLPYH_INDEXlpString 是一个字样索引表。参考对 GetCharacterPlacement 函数的说明。只适用于 Win95
ETO_OPAQUE 在正式描绘文本前，用当前的背景色填充矩形
lpRectRect，指定一个矩形，用于对文本进行格式化处理。可指定长整数 0，在不用矩形区域的前提下描绘文本
lpStringString，欲描绘的字串
nCountLong，字串中要显示出来的字符数
lpDxLong，如果不是零，这个参数就代表指向一个 Long 值数组的指针。该数组对每一对字符的间距进行了说明（采用逻辑单位）。其中第一个条目是第一和第二个字符的间距；第二个条目是第二和第三个字符的间距；以此类推。如果为零，函数就使用字体的默认间距设置

Top
GetAspectRatioFilterEx
GetAspectRatioFilterEx

VB 声明
Declare Function GetAspectRatioFilterEx Lib "gdi32" Alias "GetAspectRatioFilterEx" (ByVal hdc As Long, lpAspectRatio As SIZE) As Long
说明
我们可用 SetMapperFlags 函数要求 Windows 只选择与设备当前纵横比相符的光栅字体。这个函数可判断那种特殊选择过程中使用的纵横比是多大
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError

参数表
参数类型及说明
hdcLong，一个设备场景的句柄
lpAspectRatioSIZE，用于装载纵横比的一个结构

Top
GetCharABCWidths
GetCharABCWidths

VB 声明
Declare Function GetCharABCWidths Lib "gdi32" Alias "GetCharABCWidthsA" (ByVal hdc As Long, ByVal uFirstChar As Long, ByVal uLastChar As Long, lpabc As ABC) As Long
说明
判断 TureType 字体中一个或多个字符的 A-B-C 大小
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
uFirstCharLong，欲调查 A-B-C 尺寸的第一个字符的 ASCII 值
uLastCharLong，欲调查 A-B-C 尺寸的最后一个字符的 ASCII 值
lpabcABC，在 ABC 结构数组中的第一个条目。这个数组填充了指定的字符大小设置。该数组的长度必须足够大，足以容下要求的所有字符
注解
对于非 TureType 字体用 GetCharWidth 函数

Top
GetCharABCWidthsFloat
GetCharABCWidthsFloat

VB 声明
Declare Function GetCharABCWidthsFloat Lib "gdi32" Alias "GetCharABCWidthsFloatA" (ByVal hdc As Long, ByVal iFirstChar As Long, ByVal iLastChar As Long, lpABCF As ABCFLOAT) As Long
说明
查询一种字体中一个或多个字符的 A-B-C 尺寸
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
iFirstCharLong，欲调查 A-B-C 尺寸的第一个字符的 ASCII 值
iLastCharLong，欲调查 A-B-C 尺寸的最后一个字符的 ASCII 值
lpABCFABCFLOAT，在 ABCFLOAT 结构数组中的第一个条目。这个数组填充了指定的字

符大小设置。该数组的长度必须足够大，足以容下要求的所有字符
注解
和 GetCharABCWidths 不同，这个函数适用于任何字符。ABC 值是以浮点数的形式返回的，而且可能不是整数——具体取决于用这个函数处理非 TureType 字体时采用的转换方式
适用平台
Windows NT

Top
GetCharacterPlacement
GetCharacterPlacement

VB 声明
Declare Function GetCharacterPlacement Lib "gdi32" Alias " GetCharacterPlacementA" (ByVal hdc As Long, ByVal lpsz As String, ByVal n1 As Long, ByVal n2 As Long, lpGcpResults As GCP_RESULTS, ByVal dw As Long) As Long
说明
该函数用于了解如何用一个给定的字符显示一个字串
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
lpszString，欲分析的字串
n1Long，字串的长度
n2Long，若在 dw 参数中指定了 GCP_MAXEXTENT 常数，那么一旦显示的字串超出了由该参数指定的宽度（用逻辑单位），函数就会停止处理字串
lpGcpResultsGCP_RESULTS，在这个结构中装载为这个字串计算出来的信息
dwLong，下述常数的一个或多个：
GCP_CLASSINlpGcpResults 结构中的 lpClass 数组包含了字串中各字符的分类信息
GCP_DIACRITIC 在计算时将发音符和“废”字符考虑在内
GCP_DISPLAYZWG 显示某些字符集中使用的不可见字符，根据它们在一个词中的位置修改字符
GCP_GLPYPHSHAPE 允许对字样（字面）进行特殊处理。根据 GetFontLanguageInfo 函数的结果使用
GCP_JUSTIFY 调整字样位置，对字串进行对齐处理，使其与 n2 参数指定的范围相符
GCP_JUSTIFYINlpGcpResults 结构中的 lpDX 参数包含了计算过程中使用的对齐粗细设置
GCP_LIGATE 如当前字体支持，就用连字技术将字符合并成单独一个字符
GCP_MAXEXTENT 请参考对 n2 参数的说明
GCP_USERKERNING 计算字符位置时，使用字距表（如果有的话）可用其他标志对希伯来和阿拉伯字体进行特殊处理。这类语言按照从右到左的顺序显示文字，而且具体显示的字样由字符在一个词中的位置决定

Top
GetCharWidth

GetCharWidth, GetCharWidth32, GetCharWidthFloat

VB 声明
Declare Function GetCharWidth& Lib "gdi32" Alias "GetCharWidthA" (ByVal hDC As Long, ByVal iFirstChar As Long, ByVal iLastChar As Long, lpBuffer As Long)
Declare Function GetCharWidth32& Lib "gdi32" Alias "GetCharWidth32A" (ByVal hdc As Long, ByVal iFirstChar As Long, ByVal iLastChar As Long, lpBuffer As Long)
Declare Function GetCharWidthFloat& Lib "gdi32" Alias "GetCharWidthFloatA" (ByVal hdc As Long, ByVal iFirstChar As Long, ByVal iLastChar As Long, pxBuffer As Single)
说明
调查字体中一个或多个字符的宽度。在 Win32 环境中，请使用 GetCharWidth32 函数。用 GetCharWidthFloat 则可获得小数宽度
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hDCLong，设备场景的句柄
iFirstCharLong，要调查宽度的第一个字符的 ASCII 值
iLastCharLong，要调查宽度的最后一个字符的 ASCII 值
lpBufferLong，指定 Long 值数组的第一个条目。该数组容纳了字体的字符宽度设置
pxBufferSingle，指定 Single 值数组的第一个条目。该数组容纳了字体的字符宽度设置
注解
对于 TureType 字体，GetCharABCWidths 可获得更详细的信息

Top
GetFontData
GetFontData

VB 声明
Declare Function GetFontData Lib "gdi32" Alias "GetFontDataA" (ByVal hdc As Long, ByVal dwTable As Long, ByVal dwOffset As Long, lpvBuffer As Any, ByVal cbData As Long) As Long
说明
接收一种可缩放字体文件的数据。随后，可用这些数据将字体信息嵌入一个文档。如果需要在文档使用一种特殊字体，同时这种字体在大多数系统中都不常见，而且程序员希望文档无论如何都要显示这种字体，那么这种技术就相当有用了
在 VB 里使用
未经测试
Use with VB:Untested

Top
GetFontLanguageInfo
GetFontLanguageInfo

VB 声明

Declare Function GetFontLanguageInfo Lib "gdi32" Alias "GetFontLanguageInfo" (ByVal hdc As Long) As Long
说明
返回目前选入指定设备场景中的字体的信息
返回值
Long，如返回零，表示是简单字体；GCP_ERROR 表示出错。否则，返回下述一个或多个标志：
GCP_DBCS 双字节字符集
GCP_DIACRITIC 字体包含了发音字符
FLI_GLYPHS 字体包含了通常不会显示出来的字样
GCP_GLYPHSHAPE 字体包含了特殊字符，用于字样显示由除字符值以外的其他因素决定的场合。例如显示字样由一个字符在单词中位置决定，或者显示单个“连字”，指出这是两个字符值的组合
GCP_KASHIDA 在阿拉伯字体中使用
GCP_LIGATE 字体包含了连字字样
GCP_USERKERNING 字体包含了字距表
GCP_REORDER 字体必须记录下来，以便正确显示。随同希伯来和阿拉伯字体使用
参数表
参数类型及说明
hdcLong，设备场景的句柄
注解
从这个函数返回的值可用于设置 GetCharacterPlacement 函数的标志

Top
GetGlyphOutline
GetGlyphOutline

VB 声明
Declare Function GetGlyphOutline Lib "gdi32" Alias "GetGlyphOutlineA" (ByVal hdc As Long, ByVal uChar As Long, ByVal fuFormat As Long, lpgm As GLYPHMETRICS, ByVal cbBuffer As Long, lpBuffer As Any, lpmat2 As MAT2) As Long
说明
取得 TureType 字体中构成一个字符的曲线信息。主要用于将文本转换成曲线，以及用于字体的特殊处理（比如造字程序。无论如何，都要求程序员掌握高深的字体技术）。请参考由微软发布的 TureType 字体规格书，其中对这个字体进行了更详细的解释

Top
GetKerningPairs
GetKerningPairs

VB 声明
Declare Function GetKerningPairs Lib "gdi32" Alias "GetKerningPairsA" (ByVal hdc As Long, ByVal cPairs As Long, lpkrnpair As KERNINGPAIR) As Long
说明

取得指定字体的字距信息
返回值
Long，返回的字距对数量，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
cPairsLong，KERNINGPAIR 结构在数组中的数量，由 lpkrnpair 参数指定。如果将这个参数与 lpkrnpair 设置成零，可判断出字距表的大小
lpkrnpairKERNINGPAIR，指定 KERNINGPAIR 结构数组中的第一个条目
注解
参考 KERNINGPAIR
返回以后，结构会针对数组中的每个条目象下面这样设置字段：
wFirst 指定一个双字符序列的第一个字符；wSecond 指定第二个字符。iKernAmount 字段指定指定这两个字符的字间距。
例如，假设第一个字符是"f"，第二个是"i"。那么在这两个字符一个紧接一个显示出来时，字间距就是添加到默认字符间距上的一个逻辑距离。这个值通常为负，因为系统通常会令两个字符靠得更近

Top
GetOutlineTextMetrics
GetOutlineTextMetrics

VB 声明
Declare Function GetOutlineTextMetrics Lib "gdi32" Alias "GetOutlineTextMetricsA" (ByVal hdc As Long, ByVal cbData As Long, lpotm As OUTLINETEXTMETRIC) As Long
说明
接收与 TureType 字体内部特征有关的详细信息。请参考微软公司发布的 TureType 字体规格文件，它提供了这个函数的进一步信息

Top
GetRasterizerCaps
GetRasterizerCaps

VB 声明
Declare Function GetRasterizerCaps Lib "gdi32" Alias "GetRasterizerCaps" (lpraststat As RASTERIZER_STATUS, ByVal cb As Long) As Long
说明
了解系统是否有能力支持可缩放的字体。利用得到的结果，可判断那种系统中是否允许使用 TureType 字体
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpraststatRASTERIZER_STATUS，这个结构用于装载光栅信息

cbLong，欲复制到结构中的字符数
注解
结构目前的大小是 6 个字符，并包含在 RASTERIZER_STATUS 结构的第一个整数中

Top
GetTabbedTextExtent
GetTabbedTextExtent

VB 声明
Declare Function GetTabbedTextExtent Lib "user32" Alias "GetTabbedTextExtentA" (ByVal hdc As Long, ByVal lpString As String, ByVal nCount As Long, ByVal nTabPositions As Long, lpnTabStopPositions As Long) As Long
说明
判断一个字串占据的范围，同时考虑制表站扩充的因素。也请参考 TabbedTextOut 函数
返回值
Long，低 16 位包含了文本宽度，采用设备场景的逻辑坐标表示。高 16 位则包含了文本高度。零意味着出错
参数表
参数类型及说明
hdcLong，设备场景的句柄
lpStringString，欲计算的字串
nCountLong，字串中的字符数
nTabPositionsLong，lpnTabStopPositions 数组中的制表站数量。如果是零，则 lpnTabStopPositions 也应是 NULL（需另行创建一个函数声明，将这个参数声明成 ByVal nTabPositions&）。在这种情况下，制表站会根据当前字体的平均字符宽度，设置成默认的 8 字符间距。如 nTabPositions 是 1，那么制表站间距就会以 lpnTabStopPositions 数组的第一个条目为准
lpnTabStopPositionsLong，指定制表站位置数组的第一个条目。这种位置是按升序用设备坐标指定的
注解
进行这种计算的时候，剪切区不会考虑在内

Top
GetTextAlign
GetTextAlign

VB 声明
Declare Function GetTextAlign Lib "gdi32" Alias "GetTextAlign" (ByVal hdc As Long) As Long
说明
接收一个设备场景当前的文本对齐标志
返回值
Long，当前的文本对齐标志。GDI_ERROR 表示失败。会设置 GetLastError。文本的对齐方法由几个常数的组合决定。其中每个常数都来自下述不同的组别。参考下面总结的文本对齐标志

水平对齐标志 TA_CENTER 文本在约束矩形内居中显示
TA_LEFT 文本在约束矩形内左对齐（默认设置）
TA_RIGHT 文本在约束矩形内右对齐
垂直对齐标志定义文本输出函数的 Y 参数的含义
TA_BASELINEY 参数指定字体基线的位置
TA_BOTTOMY 参数指定约束矩形底边的位置
TA_TOPY 参数指定约束矩形顶边的位置（默认设置）
当前位置 TA_NOUPDATECP 文本输出函数不使用设备场景当前的绘图位置
TA_UPDATECP 文本输出函数使用设备场景当前的绘图位置。完成绘图后，输出函数会对当前位置进行更新。文本输出函数的 X 和 Y 参数会被忽略——绘图会以当前位置为起点
其他 TA_RTLREADING 文本输出从右到左进行。仅在 Windows95 下适用于希伯来和阿拉伯字体
参数表
参数类型及说明
hdcLong，设备场景的句柄
注解
参考对 SetTextAlign 函数的说明，进一步了解文本对齐标志的情况

Top
GetTextCharacterExtra
GetTextCharacterExtra

VB 声明
Declare Function GetTextCharacterExtra Lib "gdi32" Alias "GetTextCharacterExtraA" (ByVal hdc As Long) As Long
说明
判断额外字符间距的当前值。请参考 SetTextCharacterExtra 函数，了解进一步的情况
返回值
Long，返回 Windows 描绘文本时于字符间添加的额外空间大小
参数表
参数类型及说明
hdcLong，设备场景的句柄

Top
GetTextCharset
GetTextCharset

VB 声明
Declare Function GetTextCharset Lib "gdi32" Alias "GetTextCharset" (ByVal hdc As Long) As Long
说明
接收当前选入指定设备场景的字体的字符集标识符
返回值
Long，字符集标识符。DEFAULT_CHARSET 代表默认字符集；-1 表示出错

参数表
参数类型及说明
hdcLong，设备场景的句柄

Top
GetTextCharsetInfo
GetTextCharsetInfo

VB 声明
Declare Function GetTextCharsetInfo Lib "gdi32" Alias "GetTextCharsetInfo" (ByVal hdc As Long, lpSig As FONTSIGNATURE, ByVal dwFlags As Long) As Long
说明
获取与当前选定字体的字符集有关的详细信息
返回值
Long，字符集标识符。DEFAULT_CHARSET 代表默认字符集；-1 表示出错
参数表
参数类型及说明
hdcLong，设备场景的句柄
lpSigFONTSIGNATURE，用于装载字符集信息的结构。有关 Unicode 字体签名（signature）的资料可在 Unicode 规格文件中找到
dwFlagsLong，已保留，设为零

Top
GetTextColor
GetTextColor

VB 声明
Declare Function GetTextColor Lib "gdi32" Alias "GetTextColor" (ByVal hdc As Long) As Long
说明
判断当前字体颜色。通常也称为“前景色”
返回值
Long，文字的当前 RGB 颜色设置。如果出错，会返回 CLR_INVALID。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
注解
在 VB 里用 ForeColor 属性设置图片控件和窗体

Top
GetTextExtentExPoint
GetTextExtentExPoint

VB 声明
Declare Function GetTextExtentExPoint Lib "gdi32" Alias "GetTextExtentExPointA" (ByVal hdc

As Long, ByVal lpszStr As String, ByVal cchString As Long, ByVal nMaxExtent As Long, lpnFit As Long, alpDx As Long, lpSize As SIZE) As Long
说明
判断要填入指定区域的字符数量。也用一个数组装载每个字符的范围信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
lpszStrString，准备量度其范围的字串
cchStringLong，lpszStr 字串的长度
nMaxExtentLong，采用逻辑单位表示的水平范围
lpnFitLong，在其中保存欲填充到指定区域的字符数量。可以为 NULL（用一个别名化的声明来设置 ByVal As Long）——此时会忽略 nMaxExtent 设置
AsLong，cchString 数组的第一个条目。每个条目都要保存从字串起点到这个字符的距离（采用逻辑单位）。如果不需要这方面的信息，也可设为 NULL（用别名声明设置 ByVal As Long）
lpSizeSIZE，这个结构用于装载字串范围的高度和宽度信息
注解
可用这个函数计算自动换行输出时的字符位置

Top
GetTextExtentPoint
GetTextExtentPoint, GetTextExtentPoint32

VB 声明
Declare Function GetTextExtentPoint& Lib "gdi32" Alias "GetTextExtentPointA" (ByVal hdc As Long, ByVal lpszString As String, ByVal cbString As Long, lpSize As SIZE)
Declare Function GetTextExtentPoint32& Lib "gdi32" Alias "GetTextExtentPoint32A" (ByVal hdc As Long, ByVal lpsz As String, ByVal cbString As Long, lpSize As SIZE)
说明
判断一个字串的大小（范围）。在 Win32 环境中，最好使用 GetTextExtentPoint32，它提供了更精确的计算结果
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
lpszStringString，欲度量其范围（extent）的一个字串
cbStringLong，lpszString 字串的长度
lpSizeSIZE，这个结构用于装载字串范围的高度和宽度信息
注解
这个函数不会将剪切区考虑在内，但却考虑到了由 SetTextCharacterExtra 函数设置的任何额外空间（间距）

Top
GetTextFace
GetTextFace

VB 声明
Declare Function GetTextFace Lib "gdi32" Alias "GetTextFaceA" (ByVal hdc As Long, ByVal nCount As Long, ByVal lpFacename As String) As Long
说明
获取一种字体的字样名
返回值
Long，缓冲区中载入的字节数量。零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
nCountLong，lpFacename 字串的大小
lpFacenameString，指定一个字串缓冲区，用于装载当前选定字体的字样名称。这个缓冲区事先必须初始化成至少 nCount+1 个字符的长度
注解
类似于读取 VB 的 FontName 属性

Top
GetTextMetrics
GetTextMetrics

VB 声明
Declare Function GetTextMetrics Lib "gdi32" Alias "GetTextMetricsA" (ByVal hdc As Long, lpMetrics As TEXTMETRIC) As Long
说明
获取与选入一种设备场景的物理字体有关的信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
lpMetricsTEXTMETRIC，用于填充物理字体属性信息的一个结构

Top
GrayString
GrayString, GrayStringByString

VB 声明
Declare Function GrayString& Lib "user32" Alias "GrayStringA" (ByVal hDC As Long, ByVal hBrush As Long, ByVal lpOutputFunc As Long, ByVal lpData As Long, ByVal nCount As Long, ByVal X As Long, ByVal Y As Long, ByVal nWidth As Long, ByVal nHeight As Long)

Declare Function GrayStringByString& Lib "user32" Alias "GrayStringA" (ByVal hDC As Long, ByVal hBrush As Long, ByVal lpOutputFunc As Long, ByVal lpData As String, ByVal nCount As Long, ByVal X As Long, ByVal Y As Long, ByVal nWidth As
Long, ByVal nHeight As Long)
说明
描绘一个以灰色显示的字串。通常由 Windows 用于标识禁止状态。这个函数使用的是当前字体，但会忽略背景及文本颜色
返回值
Long，非零表示成功。如果 TextOut 函数或回调控件绘图事件返回零，那么该函数也会返回零
参数表
参数类型及说明
hDCLong，设备场景的句柄
hBrushLong，用于填充灰色的一个刷子。如果为零就用当前刷子
lpOutputFuncLong，指向一个函数的指针，该函数用于输出文本。通常设为零，表示使用 TextOut 函数。否则可将该参数设成一个进程地址。具体地址可用一个标准函数的 AddressOf 运算符返回，或者用一个回调控件的相关属性返回
lpDataLong，如果是一个字串（将 lpOutputFunc 设为 NULL），这就表示要以灰色显示的一个字串。否则就指定一个 Long 型变量，将其传递给回调函数
nCountLong，欲显示的字符数量。如果设为零，而且 lpData 是个字串，那么就用于计算字串的长度
X,YLong，将字串封闭（约束）起来的一个矩形的 X，Y 坐标
nWidthLong，将字串封闭起来的一个矩形的宽度，采用设备坐标表示。如设为零，就根据字串计算出具体的值
nHeightLong，将字串封闭起来的一个矩形的高度，采用设备坐标表示。如设为零，就根据字串计算出具体的值

Top
PolyTextOut
PolyTextOut

VB 声明
Declare Function PolyTextOut Lib "gdi32" Alias "PolyTextOutA" (ByVal hdc As Long, pptxt As POLYTEXT, cStrings As Long) As Long
说明
描绘一系列字串
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，欲在其中绘图的设备场景
pptxtPOLYTEXT，指定 POLYTEXT 结构数组中的第一个条目。该结构对要描绘字串的位置及内容进行了说明
cStringsLong，pptxt 数组中的条目数量

Top
RemoveFontResource
RemoveFontResource

VB 声明
Declare Function RemoveFontResource Lib "gdi32" Alias "RemoveFontResourceA" (ByVal lpFileName As String) As Long
说明
从 Windows 系统中删除一种字体资源。如删除的字体目前正由其他应用程序使用，则并不将其立即删除
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpFileNameString，一个字体资源文件的文件名
注解
删除一种字体资源后，注意必须调用一下 API 函数：
di% = SendMessageBynum(HWND_BROADCAST, WM_FONTCHANGE, X, Y)
其中，HWND_BROADCAST 和 WM_FONTCHANGE 都是来自 API32.TXT 文件的常数。它的作用是通知所有 Windows 应用程序字体列表已发生了变化。
注意磁盘上的字体文件本身并不会由这个函数删除

Top
SetMapperFlags
SetMapperFlags

VB 声明
Declare Function SetMapperFlags Lib "gdi32" Alias "SetMapperFlags" (ByVal hdc As Long, ByVal dwFlag As Long) As Long
说明
Windows 对字体进行映射时，可用该函数选择与目标设备的纵横比相符的光栅字体。请参考对 GetAspectRatioFilterEx 函数的解释，了解进一步的情况
返回值
Long，字体映射标志的前一个值。GDI_ERROR 表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
dwFlagLong，用 ASPECT_FILTERING 常数请求 GDI 选择与设备纵横比相符的字体

Top
SetTextAlign
SetTextAlign

VB 声明
Declare Function SetTextAlign Lib "gdi32" Alias "SetTextAlign" (ByVal hdc As Long, ByVal wFlags As Long) As Long
说明
设置文本对齐方式，并指定在文本输出过程中使用设备场景的当前位置
返回值
Long，前一个文本对齐标志，GDI_ERROR 表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
wFlagsLong，参考 GetTextAlign 函数的返回值列表
在 VB 里使用
针对自己修改的任何 VB 窗体或控件，注意确定恢复其原始的对齐排列状态。可用 GetTextAlign 函数了解目前的对齐方式是什么

Top
SetTextCharacterExtra
SetTextCharacterExtra

VB 声明
Declare Function SetTextCharacterExtra Lib "gdi32" Alias "SetTextCharacterExtraA" (ByVal hdc As Long, ByVal nCharExtra As Long) As Long
说明
描绘文本的时候，指定要在字符间插入的额外间距
返回值
Long，这个设备场景的前一个额外间距设置
参数表
参数类型及说明
hdcLong，设备场景的句柄
nCharExtraLong，要在字符间插入的额外空间，采用设备场景的逻辑坐标系统
在 VB 里使用
如改变了这个设置，注意恢复 VB 窗体或控件原来的字符间距设置

Top
SetTextColor
SetTextColor

VB 声明
Declare Function SetTextColor Lib "gdi32" Alias "SetTextColor" (ByVal hdc As Long, ByVal crColor As Long) As Long
说明
设置当前文本颜色。这种颜色也称为“前景色”
返回值
Long，文本色的前一个 RGB 颜色设定。CLR_INVALID 表示失败。会设置 GetLastError

参数表
参数类型及说明
hdcLong，设备场景的句柄
crColorLong，新的文本色
在 VB 里使用
如改变了这个设置，注意恢复 VB 窗体或控件原始的文本颜色

Top
SetTextJustification
SetTextJustification

VB 声明
Declare Function SetTextJustification Lib "gdi32" Alias "SetTextJustification" (ByVal hdc As Long, ByVal nBreakExtra As Long, ByVal nBreakCount As Long) As Long
说明
通过指定一个文本行应占据的额外空间，可用这个函数对文本进行两端对齐处理
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
nBreakExtraLong，描绘时欲添加到字串的额外空间大小
nBreakCountLong，用于分散额外空间的分隔字符的数量
在 VB 里使用
如使用了这个函数，要确定针对 VB 窗体或控件清除错误条件
注解
额外空间由行内各个分隔字符分摊。这里的“分隔字符”是由特定的字体定义的，通常都是空格字符。可用 GetTextMetrics 函数了解一种字体采用的分隔字符是什么。对文本进行两端对齐排列的时候，通常需要采取的操作步骤如下：
1、用 GetTextExtentPoint32 这个 API 函数计算字串占据的显示范围
2、决定为了使一个行两端对齐，需要加入多少额外的空间（采用逻辑坐标）。这个空间（或距离）通常等于右页边距减去文本的水平“范围”
3、计算一行文本中采用多少个间隔字符（通常是空格）
4、将额外空间以及间隔字符的数量作为参数，调用 SetTextJustification 函数
5、调用文本绘图（显示）函数
这个函数在内部维持着一种错误条件，用于纠正对齐过程中出现的误差。这样一来，我们就可以区分出行内不同部分间的额外间距（如行内使用了多种字体）。具体的方法是将行分割成几个段，然后为每一段都调用这个函数。对于一个新行，必须清除这个错误条件，方法是向 nBreakExtra 和 nBreakCount 参数传递零值，然后调用这个函数

Top
TabbedTextOut
TabbedTextOut

VB 声明
Declare Function TabbedTextOut Lib "user32" Alias "TabbedTextOutA" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long, ByVal lpString As String, ByVal nCount As Long, ByVal nTabPositions As Long, lpnTabStopPositions As Long, ByVal nTabOrigin As Long) As Long
说明
支持制表站的一个文本描绘函数。也请参考 SetTextAlign 函数
返回值
Long，返回字串的显示“范围”。其中，结果值的高 16 位代表高度，低 16 位代表宽度
参数表
参数类型及说明
hdcLong，设备场景的句柄
x,yLong，用逻辑坐标设置的一个点，指定字体的描绘（显示）起点
lpStringString，欲描绘的字串
nCountLong，字串中要正式描绘出来的字符数
nTabPositionsLong，lpnTabStopPositions 数组中的制表站数量。如果是零，lpnTabStopPositions 也应该是 NULL（需要另行创建一个声明，将参数指定成 ByVal nTabPositions& ）——在这种情况下，制表站会根据当前字体的平均字符宽度设置成默认的 8 字符间距。如 nTabPositions 为 1，那么制表站间距就会根据 lpnTabStopPositions 数组的第一个条目设置
lpnTabStopPositionsLong，指定制表站位置数组中的头一个条目。这些位置用设备坐标按升序指定。如果为负数，表示文本应该右对齐制表站，而不是默认的左对齐（仅适用于 Win95）
nTabOriginLong，指定制表站起点。如为同一行多次调用该函数，而又希望维持相同的制表起点，这个参数就显得非常重要

Top
TextOut
TextOut

VB 声明
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
说明
文本绘图函数。也请参考 SetTextAlign
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
x,yLong，绘图的起点，采用逻辑坐标
lpStringString，欲描绘的字串
nCountLong，字串中要描绘的字符数量
注解
在一个路径中，如绘图背景模式是“不透明”（opaque）——请参考 SetBkMode 函数，那么创建的轮廓将由字符单元格减去字符构成。如背景模式为透明，轮廓就由字符的字样本身构成

Top
硬件与系统函数
硬件与系统函数：共七页。第一页，第二页，第三页，第四页，第五页，第六页，第七页
ActivateKeyboardLayout 激活一个新的键盘布局。键盘布局定义了按键在一种物理性键盘上的位置与含义
Beep 用于生成简单的声音
CharToOem 将一个字串从 ANSI 字符集转换到 OEM 字符集
ClipCursor 将指针限制到指定区域
ConvertDefaultLocale 将一个特殊的地方标识符转换成真实的地方 ID
CreateCaret 根据指定的信息创建一个插入符（光标），并将它选定为指定窗口的默认插入符
DestroyCaret 清除（破坏）一个插入符
EnumCalendarInfo 枚举在指定“地方”环境中可用的日历信息
EnumDateFormats 列举指定的“当地”设置中可用的长、短日期格式
EnumSystemCodePages 枚举系统中已安装或支持的代码页
EnumSystemLocales 枚举系统已经安装或提供支持的“地方”设置
EnumTimeFormats 枚举一个指定的地方适用的时间格式
ExitWindowsEx 退出 windows，并用特定的选项重新启动
ExpandEnvironmentStrings 扩充环境字串
FreeEnvironmentStrings 翻译指定的环境字串块
GetACP 判断目前正在生效的 ANSI 代码页
ActivateKeyboardLayout
ActivateKeyboardLayout

VB 声明
Declare Function ActivateKeyboardLayout Lib "user32" Alias "ActivateKeyboardLayout" (ByVal HKL As Long, ByVal flags As Long) As Long
说明
激活一个新的键盘布局。键盘布局定义了按键在一种物理性键盘上的位置与含义
返回值
Long，如执行成功，返回前一个键盘布局的句柄；零表示失败。会设置 GetLastError
参数表
参数类型及说明
HKLLong，指定一个键盘布局的句柄。这个布局是随同 LoadKeyboardLayout 或 GetKeyboardLayoutList 函数载入的。也可用 HKL_NEXT 常数激活下一个已装载布局；或用 HKL_PREV 载入前一个布局
flagsLong，将指定的键盘移至内部键盘布局列表的起始处

Top
Beep
Beep

VB 声明
Declare Function Beep Lib "kernel32" Alias "Beep" (ByVal dwFreq As Long, ByVal dwDuration

As Long) As Long
说明
用于生成简单的声音
返回值
Long，TRUE（非零）表示成功，否则返回零。会设置 GetLastError
参数表
参数类型及说明
dwFreqLong，声音频率（从 37Hz 到 32767Hz）。在 windows95 中忽略
dwDurationLong，声音的持续时间，以毫秒为单位。如为-1，表示一直播放声音，直到再次调用该函数为止。在 windows95 中会被忽略
注解
在 windows95 中，这个函数简单的播放默认系统响铃

Top
CharToOem
CharToOem, CharToOemBuff

VB 声明
Declare Function CharToOem& Lib "user32" Alias "CharToOemA" (ByVal lpszSrc As String, ByVal lpszDst As String)
Declare Function CharToOemBuff& Lib "user32" Alias "CharToOemBuffA" (ByVal lpszSrc As String, ByVal lpszDst As String, ByVal cchDstLength As Long)
说明
将一个字串从 ANSI 字符集转换到 OEM 字符集。CharToOemBuff 允许我们指定字串中需转换的字符数量
返回值
Long，肯定是 TRUE
参数表
参数类型及说明
lpszSrcString，欲转换的字串
lpszDstString，用于包含转换结果的 OEM 字串。注意事先将字串初始化成合适的长度。可将相同的字串传递给这两个参数，执行本地转换（即在同一个字串中转换）
cchDstLengthLong，在字串 lpszSrc 中想转换的字符数量
注解
如用一个 Win32 类型库访问宽字符函数 CharToOemW，则 lpszSrc 是一个 Unicode 字串，且 lpszDst 参数绝对不能与 lpszSrc 相同

Top
ClipCursor
ClipCursor, ClipCursorBynum

VB 声明
Declare Function ClipCursor& Lib "user32" (lpRect As RECT)
Declare Function ClipCursorBynum& Lib "user32" Alias "ClipCursor" (ByVal lpRect As Long)

说明
将指针限制到指定区域。ClipCursorBynum 是一个别名，允许我们清除以前设置的指针剪切区域
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpRectRECT，指定一个矩形，用像素屏幕坐标系统表示。鼠标指针必须在这个区域内运动。如使用函数的 ClipCursorBynum 形式，则可将参数设为 Long 值，用它传递一个 0，禁止指针剪切，恢复常规运作状态
注解
指针剪切后，按 Ctrl+Alt+Del 可简单的清除剪切区域

Top
ConvertDefaultLocale
ConvertDefaultLocale

VB 声明
Declare Function ConvertDefaultLocale Lib "KERNEL32" Alias "ConvertDefaultLocale" (ByVal Locale As Long) As Long
说明
将一个特殊的地方标识符转换成真实的地方 ID
返回值
Long，如执行成功，返回实际的地方 ID。如失败则返回传递给 Locale 参数的值
参数表
参数类型及说明
LocaleLong，如设为常数 LOCALE_SYSTEM_DEFAULT 和 LOCALE_USER_DEFAULT，可分别接收默认的系统或用户地方设置。如设为零，则取得语言方面的地方设置。主语言 ID 用于接收采用了默认子语言的地方信息
注解
用 LOCALE_SYSTEM_DEFAULT 和 LOCALE_USER_DEFAULT 这两个值调用函数时，得到的结果与 GetSystemDefaultLCID 和 GetUserDefaultLCID 函数是一样的

Top
CreateCaret
CreateCaret

VB 声明
Declare Function CreateCaret Lib "user32" Alias "CreateCaret" (ByVal hwnd As Long, ByVal hBitmap As Long, ByVal nWidth As Long, ByVal nHeight As Long) As Long
说明
根据指定的信息创建一个插入符（光标），并将它选定为指定窗口的默认插入符。插入符可以是一根短线、一个方块或者一幅位图。通常用插入符指示文字在文字框中的插入位置
返回值

Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，拥有插入符的那个窗口的句柄
hBitmapLong，用作插入符的一幅位图的句柄。可以是 0 或 1；在这种情况下，插入符可通过 nWidth 和 nHeight 参数创建。如设为 1，则新插入符以灰色显示；而不是传统的黑色
nWidthLong，采用逻辑单位的插入符的宽度
nHeightLong，采用逻辑单位的插入符的高度
注解
如创建一个插入符，会同时清除原先的插入符；效果等同于 DestroyCaret 函数。在 vb 的 LostFocus 事件期间，不要试图用 DestroyCaret 函数清除一个插入符。这是由于 vb 的 LostFocus 事件不会接收 DestroyCaret 直到另一个窗口已经有焦点。因此，倘若在那个时候调用 DestroyCaret，会破坏其他窗口的插入符。如果准备自己管理插入符，可以（而且应该）在 WM_KILLFOCUS 消息期间清除插入符
在 vb 里使用
可以使用。但在应用程序切换时，标准的 vb 文本控件不能处理 GotFocus 和 LostFocus 事件。因此，很难知道何时为一个控件设置插入符。此外，vb 假设插入符为一根短竖线，并据此定义它的位置，所以其他形式的插入符可能无法正确定位

Top
DestroyCaret
DestroyCaret

VB 声明
Declare Function DestroyCaret Lib "user32" Alias "DestroyCaret" () As Long
说明
清除（破坏）一个插入符
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
注解
参考 CreateCaret 函数的注解

Top
EnumCalendarInfo
EnumCalendarInfo

VB 声明
Declare Function EnumCalendarInfo Lib "kernel32" Alias "EnumCalendarInfoA" (ByVal lpCalInfoEnumProc As Long, ByVal Locale As Long, ByVal Calendar As Long, ByVal CalType As Long) As Long
说明
枚举在指定“地方”环境中可用的日历信息
返回值
Long，TRUE（非零）表示成功，零表示失败。会将 GetLastError 设置设为下述某个常数：

ERROR_BADDB，ERROR_INVALID_FLAGS，ERROR_INVALID_PARAMETER
参数表
参数类型及说明
lpCalInfoEnumProcLong，指向为每个日历都调用的一个函数的指针。用 Addressof 运算符得到位于一个标准模块中的函数的地址
LocaleLong，用于决定具体枚举日历的“地方”设置
CalendarLong，如设为常数 ENUM_ALL_CALANDARS，表示枚举所有日历；如设为 1，枚举本地罗马日历，2 枚举英国罗马日历，3 枚举日本日历，4 枚举中国日历，5 枚举朝鲜日历
CalTypeLong，CAL_函数之一，指示欲枚举的信息类型

Top
EnumDateFormats
EnumDateFormats

VB 声明
Declare Function EnumDateFormats Lib "KERNEL32" Alias "EnumDateFormats" (ByVal lpDateFmtEnumProc As Long, ByVal Locale As Long, ByVal dwFlags As Long) As Long
说明
列举指定的“当地”设置中可用的长、短日期格式
返回值
Long，TRUE（非零）表示成功，零表示失败。会将 GetLastError 设置设为下述某个常数：
ERROR_BADDB，ERROR_INVALID_FLAGS，ERROR_INVALID_PARAMETER
参数表
参数类型及说明
lpDateFmtEnumProcLong，指向为每种日期格式都调用的一个函数的指针。用 Addressof 运算符得到位于一个标准模块中的函数的地址
LocaleLong，用于决定具体枚举格式的“地方”的设置
dwFlagsLong，DATE_SHORTDATE 或 DATE_LONGDATE 常数之一，分别用于枚举短或长日期格式

Top
EnumSystemCodePages
EnumSystemCodePages

VB 声明
Declare Function EnumSystemCodePages Lib "KERNEL32" Alias "EnumSystemCodePages" (ByVal lpCodePageEnumProc As Long, ByVal dwFlags As Long) As Long
说明
枚举系统中已安装或支持的代码页
返回值
Long，TRUE（非零）表示成功，零表示失败。会将 GetLastError 设置设为下述某个常数：
ERROR_BADDB，ERROR_INVALID_FLAGS，ERROR_INVALID_PARAMETER
参数表
参数类型及说明

lpCodePageEnumProcLong，指向为每个代码页都调用的一个函数的指针。用 Addressof 运算符得到位于一个标准模块中的函数的地址
dwFlagsLong，常数 CP_INSTALLED 表示枚举已安装的所有代码页，CP_SUPPORTED 表示枚举所有支持的代码页

Top
EnumSystemLocales
EnumSystemLocales

VB 声明
Declare Function EnumSystemLocales Lib "KERNEL32" Alias "EnumSystemLocales" (ByVal lpLocaleEnumProc As Long, ByVal dwFlags As Long) As Long
说明
枚举系统已经安装或提供支持的“地方”设置
返回值
Long，TRUE（非零）表示成功，零表示失败。会将 GetLastError 设置设为下述某个常数：ERROR_BADDB，ERROR_INVALID_FLAGS，ERROR_INVALID_PARAMETER
参数表
参数类型及说明
lpLocaleEnumProcLong，指向为每个“地方”都调用的一个函数的指针。用 Addressof 运算符得到位于一个标准模块中的函数的地址
dwFlagsLong，常数 LCID_INSTALLED 表示枚举已安装的所有地方；LCID_SUPPORTED 则枚举所有支持的地方

Top
EnumTimeFormats
EnumTimeFormats

VB 声明
Declare Function EnumTimeFormats Lib "KERNEL32" Alias "EnumTimeFormats" (ByVal lpTimeFmtEnumProc As Long, ByVal Locale As Long, ByVal dwFlags As Long) As Long
说明
枚举一个指定的地方适用的时间格式
返回值
Long，TRUE（非零）表示成功，零表示失败。会将 GetLastError 设置设为下述某个常数：ERROR_BADDB，ERROR_INVALID_FLAGS，ERROR_INVALID_PARAMETER
参数表
参数类型及说明
lpTimeFmtEnumProcLong，指向为每种时间格式都调用的一个函数的指针。用 Addressof 运算符得到位于一个标准模块中的函数的地址
LocaleLong，用于决定具体枚举格式的“地方”设置
dwFlagsLong，未用，设为零

Top

ExitWindowsEx
ExitWindowsEx

VB 声明
Declare Function ExitWindowsEx Lib "user32" Alias "ExitWindowsEx" (ByVal uFlags As Long, ByVal dwReserved As Long) As Long
说明
退出 windows，并用特定的选项重新启动
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
uFlagsLong，指定下述一个或多个标志（用 OR 运算符合并到一起）
EWX_FORCE 强迫中止没有响应的进程
EWX_LOGOFF 中止进程，然后注销
EWX_SHUTDOWN 关掉系统电源（如果可能的话，ATX 电源就可以）
EWX_REBOOT 重新引导系统
EWX_SHUTDOWN 关闭系统
dwReservedLong，保留，设为零
注解
这个函数调用后会立刻返回，系统关闭过程是在后台进行的。注意先中止自己的应用程序，使关闭过程更显平顺。当然，您的进程必须有足够的优先权，否则也不能执行这种操作

Top
ExpandEnvironmentStrings
ExpandEnvironmentStrings

VB 声明
Declare Function ExpandEnvironmentStrings Lib "kernel32" Alias "ExpandEnvironmentStringsA" (ByVal lpSrc As String, ByVal lpDst As String, ByVal nSize As Long) As Long
说明
扩充环境字串。具体操作过程与命令行处理的所为差不多。也就是说，将由百分号封闭起来的环境变量名转换成那个变量的内容。比如，“%path%”会扩充成完整路径。在 vb 里经常用于为新进程创建一个环境块
返回值
Long，lpDst 要求的缓冲区的大小。如 nSize 小于这个数字（也就是说，缓冲区太小，以至不能全容下扩充过后的字串），那么 lpDst 不会被载入。可利用这个结果改变字串的大小。零表示遇到错误。会设置 GetLastError
参数表
参数类型及说明
lpSrcString，欲扩充的字串
lpDstString，扩充过后的字串
nSizeLong，lpDst 的长度。注意预先对 lpDst 进行初始化，使其与这个长度相符

示例
```
Dim s$, dl&
Dim y As String * 5
s$ = "%PATH%"
dl& = ExpandEnvironmentStrings(s$, y, 499)
Print y
```

Top
FreeEnvironmentStrings
FreeEnvironmentStrings

VB 声明
```
Declare Function FreeEnvironmentStrings& Lib "kernel32" Alias "FreeEnvironmentStringsA" (ByVal lpsz As Long)
```
说明
翻译指定的环境字串块
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpszLong，指向一个内存块的句柄；那个内存块是以前通过 GetEnvironmentStrings 函数获得的
注解
开头的声明来自于我的资料，而在 vb 自带的 api 文本查看器中的声明为：Declare Function FreeEnvironmentStrings Lib "kernel32" Alias "FreeEnvironmentStringsA" (ByVal lpsz As String) As Long，lpsz 的类型为 String，不知是谁的错了

Top
GetACP
GetACP

VB 声明
```
Declare Function GetACP Lib "kernel32" Alias "GetACP" () As Long
```
说明
判断目前正在生效的 ANSI 代码页
返回值
Long，目前活动 ANSI 代码页的标识符。针对一种特定的语言，可能存在多个这样的代码页。可能的代码页包括下面这些：
874 泰语 932 日语
936 中文（简体）949 朝鲜语
950 中文（台湾和香港繁体）1200Unicode
1250 东欧语言 1251 西里尔语
1252 美国和西欧语言 1253 希腊语
1254 土耳其语 1255 希伯来语

1256 阿拉伯语 1257 波罗的语
注解
不要混淆 ANSI 代码页与 OEM 代码页的概念！ANSI 代码页为不同版本的 windows 定义标准的 ANSI 8 位字符集。而 OEM 代码页指定基础 DOS 代码页，由系统及键盘使用

Top
GetAsyncKeyState
GetAsyncKeyState

VB 声明
Declare Function GetAsyncKeyState Lib "user32" Alias "GetAsyncKeyState" (ByVal vKey As Long) As Integer
说明
判断函数调用时指定虚拟键的状态
返回值
Long，自对 GetAsyncKeyState 函数的上一次调用以来，如键已被按过，则位 0 设为 1；否则设为 0。如键目前处于按下状态，则位 15 设为 1；如抬起，则为 0。微软的 win32 手册指出：倘若输入焦点从属于与调用函数的输入线程不同的另一个输入线程，则返回值为 0（例如，一旦另一个程序拥有焦点，则它应返回零）。证据显示，函数实际是在整个系统的范围内工作的
参数表
参数类型及说明
vKeyLong，欲测试的虚拟键的键码
注解
如指定了 VK_LBUTTON 或 VK_RBUTTON，按钮的状态就会根据实际的按钮报告——无论是否曾用 SwapMouseButton 函数对鼠标的位置进行了交换。win32 提供了额外的一些虚拟键码，比如 VK_LSHIFT 和 VK_RSHIFT，以便在两个完全一样的键中区分出左右（也包括 Ctrl 和 Alt）

Top
GetCaretBlinkTime
GetCaretBlinkTime

VB 声明
Declare Function GetCaretBlinkTime Lib "user32" Alias "GetCaretBlinkTime" () As Long
说明
判断插入符光标的闪烁频率
返回值
Long，插入符连续两次闪烁间隔的时间，以毫秒为单位。零表示函数调用失败。会设置 GetLastError

Top
GetCaretPos
GetCaretPos

VB 声明
Declare Function GetCaretPos Lib "user32" Alias "GetCaretPos" (lpPoint As POINTAPI) As Long
说明
判断插入符的当前位置
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpPointPOINTAPI，这个结构会随同插入符在窗口客户坐标系统中的位置载入；那个窗口是插入符的父窗口

Top
GetClipCursor
GetClipCursor

VB 声明
Declare Function GetClipCursor Lib "user32" Alias "GetClipCursor" (lprc As RECT) As Long
说明
取得一个矩形，用于描述目前为鼠标指针规定的剪切区域；该区域是由 SetClipCursor 函数定义的
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lprcRECT，在屏幕坐标系统中随同当前剪切矩形载入的一个矩形。倘若没有活动的剪切，这个矩形会反映出整个显示屏幕的大小

Top
GetCommandLine
GetCommandLine

VB 声明
Declare Function GetCommandLine Lib "kernel32" Alias "GetCommandLineA" () As String
说明
获得指向当前命令行缓冲区的一个指针
返回值
Long，命令行缓冲区在内存中的地址
注解
Visual Basic Command 函数更易获取参数，但它未提供可执行的名称。使用这个函数时，要求进行内存复制操作

Top
GetComputerName

GetComputerName

VB 声明
Declare Function GetComputerName Lib "kernel32" Alias "GetComputerNameA" (ByVal lpBuffer As String, nSize As Long) As Long
说明
取得这台计算机的名称
返回值
Long，TRUE（非零）表示成功，否则返回零。会设置 GetLastError
参数表
参数类型及说明
lpBufferString，随同计算机名载入的字串缓冲区
nSizeLong，缓冲区的长度。这个变量随同返回计算机名的实际长度载入
注解
注意 nSize 参数并不是按值传递的。参考 api32.txt，了解 MAX_COMPUTER_NAME 常数的值
示例
```
Dim s$
s$ = String$(MAX_COMPUTERNAME_LENGTH+1,0)
Dim dl&
Dim sz&
sz& = MAX_COMPUTERNAME_LENGTH+1
dl& = GetComputerName(s$, sz)
```
其他
也许你会发现，MAX_COMPUTERNAME_LENGTH 常数在 vb 自带的 api 文本查看器中找不到。的确，我也没有找到。但我有一个工具：Listapi，这个常数在它那里可以找到

Top
GetCPInfo
GetCPInfo

VB 声明
Declare Function GetCPInfo Lib "kernel32" Alias "GetCPInfo" (ByVal CodePage As Long, lpCPInfo As CPINFO) As Long
说明
取得与指定代码页有关的信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
CodePageLong，欲为其载入信息的代码页的标识符。可能是一个 ANSI 或 OEM 代码页
lpCPInfoCPINFO，用于容纳代码页信息的一个结构

Top

GetCurrencyFormat
GetCurrencyFormat, GetCurrencyFormatBynum

VB 声明
Declare Function GetCurrencyFormat& Lib "kernel32" Alias "GetCurrencyFormatA" (ByVal Locale As Long, ByVal dwFlags As Long, ByVal lpValue As String, lpFormat As CURRENCYFMT, ByVal lpCurrencyStr As String, ByVal cchCurrency As Long)
Declare Function GetCurrencyFormatBynum& Lib "kernel32" Alias "GetCurrencyFormatA" (ByVal Locale As Long, ByVal dwFlags As Long, ByVal lpValue As String, ByVal lpFormat As Long, ByVal lpCurrencyStr As String, ByVal cchCurrency As Long)
说明
针对指定的“地方”设置，根据货币格式格式化一个数字
返回值
Long，格式化过后的字串的长度。零表示出错。会将 GetLastError 设置为下述值之一：ERROR_INSUFFICIENT_BUFFER，ERROR_INVALID_FLAGS，ERROR_INVALID_PARAMETER
参数表
参数类型及说明
LocaleLong，用于决定格式的地方 ID。lpFormat 参数中指定的任何信息（倘若不是 NULL）都优先于特定于地方的信息
dwFlagsLong，如指定了 lpFormat，那么该参数应该为零。否则，可设为 LOCALE_NOUSEROVERRIDE，强制使用系统地方参数——即使它们已由用户取代
lpValueString，指定欲格式化的数字。可以只有数位、一个前缀“-”号以及一个小数点
lpFormatCURRENCYFMT，可设为 NULL，使用特定于不同地方的值（用 GetCurrencyFormatBynum，则可通过 ByVal As Long 形式传递这个参数）。否则，可引用一个 CURRENCYFMT 结构，其中包含所有必要的字段，可填入需要用到的信息
lpCurrencyStrString，指定一个缓冲区，用于容纳格式化过后的字串。注意事先对字串进行初始化
cchCurrencyLong，lpCurrencyStr 缓冲区的长度。如为零，表示函数会返回需要缓冲区的大小
注解
在 vb 里，如使用一个别名，其中的 lpFormat 设为 NULL，则可以正常使用。CURRENCYFMT 结构的正确预初始化非常具有挑战性

Top
GetCursor
GetCursor

VB 声明
Declare Function GetCursor Lib "user32" Alias "GetCursor" () As Long
说明
获取目前选择的鼠标指针的句柄
返回值
Long，目前使用的指针的句柄。倘若不存在指针，则返回零

注解
这个函数返回的是当前线程的指针——不能获取其他应用程序的指针

Top
GetCursorPos
GetCursorPos

VB 声明
Declare Function GetCursorPos Lib "user32" Alias "GetCursorPos" (lpPoint As POINTAPI) As Long
说明
获取鼠标指针的当前位置
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpPointPOINTAPI，随同指针在屏幕像素坐标中的位置载入的一个结构

Top
GetDateFormat
GetDateFormat

VB 声明
Declare Function GetDateFormat Lib "kernel32" Alias "GetDateFormatA" (ByVal Locale As Long, ByVal dwFlags As Long, lpDate As SYSTEMTIME, ByVal lpFormat As String, ByVal lpDateStr As String, ByVal cchDate As Long) As Long
说明
针对指定的“当地”格式，对一个系统日期进行格式化
返回值
Long，格式化过后的字串的长度。零表示出错，会将 GetLastError 设置为下述值之一：ERROR_INSUFFICIENT_BUFFER，ERROR_INVALID_FLAGS，ERROR_INVALID_PARAMETER
参数表
参数类型及说明
LocaleLong，用于决定格式的地方 ID。lpFormat 参数中指定的任何信息（倘若不是 NULL）都优先于特定于地方的信息
dwFlagsLong，如指定了 lpFormat，那么该参数应该为零。否则，可设为 LOCALE_NOUSEROVERRIDE，强制使用系统地方参数——即使它们已由用户取代。用 DATE_SHORTDATE 或 DATE_LONGDATE 选择不同的日期格式
lpDateSYSTEMTIME，包含了一个系统日期的结构
lpFormatString，可设为 NULL，使用特定于不同地方的值（用 vbNullString 传递一个 NULL）。否则包含一个日期格式字串。对 d,dd,ddd,dddd,m,mm,mmm,mmmm,y,yy,yyyy 这样的代码，它们的用法与在 vb 格式命令中的用法是相同的。注意用 gg 指定一个“纪元”
lpDateStrString，指定一个缓冲区，用于容纳格式化过后的字串。注意事先对字串进行初始

化
cchDateLong，lpDateStr 缓冲区的长度。如为零，表示函数会返回需要缓冲区的大小

Top
GetDoubleClickTime
GetDoubleClickTime

VB 声明
Declare Function GetDoubleClickTime Lib "user32" Alias "GetDoubleClickTime" () As Long
说明
判断连续两次鼠标单击之间会被处理成双击事件的间隔时间
返回值
Long，以毫秒表示的双击时间

Top
GetEnvironmentStrings
GetEnvironmentStrings

VB 声明
Declare Function GetEnvironmentStrings& Lib "kernel32" Alias "GetEnvironmentStringsA" ()
说明
为包含了当前环境字串设置的一个内存块分配和返回一个句柄。这个内存块包含了所有环境字串。各字串间用一个 NULL 分隔；连续两个 NULL 标志着列表的结尾
返回值
Long，指向包含了环境字串的一个内存块的地址。零表示失败。会设置 GetLastError
注解
注意一定要用 FreeEnvironmentStrings 函数释放这个内存块
其他
请看从 vb 的 api 文本查看器复制的声明：Declare Function GetEnvironmentStrings Lib "kernel32" Alias "GetEnvironmentStringsA" () As String，与前面的声明返回值不同

Top
GetEnvironmentVariable
GetEnvironmentVariable

VB 声明
Declare Function GetEnvironmentVariable Lib "kernel32" Alias "GetEnvironmentVariableA" (ByVal lpName As String, ByVal lpBuffer As String, ByVal nSize As Long) As Long
说明
取得一个环境变量的值
返回值
Long，载入的环境变量的长度。如指定的环境字串不存在，就返回零。如 lpBuffer 的长度不足以全部容下字串，则返回字串的全长。随后可用这个长度分配一个足够大的缓冲区
参数表

参数类型及说明
lpNameString，欲读入的环境字串的名称
lpBufferString，随同字串装载的一个缓冲区。注意预先将其初始化到合适的长度
nSizeLong，lpBuffer 的长度

Top
GetInputState
GetInputState

VB 声明
Declare Function GetInputState Lib "user32" Alias "GetInputState" () As Long
说明
判断是否存在任何待决（等待处理）的鼠标或键盘事件
返回值
Long，非零表示成功，零表示失败
注解
在 win32 下，这个函数只返回当前输入线程的状态

Top
GetKBCodePage
GetKBCodePage

VB 声明
Declare Function GetKBCodePage Lib "user32" Alias "GetKBCodePage" () As Long
说明
由 GetOEMCP 取代，两者功能完全相同

Top
GetKeyboardLayout
GetKeyboardLayout

VB 声明
Declare Function GetKeyboardLayout Lib "user32" Alias "GetKeyboardLayout" (ByVal dwLayout As Long) As Long
说明
取得一个句柄，描述指定应用程序的键盘布局
返回值
Long，键盘布局的句柄
参数表
参数类型及说明
dwLayoutLong，欲检查的线程的标识符

Top
GetKeyboardLayoutList

GetKeyboardLayoutList

VB 声明
Declare Function GetKeyboardLayoutList Lib "user32" Alias "GetKeyboardLayoutList" (ByVal nBuff As Long, lpList As Long) As Long
说明
获得系统适用的所有键盘布局的一个列表
返回值
Long，装载到内存的键盘布局的数量
参数表
参数类型及说明
nBuffLong，lpList 数组中的条目数量。如设为零，表示获取可用键盘布局的数量
lpListLong，指定一个数组，它的元素数量至少应有 nBuff 规定的元素那么多。这个数组会随同句柄载入可用的键盘布局

Top
GetKeyboardLayoutName
GetKeyboardLayoutName

VB 声明
Declare Function GetKeyboardLayoutName Lib "user32" Alias "GetKeyboardLayoutNameA" (ByVal pwszKLID As String) As Long
说明
取得当前活动键盘布局的名称
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pwszKLIDString，长度为 KL_NAMELENGTH 个字符的字串
注解
在 NT 中，键盘布局与特定的应用程序有关。而在 windows95 中，它取决于特定的线程

Top
GetKeyboardState
GetKeyboardState

VB 声明
Declare Function GetKeyboardState Lib "user32" Alias "GetKeyboardState" (pbKeyState As Byte) As Long
说明
取得键盘上每个虚拟键当前的状态
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表

参数类型及说明
pbKeyStateLong，总共含 256 个条目的字节数组中的第一个项目。每个字节都会附带载入与虚拟键对应的状态。如开关键打开，则位 0 设为 1（开关键包括 CapsLock， NumLock，ScrollLock）；如某个键当时按下，则位 7 为 1；如已经抬起，则为 0
注解
虚拟键码常数 VK_? 作为数组的索引使用。这个函数相应于取得按键状态于一瞬间的“快照”——键按下或松开以后，数组不会自动更新。在 win32 中注意用一个字节数组避免由于 vb 向 Unicode 的内部转换而导致错误

Top
GetKeyboardType
GetKeyboardType

VB 声明
Declare Function GetKeyboardType Lib "user32" Alias "GetKeyboardType" (ByVal nTypeFlag As Long) As Long
说明
了解与正在使用的键盘有关的信息
返回值
Long，零表示出错。否则返回下述值之一
nTypeFlag=01——PC 或兼容的 83 键键盘；2——Olivetti102 键键盘；3——AT 或兼容 84 键键盘；4——增强型（IBM）101 或 102 键键盘；5——Nokia1050 键盘；6——Nokia9140 键盘；7——日文键盘
nTypeFlag=1 任何值，由厂商决定
nTypeFlag=21——10 个功能键（即 F? 键）；2——12 或 18 个功能键；3——10 个功能键；4——12 个功能键；5——10 个功能键；6——24 个功能键；7——由厂商决定
参数表
参数类型及说明
nTypeFlagLong，可设为下述值之一
0——返回键盘类型
1——返回键盘子类型
2——返回键盘上的功能键数量

Top
GetKeyNameText
GetKeyNameText

VB 声明
Declare Function GetKeyNameText Lib "user32" Alias "GetKeyNameTextA" (ByVal lParam As Long, ByVal lpBuffer As String, ByVal nSize As Long) As Long
说明
在给出扫描码的前提下，判断键名
返回值
Long，lpBuffer 中载入的键名的实际长度

参数表
参数类型及说明
lParamLong，位 0 到 5=0；位 16 到 23=按键的扫描码；位 24=增强型键盘上的扩展位；位 25=如设为 1，表示忽略左右 Shift 和 Ctrl 键的区别
lpBufferString，字串预先初始化成至少 nSize+1 字节，以便随同键名载入
nSizeLong，字串的最大长度

Top
GetKeyState
GetKeyState

VB 声明
Declare Function GetKeyState Lib "user32" Alias "GetKeyState" (ByVal nVirtKey As Long) As Integer
说明
针对已处理过的按键，在最近一次输入信息时，判断指定虚拟键的状态
返回值
Integer，如开关键打开，则位 0 设为 1（开关键包括 CapsLock， NumLock，ScrollLock）；如某个键当时正处于按下状态，则位 15 为 1；如已经抬起，则为 0
参数表
参数类型及说明
nVirtKeyLong，欲测试的虚拟键键码。对字母、数字字符（A-Z、a-z、0-9），用它们实际的 ASCII 值

Top
GetLastError
GetLastError

VB 声明
Declare Function GetLastError Lib "kernel32" Alias "GetLastError" () As Long
说明
针对之前调用的 api 函数，用这个函数取得扩展错误信息（在 vb 里使用：在 vb 中，用 Err 对象的 GetLastError 属性获取 GetLastError 的值。这样做是必要的，因为在 api 调用返回以及 vb 调用继续执行期间，vb 有时会重设 GetLastError 的值）
返回值
Long，由 api 函数决定。请参考 api32.txt 文件，其中列出了一系列错误常数；都以 ERROR_ 前缀起头。常用的错误代码见下表
ERROR_INVALID_HANDLE 无效的句柄作为一个参数传递
ERROR_CALL_NOT_IMPLEMENTED 在 win 95 下调用专为 win nt 设计的 win32 api 函数
ERROR_INVALID_PARAMETER 函数中有个参数不正确
注解
GetLastError 返回的值通过在 api 函数中调用 SetLastError 或 SetLastErrorEx 设置。函数并无必要设置上一次错误信息，所以即使一次 GetLastError 调用返回的是零值，也不能担保函数已成功执行。只有在函数调用返回一个错误结果时，这个函数指出的错误结果才是有效的。

通常，只有在函数返回一个错误结果，而且已知函数会设置 GetLastError 变量的前提下，才应访问 GetLastError；这时能保证获得有效的结果。SetLastError 函数主要在对 api 函数进行模拟的 dll 函数中使用，所以对 vb 应用程序来说是没有意义的

Top
GetLocaleInfo
GetLocaleInfo

VB 声明
Declare Function GetLocaleInfo Lib "kernel32" Alias "GetLocaleInfoA" (ByVal Locale As Long, ByVal LCType As Long, ByVal lpLCData As String, ByVal cchData As Long) As Long
说明
取得与指定“地方”有关的信息
返回值
Long，装载到缓冲区的字符数，或者 cchData 要求的缓冲区长度。零表示出错。会将 GetLastError 设为下述值之一：ERROR_INSUFFICIENT_BUFFER，ERROR_INVALID_FLAGS，ERROR_INVALID_PARAMETER
参数表
参数类型及说明
LocaleLong，欲为其获得信息的地方 ID
LCTypeLong，要取回的信息类型。参考 api32.txt 文件中带 LOCALE_ 前缀的常数。用 OR 运算符合并 LOCALE_NOUSEROVERRIDE，从而强制使用系统默认信息——即使当前用户已修改了相关设置
lpLCDataString，指定一个缓冲区，用于装载要求的信息。注意预先将字串格式化成合适的长度
cchDataLong，lpLCData 缓冲区的长度；如设为零，表示获取必要的缓冲区长度

Top
GetLocalTime
GetLocalTime

VB 声明
Declare Sub GetLocalTime Lib "kernel32" Alias "GetLocalTime" (lpSystemTime As SYSTEMTIME)
说明
在 lpSystemTime 结构中装载本地日期和时间
参数表
参数类型及说明
lpSystemTimeSYSTEMTIME，用于装载本地时间的结构

Top
GetNumberFormat
GetNumberFormat,GetNumberFormatBynum

VB 声明
Declare Function GetNumberFormat& Lib "kernel32" Alias "GetNumberFormatA" (ByVal Locale As Long, ByVal dwFlags As Long, ByVal lpValue As String, lpFormat As NUMBERFMT, ByVal lpNumberStr As String, ByVal cchNumber As Long)
Declare Function GetNumberFormatBynum& Lib "kernel32" Alias "GetNumberFormatA" (ByVal Locale As Long, ByVal dwFlags As Long, ByVal lpValue As String, ByVal lpFormat As Long, ByVal lpNumberStr As String, ByVal cchNumber As Long)
说明
针对指定的“地方”，按特定的格式格式化一个数字
返回值
Long，格式化过后的字串的长度。零表示出错。会将 GetLastError 设为下述值之一：ERROR_INSUFFICIENT_BUFFER，ERROR_INVALID_FLAGS，ERROR_INVALID_PARAMETER
参数表
参数类型及说明
LocaleLong，决定了具体格式的地方 ID。lpFormat 参数（如果不为 NULL）指定的任何信息都优先于各“地方”不同的特定信息
dwFlagsLong，如指定了 lpFormat，这个参数应为零。否则，可设为 LOCALE_NOUSEROVERRIDE，强制使用系统地方参数——无论用户是否已作出相应的修改
lpValueString，欲格式化的数字。数字可以只有数位、加在前面的一个“-”号以及一个小数点
lpFormatNUMBERFMT，可设为 NULL，表示使用本地特有的值（倘若用 GetNumberFormatBynum，则可将这个参数以 ByVal As Long 的形式传递）。否则，可引用一个 NUMBERFMT 结构，其中的各个字段载入欲使用的格式信息
lpNumberStrString，指定一个缓冲区，用于装载格式化过后的字串。注意先初始化成合适的长度
cchNumberLong，lpNumberStr 缓冲区的长度。如为零，函数会返回缓冲区必要的长度

Top
GetOEMCP
GetOEMCP

VB 声明
Declare Function GetOEMCP Lib "kernel32" Alias "GetOEMCP" () As Long
说明
判断在 OEM 和 ANSI 字符集间转换的 windows 代码页
返回值
Long，目前处于活动状态的 OEM 代码页的标识符。针对一种特定的语言，可能存在多个代码页。以下是可用代码页列表
437 默认：美国 708-720 阿拉伯代码页 737 希腊
775 波罗的 850 国际 852Slavic
855 西里尔语 857 土耳其语 860 葡萄牙语
861 冰岛语 862 希伯来语 863 加拿大法语

864 阿拉伯语 865 挪威/丹麦语 866 俄语
874 泰语 932 日语 936 中文（简体）
949 朝鲜语 950 中文（台、港繁体）1361 朝鲜语

Top
GetQueueStatus
GetQueueStatus

VB 声明
Declare Function GetQueueStatus Lib "user32" Alias "GetQueueStatus" (ByVal fuFlags As Long) As Long
说明
判断应用程序消息队列中待决（等待处理）的消息类型
返回值
Long，高字是一个 16 位的旗标字，包含了待决的消息。其中的各个位是由为 fuFlags 参数定义的同样的常数决定的。低字是一个对应的旗标字。其中各个位指出自上次调用这个函数以来，或自消息上一次处理以来，新加入的待处理消息
参数表
参数类型及说明
fuFlagsLong，一个标志（旗标）字，指定要检查的消息。标志位是由下述常数定义的
QS_KEYWM_CHAR 消息（会造成 vb KeyPressed 事件）
QS_MOUSE 任何鼠标消息
QS_MOUSEMOVEMouseMove 消息或事件
QS_MOUSEBUTTON 鼠标按钮消息或相关事件
QS_PAINT 等待处理的 Paint 消息
QS_POSTMESSAGE 投递的其他消息
QS_SENDMESSAGE 从另一个应用程序中发出的消息
QS_TIMER 计时器消息
QS_HOTKEY 队列中的一条 Hotkey 消息
注解
在 vb 里这个函数不特别有用（Use with VB:Not particularly useful.）

Top
GetSysColor
GetSysColor

VB 声明
Declare Function GetSysColor Lib "user32" Alias "GetSysColor" (ByVal nIndex As Long) As Long
说明
判断指定 windows 显示对象的颜色
返回值
Long，指定对象的 RGB 颜色
参数表

参数类型及说明
nIndexLong，一个常数，指出特定的 windows 显示对象，如下表
Windows 对象常数表
常数定义 Windows 对象常数定义 Windows 对象
COLOR_ACTIVEBORDER 活动窗口的边框 COLOR_ACTIVECAPTION 活动窗口的标题
COLOR_APPWORKSPACEMDI 桌面的背景 COLOR_BACKGROUNDwindows 桌面
COLOR_BTNFACE 按钮 COLOR_BTNHIGHLIGHT 按钮的 3D 加亮区
COLOR_BTNSHADOW 按钮的 3D 阴影 COLOR_BTNTEXT 按钮文字
COLOR_CAPTIONTEXT 窗口标题中的文字 COLOR_GRAYTEXT 灰色文字；如使用了抖动技术则为零
COLOR_HIGHLIGHT 选定的项目背景 COLOR_HIGHLIGHTTEXT 选定的项目文字
COLOR_INACTIVEBORDER 不活动窗口的边框 COLOR_INACTIVECAPTION 不活动窗口的标题
COLOR_INACTIVECAPTIONTEXT 不活动窗口的文字 COLOR_MENU 菜单
COLOR_MENUTEXT 菜单正文 COLOR_SCROLLBAR 滚动条
COLOR_WINDOW 窗口背景 COLOR_WINDOWFRAME 窗框
COLOR_WINDOWTEXT 窗口正文 COLOR_3DDKSHADOW3D 深阴影 *
COLOR_3DFACE3D 阴影化对象的正面颜色 *COLOR_3DHILIGHT3D 加亮颜色（win95）
COLOR_3DLIGHT3D 阴影化对象的浅色 *COLOR_INFOBK 工具提示的背景色 *
COLOR_INFOTEXT 工具提示的文本色 *
*：带 * 号的常数未获 NT 3.51 的支持

Top
GetSystemDefaultLangID
GetSystemDefaultLangID

VB 声明
Declare Function GetSystemDefaultLangID Lib "kernel32" Alias "GetSystemDefaultLangID" () As Integer
说明
取得系统的默认语言 ID
返回值
Integer，系统的默认语言 ID

Top
GetSystemDefaultLCID
GetSystemDefaultLCID

VB 声明
Declare Function GetSystemDefaultLCID Lib "kernel32" Alias "GetSystemDefaultLCID" () As Long
说明
取得当前的默认系统“地方”
返回值

Long，默认的系统地方 ID

Top
GetSystemInfo
GetSystemInfo

VB 声明
Declare Sub GetSystemInfo Lib "kernel32" Alias "GetSystemInfo" (lpSystemInfo As SYSTEM_INFO)
说明
在一个 SYSTEM_INFO 结构中载入与底层硬件平台有关的信息
参数表
参数类型及说明
lpSystemInfoSYSTEM_INFO，指定一个结构，用于装载适当的系统信息

Top
GetSystemMetrics
GetSystemMetrics

VB 声明
Declare Function GetSystemMetrics Lib "user32" Alias "GetSystemMetrics" (ByVal nIndex As Long) As Long
说明
返回与 windows 环境有关的信息
返回值
Long，取决于具体的常数索引
参数表
参数类型及说明
nIndexLong，常数，指定欲获取的信息；如下表所示
nIndex 常数设置
常数定义取得信息
SM_ARRANGE 设置 windows 如何排列最小化窗口的一个标志。参考 api32.txt 中的 ARW 常数
SM_CLEANBOOT 指定启动模式。0=普通模式；1=带网络支持的安全模式
SM_CMETRICS 可用系统环境的数量
SM_CMOUSEBUTTON 鼠标按钮（按键）的数量。如没有鼠标，就为零
SM_CXBORDER, SM_CYBORDER 尺寸不可变边框的大小
SM_CXCURSOR, SM_CYCURSOR 标准指针大小
SM_CXDLGFRAME, SM_CYDLGFRAME 对话框边框的大小
SM_CXDOUBLECLK, SM_CYDOUBLECLK 双击区域的大小（参考注解）
SM_CXFRAME, SM_CYFRAME 尺寸可变边框的大小（在 win95 和 nt 4.0 中使用 SM_C?FIXEDFRAME）
SM_CXFULLSCREEN, SM_CYFULLSCREEN 最大化窗口客户区的大小
SM_CXHSCROLL, SM_CYHSCROLL 水平滚动条上的箭头大小

SM_CXHTHUMB, SM_CYHTHUMB 滚动块在水平滚动条上的大小
SM_CXICON, SM_CYICON 标准图标的大小
SM_CXICONSPACING, SM_CYICONSPACING 桌面图标之间的间隔距离。在 win95 和 nt 4.0 中是指大图标的间距
SM_CXMAXIMIZED, SM_CYMAXIMIZED 最大化窗口的默认尺寸
SM_CXMAXTRACK, SM_CYMAXTRACK 改变窗口大小时，最大的轨迹宽度
SM_CXMENUCHECK, SM_CYMENUCHECK 菜单复选号位图的大小
SM_CXMENUSIZE, SM_CYMENUSIZE 菜单栏上的按钮大小
SM_CXMIN, SM_CYMIN 窗口的最小尺寸
SM_CXMINIMIZED, SM_CYMINIMIZED 最小化的窗口必须填充进去的一个矩形小于或等于 SM_C?ICONSPACING
SM_CXMINTRACK, SM_CYMINTRACK 窗口的最小轨迹宽度
SM_CXSCREEN, SM_CYSCREEN 屏幕大小
SM_CXSIZE, SM_CYSIZE 标题栏位图的大小
SM_CXSIZEFRAME, SM_CYSIZEFRAME 具有 WS_THICKFRAME 样式的窗口的大小
SM_CXSMICON, SM_CYSMICON 小图标的大小
SM_CXSMSIZE, SM_CYSMSIZE 小标题按钮的大小
SM_CXVSCROLL, SM_CYVSCROLL 垂直滚动条中的箭头按钮的大小
SM_CYCAPTION 窗口标题的高度
SM_CYKANJIWINDOWKanji 窗口的大小（Height of Kanji window）
SM_CYMENU 菜单高度
SM_CYSMCAPTION 小标题的高度
SM_CYVTHUMB 垂直滚动条上滚动块的高度
SM_DBCSENABLED 如支持双字节则为 TRUE
SM_DEBUG 如 windows 的调试版正在运行，则为 TRUE
SM_MENUDROPALIGNMENT 如弹出式菜单对齐菜单栏项目的左侧，则为零
SM_MIDEASTENABLED 允许了希伯来和阿拉伯语
SM_MOUSEPRESENT 如安装了鼠标则为 TRUE
SM_MOUSEWHEELPRESENT 如安装了带轮鼠标则为 TRUE；只适用于 nt 4.0
SM_NETWORK 如安装了网络，则设置位 0。其他位保留未用
SM_PENWINDOWS 如装载了支持笔窗口的 DLL，则表示笔窗口的句柄
SM_SECURE 如安装了安全（保密）机制，则为 TRUE
SM_SHOWSOUNDS 强制视觉提示播放声音
SM_SLOWMACHINE 系统速度太慢，但仍在运行中（System is too slow for effective use but is being run anyway）
SM_SWAPBUTTON 如左右鼠标键已经交换，则为 TRUE
注解
双击区域指定屏幕上一个特定的显示区域，只有在这个区域内连续进行两次鼠标单击，才有可能被当作双击事件处理
其他
常数 SM_ARRANGE, SM_CLEANBOOT, SM_CMETRICS, SM_C?MAXIMIZED, SM_C?MAXTRACK, SM_C?SIZEFRAME, SM_C?SMICON, SM_C?SMSIZE, SM_CYSMCAPTION, SM_SECURE, SM_SHOWSOUNDS, and SM_SLOWMACHINE 未获 NT 3.51 及更早版本的支持

Top
GetSystemPowerStatus
GetSystemPowerStatus

VB 声明
Declare Function GetSystemPowerStatus Lib "kernel32" Alias "GetSystemPowerStatus" (lpSystemPowerStatus As SYSTEM_POWER_STATUS) As Long
说明
获得与当前系统电源状态有关的信息。对便携式计算机来说，这些信息特别有用。在那些地方，可用这个函数了解有关电源和电池组的情况
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
lpSystemPowerStatusSYSTEM_POWER_STATUS，用于装载电源状态信息的结构
注解
结果的准确度取决于系统实际的电源管理能力

Top
GetSystemTime
GetSystemTime

VB 声明
Declare Sub GetSystemTime Lib "kernel32" Alias "GetSystemTime" (lpSystemTime As SYSTEMTIME)
说明
在一个 SYSTEMTIME 中载入当前系统时间，这个时间采用的是“协同世界时间”（即 UTC，也叫做 GMT）格式
参数表
参数类型及说明
lpSystemTimeSYSTEMTIME，随同当前时间载入的结构

Top
GetSystemTimeAdjustment
GetSystemTimeAdjustment

VB 声明
Declare Function GetSystemTimeAdjustment Lib "kernel32" Alias "GetSystemTimeAdjustment" (lpTimeAdjustment As Long, lpTimeIncrement As Long, lpTimeAdjustmentDisabled As Boolean) As Long
说明
Win32 可使内部系统时钟与一个外部的时钟信号源同步，方法是定时添加一个校准值。这个函数指定的所有时间都以 100ns（0.1ms）为单位递增

返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
lpTimeAdjustmentLong，添加到内部系统时钟的时间量
lpTimeIncrementLong，校准间隔时间。等于中断时钟时间
lpTimeAdjustmentDisabledBoolean，如时间调校功能关闭，则为 TRUE
示例
Dim lpTimeAdjustment&, lpTimeIncrement&, lpTimeAdjustmentDisabled&
GetSystemTimeAdjustment lpTimeAdjustment, lpTimeIncrement, lpTimeAdjustmentDisabled
Print "Time adjustment: " & lpTimeAdjustment & "Increment: " & lpTimeIncrement & "Adjustment Disabled: " & lpTimeAdjustmentDisabled

Top
GetThreadLocale
GetThreadLocale

VB 声明
Declare Function GetThreadLocale Lib "KERNEL32" Alias "GetThreadLocale" () As Long
说明
取得当前线程的地方 ID
返回值
Long，地方标识符
注解
在 vb 下，除非从一个多线程 EXE 服务器调用，否则当前线程就是当前应用程序

Top
GetTickCount
GetTickCount

VB 声明
Declare Function GetTickCount Lib "kernel32" Alias "GetTickCount" () As Long
说明
用于获取自 windows 启动以来经历的时间长度（毫秒）
返回值
Long，以毫秒为单位的 windows 运行时间

Top
GetTimeFormat
GetTimeFormat

VB 声明
Declare Function GetTimeFormat Lib "kernel32" Alias "GetTimeFormatA" (ByVal Locale As Long, ByVal dwFlags As Long, lpTime As SYSTEMTIME, ByVal lpFormat As String, ByVal

lpTimeStr As String, ByVal cchTime As Long) As Long
说明
针对当前指定的“地方”，按特定的格式格式化一个系统时间
返回值
Long，格式化过后的字串的长度。零表示出错。会将 GetLastError 设置为下述值之一：ERROR_INSUFFICIENT_BUFFER ， ERROR_INVALID_FLAGS ， ERROR_INVALID_PARAMETER
参数表
参数类型及说明
LocaleLong，决定了具体格式的地方 ID。lpFormat 参数中指定的任何信息（倘若不是 NULL）都要优先于各地方不同的特别信息
dwFlagsLong ， 如 指 定 了 lpFormat ， 那 么 该 参 数 应 该 为 零 。 否 则 ， 可 设 为 LOCALE_NOUSEROVERRIDE，表示强制使用系统地方参数——即使它们已由用户更改。用 DATE_SHORTDATE 或 DATE_LONGDATE 选择不同的日期格式
lpTimeSYSTEMTIME，用于包容系统时间的一个结构
lpFormatString，可设为 NULL，使用特定于不同地方的值（用 vbNullString 传递一个 NULL）。否则包含一个时间格式字串。对 h,hh,hhh,hhhh,m,mm,s,ss 这样的代码来说，它们的用法与在 vb 格式命令中的用法是相同的。t 和 tt 用于指定一个时间段标志（A 或 AM，P 或 PM）
lpTimeStrString，指定一个缓冲区，用于容纳格式化过后的字串。注意事先对字串的长度进行正确的预初始化
cchTimeLong，lpTimeStr 缓冲区的长度。如为零，表示函数会返回需要缓冲区的大小
注解
对于 dwFlags 参数的解释可能翻译有错误。原文为：
If lpFormat is specified, this should be zero. Otherwise, may be set to LOCALE_NOUSEROVERRIDE to force the system locale parameters to be used even if they have been overridden by the user. Use the self-explanatory constants TIME_NOMINUITESORSECONDS, TIME_NOSECONDS, or TIME_FORCE24HOURFORMAT to choose between date formats. Constant TIME_NOMARKER removes the AM or PM marker.

Top
GetTimeZoneInformation
GetTimeZoneInformation

VB 声明
Declare Function GetTimeZoneInformation Lib "kernel32" Alias "GetTimeZoneInformation" (lpTimeZoneInformation As TIME_ZONE_INFORMATION) As Long
说明
在一个 TIME_ZONE_INFORMATION 结构中载入与系统时区设置有关的信息
返回值
Long，下述常数之一：
TIME_ZONE_ID_INVALID 函数执行失败，会设置 GetLastError
TIME_ZONE_ID_UNKNOWN 时区未知（可能仍然指定了 bias 值）
TIME_ZONE_ID_STANDARD 标准时间有效

TIME_ZONE_ID_DAYLIGHT 夏令时有效
参数表
参数类型及说明
lpTimeZoneInformationTIME_ZONE_INFORMATION，用于载入时区信息的结构
注解
在 lpTimeZoneInformation 结构中为本地时间添加 bias 信息，从而获得系统时间。TIME_ZONE_INFORMATION 结构内部的 DaylightName 和 StandardName 字串肯定采用 Unicode 格式

Top
GetUserDefaultLangID
GetUserDefaultLangID

VB 声明
Declare Function GetUserDefaultLangID Lib "kernel32" Alias "GetUserDefaultLangID" () As Integer
说明
为当前用户取得默认语言 ID
返回值
Long，当前用户的语言 ID

Top
GetUserDefaultLCID
GetUserDefaultLCID

VB 声明
Declare Function GetUserDefaultLCID Lib "kernel32" Alias "GetUserDefaultLCID" () As Long
说明
取得当前用户的默认“地方”设置
返回值
Long，针对当前用户的默认地方 ID

Top
GetUserName
GetUserName

VB 声明
Declare Function GetUserName Lib "advapi32.dll" Alias "GetUserNameA" (ByVal lpBuffer As String, nSize As Long) As Long
说明
取得当前用户的名字
返回值
Long，TRUE（非零）表示成功，否则返回零。会设置 GetLastError
参数表

参数类型及说明
lpBufferString，一个字串缓冲区，预先初始化成由 nSize 指定的长度。它将用于容纳用户名
nSizeLong，初始化成 lpBuffer 的长度。返回以后，它会包含载入 lpBuffer 的字符数量
示例

```
Dim s$, cnt&, dl&
cnt& = 199
s$ = String$(200,0)
dl& = GetUserName(s$, cnt)
Debug.Print Left$(s$, cnt); cnt
```

Top
GetVersion
GetVersion

VB 声明

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

说明
判断当前运行的 Windows 和 DOS 版本
返回值
Long，低 16 位包含了 windows 版本；低字节包含了主版本号（3 代表 windows 3.10，4 代表 nt 4.0）；高字节包含了两个数位的辅助版本号（10 代表 windows 3.10，95 代表 windows 95）。高 16 位则包含了平台信息。针对 windows NT，高位设为 0；针对 windows for workgroups 上运行的 Win32s，则高位为 1
注解
在 win32 下，最好换用 GetVersionEx 函数。在 win32 下，高字不会返回 DOS 版本

Top
GetVersionEx
GetVersionEx

VB 声明

```
Declare Function GetVersionEx Lib "kernel32" Alias "GetVersionExA" (ByVal lpVersionInformation As OSVERSIONINFO) As Long
```

说明
在一个 OSVERSIONINFO 结构中载入与平台和操作系统有关的版本信息
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
lpVersionInformationOSVERSIONINFO，用于装载版本信息的结构。在正式调用函数之前，必须先将这个结构的 dwOSVersionInfoSize 字段设为结构的大小（148）

Top
HideCaret

HideCaret

VB 声明
Declare Function HideCaret Lib "user32" Alias "HideCaret" (ByVal hwnd As Long) As Long
说明
在指定的窗口隐藏插入符（光标）
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，包含了插入符的窗口的句柄。可能是零；在这种情况下，只有包容了插入符的那个窗口由活动任务拥有时，插入符才会被隐藏
注解
针对插入符的显示，windows 维护着一个内部计数器；类似于 ShowCursor 函数使用的那个。所以对 HideCaret 和 ShowCaret 的调用必须进行一番权衡

Top
IsValidCodePage
IsValidCodePage

VB 声明
Declare Function IsValidCodePage Lib "kernel32" Alias "IsValidCodePage" (ByVal CodePage As Long) As Long
说明
判断一个代码页是否有效
返回值
Long，如代码页有效，就返回 TRUE（非零）；否则返回零
参数表
参数类型及说明
CodePageLong，一个代码页的标识符。参考 GetACP 和 GetOEMCP 函数

Top
IsValidLocale
IsValidLocale

VB 声明
Declare Function IsValidLocale Lib "KERNEL32" Alias "IsValidLocale" (ByVal Locale As Long, ByVal dwFlags As Long) As Long
说明
判断地方标识符是否有效
返回值
Long，根据测试条件，如指定的地方有效，就返回 TRUE（非零）；否则返回零
参数表
参数类型及说明

LocaleLong，要检查的地方标识符
dwFlagsLong，下述常数之一：
LCID_SUPPORTED 检查“地方”是否能由系统支持
LCID_INSTALLED 检查“地方”是否已获支持并安装

Top
keybd_event
keybd_event

VB 声明
Declare Sub keybd_event Lib "user32" Alias "keybd_event" (ByVal bVk As Byte, ByVal bScan As Byte, ByVal dwFlags As Long, ByVal dwExtraInfo As Long)
说明
这个函数模拟了键盘行动
参数表
参数类型及说明
bVkByte，欲模拟的虚拟键码
bScanByte，键的 OEM 扫描码
dwFlagsLong，零；或设为下述两个标志之一
KEYEVENTF_EXTENDEDKEY 指出是一个扩展键，而且在前面冠以 0xE0 代码
KEYEVENTF_KEYUP 模拟松开一个键
dwExtraInfoLong，通常不用的一个值。api 函数 GetMessageExtraInfo 可取得这个值。允许使用的值取决于特定的驱动程序
注解
这个函数支持屏幕捕获（截图）。在 win95 和 nt4.0 下这个函数的行为不同

Top
LoadKeyboardLayout
LoadKeyboardLayout

VB 声明
Declare Function LoadKeyboardLayout Lib "user32" Alias "LoadKeyboardLayoutA" (ByVal pwszKLID As String, ByVal flags As Long) As Long
说明
载入一个键盘布局
返回值
Long，键盘布局的句柄。零表示出错
参数表
参数类型及说明
pwszKLIDString，一个 8 字符字串，用于描述键盘布局的名称。参考注解
flagsLong，下述常数的任何一种组合
KLF_ACTIVATE 载入和激活指定的布局
KLF_NOTELLSHELL 禁止一个外壳挂钩进程（a shell hook procedure）接收到 HSHELL_LANGUAGE 通告。如准备载入一系列键盘布局，就需要考虑设置这个标志，从

而改善性能（不要为最后一个载入的布局设置该标志）
KLF_REORDER 将指定的活动布局移至内部键盘布局列表的起始处
KLF_REPLACELANG 如指定语言的键盘布局已经存在，则用这个将其替换。仅适用于 win95
KLF_SUBSTITUTE_OK 在注册表中使用替换信息，为这个语言载入一个由用户指定的替换键盘布局（如果存在的话），而不是载入当前这个布局
KLF_UNLOADPREVIOUS 如 KLF_ACTIVATE 已经指定并成功，则卸载前一个布局
注解
键盘布局的名称采用“ddddnnnn”的形式。其中，nnnn 代表一个语言 ID 的字串形式，而 dddd 代表一个设备代码的字串形式。标准的美国键盘名称是“00000409”
其他
键盘布局在 win95 中取决于特定的线程；在 windows nt 中，则在整个系统的范围内有效

Top
MapVirtualKey
MapVirtualKey

VB 声明
Declare Function MapVirtualKey Lib "user32" Alias "MapVirtualKeyA" (ByVal wCode As Long, ByVal wMapType As Long) As Long
说明
根据指定的映射类型，执行不同的扫描码和字符转换
返回值
Long，取决于 wMapType 参数
参数表
参数类型及说明
wCodeLong，欲转换的源字符或扫描码
wMapTypeLong，控制映射类型，如下所示
0—— wCode 是个虚拟键码。函数返回相应的扫描码
1—— wCode 是个扫描码。函数返回相应的虚拟键码
2—— wCode 是个虚拟键码。函数返回相应的 ASCII 值（未加 Shift 组合键）

Top
MapVirtualKeyEx
MapVirtualKeyEx

VB 声明
Declare Function MapVirtualKeyEx Lib "user32" Alias "MapVirtualKeyExA" (ByVal uCode As Long, ByVal uMapType As Long, ByVal dwhkl As Long) As Long
说明
根据指定的映射类型，执行不同的扫描码和字符转换
返回值
Long，取决于 uMapType 参数
参数表
参数类型及说明

uCodeLong，欲转换的源字符或代码
uMapTypeLong，控制映射类型，如下所示
0—— uCode 是个虚拟键码。函数返回相应的扫描码
1—— uCode 是个扫描码。函数返回相应的虚拟键码
2—— uCode 是个虚拟键码。函数返回相应的 ASCII 值（未加 Shift 组合键）。针对死键，高位设为 1。如果出错，返回 NULL
dwhklLong，键盘布局的句柄
注解
利用这个函数，可在扫描码及附加的虚拟键码间转换。这些虚拟键码包括 VK_LSHIFT 和 VK_RSHIFT 等。这样一来，便可为表面上两个完全一样的键（也包括 Ctrl 和 Alt）区分左键和右键

Top
MessageBeep
MessageBeep

VB 声明
Declare Function MessageBeep Lib "user32" Alias "MessageBeep" (ByVal wType As Long) As Long
说明
播放一个系统声音。系统声音的分配方案是在控制面板里决定的
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
wTypeLong，下述值之一
0xffffffff 标准响铃
MB_ICONASTERISK 系统星号声（System asterisk sound）
MB_ICONEXCLAMATION 系统惊叹声
MB_ICONHAND 系统指针声（System hand sound）
MB_ICONQUESTION 系统提问声

Top
mouse_event
mouse_event

VB 声明
Declare Sub mouse_event Lib "user32" Alias "mouse_event" (ByVal dwFlags As Long, ByVal dx As Long, ByVal dy As Long, ByVal cButtons As Long, ByVal dwExtraInfo As Long)
说明
模拟一次鼠标事件
参数表
参数类型及说明
dwFlagsLong，下述标志的一个组合

MOUSEEVENTF_ABSOLUTEdx 和 dy 指定鼠标坐标系统中的一个绝对位置。在鼠标坐标系统中，屏幕在水平和垂直方向上均匀分割成 65535×65535 个单元
MOUSEEVENTF_MOVE 移动鼠标
MOUSEEVENTF_LEFTDOWN 模拟鼠标左键按下
MOUSEEVENTF_LEFTUP 模拟鼠标左键抬起
MOUSEEVENTF_RIGHTDOWN 模拟鼠标右键按下
MOUSEEVENTF_RIGHTUP 模拟鼠标右键按下
MOUSEEVENTF_MIDDLEDOWN 模拟鼠标中键按下
MOUSEEVENTF_MIDDLEUP 模拟鼠标中键按下
dxLong，根据是否指定了 MOUSEEVENTF_ABSOLUTE 标志，指定水平方向的绝对位置或相对运动
dyLong，根据是否指定了 MOUSEEVENTF_ABSOLUTE 标志，指定垂直方向的绝对位置或相对运动
cButtonsLong，未使用
dwExtraInfoLong，通常未用的一个值。用 GetMessageExtraInfo 函数可取得这个值。可用的值取决于特定的驱动程序
注解
进行相对运动的时候，由 SystemParametersInfo 函数规定的系统鼠标轨迹速度会应用于鼠标运行的速度

Top
OemKeyScan
OemKeyScan

VB 声明
Declare Function OemKeyScan Lib "user32" Alias "OemKeyScan" (ByVal wOemChar As Long) As Long
说明
判断 OEM 字符集中的一个 ASCII 字符的扫描码和 Shift 键状态
返回值
Long，低字包含了扫描码。高字包含了下述标志：位 0 标志着 Shift 已经按下；位 1 标志着 Ctrl 键按下；位 2 标志着 Alt 键按下。如两个字都为-1，表明字符未在 OEM 字符集中得到定义
参数表
参数类型及说明
wOemCharLong，欲转换的字符的 ASCII 值
注解
这个函数只能转换那些单击键就能生成的字符。倘若字符要通过“Alt+3 数位”（比如 Alt+255）才能出现，则不能用这个函数转换
原文：This function only translates keys that can be typed with a single keystroke. Keys that are entered using the ALT + 3 digit entry cannot be converted with this function.

Top
OemToChar

OemToChar, OemToCharBuff

VB 声明
Declare Function OemToChar& Lib "user32" Alias "OemToCharA" (ByVal lpszSrc As String, ByVal lpszDst As String)
Declare Function OemToCharBuff& Lib "user32" Alias "OemToCharBuffA" (ByVal lpszSrc As String, ByVal lpszDst As String, ByVal cchDstLength As Long)
说明
将 OEM 字符集的一个字串转换到 ANSI 字符集。OemToCharBuff 允许我们指定字串中欲转换的字符数量
返回值
Long，肯定是 TRUE
参数表
参数类型及说明
lpszSrcString，欲转换的字串
lpszDstString，用于容纳 ANSI 转换结果的一个字串。注意先将字串初始化成合适的长度。可将相同的字串传递给两个参数，以便在同一个字串中进行转换
cchDstLengthLong，字串中要转换的字符数量
注解
如使用一个 Win32 类型库访问宽字符函数 OemToCharW，那么 lpszSrc 是个 Unicode 字串，而且 lpszDst 参数绝对不能与 lpszSrc 相同

Top
SetCaretBlinkTime
SetCaretBlinkTime

VB 声明
Declare Function SetCaretBlinkTime Lib "user32" Alias "SetCaretBlinkTime" (ByVal wMSeconds As Long) As Long
说明
指定插入符（光标）的闪烁频率
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
wMSecondsLong，插入符新的闪烁间隔时间，以毫秒为单位
注解
插入符或插入标志是一种共享资源，所以闪烁时间设定会对所有应用程序的插入符产生影响。可用 GetCaretBlinkTime 函数获得最初的闪烁时间设置。以后在适当的时候，可用这个设置恢复最开始的值。参考 CreateCaret 函数的注解

Top
SetCaretPos
SetCaretPos

VB 声明
Declare Function SetCaretPos Lib "user32" Alias "SetCaretPos" (ByVal x As Long, ByVal y As Long) As Long
说明
指定插入符的位置
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
x,yLong，插入符在客户区坐标系统中的 X，Y 位置
注解
插入符是一种共享资源。在更改插入符的位置之前，程序员应确定插入符位于由当前任务拥有的一个窗口内。参考 CreateCaret 函数的注解

Top
SetComputerName
SetComputerName

VB 声明
Declare Function SetComputerName Lib "kernel32" Alias "SetComputerNameA" (ByVal lpComputerName As String) As Long
说明
设置新的计算机名
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpComputerNameString，新的计算机名称。最多可有 MAX_COMPUTERNAME_LENGTH 个字符
注解
windows95 会将任何非法字符自动转换到标准的字符集里。windows nt 则会报告出错

Top
SetCursor
SetCursor

VB 声明
Declare Function SetCursor Lib "user32" Alias "SetCursor" (ByVal hCursor As Long) As Long
说明
将指定的鼠标指针设为当前指针
返回值
Long，前一个指针的值
参数表

参数类型及说明
hCursorLong，要设为当前指针的一个指针的句柄。如设为零，表示不显示任何指针
注解
在 vb 里这个函数不能很好的工作，因为 vb 习惯在不同的时间将指针变回原来的样子

Top
SetCursorPos
SetCursorPos

VB 声明
Declare Function SetCursorPos Lib "user32" Alias "SetCursorPos" (ByVal x As Long, ByVal y As Long) As Long
说明
设置指针的位置
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
x,y 鼠标指针在屏幕像素坐标系统中的 X，Y 位置

Top
SetDoubleClickTime
SetDoubleClickTime

VB 声明
Declare Function SetDoubleClickTime Lib "user32" Alias "SetDoubleClickTime" (ByVal wCount As Long) As Long
说明
设置连续两次鼠标单击之间能使系统认为是双击事件的间隔时间
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
wCountLong，新的 DoubleClick 间隔时间，以毫秒为单位
注解
双击时间设定的变化会影响整个系统

Top
SetEnvironmentVariable
SetEnvironmentVariable

VB 声明
Declare Function SetEnvironmentVariable Lib "kernel32" Alias "SetEnvironmentVariableA" (ByVal lpName As String, ByVal lpValue As String) As Long

说明
将一个环境变量设为指定的值
返回值
Long，TRUE（非零）表示成功，否则返回零。会设置 GetLastError
参数表
参数类型及说明
lpNameString，欲设置的环境变量的名称。如具有这个名称的一个环境变量尚不存在，则函数会自动创建它
lpValueString，欲为变量设置的新值。如为 NULL，表示清除一个现成值（用 vbNullString 常数向这个函数传递一个 NULL）

Top
SetKeyboardState
SetKeyboardState

VB 声明
Declare Function SetKeyboardState Lib "user32" Alias "SetKeyboardState" (lppbKeyState As Byte) As Long
说明
设置每个虚拟键当前在键盘上的状态
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lppbKeyStateByte，固定为 256 字符长度的一个字串。windows 内部键盘状态表中的每个字符都会根据这张表中对应的虚拟键设置。每个键的状态与 GetKeyboardState 函数的结果是相同的
注解
可用这个函数设置 CapsLock， NumLock 和 ScrollLock 键的状态

Top
SetLocaleInfo
SetLocaleInfo

VB 声明
Declare Function SetLocaleInfo Lib "kernel32" Alias "SetLocaleInfoA" (ByVal Locale As Long, ByVal LCType As Long, ByVal lpLCData As String) As Long
说明
改变用户“地方”设置信息
返回值
Long，TRUE（非零）表示成功，否则返回零。会将 GetLastError 设置为下述值之一：ERROR_INVALID_ACCESS，ERROR_INVALID_FLAGS，ERROR_INVALID_PARAMETER
参数表
参数类型及说明

LocaleLong，要为其改变信息的地方 ID
LCTypeLong，欲改变的信息类型。参考 api32.txt，检视那些带 LOCALE_ 前缀的常数
lpLCDataString，这个地方信息项目的新设置
注解
这个函数不会改变系统地方设置

Top
SetLocalTime
SetLocalTime

VB 声明
Declare Function SetLocalTime Lib "kernel32" Alias "SetLocalTime" (lpSystemTime As SYSTEMTIME) As Long
说明
设置当前地方时间
返回值
Long，TRUE（非零）表示成功，否则返回零。会设置 GetLastError
参数表
参数类型及说明
lpSystemTimeSYSTEMTIME，这个结构指定了新的地方时间。该结构中的 wDayOfWeek 条目会被忽略

Top
SetSysColors
SetSysColors

VB 声明
Declare Function SetSysColors Lib "user32" Alias "SetSysColors" (ByVal nChanges As Long, lpSysColor As Long, lpColorValues As Long) As Long
说明
设置指定窗口显示对象的颜色
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
nChangesLong，欲改变的对象的数量
lpSysColorLong，按引用传递。这是一个整数数组（总共包含 nChanges 个元素）的第一个元素。每个条目都包含了一个常数，指定一个 windows 显示对象。参考 GetSysColor 函数
lpColorValuesLong，按引用传递。这是 RGB 值数组的第一个元素；该数组用于设置 lpSysColor 数组中的对象颜色

Top
SetSystemCursor
SetSystemCursor

VB 声明
Declare Function SetSystemCursor Lib "user32" Alias "SetSystemCursor" (ByVal hcur As Long, ByVal id As Long) As Long
说明
改变任何一个标准系统指针
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hcurLong，新指针
idLong，以 OCR_ 前缀起头的一个常数，用于指定标准系统指针
注解
不要破坏由 hcur 指定的指针——在必要的时候，它会由系统自行清除

Top
SetSystemTime
SetSystemTime

VB 声明
Declare Function SetSystemTime Lib "kernel32" Alias "SetSystemTime" (lpSystemTime As SYSTEMTIME) As Long
说明
设置当前系统时间
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpSystemTimeSYSTEMTIME，这个结构指定了新的地方时间。其中的 wDayOfWeek 条目会被忽略

Top
SetSystemTimeAdjustment
SetSystemTimeAdjustment

VB 声明
Declare Function SetSystemTimeAdjustment Lib "kernel32" Alias "SetSystemTimeAdjustment" (ByVal dwTimeAdjustment As Long, ByVal bTimeAdjustmentDisabled As Boolean) As Long
说明
Win32 可使内部系统时钟与一个外部的时钟信号源同步，方法是定时添加一个校准值。这个函数指定的所有时间都以 100ns（0.1ms）为单位递增
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表

参数类型及说明
dwTimeAdjustmentLong，每次时钟中断时添加到内部系统时钟的时间量
bTimeAdjustmentDisabledBoolean，TRUE 用于禁止时间调校

Top
SetThreadLocale
SetThreadLocale

VB 声明
Declare Function SetThreadLocale Lib "kernel32" Alias "SetThreadLocale" (ByVal Locale As Long) As Long
说明
为当前线程设置地方
返回值
Long，TRUE（非零）表示成功，否则返回零。会设置 GetLastError
参数表
参数类型及说明
LocaleLong，这个线程使用的地方 ID
注解
适用平台——Windows NT

Top
SetTimeZoneInformation
SetTimeZoneInformation

VB 声明
Declare Function SetTimeZoneInformation Lib "kernel32" Alias "SetTimeZoneInformation" (lpTimeZoneInformation As TIME_ZONE_INFORMATION) As Long
说明
设置系统时区信息
返回值
Long，TRUE（非零）表示成功，否则返回零。会设置 GetLastError
参数表
参数类型及说明
lpTimeZoneInformationTIME_ZONE_INFORMATION，要在其中设置当前时区信息的一个结构
注解
参考 GetTimeZoneInformation

Top
ShowCaret
ShowCaret

VB 声明

Declare Function ShowCaret Lib "user32" Alias "ShowCaret" (ByVal hwnd As Long) As Long
说明
在指定的窗口里显示插入符（光标）
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，包含了插入符的窗口的句柄。可以为零；此时，只有在插入符包含于由活动任务拥有的一个窗口时，它才会显示出来
注解
参考 HideCaret 函数

Top
ShowCursor
ShowCursor

VB 声明
Declare Function ShowCursor Lib "user32" Alias "ShowCursor" (ByVal bShow As Long) As Long
说明
控制鼠标指针的可视性
返回值
Long，显示计数（参考注解）
参数表
参数类型及说明
bShowLong，TRUE（非零）显示指针，FALSE 隐藏
注解
windows 维持着一个内部显示计数；倘若 bShow 为 TRUE，那么每调用一次这个函数，计数就会递增 1；反之，如 bShow 为 FALSE，则计数递减 1。只有在这个计数大于或等于 0 的情况下，指针才会显示出来

Top
SwapMouseButton
SwapMouseButton

VB 声明
Declare Function SwapMouseButton Lib "user32" Alias "SwapMouseButton" (ByVal bSwap As Long) As Long
说明
决定是否互换鼠标左右键的功能
返回值
Long，TRUE（非零）表示鼠标按钮的功能在调用这个函数之前已经互换；否则返回零
参数表
参数类型及说明

bSwapLong，倘若为 TRUE（非零），则互换两个鼠标按钮的功能。FALSE 则恢复正常状态
注解
鼠标是一种共享资源，所以这个函数会对系统中的所有应用程序造成影响

Top
SystemParametersInfo
SystemParametersInfo, SystemParametersInfoByval

VB 声明
Declare Function SystemParametersInfo& Lib "user32" Alias "SystemParametersInfoA" (ByVal uAction As Long, ByVal uParam As Long, lpvParam As Any, ByVal fuWinIni As Long)
Declare Function SystemParametersInfoByVal& Lib "user32" Alias "SystemParametersInfoA" (ByVal uAction As Long, ByVal uParam As Long, ByVal lpvParam As Any, ByVal fuWinIni As Long)
说明
允许获取和设置数量众多的 windows 系统参数
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
uActionLong，指定要设置的参数。参考 uAction 常数表
uParamLong，参考 uAction 常数表
lpvParamAny，按引用调用的 Integer、Long 和数据结构。对于 String 数据，请用 SystemParametersInfoByval 函数。具体用法参考 uAction 常数表
fuWinIniLong，取决于不同的参数及操作系统，随同这个函数设置的用户配置参数保存在 win.ini 或注册表里，或同时保存在这两个地方。这个参数规定了在设置系统参数的时候，是否应更新用户设置参数。可以是零（禁止更新），或下述任何一个常数：
SPIF_UPDATEINIFILE 更新 win.ini 和（或）注册表中的用户配置文件
SPIF_SENDWININICHANGE 倘若也设置了 SPIF_UPDATEINIFILE，将一条 WM_WININICHANGE 消息发给所有应用程序。否则没有作用。这调消息告诉应用程序已经改变了用户配置设置
注解
在调用这个函数之前，特别要注意将 lpvParam 参数定义成正确的数据类型

Top
SystemTimeToTzSpecificLocalTime
SystemTimeToTzSpecificLocalTime

VB 声明
Declare Function SystemTimeToTzSpecificLocalTime Lib "kernel32" Alias "SystemTimeToTzSpecificLocalTime" (lpTimeZoneInformation As TIME_ZONE_INFORMATION, lpUniversalTime As SYSTEMTIME, lpLocalTime As SYSTEMTIME) As Long
说明

将系统时间转换成地方时间
返回值
Long，TRUE（非零）表示成功，否则返回零。会设置 GetLastError
参数表
参数类型及说明
lpTimeZoneInformationTIME_ZONE_INFORMATION，包含了时区信息的结构
lpUniversalTimeSYSTEMTIME，包含系统时间的结构
lpLocalTimeSYSTEMTIME，用于装载地方时间的结构
注解
适用平台：Windows NT

Top
ToAscii
ToAscii, ToAsciiEx

VB 声明
Declare Function ToAscii& Lib "user32" (ByVal uVirtKey As Long, ByVal uScanCode As Long, lpbKeyState As Byte, lpwTransKey As Integer, ByVal fuState As Long)
Declare Function ToAsciiEx& Lib "user32" (ByVal uVirtKey As Long, ByVal uScanCode As Long, lpKeyState As Byte, lpwTransKey As Integer, ByVal fuState As Long, ByVal dwhkl As Long)
说明
根据当前的扫描码和键盘信息，将一个虚拟键转换成 ASCII 字符
返回值
Long，负值表明按键是“死”的——不能自己将自己转换成一个字符（重音键[accent keys]就是一个例子）。在给定当前键盘状态的前提下，如按键不能被转换（翻译），则返回 0。如单个字符已载入 lpwTransKey，则返回 1。如 lpwTransKey 里已载入了两个字符（需要把它分隔到两个字节里），那么返回值是 2。在当前字符集里，倘若单独一个字符不能表达键盘支持的死键或重音按键组合，就可能得到 2 的返回值
参数表
参数类型及说明
uVirtKeyLong，欲转换的虚拟键
uScanCodeLong，键的扫描码。如键处于抬起状态，会设置高位（设为 1）；如按下，则清除高位（设为 0）
lpbKeyStateByte，描述了键盘状态的一个 256 字符数组的第一个条目。参考 GetKeyboardState 函数，了解关于这个数组更多的情况
lpwTransKeyInteger，用于装载转换过后的字符的一个整数变量。可用 chr()函数将这个值转换成一个字串
fuStateLong，如一个菜单处于活动状态，则设为 1
dwhklLong，欲用于转换的一个键盘布局的句柄
注解
NumLock 键的状态会被忽略，因为虚拟键码包括了哪个信息
在微软的 win32 手册里，对 ToAsciiEx 函数的建议是将它的 lpwTransKey 参数设为 Long，而不要设为 Integer。这里的函数声明根据实际的 C 语言头，它将参数定义成一个 16 位的字（既

vb 的整数）

Top
ToUnicode
ToUnicode

VB 声明
Declare Function ToUnicode Lib "user32" Alias "ToUnicode" (ByVal wVirtKey As Long, ByVal wScanCode As Long, lpKeyState As Byte, ByVal pwszBuff As String, ByVal cchBuff As Long, ByVal wFlags As Long) As Long
说明
根据当前的扫描码和键盘信息，将一个虚拟键转换成 Unicode 字符
返回值
Long，值-1 表明按键是“死”的——不能自己将自己转换成一个字符（重音键[accent keys]就是一个例子）。在给定当前键盘状态的前提下，如按键不能被转换（翻译），则返回 0。如单个字符已载入 pwszBuff，则返回 1。如 pwszBuff 里已载入了两个或更多的字符，那么返回值是 2。在当前字符集里，倘若单独一个字符不能表达键盘支持的死键或重音按键组合，就可能得到 2 的返回值
参数表
参数类型及说明
wVirtKeyLong，欲转换的虚拟键
wScanCodeLong，键的扫描码。如键处于抬起状态，会设置高位；如按下，则清除高位
lpKeyStateByte，描述了键盘状态的一个 256 字符数组的第一个条目。参考 GetKeyboardState 函数，了解关于这个数组更多的情况
pwszBuffString，用于装载 Unicode 字符的一个字串缓冲区。注意事先对这个字串进行正确的初始化
cchBuffLong，pwszBuff 字串缓冲区的长度
wFlagsLong，如一个菜单处于活动状态，则设为 1
注解
适用平台：Windows NT

Top
UnloadKeyboardLayout
UnloadKeyboardLayout

VB 声明
Declare Function UnloadKeyboardLayout Lib "user32" Alias "UnloadKeyboardLayout" (ByVal HKL As Long) As Long
说明
卸载指定的键盘布局。参考 LoadKeyboardLayout 函数，了解更多信息
返回值
Long，如执行成功，返回键盘布局的句柄；如出错，则返回零。会设置 GetLastError
参数表
参数类型及说明

HKLLong，欲卸载的键盘布局的句柄
注解
在 windows95 下，这个函数永远都不能卸载默认的系统键盘布局

Top
VkKeyScan
VkKeyScan, VkKeyScanEx

VB 声明
Declare Function VkKeyScan% Lib "user32" Alias "VkKeyScanA" (ByVal cChar As Byte)
Declare Function VkKeyScanEx% Lib "user32" Alias "VkKeyScanExA" (ByVal ch As Byte, ByVal dwhkl As Long)
说明
针对 Windows 字符集中一个 ASCII 字符，判断虚拟键码和 Shift 键的状态
返回值
Long，低字包含了虚拟键码。高字则包含了下述标志：
位 0 指出 Shift 键已经按下；位 1 指出 Ctrl 键已经按下；位 2 指出 Alt 键已经按下
参数表
参数类型及说明
chByte，欲转换的字符的 ASCII 值
dwhklLong，转换时作为依据的键盘布局
注解
数字小键盘的转换会被忽略。这个函数可用于获取发送 WM_KEYDOWN 和 WM_KEYUP 消息时应使用的正确参数

Top
控件与消息函数
控件与消息函数：共六页。第一页，第二页，第三页，第四页，第五页，第六页
AdjustWindowRect 给定一种窗口样式，计算获得目标客户区矩形所需的窗口大小
AnyPopup 判断屏幕上是否存在任何弹出式窗口
ArrangeIconicWindows 排列一个父窗口的最小化子窗口
AttachThreadInput 连接线程输入函数
BeginDeferWindowPos 启动构建一系列新窗口位置的过程
BringWindowToTop 将指定的窗口带至窗口列表顶部
CascadeWindows 以层叠方式排列窗口
ChildWindowFromPoint 返回父窗口中包含了指定点的第一个子窗口的句柄
ClientToScreen 判断窗口内以客户区坐标表示的一个点的屏幕坐标
CloseWindow 最小化指定的窗口
CopyRect 矩形内容复制
DeferWindowPos 该函数为特定的窗口指定一个新窗口位置
DestroyWindow 清除指定的窗口以及它的所有子窗口
DrawAnimatedRects 描绘一系列动态矩形
EnableWindow 指定的窗口里允许或禁止所有鼠标及键盘输入
EndDeferWindowPos 同时更新 DeferWindowPos 调用时指定的所有窗口的位置及状态

AdjustWindowRect
AdjustWindowRect,AdjustWindowRectEx

VB 声明
Declare Function AdjustWindowRect Lib "user32" Alias "AdjustWindowRect" (lpRect As RECT, ByVal dwStyle As Long, ByVal bMenu As Long) As Long
Declare Function AdjustWindowRectEx Lib "user32" Alias "AdjustWindowRectEx" (lpRect As RECT, ByVal dsStyle As Long, ByVal bMenu As Long, ByVal dwEsStyle As Long) As Long
说明
在给定一种窗口样式的前提下，计算获得目标客户区矩形所需的窗口大小
返回值
Long，如执行成功，则返回非零值；如失败，返回零值。会设置 GetLastError
参数表
参数类型及说明
lpRectRECT，最初包含要求的客户区。由函数设为目标窗口矩形大小
dwStyleLong，窗口样式
bMenuLong，如窗口有菜单，则设为 TRUE（非零）
dwEsStyleLong，扩展窗口样式（只适用于 AdjustWindowRectEx）
注解
在调用本函数前，先用 GetWindowLong 取得一个窗体的样式。如菜单占用两行以上的空间，则函数不能正确计算大小。如程序使用了多行标题，则应使用 GetSystemMetrics

Top
AnyPopup
AnyPopup

VB 声明
Declare Function AnyPopup Lib "user32" Alias "AnyPopup" () As Long
说明
判断屏幕上是否存在任何弹出式窗口
返回值
Long，如存在弹出式菜单，则返回 TRUE（非零）
注解
对该函数来说，弹出式菜单包含所有可见的包容顶级窗口，无论弹出式还是重叠窗口

Top
ArrangeIconicWindows
ArrangeIconicWindows

VB 声明
Declare Function ArrangeIconicWindows Lib "user32" Alias "ArrangeIconicWindows" (ByVal hwnd As Long) As Long
说明
排列一个父窗口的最小化子窗口（在 vb 里使用：用于在桌面排列图标，用 GetDesktopWindow

函数获得桌面窗口的一个句柄）
返回值
Long，图标行的高度；如失败，则返回零。会设置 GetLastError
参数表
参数类型及说明
hwndLong，父窗口的句柄
注解
也可将该函数用于包含了图标化子窗口的的定制控件

Top
AttachThreadInput
AttachThreadInput

VB 声明
Declare Function AttachThreadInput Lib "user32" Alias "AttachThreadInput" (ByVal idAttach As Long, ByVal idAttachTo As Long, ByVal fAttach As Long) As Long
说明
通常，系统内的每个线程都有自己的输入队列。本函数（既“连接线程输入函数”）允许线程和进程共享输入队列。连接了线程后，输入焦点、窗口激活、鼠标捕获、键盘状态以及输入队列状态都会进入共享状态
返回值
Long，非零表示成功，零表示失败，会设置会 GetLastError
参数表
参数类型及说明
idAttachLong，欲连接线程的标识符（ID）
idAttachToLong，与 idAttach 线程连接的另一个线程的标识符
fAttachLong，TRUE（非零）连接，FALSE 撤消连接
注解
调用这个函数时，会重设键盘状态

Top
BeginDeferWindowPos
BeginDeferWindowPos

VB 声明
Declare Function BeginDeferWindowPos Lib "user32" Alias "BeginDeferWindowPos" (ByVal nNumWindows As Long) As Long
说明
启动构建一系列新窗口位置的过程（以便同时更新）。该函数会向一个内部结果返回一个句柄，这个结构容纳了与窗口位置有关的信息。随后，该结构会由对 DeferWindowPos 函数的调用填充。准备好更新所有窗口位置以后，对 EndDeferWindowPos 的一个调用可同时更新结构内所有窗口的位置
返回值
Long，内部结构的句柄。零表示失败

参数表
参数类型及说明
nNumWindowsLong，在结构中欲为其分配空间的初始窗口数量。在每次 DeferWindowPos 调用期间结构的大小会根据情况自动调节

Top
BringWindowToTop
BringWindowToTop

VB 声明
Declare Function BringWindowToTop Lib "user32" Alias "BringWindowToTop" (ByVal hwnd As Long) As Long
说明
将指定的窗口带至窗口列表顶部。倘若它部分或全部隐藏于其他窗口下面，则将隐藏的部分完全显示出来。该函数也对弹出式窗口、顶级窗口以及 MDI 子窗口产生作用
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，欲带至顶部的那个窗口的句柄
注解
这个函数也许能随同子窗口使用。函数对一个特定的输入线程来说是“本地的”——换言之，倘若某窗口并非前台应用程序的一部分，那么一旦随同该窗口调用本函数，仍会将窗口带至它自己那个应用程序的窗口列表顶部。但是，不会同时使那个应用成为前台应用程序。这意味着在调用了本函数后，窗口仍会保持隐藏状态

Top
CascadeWindows
CascadeWindows,CascadeWindowsBynum

VB 声明
Declare Function CascadeWindows% Lib "user32" (ByVal hwndParent As Long, ByVal wHow As Long, lpRect As RECT, ByVal cKids As Long, lpKids As Long)
Declare Function CascadeWindowsBynum% Lib "user32" Alias "CascadeWindows" (ByVal hwndParent As Long, ByVal wHow As Long, ByVal lpRect As Long, ByVal cKids As Long, ByVal lpKids As Long)
说明
以层叠方式排列窗口（在 vb 里使用：位于顶部或被所有的窗口没有问题。原文：No problem for top level windows or owned windows.）
返回值
Integer，排列成功的窗口数量，零表示失败
参数表
参数类型及说明
hwndParentLong，指定一个父窗口；准备对它的子窗口进行排列。用 GetDesktopWindow 函

数获得顶级窗口的句柄
wHowLong，MDITILE_SKIPDISABLED——不排列已被禁用的 MDI 子窗口
lpRectRECT，指定一个矩形，矩形区域中的窗口才会层叠处理。可设为 NULL，表示使用整个客户区
cKidsLong，在 lpKids 数组中指定的子窗口数量
lpKidsLong，子窗口列表中准备排列的第一个元素。如传递 NULL（注意将参数定义成 ByVal）。Long，则表示排列所有的子窗口（原文：Long--First element in list of child windows to arrange. Pass NULL (be sure to define parameter as ByVal - Long, to arrange all child windows.）
注解
在正式的 win32 文档里，对这个函数的说明是不正确的。这儿的参数建立在实际的 win32 C 头文件基础上。函数不能对诸如控件的子窗口产生——只对顶级窗口及 MDI 子有用。注意在 MDI 窗体的情况下，指定的父窗口应是 MDIClient 窗口的句柄，不应是 MDI 窗体本身的窗口句柄。可用 api 函数 GetParent 获得正确的句柄

Top
ChildWindowFromPoint
ChildWindowFromPoint,ChildWindowFromPointEx

VB 声明
Declare Function ChildWindowFromPoint Lib "user32" Alias "ChildWindowFromPoint" (ByVal hWnd As Long, ByVal xPoint As Long, ByVal yPoint As Long) As Long
Declare Function ChildWindowFromPointEx Lib "user32" Alias "ChildWindowFromPointEx" (ByVal hWnd As Long, ByVal pt As POINTAPI, ByVal un As Long) As Long
说明
返回父窗口中包含了指定点的第一个子窗口的句柄
返回值
Long，发现包含了指定点的第一个子窗口的句柄。如未发现任何窗口，则返回 hWnd（父窗口的句柄）。如指定点位于父窗口外部，则返回零
参数表
参数类型及说明
hWndLong，父窗口的句柄
xPointLong，点的 X 坐标，以像素为单位
yPointLong，点的 Y 坐标，以像素为单位
ptPOINTAPI，点的坐标，以像素为单位
unLong，（只适用于 ChildWindowFromPointEx）控制对窗口的搜索。参见下表
CWP_ALL 测试所有窗口
CWP_SKIPINVISIBLE 忽略不可见窗口
CWP_SKIPDISABLED 忽略已屏蔽的窗口
CWP_SKIPTRANSPARENT 忽略透明窗口

Top
ClientToScreen
ClientToScreen

VB 声明
Declare Function ClientToScreen Lib "user32" Alias "ClientToScreen" (ByVal hwnd As Long, lpPoint As POINTAPI) As Long
说明
判断窗口内以客户区坐标表示的一个点的屏幕坐标
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hwndLong，判断客户区坐标时那个窗口的句柄
lpPointPOINTAPI，用 hwnd 窗口的客户区坐标表示的点，这个参数会包含屏幕坐标系统中相同的点

Top
CloseWindow
CloseWindow

VB 声明
Declare Function CloseWindow Lib "user32" Alias "CloseWindow" (ByVal hwnd As Long) As Long
说明
最小化指定的窗口。窗口不会从内存中清除
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，欲最小化的那个窗口的句柄

Top
CopyRect
CopyRect

VB 声明
Declare Function CopyRect Lib "user32" Alias "CopyRect" (lpDestRect As RECT, lpSourceRect As RECT) As Long
说明
将矩形的 lpSourceRect 内容复制给矩形 lpDestRect
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpDestRectRECT，目标矩形结构
lpSourceRectRECT，源矩形

Top
DeferWindowPos
DeferWindowPos

VB 声明
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
说明
该函数为特定的窗口指定一个新窗口位置，并将其输入由 BeginDeferWindowPos 创建的结构，以便在 EndDeferWindowPos 函数执行期间更新
返回值
Long，返回一个新句柄，它指向的结构包含了位置更新信息。这个句柄应在对 DeferWindowPos 的后续调用以及对 EndDeferWindowPos 的结束调用中用到。如出错，则返回零值
参数表
参数类型及说明
hWinPosInfoLong，由 BeginDeferWindowPos 为后续对 DeferWindowPos 的调用返回的句柄
hwndLong，欲定位的窗口
hWndInsertAfterLong，窗口句柄。在窗口列表中，窗口 hwnd 会排列于这个窗口后面。也可用下述值之一：
HWND_BOTTOM 将窗口置于窗口列表底部
HWND_TOP 将窗口置于 Z 序列顶部；Z 序列是窗口针对分级结构中一个给定级别显示的顺序
HWND_TOPMOST 将窗口置于列表顶部，位于任何最顶级窗口的前面（请参考 WS_EX_TOPMOST 样式位）
HWND_NOTOPMOST 将窗口置于列表顶部，位于任何最顶级窗口的后面
xLong，窗口新的 x 坐标。如 hwnd 是个子窗口，那么 x 用父窗口的客户区坐标表示
yLong，窗口新的 y 坐标。如 hwnd 是个子窗口，那么 y 用父窗口的客户区坐标表示
cxLong，指定新窗口宽度
cyLong，指定新窗口高度
wFlagsLong，包含了旗标的一个整数，如下所示：
SWP_DRAWFRAME 围绕窗口画一个框
SWP_HIDEWINDOW 隐藏窗口
SWP_NOACTIVATE 不激活窗口
SWP_NOMOVE 保持当前位置（x 和 y 设定将被忽略）
SWP_NOREDRAW 窗口不自动重画
SWP_NOSIZE 保持当前大小（cx 和 cy 会被忽略）
SWP_NOZORDER 保持在窗口列表的当前位置（hWndInsertAfter 会被忽略）
SWP_SHOWWINDOW 显示窗口
SWP_NOOWNERZORDER 不改变 Z 序列的所有者
SWP_NOSENDCHANGING 窗口不发出 WM_WINDOWPOSCHANGING 消息
注解

请参考对 SetWindowPos 函数的注解。同时参考 BeginDeferWindowPos 和 EndDeferWindowPos

Top
DestroyWindow
DestroyWindow

VB 声明
Declare Function DestroyWindow Lib "user32" Alias "DestroyWindow" (ByVal hwnd As Long) As Long
说明
破坏（即清除）指定的窗口以及它的所有子窗口（在 vb 里使用：用处不大。原文：it is unlikely to be of much use.）
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，欲清除的窗口的句柄

Top
DrawAnimatedRects
DrawAnimatedRects

VB 声明
Declare Function DrawAnimatedRects Lib "user32" Alias "DrawAnimatedRects" (ByVal hwnd As Long, ByVal idAni As Long, lprcFrom As Rect, lprcTo As Rect) As Long
说明
在 lprcFrom 和 lprcTo 之间描绘一系列动态矩形
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hwndLong，要在其中描绘矩形的窗口。如设为零，表示以桌面为背景
idAniLong，windows 95 要设为 0
lprcFromRect，原始矩形
lprcToRect，目标矩形
注解
我的理解——这个函数用于在窗口缩放时产生动画效果

Top
EnableWindow
EnableWindow

VB 声明
Declare Function EnableWindow Lib "user32" Alias "EnableWindow" (ByVal hwnd As Long,

ByVal fEnable As Long) As Long
说明
在指定的窗口里允许或禁止所有鼠标及键盘输入（在 vb 里使用：在 vb 窗体和控件中使用 Enabled 属性）
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，窗口句柄
fEnableLong，非零允许窗口，零禁止

Top
EndDeferWindowPos
EndDeferWindowPos

VB 声明
Declare Function EndDeferWindowPos Lib "user32" Alias "EndDeferWindowPos" (ByVal hWinPosInfo As Long) As Long
说明
同时更新 DeferWindowPos 调用时指定的所有窗口的位置及状态
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hWinPosInfoLong，由对 DeferWindowPos 最近一次调用返回的结构句柄

Top
EnumChildWindows
EnumChildWindows

VB 声明
Declare Function EnumChildWindows Lib "user32" Alias "EnumChildWindows" (ByVal hWndParent As Long, ByVal lpEnumFunc As Long, ByVal lParam As Long) As Long
说明
为指定的父窗口枚举子窗口
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hWndParentLong，欲枚举子窗口的父窗口的句柄
lpEnumFuncLong，为每个子窗口调用的函数的指针。用 AddressOf 运算符获得函数在一个标准模块中的地址
lParamLong，在枚举期间，传递给 dwcbkd32.ocx 定制控件之 EnumWindows 事件的值。这个值的含义是由程序员规定的。（原文：Value that is passed to the EnumWindows event of the

dwcbkd32.ocx custom control during enumeration. The meaning of this value is defined by the programmer.）
注解
在 vb4 下要求 dwcbkd32.ocx 定制控件。子窗口下属的子窗口也可由这个函数枚举

Top
EnumThreadWindows
EnumThreadWindows

VB 声明
Declare Function EnumThreadWindows Lib "user32" Alias "EnumThreadWindows" (ByVal dwThreadId As Long, ByVal lpfn As Long, ByVal lParam As Long) As Long
说明
枚举与指定任务相关的窗口
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
dwThreadIdLong，某线程的标识符，它的窗口将被枚举
lpfnLong，指向一个函数的指针，要求为每个子窗口都调用这个函数。用 AddressOf 运算符获得函数在标准模式下的地址
lParamLong，在枚举期间，传递给 dwcbkd32d.ocx 定制控件之 EnumWindows 事件的值。这个值的含义是由程序员规定的
注解
子窗口下属的其他子窗口也可由这个函数枚举

Top
EnumWindows
EnumWindows

VB 声明
Declare Function EnumWindows& Lib "user32" (ByVal lpEnumFunc As Long, ByVal lParam As Long)
说明
枚举窗口列表中的所有父窗口（顶级和被所有窗口）
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
lpEnumFuncLong，指向为每个子窗口都调用的一个函数的指针。用 AddressOf 运算符获得函数在标准模式下的地址
lParamLong，在枚举期间，传递给 dwcbkd32.ocx 定制控件之 EnumWindows 事件的值。这个值的含义是由程序员规定的
注解

我的理解——在随 vb5 同时提供的 api32.txt 文件中，找不到这个函数

Top
EqualRect
EqualRect

VB 声明
Declare Function EqualRect Lib "user32" Alias "EqualRect" (lpRect1 As RECT, lpRect2 As RECT) As Long
说明
判断两个矩形结构是否相同
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpRect1RECT，要比较的矩形
lpRect2RECT，要比较的矩形

Top
FindWindow
FindWindow

VB 声明
Declare Function FindWindow Lib "user32" Alias "FindWindowA" (ByVal lpClassName As String, ByVal lpWindowName As String) As Long
说明
寻找窗口列表中第一个符合指定条件的顶级窗口（在 vb 里使用：FindWindow 最常见的一个用途是获得 ThunderRTMain 类的隐藏窗口的句柄；该类是所有运行中 vb 执行程序的一部分。获得句柄后，可用 api 函数 GetWindowText 取得这个窗口的名称；该名也是应用程序的标题）
返回值
Long，找到窗口的句柄。如未找到相符窗口，则返回零。会设置 GetLastError
参数表
参数类型及说明
lpClassNameString，指向包含了窗口类名的空中止（C 语言）字串的指针；或设为零，表示接收任何类
lpWindowNameString，指向包含了窗口文本（或标签）的空中止（C 语言）字串的指针；或设为零，表示接收任何窗口标题
注解
很少要求同时按类与窗口名搜索。为向自己不准备参数传递一个零，最简便的办法是传递 vbNullString 常数
示例
```
Dim hw&, cnt&
Dim rttitle As String * 256
hw& = FindWindow("ThunderRT5Main", vbNullString) ' ThunderRTMain under VB4
```

```
cnt = GetWindowText(hw&, rttitle, 255)
MsgBox Left$(rttitle, cnt), 0, "RTMain title"
```

Top
FindWindowEx
FindWindowEx

VB 声明
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
说明
在窗口列表中寻找与指定条件相符的第一个子窗口
返回值
Long，找到的窗口的句柄。如未找到相符窗口，则返回零。会设置 GetLastError
参数表
参数类型及说明
hWnd1Long，在其中查找子的父窗口。如设为零，表示使用桌面窗口（通常说的顶级窗口都被认为是桌面的子窗口，所以也会对它们进行查找）
hWnd2Long，从这个窗口后开始查找。这样便可利用对 FindWindowEx 的多次调用找到符合条件的所有子窗口。如设为零，表示从第一个子窗口开始搜索
lpsz1String，欲搜索的类名。零表示忽略
lpsz2String，欲搜索的类名。零表示忽略

Top
FlashWindow
FlashWindow

VB 声明
Declare Function FlashWindow Lib "user32" Alias "FlashWindow" (ByVal hwnd As Long, ByVal bInvert As Long) As Long
说明
闪烁显示指定窗口。这意味着窗口的标题和说明文字会发生变化，似乎从活动切换到非活动状态、或反向切换。通常对不活动的窗口应用这个函数，引起用户的注意
返回值
Long，如窗口在调用前处于活动状态，则返回 TRUE（非零）
参数表
参数类型及说明
hwndLong，要闪烁显示的窗口的句柄
bInvertLong，TRUE（非零）表示切换窗口标题；FALSE 返回最初状态
注解
该函数通常与一个计数器组合使用，生成连续的闪烁效果。在 windows nt 及 windows for workgroup 中，bInvert 参数会被忽略。但在 windows 95 中不会忽略

Top

GetActiveWindow
GetActiveWindow

VB 声明
Declare Function GetActiveWindow Lib "user32" Alias "GetActiveWindow" () As Long
说明
获得活动窗口的句柄
返回值
Long，活动窗口的句柄，如没有窗口处于活动状态，则返回零值

Top
GetCapture
GetCapture

VB 声明
Declare Function GetCapture Lib "user32" Alias "GetCapture" () As Long
说明
获得一个窗口的句柄，这个窗口位于当前输入线程，且拥有鼠标捕获（鼠标活动由它接收）
返回值
Long，拥有捕获的窗口的句柄。倘若当前线程中没有窗口拥有捕获，则返回零值

Top
GetClassInfo
GetClassInfo,GetClassInfoEx

VB 声明
Declare Function GetClassInfo& Lib "user32" Alias "GetClassInfoA" (ByVal hInstance As Long, ByVal lpClassName As String, lpWndClass As WNDCLASS)
Declare Function GetClassInfoEx& Lib "user32" Alias "GetClassInfoExA" (ByVal hInstance As Long, ByVal lpClassName As String, lpWndClassEx As WNDCLASSEX)
说明
取得 WNDCLASS 结构（或 WNDCLASSEX 结构）的一个副本，结构中包含了与指定类有关的信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hInstanceLong，指向拥有类的那个实例的一个句柄。如设为 NULL，则获得与标准 windows 类有关的信息
lpClassNameString，欲查找的类名。也可能是个 Long 值，其中的低字是包含类名的一个全局（共用）原子
lpWndClassWNDCLASS，（GetClassInfo）用于包含结果信息的结构
lpWndClassExWNDCLASSEX，（GetClassInfoEx）用于包含结果信息的结构
注解

如为这个函数使用了 WNDCLASSEX 参数，请务必设置其中的 cbSize 字段

Top
GetClassLong
GetClassLong

VB 声明
Declare Function GetClassLong Lib "user32" Alias "GetClassLongA" (ByVal hwnd As Long, ByVal nIndex As Long) As Long
说明
取得窗口类的一个 Long 变量条目
返回值
Long，由 nIndex 决定。零表示出错。会设置 GetLastError
参数表
参数类型及说明
hwnd 要为其获得类信息的窗口的句柄
nIndex 欲取得的信息，可能是下述任何一个常数：（正数表示一个字节偏移，用于取得在额外字节中为这个类分配的类信息）
GCL_CBCLSEXTRA 这个类结构中分配的额外字节数
GCL_CBWNDEXTRA 窗口结构里为这个类中每个窗口分配的额外字节数
GCL_HBRBACKGROUND 描绘这个类每个窗口的背景时，使用的默认刷子的句柄
GCL_HCURSOR 指向这个类窗口默认光标的句柄
GCL_HICON 这个类中窗口默认图标的句柄
GCL_HICONSM 这个类的小图标
GCL_HMODULE 这个类的模块的句柄
GCL_MENUNAME 为类菜单取得名称或资源 ID
GCL_STYLE 这个类的样式
GCL_WNDPROC 取得类窗口函数（该类窗口的默认窗口函数）的地址

Top
GetClassName
GetClassName

VB 声明
Declare Function GetClassName Lib "user32" Alias "GetClassNameA" (ByVal hwnd As Long, ByVal lpClassName As String, ByVal nMaxCount As Long) As Long
说明
为指定的窗口取得类名
返回值
Long，以字节数表示的类名长度；排除最后的空中止字符。零表示出错。会设置 GetLastError
参数表
参数类型及说明
hwndLong，欲获得类名的那个窗口的句柄
lpClassNameString，随同类名载入的缓冲区。预先至少必须分配 nMaxCount+1 个字符

nMaxCountLong，由 lpClassName 提供的缓冲区长度

Top
GetClassWord
GetClassWord

VB 声明
Declare Function GetClassWord Lib "user32" Alias "GetClassWord" (ByVal hwnd As Long, ByVal nIndex As Long) As Long
说明
为窗口类取得一个整数变量
返回值
Long，由 nIndex 决定。零表示出错。会设置 GetLastError
参数表
参数类型及说明
hwndLong，欲获得类信息的那个窗口的句柄
nIndexLong，类信息的正偏移量；这些信息是该类的额外字节中分配的
注解
我的迷惑：在某参考书上是这样声明的：Declare Function GetClassWord% Lib "user32" (ByVal hwnd As Long, ByVal nIndex As Long)，即函数的返回值为 Integer，而从 vb 自带的 api 查看器中得到的声明表明返回值是个 Long 类型

Top
GetClientRect
GetClientRect

VB 声明
Declare Function GetClientRect Lib "user32" Alias "GetClientRect" (ByVal hwnd As Long, lpRect As RECT) As Long
说明
返回指定窗口客户区矩形的大小
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，欲计算大小的目标窗口
lpRectRECT，指定一个矩形，用客户区域的大小载入（以像素为单位）
注解
lpRect 的左侧及顶部区域肯定会被这个函数设为零

Top
GetDesktopWindow
GetDesktopWindow

VB 声明
Declare Function GetDesktopWindow Lib "user32" Alias "GetDesktopWindow" () As Long
说明
获得代表整个屏幕的一个窗口（桌面窗口）句柄
返回值
Long，桌面窗口的句柄
注解
所有桌面图标都在这个窗口里拒绝。它也用于各类屏幕保护程序

Top
GetFocus
GetFocus

VB 声明
Declare Function GetFocus Lib "user32" Alias "GetFocus" () As Long
说明
获得拥有输入焦点的窗口的句柄
返回值
Long，拥有焦点的那个窗口的句柄。如没有窗口拥有输入焦点，则返回零

Top
GetForegroundWindow
GetForegroundWindow

VB 声明
Declare Function GetForegroundWindow Lib "user32" Alias "GetForegroundWindow" () As Long
说明
获得前台窗口的句柄。这里的“前台窗口”是指前台应用程序的活动窗口
返回值
Long，前台窗口的句柄
注解
windows nt 支持多个桌面，它们相互间是独立的。每个桌面都有自己的前台窗口

Top
GetLastActivePopup
GetLastActivePopup

VB 声明
Declare Function GetLastActivePopup Lib "user32" Alias "GetLastActivePopup" (ByVal hwndOwnder As Long) As Long
说明
获得在一个给定父窗口中最近激活过的弹出式窗口的句柄（在 vb 里使用：vb 应用程序不用弹出式窗口，所以这个函数并非特别有用）
返回值

Long，指向最近用过的弹出式窗口的句柄。如未发现弹出窗口。则返回 hwndOwnder
参数表
参数类型及说明
hwndOwnderLong，父窗口

Top
GetParent
GetParent

VB 声明
Declare Function GetParent Lib "user32" Alias "GetParent" (ByVal hwnd As Long) As Long
说明
判断指定窗口的父窗口
返回值
Long，父窗口的句柄。如窗口没有父，或遇到错误，则返回零。会设置 GetLastError
参数表
参数类型及说明
hwndLong，欲测试的窗口的句柄

Top
GetTopWindow
GetTopWindow

VB 声明
Declare Function GetTopWindow Lib "user32" Alias "GetTopWindow" (ByVal hwnd As Long) As Long
说明
搜索内部窗口列表，寻找隶属于指定窗口的头一个窗口的句柄
返回值
Long，位于指定窗口内部的顶级子窗口的句柄
参数表
参数类型及说明
ByValLong，想在其中搜寻顶级子的窗口。零表示寻找位于桌面的顶级窗口
注解
Z 序列中的顶级窗口也是内部窗口列表的第一个窗口

Top
GetUpdateRect
GetUpdateRect

VB 声明
Declare Function GetUpdateRect Lib "user32" Alias "GetUpdateRect" (ByVal hwnd As Long, lpRect As RECT, ByVal bErase As Long) As Long
说明

获得一个矩形，它描叙了指定窗口中需要更新的那一部分（在 vb 里使用：到一个 vb 窗体或控件里发生 paint 事件的时候，更新区域已被清除了。所以这个函数对于 vb 来说是没有意义的。然而，可用一个子类模块拦截一个窗体或控件的 WM_PAINT 消息，在 vb 自行清除之前了解更新区域在哪里）
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hwndLong，欲在其中调查更新区域的那个窗口
lpRectRECT，随同更新坐标载入的矩形
bEraseLong，设置 TRUE（非零），清除更新区域
注解
如窗口类样式拥有 CS_OWNDC 集，且窗口映射模式不是 MM_TEXT，那么更新矩形会用逻辑坐标表示

Top
GetWindow
GetWindow

VB 声明
Declare Function GetWindow Lib "user32" Alias "GetWindow" (ByVal hwnd As Long, ByVal wCmd As Long) As Long
说明
获得一个窗口的句柄，该窗口与某源窗口有特定的关系
返回值
Long，由 wCmd 决定的一个窗口的句柄。如没有找到相符窗口，或者遇到错误，则返回零值。会设置 GetLastError
参数表
参数类型及说明
hwndLong，源窗口
wCmdLong，指定结果窗口与源窗口的关系，它们建立在下述常数基础上：
GW_CHILD 寻找源窗口的第一个子窗口
GW_HWNDFIRST 为一个源子窗口寻找第一个兄弟（同级）窗口，或寻找第一个顶级窗口
GW_HWNDLAST 为一个源子窗口寻找最后一个兄弟（同级）窗口，或寻找最后一个顶级窗口
GW_HWNDNEXT 为源窗口寻找下一个兄弟窗口
GW_HWNDPREV 为源窗口寻找前一个兄弟窗口
GW_OWNER 寻找窗口的所有者
注解
兄弟或同级是指在整个分级结构中位于同一级别的窗口。如某个窗口有五个子窗口，那五个窗口就是兄弟窗口。尽管 GetWindow 可用于枚举窗口，但倘若要在枚举过程中重新定位、创建和清除窗口，那么 EnumWindows 和 EnumChildWindows 更为可靠

Top

GetWindowContextHelpId
GetWindowContextHelpId

VB 声明
Declare Function GetWindowContextHelpId Lib "user32" Alias "GetWindowContextHelpId" (ByVal hWnd As Long) As Long
说明
取得与窗口关联在一起的帮助场景 ID
返回值
Long，帮助场景 ID（如果有的话），否则返回零
参数表
参数类型及说明
hWndLong，欲获取帮助场景的那个窗口的句柄
注解
在 vb 环境中，这个函数不能取得一个 vb 窗体或控件的帮助场景的 ID 集

Top
GetWindowLong
GetWindowLong

VB 声明
Declare Function GetWindowLong Lib "user32" Alias "GetWindowLongA" (ByVal hwnd As Long, ByVal nIndex As Long) As Long
说明
从指定窗口的结构中取得信息
返回值
Long，由 nIndex 决定。零表示出错。会设置 GetLastError
参数表
参数类型及说明
hwndLong，欲为其获取信息的窗口的句柄
nIndexLong，欲取回的信息，可以是下述任何一个常数：
GWL_EXSTYLE 扩展窗口样式
GWL_STYLE 窗口样式
GWL_WNDPROC 该窗口的窗口函数的地址
GWL_HINSTANCE 拥有窗口的实例的句柄
GWL_HWNDPARENT 该窗口之父的句柄。不要用 SetWindowWord 来改变这个值
GWL_ID 对话框中一个子窗口的标识符
GWL_USERDATA 含义由应用程序规定
DWL_DLGPROC 这个窗口的对话框函数地址
DWL_MSGRESULT 在对话框函数中处理的一条消息返回的值
DWL_USER 含义由应用程序规定

Top
GetWindowPlacement

GetWindowPlacement

VB 声明
Declare Function GetWindowPlacement Lib "user32" Alias "GetWindowPlacement" (ByVal hwnd As Long, lpwndpl As WINDOWPLACEMENT) As Long
说明
获得指定窗口的状态及位置信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，欲获取信息的那个窗口的句柄
lpwndplWINDOWPLACEMENT，包含的窗口位置和状态信息的结构
注解
在调用这个函数之前，请务必设置 WINDOWPLACEMENT 结构的长度字段

Top
GetWindowRect
GetWindowRect

VB 声明
Declare Function GetWindowRect Lib "user32" Alias "GetWindowRect" (ByVal hwnd As Long, lpRect As RECT) As Long
说明
获得整个窗口的范围矩形，窗口的边框、标题栏、滚动条及菜单等都在这个矩形内
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，想获得范围矩形的那个窗口的句柄
lpRectRECT，屏幕坐标中随同窗口装载的矩形
注解
如将它与通过 GetDesktopWindow 获得的句柄联合使用，可获得对整个可视显示区域（桌面）进行说明的矩形

Top
GetWindowText
GetWindowText

VB 声明
Declare Function GetWindowText Lib "user32" Alias "GetWindowTextA" (ByVal hwnd As Long, ByVal lpString As String, ByVal cch As Long) As Long
说明
取得一个窗体的标题（caption）文字，或者一个控件的内容（在 vb 里使用：使用 vb 窗体或

控件的 caption 或 text 属性）
返回值
Long，复制到 lpString 的字串长度；不包括空中止字符。会设置 GetLastError
参数表
参数类型及说明
hwndLong，欲获取文字的那个窗口的句柄
lpStringString，预定义的一个缓冲区，至少有 cch+1 个字符大小；随同窗口文字载入
cchLong，lpString 缓冲区的长度
注解
不能用它从另一个应用程序的编辑控件中获取文字

Top
GetWindowTextLength
GetWindowTextLength

VB 声明
Declare Function GetWindowTextLength Lib "user32" Alias "GetWindowTextLengthA" (ByVal hwnd As Long) As Long
说明
调查窗口标题文字或控件内容的长短（在 vb 里使用：直接使用 vb 窗体或控件的 caption 或 text 属性）
返回值
Long，字串长度，不包括空中止字符
参数表
参数类型及说明
hwndLong，想调查文字长度的窗口的句柄

Top
GetWindowWord
GetWindowWord

VB 声明
Declare Function GetWindowWord Lib "user32" Alias "GetWindowWord" (ByVal hwnd As Long, ByVal nIndex As Long) As Integer
说明
获得指定窗口结构的信息
返回值
Long，由 nIndex 决定
参数表
参数类型及说明
hwndLong，想获取信息的那个窗口的句柄
nIndexLong，正偏移值，指出在窗口额外字节分配的空间中取得的信息

Top

InflateRect
InflateRect

VB 声明
Declare Function InflateRect Lib "user32" Alias "InflateRect" (lpRect As RECT, ByVal x As Long, ByVal y As Long) As Long
说明
这个函数用于增大或减小一个矩形的大小。x 加在右侧区域，并从左侧区域减去；如 x 为正，则能增大矩形的宽度；如 x 为负，则能减小它。y 对顶部与底部区域产生的影响是是类似的
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpRectRECT，欲修改的矩形
xLong，用这个数字修改宽度
yLong，用这个数字修改高度
注解
注意宽度和高度的实际改变量是 x 及 y 参数值的两倍

Top
IntersectRect
IntersectRect

VB 声明
Declare Function IntersectRect Lib "user32" Alias "IntersectRect" (lpDestRect As RECT, lpSrc1Rect As RECT, lpSrc2Rect As RECT) As Long
说明
这个函数在 lpDestRect 里载入一个矩形，它是 lpSrc1Rect 与 lpSrc2Rect 两个矩形的交集。如两个源矩形根本未发生重叠，则 lpDestRect 会被设为一个空矩形
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpDestRectRECT，目标矩形，用于包含 lpSrc1Rect 与 lpSrc2Rect 两个矩形的重合部分
lpSrc1RectRECT，第一个源矩形
lpSrc2RectRECT，第二个源矩形

Top
InvalidateRect
InvalidateRect,InvalidateRectBynum

VB 声明
Declare Function InvalidateRect& Lib "user32" (ByVal hwnd As Long, lpRect As RECT, ByVal bErase As Long)

Declare Function InvalidateRectBynum& Lib "user32" Alias "InvalidateRect" (ByVal hwnd As Long, ByVal lpRect As Long, ByVal bErase As Long)
说明
这个函数屏蔽一个窗口客户区的全部或部分区域。这会导致窗口在事件期间部分重画
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hwndLong，待屏蔽窗口的句柄
lpRectRECT，用于描述待屏蔽矩形部分的一个矩形结构。可用 InvalidateRectBynum 函数，同时将 lpRect 设为零（Long 数据类型），从而屏蔽（或禁用）整个窗口
bEraseLong，TRUE（非零）导致指定的区域在重画前先删除
注解
一旦系统有些更新屏幕的闲置时间可用，windows 就会重画窗口

Top
IsChild
IsChild

VB 声明
Declare Function IsChild Lib "user32" Alias "IsChild" (ByVal hWndParent As Long, ByVal hwnd As Long) As Long
说明
判断一个窗口是否为另一窗口的子或隶属窗口
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hWndParentLong，父窗口的句柄
hwndLong，欲测试的窗口的句柄

Top
IsIconic
IsIconic

VB 声明
Declare Function IsIconic Lib "user32" Alias "IsIconic" (ByVal hwnd As Long) As Long
说明
判断窗口是否已最小化
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hwndLong，待检测窗口的句柄

Top
IsRectEmpty
IsRectEmpty

VB 声明
Declare Function IsRectEmpty Lib "user32" Alias "IsRectEmpty" (lpRect As RECT) As Long
说明
判断一个矩形是否为空
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpRectRECT，要检查的矩形

Top
IsWindow
IsWindow

VB 声明
Declare Function IsWindow Lib "user32" Alias "IsWindow" (ByVal hwnd As Long) As Long
说明
判断一个窗口句柄是否有效
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hwndLong，待检查窗口的句柄
注解
如在一个程序变量里容纳了窗口句柄，为了解它是否仍然有效，就可考虑使用这个函数

Top
IsWindowEnabled
IsWindowEnabled

VB 声明
Declare Function IsWindowEnabled Lib "user32" Alias "IsWindowEnabled" (ByVal hwnd As Long) As Long
说明
判断窗口是否处于活动状态（在 vb 里使用：针对 vb 窗体和控件，请用 enabled 属性）
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明

hwndLong，待检测窗口的句柄

Top
IsWindowUnicode
IsWindowUnicode

VB 声明
Declare Function IsWindowUnicode Lib "user32" Alias "IsWindowUnicode" (ByVal hwnd As Long) As Long
说明
判断一个窗口是否为 Unicode 窗口。这意味着窗口为所有基于文本的消息都接收 Unicode 文字
返回值
Long，如是个 Unicode 窗口则返回非零值，如是个 ANSI 窗口则返回零
参数表
参数类型及说明
hwndLong，窗口句柄
注解
在 vb 里无须担心这方面的问题，windows 会自动转换引用了文本的消息，以满足窗口接收消息的需要

Top
IsWindowVisible
IsWindowVisible

VB 声明
Declare Function IsWindowVisible Lib "user32" Alias "IsWindowVisible" (ByVal hwnd As Long) As Long
说明
判断窗口是否可见
返回值
Long，如窗口可见则返回 TRUE（非零）
参数表
参数类型及说明
hwndLong，要测试的那个窗口的句柄
注解
窗口可能处于可见状态，同时仍被位于它顶部的其他可见窗口（排在内部窗口列表的前面）隐藏起来

Top
IsZoomed
IsZoomed

VB 声明

Declare Function IsZoomed Lib "user32" Alias "IsZoomed" (ByVal hwnd As Long) As Long
说明
判断窗口是否最大化
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hwndLong，欲测试的窗口的句柄

Top
LockWindowUpdate
LockWindowUpdate

VB 声明
Declare Function LockWindowUpdate Lib "user32" Alias "LockWindowUpdate" (ByVal hwndLock As Long) As Long
说明
锁定指定窗口，禁止它更新。同时只能有一个窗口处于锁定状态
返回值
Long，非零表示成功，零表示失败（比如另外已有一个窗口锁定）
参数表
参数类型及说明
hwndLockLong，欲锁定窗口的句柄。如设为零，则对窗口解锁
注解
windows 会跟踪锁定窗口的区域，并会在窗口解锁后重画它们。可用 GetDCEx 获得一个特殊的设备场景，令其与锁定窗口协同工作，从而描绘一个加锁的窗口。这种技术的一个应用场合是创建跟踪矩形（比如用于改变窗口大小的矩形）

Top
MapWindowPoints
MapWindowPoints

VB 声明
Declare Function MapWindowPoints& Lib "user32" (ByVal hwndFrom As Long, ByVal hwndTo As Long, lppt As POINTAPI, ByVal cPoints As Long)
说明
将一个窗口客户区坐标的点转换到另一窗口的客户区坐标系统（在 vb 里使用：无论向函数传递单独一个点，还是传递数组中的第一个 POINTAPI 结构，都要特别谨慎。数组中的条目数量至少等于由 cPoints 参数指定的数量）
返回值
Long，低字代表映射过程中添加给每个点的水平偏移，高字则代表垂直偏移
参数表
参数类型及说明
hwndFromLong，定义源坐标的窗口。用零或桌面窗口句柄指定屏幕坐标

hwndToLong，定义目标坐标的窗口。用零或桌面窗口句柄指定屏幕坐标
lpptPOINTAPI，点结构中待转换的第一个条目。注意 RECT 结构在内存中组织成两个连续的 POINTAPI 结构。这样就可为该函数创建一个别名，并使用 RECT 结构；而不是 POINTAPI 结构。如这样做时，注意将 cPoints 的值加倍
cPointsLong，欲转换的点数
注解
在 vb 自带的 api 查看器中复制的声明为：Declare Function MapWindowPoints Lib "user32" Alias "MapWindowPoints" (ByVal hwndFrom As Long, ByVal hwndTo As Long, lppt As Any, ByVal cPoints As Long) As Long，请注意：lppt As Any

Top
MoveWindow
MoveWindow

VB 声明
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
说明
改变指定窗口的位置和大小。顶级窗口可能受最大或最小尺寸的限制，那些尺寸优先于这里设置的参数
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hwndLong，欲移动窗口的句柄
xLong，窗口新的左侧位置
yLong，窗口新的顶部位置
nWidthLong，窗口的新宽度
nHeightLong，窗口的高宽度
bRepaintLong，如窗口此时应重画，则设为 TRUE（非零）。FALSE（零）则表明应用程序会自己决定是否重画窗口

Top
OffsetRect
OffsetRect

VB 声明
Declare Function OffsetRect Lib "user32" Alias "OffsetRect" (lpRect As RECT, ByVal x As Long, ByVal y As Long) As Long
说明
该函数通过应用一个指定的偏移，从而让矩形移动起来。x 会添加到右侧和左侧区域。y 添加到顶部和底部区域。偏移方向则取决于参数是正数还是负数，以及采用的是什么坐标系统
返回值

Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpRectRECT，欲移动的矩形
xLong，水平偏移量
yLong，垂直偏移量

Top
OpenIcon
OpenIcon

VB 声明
Declare Function OpenIcon Lib "user32" Alias "OpenIcon" (ByVal hwnd As Long) As Long
说明
恢复一个最小化的程序，并将其激活
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，欲恢复的窗口
注解
针对 vb 窗体，应使用 vb 的 WindowState 属性

Top
PtInRect
PtInRect

VB 声明
Declare Function PtInRect Lib "user32" Alias "PtInRect" (lpRect As RECT, pt As POINTAPI) As Long
说明
这个函数判断指定的点是否位于矩形 lpRect 内部
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpRectRECT，欲检查的矩形
ptPOINTAPI，欲判断的点
注解
如点位于矩形四边之内，或矩形的顶部或左侧边线上，则认为它在矩形内部。如位于矩形的右侧或底部边线，则不认为它在矩形内部

Top
RedrawWindow

RedrawWindow

VB 声明
Declare Function RedrawWindow Lib "user32" Alias "RedrawWindow" (ByVal hwnd As Long, lprcUpdate As RECT, ByVal hrgnUpdate As Long, ByVal fuRedraw As Long) As Long
说明
根据 fuRedraw 旗标的设置，重画全部或部分窗口
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，要重画的窗口的句柄。零表示更新桌面窗口
lprcUpdateRECT，窗口中需要重画的一个矩形区域
hrgnUpdateLong，一个“区”的句柄，这个区描述了要重画的窗口区域。“区”：Region
fuRedrawLong，规定具体重画操作的旗标。下列常数可组合使用，从而进行复杂的重画行动
RDW_ERASE 重画前，先清除重画区域的背景。也必须指定 RDW_INVALIDATE
RDW_FRAME 如非客户区包含在重画区域中，则对非客户区进行更新。也必须指定 RDW_INVALIDATE
RDW_INTERNALPAINT 即使窗口并非无效，也向其投递一条 WM_PAINT 消息
RDW_INVALIDATE 禁用（屏蔽）重画区域
RDW_NOERASE 禁止删除重画区域的背景
RDW_NOFRAME 禁止非客户区域重画（如果它是重画区域的一部分）。也必须指定 RDW_VALIDATE
RDW_NOINTERNALPAINT 禁止内部生成或由这个函数生成的任何待决 WM_PAINT 消息。针对无效区域，仍会生成 WM_PAINT 消息
RDW_VALIDATE 检验重画区域
RDW_ERASENOW 立即删除指定的重画区域
RDW_UPDATENOW 立即更新指定的重画区域
RDW_ALLCHILDREN 重画操作包括子窗口（前提是它们存在于重画区域）
RDW_NOCHILDREN 重画操作排除子窗口（前提是它们存在于重画区域）
注解
如针对桌面窗口应用这个函数，则应用程序必须用 RDW_ERASE 旗标重画桌面

Top
ReleaseCapture
ReleaseCapture

VB 声明
Declare Function ReleaseCapture Lib "user32" Alias "ReleaseCapture" () As Long
说明
为当前的应用程序释放鼠标捕获
返回值
Long，TRUE（非零）表示成功，零表示失败
注解

我的理解：与 SetCapture 函数一起使用，用于判断鼠标离开（mouseleave）事件

Top
ScreenToClient
ScreenToClient

VB 声明
Declare Function ScreenToClient Lib "user32" Alias "ScreenToClient" (ByVal hwnd As Long, lpPoint As POINTAPI) As Long
说明
判断屏幕上一个指定点的客户区坐标
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hwndLong，一个窗口的句柄，该窗口定义了要使用的客户区坐标系统
lpPointPOINTAPI，屏幕坐标系统中包含了屏幕点的结构。这个函数会随同相应的客户区坐标（由 hwnd 决定）载入结构

Top
ScrollWindow
ScrollWindow

VB 声明
Declare Function ScrollWindow Lib "user32" Alias "ScrollWindow" (ByVal hWnd As Long, ByVal XAmount As Long, ByVal YAmount As Long, lpRect As RECT, lpClipRect As RECT) As Long
说明
滚动窗口客户区的全部或一部分
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hWndLong，待滚动窗口的句柄
XAmountLong，水平滚动的距离。正值向右滚动，负值向左滚动
YAmountLong，垂直滚动的距离。正值向下滚动，负值向上滚动
lpRectRECT，用客户区坐标表示的一个矩形，它定义了客户区要滚动的一个部分。如设为 NULL，则滚动整个客户区。在 NULL 的情况下，子窗口和控件的位置也会随同任何无效区域移动。否则，子窗口和无效区域不会一起移动。因此，在滚动之前，如指定了 lpRect，一个明智的做法是先调用 UpdateWindow 函数
lpClipRectRECT，指定剪切区域。只有这个矩形的区域才可能滚动。该矩形优先于 lpRect。可设为 NULL

Top

ScrollWindowEx
ScrollWindowEx

VB 声明
Declare Function ScrollWindowEx Lib "user32" Alias "ScrollWindowEx" (ByVal hwnd As Long, ByVal dx As Long, ByVal dy As Long, lprcScroll As RECT, lprcClip As RECT, ByVal hrgnUpdate As Long, lprcUpdate As RECT, ByVal fuScroll As Long) As Long
说明
根据附加的选项，滚动窗口客户区的全部或部分
返回值
Long，常数值 SIMPLEREGION，COMPLEXREGION，或 NULLREGION，它们描述了无效区域的类型
参数表
参数类型及说明
hwndLong，欲滚动的窗口的句柄
dxLong，水平滚动的距离。正值向右滚动，负值向左滚动
dyLong，垂直滚动的距离。正值向下滚动，负值向上滚动
lprcScrollRECT，用客户区坐标表示的一个矩形，它定义了客户区要滚动的一个部分。如设为零，则滚动整个客户区
lprcClipRECT，指定一个剪切矩形。只有这个矩形的内容才可能滚动。该矩形优先于 lpRect。可能为零，表示不进行剪切处理（原文：Clipping rectangle. Only the area within this rectangle may be scrolled. This rectangle takes priority over lpRect. May be zero, in which case no clipping takes place.）
hrgnUpdateLong，滚动过程中随同无效区域载入的一个“区”。可能是零
lprcUpdateRECT，随同一个矩形载入的矩形结构，该矩形定义了滚动过程中无效的区域。可能是零
fuScrollLong，对滚动进行控制的旗标。可以是下述任何常数的组合
SW_ERASE 清除新无效区域的背景
SW_INVALDATE 使滚动时未覆盖的区域无效
SW_SCROLLCHILDREN 滚动区域内的子窗口进行等量移动。为避免得到无效的结果，在使用这个函数的时候，请确定子窗口或控件要么完全在滚动区域中，要么完全在滚动区域外

Top
SetActiveWindow
SetActiveWindow

VB 声明
Declare Function SetActiveWindow Lib "user32" Alias "SetActiveWindow" (ByVal hwnd As Long) As Long
说明
激活指定的窗口
返回值
Long，前一个活动窗口的句柄
参数表

参数类型及说明
hwndLong，待激活窗口的句柄
注解
在 vb 里使用这个函数要小心，它不会改变输入焦点，所以焦点可能设向一个不活动窗口，最好换用 SetFocusAPI 函数来激活窗口。如指定的窗口不从属于当前输入线程，则没有任何效果

Top
SetCapture
SetCapture

VB 声明
Declare Function SetCapture Lib "user32" Alias "SetCapture" (ByVal hwnd As Long) As Long
说明
将鼠标捕获设置到指定的窗口。在鼠标按钮按下的时候，这个窗口会为当前应用程序或整个系统接收所有鼠标输入
返回值
Long，之前拥有鼠标捕获的窗口的句柄
参数表
参数类型及说明
hwndLong，要接收所有鼠标输入的窗口的句柄
注解
我的理解：与 ReleaseCapture 函数一起使用，用于判断鼠标离开（mouseleave）事件

Top
SetClassLong
SetClassLong

VB 声明
Declare Function SetClassLong Lib "user32" Alias "SetClassLongA" (ByVal hwnd As Long, ByVal nIndex As Long, ByVal dwNewLong As) As Long
说明
为窗口类设置一个 Long 变量条目
返回值
Long，由 nIndex 指定的的类信息的前一个值。零表示出错。会设置 GetLastError
参数表
参数类型及说明
hwndLong，欲为其设置类信息的那个窗口的句柄
nIndexLong，参考 GetClassLong 函数的 nIndex 参数说明
dwNewLongLong，类信息的新值，具体取决于 nIndex
注解
使用这个函数一定要小心。记住，这里的变动会影响指定类的所有窗口，但除非窗口重画，否则那个变动不会显露出来

Top
SetClassWord
SetClassWord

VB 声明
Declare Function SetClassWord Lib "user32" Alias "SetClassWord" (ByVal hwnd As Long, ByVal nIndex As Long, ByVal wNewWord As Long) As Long
说明
为窗口类设置一个条目
返回值
Long，由 nIndex 指定的类信息的前一个值。零表示出错。会设置 GetLastError
参数表
参数类型及说明
hwndLong，欲为其获得类信息的那个窗口的句柄
nIndexLong，信息结构已分配额外空间的正偏移量
wNewWordLong，由 nIndex 指定的类信息的新值
注解
使用这个函数一定要小心。记住，这里的变动会影响指定类的所有窗口，但除非窗口重画，否则那个变动不会显露出来

Top
SetFocusAPI
SetFocusAPI

VB 声明
Declare Function SetFocusAPI& Lib "user32" Alias "SetFocus" (ByVal hwnd As Long)
说明
将输入焦点设到指定的窗口。如有必要，会激活窗口
返回值
Long，前一个拥有焦点的窗口的句柄
参数表
参数类型及说明
hwndLong，准备接收焦点的窗口的句柄
注解
在 vb 里对窗体和控件最好使用 SetFocus 方法。如指定的窗口不属于当前输入线程，则该函数是没有效果的。它用 SetFocusAPI 这个别名避免与 vb 的 SetFocus 方法发生冲突

Top
SetForegroundWindow
SetForegroundWindow

VB 声明
Declare Function SetForegroundWindow Lib "user32" Alias "SetForegroundWindow" (ByVal hwnd As Long) As Long

说明
将窗口设为系统的前台窗口。这个函数可用于改变用户目前正在操作的应用程序
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，带到前台的窗口
注解
不应随便使用它，因为一旦程序突然从后台进入前台，可能会使用户产生迷惑

Top
SetParent
SetParent

VB 声明
Declare Function SetParent Lib "user32" Alias "SetParent" (ByVal hWndChild As Long, ByVal hWndNewParent As Long) As Long
说明
指定一个窗口的新父（在 vb 里使用：利用这个函数，vb 可以多种形式支持子窗口。例如，可将控件从一个容器移至窗体中的另一个。用这个函数在窗体间移动控件是相当冒险的，但却不失为一个有效的办法。如真的这样做，请在关闭任何一个窗体之前，注意用 SetParent 将控件的父设回原来的那个）
返回值
Long，前一个父窗口的句柄
参数表
参数类型及说明
hWndChildLong，子窗口的句柄
hWndNewParentLong，hWndChild 的新父
注解
可用这个函数在运行期将 vb 控件置入容器控件内部（比如将一个按钮设成图象或窗体控件的子窗口），或者将控件从一个容器控件移至另一个。控件移至另一个父后，它的位置将由新父的坐标系统决定。这样一来，有必要重新规定控件的位置，使其能在目标位置显示出来

Top
SetRect
SetRect

VB 声明
Declare Function SetRect Lib "user32" Alias "SetRect" (lpRect As RECT, ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long) As Long
说明
设置指定矩形的内容
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError

参数表
参数类型及说明
lpRectRECT，欲设置的矩形
X1Long，左侧区域（Left）的值
Y1Long，顶部区域（Top）的值
X2Long，右侧区域（Right）的值
Y2Long，底部区域（Bottom）的值

Top
SetRectEmpty
SetRectEmpty

VB 声明
Declare Function SetRectEmpty Lib "user32" Alias "SetRectEmpty" (lpRect As RECT) As Long
说明
将矩形 lpRect 设为一个空矩形（所有字段都为空）
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpRectRECT，欲设为空的矩形

Top
SetWindowContextHelpId
SetWindowContextHelpId

VB 声明
Declare Function SetWindowContextHelpId Lib "user32" Alias "SetWindowContextHelpId" (ByVal hWnd As Long, ByVal dw As Long) As Long
说明
为指定的窗口设置帮助场景（上下文）ID
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hWndLong，窗口句柄
dwLong，新的帮助场景 ID

Top
SetWindowLong
SetWindowLong

VB 声明
Declare Function SetWindowLong Lib "user32" Alias "SetWindowLongA" (ByVal hwnd As Long,

ByVal nIndex As Long, ByVal dwNewLong As Long) As Long
说明
在窗口结构中为指定的窗口设置信息
返回值
Long，指定数据的前一个值
参数表
参数类型及说明
hwndLong，欲为其取得信息的窗口的句柄
nIndexLong，请参考 GetWindowLong 函数的 nIndex 参数的说明
dwNewLongLong，由 nIndex 指定的窗口信息的新值

Top
SetWindowPlacement
SetWindowPlacement

VB 声明
Declare Function SetWindowPlacement Lib "user32" Alias "SetWindowPlacement" (ByVal hwnd As Long, lpwndpl As WINDOWPLACEMENT) As Long
说明
设置窗口状态和位置信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，欲设置位置信息的窗口的句柄
lpwndplWINDOWPLACEMENT，这个结构包含了窗口的位置与状态信息

Top
SetWindowPos
SetWindowPos

VB 声明
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
说明
这个函数能为窗口指定一个新位置和状态。它也可改变窗口在内部窗口列表中的位置。该函数与 DeferWindowPos 函数相似，只是它的作用是立即表现出来的（在 vb 里使用：针对 vb 窗体，如它们在 win32 下屏蔽或最小化，则需重设最顶部状态。如有必要，请用一个子类处理模块来重设最顶部状态
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明

hwndLong，欲定位的窗口
hWndInsertAfterLong，窗口句柄。在窗口列表中，窗口 hwnd 会置于这个窗口句柄的后面。也可能选用下述值之一：
HWND_BOTTOM 将窗口置于窗口列表底部
HWND_TOP 将窗口置于 Z 序列的顶部；Z 序列代表在分级结构中，窗口针对一个给定级别的窗口显示的顺序
HWND_TOPMOST 将窗口置于列表顶部，并位于任何最顶部窗口的前面
HWND_NOTOPMOST 将窗口置于列表顶部，并位于任何最顶部窗口的后面
xLong，窗口新的 x 坐标。如 hwnd 是一个子窗口，则 x 用父窗口的客户区坐标表示
yLong，窗口新的 y 坐标。如 hwnd 是一个子窗口，则 y 用父窗口的客户区坐标表示
cxLong，指定新的窗口宽度
cyLong，指定新的窗口高度
wFlagsLong，包含了旗标的一个整数
SWP_DRAWFRAME 围绕窗口画一个框
SWP_HIDEWINDOW 隐藏窗口
SWP_NOACTIVATE 不激活窗口
SWP_NOMOVE 保持当前位置（x 和 y 设定将被忽略）
SWP_NOREDRAW 窗口不自动重画
SWP_NOSIZE 保持当前大小（cx 和 cy 会被忽略）
SWP_NOZORDER 保持窗口在列表的当前位置（hWndInsertAfter 将被忽略）
SWP_SHOWWINDOW 显示窗口
SWP_FRAMECHANGED 强迫一条 WM_NCCALCSIZE 消息进入窗口，即使窗口的大小没有改变
注解
窗口成为最顶级窗口后，它下属的所有窗口也会进入最顶级。一旦将其设为非最顶级，则它的所有下属和物主窗口也会转为非最顶级。Z 序列用垂直于屏幕的一根假想 Z 轴量化这种从顶部到底部排列的窗口顺序

Top
SetWindowText
SetWindowText

VB 声明
Declare Function SetWindowText Lib "user32" Alias "SetWindowTextA" (ByVal hwnd As Long, ByVal lpString As String) As Long
说明
设置窗口的标题文字或控件的内容（在 vb 里使用：针对 vb 窗体，应使用 caption 或 text 属性）
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，要设置文字的窗口的句柄
lpStringString，要设到 hwnd 窗口中的文字

Top
SetWindowWord
SetWindowWord

VB 声明
Declare Function SetWindowWord Lib "user32" Alias "SetWindowWord" (ByVal hwnd As Long, ByVal nIndex As Long, ByVal wNewWord As Long) As Long
说明
在窗口结构中为指定的窗口设置信息（在 vb 里使用：使用时要谨慎）
返回值
Long，指定数据的前一个值
参数表
参数类型及说明
hwndLong，欲为其设置信息的那个窗口的句柄
nIndexLong，参考对 GetWindowWord 函数的 nIndex 参数的说明
wNewWordLong，由 nIndex 指定的窗口信息的新值

Top
ShowOwnedPopups
ShowOwnedPopups

VB 声明
Declare Function ShowOwnedPopups Lib "user32" Alias "ShowOwnedPopups" (ByVal hwnd As Long, ByVal fShow As Long) As Long
说明
显示或隐藏由指定窗口所有的全部弹出式窗口（在 vb 里使用：并不特别有用，因为 vb 不用弹出式窗口）
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，父窗口的句柄
fShowLong，TRUE（非零）表示显示由 hwnd 所有的所有弹出式窗口；FALSE（零）则隐藏它们

Top
ShowWindow
ShowWindow

VB 声明
Declare Function ShowWindow Lib "user32" Alias "ShowWindow" (ByVal hwnd As Long, ByVal nCmdShow As Long) As Long
说明

控制窗口的可见性（在 vb 里使用：针对 vb 窗体及控件，请使用对应的 vb 属性）
返回值
Long，如窗口之前是可见的，则返回 TRUE（非零），否则返回 FALSE（零）
参数表
参数类型及说明
hwndLong，窗口句柄，要向这个窗口应用由 nCmdShow 指定的命令
nCmdShowLong，为窗口指定可视性方面的一个命令。请用下述任何一个常数
SW_HIDE 隐藏窗口，活动状态给令一个窗口
SW_MINIMIZE 最小化窗口，活动状态给令一个窗口
SW_RESTORE 用原来的大小和位置显示一个窗口，同时令其进入活动状态
SW_SHOW 用当前的大小和位置显示一个窗口，同时令其进入活动状态
SW_SHOWMAXIMIZED 最大化窗口，并将其激活
SW_SHOWMINIMIZED 最小化窗口，并将其激活
SW_SHOWMINNOACTIVE 最小化一个窗口，同时不改变活动窗口
SW_SHOWNA 用当前的大小和位置显示一个窗口，不改变活动窗口
SW_SHOWNOACTIVATE 用最近的大小和位置显示一个窗口，同时不改变活动窗口
SW_SHOWNORMAL 与 SW_RESTORE 相同

Top
ShowWindowAsync
ShowWindowAsync

VB 声明
Declare Function ShowWindowAsync Lib "user32" Alias "ShowWindowAsync" (ByVal hWnd As Long, ByVal nCmdShow As Long) As Long
说明
与 ShowWindow 相似，只是这时的 ShowWindow 命令会投递到指定的窗口，然后进行异步处理。这样一来，就可控制从属于另一个进程的窗口的可视情况。同时无须担心另一个进程挂起的时候，自己的应用程序也会牵连其中
返回值
Long，如窗口之前是可见的，则返回 TRUE（非零），否则返回 FALSE（零）
参数表
参数类型及说明
hWndLong，欲接收 ShowWindow 命令的窗口
nCmdShowLong，与 ShowWindow 相同

Top
SubtractRect
SubtractRect

VB 声明
Declare Function SubtractRect Lib "user32" Alias "SubtractRect" (lprcDst As RECT, lprcSrc1 As RECT, lprcSrc2 As RECT) As Long
说明

这个函数会装载矩形 lprcDst，它是在矩形 lprcSrc1 中减去 lprcSrc2 得到的结果
返回值
Long，TRUE（非零）表示成功，零表示失败
参数表
参数类型及说明
lprcDstRECT，准备包容 lprcSrc1 减 lprcSrc2 结果的目标矩形
lprcSrc1RECT，第一个源矩形
lprcSrc2RECT，第二个源矩形
注解
lprcSrc1 与 lprcSrc2 这两个矩形必须在水平或垂直方向完全相交。换句话说，倘若在 lprcSrc1 中拿掉与 lprcSrc2 相交的 lprcDst 部分，结果必须是个矩形

Top
TileWindows
TileWindows

VB 声明
Declare Function TileWindows% Lib "user32" (ByVal hwndParent As Long, ByVal wHow As Long, lpRect As RECT, ByVal cKids As Long, lpKids As Long)
Declare Function TileWindowsBynum% Lib "user32" Alias "TileWindows" (ByVal hwndParent As Long, ByVal wHow As Long, ByVal lpRect As Long, ByVal cKids As Long, ByVal lpKids As Long)
说明
以平铺顺序排列窗口（在 vb 里使用：对顶级窗口和 MDI 子窗口有效）
返回值
Integer，成功排列的窗口数量，零表示失败
参数表
参数类型及说明
hwndParentLong，欲对其子窗口进行排列的父窗口。可用 GetDesktopWindow 函数获得顶级窗口（桌面）的句柄
wHowLong，MDITILE_HORIZONTAL 或 MDITILE_VERTICAL，用于设置平铺方向（水平或垂直）
lpRectLong，要在其中平铺窗口的矩形，可设为 NULL，表示用整个客户区域
cKidsLong，lpKids 数组中指定的子窗口数量
lpKidsLong，欲排列子窗口列表的第一个元素。如传递 NULL（务必将参数定义成 ByVal Long），可排列所有子窗口
注解
这个函数不能对诸如控件的子窗口产生作用——只对对顶级窗口和 MDI 子窗口有用。在 MDI 窗口的情况下，指定的父窗口应是 MDIClient 窗口的句柄，不应是 MDI 窗体本身的窗口句柄。可用 GetParent 获得正确的句柄

Top
UnionRect
UnionRect

VB 声明
Declare Function UnionRect Lib "user32" Alias "UnionRect" (lpDestRect As RECT, lpSrc1Rect As RECT, lpSrc2Rect As RECT) As Long
说明
这个函数会装载一个 lpDestRect 目标矩形，它是 lpSrc1Rect 和 lpSrc2Rect 联合起来的结果。目标矩形的所有点都同时位于两个源矩形里；也即是它们的一个交集
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpDestRectRECT，用于容纳 lpSrc1Rect 和 lpSrc2Rect 联合运算结果的目标矩形
lpSrc1RectRECT，第一个源矩形
lpSrc2RectRECT，第二个源矩形

Top
UpdateWindow
UpdateWindow

VB 声明
Declare Function UpdateWindow Lib "user32" Alias "UpdateWindow" (ByVal hwnd As Long) As Long
说明
强制立即更新窗口，窗口中以前屏蔽的所有区域都会重画（在 vb 里使用：如 vb 窗体或控件的任何部分需要更新，可考虑直接使用 refresh 方法
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hwndLong，欲更新窗口的句柄

Top
ValidateRect
ValidateRect

VB 声明
Declare Function ValidateRect& Lib "user32" (ByVal hwnd As Long, lpRect As RECT)
Declare Function ValidateRectBynum& Lib "user32" Alias "ValidateRect" (ByVal hwnd As Long, ByVal lpRect As Long)
说明
校验窗口的全部或部分客户区。这样便可告之 windows 指定的区域不需要重画
返回值
Long，非零表示成功，零表示失败
参数表

参数类型及说明
hwndLong，欲检验的窗口句柄
lpRectRECT，指定一个矩形结构，用于描叙欲校验的矩形部分。可使用 ValidateRectBynum，同时将 lpRect 设为零（Long 数据类型），从而对整个窗口进行校验

Top
WindowFromPoint
WindowFromPoint

VB 声明
Declare Function WindowFromPoint Lib "user32" Alias "WindowFromPoint" (ByVal xPoint As Long, ByVal yPoint As Long) As Long
说明
返回包含了指定点的窗口的句柄。忽略屏蔽、隐藏以及透明窗口
返回值
Long，包含了指定点的窗口的句柄。如指定的点处没有窗口存在，则返回零
参数表
参数类型及说明
xPointLong，x 点值
yPointLong，y 点值

Top
位图、图标和光栅运算函数
位图、图标和光栅运算函数，共三页。第一页，第二页，第三页
BitBlt 将一幅位图从一个设备场景复制到另一个
CopyIcon 制作指定图标或鼠标指针的一个副本。这个副本从属于发出调用的应用程序
CopyImage 复制位图、图标或指针，同时在复制过程中进行一些转换工作
CreateBitmap 按照规定的格式创建一幅与设备有关位图
CreateBitmapIndirect 创建一幅与设备有关位图
CreateCompatibleBitmap 创建一幅与设备有关位图，它与指定的设备场景兼容
CreateCursor 创建一个鼠标指针
CreateDIBitmap 根据一幅与设备无关的位图创建一幅与设备有关的位图
CreateDIBSection 创建一个 DIBSection
CreateIcon 创建一个图标
CreateIconIndirect 创建一个图标
DestroyCursor 清除指定的鼠标指针，并释放它占用的所有系统资源
DestroyIcon 清除图标
DrawIcon 在指定的位置画一个图标
DrawIconEx 描绘一个图标或鼠标指针。与 DrawIcon 相比，这个函数提供了更多的功能
ExtractAssociatedIcon 判断一个可执行程序或 DLL 中是否存在图标，或是否有图标与系统注册表中指定的文件存在关联并提取之
BitBlt
BitBlt

VB 声明
Declare Function BitBlt Lib "gdi32" Alias "BitBlt" (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
说明
将一幅位图从一个设备场景复制到另一个。源和目标 DC 相互间必须兼容
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hDestDCLong，目标设备场景
x,yLong，对目标 DC 中目标矩形左上角位置进行描述的那个点。用目标 DC 的逻辑坐标表示
nWidth,nHeightLong，欲传输图象的宽度和高度
hSrcDCLong，源设备场景。如光栅运算未指定源，则应设为 0
xSrc,ySrcLong，对源 DC 中源矩形左上角位置进行描述的那个点。用源 DC 的逻辑坐标表示
dwRopLong，传输过程要执行的光栅运算
注解
在 NT 环境下，如在一次世界传输中要求在源设备场景中进行剪切或旋转处理，这个函数的执行会失败
如目标和源 DC 的映射关系要求矩形中像素的大小必须在传输过程中改变，那么这个函数会根据需要自动伸缩、旋转、折叠、或切断，以便完成最终的传输过程

Top
CopyIcon
CopyIcon

VB 声明
Declare Function CopyIcon Lib "user32" Alias "CopyIcon" (ByVal hIcon As Long) As Long
说明
制作指定图标或鼠标指针的一个副本。这个副本从属于发出调用的应用程序
返回值
Long，新图标的句柄，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hIconLong，欲复制的图标或指针的句柄
注解
由于这个函数在 win32 中可复制鼠标指针，所以 win32 未提供对 win16 函数 CopyCursor 的支持
从一个 DLL 或其他应用程序载入一个图标后，只有 DLL 或应用程序保持载入状态，那个图标才可使用。这个函数允许我们制作图标从属于自己应用程序的一个副本。一旦不再需要，必须用 DestroyIcon 函数将其清除，以释放占用的系统资源

Top

CopyImage
CopyImage

VB 声明
Declare Function CopyImage Lib "user32" Alias "CopyImage" (ByVal handle As Long, ByVal un1 As Long, ByVal n1 As Long, ByVal n2 As Long, ByVal un2 As Long) As Long
说明
复制位图、图标或指针，同时在复制过程中进行一些转换工作
返回值
Long，执行成功则返回新图象的句柄，零表示失败。会设置 GetLastError
参数表
参数类型及说明
handleLong，欲复制的图象的句柄
un1Long，下述常数之一：MAGE_BITMAP， IMAGE_CURSOR 或 IMAGE_ICON
n1Long，副本以像素表示的宽度
n2Long，副本以像素表示的高度
un2Long，下述常数任意组合：
LR_DELETEORG 删除原来的图象
LR_COPYRETURNORG 忽略 n1 和 n2 设置
LR_MONOCHROME 创建一个单色副本
LR_COPYFROMRESOURCE 在原始资源的基础上创建一个副本，原始图象即是从那个资源中载入的。假设我们想为一个 32×32 的图标制作一个 64×64 的副本。如果不设这个标志，CopyImage 会直接放大原来的图标。而使用这个标志后，CopyImage 首先检查资源文件中是否存在这个图标的一个 64×64 版本，如果存在，就直接载入品质更好的图象
注解
这个函数通常在希望复制已选入其他设备场景的一幅位图时使用——例如，复制已成为 ImageList 控件一部分的某幅位图。选定的位图将不能使用，因为一次只能将位图选入一个设备场景

Top
CreateBitmap
CreateBitmap

VB 声明
Declare Function CreateBitmap Lib "gdi32" Alias "CreateBitmap" (ByVal nWidth As Long, ByVal nHeight As Long, ByVal nPlanes As Long, ByVal nBitCount As Long, lpBits As Any) As Long
说明
按照规定的格式创建一幅与设备有关位图
返回值
Long，执行成功返回位图的句柄，零表示失败
参数表
参数类型及说明
nWidthLong，位图宽度，以像素为单位

nHeightLong，位图高度，以像素为单位
nPlanesLong，色层数量
nBitCountLong，每像素的位数
lpBitsAny，指向欲载入位图的数据的指针。可设为零，表示不对位图进行初始化（用 ByVal 传递一个零值）。这个数据的格式必须与设备的要求相符。扫描线必须对齐 16 位字边界
注解
一旦不再需要，记住用 DeleteObject 函数释放位图占用的内存和资源
可用这个函数创建单色位图（1 层，每像素一位）。对于彩色位图，则应使用 CreateCompatibleBitmap。这个函数可以胜任工作；但要注意，用它创建的位图在使用时会稍慢一些，因为 Windows 每次使用的时候都必须检查它的位图格式
如果 nWidth 和 nHeight 为零，返回的位图就是一个 1×1 的单色位图

Top
CreateBitmapIndirect
CreateBitmapIndirect

VB 声明
Declare Function CreateBitmapIndirect Lib "gdi32" Alias "CreateBitmapIndirect" (lpBitmap As BITMAP) As Long
说明
创建一幅与设备有关位图
返回值
Long，执行成功返回位图句柄，零表示失败
参数表
参数类型及说明
lpBitmapBITMAP，对一幅逻辑位图进行描述的结构。这个结构中的字段与 CreateBitmap 函数的参数大致对应
注解
如使用由 GetObject 函数获得的 BITMAP 结构，注意 GetObject 不会取回位图数据——只有位图的大小和配置信息
如果 nWidth 和 nHeight 为零，返回的位图就是一个 1×1 的单色位图。也请参考对 CreateBitmap 函数的注解

Top
CreateCompatibleBitmap
CreateCompatibleBitmap

VB 声明
Declare Function CreateCompatibleBitmap Lib "gdi32" Alias "CreateCompatibleBitmap" (ByVal hdc As Long, ByVal nWidth As Long, ByVal nHeight As Long) As Long
说明
创建一幅与设备有关位图，它与指定的设备场景兼容
返回值
Long，执行成功返回位图句柄，零表示失败

参数表
参数类型及说明
hdcLong，设备场景的句柄
nWidthLong，位图宽度，以像素为单位
nHeightLong，位图高度，以像素为单位
注解
内存设备场景即与彩色位图兼容，也与单色位图兼容。这个函数的作用是创建一幅与当前选入 hdc 中的场景兼容。对一个内存场景来说，默认的位图是单色的。倘若内存设备场景有一个 DIBSection 选入其中，这个函数就会返回 DIBSection 的一个句柄。如 hdc 是一幅设备位图，那么结果生成的位图就肯定兼容于设备（也就是说，彩色设备生成的肯定是彩色位图）
如果 nWidth 和 nHeight 为零，返回的位图就是一个 1×1 的单色位图
一旦位图不再需要，一定用 DeleteObject 函数释放它占用的内存及资源

Top
CreateCursor
CreateCursor

VB 声明
Declare Function CreateCursor Lib "user32" Alias "CreateCursor" (ByVal hInstance As Long, ByVal nXhotspot As Long, ByVal nYhotspot As Long, ByVal nWidth As Long, ByVal nHeight As Long, lpANDbitPlane As Any, lpXORbitPlane As Any) As Long
说明
创建一个鼠标指针
返回值
Long，执行成功返回指针的句柄，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hInstanceLong，准备拥有指针的应用程序的实例的句柄。可用 GetWindowWord 函数获得拥有一个窗体或控件的一个实例的句柄
nXhotspot,nYhotspotLong，鼠标指针图象中代表准确指针位置的 X，Y 坐标
nWidthLong，指针图象的宽度。可用 GetSystemMetrics 函数判断一个特定设备的正确编号。VGA 的编号是 32
nHeightLong，指针图象的高度。可用 GetSystemMetrics 函数判断一个特定设备的正确编号。VGA 的编号是 32
lpANDbitPlaneAny，指向 AND 位图数据的指针
lpXORbitPlaneAny，指向 XOR 位图数据的指针
注解
一旦不再需要，注意用 DestroyCursor 函数释放鼠标指针占用的内存及资源

Top
CreateDIBitmap
CreateDIBitmap

VB 声明

Declare Function CreateDIBitmap Lib "gdi32" Alias "CreateDIBitmap" (ByVal hdc As Long, lpInfoHeader As BITMAPINFOHEADER, ByVal dwUsage As Long, lpInitBits As Any, lpInitInfo As BITMAPINFO, ByVal wUsage As Long) As Long
说明
根据一幅与设备无关的位图创建一幅与设备有关的位图
返回值
Long，执行成功返回位图句柄，零表示失败
参数表
参数类型及说明
hdcLong，一个设备场景的句柄，该设备场景定义了要创建的与设备有关位图的配置信息
lpInfoHeaderBITMAPINFOHEADER，对 DIB（与设备无关位图）的格式进行描述的一个结构
dwUsageLong，如不应对位图数据进行初始化，那么设为零。如设为 CBM_INIT，表示根据 lpInitBits 和 lpInitInfo 参数对位图进行初始化
lpInitBitsAny，指向 DIB 格式中的位图数据的一个指针，格式是由 lpInitInfo 指定的
lpInitInfoBITMAPINFO，这个结构对 lpInitBits DIB 的格式及颜色进行了说明
wUsageLong，下述常数之一：
DIB_PAL_COLORS——颜色表包含对当前选定的调色板的索引
DIB_RGB_COLORS——颜色表包含了 RGB 颜色
注解
一旦不再需要，记住用 DeleteObject 函数释放位图占用的内存及资源

Top
CreateDIBSection
CreateDIBSection

VB 声明
Declare Function CreateDIBSection Lib "gdi32" Alias "CreateDIBSection" (ByVal hDC As Long, pBitmapInfo As BITMAPINFO, ByVal un As Long, ByVal lplpVoid As Long, ByVal handle As Long, ByVal dw As Long) As Long
说明
创建一个 DIBSection。这是一个 GDI 对象，可象一幅与设备有关位图那样使用。但是，它在内部作为一幅与设备无关位图保存
返回值
Long，执行成功返回 DIBSection 位图的句柄，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hDCLong，一个设备场景的句柄。如 dw 设为 DIB_PAL_COLORS，那么 DIB 颜色表就会用来自逻辑调色板的颜色进行初始化
pBitmapInfoBITMAPINFO，这个结构初始化成欲创建的那幅位图的配置数据
unLong，下述常数之一：
DIB_PAL_COLORSBITMAPINFO 包含了一个 16 位调色板索引的数组
DIB_RGB_COLORSBITMAPINFO 包含了一个颜色表，其中保存有 32 位颜色（RGBQUAD）
lplpVoidLong，用于载入 DIBSection 数据区的内存地址

handleLong，指向一个文件映射对象的可选句柄，位图将在其中创建。如设为零，Windows 会自动分配内存
dwLong，如指定了句柄，就用这个参数指定位图数据在文件映射对象中的偏移量
注解
一旦不再需要，记住用 DeleteObject 函数删除 DIBSection 位图
如 Windows 分配了一个内存缓冲区，那么对象删除以后，缓冲区也会自动删除。如指定了一个文件映射对象，则不会自动将其清除
在直接访问 DIB 内存之前，首先必须保证 Windows 已完成了绘图（记住，Windows 可能对绘图操作进行了排列处理）。通过调用 gdiFlush 函数，可确保完成所有未决的绘图操作

Top
CreateIcon
CreateIcon

VB 声明
Declare Function CreateIcon Lib "user32" Alias "CreateIcon" (ByVal hInstance As Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal nPlanes As Byte, ByVal nBitsPixel As Byte, lpANDbits As Byte, lpXORbits As Byte) As Long
说明
创建一个图标
返回值
Long，执行成功返回图标的句柄，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hInstanceLong，准备拥有图标的应用程序的实例的句柄。可用 GetWindowWord 函数获得拥有一个窗体或控件的一个实例的句柄
nWidthLong，图标图象的宽度。可用 GetSystemMetrics 函数判断一个特定设备的正确编号。VGA 的编号是 32
nHeightLong，图标图象的高度。可用 GetSystemMetrics 函数判断一个特定设备的正确编号。VGA 的编号是 32
nPlanesByte，lpXORbits 数据数组中的色层数量
nBitsPixelByte，lpXORbits 数据数组中每像素的位数
lpANDbitsByte，指向 AND 位图数据的指针
lpXORbitsByte，指向 XOR 位图数据的指针
注解
一旦不再需要，注意用 DestroyIcon 函数释放鼠标指针占用的内存及资源

Top
CreateIconIndirect
CreateIconIndirect

VB 声明
Declare Function CreateIconIndirect Lib "user32" Alias "CreateIconIndirect" (piconinfo As ICONINFO) As Long

说明
创建一个图标
返回值
Long，执行成功返回图标的句柄，零表示失败。会设置 GetLastError
参数表
参数类型及说明
piconinfoICONINFO，这个结构包含的信息与欲创建的图标有关。图标是随同两幅参考位图创建出来的。这个函数允许我们在给定 XOR 和 AND 位图的前提下创建图标，而不是给出实际的掩模数组
注解
一旦不再需要，注意用 DestroyIcon 函数释放鼠标指针占用的内存及资源

Top
DestroyCursor
DestroyCursor

VB 声明
Declare Function DestroyCursor Lib "user32" Alias "DestroyCursor" (ByVal hCursor As Long) As Long
说明
清除指定的鼠标指针，并释放它占用的所有系统资源。不要用这个函数清除随同 LoadCursor 函数载入的系统指针资源
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hCursorLong，欲清除的指针对象的句柄

Top
DestroyIcon
DestroyIcon

VB 声明
Declare Function DestroyIcon Lib "user32" Alias "DestroyIcon" (ByVal hIcon As Long) As Long
说明
清除图标
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hIconLong，图标句柄
注解
不要用这个函数清除随同 LoadIcon 函数载入的系统固有图标

Top
DrawIcon
DrawIcon

VB 声明
Declare Function DrawIcon Lib "user32" Alias "DrawIcon" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long, ByVal hIcon As Long) As Long
说明
在指定的位置画一个图标
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景
x,yLong，想描绘图标的位置（逻辑坐标）
hIconLong，欲描绘图标的句柄

Top
DrawIconEx
DrawIconEx

VB 声明
Declare Function DrawIconEx Lib "user32" Alias "DrawIconEx" (ByVal hdc As Long, ByVal xLeft As Long, ByVal yTop As Long, ByVal hIcon As Long, ByVal cxWidth As Long, ByVal cyWidth As Long, ByVal istepIfAniCur As Long, ByVal hbrFlickerFreeDraw As Long, ByVal diFlags As Long) As Long
说明
描绘一个图标或鼠标指针。与 DrawIcon 相比，这个函数提供了更多的功能
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，要在其中画图的一个设备场景的句柄
xLeft,yTopLong，图标左上角的位置，用逻辑坐标表示
hIconLong，要描绘的图标的句柄
cxWidth,cyWidthLong，希望的图标宽度和高度。图标会自动缩放，与指定的值相符
istepIfAniCurLong，如果 hIcon 是个动画指针，那么该参数指定描绘动画中的哪幅图象。注意 Win32 不能区分图标和指针
hbrFlickerFreeDrawLong，如设为一个刷子句柄，那么函数会将图标画入一幅内存位图，并用背景色填充。随后，将图象直接复制到指定的位置。这样做可绘图时减少闪烁（因为画图过程中重现）
diFlagsLong，下述常数之一：
DI_COMPAT 描绘标准的系统指针，而不是指定的图象
DI_DEFAULTSIZE 忽略 cxWidth 和 cyWidth 设置，并采用原始的图标大小

DI_IMAGE 绘图时使用图标的 XOR 部分（即图标没有透明区域）
DI_MASK 绘图时使用图标的 MASK 部分（如单独使用，可获得图标的掩模）
DI_NORMAL 用常规方式绘图（合并 DI_IMAGE 和 DI_MASK）
注解
应检查 Windows95 是否与指定的标志及参数兼容。Win32 用户手册宣称函数与 Windows 95 是兼容的，但在实际运用中发现它有一定的限制

Top
ExtractAssociatedIcon
ExtractAssociatedIcon

VB 声明
Declare Function ExtractAssociatedIcon Lib "shell32.dll" Alias "ExtractAssociateIconA" (ByVal hInst As Long, ByVal lpIconPath As String, lpiIcon As Long) As Long
说明
这个函数可判断一个可执行程序或 DLL 中是否存在图标，或是否有图标与系统注册表中指定的文件存在关联。随后，它允许我们提取出那些图标
返回值
Long，如果找到任何图标，就返回图标的句柄；否则返回零
参数表
参数类型及说明
hInstLong，当前应用程序的实例句柄。也可用 GetWindowWord 函数取得拥有一个窗体或控件的示例的句柄
lpIconPathString，指定一个文件名，准备从该文件中提取图标。如果文件并非执行程序或 DLL 本身，但通过系统注册表与一个可执行文件关联，就用这个字串装载可执行程序的名字
lpiIconLong，在其中装载图标在可执行文件中的资源标识符
注解
注意至少要把 lpIconPath 字串定义成 MAX_PATH 个字符的长度

Top
ExtractIcon
ExtractIcon

VB 声明
Declare Function ExtractIcon Lib "shell32.dll" Alias "ExtractIconA" (ByVal hInst As Long, ByVal lpszExeFileName As String, ByVal nIconIndex As Long) As Long
说明
判断一个可执行文件或 DLL 中是否有图标存在，并将其提取出来
返回值
Long，如成功，返回指向图标的句柄；如文件中不存在图标，则返回零。如果 nIconIndex 设为-1，就返回文件中的图标总数
参数表
参数类型及说明

hInstLong，当前应用程序的实例句柄。也可用 GetWindowWord 函数取得拥有一个窗体或控件的实例的句柄
lpszExeFileNameString，在其中提取图标的那个程序的全名
nIconIndexLong，欲获取的图标的索引。如果为-1，表示取得文件中的图标总数

Top
GetBitmapBits
GetBitmapBits

VB 声明
Declare Function GetBitmapBits Lib "gdi32" Alias "GetBitmapBits" (ByVal hBitmap As Long, ByVal dwCount As Long, lpBits As Any) As Long
说明
将来自位图的二进制位复制到一个缓冲区
返回值
Long，如执行成功，返回位图中的字节数量；零表示失败。会设置 GetLastError
参数表
参数类型及说明
hBitmapLong，位图的句柄
dwCountLong，欲复制的字节数。如设为零，表示取得位图中的字节数
lpBitsAny，指向容纳位图位的一个缓冲区的指针。注意事先将缓冲区至少初始化成 dwCount 个字节
注解
虽然这个函数能正常工作，但强烈建议使用与设备无关的位图（GetDIBits）

Top
GetBitmapDimensionEx
GetBitmapDimensionEx

VB 声明
Declare Function GetBitmapDimensionEx Lib "gdi32" Alias "GetBitmapDimensionEx" (ByVal hBitmap As Long, lpDimension As SIZE) As Long
说明
取得一幅位图的宽度和高度。它们是由 SetBitmapDimensionEx 函数设置的。Windows 不会使用位图的大小
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hBitmapLong，位图句柄
lpDimensionSIZE，这个结构用于容纳由 SetBitmapDimensionEx 函数设置的位图的大小。这个大小以 1mm 的十分之一为单位
注解
参考 SetBitmapDimensionEx 函数

Top
GetDIBColorTable
GetDIBColorTable

VB 声明
Declare Function GetDIBColorTable Lib "gdi32" Alias "GetDIBColorTable" (ByVal hDC As Long, ByVal un1 As Long, ByVal un2 As Long, pRGBQuad As RGBQUAD) As Long
说明
从选入设备场景的 DIBSection 中取得颜色表信息
返回值
Long，取回的颜色条目数量，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hDCLong，已选入了一个 DIBSection 对象的设备场景
un1Long，颜色表中欲取回的第一个条目的索引
un2Long，欲取回的条目数量
pRGBQuadRGBQUAD，这个结构数组用于装载颜色表信息的第一个条目
注解
如 DIBSection 为每个像素使用了 8 个以上的二进制位，则不会再用颜色表

Top
GetDIBits
GetDIBits

VB 声明
Declare Function GetDIBits Lib "gdi32" Alias "GetDIBits" (ByVal aHDC As Long, ByVal hBitmap As Long, ByVal nStartScan As Long, ByVal nNumScans As Long, lpBits As Any, lpBI As BITMAPINFO, ByVal wUsage As Long) As Long
说明
将来自一幅位图的二进制位复制到一幅与设备无关的位图里
返回值
Long，非零表示成功，零表示失败。在 Windows 95 中，返回值是返回的扫描线数量
参数表
参数类型及说明
aHDCLong，定义了与设备有关位图 hBitmap 的配置信息的一个设备场景的句柄
hBitmapLong，源位图的句柄。绝对不能将这幅位图选入设备场景
nStartScanLong，欲复制到 DIB 中的第一条扫描线的编号
nNumScansLong，欲复制的扫描线数量
lpBitsAny，指向一个缓冲区的指针。这个缓冲区将用于装载采用 DIB 格式的信息，但不取回数据（用 ByVal 传递零值）
lpBIBITMAPINFO，对 lpBits DIB 的格式及颜色进行说明的一个结构。在 BITMAPINFOHEADER 结构中，从 biSize 到 biCompression 之间的所有字段都必须初始化
wUsageLong，下述常数之一：

DIB_PAL_COLORS 在颜色表中装载一个 16 位所以数组，它们与当前选定的调色板有关
DIB_RGB_COLORS 在颜色表中装载 RGB 颜色
注解
起始扫描线与起点有关。除非将 BITMAPINFOHEADER 结构的 biHeight 字段设为负值，否则起点就位于左下角

Top
GetIconInfo
GetIconInfo

VB 声明
Declare Function GetIconInfo Lib "user32" Alias "GetIconInfo" (ByVal hIcon As Long, piconinfo As ICONINFO) As Long
说明
取得与图标有关的信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hIconLong，图标句柄
piconinfoICONINFO，这个结构装载的信息与包含了 XOR 及 AND 位图的图标有关
注解
函数返回时，由 ICONINFO 结构载入的位图必须由应用程序删去

Top
GetStretchBltMode
GetStretchBltMode

VB 声明
Declare Function GetStretchBltMode Lib "gdi32" Alias "GetStretchBltMode" (ByVal hdc As Long) As Long
说明
判断 StretchBlt 和 StretchDIBits 函数采用的伸缩模式。伸缩模式决定了 Windows 如何控制在伸缩过程中剔除的扫描线
返回值
Long，取得当前的伸缩模式。零表示出错
参数表
参数类型及说明
hdcLong，设备场景的句柄
注解
参考 SetStretchBltMode 函数

Top
LoadBitmap

LoadBitmap, LoadBitmapBynum

VB 声明
Declare Function LoadBitmap& Lib "user32" Alias "LoadBitmapA" (ByVal hInstance As Long, ByVal lpBitmapName As String)
Declare Function LoadBitmapBynum& Lib "user32" Alias "LoadBitmapA" (ByVal hInstance As Long, ByVal lpBitmapName As Long)
说明
从指定的模块或应用程序实例中载入一幅位图。LoadBitmapBynum 是 LoadBitmap 函数的类型安全声明
返回值
Long，已载入的位图的句柄，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hInstanceLong，一个 DLL 的模块句柄；或者一个实例句柄，它指定了包含位图的可执行文件
lpBitmapNameString，作为一个字串，指定欲载入的位图资源。作为一个长整数，指定欲载入的资源 ID。或者一个常数，代表某幅固有系统位图。如装载一幅固有系统位图，hInstance 参数应设为 0。常数在 api32.txt 文件中用前缀 OBM_ 标志。OBM_REDUCE 引用的是标题栏右侧显示的下箭头位图，用于最小化一个窗口。参考 api32.txt，其中有系统固有位图的完整列表
注解
一旦系统位图不再需要，必须用 DeleteObject 函数删除由这个函数取得的位图

Top
LoadCursor
LoadCursor, LoadCursorBynum

VB 声明
Declare Function LoadCursor& Lib "user32" Alias "LoadCursorA" (ByVal hInstance As Long, ByVal lpCursorName As String)
Declare Function LoadCursorBynum& Lib "user32" Alias "LoadCursorA" (ByVal hInstance As Long, ByVal lpCursorName As Long)
说明
从指定的模块或应用程序实例中载入一个鼠标指针。LoadCursorBynum 是 LoadCursor 函数的类型安全声明
返回值
Long，执行成功则返回已载入的指针的句柄；零表示失败。在 Windows 95 和 Win16 环境中，这个函数只能载入标准尺寸的图标
参数表
参数类型及说明
hInstanceLong，一个 DLL 的模块句柄；或者一个实例句柄，指定包含了鼠标指针的可执行程序
lpCursorNameString，作为一个字串，指定欲载入的指针资源。作为一个长整数值，指定欲

载入的资源 ID；或者设置一个常数，代表某幅固有系统指针。如装载的是一个固有系统指针，注意 hInstance 参数应设为零。在 api32.txt 文件中以前缀 IDC_ 作为标志
注解
不要清除固有系统指针或从属于其他应用程序的指针。注意 lpCursorName 引用的是一个指针资源。因为假如名字引用的是一个有效的资源。即使它并非指针资源，该函数也会返回一个非零的句柄

Top
LoadCursorFromFile
LoadCursorFromFile

VB 声明
Declare Function LoadCursorFromFile Lib "user32" Alias "LoadCursorFromFileA" (ByVal lpFileName As String) As Long
说明
在一个指针文件或一个动画指针文件（扩展名分别是.cur 和.ani）的基础上创建一个指针
返回值
Long，执行成功则返回指向指针的一个句柄，零表示失败。如果失败，会将 GetLastError 设置为常数 ERROR_FILE_NOT_FOUND
参数表
参数类型及说明
lpFileNameString，包含指针的那个文件的名字
注解
如果将 lpFileName 定义成 Long 型数值（ByVal As Long），就可以（理论上）传递带 OCR_ 前缀的常数，以便获取一个固有系统指针。这一做法在 Windows NT 3.5 下不适用

Top
LoadIcon
LoadIcon, LoadIconBynum

VB 声明
Declare Function LoadIcon& Lib "user32" Alias "LoadIconA" (ByVal hInstance As Long, ByVal lpIconName As String)
Declare Function LoadIconBynum& Lib "user32" Alias "LoadIconA" (ByVal hInstance As Long, ByVal lpIconName As Long)
说明
从指定的模块或应用程序实例中载入一个图标。其中，LoadIconBynum 是 LoadIcon 函数的类型安全声明
返回值
Long，执行成功则返回已载入的图标的句柄；零表示失败。会设置 GetLastError
参数表
参数类型及说明
hInstanceLong，一个 DLL 的模块句柄；或者一个实例句柄，指定包含了图标的可执行程序
lpIconNameString，作为一个字串，指定欲载入的图标资源。作为一个长整数值，指定欲载

入的资源 ID；或者设置一个常数，代表某幅固有系统图标。如装载的是一个固有系统图标，注意 hInstance 参数应设为零。在 api32.txt 文件中以前缀 IDI_ 作为标志

注解
在 Windows 95 和 Win16 环境中，这个函数只能载入标准尺寸的图标。用 GetSystemMetrics 可以获得 SM_CXICON 和 SM_CYICON 标准图标设置。用 LoadImage 则可载入非标准尺寸的图标
文件中可能包含了同一个图标资源的多种版本。这个函数会自动挑选与当前显示模式最相配的一种

Top
LoadImage
LoadImage, LoadImageBynum

VB 声明
Declare Function LoadImage& Lib "user32" Alias "LoadImageA" (ByVal hInst As Long, ByVal lpsz As String, ByVal un1 As Long, ByVal n1 As Long, ByVal n2 As Long, ByVal un2 As Long)
Declare Function LoadImageBynum& Lib "user32" Alias "LoadImageA" (ByVal hInst As Long, ByVal lpsz As Long, ByVal un1 As Long, ByVal n1 As Long, ByVal n2 As Long, ByVal un2 As Long)
说明
载入一个位图、图标或指针
返回值
Long，执行成功则返回对象的一个句柄；零表示失败
参数表
参数类型及说明
hInstLong，要从其中载入图象的 DLL 或应用程序模块或实例句柄。零表示装载一幅固有图象
lpszString，欲载入图象的名字。如指定了 hInst，就用这个参数指定资源或资源的标志符（标志符是一个长整数）。如 hInst 为空，而且已指定了 LR_LOADFROMFILE，那么这个参数代表文件名（位图、图标或指针文件）。如果是个 Long 型值，这个参数就代表固有位图、图标或指针的编号
un1Long，下述常数之一，指定了欲载入的图象类型：IMAGE_BITMAP， IMAGE_CURSOR，IMAGE_ICON
n1,n2Long，要求的图象宽度和高度。图象会根据情况自动伸缩。如设为零，表示用图象的默认大小
un2Long，下述常数的任意组合，它们都在 api32.txt 文件中得到了定义：
LR_DEFAULTCOLOR 以常规方式载入图象
LR_LOADREALSIZE 不对图象进行缩放处理。忽略 n1 和 n2 的设置
LR_CREATEDIBSECTION 如果指定了 IMAGE_BITMAP，就返回 DIBSection 的句柄，而不是位图的句柄
LR_DEFAULTSIZE 如果 n1 和 n2 为零，就使用由系统定义的图象默认大小，而不是图象本身定义的大小
LR_LOADFROMFILE 如 hInst 为零，lpsz 就代表要载入适当类型的一个文件的名字，仅适

用于 Win95
LR_LOADMAP3DCOLORS 将图象中的深灰、灰、以及浅灰像素都替换成 COLOR_3DSHADOW，COLOR_3DFACE 以及 COLOR_3DLIGHT 的当前设置
LR_LOADTRANSPARENT 与图象中第一个像素相符的所有像素都由系统替换
LR_MONOCHROME 将图象转换成单色
LR_SHARED 将图象作为一个共享资源载入。在 NT 4.0 中装载固有资源时要用到这个设置

Top
MaskBlt
MaskBlt

VB 声明
Declare Function MaskBlt Lib "gdi32" Alias "MaskBlt" (ByVal hdcDest As Long, ByVal nXDest As Long, ByVal nYDest As Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal hdcSrc As Long, ByVal nXSrc As Long, ByVal nYSrc As Long, ByVal hbmMask As Long, ByVal xMask As Long, ByVal yMask As Long, ByVal dwRop As Long) As Long
说明
执行复杂的图象传输，同时进行掩模（MASK）处理
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcDestLong，目标设备场景
nXDest,nYDestLong，目标图象左上角的 x,y 坐标
nWidth,nHeightLong，图象在目标设备场景中的宽度和高度
hdcSrcLong，源设备场景
nXSrc,nYSrcLong，源图象的左上角 x,y 坐标
hbmMaskLong，作为掩模使用的一幅单色位图的句柄。如果 dwRop 代码包括一个源，那么这幅位图必须与源尺寸相同，否则必须与目标尺寸相符
xMask,yMaskLong，单色掩模位图的 x,y 偏移。这样便允许我们创建一幅使用了多个掩模的大型位图
dwRopLong，一种特殊的光栅运算，在传输过程中使用
适用平台
Windows NT
注解
dwRop 代码是一种非标准的光栅运算代码，由两个普通的光栅运算代码组成：一个前景代码以及一个背景代码
在掩模位图设为 1 的每个像素处，都在传输过程中应用前景转换处理。如对应的掩蔽位图像素为零，则应用背景转换。如果未指定掩模位图，那么这个函数就执行与 BitBlt 相同的操作
如果对源应用了旋转或剪切处理，则函数调用会失败
注意可用 GetDeviceCaps 判断这个函数是否得到了一个特定设备场景的支持

Top
PatBlt

PatBlt

VB 声明
Declare Function PatBlt Lib "gdi32" Alias "PatBlt" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal dwRop As Long) As Long
说明
在当前选定的刷子的基础上，用一个图案填充指定的设备场景
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，欲描绘的一个设备场景的句柄
x,yLong，对目标 DC 中目标矩形左上角位置进行定义的一个点，用逻辑坐标表示
nWidth,nHeightLong，目标矩形的宽度和高度，用逻辑坐标表示
dwRopLong，传输过程中欲进行的光栅运算。对一个源进行引用的光栅运算也许不能在这个函数中使用
注解
可用 GetDeviceCaps 判断这个函数是否得到了一个特定设备场景的支持

Top
PlgBlt
PlgBlt

VB 声明
Declare Function PlgBlt Lib "gdi32" Alias "PlgBlt" (ByVal hdcDest As Long, lpPoint As POINTAPI, ByVal hdcSrc As Long, ByVal nXSrc As Long, ByVal nYSrc As Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal hbmMask As Long, ByVal xMask As Long, ByVal yMask As Long) As Long
说明
复制一幅位图，同时将其转换成一个平行四边形。利用它可对位图进行旋转处理
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcDestLong，图象使用的目标设备场景
lpPointPOINTAPI，POINTAPI 结构数组中使用的第一个条目。第一个点对应于一个平行四边形左上角位置；第二个点代表右下角位置；第三个点代表左下角位置；第四个点是在前三个点的基础上导出的
hdcSrcLong，图象的源设备场景
nXSrc,nYSrcLong，源图象左上角的 x,y 坐标，采用逻辑坐标系统表示
nWidth,nHeightLong，源图象大小，用逻辑坐标表示
hbmMaskLong，一个可选的句柄，指向一个单色掩模。如设定了这个参数，那么只有与掩模值 1 对应的二进制位才会传输到目的地
xMask,yMaskLong，掩模位图欲使用区域左上角的 x,y 坐标

适用平台
Windows NT
注解
如果对源图象应用了旋转或剪切处理，这个函数的执行就会失败。可用 GetDeviceCaps 判断这个函数是否得到了一个特定设备场景的支持

Top
sample
例程
图形菜单（5K）这个例程有完整的中文注释！演示如何实现图形的菜单条目，用到了几个菜单的 api 函数，本站均有介绍
屏蔽文本框菜单（2.08K）用 api 函数拦截消息，从而屏蔽掉文本框菜单，参考文档中的屏蔽文本框菜单一文
奇形怪状的窗体（6.19K）圆角矩形的窗体中有一个椭圆形的洞。参考奇形怪状的窗体一文
系统菜单（8.89K）在系统菜单里加上自己的菜单项，并可以恢复到原来的菜单
提取图标（9.27K）从 DLL 和 EXE 文件中提取图标，并显示在图片框上
API 函数示例（2.6K）用于列表框
SetBitmapBits
SetBitmapBits

VB 声明
Declare Function SetBitmapBits Lib "gdi32" Alias "SetBitmapBits" (ByVal hBitmap As Long, ByVal dwCount As Long, lpBits As Any) As Long
说明
将来自缓冲区的二进制位复制到一幅位图
返回值
Long，执行成功则返回字节数量，零表示失败
参数表
参数类型及说明
hBitmapLong，位图的句柄
dwCountLong，欲复制的字节数量
lpBitsAny，指向一个缓冲区的指针。这个缓冲区包含了为位图正确格式化的位图位
注解
在 Win32 中，应使用与设备无关位图

Top
SetBitmapDimensionEx
SetBitmapDimensionEx

VB 声明
Declare Function SetBitmapDimensionEx Lib "gdi32" Alias "SetBitmapDimensionEx" (ByVal hbm As Long, ByVal nX As Long, ByVal nY As Long, lpSize As SIZE) As Long
说明
设置一幅位图的宽度。以一毫米的十分之一为单位。Windows 不使用这种信息，但也许能通

过 GetBitmapDimensionEx 函数获取
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hbmLong，一幅位图的句柄
nX,nYLong，位图的建议大小，以 0.1mm 为单位
lpSizeSIZE，用于载入前一个位图大小的结构

Top
SetDIBits
SetDIBits

VB 声明
Declare Function SetDIBits Lib "gdi32" Alias "SetDIBits" (ByVal hdc As Long, ByVal hBitmap As Long, ByVal nStartScan As Long, ByVal nNumScans As Long, lpBits As Any, lpBI As BITMAPINFO, ByVal wUsage As Long) As Long
说明
将来自与设备无关位图的二进制位复制到一幅与设备有关的位图里
返回值
Long，执行成功则返回扫描线的数量，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，指向一个设备场景的句柄，那个设备场景定义了与设备有关位图（hBitmap）的配置
hBitmapLong，目标位图的一个句柄。这幅位图绝对不能选入一个设备场景
nStartScanLong，lpBits 数组中第一条扫描线的编号。如 lpBI 之 BITMAPINFOHEADER 部分的 biHeight 字段是正数，那么这条扫描线就会从位图的底部开始计算；如果是负数，就从顶部开始计算
nNumScansLong，欲复制的扫描线数量
Any，指向一个缓冲区的指针。这个缓冲区包含了以 DIB 格式描述的位图数据；这种格式是由 lpBI 指定的
lpBIBITMAPINFO，对 lpBits DIB 的格式和颜色进行描述的一个结构
wUsageLong，下述常数之一
DIB_PAL_COLORS 颜色表是一个整数数组，其中包含了与目前选入 hdc 设备场景的调色板相关的索引
DIB_RGB_COLORS 颜色表包含了 RG 颜色
注解
用 GetDeviceCaps 判断设备是否支持这个函数

Top
SetDIBitsToDevice
SetDIBitsToDevice

VB 声明
Declare Function SetDIBitsToDevice Lib "gdi32" Alias "SetDIBitsToDevice" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long, ByVal dx As Long, ByVal dy As Long, ByVal SrcX As Long, ByVal SrcY As Long, ByVal Scan As Long, ByVal NumScans As Long, Bits As Any, BitsInfo As BITMAPINFO, ByVal wUsage As Long) As Long
说明
将一幅与设备无关位图的全部或部分数据直接复制到一个设备。这个函数在设备中定义了一个目标矩形，以便接收位图数据。它也在 DIB 中定义了一个源矩形，以便从中提取数据
返回值
Long，执行成功则返回扫描线的数量，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，一个设备场景的句柄。该场景用于接收位图数据
x,yLong，用逻辑坐标表示的目标矩形的起点
dx,dyLong，用目标矩形的设备单位表示的宽度及高度
SrcX,SrcYLong，用设备坐标表示的源矩形在 DIB 中的起点
ScanLong，Bits 数组中第一条扫描线的编号。如 BitsInfo 之 BITMAPINFOHEADER 部分的 biHeight 字段是正数，那么这条扫描线就会从位图的底部开始计算；如果是负数，就从顶部开始计算
NumScansLong，欲复制的扫描线数量
BitsAny，指向一个缓冲区的指针。这个缓冲区包含了以 DIB 格式描述的位图数据；这种格式是由 BitsInfo 指定的
BitsInfoBITMAPINFO，对 Bits DIB 的格式和颜色进行描述的一个结构
wUsageLong，下述常数之一
DIB_PAL_COLORS 颜色表是一个整数数组，其中包含了与目前选入 hdc 设备场景的调色板相关的索引
DIB_RGB_COLORS 颜色表包含了 RG 颜色
注解
用 GetDeviceCaps 判断设备是否支持这个函数

Top
SetStretchBltMode
SetStretchBltMode

VB 声明
Declare Function SetStretchBltMode Lib "gdi32" Alias "SetStretchBltMode" (ByVal hdc As Long, ByVal nStretchMode As Long) As Long
说明
指定 StretchBlt 和 StretchDIBits 函数的伸缩模式。这种伸缩模式定义了 Windows 如何对伸缩过程中剔除的扫描线进行控制。对于 VB 窗体和控件，倘若在 API 绘图过程中使用这个函数，建议恢复原来的 StretchBlt 模式
返回值
Long，上一次伸缩模式的值，零表示失败
参数表

参数类型及说明
hdcLong，设备场景的句柄
nStretchModeLong，新伸缩模式建立在下述任何一个常数的基础上，它们均在 API32.TXT 文件中得到了定义：
STRETCH_ANDSCANS 默认设置。剔除的线段与剩下的线段进行 AND 运算。这个模式通常应用于采用了白色背景的单色位图
STRETCH_DELETESCANS 剔除的线段被简单的清除。这个模式通常用于彩色位图
STRETCH_ORSCANS 剔除的线段与剩下的线段进行 OR 运算。这个模式通常应用于采用了白色背景的单色位图
STRETCH_HALFTONE 目标位图上的像素块被设为源位图上大致近似的块。这个模式要明显慢于其他模式
注解
如果要对伸缩模式有一个更深刻的印象，可想象一下对白色图象中的一条白色细线进行压缩。压缩过程中，像素会从图象中删去。为避免线段消失，在删除它们之前，有必要先对线段中的像素与邻近像素进行 AND 运算。为达到这个目的，应考虑选用 STRETCH_ANDSCANS 伸缩模式

Top
StretchBlt
StretchBlt

VB 声明
Declare Function StretchBlt Lib "gdi32" Alias "StretchBlt" (ByVal hdc 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 nSrcWidth As Long, ByVal nSrcHeight As Long, ByVal dwRop As Long) As Long
说明
将一幅位图从一个设备场景复制到另一个。源和目标 DC 相互间必须兼容。这个函数会在设备场景中定义一个目标矩形，并在位图中定义一个源图象。源矩形会根据需要进行伸缩，以便与目标矩形的大小相符
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，目标设备场景
x,yLong，目标矩形左上角的 x,y 坐标，以逻辑坐标表示
nWidth,nHeightLong，目标矩形的宽度和高度，以逻辑坐标表示
hSrcDCLong，源设备场景。如光栅运算未指定一个源，则这个参数应为零
xSrc,ySrcLong，用源 DC 的逻辑坐标表示的源矩形左上角位置
nSrcWidth,nSrcHeightLong，分别指定用逻辑单位（以源 DC 为基础）传输的一幅图象的宽度和高度。如其中有一个参数的符号（指正负号）与对应的目标参数不符，位图就会在对应的轴上作镜像转换处理
dwRopLong，传输过程中进行的光栅运算。如刷子属于光栅运算的一部分，就使用选入目标 DC 的刷子

注解
可用 GetDeviceCaps 函数判断特定的设备场景是否支持此函数
不可选择对源位图进行剪切或旋转处理，源位图也不能是一个图元文件设备场景

Top
StretchDIBits
StretchDIBits

VB 声明
Declare Function StretchDIBits Lib "gdi32" Alias "StretchDIBits" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long, ByVal dx As Long, ByVal dy As Long, ByVal SrcX As Long, ByVal SrcY As Long, ByVal wSrcWidth As Long, ByVal wSrcHeight As Long, lpBits As Any, lpBitsInfo As BITMAPINFO, ByVal wUsage As Long, ByVal dwRop As Long) As Long
说明
将一幅与设备无关位图的全部或部分数据直接复制到指定的设备场景。这个函数在设备场景中定义了一个目标矩形，用于接收位图数据。它也在 DIB 中定义了一个源矩形，以便从中提取数据。根据设备场景的 StretchBlt 模式（由 SetStretchBltMode 函数决定），源矩形会根据需要调整，以便符合目标矩形的要求
返回值
Long，如函数执行成功，返回欲复制的扫描线的数量；如返回常数 GDI_ERROR，表示出错
参数表
参数类型及说明
hdcLong，一个设备场景的句柄。该场景用于接收位图数据
x,yLong，用逻辑坐标表示的目标矩形的起点
dx,dyLong，目标矩形的宽度及高度，以逻辑坐标表示
SrcX,SrcYLong，用设备坐标表示的源矩形在 DIB 中的起点
wSrcWidth,wSrcHeightLong，源矩形的宽度与高度，用设备坐标表示。如其中有一个参数的符号（指正负号）与对应的目标参数不符，位图就会在对应的轴上作镜像转换
lpBitsAny，指向一个缓冲区的指针。这个缓冲区包含了以 DIB 格式描述的位图数据；这种格式是由 lpBitsInfo 指定的
lpBitsInfoBITMAPINFO，对 lpBits DIB 的格式和颜色进行描述的一个结构
wUsageLong，下述常数之一
DIB_PAL_COLORS 颜色表是一个整数数组，其中包含了与目前选入 hdc 设备场景的调色板相关的索引
DIB_RGB_COLORS 颜色表包含了 RG 颜色
dwRopLong，欲进行的光栅运算

Top
绘图函数
绘图函数：共七页。第一页，第二页，第三页，第四页，第五页，第六页，第七页
AbortPath 抛弃选入指定设备场景中的所有路径。也取消目前正在进行的任何路径的创建工作
AngleArc 用一个连接弧画一条线
Arc 画一个圆弧

BeginPath 启动一个路径分支
CancelDC 取消另一个线程里的长时间绘图操作
Chord 画一个弦
CloseEnhMetaFile 关闭指定的增强型图元文件设备场景，并将新建的图元文件返回一个句柄
CloseFigure 描绘到一个路径时，关闭当前打开的图形
CloseMetaFile 关闭指定的图元文件设备场景，并向新建的图元文件返回一个句柄
CopyEnhMetaFile 制作指定增强型图元文件的一个副本（拷贝）
CopyMetaFile 制作指定（标准）图元文件的一个副本
CreateBrushIndirect 在一个 LOGBRUSH 数据结构的基础上创建一个刷子
CreateDIBPatternBrush 用一幅与设备无关的位图创建一个刷子，以便指定刷子样式（图案）
CreateEnhMetaFile 创建一个增强型的图元文件设备场景
CreateHatchBrush 创建带有阴影图案的一个刷子
CreateMetaFile 创建一个图元文件设备场景
AbortPath
AbortPath

VB 声明
Declare Function AbortPath Lib "gdi32" Alias "AbortPath" (ByVal hdc As Long) As Long
说明
抛弃选入指定设备场景中的所有路径。也取消目前正在进行的任何路径的创建工作
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景

Top
AngleArc
AngleArc

VB 声明
Declare Function AngleArc Lib "gdi32" Alias "AngleArc" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long, ByVal dwRadius As Long, ByVal eStartAngle As Double, ByVal eSweepAngle As Double) As Long
说明
用一个连接弧画一条线，参考注解
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hdcLong，要在其中作图的设备场景
x,yLong，对弧进行描述的一个圆的中心点坐标
dwRadiusLong，圆的半径
eStartAngleDouble，线同圆连接时的角度（以度数为单位）

eSweepAngleDouble，弧在圆上占据的范围（以度数为单位）
注解
注意 eStartAngle 和 eSweepAngle 参数是以度数为单位指定的，而且应该是单精度数（Single）而不是双精度。相应的函数声明为：Declare Function AngleArc& Lib "gdi32" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long, ByVal dwRadius As Long, ByVal eStartAngle As Single, ByVal eSweepAngle As Single)。
我的理解：本文开头的函数声明复制于 vb 的 api 文本查看器，此处的声明来自于我的参考资料，也不知谁对谁错。参数表的说明，按 vb 的 api 文本查看器中复制来的声明中的数据类型。请使用者注意

Top
Arc
Arc,ArcTo

VB 声明
Declare Function Arc& Lib "gdi32" (ByVal hdc As Long, ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long, ByVal X3 As Long, ByVal Y3 As Long, ByVal X4 As Long, ByVal Y4 As Long)
Declare Function ArcTo& Lib "gdi32" (ByVal hdc As Long, ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long, ByVal X3 As Long, ByVal Y3 As Long, ByVal X4 As Long, ByVal Y4 As Long)
说明
象注解中那样画一个圆弧。（X1,Y1）和（X2,Y2）定义了椭圆的一个范围（约束）矩形。从矩形中心点到点（X3,Y3）的一条线段与椭圆的交点标志着圆弧的起点。而到（X4,Y4）的一条线与椭圆的交点则标志着圆弧的终点。ArcTo 函数会将当前画笔位置设为弧的终点，而 Arc 函数则不会对当前的画笔位置造成影响
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hdcLong，一个显示场景的句柄
X1,Y1Long，指定围绕椭圆的一个矩形的左上角位置
X2,Y2Long，指定围绕椭圆的一个矩形的右下角位置
X3,Y3Long，指定圆弧起点
X4,Y4Long，指定圆弧终点
注解
在 win16 和 win95 中，约束矩形的宽度和高度必须在 3——32766 间。绘图方向肯定是逆时针方向。
在 win nt 中：绘图方向由 SetArcDirection 函数决定。默认为逆时针方向

Top
BeginPath
BeginPath

VB 声明
Declare Function BeginPath Lib "gdi32" Alias "BeginPath" (ByVal hdc As Long) As Long
说明
启动一个路径分支。在这个命令后执行的 GDI 绘图命令会自动成为路径的一部分。对线段的连接会结合到一起。设备场景中任何现成的路径都会被清除。参考下表，其中列出的函数都可记录到路径中
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，欲在其中记录的设备场景

合法的路径函数
函数 Windows NTWindows 95 函数 Windows NTWindows 95
AngleArcYesNoArcYesNo
ArcToYesNoChordYesNo
EllipseYesNoExtTextOutYesYes
LineToYesYesMoveToExYesYes
PieYesNoPolyBezierYesYes
PolyBezierToYesYesPolyDrawYesNo
PolygonYesYesPolylineYesYes
PolylineToYesYesPolyPolygonYesYes
PolyPolylineYesYesRectangleYesNo
RoundRectYesNoTextOutYesYes

Top
CancelDC
CancelDC

VB 声明
Declare Function CancelDC Lib "gdi32" Alias "CancelDC" (ByVal hdc As Long) As Long
说明
取消另一个线程里的长时间绘图操作
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hdcLong，要在其中取消绘图操作的设备场景
注解
绘图操作有时可能会相当花时间。在多线程应用程序中，这个命令允许一个线程终止另一个在线程里的绘图操作

Top

Chord
Chord

VB 声明
Declare Function Chord Lib "gdi32" Alias "Chord" (ByVal As Long, ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long, ByVal X3 As Long, ByVal Y3 As Long, ByVal X4 As Long, ByVal Y4 As Long) As Long
说明
象注解中那样画一个弦。(X1,Y1)和(X2,Y2)定义了椭圆的一个范围(约束)矩形。点(X3,Y3)和点（X4,Y4）定义了一条线段。该线段与椭圆之间的区域就是“弦”
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，一个显示场景的句柄
X1,Y1Long，指定围绕椭圆的一个矩形的左上角位置
X2,Y2Long，指定围绕椭圆的一个矩形的右下角位置
X3,Y3Long，指定与椭圆相交的一条线的一个点
X4,Y4Long，指定与椭圆相交的一条线的另一个点
注解
在 win16 和 win95 中，约束矩形的宽度和高度必须在 3——32766 个单位之间。请参考 Arc 获得更详细的图例

Top
CloseEnhMetaFile
CloseEnhMetaFile

VB 声明
Declare Function CloseEnhMetaFile Lib "gdi32" Alias "CloseEnhMetaFile" (ByVal hdc As) As Long
说明
关闭指定的增强型图元文件设备场景，并将新建的图元文件返回一个句柄
返回值
Long，指向增强型图元文件的一个句柄。也许能用 PlayEnhMetaFile 函数回放图元文件。零表示出错（原文：A handle to the enhanced metafile. The PlayEnhMetaFile function may be used to play the metafile. Zero on error.）
参数表
参数类型及说明
hdcLong，由 CreateEnhMetaFile 函数返回的一个图元文件设备场景

Top
CloseFigure
CloseFigure

VB 声明
Declare Function CloseFigure Lib "gdi32" Alias "CloseFigure" (ByVal hdc As Long) As Long
说明
描绘到一个路径时，关闭当前打开的图形
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，包含了一个打开的 Path 分支的设备场景
注解
如准备在一个路径里描绘一系列线段，就会有一个打开的图形。调用这个函数的时候，windows 会在当前位置和图形的起始处（通常是最后一个 MoveToEx 操作设置画笔位置的地方）画一条线。图中的这条线和第一条线会连接到一起。注意倘若自己描绘这条线，图形仍会处于打开状态——即使起点与终点相同。这样便造成了与几何画笔的差异。利用 CloseFigure，线段会连接到一起——否则它们会用笔尖显示出来。一旦关闭了图形，在路径中画的下一条线就会从一幅新图形开始。打开的图形会被那些用于填充路径的函数自动关闭

Top
CloseMetaFile
CloseMetaFile

VB 声明
Declare Function CloseMetaFile Lib "gdi32" Alias "CloseMetaFile" (ByVal hMF As Long) As Long
说明
关闭指定的图元文件设备场景，并向新建的图元文件返回一个句柄
返回值
Long，指向图元文件的一个句柄。也许能用 PlayMetaFile 回放图元文件。零表示出错
参数表
参数类型及说明
hMFLong，由 CreateMetaFile 函数返回的一个图元文件设备场景
注解
图元文件不再需要后，一定要用 DeleteMetaFile 函数删除图元文件，并释放它的资源

Top
CopyEnhMetaFile
CopyEnhMetaFile

VB 声明
Declare Function CopyEnhMetaFile Lib "gdi32" Alias "CopyEnhMetaFileA" (ByVal hemfSrc As Long, ByVal lpszFile As String) As Long
说明
制作指定增强型图元文件的一个副本（拷贝）
返回值

Long，如执行成功，返回副本的句柄。零表示失败。会设置 GetLastError
参数表
参数类型及说明
hemfSrcLong，欲复制的增强型图元文件的句柄
lpszFileString，副本的文件名（要在磁盘上创建一个新的图元文件）。可用 vbNullString 向这个参数传递一个 NULL 值，以便在内存中创建副本

Top
CopyMetaFile
CopyMetaFile

VB 声明
Declare Function CopyMetaFile Lib "gdi32" Alias "CopyMetaFileA" (ByVal hMF As Long, ByVal lpFileName As String) As Long
说明
制作指定（标准）图元文件的一个副本
返回值
Long，如执行成功，则返回副本的句柄。零表示失败。会设置 GetLastError
参数表
参数类型及说明
hMFLong，指向欲复制的图元文件的句柄
lpFileNameString，新图元文件的文件名。可用 vbNullString 向这个参数传递一个 NULL 值，以便在内存中创建副本

Top
CreateBrushIndirect
CreateBrushIndirect

VB 声明
Declare Function CreateBrushIndirect Lib "gdi32" Alias "CreateBrushIndirect" (lpLogBrush As LOGBRUSH) As Long
说明
在一个 LOGBRUSH 数据结构的基础上创建一个刷子
返回值
Long，如执行成功，返回指向新刷子的一个句柄。零表示失败
参数表
参数类型及说明
lpLogBrushLOGBRUSH
注解
如不再需要，请用 DeleteObject 函数删除刷子。也请参考 CreateBrush 函数，它的参数与 LOGBRUSH 结构的字段是对应的

Top
CreateDIBPatternBrush

CreateDIBPatternBrush,CreateDIBPatternBrushPt

VB 声明
Declare Function CreateDIBPatternBrush& Lib "gdi32" (ByVal hPackedDIB As Long, ByVal wUsage As Long)
Declare Function CreateDIBPatternBrushPt& Lib "gdi32" (lpPackedDIB As Any, ByVal wUsage As Long)
说明
用一幅与设备无关的位图创建一个刷子，以便指定刷子样式（图案）
返回值
Long，如执行成功，返回指向刷子的一个句柄。零表示失败
参数表
参数类型及说明
hPackedDIB,lpPackedDIBLong，hPackedDIB 是指向一个内存块的全局内存句柄。那个内存块包含了一个 BITMAPINFO 结构，后面跟随一幅与设备无关的位图。lpPackedDIB 是具有相同配置的一个内存块的地址。如指定了单色 DIB，DIB 颜色就会忽略，而换用文本和背景颜色
wUsageLong，下述常数之一：
DIB_PAL_COLORSDIB 颜色表，包含了当前逻辑调色板的索引
DIB_RGB_COLORSDIB 颜色表，包含了 32 位的 RGB 色值
注解
编制 win32 应用程序的时候，最好使用 CreateDIBPatternBrushPt

Top
CreateEnhMetaFile
CreateEnhMetaFile

VB 声明
Declare Function CreateEnhMetaFile Lib "gdi32" Alias "CreateEnhMetaFileA" (ByVal hdcRef As Long, ByVal lpFileName As String, lpRect As RECT, ByVal lpDescription As String) As Long
说明
创建一个增强型的图元文件设备场景。绘图操作也许在这个设备场景中执行。调用 CloseEnhMetaFile 函数关闭了这个设备场景后，会创建一个图元文件句柄，在其中包含记录下来的绘图命令序列。随后，可在任何设备场景中回放这些命令
返回值
Long，一个增强型图元文件设备场景。零表示函数执行出错。不要将这个设备场景与图元文件句柄混淆起来。图元文件设备场景用于描绘图元文件——这与 GDI 绘图函数作为参数使用的其他任何设备场景是一样的。调用 CloseEnhMetaFile 函数的时候，会获得实际的图元文件句柄
参数表
参数类型及说明
hdcRefLong，一个参考设备场景。函数会用该设备场景在图元文件中保存与创建图元文件的那个设备的分辨率有关的信息。如设为零，表示将整个显示器（屏幕）作为参考设备使用
lpFileNameString，这个图元文件的磁盘文件名。文件应有一个.EMF 扩展名。可用 vbNullString

传递一个 NULL，从而创建内存图元文件
lpRectRECT，一个约束矩形，用于描述图元文件的大小和位置（以 0.01 毫米为单位）。可用它精确定义图元文件的物理尺寸
lpDescriptionString，对图元文件的一段说明。包括创建应用程序的名字、一个 NULL 字符、对图元文件的一段说明以及两个 NULL 字符。如："My app" & chr$(0) & "my metafile" & chr$(0) & chr$(0)。如果不愿意包含一段说明，也可设为 vbNullString
注解
与标准图元文件相比，增强型图元文件的一个优点在于它们包括了对图元文件实际大小和位置进行描述的信息，这些信息与它最开始创建时的情况相符。windows 和绘图程序可读取这种信息，在任何设备上实际重现图元文件

Top
CreateHatchBrush
CreateHatchBrush

VB 声明
Declare Function CreateHatchBrush Lib "gdi32" Alias "CreateHatchBrush" (ByVal nIndex As Long, ByVal crColor As Long) As Long
说明
创建带有阴影图案的一个刷子（阴影图案见注解）
返回值
Long，如执行成功，返回指向新刷子的一个句柄。否则返回零。注意在不需要时，用 DeleteObject 清除刷子
参数表
参数类型及说明
nIndexLong，象下图那样指定一种阴影类型
crColorLong，指定刷子的 RGB 前景色
注解

Top
CreateMetaFile
CreateMetaFile

VB 声明
Declare Function CreateMetaFile Lib "gdi32" Alias "CreateMetaFileA" (ByVal lpString As String) As Long
说明
创建一个图元文件设备场景。绘图操作也许能在这个设备中执行。调用 CloseMetaFile 函数关闭了这个设备场景后，会创建一个图元文件句柄，其中包含了记录下来的绘图命令序列。随后，可在任何设备场景中回放这些命令
返回值
Long，指向图元文件设备场景的一个句柄。零表示出错。注意不要把这个设备场景与图元文件句柄混淆起来。图元文件设备场景用于描绘图元文件——它的用法与 GDI 绘图函数作

为参数使用的其他任何设备场景都是一样的。以后调用了 CloseMetaFile 函数以后，会得到实际的图元文件句柄
参数表
参数类型及说明
lpStringString，欲用来容纳图元文件的文件名。用 vbNullString 可传递一个零值，从而创建一个内存句柄
注解
尽管这个函数能创建基于磁盘的图元文件，但这些图元文件并非通常在 windows 下使用的标准“可放置”图元文件格式

Top
CreatePatternBrush
CreatePatternBrush

VB 声明
Declare Function CreatePatternBrush Lib "gdi32" Alias "CreatePatternBrush" (ByVal hBitmap As Long) As Long
说明
用指定了刷子图案的一幅位图创建一个刷子
返回值
Long，如执行成功，则返回新刷子的一个句柄；否则返回零
参数表
参数类型及说明
hBitmapLong，指向一幅位图的句柄。如指定了单色位图，文本和背景色就会在图案中使用
注解
一旦刷子不再需要，记得用 DeleteObject 函数将其删除。不要在这个函数里使用作为 DIB 分区创建的位图

Top
CreatePen
CreatePen

VB 声明
Declare Function CreatePen Lib "gdi32" Alias "CreatePen" (ByVal nPenStyle As Long, ByVal nWidth As Long, ByVal crColor As Long) As Long
说明
用指定的样式、宽度和颜色创建一个画笔
返回值
Long，如函数执行成功，就返回指向新画笔的一个句柄；否则返回零
参数表
参数类型及说明
nPenStyleLong，指定画笔样式，可以是下述常数之一
PS_SOLID 画笔画出的是实线
PS_DASH 画笔画出的是虚线（nWidth 必须是 1）

PS_DOT 画笔画出的是点线（nWidth 必须是 1）
PS_DASHDOT 画笔画出的是点划线（nWidth 必须是 1）
PS_DASHDOTDOT 画笔画出的是点-点-划线（nWidth 必须是 1）
PS_NULL 画笔不能画图
PS_INSIDEFRAME 画笔在由椭圆、矩形、圆角矩形、饼图以及弦等生成的封闭对象框中画图。如指定的准确 RGB 颜色不存在，就进行抖动处理
nWidthLong，以逻辑单位表示的画笔的宽度
crColorLong，画笔的 RGB 颜色
注解
一旦不再需要画笔，记得用 DeleteObject 函数将其删除

Top
CreatePenIndirect
CreatePenIndirect

VB 声明
Declare Function CreatePenIndirect Lib "gdi32" Alias "CreatePenIndirect" (lpLogPen As LOGPEN) As Long
说明
根据指定的 LOGPEN 结构创建一个画笔
返回值
Long，如执行成功，返回指向新画笔的一个句柄；否则返回零
参数表
参数类型及说明
lpLogPenLOGPEN，逻辑画笔结构。这个结构与 CreatePen 函数的参数非常接近
注解
一旦不再需要画笔，记得用 DeleteObject 函数将其删除

Top
CreateSolidBrush
CreateSolidBrush

VB 声明
Declare Function CreateSolidBrush Lib "gdi32" Alias "CreateSolidBrush" (ByVal crColor As Long) As Long
说明
用纯色创建一个刷子
返回值
Long，如执行成功，返回新刷子的一个句柄；否则返回零
参数表
参数类型及说明
crColorLong，数字的 RGB 彩色
注解
一旦刷子不再需要，就用 DeleteObject 函数将其删除

Top
DeleteEnhMetaFile
DeleteEnhMetaFile

VB 声明
Declare Function DeleteEnhMetaFile Lib "gdi32" Alias "DeleteEnhMetaFile" (ByVal hemf As Long) As Long
说明
删除指定的增强型图元文件
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hemfLong，增强型图元文件的句柄
注解
如图元文件是建立在一个磁盘文件基础上的，那么文件本身就不会由函数删除。这样有可能由 GetEnhMetaFile 函数再次将其打开

Top
DeleteMetaFile
DeleteMetaFile

VB 声明
Declare Function DeleteMetaFile Lib "gdi32" Alias "DeleteMetaFile" (ByVal hMF As Long) As Long
说明
删除指定的图元文件
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hMFLong，图元文件的句柄
注解
如图元文件是建立在一个磁盘文件基础上的，那么文件本身就不会由函数删除。这样有可能造成由 GetEnhMetaFile 函数再次将其打开。然而，这个图元文件并不采用标准可放置图元文件格式

Top
DeleteObject
DeleteObject

VB 声明
Declare Function DeleteObject Lib "gdi32" Alias "DeleteObject" (ByVal hObject As Long) As

Long
说明
用这个函数删除 GDI 对象，比如画笔、刷子、字体、位图、区域以及调色板等等。对象使用的所有系统资源都会被释放
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hObjectLong，一个 GDI 对象的句柄
注解
不要删除一个已选入设备场景的画笔、刷子或位图。如删除以位图为基础的阴影（图案）刷子，位图不会由这个函数删除——只有刷子被删掉

Top
DrawEdge
DrawEdge

VB 声明
Declare Function DrawEdge Lib "user32" Alias "DrawEdge" (ByVal hdc As Long, qrc As RECT, ByVal edge As Long, ByVal grfFlags As Long) As Long
说明
用指定的样式描绘一个矩形的边框
返回值
Long，TRUE（非零）表示成功，否则返回零。会设置 GetLastError。（在 vb 里使用：推荐使用。利用这个函数，我们没有必要再使用许多 3D 边框和面板。所以就资源和内存的占用率来说，这个函数的效率要高得多。它可在一定程度上提升性能）
参数表
参数类型及说明
hdcLong，要在其中绘图的设备场景
qrcRECT，要为其描绘边框的矩形
edgeLong，带有前缀 BDR_的两个常数的组合。一个指定内部边框是上凸还是下凹；另一个则指定外部边框。有时能换用带 EDGE_前缀的常数。
grfFlagsLong，带有 BF_前缀的常数的组合
注解
由于这是一个 GDI 函数，所以矩形坐标是逻辑坐标

Top
DrawEscape
DrawEscape

VB 声明
Declare Function DrawEscape Lib "gdi32" Alias "DrawEscape" (ByVal hdc As Long, ByVal nEscape As Long, ByVal cbInput As Long, ByVal lpszInData As String) As Long
说明

换码（Escape）函数将数据直接发至显示设备驱动程序（在 vb 里使用：能够使用。但由于 Escape 对设备有较强的依赖性，所以除非万不得以，尽量不要用它）
返回值
Long，非零表示成功，零表示错误（QUERYESCSUPPORT 换码例外；返回 TRUE 表示支持特定的换码；否则返回零）
参数表
参数类型及说明
hdcLong，显示设备的设备场景
nEscapeLong，指定欲执行的换码的一个常数
cbInputLong，输入缓冲区的长度
lpszInDataString，输入字串或缓冲区
注解
可用 QUERYESCSUPPORT 换码判断一个特定的换码是否得到当前显示驱动程序的支持

Top
DrawFocusRect
DrawFocusRect

VB 声明
Declare Function DrawFocusRect Lib "user32" Alias "DrawFocusRect" (ByVal hdc As Long, lpRect As RECT) As Long
说明
画一个焦点矩形。这个矩形是在标志焦点的样式中通过异或运算完成的（焦点通常用一个点线表示）。如用同样的参数再次调用这个函数，就表示删除焦点矩形
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
lpRectRECT，要在逻辑坐标中描绘的矩形

Top
DrawFrameControl
DrawFrameControl

VB 声明
Declare Function DrawFrameControl Lib "user32" Alias "DrawFrameControl" (ByVal hDC As Long, lpRect As RECT, ByVal un1 As Long, ByVal un2 As Long) As Long
说明
这个函数用于描绘一个标准控件。例如，可描绘一个按钮或滚动条的帧（原文：This function draws a standard control. For example, you can draw the frame of a button or scrollbar.）
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表

参数类型及说明
hDCLong，要在其中作画的设备场景
lpRectRECT，指定帧的位置及大小的一个矩形
un1Long，指定帧类型的一个常数。这些常数包括 DFC_BUTTON，DFC_CAPTION，DFC_MENU，以及 DFC_SCROLL
un2Long，一个常数，指定欲描绘的帧的状态。由带有前缀 DFCS_的一个常数构成

Top
DrawState
DrawState

VB 声明
Declare Function DrawState Lib "user32" Alias "DrawStateA" (ByVal hDC As Long, ByVal hBrush As Long, ByVal lpDrawStateProc As Long, ByVal lParam As Long, ByVal wParam As Long, ByVal n1 As Long, ByVal n2 As Long, ByVal n3 As Long, ByVal n4 As Long, ByVal un As Long) As Long
说明
这个函数可为一幅图象或绘图操作应用各式各样的效果
返回值
Long，TRUE（非零）表示成功，FALSE 表示失败
参数表
参数类型及说明
hDCLong，要在其中绘图的设备场景
hBrushLong，如状态为 DSS_MONO（在 un 参数中设定），则指定一个刷子句柄
lpDrawStateProcLong，指向一个函数地址的指针。如图象类型为 DST_COMPLEX，必须设置这个参数。对于 DST_TEXT，则可设可不设
lParamLong，由图象的类型决定
wParamLong，由图象的类型决定
n1Long，图象的水平位置
n2Long，图象的垂直位置
n3Long，图象的宽度。如图象类型为 DST_COMPLEX，必须设置这个参数。而对于其他类型，则可以设为零。如为零，表示该参数在图象的基础上计算
n4Long，图象的高度。如图象类型为 DST_COMPLEX，必须设置这个参数。而对于其他类型，则可以设为零。如为零，表示该参数在图象的基础上计算
unLong，图象类型和状态的一个组合。参见下表
图象类型
DST_BITMAPlParam 中的句柄
DST_COMPLEX 绘图在由 lpDrawStateProc 参数指定的回调函数期间执行。lParam 和 wParam 会传递给回调事件
DST_ICONlParam 包括图标句柄
DST_TEXTlParam 代表文字的地址（可使用一个字串别名），wParam 代表字串的长度
DST_PREFIXTEXT 与 DST_TEXT 类似，只是 & 字符指出为下各字符加上下划线
图象状态常数
DSS_NORMAL 普通图象

DSS_UNION 图象进行抖动处理
DSS_DISABLED 图象具有浮雕效果
DSS_MONO 用 hBrush 描绘图象
DSS_RIGHT 手册未正式说明——经实验证明没有什么作用（原文：Undocumented-experimentation seems to show no effect.）
注解
windows95 用它获得我们应用于图象的一些视觉效果；例如，可使位图或其他图象在视觉上进入禁用或抖动状态。对于位图和图标，它在描绘位图或图标的时候应用一种效果。对于文本，既可让函数画出文本，也可在一个回调函数中执行自己的绘图操作。对于复杂的（用户自定义）图象，则必须用一个回调函数。在回调函数执行过程中，用自己的代码将自己希望的任何东西画入设备场景。在这之后，利用 DrawState 函数应用希望的效果

Top
Ellipse
Ellipse

VB 声明
Declare Function Ellipse Lib "gdi32" Alias "Ellipse" (ByVal hdc As Long, ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long) As Long
说明
描绘一个椭圆，由指定的矩形围绕。椭圆用当前选择的画笔描绘，并用当前选择的刷子填充
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
X1，Y1Long，约束矩形采用逻辑坐标的左上角位置
X2，Y2Long，约束矩形采用逻辑坐标的右下角位置

Top
EndPath
EndPath

VB 声明
Declare Function EndPath Lib "gdi32" Alias "EndPath" (ByVal hdc As Long) As Long
说明
停止定义一个路径。如执行成功，BeginPath 函数调用和这个函数之间发生的所有绘图操作都会正式成为指定设备场景的路径
返回值
Long，非零表示成功，零表示失败。会将 GetLastError 设置为下述值之一：ERROR_CAN_NOT_COMPLETE 或 ERROR_INVALID_PARAMETER
参数表
参数类型及说明
hdcLong，设备场景

Top
EnumEnhMetaFile
EnumEnhMetaFile

VB 声明
Declare Function EnumEnhMetaFile Lib "gdi32" Alias "EnumEnhMetaFile" (ByVal hdc As Long, ByVal hemf As Long, ByVal lpEnhMetaFunc As Long, lpData As Any, lpRect As RECT) As Long
说明
针对一个增强型图元文件，列举其中单独的图元文件记录。每条图元文件记录都由单个 GDI 命令组成。可伴随 PlayEnhMetaFileRecord 函数使用，选择性的回放图元文件的一部分
返回值
Long，如图元文件的所有记录都列举出来，就返回 TRUE（非零）；否则返回零
参数表
参数类型及说明
hdcLong，用于输出的设备场景的句柄。只有在用回调函数进行绘图操作的时候，才要求设置这个参数
hemfLong，欲列举的一个增强型图元文件的句柄
lpEnhMetaFuncLong，指向为每个图元文件调用的一个函数的指针
lpData 用户自定义的值
lpRectRECT，定义图元文件边框的一个矩形

Top
EnumMetaFile
EnumMetaFile

VB 声明
Declare Function EnumMetaFile Lib "gdi32" Alias "EnumMetaFile" (ByVal hDC As Long, ByVal hMetafile As Long, ByVal lpMFEnumProc As Long, ByVal lParam As Long) As Long
说明
为一个标准的 windows 图元文件枚举单独的图元文件记录。每条图元文件记录都包含了单个 GDI 绘图命令。可与 PlayMetaFileRecord 函数联合使用，选择性的回放图元文件的某一部分
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hDCLong，用于输出的设备场景的句柄。只有在用回调函数进行绘图操作的时候，才要求设置这个参数
hMetafileLong，欲列举的一个标准图元文件的句柄
lpMFEnumProcLong，指向为每个图元文件调用的一个函数的指针
lParamLong，用户自定义的值

Top

EnumObjects
EnumObjects

VB 声明
Declare Function EnumObjects Lib "gdi32" Alias "EnumObjects" (ByVal hDC As Long, ByVal n As Long, ByVal lpGOBJEnumProc As Long, lpVoid As Any) As Long
说明
枚举可随同指定设备场景使用的画笔和刷子
返回值
Long，如函数要枚举的对象太多，就返回-1。否则由用户自己定义
参数表
参数类型及说明
hDCLong，设备场景的句柄
nLong，欲枚举的对象的类型。请查找带 OBJ_前缀的常数，这样可得到一个对象列表。win32 手册建议只使用 OBJ_PEN 和 OBJ_BRUSH 两个常数
lpGOBJEnumProcLong，指向为每个 GDI 对象调用的指针
lpVoid 枚举过程中传递给回调函数的值

Top
ExtCreatePen
ExtCreatePen

VB 声明
Declare Function ExtCreatePen Lib "gdi32" Alias "ExtCreatePen" (ByVal dwPenStyle As Long, ByVal dwWidth As Long, lplb As LOGBRUSH, ByVal dwStyleCount As Long, lpStyle As Long) As Long
说明
创建一个扩展画笔（装饰或几何）
返回值
Long，如执行成功，返回一个指向扩展画笔的句柄。零表示执行出错。一旦不再需要，记得用 DeleteObject 将画笔删除
参数表
参数类型及说明
dwPenStyleLong，画笔样式来自下述常数组的任何一个常数的组合（OR 运算）：
PS_COSMETIC or PS_GEOMETRIC 画笔的类型
PS_ALTERNATE, PS_SOLID, PS_DASH, PS_DOT, PS_DASHDOT, PS_DASHDOTDOT, PS_NULL, PS_USERSTYLE, PS_INSIDEFRAME 画笔的样式
PS_ENDCAP_???画笔的笔尖
PS_JOIN_???在图形中连接线段或在路径中连接直线的方式
dwWidthLong，指定线宽。几何画笔的线宽肯定是 1
lplbLOGBRUSH，lbColor 代表画笔颜色。对于装饰画笔，lbStyle 为 PS_SOLID；对于几何画笔，lbStyle 则代表实际的样式。针对几何画笔，必须设置其他所有字体
dwStyleCountLong，如指定了 PS_USERSTYLE，则代表 lpStyle 数组中的条目数量
lpStyleLong，指定 PS_USERSTYLE 的“线段/空白”对（原文：Line/space pairs for

PS_USERSTYLE）

Top
ExtFloodFill
ExtFloodFill

VB 声明
Declare Function ExtFloodFill Lib "gdi32" Alias "ExtFloodFill" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long, ByVal crColor As Long, ByVal wFillType As Long) As Long
说明
在指定的设备场景里，用当前选择的刷子填充一个区域
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
x,yLong，开始填充的一个点，采用逻辑坐标表示
crColorLong，要使用的边界颜色
wFillTypeLong，欲执行的填充类型，由下述任何一个常数决定
FLOODFILLBORDER 等同于 FloodFill 函数的功能
FLOODFILLSURFACE 从指定的点向外填充，只到找到了 crColor 颜色（在边框采用了多种颜色时使用）
注解
如指定了 FLOODFILLBORDER，那么 x,y 点绝对不能为 crColor 颜色。如指定了 FLOODFILLSURFACE，那么 x,y 点必须是 crColor 颜色。这个函数只能在光栅设备中使用。可用 GetDeviceCaps 函数判断设备是否支持这个函数
提示
一旦指定了 FLOODFILLBORDER，务必保证初始点的颜色没有 crColor。如果使用的是 FLOODFILLSURFACE，务必保证初始点有颜色 crColor（这是函数执行失败最常见的两个原因）。注意保证初始点位于剪切区内

Top
FillPath
FillPath

VB 声明
Declare Function FillPath Lib "gdi32" Alias "FillPath" (ByVal hdc As Long) As Long
说明
关闭路径中任何打开的图形，并用当前刷子填充
返回值
Long，非零表示成功，零表示失败。会将 GetLastError 设置为下述值之一：ERROR_CAN_NOT_COMPLETE，ERROR_INVALID_PARAMETER，ERROR_NOT_ENOUGH_MEMORY
参数表

hdcLong，欲在其中操作的设备场景
注解
函数执行完毕后，选定的路径会自行清除

Top
FillRect
FillRect

VB 声明
Declare Function FillRect Lib "user32" Alias "FillRect" (ByVal hdc As Long, lpRect As RECT, ByVal hBrush As Long) As Long
说明
用指定的刷子填充一个矩形
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
lpRectRECT，对填充区域进行描述的一个矩形，采用逻辑坐标
hBrushLong，欲使用的刷子的句柄
注解
矩形的右边和底边不会描绘

Top
FlattenPath
FlattenPath

VB 声明
Declare Function FlattenPath Lib "gdi32" Alias "FlattenPath" (ByVal hdc As Long) As Long
说明
将一个路径中的所有曲线都转换成线段
返回值
Long，非零表示成功，零表示失败。会将 GetLastError 设置为下述值之一：ERROR_CAN_NOT_COMPLETE，ERROR_INVALID_PARAMETER，ERROR_NOT_ENOUGH_MEMORY
参数表
参数类型及说明
hdcLong，包含了路径的设备场景

Top
FloodFill
FloodFill

VB 声明

Declare Function FloodFill Lib "gdi32" Alias "FloodFill" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long, ByVal crColor As Long) As Long
说明
用当前选定的刷子在指定的设备场景中填充一个区域。区域是由颜色 crColor 定义的
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
x,yLong，开始填充的那个点，用逻辑坐标表示
crColorLong，欲使用的边界颜色。由这个颜色包围的表面会被填充
注解
点 x,y 绝对不能有颜色 crColor，而且必须在剪切区域内。这个函数只对光栅设备有效，请参考 ExtFloodFill 的注解

Top
FrameRect
FrameRect

VB 声明
Declare Function FrameRect Lib "user32" Alias "FrameRect" (ByVal hdc As Long, lpRect As RECT, ByVal hBrush As Long) As Long
说明
用指定的刷子围绕一个矩形画一个边框（组成一个帧），边框的宽度是一个逻辑单位
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
lpRectRECT，对要描绘的边框进行描述的一个矩形。这等效于将画笔设成一个单位的宽度，然后用矩形函数画出一个矩形
hBrushLong，欲使用的刷子的句柄
注解
lpRect 的顶部边距值必须小于底部边距值。lpRect 的左侧边距值必须小于右侧边距值

Top
GdiComment
GdiComment

VB 声明
Declare Function GdiComment Lib "gdi32" Alias "GdiComment" (ByVal hdc As Long, ByVal cbSize As Long, lpData As Byte) As Long
说明
为指定的增强型图元文件设备场景添加一条注释信息

返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hdcLong，目标增强型图元文件设备场景的句柄
cbSizeLong，欲嵌入图元文件的数据长度
lpDataByte，一个注释结构或一个缓冲区的 Long 内存地址，其中包含了欲添加的注释正文
注解
尽管可在图元文件中嵌入任何专用或私有（Private）信息，但只有几种全局数据格式能够嵌入。如将缓冲区看作一个 32 位 Long 型值的数组，则全局注释的值就是下面这个样子：
第一个条目是常数 GDICOMMENT_IDENTIFIER
第二个条目如下所示：
首先是一个 GDICOMMENT_WINDOWS_METAFILE——在增强型图元文件中嵌入一个标准图元文件。它的后面跟随下述值之一：
□ 标准图元文件的版本号
□ 一个校验和（checksum）值：所有图元文件数据的总和——包括这个值——必须是零
□ 零
□ 后面跟随的窗口图元文件的大小
GDICOMMENT_BEGINGROUP——标志一组绘图命令在增强型图元文件在中的起始处。它的后面跟随：
□ 四个 Long 值。定义一个 RECT 结构。结构中包含了绘图命令的约束矩形
□ 可选的 Unicode 字串的长度。字串中包含对命令组的说明文字。如不想提供说明，可设为零
GDICOMMENT_ENDGROUP——标志增强型图元文件中的一组绘图命令的结尾
GDICOMMENT_MULTIFORMATS——以不同的格式嵌入一幅处理过的图象。例如，可利用这个注释在一个增强型图元文件中嵌入一个封装式 PostScript 图象。回放这条记录的时候，windows 会重画它能描绘的第一组格式。它的后面跟随：
□ 四个 Long 值。定义一个 RECT 结构。结构中包含了绘图命令的约束矩形
□ 包括在注释中的格式数量
□ 一系列 EMRFORMAT 结构，每种格式使用一个

Top
GdiFlush
GdiFlush

VB 声明
Declare Function GdiFlush Lib "gdi32" Alias "GdiFlush" () As Long
说明
执行任何未决的绘图操作
返回值
Long，如所有未决的绘图操作都成功完成，就返回 TRUE（非零）。如任何一个操作失败，就返回零值
注解
通过成批合并绘图操作命令，win32 图形子系统（GDI）可改善绘图的性能。如调用一系列

绘图命令，他们都返回布尔值（TRUE 表示成功，零表示失败），就可将他们置于一个内部 GDI 队列里。此时，函数可以立即返回。随后，GDI 子系统会执行这些待决的绘图命令。可考虑一种最常见的情况。在这种情况下，系统安装了一块显示卡。卡上自带图形处理器或加速器。画图的时候，GDI 只需将图形命令简单的发送给显示卡，另其完成实际的操作。如果必须等待每个绘图命令都完成并返回，系统和应用程序的性能就会受到显示卡绘图速度的极大限制。所以在这个时候，GDI 将绘图命令置于一个名为“批”（Batch）的队列里。这样一来，系统和应用程序就能继续运行，同时仍然让显示卡进行绘图操作
GdiFlush 命令指示应用程序进入等待状态，直到所有待决的绘图操作完成为止。如执行的是一个特殊的 GDI 绘图命令，它不会返回一个布尔值，那么也会面临这种情况。例如，GetPixel 函数需要读取一个像素值。但除非所有待决的绘图完成，否则该函数不能可靠的完成工作

Top
GdiGetBatchLimit
GdiGetBatchLimit

VB 声明
Declare Function GdiGetBatchLimit Lib "gdi32" Alias "GdiGetBatchLimit" () As Long
说明
判断有多少个 GDI 绘图命令位于队列中
返回值
Long，待决绘图命令的最大数量，零表示出错。会设置 GetLastError
注解
参考对 GdiFlush 的注解

Top
GdiSetBatchLimit
GdiSetBatchLimit

VB 声明
Declare Function GdiSetBatchLimit Lib "gdi32" Alias "GdiSetBatchLimit" (ByVal dwLimit As Long) As Long
说明
指定有多少个 GDI 绘图命令能够进入队列
返回值
Long，如执行成功，返回前一个限制；零表示出错。会设置 GetLastError
参数表
参数类型及说明
dwLimitLong，可排队的绘图操作最大数量；0 意味着恢复默认值；1 表示禁止绘图命令排队
注解
参考对 GdiFlush 的注解

Top
GetArcDirection

GetArcDirection

VB 声明
Declare Function GetArcDirection Lib "gdi32" Alias "GetArcDirection" (ByVal hdc As Long) As Long
说明
画圆弧的时候，判断当前采用的绘图方向
返回值
Long，常数 AD_COUNTERCLOCKWISE（逆时针）或 AD_CLOCKWISE（顺时针），零表示出错
参数表
参数类型及说明
hdcLong，要查询的设备场景

Top
GetBkColor
GetBkColor

VB 声明
Declare Function GetBkColor Lib "gdi32" Alias "GetBkColor" (ByVal hdc As Long) As Long
说明
取得指定设备场景当前的背景颜色
返回值
Long，当前背景色的 RGB 颜色值
参数表
参数类型及说明
hdcLong，欲查询背景颜色的一个设备场景

Top
GetBkMode
GetBkMode

VB 声明
Declare Function GetBkMode Lib "gdi32" Alias "GetBkMode" (ByVal hdc As Long) As Long
说明
针对指定的设备场景，取得当前的背景填充模式
返回值
Long，下述常数之一，零意味着出错
OPAQUE 将文本、阴影刷以及虚线画笔线段的背景设为当前的背景色
TRANSPARENT 不修改文本、阴影刷以及虚线画笔线段的背景
参数表
参数类型及说明
hdcLong，设备场景的句柄

Top
GetBrushOrgEx
GetBrushOrgEx

VB 声明
Declare Function GetBrushOrgEx Lib "gdi32" Alias "GetBrushOrgEx" (ByVal hDC As Long, lpPoint As POINTAPI) As Long
说明
判断指定设备场景中当前选定刷子起点
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hDCLong，设备场景的句柄
lpPointPOINTAPI，这个结构用来装载当前刷子的起点

Top
GetCurrentObject
GetCurrentObject

VB 声明
Declare Function GetCurrentObject Lib "gdi32" Alias "GetCurrentObject" (ByVal hdc As Long, ByVal uObjectType As Long) As Long
说明
用于获得指定类型的当前选定对象
返回值
Long，当前选定对象的句柄，零表示错误
参数表
参数类型及说明
hdcLong，欲查询的设备场景
uObjectTypeLong，对象类型。可以是下述常数之一：OBJ_PEN，OBJ_BRUSH，OBJ_PAL，OBJ_FONT，或 OBJ_BITMAP

Top
GetCurrentPositionEx
GetCurrentPositionEx

VB 声明
Declare Function GetCurrentPositionEx Lib "gdi32" Alias "GetCurrentPositionEx" (ByVal hdc As Long, lpPoint As POINTAPI) As Long
说明
在指定的设备场景中取得当前的画笔位置
返回值
Long，非零表示成功，零表示失败

参数表
参数类型及说明
hdcLong，设备场景的句柄
lpPointPOINTAPI，用于装载当前位置

Top
GetEnhMetaFile
GetEnhMetaFile

VB 声明
Declare Function GetEnhMetaFile Lib "gdi32" Alias "GetEnhMetaFileA" (ByVal lpszMetaFile As String) As Long
说明
取得磁盘文件中包含的一个增强型图元文件的图元文件句柄（原文：Retrieves a metafile handle to an enhanced metafile contained in a disk file）
返回值
Long，指向图元文件的句柄。零表示出错
参数表
参数类型及说明
lpszMetaFileString，包含了增强型图元文件的一个磁盘文件的名字

Top
GetEnhMetaFileBits
GetEnhMetaFileBits

VB 声明
Declare Function GetEnhMetaFileBits Lib "gdi32" Alias "GetEnhMetaFileBits" (ByVal hemf As Long, ByVal cbBuffer As Long, lpbBuffer As Byte) As Long
说明
将指定的增强型图元文件复制到一个内存缓冲区里。这个函数可用于取得一个图元文件的原始数据，以便将其保存到磁盘文件中
返回值
Long，如 lpbBuffer 为零，则必须设置缓冲区的长度（用 ByVal As Long 在这种情况下传递一个 NULL 参数）。如执行成功，返回缓冲区实际载入的字节数；零表示执行失败
参数表
参数类型及说明
hemfLong，指向一个增强型图元文件的句柄
cbBufferLong，lpbBuffer 缓冲区的长度
lpbBufferByte，长度为 cbBuffer 的一个缓冲区的头一个字节。可用一个别名，将这个参数定义成 ByVal As Long，以便传递一个内存地址
注解
调用了这个函数后，图元文件句柄 hemf 仍然有效

Top

GetEnhMetaFileDescription
GetEnhMetaFileDescription

VB 声明
Declare Function GetEnhMetaFileDescription Lib "gdi32" Alias "GetEnhMetaFileDescriptionA" (ByVal hemf As Long, ByVal cchBuffer As Long, ByVal lpszDescription As String) As Long
说明
返回对一个增强型图元文件的说明
返回值
Long，如 lpszDescription 为 NULL（使用 vbNullString），则必须设置缓冲区的长度。如函数执行成功，返回实际载入缓冲区的字节数；如不存在说明信息则返回零
参数表
参数类型及说明
hemfLong，目标增强型图元文件的句柄
cchBufferLong，lpszDescription 缓冲区的长度
lpszDescriptionString，指定一个预先初始化好的字串缓冲区，准备随同图元文件说明载入。参考 CreateEnhMetaFile 函数，了解增强型图元文件说明字串的具体格式

Top
GetEnhMetaFileHeader
GetEnhMetaFileHeader

VB 声明
Declare Function GetEnhMetaFileHeader Lib "gdi32" Alias "GetEnhMetaFileHeader" (ByVal hemf As Long, ByVal cbBuffer As Long, lpemh As ENHMETAHEADER) As Long
说明
取得增强型图元文件的图元文件头
返回值
Long，若 lpemh 为零（用 ByVal As Long 在这种情况下传递一个 NULL 参数）则必须设置缓冲区的长度。如执行成功，返回缓冲区中实际载入的字节数；零表示失败
参数表
参数类型及说明
hemfLong，指向一个增强型图元文件的句柄
cbBufferLong，ENHMETAHEADER 结构的大小
lpemhENHMETAHEADER

Top
GetEnhMetaFilePaletteEntries
GetEnhMetaFilePaletteEntries

VB 声明
Declare Function GetEnhMetaFilePaletteEntries Lib "gdi32" Alias "GetEnhMetaFilePaletteEntries" (ByVal hemf As Long, ByVal cEntries As Long, lppe As PALETTEENTRY) As Long

说明
取得增强型图元文件的全部或部分调色板
返回值
Long，如 lppe 为零（用 ByVal As Long 在这种情况下传递一个 NULL 参数）则必须设置缓冲区的长度。如执行成功，返回缓冲区实际载入的条目数量；如图元文件中不存在调色板就返回一个零值。如返回 GDI_ERROR，表示函数执行出错
参数表
参数类型及说明
hemfLong，增强型图元文件的句柄
cEntriesLong，欲取回的条目数量
lppePALETTEENTRY，一个 PALETTEENTRY 结构的数组，用于装载增强型图元文件的调色板条目。注意至少要包括 cEntries 个结构

Top
GetMetaFile
GetMetaFile

VB 声明
Declare Function GetMetaFile Lib "gdi32" Alias "GetMetaFileA" (ByVal lpFileName As String) As Long
说明
取得包含在一个磁盘文件中的图元文件的图元文件句柄
返回值
Long，指向已载入的图元文件的一个句柄，零表示错误
参数表
参数类型及说明
lpFileNameString，包含了一个图元文件的磁盘文件的名称
注解
这个函数在 vb 里没有什么用。因为 windows 图元文件通常保存为可放置的图元文件格式，而该函数不能识别这种格式

Top
GetMetaFileBitsEx
GetMetaFileBitsEx

VB 声明
Declare Function GetMetaFileBitsEx Lib "gdi32" Alias "GetMetaFileBitsEx" (ByVal hMF As Long, ByVal nSize As Long, lpvData As Any) As Long
说明
将指定的图元文件复制到一个内存缓冲区。这个函数可用于取得一个图元文件的原始数据，以便转存到磁盘文件
返回值
Long，如 lpvData 为零（在这种情况下用 ByVal As Long 传递一个 NULL 参数），那么必须指定缓冲区的长度。如执行成功，返回实际载入缓冲区的字节数，零表示出错

参数表
参数类型及说明
hMFLong，标准 windows 图元文件的句柄
nSizeLong，lpvData 缓冲区的长度
lpvData 任何类型，一个字节数组中的第一个条目，该数组用于装载图元文件数据。用 ByVal As Long 可指定空值或缓冲区的内存地址

Top
GetMiterLimit
GetMiterLimit

VB 声明
Declare Function GetMiterLimit& Lib "gdi32" (ByVal hdc As Long, peLimit As Single)
说明
取得设备场景的斜率限制（Miter）设置——斜率限制是指斜角长度与线宽间的比率
返回值
Long，
参数表
参数类型及说明
hdcLong，设备场景的句柄
peLimitSingle，随同当前斜率设置载入一个 Single 值
注解
开头的声明来自于我的资料，而在 vb 的 api 文本查看器里的声明如下：Declare Function GetMiterLimit Lib "gdi32" Alias "GetMiterLimit" (ByVal hdc As Long, peLimit As Double) As Long，参数 peLimit 的类型不同，不知是谁错了。

Top
GetNearestColor
GetNearestColor

VB 声明
Declare Function GetNearestColor Lib "gdi32" Alias "GetNearestColor" (ByVal hdc As Long, ByVal crColor As Long) As Long
说明
根据设备的显示能力，取得与指定颜色最接近的一种纯色
返回值
Long，取得与指定颜色最接近的一种颜色，这种颜色可由设备场景实际显示出来（给定当前系统调色板）。如返回 CLR_INVALID，表示函数执行出错。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
crColorLong，欲测试的 RGB 颜色

Top

GetObjectAPI
GetObjectAPI

VB 声明
Declare Function GetObjectAPI& Lib "gdi32" Alias "GetObjectA" (ByVal hObject As Long, ByVal nCount As Long, lpObject As Any)
说明
取得对指定对象进行说明的一个结构。windows 手册建议用 GetObject 这个名字来引用该函数。GetObjectAPI 在 vb 中用于避免与 GetObject 关键字混淆
返回值
Long，如 lpObject 设为零（用 ByVal As Long 在这种情况下传递一个 NULL 参数），则必须设置缓冲区的长度。如执行成功，返回载入结构内部的实际字节数；如失败，返回零值
参数表
参数类型及说明
hObjectLong，画笔、刷子、字体、位图或调色板等对象的句柄
nCountLong，欲取回的字节数。通常是由 lpObject 定义的那个结构的长度
lpObject 任何类型，用于容纳对象数据的结构。针对画笔，通常是一个 LOGPEN 结构；针对扩展画笔，通常是 EXTLOGPEN；针对字体是 LOGBRUSH；针对位图是 BITMAP；针对 DIBSection 位图是 DIBSECTION；针对调色板，应指向一个整型变量，代表调色板中的条目数量

Top
GetObjectType
GetObjectType

VB 声明
Declare Function GetObjectType Lib "gdi32" Alias "GetObjectType" (ByVal hgdiobj As Long) As Long
说明
判断由指定句柄引用的 GDI 对象的类型
返回值
Long，带有 OBJ_前缀的一个常数，标志着一种特定的对象。零表示错误
参数表
参数类型及说明
hgdiobjLong，一个 GDI 对象句柄

Top
GetPath
GetPath

VB 声明
Declare Function GetPath Lib "gdi32" Alias "GetPath" (ByVal hdc As Long, lpPoint As POINTAPI, lpTypes As Byte, ByVal nSize As Long) As Long
说明

取得对当前路径进行定义的一系列数据
返回值
Long，载入数组的点数（如 nSize 设为零，则返回要求的条目数量）。如数组空间不够大，不足以容下所有的点，就返回 -1。会将 GetLastError 设置为下述值之一：ERROR_CAN_NOT_COMPLETE，ERROR_INVALID_PARAMETER，ERROR_BUFFER_OVERFLOW
参数表
参数类型及说明
hdcLong，包含了路径的设备场景
lpPointPOINTAPI，一个 POINTAPI 结构数组中的第一个元素。这个数组为路径中的每个段（segment）都要载入坐标数据。具体的信息是采用逻辑坐标提供的
lpTypesByte，一个字节数组中的第一个元素；这个数组定义了与每个坐标对应的操作类型。其中包括：
PT_MOVETO 坐标是一个新子路径的起始处
PT_LINETO 坐标是来自前一个坐标的一条线的终点
PT_BEZIERTO 肯定以三点一组的形式出现。头两个点是控制点，第三个是贝塞尔（Bezier）曲线的终点。PT_LINETO 和 PT_BEZIERTO 也许能与 PT_CLOSEFIGURE 联合使用。在这种情况下，它代表一幅图象的最后一个点。将这个点与子路径的第一个连接起来后，路径就会封闭
nSizeLong，lpPoint 和 lpTypes 数组的大小。如设为零，表示取得要求的数组大小
注解
尽管路径信息是在设备坐标的内部保存的，这个函数的所有坐标都是用逻辑坐标返回的。具体坐标取决于当前的坐标系统及转换设置。可用 FlattenPath 函数强迫路径中的所有点都成为 PT_MOVETO 和 PT_LINETO 类型

Top
GetPixel
GetPixel

VB 声明
Declare Function GetPixel Lib "gdi32" Alias "GetPixel" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long) As Long
说明
在指定的设备场景中取得一个像素的 RGB 值
返回值
Long，指定点的 RGB 颜色。如指定的点位于设备场景的剪切区之外，则返回 CLR_INVALID
参数表
参数类型及说明
hdcLong，一个设备场景的句柄
x,yLong，逻辑坐标中要检查的点
注解
用 GetDeviceCaps 判断设备是否支持本函数

Top

GetPolyFillMode
GetPolyFillMode

VB 声明
Declare Function GetPolyFillMode Lib "gdi32" Alias "GetPolyFillMode" (ByVal hdc As Long) As Long
说明
针对指定的设备场景，获得多边形填充模式。关于填充模式见注解
返回值
Long，常数 ALTERNATE 或 WINDING。零表示失败
参数表
参数类型及说明
hdcLong，设备场景的句柄
注解
填充模式 1——ALTERNATE：为判断一个点是否位于填充区，windows 会从这个点到图形外部画一条假想的线。每与一条线相交，计数器就会增 1。如最后一个记数是奇数，则填充这个点；如果是偶数，则保留原样不变
填充模式 2——WINDING：为判断一个点是否位于填充区，windows 会从这个点到图形外部画一条假想的线。windows 会跟踪画出每个顶点（线段）的方向。这条假想的线每次穿过一个顶点时，而且顶点的 Y 方向为正，则减一个记数。如结果记数不是零，就表明该点位于填充区域

Top
GetROP2
GetROP2

VB 声明
Declare Function GetROP2 Lib "gdi32" Alias "GetROP2" (ByVal hdc As Long) As Long
说明
针对指定的设备场景，取得当前的绘图模式。这样可定义绘图操作如何与正在显示的图象合并起来
返回值
Long，请参考绘图模式常数表
参数表
参数类型及说明
hdcLong，设备场景的句柄
注解
这个函数只对光栅设备有效

绘图模式常数表
常数 DrawMode 像素值
R2_BLACKvbBlackness 黑色
R2_WHITEvbWhitness 白色
R2_NOPvbNop 不变

R2_NOTvbInvert 当前显示颜色的反色
R2_COPYPENvbCopyPen 画笔颜色
R2_NOTCOPYPENvbNotCopyPenR2_COPYPEN 的反色
R2_MERGEPENNOTvbMergePenNot 显示颜色的反色与画笔颜色进行 OR 运算
R2_MASKPENNOTvbMaskPenNot 显示颜色的反色与画笔颜色进行 AND 运算
R2_MERGENOTPENvbMergeNotPen 画笔颜色的反色与显示颜色进行 OR 运算
R2_MASKNOTPENvbMaskNotPen 画笔颜色的反色与显示颜色进行 AND 运算
R2_MERGEPENvbMergePen 画笔颜色与显示颜色进行 OR 运算
R2_NOTMERGEPENvbNotMergePenR2_MERGEPEN 的反色
R2_MASKPENvbMaskPen 显示颜色与画笔颜色进行 AND 运算
R2_NOTMASKPENvbNotMaskPenR2_MASKPEN 的反色
R2_XORPENvbXorPen 显示颜色与画笔颜色进行异或运算
R2_NOTXORPENvbNotXorPenR2_XORPEN 的反色

Top
GetStockObject
GetStockObject

VB 声明
Declare Function GetStockObject Lib "gdi32" Alias "GetStockObject" (ByVal nIndex As Long) As Long
说明
取得一个固有对象（Stock）。这是可由任何应用程序使用的 windows 标准对象之一
返回值
Long，指向指定对象的一个句柄。零表示出错
参数表
参数类型及说明
nIndexLong，下述表格中定义的任何常数之一
BLACK_BRUSH 黑色刷子 DKGRAY_BRUSH 黑灰色刷子
GRAY_BRUSH 灰色刷子 HOLLOW_BRUSH 凹刷子
LTGRAY_BRUSH 浅灰色刷子 NULL_BRUSH 空刷子
WHITE_BRUSH 白色刷子 BLACK_PEN 黑色画笔
NULL_PEN 空画笔 WHITE_PEN 白色画笔
ANSI_FIXED_FONT 采用 windows（ANSI）字符集的等宽字体 ANSI_VAR_FONT 采用 windows（ANSI）字符集的不等宽字体
DEVICE_DEFAULT_FONT 设备使用的默认字体（NT）DEFAULT_GUI_FONT 用户界面的默认字体，包括菜单和对话框字体（Windows 95）
OEM_FIXED_FONTOEM 字符集的固有字体 SYSTEM_FONT 屏幕系统字体。这是用于菜单、对话框等等的默认不等宽字体
SYSTEM_FIXED_FONT 屏幕系统字体。这是用于菜单、对话框等等的默认等宽字体（在 windows 3.0 之前使用）DEFAULT_PALETTE 默认调色板
注解
固有刷子的起点可能不会改变。不应用 DeleteObject 函数删除这些对象。不要对那些不具备 CS_HREDRAW 和 CS_VREDRAW 类样式的窗口使用 DK_GRAY_BRUSH，GRAY_BRUSH

和 LTGRAY_BRUSH 刷子

Top
GetSysColorBrush
GetSysColorBrush

VB 声明
Declare Function GetSysColorBrush Lib "user32" Alias "GetSysColorBrush" (ByVal nIndex As Long) As Long
说明
为任何一种标准系统颜色取得一个刷子
返回值
Long，针对一种系统颜色的一个固有刷子的句柄。零表示出错
参数表
参数类型及说明
nIndexLong，系统颜色索引，也即带有 COLOR_前缀的某个常数。参考 GetSysColor
注解
不要用 DeleteObject 函数删除这些刷子。它们是由系统拥有的固有对象。不要将这些刷子指定成一种窗口类的默认刷子

Top
GetWinMetaFileBits
GetWinMetaFileBits

VB 声明
Declare Function GetWinMetaFileBits Lib "gdi32" Alias "GetWinMetaFileBits" (ByVal hemf As Long, ByVal cbBuffer As Long, lpbBuffer As Byte, ByVal fnMapMode As Long, ByVal hdcRef As Long) As Long
说明
通过在一个缓冲区中填充用于标准图元文件的数据，将一个增强型图元文件转换成标准 windows 图元文件
返回值
Long，以字节数表示的图元文件长度。如 lpbBuffer 为 NULL（在这种情况下用一个别名指定 ByVal As Long，从而传递一个 NULL 值）——返回字节数组的长度。零表示出错（原文: The size in bytes of the metafile. If lpbBuffer is null (use an alias with the parameter specified ByVal as Long to pass null to this function)-returns the required size of the byte array. Zero on error.）
参数表
参数类型及说明
hemfLong，欲转换的增强型图元文件的句柄。函数调用完毕后，该句柄仍然保持有效
cbBufferLong，目标缓冲区的长度
lpbBufferByte，作为目标缓冲区使用的一个字节数组的第一个字节。这个数组的长度至少为 cbBuffer 个字节
fnMapModeLong，转换时采用的映射（对应）模型。通常用 MM_ANISOTROPIC 创建一个

可扩展的图元文件
hdcRefLong，一个参考设备场景，用于决定新图元文件采用的参考分辨率
注解
有些增强型图元文件命令没有对应的标准图元文件命令。这些命令会转换成最接近的命令，或者丢弃。结果生成的图元文件已指定了窗口的显示范围。窗口的起点是 0，0

Top
InvertRect
InvertRect

VB 声明
Declare Function InvertRect Lib "user32" Alias "InvertRect" (ByVal hdc As Long, lpRect As RECT) As Long
说明
通过反转每个像素的值，从而反转一个设备场景中指定的矩形
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
lpRectRECT，要反转的矩形，用逻辑坐标指定
注解
反转是一种光栅操作

Top
LineDDA
LineDDA

VB 声明
Declare Function LineDDA Lib "gdi32" Alias "LineDDA" (ByVal n1 As Long, ByVal n2 As Long, ByVal n3 As Long, ByVal n4 As Long, ByVal lpLineDDAProc As Long, ByVal lParam As Long) As Long
说明
枚举指定线段中的所有点
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
n1,n2Long，线段的 x,y 起点坐标
n3,n4Long，线段的 x,y 终点坐标
lpLineDDAProcLong，vb5 中的一个函数地址
lParamLong，枚举过程中传递给回调函数的用户自定义值
注解
通常用这个函数执行自定义的线段作图——例如，可将一条线中的其他每个像素都设成不同

的颜色。在 MM_TEXT 模式下，每个点都对应于设备中的一个像素——在这种模式下，也可用这个函数进行线段中的击中测试。线段中的最后一个点不会枚举出来

Top
LineTo
LineTo

VB 声明
Declare Function LineTo Lib "gdi32" Alias "LineTo" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long) As Long
说明
用当前画笔画一条线，从当前位置连到一个指定的点。这个函数调用完毕，当前位置变成 x,y 点
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hdcLong，设备场景的句柄
x,yLong，线段终点位置，采用逻辑坐标表示。这个点不会实际画出来；它不属于线段的一部分
注解
如重复调用这个函数和一个几何画笔，从而创建一系列线段，那么除非在一个路径的场景中调用，否则不会认为这些线段已结合到一起

Top
MoveToEx
MoveToEx

VB 声明
Declare Function MoveToEx Lib "gdi32" Alias "MoveToEx" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long, lpPoint As POINTAPI) As Long
说明
为指定的设备场景指定一个新的当前画笔位置。前一个位置保存在 lpPoint 中
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hdcLong，指向一个设备场景的句柄
x,yLong，采用逻辑坐标表示的新画笔位置
lpPointPOINTAPI，用于保存前一个画笔位置。可以为 NULL（将参数改为 ByVal As Long，以传递一个空参数）
注解
在一个路径分支中描绘的时候，这个函数会创建一个新的子路径

Top
PaintDesktop
PaintDesktop

VB 声明
Declare Function PaintDesktop Lib "user32" Alias "PaintDesktop" (ByVal hdc As Long) As Long
说明
在指定的设备场景中描绘桌面墙纸图案
返回值
Long，TRUE（非零）表示成功，否则返回零
参数表
参数类型及说明
hdcLong，要在其中填充的设备场景

Top
PathToRegion
PathToRegion

VB 声明
Declare Function PathToRegion Lib "gdi32" Alias "PathToRegion" (ByVal hdc As Long) As Long
说明
将当前选定的路径转换到一个区域里
返回值
Long，新区域的句柄。零表示错误。会将 GetLastError 设为下述值之一：ERROR_CAN_NOT_COMPLETE，ERROR_INVALID_PARAMETER，ERROR_NOT_ENOUGH_MEMORY
参数表
参数类型及说明
hdcLong，包含了欲转换的路径的设备场景
注解
函数执行完毕后，路径会自动清除

Top
Pie
Pie

VB 声明
Declare Function Pie Lib "gdi32" Alias "Pie" (ByVal hdc As Long, ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long, ByVal X3 As Long, ByVal Y3 As Long, ByVal X4 As Long, ByVal Y4 As Long) As Long
说明
象注解里那样画一个饼图，X1，Y1，X2，Y2 指定椭圆的一个约束矩形。从矩形的中心分别向 X3，Y3 和 X4，Y4 画一条线，这两条线与椭圆的交点定义了饼图占据椭圆的面积
返回值

Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，指定一个显示场景的句柄
X1,Y1Long，指定椭圆约束矩形的左上角位置
X2,Y2Long，指定椭圆约束矩形的右下角位置
X3,Y3Long，指定饼图的一个斜边
X4,Y4Long，指定饼图的另一个斜边
注解
在 win95 和 win16 中，约束矩形的宽度和高度必须在 3-32766 之间
参考 Arc 函数

Top
PlayEnhMetaFile
PlayEnhMetaFile

VB 声明
Declare Function PlayEnhMetaFile Lib "gdi32" Alias "PlayEnhMetaFile" (ByVal hdc As Long, ByVal hemf As Long, lpRect As RECT) As Long
说明
在指定的设备场景中画一个增强型图元文件。与标准图元文件不同，完成回放后，增强型图元文件会恢复设备场景以前的状态
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，目标设备场景
hemfLong，欲描绘的增强型图元文件的句柄
lpRectRECT，一个约束矩形，定义了在哪里描绘图元文件

Top
PlayEnhMetaFileRecord
PlayEnhMetaFileRecord

VB 声明
Declare Function PlayEnhMetaFileRecord Lib "gdi32" Alias "PlayEnhMetaFileRecord" (ByVal hdc As Long, lpHandletable As HANDLETABLE, lpEnhMetaRecord As ENHMETARECORD, ByVal nHandles As Long) As Long
说明
回放单独一条增强型图元文件记录。可与 EnumEnhMetaFile 函数联合使用，只回放选定的图元文件记录。这个函数的参数设置与那些由 EnumMetaFile 回调函数返回的值是相似的
返回值
Long，非零表示成功，零表示失败
参数表

参数类型及说明
hdcLong，欲在其中绘图的设备场景
lpHandletableHANDLETABLE，使用的句柄的一个数组（An array of handles used.）
lpEnhMetaRecordENHMETARECORD，一个特定的结构（或指向结构的指针），包含了增强型图元文件记录
nHandlesLong，句柄表中的句柄数量

Top
PlayMetaFile
PlayMetaFile

VB 声明
Declare Function PlayMetaFile Lib "gdi32" Alias "PlayMetaFile" (ByVal hdc As Long, ByVal hMF As Long) As Long
说明
在指定的设备场景中回放一个图元文件。图元文件中记录的 GDI 操作会针对设备场景而执行
返回值
Long，
参数表
参数类型及说明
hdcLong，要在其中回放图元文件的一个设备场景的句柄
hMFLong，欲回放的一个图元文件的句柄
注解
在 vb 里，图元文件有能力改变一个设备场景的对象及映射模式。注意在调用这个函数之前，必须保存一个 vb 窗体或图片控件设备场景的状态

Top
PlayMetaFileRecord
PlayMetaFileRecord

VB 声明
Declare Function PlayMetaFileRecord Lib "gdi32" Alias "PlayMetaFileRecord" (ByVal hdc As Long, lpHandletable As HANDLETABLE, lpMetaRecord As METARECORD, ByVal nHandles As Long) As Long
说明
回放来自图元文件的单条记录（每条记录都包含了单个 GDI 绘图命令）。可与 EnumMetaFile 函数联合使用，只回放那些选定的图元文件记录。这个函数的参数与那些由 EnumMetaFile 回调函数返回的值相似
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hdcLong，欲在其中回放图元文件记录 GDI 命令的一个设备场景的句柄

lpHandletableHANDLETABLE，一个整数数组的第一个条目，该数组用于容纳图元文件使用的 GDI 对象的句柄
lpMetaRecordMETARECORD，指定单条图元文件记录
nHandlesLong，图元文件句柄表格中的句柄数目

Top
PolyBezier
PolyBezier, PolyBezierTo

VB 声明
Declare Function PolyBezier& Lib "gdi32" (ByVal hdc As Long, lppt As POINTAPI, ByVal cPoints As Long)
Declare Function PolyBezierTo& Lib "gdi32" (ByVal hdc As Long, lppt As POINTAPI, ByVal cCount As Long)
说明
描绘一条或多条贝塞尔（Bezier）曲线。PolyBezierTo 用于将当前画笔位置设为前一条曲线的终点
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hdcLong，要在其中绘图的设备场景
lpptPOINTAPI，指定一个 POINTAPI 结构数组。其中的第一个结构指定了起点。剩下的点三个一组——包括两个控件点和一个终点
原文：An array of POINTAPI structures. The first structure specifies the starting point. The remaining points are in groups of three, consisting of two control points and an end point.
cPointsLong，lppt 数组的总点数

Top
PolyDraw
PolyDraw

VB 声明
Declare Function PolyDraw Lib "gdi32" Alias "PolyDraw" (ByVal hdc As Long, lppt As POINTAPI, lpbTypes As Byte, ByVal cCount As Long) As Long
说明
描绘一条复杂的曲线，由线段及贝塞尔曲线组成
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hdcLong，用于绘图的设备场景
lpptPOINTAPI，POINTAPI 结构数组的第一个元素。这个数组用于为描绘的每一段都载入坐标数据。这些信息是用逻辑坐标提供的

lpbTypesByte，一个字节数组的第一个元素。这个数组定义了与每个坐标对应的操作类型。其中包括：
PT_MOVETO 坐标是一幅新打开图形的起点
PT_LINETO 坐标是来自前一个坐标的一条线的终点
PT_BEZIERTO 以三点一组的形式出现。头两个点是控制点，第三个是贝塞尔曲线的终点。
PT_LINETO 和 PT_BEZIERTO 也许能与 PT_CLOSEFIGURE 联合使用。在这种情况下，它代表一幅图形的最后一个点。将这个点与图形的第一个点连接起来后，图形就会封闭
cCountLong，lppt 和 lpbTypes 数组的大小，设为零表示取得需要的数组大小
原文：The size of the lpPoint and lpTypes array. Set to zero to retrieve the required array size.
注解
当前的画笔位置设为最后一条线段或 lppt 数组中的曲线的终点

Top
Polygon
Polygon

VB 声明
Declare Function Polygon Lib "gdi32" Alias "Polygon" (ByVal hdc As Long, lpPoint As POINTAPI, ByVal nCount As Long) As Long
说明
描绘一个多边形，由两点或三点的任意系列构成。windows 会将最后一个点与第一个点连接起来，从而封闭多边形。多边形的边框用当前选定的画笔描绘，多边形用当前选定的刷子填充
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，用于描绘的设备场景
lpPointPOINTAPI，在 nCount POINTAPI 结构数组中的第一个 POINTAPI 结构
nCountLong，多边形的总点数（顶点数）
注解
GetPolyFillMode 和 SetPolyFillMode 函数决定了如何在多边形内部填充

Top
Polyline
Polyline, PolyLineTo

VB 声明
Declare Function Polyline Lib "gdi32" Alias "Polyline" (ByVal hdc As Long, lpPoint As POINTAPI, ByVal nCount As Long) As Long
Declare Function PolylineTo Lib "gdi32" Alias "PolylineTo" (ByVal hdc As Long, lppt As POINTAPI, ByVal cCount As Long) As Long
说明
用当前画笔描绘一系列线段。使用 PolylineTo 函数时，当前位置会设为最后一条线段的终点。

它不会由 Polyline 函数改动
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hdcLong，要在其中绘图的设备场景
lpPointPOINTAPI，nCount POINTAPI 结构数组中的第一个 POINTAPI 结构
nCountLong，lpPoint 数组中的点数。会从第一个点到第二个点画一条线，以次类推

Top
PolyPolygon
PolyPolygon

VB 声明
Declare Function PolyPolygon Lib "gdi32" Alias "PolyPolygon" (ByVal hdc As Long, lpPoint As POINTAPI, lpPolyCounts As Long, ByVal nCount As Long) As Long
说明
用当前选定画笔描绘两个或多个多边形。根据由 SetPolyFillMode 函数指定的多边形填充模式，用当前选定的刷子填充它们。每个多边形都必须是封闭的
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，要在其中绘图的设备场景
lpPointPOINTAPI，nCount POINTAPI 结构数组中的第一个 POINTAPI 结构
lpPolyCountsLong，在 Long 值数组中的第一个条目。每个条目都包含了一定数量的点，用于构成一个封闭的多边形。lpPoint 数组由一系列封闭的多边形构成，每个在 lpPolyCounts 数组中都有一个条目
nCountLong，要描绘的多边形总数（就是 lpPolyCounts 数组的大小）。至少为 2

Top
PolyPolyline
PolyPolyline

VB 声明
Declare Function PolyPolyline Lib "gdi32" Alias "PolyPolyline" (ByVal hdc As Long, lppt As POINTAPI, lpdwPolyPoints As Long, ByVal cCount As Long) As Long
说明
用当前选定画笔描绘两个或多个多边形
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hdcLong，要在其中绘图的设备场景

lpPointPOINTAPI，nCount POINTAPI 结构数组中的第一个 POINTAPI 结构
lpdwPolyPointsLong，在 Long 值数组中的第一个条目。每个条目都包含了构成一个多边形的点数。lpPoint 数组由一系列多边形构成，每个在 lpdwPolyPoints 数组中都有一个条目
cCountLong，要描绘的多边形总数（就是 lpdwPolyPoints 数组的大小）
注解
这个函数几乎与 PolyPolygon 函数完全一致，只是多边形不会填充，也不需要封闭

Top
Rectangle
Rectangle

VB 声明
Declare Function Rectangle Lib "gdi32" Alias "Rectangle" (ByVal hdc As Long, ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long) As Long
说明
用当前选定的画笔描绘矩形，并用当前选定的刷子进行填充
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
X1,Y1Long，指定矩形左上角位置
X2,Y2Long，指定矩形右下角位置

Top
RoundRect
RoundRect

VB 声明
Declare Function RoundRect Lib "gdi32" Alias "RoundRect" (ByVal hdc As Long, ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long, ByVal X3 As Long, ByVal Y3 As Long) As Long
说明
用当前选定的画笔画一个圆角矩形，并用当前选定的刷子在其中填充。X3 和 Y3 定义了用于生成圆角的椭圆
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，用于绘图的设备场景
X1,Y1Long，对矩形左上角位置进行说明的 X，Y 坐标
X2,Y2Long，对矩形右下角位置进行说明的 X，Y 坐标
X3Long，用于生成圆角效果的一个椭圆的宽度。取值范围从零（表示不加圆角），一直到矩形的宽度（全圆）

Y3Long，用于生成圆角效果的一个椭圆的高度。取值范围从零（表示不加圆角），一直到矩形的高度（全圆）

Top
SelectClipPath
SelectClipPath

VB 声明
Declare Function SelectClipPath Lib "gdi32" Alias "SelectClipPath" (ByVal hdc As Long, ByVal iMode As Long) As Long
说明
将设备场景当前的路径合并到剪切区域里
返回值
Long，非零表示成功，零表示失败。会将 GetLastError 设置为下述值之一：ERROR_CAN_NOT_COMPLETE，ERROR_INVALID_PARAMETER，ERROR_NOT_ENOUGH_MEMORY
参数表
参数类型及说明
hdcLong，包含了欲合并到剪切区域的路径的一个设备场景的句柄
iModeLong，决定如何将路径与当前剪切区域合并到一起。可选下述常数之一：
RGN_AND 新的剪切区域只包含了在路径和当前剪切区域都存在的点
RGN_COPY 新剪切区域设为路径
RGN_DIFF 新剪切区域只包含了当前剪切区域的点，但这些点不在路径中
RGN_OR 新的剪切区域包含了在路径或当前剪切区域中的点
RGN_XOR 新的剪切区域只包含了存在于路径或当前剪切区域中的点，但两地共有的点不包括在内

Top
SelectObject
SelectObject

VB 声明
Declare Function SelectObject Lib "gdi32" Alias "SelectObject" (ByVal hdc As Long, ByVal hObject As Long) As Long
说明
每个设备场景都可能有选入其中的图形对象。其中包括位图、刷子、字体、画笔以及区域等等。一次选入设备场景的只能有一个对象。选定的对象会在设备场景的绘图操作中使用。例如，当前选定的画笔决定了在设备场景中描绘的线段颜色及样式
返回值
Long，与以前选入设备场景的相同 hObject 类型的一个对象的句柄，零表示出错。如选定的对象是一个区域（Region），结果就是下列常数之一：SIMPLEREGION，COMPLEXREGION 或 NULLREGION 对区域进行描述，GDI_ERROR 表示出错
参数表
参数类型及说明

hdcLong，一个设备场景的句柄
hObjectLong，一个画笔、位图、刷子、字体或区域的句柄
注解
返回值通常用于获得选入 DC 的对象的原始值。绘图操作完成后，原始的对象通常选回设备场景。在清除一个设备场景前，务必注意恢复原始的对象

Top
SetArcDirection
SetArcDirection

VB 声明
Declare Function SetArcDirection Lib "gdi32" Alias "SetArcDirection" (ByVal hdc As Long, ByVal ArcDirection As Long) As Long
说明
设置圆弧的描绘方向
返回值
Long，如执行成功，返回原始的圆弧方向；零意味着出错
参数表
参数类型及说明
hdcLong，要设置的设备场景
ArcDirectionLong，AD_CLOCKWISE （顺时针）或 AD_COUNTERCLOCKWISE （逆时针）
注解
可应用于下列函数：Arc， ArcTo， Chord， Ellipse， Pie， Rectangle 和 RoundRect

Top
SetBkColor
SetBkColor

VB 声明
Declare Function SetBkColor Lib "gdi32" Alias "SetBkColor" (ByVal hdc As Long, ByVal crColor As Long) As Long
说明
为指定的设备场景设置背景颜色。背景颜色用于填充阴影刷子、虚线画笔以及字符（如背景模式为 OPAQUE）中的空隙。也在位图颜色转换期间使用。参考 SetBkMode
返回值
Long，前一个背景色，CLR_INVALID 表示出错
参数表
参数类型及说明
hdcLong，设备场景的句柄
crColorLong，新背景颜色的 RGB 颜色值
注解
背景实际是设备能够显示的最接近于 crColor 的颜色

Top
SetBkMode
SetBkMode

VB 声明
Declare Function SetBkMode Lib "gdi32" Alias "SetBkMode" (ByVal hdc As Long, ByVal nBkMode As Long) As Long
说明
指定阴影刷子、虚线画笔以及字符中的空隙的填充方式
返回值
Long，前一个背景模式的值
参数表
参数类型及说明
hdcLong，设备场景的句柄
nBkModeLong，下述常数之一：
OPAQUE 用当前的背景色填充虚线画笔、阴影刷子以及字符的空隙
TRANSPARENT 透明处理，即不作上述填充
注解
背景模式不会影响用扩展画笔描绘的线条

Top
SetBrushOrgEx
SetBrushOrgEx

VB 声明
Declare Function SetBrushOrgEx Lib "gdi32" Alias "SetBrushOrgEx" (ByVal hdc As Long, ByVal nXOrg As Long, ByVal nYOrg As Long, lppt As POINTAPI) As Long
说明
为指定的设备场景设置当前选定刷子的起点
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
nXOrg,nYOrgLong，刷子的新起点，采用设备坐标表示。取值范围在 0-7 之间（不需要更大的值，因为 windows95 下最大的刷子的尺寸是 8×8；而这个函数在 NT 下是不必要的）
lpptPOINTAPI，用于装载前一个刷子的起点
注解
Windows NT 会自动设置刷子的起点，所以不应在 NT 下使用这个函数
在 vb 里使用
注意完成以后一定要将设备场景的刷子起点设为 0，0。既可明确指定坐标，也可用 RestoreDC 函数恢复恢复原始的 DC

Top

SetEnhMetaFileBits
SetEnhMetaFileBits

VB 声明
Declare Function SetEnhMetaFileBits Lib "gdi32" Alias "SetEnhMetaFileBits" (ByVal cbBuffer As Long, lpData As Byte) As Long
说明
用指定内存缓冲区内包含的数据创建一个增强型图元文件。在从磁盘读入原始的图元文件数据后（最开始是由 GetEnhMetaFileBits 函数获得的），通常再用这个函数创建一个图元文件
返回值
Long，如执行成功，返回一个增强型图元文件句柄；否则返回零
参数表
参数类型及说明
cbBufferLong，lpData 数组的长度
lpDataByte，一个字节数组的头一个条目，这个数组内包含了图元文件数据

Top
SetMetaFileBitsEx
SetMetaFileBitsEx

VB 声明
Declare Function SetMetaFileBitsEx Lib "gdi32" Alias "SetMetaFileBitsEx" (ByVal nSize As Long, lpData As Byte) As Long
说明
用包含在指定内存缓冲区内的数据结构创建一个图元文件。在从磁盘读入原始的图元文件数据后（最开始是由 GetMetaFileBitsEx 函数获得的），通常再用这个函数创建一个图元文件
返回值
Long，如执行成功，返回一个标准图元文件的句柄；零意味着失败
参数表
参数类型及说明
nSizeLong，lpData 数组的长度
lpDataByte，一个字节数组的头一个条目，这个数组内包含了图元文件数据

Top
SetMiterLimit
SetMiterLimit

VB 声明
Declare Function SetMiterLimit& Lib "gdi32" (ByVal hdc As Long, ByVal eNewLimit As Single, peOldLimit As Single)
说明
设置设备场景当前的斜率限制
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError

参数表
参数类型及说明
hdcLong，欲设置的设备场景
eNewLimitSingle，新的斜率限制
peOldLimitSingle，用于容纳原始斜率限制的一个单精度值
注解
斜率限制是指斜角长度与线宽之间的比率
其他
请看 vb 的 api 文本查看器里的声明：Function SetMiterLimit Lib "gdi32" Alias "SetMiterLimit" (ByVal hdc As Long, ByVal eNewLimit As Double, peOldLimit As Double) As Long

Top
SetPixel
SetPixel

VB 声明
Declare Function SetPixel Lib "gdi32" Alias "SetPixel" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long, ByVal crColor As Long) As Long
说明
在指定的设备场景中设置一个像素的 RGB 值
返回值
Long，指定点的实际 RGB 颜色。如设备不支持指定的准确颜色，则返回的值会与 crColor 有所不同。如指定的点不能设置，则会返回-1（例如，指定的点可能位于设备场景剪切区外面）。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
x,yLong，要设置的点，用逻辑坐标表示
crColorLong，指定像素的新 RGB 颜色
注解
可用 GetDeviceCaps 判断一个设备是否支持这个函数

Top
SetPixelV
SetPixelV

VB 声明
Declare Function SetPixelV Lib "gdi32" Alias "SetPixelV" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long, ByVal crColor As Long) As Long
说明
在指定的设备场景中设置一个像素的 RGB 值
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表

参数类型及说明
hdcLong，一个设备场景的句柄
x,yLong，要设置的点，用逻辑坐标表示
crColorLong，指定像素的新 RGB 颜色值
注解
这个函数比 SetPixel 快一些，但不会返回设置的实际颜色。可用 GetDeviceCaps 判断设备是否支持这个函数

Top
SetPolyFillMode
SetPolyFillMode

VB 声明
Declare Function SetPolyFillMode Lib "gdi32" Alias "SetPolyFillMode" (ByVal hdc As Long, ByVal nPolyFillMode As Long) As Long
说明
设置多边形的填充模式。参考 GetPolyFillMode 函数的注解
返回值
Long，如执行成功，返回前一种多边形填充模式。零表示出错
参数表
参数类型及说明
hdcLong，设备场景的句柄
nPolyFillModeLong，下述常数之一：
ALTERNATE 交替填充
WINDING 根据绘图方向填充

Top
SetROP2
SetROP2

VB 声明
Declare Function SetROP2 Lib "gdi32" Alias "SetROP2" (ByVal hdc As Long, ByVal nDrawMode As Long) As Long
说明
设置指定设备场景的绘图模式。与 vb 的 DrawMode 属性完全一致
返回值
Long，如执行成功，返回前一个绘图模式；零表示出错
参数表
参数类型及说明
hdcLong，设备场景的句柄
nDrawModeLong，设备场景的新绘图模式。参考 GetROP2 函数的绘图模式常数表
注解
在 vb 里，为一个 vb 窗体或图片控件使用设备场景的时候，这个函数会设置 DrawMode 属性

Top
SetWinMetaFileBits
SetWinMetaFileBits

VB 声明
Declare Function SetWinMetaFileBits Lib "gdi32" Alias "SetWinMetaFileBits" (ByVal cbBuffer As Long, lpbBuffer As Byte, ByVal hdcRef As Long, lpmfp As METAFILEPICT) As Long
说明
将一个标准 Windows 图元文件转换成增强型图元文件
返回值
Long，如执行成功，返回一个增强型图元文件（位于内存中）的句柄；零意味着出错
参数表
参数类型及说明
cbBufferLong，lpbBuffer 数组的长度
lpbBufferByte，一个字节数组的头一个条目，这个数组包含了标准图元文件数据。数据是用 GetMetaFileBitsEx 或 GetWinMetaFileBits 函数获得的
hdcRefLong，用于决定原始格式及图元文件分辨率的一个参考设备场景。可以为零，表示采用显示器分辨率
lpmfpMETAFILEPICT，定义图元文件附加参考信息的一个结构。可设为 NULL（用一个别名传递 NULL 值，将参数定义成 ByVal As Long）；此时，会假定使用当前显示器的 MM_ANISOTROPIC 映射模式

Top
StrokeAndFillPath
StrokeAndFillPath

VB 声明
Declare Function StrokeAndFillPath Lib "gdi32" Alias "StrokeAndFillPath" (ByVal hdc As Long) As Long
说明
针对指定的设备场景，关闭路径上打开的所有区域。用当前画笔描绘路径的一个轮廓，并用当前刷子填充路径
返回值
Long，TRUE（非零）表示成功，否则返回零。会将 GetLastError 设置为下述值：ERROR_CAN_NOT_COMPLETE，ERROR_INVALID_PARAMETER，ERROR_NOT_ENOUGH_MEMORY
参数表
参数类型及说明
hdcLong，包含了要描绘和填充的那个路径的一个设备场景
注解
函数执行完后记住清除路径

Top
StrokePath

StrokePath

VB 声明
Declare Function StrokePath Lib "gdi32" Alias "StrokePath" (ByVal hdc As Long) As Long
说明
用当前画笔描绘一个路径的轮廓。打开的图形不会被这个函数关闭
返回值
Long，TRUE（非零）表示成功，否则返回零。会将 GetLastError 设置为下述值：ERROR_CAN_NOT_COMPLETE ， ERROR_INVALID_PARAMETER ， ERROR_NOT_ENOUGH_MEMORY
参数表
参数类型及说明
hdcLong，包含了要描绘和填充的那个路径的一个设备场景

Top
UnrealizeObject
UnrealizeObject

VB 声明
Declare Function UnrealizeObject Lib "gdi32" Alias "UnrealizeObject" (ByVal hObject As Long) As Long
说明
将一个刷子对象选入设备场景之前，如刷子的起点准备用 SetBrushOrgEx 修改，则必须先调用本函数
返回值
Long，TRUE（非零）表示成功，否则返回零。会设置 GetLastError
参数表
参数类型及说明
hObjectLong，刷子或逻辑调色板的句柄
注解
Windows NT 会自动跟踪刷子的起点，所以如果对象是一个刷子，那么这个函数不会产生任何影响。在那个时候，它会返回 TRUE

Top
WidenPath
WidenPath

VB 声明
Declare Function WidenPath Lib "gdi32" Alias "WidenPath" (ByVal hdc As Long) As Long
说明
根据选定画笔的宽度，重新定义当前选定的路径。例如，假设路径描述了一个封闭的矩形。面积为 10×10 像素，而且用一个 3 像素宽的画笔描绘。此时，加宽后的路径将由一个 12×12 的矩形构成
返回值

Long，TRUE（非零）表示成功，否则返回零。会将 GetLastError 设置为下述值：ERROR_CAN_NOT_COMPLETE，ERROR_INVALID_PARAMETER，ERROR_NOT_ENOUGH_MEMORY
参数表
参数类型及说明
hdcLong，包含路径的一个设备的句柄
注解
所有贝塞尔曲线都会由这个函数转换成线段

Top
打印函数
打印函数，共五页。第一页，第二页，第三页，第四页，第五页
AbortDoc 取消一份文档的打印
AbortPrinter 删除与一台打印机关联在一起的缓冲文件
AddForm 为打印机的表单列表添加一个新表单
AddJob 用于获取一个有效的路径名，以便用它为作业创建一个后台打印文件。它也会为作业分配一个作业编号
AddMonitor 为系统添加一个打印机监视器
AddPort 启动“添加端口”对话框，允许用户在系统可用端口列表中加入一个新端口
AddPrinter 在系统中添加一台新打印机
AddPrinterConnection 连接指定的打印机
AddPrinterDriver 为指定的系统添加一个打印驱动程序
AddPrintProcessor 为指定的系统添加一个打印处理器
AddPrintProvidor 为系统添加一个打印供应商
AdvancedDocumentProperties 启动打印机文档设置对话框
ClosePrinter 关闭一个打开的打印机对象
ConfigurePort 针对指定的端口，启动一个端口配置对话框
ConnectToPrinterDlg 启动连接打印机对话框，用它同访问网络的打印机连接
DeleteForm 从打印机可用表单列表中删除一个表单
AbortDoc
AbortDoc

VB 声明
Declare Function AbortDoc Lib "gdi32" Alias "AbortDoc" (ByVal hdc As Long) As Long
说明
取消一份文档的打印。自上次调用 StartDoc 函数以来的所有输出都会被取消。如对打印机进行了配置，令其在正式打印文档之前先在打印缓冲区内对文档进行排队，那么文档的任何一部分都不会打印；否则，就可能出现文档打印到一半被取消的情况
返回值
Long，大于零表示成功，SP_ERROR 表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
在 VB 里使用

如将这个函数用于由打印机对象的 hDC 属性指定的打印机设备场景，那么它可以正常发挥作用。然而，倘若之后调用了 EndDoc 方法，却有可能得到一条打印机出错消息。当大家结合 API 打印函数与 VB 打印机方法的时候，强烈建议对打印机的错误进行跟踪捕获；或干脆避免这种结合

Top
AbortPrinter
AbortPrinter

VB 声明
Declare Function AbortPrinter Lib "winspool.drv" Alias "AbortPrinter" (ByVal hPrinter As Long) As Long
说明
删除与一台打印机关联在一起的缓冲文件
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，一个已打开的打印机对象的句柄（用 OpenPrinter 获得）
注解
如打印机没有使用后台打印缓冲文件，那么该函数将无法发挥作用。例如，后台打印程序可将数据直接发给打印机

Top
AddForm
AddForm

VB 声明
Declare Function AddForm& Lib "spoolss.dll" Alias "AddFormA" (ByVal hPrinter As Long, ByVal Level As Long, pForm As FORM_INFO_1)
说明
为打印机的表单列表添加一个新表单。“表单”描述了一个页面大小及布局，提供了一种与设备无关的机制，可实现 Windows NT 下的纸张尺寸的标准化
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，一个已打开的打印机对象的句柄（用 OpenPrinter 获得）
LevelLong，设为 1
pFormFORM_INFO_1，对表单进行描述的一个结构
适用平台
Windows NT
其他
在 VB 的 API 文本查看器里复制的声明如下：

Declare Function AddForm Lib "winspool.drv" Alias "AddFormA" (ByVal hPrinter As Long, ByVal Level As Long, pForm As Byte) As Long

Top
AddJob
AddJob

VB 声明
Declare Function AddJob Lib "winspool.drv" Alias "AddJobA" (ByVal hPrinter As Long, ByVal Level As Long, pData As Byte, ByVal cdBuf As Long, pcbNeeded As Long) As Long
说明
用于获取一个有效的路径名，以便用它为作业创建一个后台打印文件。它也会为作业分配一个作业编号
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，一个已打开的打印机对象的句柄（用 OpenPrinter 获得）
LevelLong，设为 1
pDataByte，缓冲区会引用一个 ADDJOB_INFO_1 结构
cdBufLong，pData 缓冲区中的字符数量
pcbNeededLong，指向一个 Long 型变量的指针，该变量用于保存请求的缓冲区长度，或者实际读入的字节数量
注解
调用这个函数以后，可创建指定的文件，向其中写入数据，然后用 API 函数 ScheduleJob 令其将数据发给打印机

Top
AddMonitor
AddMonitor

VB 声明
Declare Function AddMonitor Lib "winspool.drv" Alias "AddMonitorA" (ByVal pName As String, ByVal Level As Long, pMonitors As Byte) As Long
说明
为系统添加一个打印机监视器
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，欲在其中安装监视器的一个服务器的名字。对于本地（本机）监视器，请设置成 vbNullString
LevelLong，设为 2
pMonitorsByte，指定一个结构中的第一个字节。那个结构又包含了一个 MONITOR_INFO_2

结构

Top
AddPort
AddPort

VB 声明
Declare Function AddPort Lib "winspool.drv" Alias "AddPortA" (ByVal pName As String, ByVal hwnd As Long, ByVal pMonitorName As String) As Long
说明
启动“添加端口”对话框，允许用户在系统可用端口列表中加入一个新端口
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，欲在其中安装端口的一个服务器的名字。对本地（本机）端口，请设置成 vbNullString
hwndLong，指定 AddPort 对话框的父窗口的句柄
pMonitorNameString，用于指定端口的一个监视器的名称

Top
AddPrinter
AddPrinter

VB 声明
Declare Function AddPrinter Lib "winspool.drv" Alias "AddPrinterA" (ByVal pName As String，ByVal Level As Long， pPrinter As Any) As Long
说明
在系统中添加一台新打印机
返回值
Long，如执行成功，返回一台新打印机的句柄；零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，欲在其中安装打印机的一个服务器的名字。对本地打印机，设为 vbNullString
LevelLong，设为 2
pPrinterAny，指定一个缓冲区的第一个条目。该缓冲区包含了一个 PRINTER_INFO_2 结构。结构中的下述字段会设为有效值：pPrinterName，pPortName，pDriverName，pPrintProcessor 和 pDataType。也可象 PRINTER_INFO_2 那样设置 pPrinter 字段。也可以设置下述字段：Attributes， DefaultPriority， pComment， pDevMode， pLocation， pParameters， Priority，pSecurityDescriptor， pSepFile， pShareName， StartTime 和 UntilTime。而其他字段都应置空
注解
在 NT 下，调用者必须有足够的权限对指定服务器上的打印机进行配置

Top
AddPrinterConnection
AddPrinterConnection

VB 声明
Declare Function AddPrinterConnection Lib "winspool.drv" Alias "AddPrinterConnectionA" (ByVal pName As String) As Long
说明
连接指定的打印机
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，网络上的一台打印机的名字

Top
AddPrinterDriver
AddPrinterDriver

VB 声明
Declare Function AddPrinterDriver Lib "winspool.drv" Alias "AddPrinterDriverA" (ByVal pName As String, ByVal Level As Long, pDriverInfo As Any) As Long
说明
为指定的系统添加一个打印驱动程序
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，指定要在其中安装驱动程序的一台服务器的名字。对于本地系统，设为 vbNullString
LevelLong，2 或 3（2 仅适用于 NT 3.51）
pDriverInfoAny，指定一个缓冲区，其中包含了一个 DRIVER_INFO_2 或 DRIVER_INFO_3 结构，它们指定了要添加的驱动程序
注解
在调用这个函数之前，所有驱动程序文件都必须位于适当的目录

Top
AddPrintProcessor
AddPrintProcessor

VB 声明
Declare Function AddPrintProcessor Lib "winspool.drv" Alias "AddPrintProcessorA" (ByVal pName As String, ByVal pEnvironment As String, ByVal pPathName As String, ByVal pPrintProcessorName As String) As Long

说明
为指定的系统添加一个打印处理器
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，指定要在其中安装驱动程序的一台服务器的名字。对于本地系统，设为 vbNullString
pEnvironmentString，要在其中添加打印处理器的一个环境（如“Windows NT x86”）。对于当前（本地）系统环境，则设为 vbNullString
pPathNameString，包含了打印管理器的一个文件的名字。文件必须位于打印处理器目录中
pPrintProcessorNameString，打印处理器的名字

Top
AddPrintProvidor
AddPrintProvidor

VB 声明
Declare Function AddPrintProvidor Lib "winspool.drv" Alias "AddPrintProvidorA" (ByVal pName As String, ByVal Level As Long, pProvidorInfo As Byte) As Long
说明
为系统添加一个打印供应商
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，指出要在其中安装打印供应商的一台服务器的名字。对于本地系统，设为 vbNullString
LevelLong，设为 1
pProvidorInfoByte，包含了一个 PROVIDOR_INFO_1 结构的缓冲区

Top
AdvancedDocumentProperties
AdvancedDocumentProperties

VB 声明
Declare Function AdvancedDocumentProperties Lib "winspool.drv" Alias "AdvancedDocumentPropertiesA" (ByVal hwnd As Long, ByVal hPrinter As Long, ByVal pDeviceName As String, pDevModeOutput As DEVMODE, pDevModeInput As DEVMODE) As Long
说明
启动打印机文档设置对话框。这个函数几乎完全等价于调用 DocumentProperties 函数，同时将 fMode 设为 DM_IN_PROMPT。请参考对 DocumentProperties 函数的说明，了解这个函数的详细情况

返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
注解
将 pDevModeOutput 设为 0 后可得到要求的 DEVMODE 结构的大小

Top
ClosePrinter
ClosePrinter

VB 声明
Declare Function ClosePrinter Lib "winspool.drv" Alias "ClosePrinter" (ByVal hPrinter As Long) As Long
说明
关闭一个打开的打印机对象
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，一个打开的打印机对象的句柄

Top
ConfigurePort
ConfigurePort

VB 声明
Declare Function ConfigurePort Lib "winspool.drv" Alias "ConfigurePortA" (ByVal pName As String, ByVal hwnd As Long, ByVal pPortName As String) As Long
说明
针对指定的端口，启动一个端口配置对话框
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，欲对其端口进行配置的一台服务器的名字。对于本地系统，请设为 vbNullString
hwndLong，对话框父窗口的句柄
pPortNameString，端口名

Top
ConnectToPrinterDlg
ConnectToPrinterDlg

VB 声明
Declare Function ConnectToPrinterDlg Lib "winspool.drv" Alias "ConnectToPrinterDlg" (ByVal hwnd As Long, ByVal flags As Long) As Long

说明
启动连接打印机对话框，用它同访问网络的打印机连接
返回值
Long，已连接或选择的打印机的句柄，零意味着失败或用户取消了操作
参数表
参数类型及说明
hwndLong，对话框的父窗口句柄
flagsLong，保留，设为零

Top
DeleteForm
DeleteForm

VB 声明
Declare Function DeleteForm Lib "winspool.drv" Alias "DeleteFormA" (ByVal hPrinter As Long, ByVal pFormName As String) As Long
说明
从打印机可用表单列表中删除一个表单
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，一个打开的打印机的名字（用 OpenPrinter 获得）
pFormNameString，欲删除的表单的名字
适用平台
Windows NT
注解
不能删除内建表单。请参考 AddForm 函数了解进一步的情况

Top
DeleteMonitor
DeleteMonitor

VB 声明
Declare Function DeleteMonitor Lib "winspool.drv" Alias "DeleteMonitorA" (ByVal pName As String, ByVal pEnvironment As String, ByVal pMonitorName As String) As Long
说明
删除指定的打印监视器
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，监视器所在的服务器的名字。用 vbNullString 指定本地系统
pEnvironmentString，欲在其中删除监视器的环境。用 vbNullString 指定当前系统

pMonitorNameString，欲删除监视器的名字
注解
参考 AddMonitor 函数，了解进一步的信息

Top
DeletePort
DeletePort

VB 声明
Declare Function DeletePort Lib "winspool.drv" Alias "DeletePortA" (ByVal pName As String, ByVal hwnd As Long, ByVal pPortName As String) As Long
说明
启动“删除端口”对话框，允许用户从当前系统删除一个端口
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，欲在其中删除端口的服务器的名字。vbNullString 表示使用本地端口
hwndLong，DeletePort 对话框的父窗口的句柄
pPortNameString，欲删除的端口的名字
注解
如打印机已连接到端口，则函数执行会失败

Top
DeletePrinter
DeletePrinter

VB 声明
Declare Function DeletePrinter Lib "winspool.drv" Alias "DeletePrinter" (ByVal hPrinter As Long) As Long
说明
将指定的打印机标志为从系统中删除
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，已打开的打印机对象的句柄
注解
除非所有待决的打印作业都已完成，否则打印机不会实际删除

Top
DeletePrinterConnection
DeletePrinterConnection

VB 声明
Declare Function DeletePrinterConnection Lib "winspool.drv" Alias "DeletePrinterConnectionA" (ByVal pName As String) As Long
说明
删除与指定打印机的连接
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，要删除的打印机连接

Top
DeletePrinterDriver
DeletePrinterDriver

VB 声明
Declare Function DeletePrinterDriver Lib "winspool.drv" Alias "DeletePrinterDriverA" (ByVal pName As String, ByVal pEnvironment As String, ByVal pDriverName As String) As Long
说明
从系统删除一个打印机驱动程序
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，服务器的名字。针对本地驱动程序设为 vbNullString
pEnvironmentString，指定欲在其中删除驱动程序的一个环境（如“Windows NT x86”）。对于当前（本地）系统环境，则设为 vbNullString
pDriverNameString，要删除的驱动程序的名字
注解
驱动程序文件不会从系统物理性的删除，只是不能继续使用

Top
DeletePrintProcessor
DeletePrintProcessor

VB 声明
Declare Function DeletePrintProcessor Lib "winspool.drv" Alias "DeletePrintProcessorA" (ByVal pName As String, ByVal pEnvironment As String, ByVal pPrintProcessorName As String) As Long
说明
从指定系统删除一个打印处理器
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表

参数类型及说明
pNameString，服务器的名字。针对本地处理器设为 vbNullString
pEnvironmentString，指定欲删除打印处理器的一个环境（如“Windows NT x86”）。对于当前（本地）系统环境，则设为 vbNullString
pPrintProcessorNameString，要删除的打印处理器的名字

Top
DeletePrintProvidor
DeletePrintProvidor

VB 声明
Declare Function DeletePrintProvidor Lib "winspool.drv" Alias "DeletePrintProvidorA" (ByVal pName As String, ByVal pEnvironment As String, ByVal pPrintProvidorName As String) As Long
说明
从系统中删除一个打印供应商
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，服务器的名字。针对本地服务器设为 vbNullString
pEnvironmentString，指定欲删除打印供应商的一个环境（如“Windows NT x86”）。对于当前（本地）系统环境，则设为 vbNullString
pPrintProvidorNameString，要删除的打印供应商的名字

Top
DeviceCapabilities
DeviceCapabilities

VB 声明
Declare Function DeviceCapabilities Lib "winspool.drv" Alias "DeviceCapabilitiesA" (ByVal lpDeviceName As String, ByVal lpPort As String, ByVal iIndex As Long, ByVal lpOutput As String, lpDevMode As DEVMODE) As Long
说明
利用这个函数可获得与一个设备的能力有关的信息
返回值
Long，由 iIndex 参数的值决定，请参考设备能力常数表。如函数执行失败，或打印机的驱动程序不支持这个函数，那么函数就会返回-1
参数表
参数类型及说明
lpDeviceNameString，设备名
lpPortString，指定连接了指定设备的那个端口
iIndexLong，欲测试的能力。请参考设备能力常数表，其中列出了可选的值
lpOutputString，指定一个缓冲区的地址，能力数据会装载到这个缓冲区中。在设备能力常数表中，针对每个 fwCapabilities 值的缓冲区的内容都进行了总结。这个表格同时总结了应将

参数设为 vbNullString 的一些情况
lpDevModeDEVMODE，一个 DEVMODE 结构的地址，或者为零。如指定了那个结构，函数会根据这个结构的设置来接收信息。如果为零，函数就会根据打印机驱动程序的默认值接收信息
注解
使用 lpOutput 时要注意：在许多时候，这个函数会返回一系列名称的列表。例如，假设将 fwCapabilities 标志设为 DC_PAPERNAMES，那么就会得到一系列支持的纸张尺寸的名字。在这种情况下，lpOutput 缓冲区应该是一个 String 变量，而且根据设备能力常数表的总结预先初始化成合适的长度。函数会在缓冲区中载入所有名称，而且每个名称在字串中都占用固定的空间。所以，我们完全能用 Mid 函数提取出每一个条目。
某些情况下，lpOutput 需要指向一个数值数组的指针

设备能力常数表
fwCapabilities 说明
DC_BINADJUST 返回来自 API32.TXT 的某个常数。它应带有 DCBA_ 前缀，用于指定当前纸张源的正确纸张方向。仅适用于 Win95
DC_BINNAMES 如 lpOutput 为零，就返回由打印机支持的纸匣数量。否则，lpOutput 应指向一个缓冲区（长度至少为 24×纸匣数）。每 24 个字节都会保存一个纸匣的 NULL 中止名称
DC_BINS 如 lpOutput 为零，就返回由打印机支持的纸匣数量。否则，lpOutput 应指向一个整数数组（长度至少为纸匣数量）。这些值对应于为 DEVMODE 结构定义的 DMBIN_??? 常数
DC_COPIES 返回打印机能够打印的最大副本数量
DC_DATATYPE_PRODUCED 接收由打印机支持的一系列数据类型。这些类型可作为由 StartDoc 函数使用的 DOCINFO 结构的输出数据类型提供。如这个函数返回-1，那么支持的唯一数据类型就是 RAW 。仅适用于 Win95
DC_DRIVER 返回打印机驱动程序的版本号
DC_DUPLEX 如打印机有双面打印功能，就返回 1；否则返回 0
DC_EMF_COMPLIANT 如打印机能直接支持增强型图元文件，就返回 TRUE。仅适用于 Win95
DC_ENUMRESOLUTIONS 如 lpOutput 为零，就返回由打印机支持的分辨率数量。否则，lpOutput 应该是一个指向 Long 型数组的指针。该数组至少应包含（2×分辨率数量）个条目。每对条目都反映出水平和垂直分辨率（以每英寸的点数——dpi——为单位）
DC_EXTRA 返回与具体设备有关的特殊字节，它们要为这个设备追加到 DEVMODE 结构后面
DC_FIELDS 针对设备默认的 DEVMODE 数据结构，返回 dmFields 字段的值
DC_FILEDEPENDENCIES 如 lpOutput 为零，就返回打印机驱动程序要求的文件数量。否则，lpOutput 应指向一个至少有（64×文件数）个字节的缓冲区。每 64 个字节都会保存一个请求文件的 NULL 中止名称
DC_MAXEXTENT 返回一个 Long 型值，其中包含打印机支持的最大纸张长度和宽度。其中，低字（16 位）包含的是宽度数据。它们是由 dmPaperWidth 和 dmPaperLength 这两个 DEVMODE 字段的最大值
DC_MINEXTENT 返回一个 Long 型值，其中包含打印机支持的最小纸张长度和宽度。其中，低字（16 位）包含的是宽度数据。它们是由 dmPaperWidth 和 dmPaperLength 这两个

DEVMODE 字段的最大值
DC_ORIENTATION 返回横向模式和纵向模式间的旋转度数。如果是零，表示驱动程序不支持横向打印模式。对于激光打印机，90 度是最常见的一个设置；而对于点阵式打印机，一般都是 270 度
DC_PAPERNAMES 如 lpOutput 为零，就返回由打印机支持的纸张尺寸数量。否则，lpOutput 就应指向一个缓冲区（长度至少为 64×纸张尺寸种数）。每 64 个字节都会装载一种支持的纸张尺寸的空中止名称
DC_PAPERS 如 lpOutput 为零，就返回由打印机支持的纸张尺寸数量。否则，lpOutput 就应指向一个整数数组（长度至少为纸张的尺寸种数）。值对应于为 DEVMODE 结构定义的 DMPAPER_??? 常数
DC_SIZE 返回打印机 DEVMODE 数据结构的 dmSize 字段
DC_TRUETYPE 下述常数之一：
DCTT_BITMAP 设备能将 TrueType 字体当作图形打印
DCTT_DOWNLOAD 设备能下载 TrueType 字体
DCTT_OUTLINE 设备能下载轮廓型 TrueType 字体
DCTT_SUBDEV 设备能取代与对应的 TrueType 字体兼容的内建字体
DC_VERSION 返回设备驱动程序的规格版本号

Top
DocumentProperties
DocumentProperties

VB 声明
Declare Function DocumentProperties& Lib "winspool.dll" Alias "DocumentPropertiesA" (ByVal hwnd As Long, ByVal hPrinter As Long, ByVal pDeviceName As String, ByVal pDevModeOutput As Long, ByVal pDevModeInput As Long, ByVal fMode As Long)
说明
这是一个灵活的打印机配置控制函数。该函数定义了两个 DEVMODE 结构，可在创建一个设备场景时为单个应用程序改变打印机设置。甚至能在文档打印期间改变打印机设置
返回值
Long，由 fMode 字段的值决定。如下所示：
若 fMode 为零，这个函数就返回 DEVMODE 结构的尺寸。注意这个结构可能比类型定义文件 API32.TXT 中规定的尺寸大
若 fMode 设置了 DM_IN_PROMPT 标志，那么打印机设置对话框就会出现。在这种情况下，返回值将是常数 IDOK 或 IDCANCEL——具体由用户关闭对话框时按下的按钮决定
在其他任何情况下，该函数执行成功后会返回 IDOK。而在任何情况下，如函数执行失败，都会返回一个负数
参数表
参数类型及说明
hwndLong，对话框父窗口的句柄。这通常是当前的活动窗体
hPrinterLong，一个已打开的打印机对象的句柄
pDeviceNameString，打印机的名字
pDevModeOutputLong，指向一个 DEVMODE 数据结构的指针。请参考 DocumentProperties 运行模式表。注意这个指针必须引用一个足够大的缓冲区，它能同时容下专用打印机驱动程

序数据，以及标准的 DEVMODE 结构
pDevModeInputLong，指向一个 DEVMODE 数据结构的指针。请参考 DocumentProperties 运行模式表
fModeLong，决定这个函数运作模式的一个标志。请参考 DocumentProperties 运行模式表

DocumentProperties 运行模式表
常数标志运行模式
无不使用 pDevModeInput。pDevModeOutput 可能为零。函数会返回由这两个参数引用的 DEVMODE 结构需要的大小
DM_IN_BUFFERpDevModeInput 缓冲区应载入打印机驱动程序的新位置。在调用这个函数判断应使用结构中的哪些字段前，应设置结构的 dmFields 字段
DM_IN_PROMPT 显示出打印机设置对话框，以便用户指定输出时采用的打印机设置。如指定了 DM_IN_BUFFER，那么在显示对话框前，输入缓冲区中指定的任何字段都会与当前的打印机 DEVMODE 结构合并起来
DM_OUT_BUFFER 令打印机设置信息输出到由 pDevModeOutput 参数指定的缓冲区。这些设置由两个输入标志决定，而且由此反映了原始的输入结构、当前的打印机设置以及用户在打印机设置对话框中作出的任何修改。如未指定这个标志，lpdmOutput 参数就可以设为零

Top
EndDocAPI
EndDocAPI

VB 声明
Declare Function EndDocAPI& Lib "gdi32" Alias "EndDoc" (ByVal hDC As Long)
说明
结束一个成功的打印作业
返回值
Long，大于零表示成功，小于或等于零表示失败。会设置 GetLastError
参数表
参数类型及说明
hDCLong，设备场景的句柄

Top
EndDocPrinter
EndDocPrinter

VB 声明
Declare Function EndDocPrinter Lib "winspool.drv" Alias "EndDocPrinter" (ByVal hPrinter As Long) As Long
说明
在后台打印程序的级别指定一个文档的结束
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表

参数类型及说明
hPrinterLong，一个已打开的打印机的句柄（用用 OpenPrinter 获得）
注解
在应用程序的级别并非特别有用。后台打印程序用它标识打印作业中一个文档的结束

Top
EndPage
EndPage

VB 声明
Declare Function EndPage Lib "gdi32" Alias "EndPage" (ByVal hdc As Long) As Long
说明
用这个函数完成一个页面的打印，并准备设备场景，以便打印下一个页
返回值
Long，大于零表示成功，小于或等于零表示失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
在 VB 里使用
使用 VB 的打印机对象时，应该使用 NewPage 方法来代替这个函数
注解
这个函数的执行可能相当耗时——具体取决于正在进行的绘图操作的复杂程度、使用什么操作系统以及打印机与本机连接还是与远程主机连接。为了让用户能随时取消打印，在执行期间，这个函数会定时调用由 API 调用 SetAbortProc 指定的取消例程

Top
EndPagePrinter
EndPagePrinter

VB 声明
Declare Function EndPagePrinter Lib "winspool.drv" Alias "EndPagePrinter" (ByVal hPrinter As Long) As Long
说明
指定一个页在打印作业中的结尾
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，一个已打开的打印机对象的句柄（用 OpenPrinter 获得）
注解
在应用程序的级别并非特别有用，后台打印程序用它在实际打印机数据中标识一个页的结尾

Top
EnumForms

EnumForms

VB 声明
Declare Function EnumForms Lib "winspool.drv" Alias "EnumFormsA" (ByVal hPrinter As Long, ByVal Level As Long, pForm As Byte, ByVal cbBuf As Long, pcbNeeded As Long, pcReturned As Long) As Long
说明
枚举一台打印机可用的表单
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，一个已打开的打印机对象的句柄（用 OpenPrinter 获得）
LevelLong，设为 1
pFormByte，一个包含 FORM_INFO_1 结构的缓冲区
cbBufLong，pForm 缓冲区中的字符数量
pcbNeededLong，指向一个 Long 型变量的指针，该变量用于保存请求的缓冲区长度，或者实际读入的字节数量
pcReturnedLong，载入缓冲区的结构数量（用于那些能返回多个结构的函数）
适用平台
Windows NT
注解
参考 AddForm 函数，了解进一步的情况

Top
EnumJobs
EnumJobs

VB 声明
Declare Function EnumJobs Lib "winspool.drv" Alias "EnumJobsA" (ByVal hPrinter As Long, ByVal FirstJob As Long, ByVal NoJobs As Long, ByVal Level As Long, pJob As Byte, ByVal cdBuf As Long, pcbNeeded As Long, pcReturned As Long) As Long
说明
枚举打印队列中的作业
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，一个已打开的打印机对象的句柄（用 OpenPrinter 获得）
FirstJobLong，作业列表中要枚举的第一个作业的索引（注意编号从 0 开始）
NoJobsLong，要枚举的作业数量
LevelLong，1 或 2
pJobByte，包含 JOB_INFO_1 或 JOB_INFO_2 结构的缓冲区
cbBufLong，pJob 缓冲区中的字符数量

pcbNeededLong，指向一个 Long 型变量的指针，该变量用于保存请求的缓冲区长度，或者实际读入的字节数量
pcReturnedLong，载入缓冲区的结构数量（用于那些能返回多个结构的函数）
适用平台

Top
EnumMonitors
EnumMonitors

VB 声明
Declare Function EnumMonitors Lib "winspool.drv" Alias "EnumMonitorsA" (ByVal pName As String, ByVal Level As Long, pMonitors As Byte, ByVal cbBuf As Long, pcbNeeded As Long, pcReturned As Long) As Long
说明
枚举可用的打印监视器
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，服务器的名字。用 vbNullString 指定本地系统
LevelLong，设为 1
pMonitorsByte，包含 MONINTOR_INFO_1 结构的缓冲区
cbBufLong，pMonitors 缓冲区中的字符数量
pcbNeededLong，指向一个 Long 型变量的指针，该变量用于保存请求的缓冲区长度，或者实际读入的字节数量
pcReturnedLong，载入缓冲区的结构数量（用于那些能返回多个结构的函数）
注解
MONINTOR_INFO_1 结构有一个字段包含了打印监视器的名字

Top
EnumPorts
EnumPorts

VB 声明
Declare Function EnumPorts Lib "winspool.drv" Alias "EnumPortsA" (ByVal pName As String, ByVal Level As Long, ByVal lpbPorts As Long, ByVal cbBuf As Long, pcbNeeded As Long, pcReturned As Long) As Long
说明
枚举一个系统可用的端口
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，指定服务器的名字。用 vbNullString 指定本地系统

LevelLong，1 或 2（1 用于 NT 3.51），分别指定 PORT_INFO_1 或 PORT_INFO_2
lpbPortsLong，包含 PORT_INFO_1 或 PORT_INFO_2 结构的缓冲区
cbBufLong，lpbPorts 缓冲区中的字符数量
pcbNeededLong，指向一个 Long 型变量的指针，该变量用于保存请求的缓冲区长度，或者实际读入的字节数量
pcReturnedLong，载入缓冲区的结构数量（用于那些能返回多个结构的函数）
注解
参考 AddPort 函数，了解进一步的情况

Top
EnumPrinterDrivers
EnumPrinterDrivers

VB 声明
Declare Function EnumPrinterDrivers Lib "winspool.drv" Alias "EnumPrinterDriversA" (ByVal pName As String, ByVal pEnvironment As String, ByVal Level As Long, pDriverInfo As Byte, ByVal cdBuf As Long, pcbNeeded As Long, pcRetruned As Long) As Long
说明
枚举指定系统中已安装的打印机驱动程序
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，指定服务器的名字。用 vbNullString 指定本地系统
pEnvironmentString，欲在其中对驱动程序进行枚举的环境（如：Windows NT x86）。如设为 vbNullString，表示使用当前（本地）系统环境
LevelLong，1，2 或 3（3 仅适用于 Windows 95 和 NT 4.0）
pDriverInfoByte，包含 DRIVER_INFO_1， DRIVER_INFO_2 或 DRIVER_INFO_3 结构的缓冲区
cbBufLong，pDriverInfo 缓冲区中的字符数量
pcbNeededLong，指向一个 Long 型变量的指针，该变量用于保存请求的缓冲区长度，或者实际读入的字节数量
pcReturnedLong，载入缓冲区的结构数量（用于那些能返回多个结构的函数）

Top
EnumPrinters
EnumPrinters

VB 声明
Declare Function EnumPrinters Lib "winspool.drv" Alias "EnumPrintersA" (ByVal flags As Long, ByVal name As String, ByVal Level As Long, pPrinterEnum As Byte, ByVal cdBuf As Long, pcbNeeded As Long, pcReturned As Long) As Long
说明
枚举系统中安装的打印机

返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
flagsLong，一个或多个下述标志
PRINTER_ENUM_LOCAL 枚举本地打印机（包括 Windows 95 中的网络打印机）。名字会被忽略
PRINTER_ENUM_NAME 枚举由 name 参数指定的打印机。其中的名字可以是一个供应商、域或服务器。如 name 为 NULL，则枚举出可用的打印机
PRINTER_ENUM_SHARE 枚举共享打印机（必须同其他常数组合使用）
PRINTER_ENUM_CONNECTIONS 枚举网络连接列表中的打印机（即使目前没有连接——仅适用于 NT）
PRINTER_ENUM_NETWORK 枚举通过网络连接的打印机。级别（Level）必须为 1。仅适用于 NT
PRINTER_ENUM_REMOTE 枚举通过网络连接的打印机和打印服务器。级别必须为 1。仅适用于 NT
nameString，vbNullString 表示枚举同本机连接的打印机。否则由标志和级别决定
LevelLong，1，2，4 或 5（4 仅适用于 NT；5 仅适用于 Win95 和 NT 4.0），指定欲枚举的结构的类型。如果是 1，则 name 参数由标志设置决定。如果是 2 或 5，那么 name 就代表欲对其打印机进行枚举的服务器的名字；或者为 vbNullString。如果是 4，那么只有 PRINTER_ENUM_LOCAL 和 PRINTER_ENUM_CONNECTIONS 才有效。名字必须是 vbNullString
pPrinterEnumByte，包含 PRINTER_ENUM_x 结构的缓冲区，其中的 x 代表级别（Level）
cbBufLong，pPrinterEnum 缓冲区中的字符数量
pcbNeededLong，指向一个 Long 型变量的指针，该变量用于保存请求的缓冲区长度，或者实际读入的字节数量
pcReturnedLong，载入缓冲区的结构数量（用于那些能返回多个结构的函数）
注解
第 4 和第 5 级将它们的结构建立在系统注册表的基础上，而且比第 2 级快得多。后者要求每台打印机都处于打开状态
请参考微软 Win32 手册，了解这个函数进一步的情况

Top
EnumPrintProcessorDatatypes
EnumPrintProcessorDatatypes

VB 声明
Declare Function EnumPrintProcessorDatatypes Lib "winspool.drv" Alias "EnumPrintProcessorDatatypesA" (ByVal pName As String, ByVal pPrintProcessorName As String, ByVal Level As Long, pDatatypes As Byte, ByVal cdBuf As Long, pcbNeeded As Long, pcRetruned As Long) As Long
说明
枚举由一个打印处理器支持的数据类型
返回值

Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，指定服务器的名字。用 vbNullString 表示使用本地系统
pPrintProcessorNameString，欲对其数据类型进行枚举的打印处理器的名字
LevelLong，设为 1
pDatatypesByte，包含 DATATYPES_INFO_1 结构的缓冲区
cbBufLong，pDatatypes 缓冲区中的字符数量
pcbNeededLong，指向一个 Long 型变量的指针，该变量用于保存请求的缓冲区长度，或者实际读入的字节数量
pcReturnedLong，载入缓冲区的结构数量（用于那些能返回多个结构的函数）
注解
DATATYPES_INFO_1 结构包含了单个字段，其中保存了数据类型的名称

Top
EnumPrintProcessors
EnumPrintProcessors

VB 声明
Declare Function EnumPrintProcessors Lib "winspool.drv" Alias "EnumPrintProcessorsA" (ByVal pName As String, ByVal pEnvironment As String, ByVal Level As Long, pPrintProcessorInfo As Byte, ByVal cdBuf As Long, pcbNeeded As Long, pcReturned As Long) As Long
说明
枚举系统中可用的打印处理器
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，指定服务器的名字。用 vbNullString 表示使用本地系统
pEnvironmentString，欲枚举的打印处理器的环境（如： Windows NT x86）。如设为 vbNullString，表示使用当前（本地）系统环境
LevelLong，设为 1
pPrintProcessorInfoByte，包含 PRINTPROCESSOR_INFO_1 结构的缓冲区
cbBufLong，pPrintProcessorInfo 缓冲区中的字符数量
pcbNeededLong，指向一个 Long 型变量的指针，该变量用于保存请求的缓冲区长度，或者实际读入的字节数量
pcReturnedLong，载入缓冲区的结构数量（用于那些能返回多个结构的函数）

Top
Escape
Escape

VB 声明
Declare Function Escape Lib "gdi32" Alias "Escape" (ByVal hdc As Long, ByVal nEscape As

Long, ByVal nCount As Long, ByVal lpInData As String, lpOutData As Any) As Long
说明
一个灵活的设备控制函数
返回值
Long，对于 QUERYESCSUPPORT，如支持指定的换码，则返回 TRUE（非零）；否则返回零。对于 PASSTHROUGH，大于零值表示成功；如指定的换码不支持，则返回零；如果出错，则返回负值
参数表
参数类型及说明
hdcLong，设备场景的句柄
nEscapeLong，换码数量，由 API32.TXT 文件中的一个常数定义。这决定了具体的运作方式。请参考注解
nCountLong，lpInData 缓冲区的大小，用字节数表示
lpInDataString，由换码类型决定。对于 QUERYESCSUPPORT，这代表指向一个整数变量的指针，那个变量包含了要测试的换码值。对于 PASSTHROUGH，这代表指向一个数据块的指针，那个数据块包含于要发送数据的头 16 位字节数量中。数据块剩余的部分包含了要发送给打印机的实际数据缓冲区
lpOutDataAny，指定一个输出缓冲区，它的具体使用由换码决定。它不由 QUERYESCSUPPORT 或 PASSTHROUGH 使用，而且应设为 NULL（ByVal 0&）
注解
只有两个换码在 Win32 环境中经常用到。请用 QUERYESCSUPPORT 换码判断一个换码是否得到了驱动程序的支持。用 PASSTHROUGH 换码将原始数据直接发给一台打印机。其他换码在 Win32 仍然得到了支持，但目的只是为了与 Win16 保持兼容

Top
FindClosePrinterChangeNotification
FindClosePrinterChangeNotification

VB 声明
Declare Function FindClosePrinterChangeNotification Lib "winspool.drv" Alias "FindClosePrinterChangeNotification" (ByVal hChange As Long) As Long
说明
关闭用 FindFirstPrinterChangeNotification 函数获取的一个打印机通告对象
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hChangeLong，欲关闭的打印机通告对象句柄
适用平台
Windows NT

Top
FindFirstPrinterChangeNotification
FindFirstPrinterChangeNotification

VB 声明
Declare Function FindFirstPrinterChangeNotification& Lib "winspool.dll" (ByVal hPrinter As Long, ByVal fdwFlags As Long, ByVal fdwOptions As Long, pPrinterNotifyOptions As Byte)
说明
创建一个新的改变通告对象，以便我们注意打印机状态的各种变化
返回值
Long，执行成功则返回改变通告对象的句柄。INVALID_HANDLE_VALUE 表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，一个已打开的打印机对象的句柄（用 OpenPrinter 获得）
fdwFlagsLong，来自 API32.TXT 文件的、带有 PRINTER_CHANGE_??前缀的某个常数，它们对要观察的对象进行了描述。如 pPrinterNotifyOptions 不为零，那么可将这个参数设为零
fdwOptionsLong，保留，设为零
pPrinterNotifyOptionsByte，指定一个缓冲区，其中包含了一个 PRINTER_NOTIFY_OPTIONS 结构。而这个结构又包含了指向一个或多个 PRINTER_NOTIFY_OPTIONS_TYPE 结构的指针。可将这个参数设为零（将声明方式改为 ByVal As Long 并传递零值），以便用 fdwFlags 字段指定想观察的变化
适用平台
Windows NT
其他
以下的声明是从 VB 的 API 文本查看器里复制的：
Declare Function FindFirstPrinterChangeNotification Lib "winspool.drv" Alias "FindFirstPrinterChangeNotification" (ByVal hPrinter As Long, ByVal fdwFlags As Long, ByVal fdwOptions As Long, ByVal pPrinterNotifyOptions As String) As Long

Top
FindNextPrinterChangeNotification
FindNextPrinterChangeNotification

VB 声明
Declare Function FindNextPrinterChangeNotification& Lib "winspool.dll" (ByVal hChange As Long, pdwChange As Long, ByVal pvReserved As Long, ByVal ppPrinterNotifyInfo As Long)
说明
用这个函数判断触发一次打印机改变通告信号的原因
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hChangeLong，一个打印机通告改变对象的句柄
pdwChangeLong，指定用于装载特定标志的一个 Long 型值，该标志标志着信号的来源。请在 API32.TXT 文件中寻找以 PRINTER_CHANGE_???前缀开头的常数
pvReservedLong，指定一个 PRINTER_NOTIFY_OPTIONS 结构的地址。如这个结构的 Flags

字段设为 PRINTER_NOTIFY_OPTIONS_REFRESH，那么 ppPrinterNotifyInfo 缓冲区就会载入正在监视的所有事件的状态——并不仅是那些触发了通告信号的事件。结构中所有其他字段会被忽略。可设为 NULL（零），表示只返回与状态改变有关信息
ppPrinterNotifyInfoLong，由系统分配的一个缓冲区的地址。完成后，应该用 FreePrinterNotifyInfo 函数将这个缓冲区删除。缓冲区内包含了一个 PRINTER_NOTIFY_INFO 结构，其后跟随一系列 PRINTER_NOTIFY_INFO_DATA 结构（具体数量由第一个结构决定）
适用平台
Windows NT
其他
在 VB 的 API 文本查看器中复制的声明如下：
Declare Function FindNextPrinterChangeNotification Lib "winspool.drv" Alias "FindNextPrinterChangeNotification" (ByVal hChange As Long, pdwChange As Long, ByVal pvReserved As String, ByVal ppPrinterNotifyInfo As Long) As Long

Top
FreePrinterNotifyInfo
FreePrinterNotifyInfo

VB 声明
Declare Function FreePrinterNotifyInfo Lib "winspool.drv" (ByVal addr As Long) As Long
说明
释放由 FindNextPrinterChangeNotification 函数分配的一个缓冲区
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
addrLong，指定由 FindNextPrinterChangeNotification 函数分配的一个系统缓冲区的地址
适用平台
Windows NT

Top
GetForm
GetForm

VB 声明
Declare Function GetForm Lib "winspool.drv" Alias "GetFormA" (ByVal hPrinter As Long, ByVal pFormName As String, ByVal Level As Long, pForm As Byte, ByVal cbBuf As Long, pcbNeeded As Long) As Long
说明
取得与指定表单有关的信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表

参数类型及说明
hPrinterLong，一个已打开的打印机的句柄（用 OpenPrinter 获得）
pFormNameString，想获取信息的一个表单的名字
LevelLong，设为 1
pFormByte，包含 FORM_INFO_1 结构的缓冲区
cbBufLong，pForm 缓冲区中的字符数量
pcbNeededLong，指向一个 Long 型变量的指针，该变量用于保存请求的缓冲区长度，或者实际读入的字节数量
适用平台
Windows NT

Top
GetJob
GetJob

VB 声明
Declare Function GetJob Lib "winspool.drv" Alias "GetJobA" (ByVal hPrinter As Long, ByVal JobId As Long, ByVal Level As Long, pJob As Byte, ByVal cdBuf As Long, pcbNeeded As Long) As Long
说明
获取与指定作业有关的信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，一个已打开的打印机的句柄（用 OpenPrinter 获得）
JobIdLong，作业编号
LevelLong，1 或 2
pJobByte，包含 JOB_INFO_1 或 JOB_INFO_2 结构的缓冲区，结构中包含了与打印作业有关的信息
cbBufLong，pJob 缓冲区中的字符数量
pcbNeededLong，指向一个 Long 型变量的指针，该变量用于保存请求的缓冲区长度，或者实际读入的字节数量

Top
GetPrinter
GetPrinter

VB 声明
Declare Function GetPrinter Lib "winspool.drv" Alias "GetPrinterA" (ByVal hPrinter As Long, ByVal Level As Long, pPrinter As Any, ByVal cbBuf As Long, pcbNeeded As Long) As Long
说明
取得与指定打印机有关的信息
返回值

Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，一个已打开的打印机的句柄（用 OpenPrinter 获得）
LevelLong，1，2，3（仅适用于 NT），4（仅适用于 NT），或者 5（仅适用于 Windows 95 和 NT 4.0）
pPrinterAny，包含 PRINTER_INFO_x 结构的缓冲区。x 代表级别
cbBufLong，pPrinterEnum 缓冲区中的字符数量
pcbNeededLong，指向一个 Long 型变量的指针，该变量用于保存请求的缓冲区长度，或者实际读入的字节数量
注解
只有在发出调用的应用程序有足够的权限时，PRINTER_INFO_x 结构中的一些字段才能够被读取。这种权限由系统当前的安全设置决定

Top
GetPrinterData
GetPrinterData

VB 声明
Declare Function GetPrinterData Lib "winspool.drv" Alias "GetPrinterDataA" (ByVal hPrinter As Long, ByVal pValueName As String, pType As Long, pData As Byte, ByVal nSize As Long, pcbNeeded As Long) As Long
说明
为打印机设置注册表配置信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，一个已打开的打印机的句柄（用 OpenPrinter 获得）
pValueNameString，欲设置的注册表值的名称
pTypeLong，指定数据类型。使用来自 API32.TXT 的、以 REG_??开头的一个常数
pDataByte，指定一个 Byte 数组以接收数据
nSizeLong，以字节表示的 pData 数组的长度
pcbNeededLong，指向一个 Long 型变量的指针，该变量用于保存请求的缓冲区长度，或者实际读入的字节数量

Top
GetPrinterDriver
GetPrinterDriver

VB 声明
Declare Function GetPrinterDriver Lib "winspool.drv" Alias "GetPrinterDriverA" (ByVal hPrinter As Long, ByVal pEnvironment As String, ByVal Level As Long, pDriverInfo As Byte, ByVal cdBuf As Long, pcbNeeded As Long) As Long

说明
针对指定的打印机，获取与打印机驱动程序有关的信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，一个已打开的打印机的句柄（用 OpenPrinter 获得）
pEnvironmentString，欲获取的驱动程序环境（如：Windows NT x86）。如设为 vbNullString，表示使用当前（本地）系统环境
LevelLong，1，2 或 3（仅适用于 Windows 95 和 NT 4.0）
pDriverInfoByte，载入一个 DRIVER_INFO_x 结构的缓冲区。其中的 x 代表级别（Level）设置
cbBufLong，pDriverInfo 缓冲区中的字符数量
pcbNeededLong，指向一个 Long 型变量的指针，该变量用于保存请求的缓冲区长度，或者实际读入的字节数量

Top
GetPrinterDriverDirectory
GetPrinterDriverDirectory

VB 声明
Declare Function GetPrinterDriverDirectory Lib "winspool.drv" Alias "GetPrinterDriverDirectoryA" (ByVal pName As String, ByVal pEnvironment As String, ByVal Level As Long, pDriverDirectory As Byte, ByVal cdBuf As Long, pcbNeeded As Long) As Long
说明
判断指定系统中包含了打印机驱动程序的目录是什么
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，服务器的名字。如设为 vbNullString，表示使用本地系统
pEnvironmentString，欲在其中获取目录的一个环境（如：Windows NT x86）。vbNullString 表示使用当前（本地）系统环境
LevelLong，设为 1
pDriverDirectoryByte，指定一个缓冲区，其中会载入打印机驱动程序目录的完整路径名。可定义成 ByVal As String，以便将字节数组分配给一个字串，从而避免执行 ANSI 到 Unicode 格式的转换
cbBufLong，pDriverDirectory 缓冲区中的字符数量
pcbNeededLong，指向一个 Long 型变量的指针，该变量用于保存请求的缓冲区长度，或者实际读入的字节数量

Top
GetPrintProcessorDirectory
GetPrintProcessorDirectory

VB 声明
Declare Function GetPrintProcessorDirectory Lib "winspool.drv" Alias "GetPrintProcessorDirectoryA" (ByVal pName As String, ByVal pEnvironment As String, ByVal Level As Long, ByVal pPrintProcessorInfo As String, ByVal cdBuf As Long, pcbNeeded As Long) As Long
说明
判断指定系统中包含了打印机处理器驱动程序及文件的目录
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pNameString，指定服务器的名字。设置成 vbNullString 则表示使用本地系统
pEnvironmentString，欲在其中获取目录的一个环境（如：Windows NT x86）。用 vbNullString 表示使用当前（本地）系统环境
LevelLong，设为 1
pPrintProcessorInfoString，指定一个缓冲区，其中载入打印机处理器目录的完整路径。可定义成 ByVal As String，以便将字节数组分配给一个字串，从而取消进行 ANSI 到 Unicode 转换的必要
cbBufLong，pPrintProcessorInfo 缓冲区中的字符数量
pcbNeededLong，指向一个 Long 型变量的指针，该变量用于保存请求的缓冲区长度，或者实际读入的字节数量

Top
OpenPrinter
OpenPrinter

VB 声明
Declare Function OpenPrinter Lib "winspool.drv" Alias "OpenPrinterA" (ByVal pPrinterName As String, phPrinter As Long, pDefault As PRINTER_DEFAULTS) As Long
说明
打开指定的打印机，并获取打印机的句柄
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
pPrinterNameString，要打开的打印机的名字
phPrinterLong，用于装载打印机的句柄
pDefaultPRINTER_DEFAULTS，这个结构保存要载入的打印机信息

Top
PrinterMessageBox
PrinterMessageBox

VB 声明
Declare Function PrinterMessageBox Lib "winspool.drv" Alias "PrinterMessageBoxA" (ByVal hPrinter As Long, ByVal error As Long, ByVal hwnd As Long, ByVal pText As String, ByVal pCaption As String, ByVal dwType As Long) As Long
说明
在拥有指定打印作业的系统上显示一个打印机出错消息框。如一名用户在远程登录，这种做法便相当有用
返回值
Long，IDOK， IDRETRY 或 IDCANCEL；由用户的输入决定（如消息框在远程系统显示，则肯定是 IDOK）
参数表
参数类型及说明
hPrinterLong，出现错误的打印机的句柄
errorLong，ERROR_OUT_OF_PAPER（缺纸）或 ERROR_NOT_READY（未就绪）
hwndLong，指定消息框的父窗口。可以为 NULL
pTextLong，欲显示的消息正文
pCaptionLong，消息框的标题
dwTypeLong，指定任何一个标准的 MessageBox 标志。建议使用 MB_ICONSTOP 或 MB_RETRYCANCEL 或 MB_SETFOREGROUND
适用平台
Windows NT

Top
PrinterProperties
PrinterProperties

VB 声明
Declare Function PrinterProperties Lib "winspool.drv" Alias "PrinterProperties" (ByVal hwnd As Long, ByVal hPrinter As Long) As Long
说明
启动打印机属性对话框，以便对打印机进行配置
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，对话框的父窗口
hPrinterLong，一个已打开的打印机的句柄
注解
如打印机打开的时候没有使用足够的访问权限，对话框的有些功能也许会禁止使用

Top
ReadPrinter
ReadPrinter

VB 声明
Declare Function ReadPrinter Lib "winspool.drv" Alias "ReadPrinter" (ByVal hPrinter As Long, pBuf As Any, ByVal cdBuf As Long, pNoBytesRead As Long) As Long
说明
从打印机读入数据
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，一个已打开的打印机的句柄（用 OpenPrinter 获得）
pBufAny，指定一个缓冲区或结构，用于装载来自打印机的数据
cdBufLong，欲读入的缓冲区大小或字节数
pNoBytesReadLong，用于装载实际读取字节数的一个变量
注解
为使这个函数正常使用，端口必须是双向的

Top
ResetDC
ResetDC

VB 声明
Declare Function ResetDC Lib "gdi32" Alias "ResetDCA" (ByVal hdc As Long, lpInitData As DEVMODE) As Long
说明
根据提供的 DEVMODE 结构，对一个设备场景进行重设。这样便允许我们在打印期间改变打印机的配置。利用这个函数，可将文档中的某个页改为横向打印。可试着用 DocumentProperties 函数取得一个设备的默认 DEVMODE 结构
返回值
Long，执行成功则返回设备场景的句柄，零表示失败
参数表
参数类型及说明
hdcLong，设备场景的句柄
lpInitDataDEVMODE，指定一个缓冲区的第一个字节。该缓冲区包含了用于那个设备的一个有效 DEVMODE 结构。记住在这个缓冲区中包括设备专用的数据区
注解
这个函数可成功用于由 VB 的 Printer 对象的 hDC 属性返回的设备场景上
注意一定要正确设置 lpdm 的 dmFields 字段
这个函数在 StartPage 和 EndPage 之间会被禁用——即只能在页与页之间调用这个函数，不能在页内调用
驱动程序、设备和输出端口不可以用这个函数更改

Top
ResetPrinter
ResetPrinter

VB 声明
Declare Function ResetPrinter Lib "winspool.drv" Alias "ResetPrinterA" (ByVal hPrinter As Long, pDefault As PRINTER_DEFAULTS) As Long
说明
改变指定打印机的默认数据类型及文档设置
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，欲修改的一台打印机的句柄
pDefaultPRINTER_DEFAULTS，定义了打印机新设置的一个结构。参考 OpenPrinter 函数的说明，了解这个结构进一步的细节。结构中的 DesiredAccess 字段会被忽略
适用平台
Windows NT

Top
ScheduleJob
ScheduleJob

VB 声明
Declare Function ScheduleJob Lib "winspool.drv" Alias "ScheduleJob" (ByVal hPrinter As Long, ByVal JobId As Long) As Long
说明
提交一个要打印的作业
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，一台已打开的打印机句柄
JobIdLong，以前用 AddJob 函数获得的作业编号
注解
参考 AddJob 函数以获得进一步的信息

Top
SetAbortProc
SetAbortProc

VB 声明
Declare Function SetAbortProc Lib "gdi32" Alias "SetAbortProc" (ByVal hDC As Long, ByVal lpAbortProc As Long) As Long
说明
我们可以为 Windows 提供一个特殊的函数，令其在扩展打印操作过程中调用。这个函数叫“取消函数”。其结果告诉 Windows 是继续打印操作，还是立即取消

SetAbortProc 函数的作用是为 windows 指定取消函数的地址。由于 VB 不支持函数地址的概念，所以要使用特定的通用回调定制控件，否则就不能使用这个函数
返回值
Long，如结果大于零，表示执行成功；SP_ERROR 表示出错。会设置 GetLastError
参数表
参数类型及说明
hDCLong，一个设备场景的句柄。
lpAbortProcLong，一个取消函数的地址
在 VB 里使用
如随同 VB 的打印机对象使用这个函数，就可能干扰正常的 VB 打印机制。这个函数的确可以在 VB 环境中使用，但有可能造成打印机出错。因此，在下一次使用 Printer.NewPage 方法的时候，有必要用适当的机制捕获这种错误。如果在自己的程序中为 VB 的 Printer 对象设置了一个取消函数，那么建议您完整测试代码
如果在自己创建的一个设备场景中打印，那么这个函数的使用没有丝毫问题

Top
SetForm
SetForm

VB 声明
Declare Function SetForm& Lib "spoolss.dll" Alias "SetFormA" (ByVal hPrinter As Long, ByVal pFormName As String, ByVal Level As Long, pForm As Byte)
说明
为指定的表单设置信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，指定一个打开打印机的句柄（用 OpenPrinter 取得）
pFormNameString，欲设置的表单的名字
LevelLong，设为 1
pFormByte，包含一个有效 FORM_INFO_1 结构的缓冲区
适用平台
Windows NT
注解
请参考 AddForm 函数。

Top
SetJob
SetJob

VB 声明
Declare Function SetJob Lib "winspool.drv" Alias "SetJobA" (ByVal hPrinter As Long, ByVal JobId As Long, ByVal Level As Long, pJob As Byte, ByVal Command As Long) As Long

说明
对一个打印作业的状态进行控制
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，指定一个打开打印机的句柄（用 OpenPrinter 取得）
JobIdLong，要修改的作业的编号
LevelLong，0，1 或 2
pJobByte，指定一个缓冲区。如级别（Level）设为 1 或 2，那该缓冲区就包含了一个 JOB_INFO_1 或 JOB_INFO_2 结构。如级别为 0，缓冲区为 NULL（变成 ByVal As Long，以便传递零值）。如指定了一个结构，则来自那个结构的信息会用于改变打印作业的设置（除 JobId，pPrinterName，pMachineName，pDriverName，Size，Submitte 以及 Time 字段外）
CommandLong，下述常数之一：
JOB_CONTROL_CANCEL 取消作业
JOB_CONTROL_PAUSE 暂停作业
JOB_CONTROL_RESTART 重新启动一个已开始打印的作业
JOB_CONTROL_RESUME 恢复一个暂停的作业

Top
SetPrinter
SetPrinter

VB 声明
Declare Function SetPrinter Lib "winspool.drv" Alias "SetPrinterA" (ByVal hPrinter As Long, ByVal Level As Long, pPrinter As Byte, ByVal Command As Long) As Long
说明
对一台打印机的状态进行控制
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，指定一个已打开的打印机的句柄（用 OpenPrinter 取得）
LevelLong，0，2 或 3（4 或 5 用于 windows95，5 或 6 用于 NT 4.0）。如 Command 不是零，则这个参数必须是零
pPrinterByte，包含一个 PRINTER_INFO_x 的结构的缓冲区，其中的 x 代表级别的设定（Level）。假如级别为零，并且 Command 设为 PRINTER_CONTROL_SET_STATUS，那缓冲区就包含了一个 PRINTER_CONTROL_STATUS 结构。否则，如级别为零，就设为 NULL（要把声明变成 ByVal As Long，以便传递零值）
CommandLong，下述值之一：
零根据 PRINTER_INFO_x 结构改变打印机
PRINTER_CONTROL_PAUSE 暂停打印机
PRINTER_CONTROL_PURGE 删除打印机的所有作业
PRINTER_CONTROL_RESUME 恢复一台暂停的打印机

PRINTER_CONTROL_SET_STATUS 载入打印机的 PRINTER_CONTROL_STATUS 结构（不可在 NT 3.51 下使用）
注解
在 PRINTER_INFO_2 结构的基础上设置打印机状态时，pServerName，AveragePPM，Status 和 cJobs 字段都会被忽略

Top
SetPrinterData
SetPrinterData

VB 声明
Declare Function SetPrinterData Lib "winspool.drv" Alias "SetPrinterDataA" (ByVal hPrinter As Long, ByVal pValueName As String, ByVal dwType As Long, pData As Byte, ByVal cbData As Long) As Long
说明
设置打印机的注册表配置信息
返回值
Long，ERROR_SUCCESS 表示成功，一个错误值表示失败。
参数表
参数类型及说明
hPrinterLong，指定一个已打开的打印机的句柄（用 OpenPrinter 取得）
pValueName String，欲设置的注册表值名
dwTypeLong，定义数据的类型，使用来自 API32.txt 文件的、以 REG_起头的一个常数
pDataByte，指定一个缓冲区的第一个条目，缓冲区中包含了要设置的适当的数据类型
cbDataLong，缓冲区 pData 的长度

Top
StartDoc
StartDoc

VB 声明
Declare Function StartDoc Lib "gdi32" Alias "StartDocA" (ByVal hdc As Long, lpdi As DOCINFO) As Long
说明
开始一个打印作业
返回值
Long，如执行成功，返回文档的作业编号，常数 SP_ERROR 失败。会设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景的句柄
lpdiDOCINFO，对文档定义的一个结构

Top
StartDocPrinter

StartDocPrinter

VB 声明
Declare Function StartDocPrinter Lib "winspool.drv" Alias "StartDocPrinterA" (ByVal hPrinter As Long, ByVal Level As Long, pDocInfo As Byte) As Long
说明
在后台打印的级别启动一个新文档
返回值
Long，非零表示成功，零表示失败。
参数表
参数类型及说明
hPrinterLong，指定一个已打开的打印机的句柄（用 OpenPrinter 取得）
LevelLong，1 或 2（仅用于 win95）
pDocInfoByte，包含一个 DOC_INFO_1 或 DOC_INFO_2 结构的缓冲区
注解
在应用程序的级别并非特别有用。后台打印程序用它标识一个文档的开始

Top
StartPage
StartPage

VB 声明
Declare Function StartPage Lib "gdi32" Alias "StartPage" (ByVal hdc As Long) As Long
说明
打印一个新页前要先调用这个函数
返回值
Long，非零表示成功，零表示失败
参数表
参数类型及说明
hdcLong，一个设备场景的句柄
在 VB 里使用
操作 VB 的打印机对象时，注意不要使用这个函数

Top
StartPagePrinter
StartPagePrinter

VB 声明
Declare Function StartPagePrinter Lib "winspool.drv" Alias "StartPagePrinter" (ByVal hPrinter As Long) As Long
说明
在打印作业中指定一个新页的开始
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError

参数表
参数类型及说明
hPrinterLong，指定一个已打开的打印机的句柄（用 OpenPrinter 取得）
注解
在应用程序的级别并非特别有用。后台打印程序用它在实际的打印机数据中标识一个文档的开始

Top
WritePrinter
WritePrinter

VB 声明
Declare Function WritePrinter Lib "winspool.drv" Alias "WritePrinter" (ByVal hPrinter As Long, pBuf As Any, ByVal cdBuf As Long, pcWritten As Long) As Long
说明
将发送目录中的数据写入打印机
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hPrinterLong，指定一个已打开的打印机的句柄（用 OpenPrinter 取得）
pBufAny，包含了要写入打印机的数据的一个缓冲区或结构
cdBufLong，pBuf 缓冲区的长度
pcWrittenLong，指定一个 Long 型变量，用于装载实际写入的字节数

Top
设备场景函数
设备场景函数，共五页。第一页，第二页，第三页，第四页，第五页
CombineRgn 将两个区域组合为一个新区域
CombineTransform 驱动世界转换。它相当于依顺序进行两次转换
CreateCompatibleDC 创建一个与特定设备场景一致的内存设备场景
CreateDC 为专门设备创建设备场景
CreateEllipticRgn 创建一个椭圆
CreateEllipticRgnIndirect 创建一个内切于特定矩形的椭圆区域
CreateIC 为专用设备创建一个信息场景
CreatePolygonRgn 创建一个由一系列点围成的区域
CreatePolyPolygonRgn 创建由多个多边形构成的区域。每个多边形都应是封闭的
CreateRectRgn 创建一个矩形区域
CreateRectRgnIndirect 创建一个矩形区域
CreateRoundRectRgn 创建一个圆角矩形
DeleteDC 删除专用设备场景或信息场景，释放所有相关窗口资源
DPtoLP 将点阵从设备坐标转换到专用设备场景逻辑坐标
EqualRgn 确定两个区域是否相等
ExcludeClipRect 从专用设备场景的剪裁区中去掉一个矩形区。矩形内不能进行绘图

AddFontResource
AddFontResource

VB 声明
Declare Function AddFontResource Lib "gdi32" Alias "AddFontResourceA" (ByVal lpFileName As String) As Long
说明
在 Windows 系统中添加一种字体资源。添加完毕后，该字体即可由任何 Windows 应用程序调用
返回值
Long，添加的字体数量，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpFileNameString，字体资源文件的文件名。可以是.FON， .FNT， .TTF 或 .FOT 文件
注解
添加了一种资源后必须调用下述 API 函数：
di% = SendMessageBynum(HWND_BROADCAST, WM_FONTCHANGE, x, y)
其中，HWND_BROADCAST、WM_FONTCHANGE 使用来自 API32.TXT 文件的值。这样便可告诉所有 Windows 应用程序字体列表已发生了变化
示例
Call AddFontResource("myfont.ttf")
Call SendMessage(HWND_BROADCAST, WM_FONTCHANGE, 0, 0)

Top
CombineRgn
CombineRgn

VB 声明
Declare Function CombineRgn Lib "gdi32" Alias "CombineRgn" (ByVal hDestRgn As Long, ByVal hSrcRgn1 As Long, ByVal hSrcRgn2 As Long, ByVal nCombineMode As Long) As Long
说明
将两个区域组合为一个新区域
返回值
Long，下列常数之一：
COMPLEXREGION：区域有互相交叠的边界
SIMPLEREGION：区域边界没有互相交叠
NULLREGION：区域为空
ERRORAPI：不能创建组合区域
参数表
参数类型及说明
hDestRgnLong，包含组合结果的区域句柄
hSrcRgn1Long，源区域 1
hSrcRgn2Long，源区域 2
nCombineModeLong，组合两区域的方法。可设为下述常数

RGN_ANDhDestRgn 被设置为两个源区域的交集
RGN_COPYhDestRgn 被设置为 hSrcRgn1 的拷贝
RGN_DIFFhDestRgn 被设置为 hSrcRgn1 中与 hSrcRgn2 不相交的部分
RGN_ORhDestRgn 被设置为两个区域的并集
RGN_XORhDestRgn 被设置为除两个源区域 OR 之外的部分

Top
CombineTransform
CombineTransform

VB 声明
Declare Function CombineTransform Lib "gdi32" Alias "CombineTransform" (lpxformResult As xform, lpxform1 As xform, lpxform2 As xform) As Long
说明
驱动世界转换。它相当于依顺序进行两次转换
返回值
Long，执行成功为 TRUE（非零），失败则为零
参数表
参数类型及说明
lpxformResultxform，保存转换结果的结构
lpxform1xform，按顺序的第一个结构
xformxform，按顺序的第二个结构
适用平台
Windows NT

Top
CreateCompatibleDC
CreateCompatibleDC

VB 声明
Declare Function CreateCompatibleDC Lib "gdi32" Alias "CreateCompatibleDC" (ByVal hdc As Long) As Long
说明
创建一个与特定设备场景一致的内存设备场景
返回值
Long，新设备场景句柄，若出错则为零
参数表
参数类型及说明
hdcLong，设备场景句柄。新的设备场景将与它一致。也可能为 0 以创建一个与屏幕一致的设备场景
注解
在绘制之前，先要为该设备场景选定一个位图。不再需要时，该设备场景可用 DeleteDC 函数删除。删除前，其所有对象应回复初始状态

Top
CreateDC
CreateDC, CreateDCBynum

VB 声明
Declare Function CreateDC& Lib "gdi32" Alias "CreateDCA" (ByVal lpDriverName As String, ByVal lpDeviceName As String, ByVal lpOutput As String, lpInitData As DEVMODE)
Declare Function CreateDCBynum& Lib "gdi32" Alias "CreateDCA" (ByVal lpDriverName As String, ByVal lpDeviceName As String, ByVal lpOutput As String, lpInitData As Long)
说明
为专门设备创建设备场景
返回值
Long，新设备场景句柄，若出错则为零
参数表
参数类型及说明
lpDriverNameString，用 vbNullString 传递 null 值给该参数，除非：1、用 DISPLAY，是获取整个屏幕的设备场景；2、用 WINSPOOL，则是访问打印驱动
lpDeviceNameString，所用专门设备的名称。该名由打印管理器分配显示
lpOutputString，用 vbNullString 传递 null 值给该参数
lpInitDataDEVMODE，这个结构保存初始值。用 CreateDCBynum 传递 0（NULL）值则适用默认设置
注解
在绘制之前，先要为该设备场景选定一个位图。不再需要时，该设备场景可用 DeleteDC 函数删除。删除前，其所有对象应回复初始状态。若有设备初始设置可用 DocumentProperties API 函数载入 DEVMODE 结构。使用屏幕设备场景（DISPLAY）时要小心，因为它会干扰其他应用程序的外观
示例：靠近屏幕左上角画一个矩形
dc& = CreateDCBynum("DISPLAY", vbNullString, vbNullString, 0)
dl& = Rectangle(dc&, 5, 5, 100, 100)

Top
CreateEllipticRgn
CreateEllipticRgn

VB 声明
Declare Function CreateEllipticRgn Lib "gdi32" Alias "CreateEllipticRgn" (ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long) As Long
说明
创建一个椭圆，该椭圆与 X1，Y1 和 X2，Y2 坐标点确定的矩形内切
返回值
Long，执行成功则为区域句柄，失败则为零
参数表
参数类型及说明
X1,Y1Long，矩形左上角 X，Y 坐标

X2,Y2Long，矩形右下角 X，Y 坐标
注解
不用时一定要用 DeleteObject 函数删除区域。用 Ellipse API 函数绘出的椭圆与该椭圆区域不完全相同，因为本函数的绘图计算不包括矩形的下边和右边

Top
CreateEllipticRgnIndirect
CreateEllipticRgnIndirect

VB 声明
Declare Function CreateEllipticRgnIndirect Lib "gdi32" Alias "CreateEllipticRgnIndirect" (lpRect As Rect) As Long
说明
创建一个内切于特定矩形的椭圆区域
返回值
Long，执行成功则返回区域句柄，失败则为零
参数表
参数类型及说明
lpRectLong，定义要创建的椭圆区域尺寸的矩形
注解
不用时一定要用 DeleteObject 函数删除该区域

Top
CreateFont
CreateFont

VB 声明
Declare Function CreateFont Lib "gdi32" Alias "CreateFontA" (ByVal H As Long, ByVal W As Long, ByVal E As Long, ByVal O As Long, ByVal W As Long, ByVal I As Long, ByVal u As Long, ByVal S As Long, ByVal C As Long, ByVal OP As Long, ByVal CP As Long, ByVal Q As Long, ByVal PAF As Long, ByVal F As String) As Long
说明
用指定的属性创建一种逻辑字体
返回值
Long，执行成功则返回逻辑字体的句柄，零表示失败。会设置 GetLastError
参数表
参数类型及说明
HLong，IfHeight
WLong，IfWidth
ELong，IfEscapement
OLong，IfOrientation
WLong，IfWeight
ILong，IfItalic
uLong，IfUnderline

SLong，IfStrikeOut
CLong，IfCharSet
OPLong，IfOutputPrecision
CPLong，IfClipPrecision
QLong，IfQuality
PAFLong，IfPitchAndFamily
FString，IfFaceName
注解
VB 的字体属性在选择字体的时候显得更有效

Top
CreateFontIndirect
CreateFontIndirect

VB 声明
Declare Function CreateFontIndirect Lib "gdi32" Alias "CreateFontIndirectA" (lpLogFont As LOGFONT) As Long
说明
用指定的属性创建一种逻辑字体
返回值
Long，执行成功则返回逻辑字体句柄，零表示失败
参数表
参数类型及说明
lpLogFontLOGFONT，这个结构定义了逻辑字体请求的属性
注解
VB 的字体属性在选择字体的时候显得更有效

Top
CreateIC
CreateIC

VB 声明
Declare Function CreateIC Lib "gdi32" Alias "CreateICA" (ByVal lpDriverName As String, ByVal lpDeviceName As String, ByVal lpOutput As String, lpInitData As DEVMODE) As Long
说明
为专用设备创建一个信息场景。信息场景可用来快速获取某设备的信息而无须创建设备场景这样的系统开销。它可作为参数传递给 GetDeviceCaps 一类的信息函数以替代设备场景参数
返回值
Long，执行成功为信息场景句柄，失败则为零
参数表
参数类型及说明
lpDriverNameString，用 vbNullString 传递 null 值给该参数，除非：1、用 DISPLAY，是获取整个屏幕的设备场景；2、用 WINSPOOL，则是访问打印驱动
lpDeviceNameString，所用专门设备的名称。该名由打印管理器分配显示

lpOutputString，用 vbNullString 传递 null 值给该参数
lpInitDataDEVMODE，这个结构保存初始值
注解
Long，不用时一定要用 DeleteDC 函数删除设备场景。进一步的说明参考 CreateDC 函数
示例：为一个名为“Color Stylus”的打印机取回信息场景
dc& = CreateICBynum("WINSPOOL", "Color Stylus", vbNullString,0)

Top
CreatePolygonRgn
CreatePolygonRgn

VB 声明
Declare Function CreatePolygonRgn Lib "gdi32" Alias "CreatePolygonRgn" (lpPoint As POINTAPI, ByVal nCount As Long, ByVal nPolyFillMode As Long) As Long
说明
创建一个由一系列点围成的区域。windows 在需要时自动将最后点与第一点相连以封闭多边形
返回值
Long，执行成功为创建的区域句柄，失败则为 0
参数表
参数类型及说明
lpPointPOINTAPI，nCount 个 POINTAPI 结构中的第一个 POINTAPI 结构
nCountLong，多边形的点数
nPolyFillModeLong，描述多边形填充模式。可为 ALTERNATE 或 WINDING 常数。参考 SetPolyFillMode 函数对多边形填充模式的解释
注解
不用时一定要用 DeleteObject 函数删除该区域

Top
CreatePolyPolygonRgn
CreatePolyPolygonRgn

VB 声明
Declare Function CreatePolyPolygonRgn Lib "gdi32" Alias "CreatePolyPolygonRgn" (lpPoint As POINTAPI, lpPolyCounts As Long, ByVal nCount As Long, ByVal nPolyFillMode As Long) As Long
说明
创建由多个多边形构成的区域。每个多边形都应是封闭的
返回值
Long，执行成功则为创建区域的句柄，失败则为零
参数表
参数类型及说明
lpPointPOINTAPI，nCount 个 POINTAPI 结构中的第一个 POINTAPI 结构
lpPolyCountsLong，长整数阵列的第一个入口。每个入口包含构成一个封闭多边形的点数。

lpPoint 阵列组成了一系列多边形，每个多边形在 lpPolyCounts 中有一个入口
nCountLong，多边形的点数
nPolyFillModeLong，描述多边形填充模式。可为 ALTERNATE 或 WINDING 常数。参考 SetPolyFillMode 函数对多边形填充模式的解释
注解
不用时一定要用 DeleteObject 函数删除该区域

Top
CreateRectRgn
CreateRectRgn

VB 声明
Declare Function CreateRectRgn Lib "gdi32" Alias "CreateRectRgn" (ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long) As Long
说明
创建一个由点 X1，Y1 和 X2，Y2 描述的矩形区域
返回值
Long，执行成功为区域句柄，失败则为零
参数表
参数类型及说明
X1,Y1Long，矩形左上角 X，Y 坐标
X2,Y2Long，矩形右下角 X，Y 坐标
注解
不用时一定要用 DeleteObject 函数删除该区域
这个矩形的下边和右边不包含在区域之内

Top
CreateRectRgnIndirect
CreateRectRgnIndirect

VB 声明
Declare Function CreateRectRgnIndirect Lib "gdi32" Alias "CreateRectRgnIndirect" (lpRect As) As Long
说明
创建一个由 lpRect 确定的矩形区域
返回值
Long，执行成功则为区域句柄，失败则为 0
参数表
参数类型及说明
lpRectRECT，要用来创建区域的矩形
注解
参考 CreateRectRgn 的注解

Top

CreateRoundRectRgn
CreateRoundRectRgn

VB 声明
Declare Function CreateRoundRectRgn Lib "gdi32" Alias "CreateRoundRectRgn" (ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long, ByVal X3 As Long, ByVal Y3 As Long) As Long
说明
创建一个圆角矩形，该矩形由 X1，Y1-X2，Y2 确定，并由 X3，Y3 确定的椭圆描述圆角弧度
返回值
Long，执行成功则为区域句柄，失败则为 0
参数表
参数类型及说明
X1,Y1Long，矩形左上角的 X，Y 坐标
X2,Y2Long，矩形右下角的 X，Y 坐标
X3Long，圆角椭圆的宽。其范围从 0（没有圆角）到矩形宽（全圆）
Y3Long，圆角椭圆的高。其范围从 0（没有圆角）到矩形高（全圆）
注解
不用时一定要用 DeleteObject 函数删除该区域
用该函数创建的区域与用 RoundRect API 函数画的圆角矩形不完全相同，因为本矩形的右边和下边不包括在区域之内

Top
CreateScalableFontResource
CreateScalableFontResource

VB 声明
Declare Function CreateScalableFontResource Lib "gdi32" Alias "CreateScalableFontResourceA" (ByVal fHidden As Long, ByVal lpszResourceFile As String, ByVal lpszFontFile As String, ByVal lpszCurrentPath As String) As Long
说明
为一种 TureType 字体创建一个资源文件，以便能用 API 函数 AddFontResource 将其加入 Windows 系统。字体信息本身并不复制到字体资源文件中；相反，资源文件中包含了欲使用的 TureType 文件的名字
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
fHiddenLong，如果是零，表示创建一个普通的字体资源；如果是 1，表示创建一个只读字体资源，它只能在文档中嵌入使用
lpszResourceFileString，欲创建的资源文件的名字。普通文件使用.FOT 扩展名，只读文件使用.FOR 扩展名
lpszFontFileString，TureType 字体文件文件的文件名。如果其中包含了一个路径，就到指定

的路径寻找字体文件，同时不使用 lpszCurrentPath 参数指定的位置。而且在调用 AddFontResource 函数之前，会将字体复制到 Windows 的 SYSTEM 目录
lpszCurrentPathString，由 lpszFontFile 参数决定

Top
DeleteDC
DeleteDC

VB 声明
Declare Function DeleteDC Lib "gdi32" Alias "DeleteDC" (ByVal hdc As Long) As Long
说明
删除专用设备场景或信息场景，释放所有相关窗口资源。不要将它用于 GetDC 函数取回的设备场景
返回值
Long，执行成功则为非零，失败则为零
参数表
参数类型及说明
hdcLong，将要删除的设备场景
注解
若有对象被选入设备场景，则在调用本函数前应将它们选出。为此，可将初始对象回选入 DC，也可用 SaveDC， RestoreDC 函数对回复 DC 为其创建时的状态
在 vb 里使用
不要将它用于由 vb hdc 属性获取的设备场景句柄

Top
DPtoLP
DPtoLP

VB 声明
Declare Function DPtoLP Lib "gdi32" Alias "DPtoLP" (ByVal hdc As Long, lpPoint As POINTAPI, ByVal nCount As Long) As Long
说明
将点阵从设备坐标转换到专用设备场景逻辑坐标
返回值
Long，执行成功为非零值，失败则为零
参数表
参数类型及说明
hdcLong，确定逻辑坐标系统的设备场景句柄
lpPointPOINTAPI，包含有设备坐标点的一个或多个 POINTAPI 结构的第一入口。每个入口都将被转换为逻辑坐标（若为世界转换则转换为世界坐标）
nCountLong，lpPoint 阵列中的入口数

Top
DrawText

DrawText

VB 声明
Declare Function DrawText Lib "user32" Alias "DrawTextA" (ByVal hdc As Long, ByVal lpStr As String, ByVal nCount As Long, lpRect As RECT, ByVal wFormat As Long) As Long
说明
将文本描绘到指定的矩形中
返回值
Long，描绘文字的高度
参数表
参数类型及说明
hdcLong，欲在其中显示文字的一个设备场景的句柄
lpStrString，欲描绘的文本字串
nCountLong，欲描绘的字符数量。如果要描绘整个字串（直到空中止符），则可将这个参数设为-1
lpRectRECT，指定用于绘图的一个格式化矩形（采用逻辑坐标）
wFormatLong，一个标志位数组，决定了以何种形式执行绘图。参考下面总结的常数类型列表
标志常数说明
DT_BOTTOM 必须同时指定 DT_SINGLE。指示文本对齐格式化矩形的底边
DT_CALCRECT 象下面这样计算格式化矩形：多行绘图时矩形的底边根据需要进行延展，以便容下所有文字；单行绘图时，延展矩形的右侧。不描绘文字。由 lpRect 参数指定的矩形会载入计算出来的值
DT_CENTER 文本垂直居中
DT_EXPANDTABS 描绘文字的时候，对制表站进行扩展。默认的制表站间距是 8 个字符。但是，可用 DT_TABSTOP 标志改变这项设定
DT_EXTERNALLEADING 计算文本行高度的时候，使用当前字体的外部间距属性（the external leading attribute）
DT_LEFT 文本左对齐
DT_NOCLIP 描绘文字时不剪切到指定的矩形
DT_NOPREFIX 通常，函数认为 & 字符表示应为下一个字符加上下划线。该标志禁止这种行为
DT_RIGHT 文本右对齐
DT_SINGLELINE 只画单行
DT_TABSTOP 指定新的制表站间距，采用这个整数的高 8 位
DT_TOP 必须同时指定 DT_SINGLE。指示文本对齐格式化矩形的顶部
DT_VCENTER 必须同时指定 DT_SINGLE。指示文本对齐格式化矩形的中部
DT_WORDBREAK 进行自动换行。如用 SetTextAlign 函数设置了 TA_UPDATECP 标志，这里的设置则无效

Top
DrawTextEx
DrawTextEx

VB 声明
Declare Function DrawTextEx Lib "user32" Alias "DrawTextExA" (ByVal hDC As Long, ByVal lpsz As String, ByVal n As Long, lpRect As RECT, ByVal un As Long, lpDrawTextParams As DRAWTEXTPARAMS) As Long
说明
与 DrawText 相似，只是加入了更多的功能
返回值
Long，描绘文字的高度
参数表
参数类型及说明
hDCLong，要在其中绘图的一个设备场景的句柄
lpszString，欲描绘的文本字串
nLong，欲描绘的字符数量。如果要描绘整个字串（直到空中止符），则可将这个参数设为-1
lpRectRECT，指定用于绘图的一个格式化矩形（采用逻辑坐标）
unLong，一个标志位。决定了以何种形式执行绘图。参考 DrawText 的 wFormat 参数和下表。其中下表列出的是新增的常数
标志常数说明
DT_EDITCONTROL 对一个多行编辑控件进行模拟。不显示部分可见的行
DT_ENDELLIPSES 倘若字串不能在矩形里全部容下，就在末尾显示省略号
DT_PATHELLIPSES 如字串包含了 \ 字符，就用省略号替换字串内容，使其能在矩形中全部容下。例如，一个很长的路径名可能换成这样显示——c:\windows\...\doc\readme.txt
DT_MODIFYSTRING 如指定了 DT_ENDELLIPSES 或 DT_PATHELLIPSES，就会对字串进行修改，使其与实际显示的字串相符
DT_RTLREADING 如选入设备场景的字体属于希伯来或阿拉伯语系，就从右到左描绘文字
lpDrawTextParamsDRAWTEXTPARAMS，这个结构包含了附加的绘图参数

Top
EnumFontFamilies
EnumFontFamilies

VB 声明
Declare Function EnumFontFamilies Lib "gdi32" Alias "EnumFontFamiliesA" (ByVal hdc As Long, ByVal lpszFamily As String, ByVal lpEnumFontFamProc As Long, ByVal lParam As Long) As Long
说明
列举指定设备可用的字体
返回值
Long，由回调函数返回的前一个值
参数表
参数类型及说明
hdcLong，设备场景的句柄
lpszFamilyString，欲枚举的字体家族。如指定 vbNullString，可枚举出每种可用字体家族中的一种字体
lpEnumFontFamProcLong，欲调用的函数地址。这个地址是用 AddressOf 运算符为来自一个

标准模块的函数进行操作，或者利用某个回调控件得到
lParamLong，指定希望传递给回调函数的一个用户自定义值
注解
这个函数取代了 API 函数 EnumFonts，因为它能对 TureType 字体样式说明进行控制
只有实际存在的字体才会列举出来，那些可由 GDI 合成的字体不会列出

Top
EnumFontFamiliesEx
EnumFontFamiliesEx

VB 声明
Declare Function EnumFontFamiliesEx Lib "gdi32" Alias "EnumFontFamiliesExA" (ByVal hdc As Long, lpLogFont As LOGFONT, ByVal lpEnumFontProc As Long, ByVal lParam As Long, ByVal dw As Long) As Long
说明
根据一个 LOGFONT 结构提供的信息，列举指定设备可用的字体
返回值
Long，由回调函数返回的前一个值
参数表
参数类型及说明
hdcLong，设备场景的句柄
lpLogFontLOGFONT，这个结构指定了欲枚举的字体。此时用到的字段包括：lfCharSet，lfFaceName 和 lfPitchAndFamily。其他所有字段都会忽略
lpEnumFontFamProcLong，欲调用的函数地址。这个地址是用 AddressOf 运算符为来自一个
标准模块的函数进行操作，或者利用某个回调控件得到
lParamLong，指定希望传递给回调函数的一个用户自定义值
dwLong，保留，设为零
注解
参见 EnumFontFamilies 函数的注解

Top
EqualRgn
EqualRgn

VB 声明
Declare Function EqualRgn Lib "gdi32" Alias "EqualRgn" (ByVal hSrcRgn1 As Long, ByVal hSrcRgn2 As Long) As Long
说明
确定两个区域是否相等
返回值
Long，若两区域相等为非零值。若不等为 0。若有一个区域无效则返回 ERRORAPI
参数表
参数类型及说明
hSrcRgn1Long，一个区域的句柄

hSrcRgn2Long，区域句柄

Top
ExcludeClipRect
ExcludeClipRect

VB 声明
Declare Function ExcludeClipRect Lib "gdi32" Alias "ExcludeClipRect" (ByVal hdc As Long, ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long) As Long
说明
从专用设备场景的剪裁区中去掉一个由点 X1，Y1 和 X2，Y2 确定的矩形区。矩形内不能进行绘图
返回值
Long，返回以下常数之一以描述所得剪裁区：
COMPLEXREGION：区域边界互相交叠
SIMPLEREGION：区域边界没有互相交叠
NULLREGION：区域为空
ERRORAPI：发生了错误
参数表
参数类型及说明
hdcLong，要修改的设备场景
X1,Y1Long，矩形（逻辑坐标）左上角的 X，Y 坐标
X2,Y2Long，矩形（逻辑坐标）右下角的 X，Y 坐标，矩形的右边和下边不会从剪裁区中移走

Top
ExcludeUpdateRgn
ExcludeUpdateRgn

VB 声明
Declare Function ExcludeUpdateRgn Lib "user32" Alias "ExcludeUpdateRgn" (ByVal hdc As Long, ByVal hwnd As Long) As Long
说明
从专用设备场景剪裁区去掉指定窗口的刷新区域。这防止对无效区绘图（该无效区将推迟一点再刷新）
返回值
Long，见 ExcludeClipRect 函数返回值
参数表
参数类型及说明
hdcLong，设备场景，其剪裁区将被修改以排除 hwnd 窗口的刷新区
hwndLong，窗口句柄
在 vb 中使用
设您要完成一个较长或复杂的绘图操作，该函数可允许您避免画到现在没有遮盖而随后即将画的区域。这可提高性能并消除闪烁效果

Top
ExtCreateRegion
ExtCreateRegion

VB 声明
Declare Function ExtCreateRegion Lib "gdi32" Alias "ExtCreateRegion" (lpXform As xform, ByVal nCount As Long, lpRgnData As RGNDATA) As Long
说明
根据世界转换修改区域
返回值
Long，执行成功为已转换区域的句柄，失败则为 0
参数表
参数类型及说明
lpXformxform，将用于由 lpRgnData 指定区域的转换
nCountLong，lpRgnData 数据结构或缓冲区的字节数
lpRgnDataRGNDATA，用 GetRegionData 定义区域时载入的一个结构
注解
在 win95 下，只有缩放和平移转换才能用该函数。它不支持旋转和剪切

Top
ExtSelectClipRgn
ExtSelectClipRgn

VB 声明
Declare Function ExtSelectClipRgn Lib "gdi32" Alias "ExtSelectClipRgn" (ByVal hdc As Long, ByVal hRgn As Long, ByVal fnMode As Long) As Long
说明
将指定区域组合到设备场景的当前剪裁区
返回值
Long，返回下列常数之一，以描述所得剪裁区：
COMPLEXREGION：区域边界互相交叠
SIMPLEREGION：区域边界没有互相交叠
NULLREGION：区域为空
ERRORAPI：发生了错误
参数表
参数类型及说明
hdcLong，剪裁区将被修改的设备场景的句柄
hRgnLong，源区域句柄。若 fnMode 为 RGN_COPY，它可为 NULL
fnModeLong，下列常数之一：
RGN_AND 新剪裁区包括即属 hRgn 又属当前剪裁区的部分
RGN_COPY 新剪裁区为 hRgn 区域
RGN_DIFF 新剪裁区为当前剪裁区除掉其在 hRgn 区内的部分
RGN_OR 新剪裁区包括属 hRgn 或当前属当前剪裁区的部分

RGN_XOR 新剪裁区包括属 hRgn 或当前属当前剪裁区的部分，但要除去同属两者的部分
注解
本函数对 hRgn 区域没有影响，执行后可毁去（destroy）该区域

Top
FillRgn
FillRgn

VB 声明
Declare Function FillRgn Lib "gdi32" Alias "FillRgn" (ByVal hdc As Long, ByVal hRgn As Long, ByVal hBrush As Long) As Long
说明
用指定刷子填充指定区域
返回值
Long，执行成功为非零值，失败则为 0
参数表
参数类型及说明
hdcLong，设备场景句柄
hRgnLong，以数据设备坐标填充的区域句柄
hBrushLong，要用的刷子的句柄

Top
FrameRgn
FrameRgn

VB 声明
Declare Function FrameRgn Lib "gdi32" Alias "FrameRgn" (ByVal hdc As Long, ByVal hRgn As Long, ByVal hBrush As Long, ByVal nWidth As Long, ByVal nHeight As Long) As Long
说明
用指定刷子围绕指定区域画一个外框
返回值
Long，执行成功返回非零值，失败为 0
参数表
参数类型及说明
hdcLong，设备场景句柄
hRgnLong，将数据设备坐标填充的区域句柄
hBrushLong，将用的刷子句柄
nWidthLong，垂直边框宽度（以设备单元为单位）
nHeightLong，水平边框高度（以设备单元为单位）

Top
GetBoundsRect
GetBoundsRect

VB 声明
Declare Function GetBoundsRect Lib "gdi32" Alias "GetBoundsRect" (ByVal hdc As Long, lprcBounds As RECT, ByVal flags As Long) As Long
说明
获取指定设备场景的边界矩形。每个设备场景都有一个边界矩形，程序员可用它来堆放表示当前图象边界的信息
返回值
Long，出错为 0，否则为下列常数之一：
DCB_SET：边界矩形非空
DCB_RESET：边界矩形为空
DCB_ENABLE：边界矩形正被堆放
DCB_DISABLE：边界矩形当前没有被堆放
参数表
参数类型及说明
hdcLong，边界矩形对应的设备场景
lprcBoundsRECT，装载设备场景 hdc 的当前边界矩形
flagsLong，可设为常数 DCB_RESET 以清除边界矩形，否则设为零
注解
详情参考 SetBoundsRect

Top
GetClipBox
GetClipBox

VB 声明
Declare Function GetClipBox Lib "gdi32" Alias "GetClipBox" (ByVal hdc As Long, lpRect As RECT) As Long
说明
获取完全包含指定设备场景剪裁区的最小矩形
返回值
Long，返回下列常数之一，以描述所得剪裁区：
COMPLEXREGION：区域边界互相交叠
SIMPLEREGION：区域边界没有互相交叠
NULLREGION：区域为空
ERRORAPI：发生了错误
参数表
参数类型及说明
hdcLong，设备场景句柄
lpRectRECT，装载包含设备场景剪裁区的矩形

Top
GetClipRgn
GetClipRgn

VB 声明
Declare Function GetClipRgn Lib "gdi32" Alias "GetClipRgn" (ByVal hdc As Long, ByVal hRgn As Long) As Long
说明
获取设备场景当前剪裁区
返回值
Long，下列常数之一：
1——执行成功，剪裁区存在
0——执行成功，没有定义剪裁区
-1——出错
参数表
参数类型及说明
hdcLong，想要访问的设备场景剪裁区
hRgnLong，已存在的区域句柄。该区域将被设为设备场景当前剪裁区
注解
hRgn 装载剪裁区拷贝。它后面的改变不影响剪裁区。该函数只能载入由应用程序设置的剪裁区，那些由 BeginPaint 一类函数设置的内部剪裁区不能被函数访问

Top
GetDC
GetDC

VB 声明
Declare Function GetDC Lib "user32" Alias "GetDC" (ByVal hwnd As Long) As Long
说明
获取指定窗口的设备场景
返回值
Long，指定窗口的设备场景句柄，出错则为 0
参数表
参数类型及说明
hwndLong，将获取其设备场景的窗口的句柄。若为 0，则要获取整个屏幕的 DC
注解
若窗口所属类具有 CS_OWNDC， CS_CLASSDC 或 CS_PARENTDC 样式，则获取的设备场景属窗口或类专有。vb 的窗体和图片框控件也是这种情况，它用该函数取得的结果和控件的 hdc 属性相同（在 autoredraw 为 FALSE 时）。您无须考虑取回的窗体或图片框控件设备场景的默认状态，特别是绘图对象。另外，默认状态随着窗体和控件 autoredraw 属性的设置而不同。在设备场景释放前您必须回复其状态为初始值。对于没有 CS_OWNDC，CS_CLASSDC 或 CS_PARENTDC 样式的窗口的设备场景，可从通用 windows 缓存中获取，其状态为默认值。缓存中可用设备场景数量是有限的，因此只要可能就释放设备场景
用本函数获取的设备场景一定要用 ReleaseDC 函数释放，不能用 DeleteDC

Top
GetDCEx
GetDCEx

VB 声明
Declare Function GetDCEx Lib "user32" Alias "GetDCEx" (ByVal hwnd As Long, ByVal hrgnclip As Long, ByVal fdwOptions As Long) As Long
说明
为指定窗口获取设备场景。相比 GetDC，本函数提供了更多的选项
返回值
Long，执行成功为指定窗口设备场景句柄。出错则为 0
参数表
参数类型及说明
hwndLong，窗口句柄
hrgnclipLong，窗口剪裁区
fdwOptionsLong，标志字。根据下列常数设置各位：
DCX_CACHE 不管窗口类的样式，从 windows 缓存获取设备场景
DCX_CLIPCHILREN 所有可见的子窗口区都要从 DC 的剪裁区中排除
DCX_CLIPSIBLINGS 窗口 hWnd 上的所有可见兄弟窗口都要从 DC 的剪裁区中排除
DCX_EXCLUDERGN 从 DC 剪裁区中排除由 hrgnclip 指定的区域
DCX_EXCLUDEUPDATE 从设备场景剪裁区中排除刷新区域
DCX_INTERSECTRGN 由 hrgnclip 指定的区域与设备场景剪裁区相交
DCX_INTERSECTUPDATE 指定区域与设备场景刷新区域相交
DCX_LOCKWINDOWUPDATE 该标志为允许向窗口绘图，即使它由于 LockWindowUpdate 的调用被锁住
DCX_NORESETATTRS 设备场景释放后不被重置为默认状态
DCX_PARENTCLIP 放弃 CS_PARENTDC 类样式设置。DC 的起点设为 hWnd 窗口的左上角
DCX_WINDOWA device context is returned for the entire window rectangle rather than just the client area of the window
DCX_VALIDATECombine with DCX_INTERSECTUPDATE, validates the clipping region
注解
若窗口所属类具有 CS_OWNDC， CS_CLASSDC 或 CS_PARENTDC 样式，则获取的设备场景属窗口或类专有。这时，设备场景状态不能从初值修改。vb 的窗体和控件通常是这种情况。否则，置 DCX_CACHE 位以从通用 windows 缓冲区恢复设备场景。若不置该位，则函数返回 0。DC 的状态位默认设置。从缓存获取的设备场景用过后要用 ReleaseDC 函数释放以防止系统死锁，因为 windows 只有 5 个缓存 DC 可用
其他情况参见 GetDC 函数注解

Top
GetDCOrgEx
GetDCOrgEx

VB 声明
Declare Function GetDCOrgEx Lib "gdi32" Alias "GetDCOrgEx" (ByVal hdc As Long, lpPoint As POINTAPI) As Long
说明
获取指定设备场景起点位置（以屏幕坐标表示）。例如，设 DC 起点为窗口客户区左上角，

则该函数返回值为该角在屏幕中的位置（以屏幕像素坐标表示）
返回值
Long，执行成功为非零值，否则为零
参数表
参数类型及说明
hdcLong，设备场景句柄
lpPointPOINTAPI，装载设备场景起点的屏幕坐标
示例：在窗体模块中使用时，该代码装载窗体客户区左上角坐标（以屏幕坐标表示）。它也是窗体 hDC 属性的起点

```
Dim dl&
Dim pt As POINTAPI
dl& = GetDCOrgEx(hDC, pt)
```

Top
GetDeviceCaps
GetDeviceCaps

VB 声明

```
Declare Function GetDeviceCaps Lib "gdi32" Alias "GetDeviceCaps" (ByVal hdc As Long, ByVal nIndex As Long) As Long
```

说明
根据指定设备场景代表的设备的功能返回信息
返回值
Long，参见 GetDeviceCaps 索引表
参数表
参数类型及说明
hdcLong，要查询其设备的信息的设备场景
nIndexLong，根据 GetDeviceCaps 索引表所示常数确定返回信息的类型

Top
GetGraphicsMode
GetGraphicsMode

VB 声明

```
Declare Function GetGraphicsMode Lib "gdi32" Alias "GetGraphicsMode" (ByVal hdc As Long) As Long
```

说明
确定是否允许增强图形模式（世界转换）
返回值
Long，下列常数之一：
GM_COMPATIBLE：图形模式兼容 win16
GM_ADVANCED：允许世界转换
参数表
参数类型及说明

hdcLong，其模式将被测试的设备场景
注解
只有 Windows NT 支持世界转换

Top
GetMapMode
GetMapMode

VB 声明
Declare Function GetMapMode Lib "gdi32" Alias "GetMapMode" (ByVal hdc As Long) As Long
说明
为特定设备场景调入映象模式
返回值
Long，当前设备场景的映象模式。关于映象模式的说明参见 SetMapMode 函数
参数表
参数类型及说明
hdcLong，其模式将被测试的设备场景

Top
GetRegionData
GetRegionData

VB 声明
Declare Function GetRegionData Lib "gdi32" Alias "GetRegionDataA" (ByVal hRgn As Long, ByVal dwCount As Long, lpRgnData As RgnData) As Long
说明
装入描述一个区域信息的 RgnData 结构或缓冲区
返回值
Long，如果结构足够大以装入区域的数据，返回 1；出错时返回 0。如果 lpRgnData 不够大，不能装入区域数据，则返回需要的结构大小
参数表
参数类型及说明
hRgnLong，包含信息的区域的句柄
dwCountLong，RgnData 结构的大小
lpRgnDataRgnData，这个结构用以装入区域信息
注解
RgnData 是一个描述区域的定长结构。Buffer 是存放区域数据的缓冲区。缓冲区实际需要的大小取决于区域的复杂程度（显然，1 字节是永远不够的）。有两个选择：
1、将 RgnData 重定义为永远不会用到的一个大尺寸。这是需要的，因为 vb 不允许动态重定义结构的大小
2、分配一个字节数组并用它来代替 RgnData 结构。这要求将 As RgnData 换为 As Byte 来改变函数的 API 声明，并且传送字节数组的第一个元素
如果以后要访问 RGNDATAHEADER 结构的元素，需要用一个内存拷贝例程将数据从缓冲区拷贝到一个特别定义的 RGNDATAHEADER 结构中

Top
GetRgnBox
GetRgnBox

VB 声明
Declare Function GetRgnBox Lib "gdi32" Alias "GetRgnBox" (ByVal hRgn As Long, lpRect As RECT) As Long
说明
获取完全包含指定区域的最小矩形
返回值
Long，下列常数之一，以描述当前剪裁区：
COMPLEXREGION：该区域有互相交叠的边界
SIMPLEREGION：该区域边界没有互相交叠
NULLREGION：区域为空
ERRORAPI：发生了错误
参数表
参数类型及说明
hRgnLong，区域句柄
lpRectRECT，矩形结构，装载完全包含指定区域的矩形

Top
GetUpdateRgn
GetUpdateRgn

VB 声明
Declare Function GetUpdateRgn Lib "user32" Alias "GetUpdateRgn" (ByVal hwnd As Long, ByVal hRgn As Long, ByVal fErase As Long) As Long
说明
确定指定窗口的刷新区域。该区域当前无效，需要刷新
返回值
Long，下列常数之一，以描述当前剪裁区：
COMPLEXREGION：该区域有互相交叠的边界
SIMPLEREGION：该区域边界没有互相交叠
NULLREGION：区域为空
ERRORAPI：发生了错误
参数表
参数类型及说明
hwndLong，将确定刷新区域的窗口的句柄
hRgnLong，装载 hwnd 窗口刷新区域的区域句柄
fEraseLong，为非零值表示窗口背景应擦除，客户区外的窗口部分是被重画
原文：TRUE (nonzero) to specify that the window background should be erased and parts of the window outside of the client area should be redrawn.

Top
GetViewportExtEx
GetViewportExtEx

VB 声明
Declare Function GetViewportExtEx Lib "gdi32" Alias "GetViewportExtEx" (ByVal hdc As Long, lpSize As SIZE) As Long
说明
获取设备场景视口（viewport）范围
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hdcLong，设备场景句柄
lpSizeSIZE，装载 DC 视口水平和垂直范围（以设备单元表示）的结构

Top
GetViewportOrgEx
GetViewportOrgEx

VB 声明
Declare Function GetViewportOrgEx Lib "gdi32" Alias "GetViewportOrgEx" (ByVal hdc As Long, lpPoint As POINTAPI) As Long
说明
获取设备场景视口起点
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hdcLong，设备场景句柄
lpPointPOINTAPI，装载视口起点的结构

Top
GetWindowDC
GetWindowDC

VB 声明
Declare Function GetWindowDC Lib "user32" Alias "GetWindowDC" (ByVal hwnd As Long) As Long
说明
获取整个窗口（包括边框、滚动条、标题栏、菜单等）的设备场景
返回值
Long，执行成功为窗口设备场景，失败则为 0
参数表

参数类型及说明
hwndLong，将获取其设备场景的窗口
注解
不推荐在 vb 里使用这个函数。用完后一定要用 ReleaseDC 函数释放场景

Top
GetWindowExtEx
GetWindowExtEx

VB 声明
Declare Function GetWindowExtEx Lib "gdi32" Alias "GetWindowExtEx" (ByVal hdc As Long, lpSize As SIZE) As Long
说明
获取指定设备场景的窗口范围
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hdcLong，设备场景句柄
lpSizeSIZE，装载设备场景逻辑窗口水平和垂直范围（以逻辑单元表示）的结构

Top
GetWindowOrgEx
GetWindowOrgEx

VB 声明
Declare Function GetWindowOrgEx Lib "gdi32" Alias "GetWindowOrgEx" (ByVal hdc As Long, lpPoint As POINTAPI) As Long
说明
获取指定设备场景的逻辑窗口的起点
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hdcLong，设备场景句柄
lpPointPOINTAPI，装载逻辑窗口起点的结构

Top
GetWindowRgn
GetWindowRgn

VB 声明
Declare Function GetWindowRgn Lib "user32" Alias "GetWindowRgn" (ByVal hWnd As Long, ByVal hRgn As Long) As Long

说明
根据以前 SetWindowRgn 函数的定义获取窗口区域
返回值
Long，下列常数之一，以描述当前剪裁区：
COMPLEXREGION：该区域有互相交叠的边界
SIMPLEREGION：该区域边界没有互相交叠
NULLREGION：区域为空
ERRORAPI：发生了错误
参数表
参数类型及说明
hWndLong，要获取区域的窗口
hRgnLong，区域句柄，若已设置有一个区域，则装载当前窗口区域的拷贝
注解
参考 SetWindowRgn 函数

Top
GetWorldTransform
GetWorldTransform

VB 声明
Declare Function GetWorldTransform Lib "gdi32" Alias "GetWorldTransform" (ByVal hdc As Long, lpXform As xform) As Long
说明
如果有世界转换，为设备场景获取当前世界转换
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hdcLong，设备场景句柄
lpXformxform，装载设备场景当前世界转换的结构
适用平台
Windows NT

Top
IntersectClipRect
IntersectClipRect

VB 声明
Declare Function IntersectClipRect Lib "gdi32" Alias "IntersectClipRect" (ByVal hdc As Long, ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long) As Long
说明
为指定设备定义一个新的剪裁区，该区为当前剪裁区与由点 X1，Y1 和 X2，Y2 定义的矩形的交集
返回值

Long，下列常数之一，以描述当前剪裁区：
COMPLEXREGION：该区域有互相交叠的边界
SIMPLEREGION：该区域边界没有互相交叠
NULLREGION：区域为空
ERRORAPI：发生了错误
参数表
参数类型及说明
hdcLong，设备场景
X1,Y1Long，矩形左上角 X，Y 坐标
X2,Y2Long，矩形右下角 X，Y 坐标

Top
InvalidateRgn
InvalidateRgn

VB 声明
Declare Function InvalidateRgn Lib "user32" Alias "InvalidateRgn" (ByVal hwnd As Long, ByVal hRgn As Long, ByVal bErase As Long) As Long
说明
使窗口指定区域不活动，并将它加入窗口刷新区，使之可随后被重画
返回值
Long，TRUE
参数表
参数类型及说明
hwndLong，窗口句柄
hRgnLong，不活动区域句柄，该区域以窗口客户坐标定义。若该句柄为 0，则将使整个窗口客户区不活动
bEraseLong，为非零值则表示要在刷新前擦除
注解
若 bErase 为 TRUE，则刷新前整个刷新区都将被擦除而不只是定义的区域

Top
InvertRgn
InvertRgn

VB 声明
Declare Function InvertRgn Lib "gdi32" Alias "InvertRgn" (ByVal hdc As Long, ByVal hRgn As Long) As Long
说明
通过颠倒每个像素值反转设备场景指定区域
返回值
Long，执行成功为非零值，失败为零
参数表
参数类型及说明

hdcLong，设备场景句柄
hRgnLong，将反转的设备区域

Top
LPtoDP
LPtoDP

VB 声明
Declare Function LPtoDP Lib "gdi32" Alias "LPtoDP" (ByVal hdc As Long, lpPoint As POINTAPI, ByVal nCount As Long) As Long
说明
将点阵从指定设备场景逻辑坐标转换为设备坐标
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hdcLong，定义了逻辑坐标系统的设备场景句柄
lpPointPOINTAPI，包含逻辑坐标的点的一个或多个 POINTAPI 结构的第一入口，每个入口将被转换为设备坐标
nCountLong，lpPoint 阵列中的入口数
注解
若存在世界转换，lpPoint 阵列中的点则是以世界坐标表示的

Top
ModifyWorldTransform
ModifyWorldTransform

VB 声明
Declare Function ModifyWorldTransform Lib "gdi32" Alias "ModifyWorldTransform" (ByVal hdc As Long, lpXform As xform, ByVal iMode As Long) As Long
说明
根据指定的模式修改世界转换
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hdcLong，设备场景，其世界坐标将被修改
lpXformxform，这个结构内含根据 iMode 参数附加的转换
iModeLong，下列常数之一：
MWT_IDENTITY 设置世界转换为默认的特性转换（用该特性转换时，世界转换无效），忽略 lpXform 参数
MWT_LEFTMULTIPLYlpXform 转换被当前转换乘，乘积设为新的世界转换
MWT_RIGHTMULTIPLY 当前转换被 lpXform 转换乘，乘积设为新的世界转换
注解

与一般的乘法不同，矩阵乘法的结果与矩阵顺序有关。在 MWT_LEFTMULTIPLY 模式，lpXform 为左边的操作数。请参考高等代数相关书籍。
调用本函数前先应调用 SetGraphicsMode 函数设置 GM_ADVANCED 图形模式
适用平台
Windows NT

Top
OffsetClipRgn
OffsetClipRgn

VB 声明
Declare Function OffsetClipRgn Lib "gdi32" Alias "OffsetClipRgn" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long) As Long
说明
按指定量平移设备场景剪裁区
返回值
Long，下列常数之一，以描述当前剪裁区：
COMPLEXREGION：该区域有互相交叠的边界
SIMPLEREGION：该区域边界没有互相交叠
NULLREGION：区域为空
ERRORAPI：发生了错误
参数表
参数类型及说明
hdcLong，设备场景
xLong，以逻辑单元表示的水平偏移
yLong，以逻辑单元表示的垂直偏移

Top
OffsetRgn
OffsetRgn

VB 声明
Declare Function OffsetRgn Lib "gdi32" Alias "OffsetRgn" (ByVal hRgn As Long, ByVal x As Long, ByVal y As Long) As Long
说明
按指定偏移量平移指定区域
返回值
Long，下列常数之一，以描述当前剪裁区：
COMPLEXREGION：该区域有互相交叠的边界
SIMPLEREGION：该区域边界没有互相交叠
NULLREGION：区域为空
ERRORAPI：发生了错误
参数表
参数类型及说明

hRgnLong，区域句柄
xLong，以逻辑坐标表示的水平偏移量
yLong，以逻辑坐标表示的垂直偏移量

Top
OffsetViewportOrgEx
OffsetViewportOrgEx

VB 声明
Declare Function OffsetViewportOrgEx Lib "gdi32" Alias "OffsetViewportOrgEx" (ByVal hdc As Long, ByVal nX As Long, ByVal nY As Long, lpPoint As POINTAPI) As Long
说明
平移设备场景视口区域
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hdcLong，设备场景句柄
nX,nYLong，将加到视口起点的水平和垂直偏移量（以设备坐标表示）
lpPointPOINTAPI，装载设备场景视口原先的起点的结构，以设备坐标表示

Top
OffsetWindowOrgEx
OffsetWindowOrgEx

VB 声明
Declare Function OffsetWindowOrgEx Lib "gdi32" Alias "OffsetWindowOrgEx" (ByVal hdc As Long, ByVal nX As Long, ByVal nY As Long, lpPoint As POINTAPI) As Long
说明
平移指定设备场景窗口起点
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hdcLong，设备场景句柄
nX,nYLong，以逻辑坐标表示的窗口起点
lpPointPOINTAPI，装载 DC 窗口原有的起点的的结构，以逻辑坐标表示

Top
PaintRgn
PaintRgn

VB 声明
Declare Function PaintRgn Lib "gdi32" Alias "PaintRgn" (ByVal hdc As Long, ByVal hRgn As

Long) As Long
说明
用当前刷子背景色填充指定区域
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hdcLong，设备场景句柄
hRgnLong，将填充的区域句柄

Top
PtInRegion
PtInRegion

VB 声明
Declare Function PtInRegion Lib "gdi32" Alias "PtInRegion" (ByVal hRgn As Long, ByVal x As Long, ByVal y As Long) As Long
说明
确定点是否在指定区域内
返回值
Long，若点在区域内为非零值，否则为 0
参数表
参数类型及说明
hRgnLong，区域句柄
xLong，以逻辑坐标表示的点的 X 坐标
yLong，以逻辑坐标表示的点的 Y 坐标
在 vb 里使用
在测试复杂图象时非常有用

Top
PtVisible
PtVisible

VB 声明
Declare Function PtVisible Lib "gdi32" Alias "PtVisible" (ByVal hdc As Long, ByVal x As Long, ByVal y As Long) As Long
说明
确定指定点是否可见（即，点是否在设备场景剪裁区内）
返回值
Long，若点在指定设备场景剪裁区内为非零值，否则为 0
参数表
参数类型及说明
hdcLong，设备场景句柄
xLong，要检查的点的 x,y 坐标

y

Top
RectInRegion
RectInRegion

VB 声明
Declare Function RectInRegion Lib "gdi32" Alias "RectInRegion" (ByVal hRgn As Long, lpRect As RECT) As Long
说明
确定矩形是否有部分在指定区域内
返回值
Long，若矩形有部分在指定区域内为非零值，否则为 0
参数表
参数类型及说明
hRgnLong，区域句柄
lpRectRECT，要测试的矩形结构
注解
矩形的底边和右边不包括在 lpRect 内

Top
RectVisible
RectVisible

VB 声明
Declare Function RectVisible Lib "gdi32" Alias "RectVisible" (ByVal hdc As Long, lpRect As RECT) As Long
说明
确定指定矩形是否有部分可见（是否在设备场景剪裁区内）
返回值
Long，矩形有部分在指定设备场景剪裁区内为非零值，否则为零
参数表
参数类型及说明
hdcLong，设备场景句柄
lpRectRECT，用于测试是否可见的矩形（用逻辑坐标）
注解
矩形的底边和右边不包括在 lpRect 内

Top
ReleaseDC
ReleaseDC

VB 声明
Declare Function ReleaseDC Lib "user32" Alias "ReleaseDC" (ByVal hwnd As Long, ByVal hdc

As Long) As Long
说明
释放由调用 GetDC 或 GetWindowDC 函数获取的指定设备场景。它对类或私有设备场景无效（但这样的调用不会造成损害）
返回值
Long，执行成功为 1，否则为 0
参数表
参数类型及说明
hwndLong，要释放的设备场景相关的窗口句柄
hdcLong，要释放的设备场景句柄
注解
对那些用 CreateDC 一类的 DC 创建函数生成的设备场景，不要用本函数

Top
RestoreDC
RestoreDC

VB 声明
Declare Function RestoreDC Lib "gdi32" Alias "RestoreDC" (ByVal hdc As Long, ByVal nSavedDC As Long) As Long
说明
从设备场景堆栈恢复一个原先保存的设备场景
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hdcLong，将恢复的设备场景句柄
nSavedDCLong，将恢复的设备场景 ID 号，它由 SaveDC 函数返回。若为-1，则恢复最近的设备场景
注解
若 nSavedDC 所指的不是栈顶的 DC，则栈中 nSavedDC 之上的所有设备场景都要从栈中移出，详情参见 SaveDC 函数

Top
SaveDC
SaveDC

VB 声明
Declare Function SaveDC Lib "gdi32" Alias "SaveDC" (ByVal hdc As Long) As Long
说明
将指定设备场景状态保存到 Windows 设备场景堆栈。DC 的映射模式、窗口和视口缩放、剪裁区、所选对象列表都要保存。本函数通常在要对设备场景作临时改动时使用，DC 属性可用 RestoreDC 函数恢复
返回值

Long，一个代表该设备场景的整数，据此可用 RestoreDC 函数恢复。出错则为 0
参数表
参数类型及说明
hdcLong，要保存的设备场景句柄
注解
设备场景被保存到内部堆栈，这意味着可多次保存一个设备场景状态。例如，您可在进入某个函数时保存设备场景，在其中完成某些 api 调用，再保存一次，作更多的调用后，恢复前一状态，再作其他 api 函数调用，最后退出该函数前恢复原有设备场景状态。可保存的设备场景个数仅受限于内存大小

Top
ScaleViewportExtEx
ScaleViewportExtEx

VB 声明
Declare Function ScaleViewportExtEx Lib "gdi32" Alias "ScaleViewportExtEx" (ByVal hdc As Long, ByVal nXnum As Long, ByVal nXdenom As Long, ByVal nYnum As Long, ByVal nYdenom As Long, lpSize As SIZE) As Long
说明
缩放设备场景视口的范围
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hdcLong，设备场景句柄
nXnum,nYnumLong，当前 X，Y 范围将与本数值相乘
nXdenom,nYdenomLong，当前 X，Y 范围将被本数值除
lpSizeSIZE，装载原有水平和垂直视口范围的结构
注解
本函数仅在 MM_ISOTROPIC 和 MM_ANISOTROPIC 映射模式下有效，在 MM_ISOTROPIC 模式，要在设置视口范围之前设置窗口范围

Top
ScaleWindowExtEx
ScaleWindowExtEx

VB 声明
Declare Function ScaleWindowExtEx Lib "gdi32" Alias "ScaleWindowExtEx" (ByVal hdc As Long, ByVal nXnum As Long, ByVal nXdenom As Long, ByVal nYnum As Long, ByVal nYdenom As Long, lpSize As SIZE) As Long
说明
缩放指定设备场景窗口范围
返回值
Long，执行成功为非零值，失败为 0

参数表
参数类型及说明
hdcLong，设备场景句柄
nXnum,nYnumLong，当前 X，Y 范围将与本数值相乘
nXdenom,nYdenomLong，当前 X，Y 范围将被本数值除
lpSizeSIZE，装载原有水平和垂直视口范围的结构
注解
本函数仅在 MM_ISOTROPIC 和 MM_ANISOTROPIC 映射模式下有效，在 MM_ISOTROPIC 模式，要在设置视口范围之前设置窗口范围

Top
ScrollDC
ScrollDC

VB 声明
Declare Function ScrollDC Lib "user32" Alias "ScrollDC" (ByVal hdc As Long, ByVal dx As Long, ByVal dy As Long, lprcScroll As RECT, lprcClip As RECT, ByVal hrgnUpdate As Long, lprcUpdate As RECT) As Long
说明
在窗口（由设备场景代表）中水平和（或）垂直滚动矩形
返回值
Long，执行成功为非零值，失败为 0，并设置 GetLastError
参数表
参数类型及说明
hdcLong，设备场景句柄
dxLong，将滚动的水平单位
dyLong，将滚动的垂直单位
lprcScrollRECT，要滚动的矩形
lprcClipRECT，定义剪裁区的矩形，滚动只能在该区内进行
hrgnUpdateLong，该区域设置为客户坐标系统中不被滚动操作覆盖的区，也可为 0，则表示没有设置刷新区域
lprcUpdateRECT，该矩形为客户坐标系统中不被滚动操作覆盖的区，也可为 0（要求修改 vb 声明使之能接收 lprcUpdate 为长整型参数）
注解
若 hrgnUpdate 和 lprcUpdate 都为 NULL，windows 将不能为该 DC 计算刷新区

Top
SelectClipRgn
SelectClipRgn

VB 声明
Declare Function SelectClipRgn Lib "gdi32" Alias "SelectClipRgn" (ByVal hdc As Long, ByVal hRgn As Long) As Long
说明

为指定设备场景选择新的剪裁区
返回值
Long，下列常数之一，以描述当前剪裁区：
COMPLEXREGION：该区域有互相交叠的边界
SIMPLEREGION：该区域边界没有互相交叠
NULLREGION：区域为空
ERRORAPI：发生了错误（保留原有剪裁区）
参数表
参数类型及说明
hdcLong，将设置新剪裁区的设备场景
hRgnLong，将为设备场景设置剪裁区的句柄，该区域使用设备坐标
注解
该函数拷贝剪裁区；这样同一剪裁区可用于多个设备场景，删除它不会影响每个设备场景的剪裁区。某些打印机文本和图形有不同的坐标系统，这时，应选用精度高的坐标系统——例如，点阵打印机用抖动算法，其文本精度比图形精度高，则应选用文本坐标系统
若使用 WM_PAINT 消息的子类，处理该消息使您用本函数增加的剪裁区不能超过窗口的不活动区

Top
SetBoundsRect
SetBoundsRect

VB 声明
Declare Function SetBoundsRect Lib "gdi32" Alias "SetBoundsRect" (ByVal hdc As Long, lprcBounds As RECT, ByVal flags As Long) As Long
说明
设置指定设备场景的边界矩形。每个设备场景都有一个边界矩形，编程者可用来堆放表示当前图象边界的信息。当允许边界矩形堆放时，每次 windows 向设备场景绘图，绘图位置都与边界矩形比较，若有必要就要扩展边界矩形以包括绘图位置。这提供了一种方法可确定设备场景的哪些部分已画了
返回值
Long，调用本函数前的边界设置：
DCB_SET：边界矩形不为空
DCB_RESET：边界矩形为空
DCB_ENABLE：边界矩形已被堆放
DCB_DISABLE：边界矩形没被堆放
参数表
参数类型及说明
hdcLong，边界矩形对应的设备场景
lprcBoundsRECT，用于设置边界矩形的矩形
flagsLong，标志集合，可用下列常数组合而成
DCB_ACCUMULATE 新的边界矩形由当前边界矩形和 lprcBounds 联合而成
DCB_DISABLE 关闭边界堆放，该值为默认值
DCB_ENABLE 打开边界堆放

DCB_RESET 清除原有边界矩形，用 DCB_ACCUMULATE 设置新的边界矩形为 lprcBounds

Top
SetGraphicsMode
SetGraphicsMode

VB 声明
Declare Function SetGraphicsMode Lib "gdi32" Alias "SetGraphicsMode" (ByVal hdc As Long, ByVal iMode As Long) As Long
说明
允许或禁止增强图形模式，以提供某些支持（包括世界转换）
返回值
Long，执行成功为原来的图形模式，失败为 0
参数表
参数类型及说明
hdcLong，将为其设置模式的设备场景
iModeLong，下列值之一：
GM_COMPATIBLE：设置 win16 兼容模式
GM_ADVANCED：允许增强图形模式，包括世界转换
注解
只有 Windows NT 支持世界转换
若您已进入增强图形模式并已设置有世界转换，则您在将图形模式改回 GM_COMPATIBLE 之前，必须用 SetWorldTransform 或 ModifyWorldTransform 函数将世界转换改回默认的特性转换。关于特性转换更多的信息参考 ModifyWorldTransform

Top
SetMapMode
SetMapMode

VB 声明
Declare Function SetMapMode Lib "gdi32" Alias "SetMapMode" (ByVal hdc As Long, ByVal nMapMode As Long) As Long
说明
设置指定设备场景的映射模式
返回值
Long，执行成功为设备场景原来的映射模式，失败为 0
参数表
参数类型及说明
hdcLong，设备场景句柄
nMapModeLong，下列常数之一：
MM_ANISOTROPIC 视口和窗口范围可完全任意
MM_HIENGLISH 逻辑单元为 0.001 inch，起点在左下角
MM_HIMETRIC 逻辑单元为 0.01 millimeter，起点在左下角
MM_ISOTROPIC 视口和窗口范围任意，只是 x 和 y 逻辑单元尺寸要相同

MM_LOENGLISH 逻辑单元为 0.01 inch，起点在左下角
MM_HIMETRIC 逻辑单元为 0.1 millimeter，起点在左下角
MM_TEXT 逻辑单元为一个像素
MM_TWIPS 逻辑单元为 1 twip (1/1440 inch)，起点在左下角

Top
SetRectRgn
SetRectRgn

VB 声明
Declare Function SetRectRgn Lib "gdi32" Alias "SetRectRgn" (ByVal hRgn As Long, ByVal X1 As Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long) As Long
说明
设置区域为 X1，Y1 和 X2，Y2 描述的矩形
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hRgnLong，该区域将被设置为指定矩形
X1,Y1Long，矩形左上角 X，Y 坐标
X2,Y2Long，矩形右下角 X，Y 坐标
注解
本函数与 CreateRectRgn 相同，只是它是设置一个已存在区域而不是创建一个新区域
矩形的底和右边不包括在区域内

Top
SetViewportExtEx
SetViewportExtEx

VB 声明
Declare Function SetViewportExtEx Lib "gdi32" Alias "SetViewportExtEx" (ByVal hdc As Long, ByVal nX As Long, ByVal nY As Long, lpSize As SIZE) As Long
说明
设置设备场景视口范围
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hdcLong，设备场景句柄
nX,nYLong，视口水平和垂直范围
lpSizeSIZE，装载 DC 视口原来的水平和垂直范围（以设备单元表示）的结构
注解
本函数只在 MM_ISOTROPIC 和 MM_ANISOTROPIC 映射模式下有效。在 MM_ISOTROPIC 模式下，设置视口范围必须在窗口范围之前

Top
SetViewportOrgEx
SetViewportOrgEx

VB 声明
Declare Function SetViewportOrgEx Lib "gdi32" Alias "SetViewportOrgEx" (ByVal hdc As Long, ByVal nX As Long, ByVal nY As Long, lpPoint As POINTAPI) As Long
说明
设置设备场景视口起点
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hdcLong，设备场景句柄
nX,nYLong，以设备坐标表示的视口起点
lpPointPOINTAPI，装载 DC 视口原来的起点的结构

Top
SetWindowExtEx
SetWindowExtEx

VB 声明
Declare Function SetWindowExtEx Lib "gdi32" Alias "SetWindowExtEx" (ByVal hdc As Long, ByVal nX As Long, ByVal nY As Long, lpSize As SIZE) As Long
说明
设置指定设备场景窗口范围
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hdcLong，设备场景句柄
nX,nYLong，窗口水平和垂直范围
lpSizeSIZE，装载设备场景原来的水平和垂直窗口范围（以逻辑单元表示）的结构
注解
本函数只在 MM_ISOTROPIC 和 MM_ANISOTROPIC 映射模式下有效。在 MM_ISOTROPIC 模式下，设置视口范围必须在窗口范围之前

Top
SetWindowOrgEx
SetWindowOrgEx

VB 声明
Declare Function SetWindowOrgEx Lib "gdi32" Alias "SetWindowOrgEx" (ByVal hdc As Long,

ByVal nX As Long, ByVal nY As Long, lpPoint As POINTAPI) As Long
说明
设置指定设备场景窗口起点
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hdcLong，设备场景句柄
nX,nYLong，以逻辑坐标表示的窗口起点
lpPointPOINTAPI，装载原来窗口起点的结构

Top
SetWindowRgn
SetWindowRgn

VB 声明
Declare Function SetWindowRgn Lib "user32" Alias "SetWindowRgn" (ByVal hWnd As Long, ByVal hRgn As Long, ByVal bRedraw As Boolean) As Long
说明
这是那些很难有人注意到的对编程者来说是个巨大的宝藏的隐含的 API 函数中的一个。本函数允许您改变窗口的区域。
通常所有窗口都是矩形的——窗口一旦存在就含有一个矩形区域。本函数允许您放弃该区域。这意味着您可以创建圆的、星形的窗口，也可以将它分为两个或许多部分——实际上可以是任何形状
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hWndLong，将设置其区域的窗口
hRgnLong，将设置的区域的句柄，一旦设置了该区域，就不能使用或修改该区域句柄，也不要删除它
bRedrawBoolean，若为 TRUE，则立即重画窗口
注解
为区域指定的所有坐标都以窗口坐标（不是客户坐标）表示，它们以整个窗口（包括标题栏和边框）的左上角为起点

Top
SetWorldTransform
SetWorldTransform

VB 声明
Declare Function SetWorldTransform Lib "gdi32" Alias "SetWorldTransform" (ByVal hdc As Long, lpXform As xform) As Long
说明

设置世界转换
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hdcLong，设备场景
lpXformxform，一个包含将用世界转换的结构
注解
适用于 Windows NT
在使用世界转换前必须调用 SetGraphicsMode 函数设置图形模式为 GM_ADVANCED 模式

Top
ValidateRgn
ValidateRgn

VB 声明
Declare Function ValidateRgn Lib "user32" Alias "ValidateRgn" (ByVal hwnd As Long, ByVal hRgn As Long) As Long
说明
激活窗口中指定区域，把它从刷新区移走
返回值
Long，执行成功为非零值，失败为 0
参数表
参数类型及说明
hwndLong，窗口句柄
hRgnLong，定义要激活区的区域句柄，该区域以窗口客户坐标表示

Top
WindowFromDC
WindowFromDC

VB 声明
Declare Function WindowFromDC Lib "user32" Alias "WindowFromDC" (ByVal hdc As Long) As Long
说明
取回与某一设备场景相关的窗口的句柄
返回值
Long，执行成功为设备场景对应的窗口的句柄，失败为 0
参数表
参数类型及说明
hdcLong，设备场景句柄

Top
进程和线程函数

CancelWaitableTimer 这个函数用于取消一个可以等待下去的计时器操作
CallNamedPipe 这个函数由一个希望通过管道通信的一个客户进程调用
ConnectNamedPipe 指示一台服务器等待下去，直至客户机同一个命名管道连接
CreateEvent 创建一个事件对象
CreateMailslot 创建一个邮路。返回的句柄由邮路服务器使用（收件人）
CreateMutex 创建一个互斥体（MUTEX）
CreateNamedPipe 创建一个命名管道。返回的句柄由管道的服务器端使用
CreatePipe 创建一个匿名管道
CreateProcess 创建一个新进程（比如执行一个程序）
CreateSemaphore 创建一个新的信号机
CreateWaitableTimer 创建一个可等待的计时器对象
DisconnectNamedPipe 断开一个客户与一个命名管道的连接
DuplicateHandle 在指出一个现有系统对象当前句柄的情况下，为那个对象创建一个新句柄
ExitProcess 中止一个进程
FindCloseChangeNotification 关闭一个改动通知对象
FindExecutable 查找与一个指定文件关联在一起的程序的文件名
CallNamedPipe
CallNamedPipe

VB 声明
Declare Function CallNamedPipe Lib "kernel32" Alias "CallNamedPipeA" (ByVal lpNamedPipeName As String, lpInBuffer As Any, ByVal nInBufferSize As Long, lpOutBuffer As Any, ByVal nOutBufferSize As Long, lpBytesRead As Long, ByVal nTimeOut As Long) As Long
说明
这个函数由一个希望通过管道通信的一个客户进程调用。如有可能，它就同一个管道连接（在必要的情况下等候管道可用）。随后，它对指定的数据进行读写，然后将管道关闭
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpNamedPipeNameString，欲打开管道的名称
lpInBufferAny，包含了要写入管道的数据的一个内存缓冲区
nInBufferSizeLong，lpInBuffer 缓冲区中的字符数量
lpOutBufferAny，指定一个内存缓冲区，用于装载从管道中读出的数据
nOutBufferSizeLong，指定一个长整数变量，用于装载来自管道的数据
lpBytesReadLong，指定从管道中读出的字节数。会阅读单条消息。如 lpOutBuffer 的容量不够大，不能容下整条消息，则函数会返回 FALSE，而且 GetLastError 会设为 ERROR_MORE_DATA（消息中留下的任何字节都会丢失）
nTimeOutLong，下述常数之一：
NMPWAIT_NOWAIT 如管道不可用，则立即返回一个错误
NMPWAIT_WAIT_FOREVER 永远等候管道可用
NMPWAIT_USE_DEFAULT_WAIT 使用管道的默认超时设置，这个设置是用 CreateNamedPipe 函数指定的

Top
CancelWaitableTimer
CancelWaitableTimer

VB 声明
Declare Function CancelWaitableTimer Lib "kernel32" (ByVal hTimer As Long)
说明
这个函数用于取消一个可以等待下去的计时器操作。计时器保持它当前的状态，而且除非用 SetWaitableTimer 函数明确启动，否则它不会重新启动
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hTimerLong，可等待计时器的句柄
适用平台
Windows NT

Top
ConnectNamedPipe
ConnectNamedPipe

VB 声明
Declare Function ConnectNamedPipe Lib "kernel32" Alias "ConnectNamedPipe" (ByVal hNamedPipe As Long, lpOverlapped As OVERLAPPED) As Long
说明
指示一台服务器等待下去，直至客户机同一个命名管道连接
返回值
Long，如 lpOverlapped 为 NULL，那么：
□ 如管道已连接，就返回 Ture（非零）；如发生错误，或者管道已经连接，就返回零（GetLastError 此时会返回 ERROR_PIPE_CONNECTED）
□ lpOverlapped 有效，就返回零；如管道已经连接，GetLastError 会返回 ERROR_PIPE_CONNECTED；如重叠操作成功完成，就返回 ERROR_IO_PENDING。在这两种情况下，倘若一个客户已关闭了管道，且服务器尚未用 DisconnectNamedPipe 函数同客户断开连接，那么 GetLastError 都会返回 ERROR_NO_DATA
参数表
参数类型及说明
hNamedPipeLong，管道的句柄
lpOverlappedOVERLAPPED，如设为 NULL（传递 ByVal As Long），表示将线程挂起，直到一个客户同管道连接为止。否则就立即返回；此时，如管道尚未连接，客户同管道连接时就会触发 lpOverlapped 结构中的事件对象。随后，可用一个等待函数来监视连接
适用平台
Windows NT
注解

可用这个函数将一个管道换成同另一个客户连接，但首先必须用 DisconnectNamedPipe 函数断开同当前进程的连接

Top
CreateEvent
CreateEvent

VB 声明
Declare Function CreateEvent Lib "kernel32" Alias "CreateEventA" (lpEventAttributes As SECURITY_ATTRIBUTES, ByVal bManualReset As Long, ByVal bInitialState As Long, ByVal lpName As String) As Long
说明
创建一个事件对象
返回值
Long，如执行成功，返回事件对象句柄；零表示出错。会设置 GetLastError。即使返回一个有效的句柄，但同时指出指定的名字已经存在，GetLastError 也会设为 ERROR_ALREADY_EXISTS
参数表
参数类型及说明
lpEventAttributesSECURITY_ATTRIBUTES，指定一个结构，用于设置对象的安全特性。如变成 ByVal As Long，并传递零值，则表明使用对象默认的安全设置
bManualResetLong，如果为 TRUE，表示创建一个人工重设事件；如果为 FALSE，表示创建一个自动重设事件
bInitialStateLong，如事件应内部进入触发状态，则为 TRUE
lpNameString，指定事件对象的名字。用 vbNullString 创建一个未命名事件对象。如已经存在拥有这个名字的一个事件，则现有的命名事件就会打开。这个名字可能不与一个现有互斥体、信号机、可等待计时器或文件映射的名字相符
注解
一旦不再需要，注意一定要用 CloseHandle 关闭事件句柄。如对象的所有句柄都已关闭，对象也会自动删除

Top
CreateMailslot
CreateMailslot

VB 声明
Declare Function CreateMailslot Lib "kernel32" Alias "CreateMailslotA" (ByVal lpName As String, ByVal nMaxMessageSize As Long, ByVal lReadTimeout As Long, lpSecurityAttributes As SECURITY_ATTRIBUTES) As Long
说明
创建一个邮路。返回的句柄由邮路服务器使用（收件人）
返回值
Long，如执行成功，返回邮路的句柄；INVALID_HANDLE_VALUE 表示失败。会设置 GetLastError

参数表
参数类型及说明
lpNameString，指定邮路的名字，采用的形式如下：\\.\邮路\[路径\]邮路名
nMaxMessageSizeLong，指定一个邮路消息的最大长度。零表示无限长。请注意，对于穿越一个网络域到多个邮路的广播消息，最大长度是 400
lReadTimeoutLong，等待指定的数据时，用这个参数指定邮路使用的默认超时设置，以毫秒为单位。零表示不等待。常数 MAILSLOT_WAIT_FOREVER 表示一直等到数据到达
lpSecurityAttributesSECURITY_ATTRIBUTES，指定一个结构，或传递零值（将参数声明为 ByVal As Long，并传递零值），表示使用不允许继承的默认描述符

Top
CreateMutex
CreateMutex

VB 声明
Declare Function CreateMutex Lib "kernel32" Alias "CreateMutexA" (lpMutexAttributes As SECURITY_ATTRIBUTES, ByVal bInitialOwner As Long, ByVal lpName As String) As Long
说明
创建一个互斥体（MUTEX）
返回值
Long，如执行成功，就返回互斥体对象的句柄；零表示出错。会设置 GetLastError。即使返回的是一个有效句柄，但倘若指定的名字已经存在，GetLastError 也会设为 ERROR_ALREADY_EXISTS
参数表
参数类型及说明
lpMutexAttributesSECURITY_ATTRIBUTES，指定一个 SECURITY_ATTRIBUTES 结构，或传递零值（将参数声明为 ByVal As Long，并传递零值），表示使用不允许继承的默认描述符
bInitialOwnerLong，如创建进程希望立即拥有互斥体，则设为 TRUE。一个互斥体同时只能由一个线程拥有
lpNameString，指定互斥体对象的名字。用 vbNullString 创建一个未命名的互斥体对象。如已经存在拥有这个名字的一个事件，则打开现有的已命名互斥体。这个名字可能不与现有的事件、信号机、可等待计时器或文件映射相符
注解
一旦不再需要，注意必须用 CloseHandle 函数将互斥体句柄关闭。从属于它的所有句柄都被关闭后，就会删除对象
进程中止前，一定要释放互斥体，如不慎未采取这个措施，就会将这个互斥体标记为废弃，并自动释放所有权。共享这个互斥体的其他应用程序也许仍然能够用它，但会接收到一个废弃状态信息，指出上一个所有进程未能正常关闭。这种状况是否会造成影响取决于涉及到的具体应用程序

Top
CreateNamedPipe
CreateNamedPipe

VB 声明
Declare Function CreateNamedPipe Lib "kernel32" Alias "CreateNamedPipeA" (ByVal lpName As String, ByVal dwOpenMode As Long, ByVal dwPipeMode As Long, ByVal nMaxInstances As Long, ByVal nOutBufferSize As Long, ByVal nInBufferSize As Long, ByVal nDefaultTimeOut As Long, lpSecurityAttributes As SECURITY_ATTRIBUTES) As Long
说明
创建一个命名管道。返回的句柄由管道的服务器端使用
返回值
Long，如执行成功，返回管道的句柄。INVALID_HANDLE_VALUE 表示失败。会设置 GetLastError
参数表
参数类型及说明
lpNameString，指定管道名，采用的形式是：\\.\管道\管道名。最多可达 256 个字符的长度，而且不用区分大小写。如果存在指定名字的一个管道，则创建那个管道的一个新实例
dwOpenModeLong，下述常数组的一个组合
下述常数之一（对于管道的所有实例都要一样）：
PIPE_ACCESS_DUPLEX 管道是双向的
PIPE_ACCESS_INBOUND 数据从客户端流到服务器端
PIPE_ACCESS_OUTBOUND 数据从服务器端流到客户端
下述常数的任意组合
FILE_FLAG_WRITE_THROUGH 在网络中建立的字节型管道内，强迫数据在每次读写操作的时候通过网络传输。否则传输就可能延迟
FILE_FLAG_OVERLAPPED 允许（但不要求）用这个管道进行异步（重叠式）操作
常数 WRITE_DAC，WRITE_OWNER 和 ACCESS_ SYSTEM_SECURITY 提供了附加的安全选项
dwPipeModeLong，下述常数组的一个组合：
下述常数之一（管道的所有实例都必须指定相同的常数）
PIPE_TYPE_BYTE 数据作为一个连续的字节数据流写入管道
PIPE_TYPE_MESSAGE 数据用数据块（名为“消息”或“报文”）的形式写入管道
下述常数之一：
PIPE_READMODE_PIPE 数据以单独字节的形式从管道中读出
PIPE_READMODE_MESSAGE 数据以名为“消息”的数据块形式从管道中读出（要求指定 PIPE_TYPE_MESSAGE）
下述常数之一：
PIPE_WAIT 同步操作在等待的时候挂起线程
PIPE_NOWAIT（不推荐！）同步操作立即返回。这样可为异步传输提供一种落后的实现方法，已由 Win32 的重叠式传输机制取代了
nMaxInstancesLong，这个管道能够创建的最大实例数量。必须是 1 到常数 PIPE_UNLIMITED_INSTANCES 间的一个值。它对于管道的所有实例来说都应是相同的
nOutBufferSizeLong，建议的输出缓冲区长度；零表示用默认设置
nInBufferSizeLong，建议的输入缓冲区长度；零表示用默认设置
nDefaultTimeOutLong，管道的默认等待超时。对一个管道的所有实例来说都应相同
lpSecurityAttributesSECURITY_ATTRIBUTES，指定一个 SECURITY_ATTRIBUTES 结构，或者传递零值（将参数声明为 ByVal As Long，并传递零值），以便使用不允许继承的一个默

认描述符
适用平台
Windows NT

Top
CreatePipe
CreatePipe

VB 声明
Declare Function CreatePipe Lib "kernel32" Alias "CreatePipe" (phReadPipe As Long, phWritePipe As Long, lpPipeAttributes As SECURITY_ATTRIBUTES, ByVal nSize As Long) As Long
说明
创建一个匿名管道
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
phReadPipeLong，指定一个变量，设为管道读入（输出）端的一个句柄
phWritePipeLong，指定一个变量，设为管道写入（输入）端的一个句柄
lpPipeAttributesSECURITY_ATTRIBUTES，指定一个 SECURITY_ATTRIBUTES 结构，或者传递零值（将参数声明为 ByVal As Long，并传递零值），以便使用不允许继承的一个默认描述符
nSizeLong，管道缓冲区的建议大小。零表示用默认值
注解
匿名管道不允许异步操作，所以如在一个管道中写入数据，且缓冲区已满，那么除非另一个进程从管道中读出数据，从而腾出了缓冲区的空间，否则写入函数不会返回

Top
CreateProcess
CreateProcess

VB 声明
Declare Function CreateProcess Lib "kernel32" Alias "CreateProcessA" (ByVal lpApplicationName As String, ByVal lpCommandLine As String, lpProcessAttributes As SECURITY_ATTRIBUTES, lpThreadAttributes As SECURITY_ATTRIBUTES, ByVal bInheritHandles As Long, ByVal dwCreationFlags As Long, lpEnvironment As Any, ByVal lpCurrentDriectory As String, lpStartupInfo As STARTUPINFO, lpProcessInformation As PROCESS_INFORMATION) As Long
说明
创建一个新进程（比如执行一个程序）
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表

参数类型及说明
lpApplicationNameString，要执行的应用程序的名字。可设为 vbNullString；在这种情况下，应用程序的名字应在 lpCommandLine 参数的起始处出现
lpCommandLineString，要执行的命令行。可用 GetCommandLine 函数取得一个进程使用的命令行。Windows 会尽可能地根据下述搜索顺序来查找执行文件：
（1）包含了父进程执行文件的目录
（2）父进程当前的目录
（3）由 GetSystemDirectory 返回的系统目录
（4）仅适于 windows NT：16 位系统目录
（5）由 GetWindowDirectory 返回的 Windows 目录
（6）由 PATH 环境变量指定的目录
lpProcessAttributesSECURITY_ATTRIBUTES，指定一个 SECURITY_ATTRIBUTES 结构，或传递零值（将参数声明为 ByVal As Long，并传递零值）——表示采用不允许继承的默认描述符。该参数定义了进程的安全特性
lpThreadAttributesSECURITY_ATTRIBUTES，指定一个 SECURITY_ATTRIBUTES 结构，或传递零值（将参数声明为 ByVal As Long，并传递零值）——表示采用不允许继承的默认描述符。该参数定义了进程之主线程的安全特性
bInheritHandlesLong，TRUE 表示允许当前进程中的所有句柄都由新建的子进程继承
dwCreationFlagsLong，来自 API32.TXT 文件的一个或多个下述常数之一，它们都带有前缀 CREATE_。下面这些用于 VB 程序员：
CREATE_SEPARATE_WOW_VDM（仅适用于 NT）启动一个 16 位的 Windows 应用程序时，强迫它在自己的内存空间运行
CREATE_SHARED_WOW_VDM（仅适用于 NT）启动一个 16 位的 Windows 应用程序时，强迫它在共享的 16 位虚拟机（VM）内运行
CREATE_SUSPENDED 立即挂起新进程。除非调用了 ResumeThread 函数函数，否则它不会恢复运行
也可能是下述常数之一，用于指定优先级
IDLE_PRIORITY_CLASS 新进程应该有非常低的优先级——只有在系统空闲的时候才能运行。基本值是 4
HIGH_PRIORITY_CLASS 新进程有非常高的优先级，它优先于大多数应用程序。基本值是 13。注意尽量避免采用这个优先级
NORMAL_PRIORITY_CLASS 标准优先级。如进程位于前台，则基本值是 9；如在后台，则优先值是 7
不要在 VB 中使用 REALTIME_PRIORITY_CLASS
lpEnvironmentAny，指向一个环境块的指针（环境缓冲区的头一个字符，或者环境块的地址）
lpCurrentDriectoryString，新进程的当前目录路径。调用函数的时候，可用 vbNullString 指定当前目录
lpStartupInfoSTARTUPINFO，指定一个 STARTUPINFO 结构，其中包含了创建进程时使用的附加信息
lpProcessInformationPROCESS_INFORMATION，该结构用于容纳新进程的进程和线程标识符。大多数情况下，一旦这个函数返回，父应用程序都会关闭两个句柄。

Top
CreateSemaphore

CreateSemaphore

VB 声明
Declare Function CreateSemaphore Lib "kernel32" Alias "CreateSemaphoreA" (lpSemaphoreAttributes As SECURITY_ATTRIBUTES, ByVal lInitialCount As Long, ByVal lMaximumCount As Long, ByVal lpName As String) As Long
说明
创建一个新的信号机
返回值
Long，如执行成功，返回信号机对象的句柄；零表示出错。会设置 GetLastError。即使返回一个有效的句柄，但倘若它指出同名的一个信号机已经存在，那么 GetLastError 也会返回 ERROR_ALREADY_EXISTS
参数表
参数类型及说明
lpSemaphoreAttributesSECURITY_ATTRIBUTES，指定一个 SECURITY_ATTRIBUTES 结构，或传递零值（将参数声明为 ByVal As Long，并传递零值）——表示采用不允许继承的默认描述符。该参数定义了信号机的安全特性
lInitialCountLong，设置信号机的初始计数。可设置零到 lMaximumCount 之间的一个值
lMaximumCountLong，设置信号机的最大计数
lpNameString，指定信号机对象的名称。用 vbNullString 可创建一个未命名的信号机对象。如果已经存在拥有这个名字的一个信号机，就直接打开现成的信号机。这个名字可能不与一个现有的互斥体、事件、可等待计时器或文件映射的名称相符
注解
一旦不再需要，一定记住用 CloseHandle 关闭信号机的句柄。它的所有句柄都关闭以后，对象自己也会删除
一旦值大于零，信号机就会触发（发出信号）。ReleaseSemaphore 函数的作用是增加信号机的计数。如果成功，就调用信号机上的一个等待函数来减少它的计数

Top
CreateWaitableTimer
CreateWaitableTimer

VB 声明
Declare Function CreateWaitableTimer Lib "kernel32" Alias "CreateWaitableTimerA" (lpSemaphoreAttributes As SECURITY_ATTRIBUTES, ByVal bManualReset As Long, ByVal lpName As String) As Long
说明
创建一个可等待的计时器对象
返回值
Long，如执行成功，返回可等待计时器对象的句柄；零表示出错。会设置 GetLastError。即使返回一个有效的句柄，但倘若它指出同名的一个计时器对象已经存在，那么 GetLastError 也会返回 ERROR_ALREADY_EXISTS
参数表
参数类型及说明

lpSemaphoreAttributesSECURITY_ATTRIBUTES，指定一个结构，用于设置对象的安全特性。如将参数声明为 ByVal As Long，并传递零值，就可使用对象的默认安全设置
bManualResetLong，如果为 TRUE，表示创建一个人工重设计时器；如果为 FALSE，则创建一个自动重设计时器
lpNameString，指定可等待计时器对象的名称。用 vbNullString 可创建一个未命名的计时器对象。如果已经存在拥有这个名字的一个可等待计时器，就直接打开现成的可等待计时器。这个名字可能不与一个现有的互斥体、事件、信号机或文件映射的名称相符
注解
一旦不再需要，一定记住用 CloseHandle 关闭计时器对象的句柄。它的所有句柄都关闭以后，对象自己也会删除

Top
DisconnectNamedPipe
DisconnectNamedPipe

VB 声明
Declare Function DisconnectNamedPipe Lib "kernel32" Alias "DisconnectNamedPipe" (ByVal hNamedPipe As Long) As Long
说明
断开一个客户与一个命名管道的连接
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hNamedPipeLong，管道的句柄
适用平台
Windows NT
注解
如客户尚未在它自己那端关闭管道句柄，下次试图访问管道的时候就会发生错误

Top
DuplicateHandle
DuplicateHandle

VB 声明
Declare Function DuplicateHandle Lib "kernel32" Alias "DuplicateHandle" (ByVal hSourceProcessHandle As Long, ByVal hSourceHandle As Long, ByVal hTargetProcessHandle As Long, lpTargetHandle As Long, ByVal dwDesiredAccess As Long, ByVal bInheritHandle As Long, ByVal dwOptions As Long) As Long
说明
在指出一个现有系统对象当前句柄的情况下，为那个对象创建一个新句柄。当前句柄可能位于一个不同的进程
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError

参数表
参数类型及说明
hSourceProcessHandleLong，拥有源句柄的那个进程的句柄。如源句柄从属于当前进程，则使用 GetCurrentProcess
hSourceHandleLong，指定对象的现有句柄。
hTargetProcessHandleLong，即将拥有新对象句柄的一个进程的句柄。如源句柄从属于当前进程，则使用 GetCurrentProcess
lpTargetHandleLong，指定用于装载新句柄的一个长整型变量
dwDesiredAccessLong，新句柄要求的安全访问级别。如 dwOptions 已指定了 DUPLICATE_SAME_ACCESS，那么忽略这里的设置。可以进行的访问由对象的类型决定，它们在不同系统对象的访问常数表里进行了总结
bInheritHandleLong，如新句柄可由 hSourceProcessHandle 的子进程继承，则为 TRUE
dwOptionsLong，下列常数的一个或两个：
DUPLICATE_SAME_ACCESS 新句柄拥有与原始句柄相同的安全访问特征
DUPLICATE_CLOSE_SOURCE 原始句柄已经关闭。即使发生错误。它也要关闭
注解
在一个进程中，这个函数可根据位于不同进程内的现有句柄创建一个新句柄。可以从这两个进程中发出对这个函数的调用。进程必须提供 PROCESS_DUP_HANDLE 访问权限，否则函数执行不能成功
句柄可以重复的对象包括控制台、文件（包括通信设备）、文件映射、事件、可等待计时器、互斥体、管道、进程、注册表项、信号机以及线程

Top
ExitProcess
ExitProcess

VB 声明
Declare Sub ExitProcess Lib "kernel32" Alias "ExitProcess" (ByVal uExitCode As Long)
说明
中止一个进程
参数表
参数类型及说明
uExitCodeLong，指定想中断的那个进程的一个退出代码
在 VB 中使用
应尽量避免用该函数来关闭进程。不要在自己的 VB 程序中使用它。此时，应试着向要关闭的那个程序的主窗口投递一条 WM_CLOSE 消息

Top
FindCloseChangeNotification
FindCloseChangeNotification

VB 声明
Declare Function FindCloseChangeNotification Lib "kernel32" Alias "FindCloseChangeNotification" (ByVal hChangeHandle As Long) As Long

说明
关闭一个改动通知对象
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hChangeHandleLong，要关闭的那个文件改动通知对象的句柄

Top
FindExecutable
FindExecutable

VB 声明
Declare Function FindExecutable Lib "shell32.dll" Alias "FindExecutableA" (ByVal lpFile As String, ByVal lpDirectory As String, ByVal lpResult As String) As Long
说明
查找与一个指定文件关联在一起的程序的文件名。可用 Windows 注册表编辑器将文件类型与特定的应用程序关联到一起。比如，扩展名为.TXT 的文本文件通常与 Windows 记事本（Notepad.exe）关联到一起。如在文件管理器中双击含.TXT 扩展名的一个文件，会自动启动记事本，并在其中载入文本文件
返回值
Long，大于 32 表示成功；31 表示不存在文件类型的关联；0 表示系统内存或资源不足；ERROR_FILE_NOT_FOUND 表示指定的文件不存在；ERROR_PATH_NOT_FOUND 表示指定的路径不存在；ERROR_BAD_FORMAT 表示执行格式无效
参数表
参数类型及说明
lpFileString，指定要为其查找相关程序的一个文件名或程序名
lpDirectoryString，要使用的默认目录的完整路径
lpResultString，指定一个字串缓冲区，用于装载可执行程序的名字。注意这个字串预先至少都要初始化成 MAX_PATH 个字符的长度

Top
FindFirstChangeNotification
FindFirstChangeNotification

VB 声明
Declare Function FindFirstChangeNotification Lib "kernel32" Alias "FindFirstChangeNotificationA" (ByVal lpPathName As String, ByVal bWatchSubtree As Long, ByVal dwNotifyFilter As Long) As Long
说明
创建一个文件通知对象。该对象用于监视文件系统发生的变化
返回值
Long，如成功，返回一个改变通知对象的句柄；INVALID_HANDLE_VALUE 表示失败。会设置 GetLastError

参数表
参数类型及说明
lpPathNameString，要监视的目录
bWatchSubtreeLong，如果为 TRUE，表示监视 lpPathName 的所有子目录
dwNotifyFilterLong，带有前缀 FILE_NOTIFY_CHANGE_???前缀的一个或多个常数，它们指定了对象发出信号的条件
注解
用 FindCloseChangeNotification 函数关闭句柄，不要用 CloseHandle 函数

Top
FindNextChangeNotification
FindNextChangeNotification

VB 声明
Declare Function FindNextChangeNotification Lib "kernel32" Alias "FindNextChangeNotification" (ByVal hChangeHandle As Long) As Long
说明
重设一个文件改变通知对象，令其继续监视下一次变化
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hChangeHandleLong，要重设的那个文件改变通知对象的句柄

Top
FreeLibrary
FreeLibrary

VB 声明
Declare Function FreeLibrary Lib "kernel32" Alias "FreeLibrary" (ByVal hLibModule As Long) As Long
说明
释放指定的动态链接库，它们早先是用 LoadLibrary API 函数装载的
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hLibModuleLong，要释放的一个库句柄
在 VB 里使用
只能用这个函数释放那些由应用程序明确装载的 DLL。对 LoadLibrary 的每一次调用都应该有一个对应的 FreeLibrary 调用

Top
GetCurrentProcess

GetCurrentProcess

VB 声明
Declare Function GetCurrentProcess Lib "kernel32" Alias "GetCurrentProcess" () As Long
说明
获取当前进程的一个伪句柄
返回值
Long，当前进程的伪句柄
注解
只要当前进程需要一个进程句柄，就可以使用这个伪句柄。该句柄可以复制，但不可继承。不必调用 CloseHandle 函数来关闭这个句柄

Top
GetCurrentProcessId
GetCurrentProcessId

VB 声明
Declare Function GetCurrentProcessId Lib "kernel32" Alias "GetCurrentProcessId" () As Long
说明
获取当前进程一个唯一的标识符
返回值
Long，当前的进程标识符

Top
GetCurrentThread
GetCurrentThread

VB 声明
Declare Function GetCurrentThread Lib "kernel32" Alias "GetCurrentThread" () As Long
说明
获取当前线程的一个伪句柄
返回值
Long，当前线程的伪句柄
注解
只要当前线程需要使用一个线程句柄，就可以使用这个伪句柄（但在其他任务线程中都无效）。该句柄可以复制，但不可继承。不必调用 CloseHandle 函数来关闭这个句柄

Top
GetCurrentThreadId
GetCurrentThreadId

VB 声明
Declare Function GetCurrentThreadId Lib "kernel32" Alias "GetCurrentThreadId" () As Long
说明

获取当前线程一个唯一的线程标识符
返回值
Long，当前的线程标识符

Top
GetExitCodeProcess
GetExitCodeProcess

VB 声明
Declare Function GetExitCodeProcess Lib "kernel32" Alias "GetExitCodeProcess" (ByVal hProcess As Long, lpExitCode As Long) As Long
说明
获取一个已中断进程的退出代码
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hProcess Long，想获取退出代码的一个进程的句柄
lpExitCodeLong，用于装载进程退出代码的一个长整数变量。如进程尚未中止，则设为常数 STILL_ACTIVE

Top
GetExitCodeThread
GetExitCodeThread

VB 声明
Declare Function GetExitCodeThread Lib "kernel32" Alias "GetExitCodeThread" (ByVal hThread As Long, lpExitCode As Long) As Long
说明
获取一个已中止线程的退出代码
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hThreadLong，想获取退出代码的一个线程的句柄
lpExitCodeLong，用于装载线程退出代码的一个长整数变量。如线程尚未中断，则设为常数 STILL_ACTIVE

Top
GetHandleInformation
GetHandleInformation

VB 声明
Declare Function GetHandleInformation Lib "kernel32" Alias "GetHandleInformation" (ByVal

hObject As Long, lpdwFlags As Long) As Long
说明
获取与一个系统对象句柄有关的信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hObjectLong，要测试的句柄
lpdwFlagsLong，设置了下述一个或多个常数的位标志：
HANDLE_FLAG_INHERIT 对象可由一个子进程继承
HANDLE_FLAG_PROTECT_FROM_CLOSE 句柄不可用 CloseHandle 函数关闭
适用平台
Windows NT

Top
GetMailslotInfo
GetMailslotInfo

VB 声明
Declare Function GetMailslotInfo Lib "kernel32" Alias "GetMailslotInfo" (ByVal hMailslot As Long, lpMaxMessageSize As Long, lpNextSize As Long, lpMessageCount As Long, lpReadTimeout As Long) As Long
说明
获取与一个邮路有关的信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hMailslotLong，指定一个邮路的句柄
lpMaxMessageSizeLong，指定一个长整数变量，用于装载这个邮路的最大消息长度
lpNextSizeLong，指定一个长整数变量，用于装载下一条消息的长度。如没有消息准备好，则可设为常数 MAILSLOT_NO_MESSAGE
lpMessageCountLong，指定一个长整数变量，用于装载邮路中准备好的消息数量
lpReadTimeoutLong，指定一个长整数变量，用于装载邮路的默认阅读超时

Top
GetModuleFileName
GetModuleFileName

VB 声明
Declare Function GetModuleFileName Lib "kernel32" Alias "GetModuleFileNameA" (ByVal hModule As Long, ByVal lpFileName As String, ByVal nSize As Long) As Long
说明
获取一个已装载模板的完整路径名称

返回值
Long，如执行成功，返回复制到 lpFileName 的实际字符数量；零表示失败。会设置 GetLastError
参数表
参数类型及说明
hModuleLong，一个模块的句柄。可以是一个 DLL 模块，或者是一个应用程序的实例句柄
lpFileNameString，指定一个字串缓冲区，要在其中容纳文件的用 NULL 字符中止的路径名，hModule 模块就是从这个文件装载进来的
nSizeLong，装载到缓冲区 lpFileName 的最大字符数量
注解
在 Windows 95 下，函数会核查应用程序的内部版本号是否为 4.0 或更大的一个数字。如果是，就返回一个长文件名，否则返回短文件名

Top
GetModuleHandle
GetModuleHandle

VB 声明
Declare Function GetModuleHandle Lib "kernel32" Alias "GetModuleHandleA" (ByVal lpModuleName As String) As Long
说明
获取一个应用程序或动态链接库的模块句柄
返回值
Long，如执行成功成功，则返回模块句柄。零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpModuleNameString，指定模块名，这通常是与模块的文件名相同的一个名字。例如，NOTEPAD.EXE 程序的模块文件名就叫作 NOTEPAD
注解
只有在当前进程的场景中，这个句柄才会有效

Top
GetPriorityClass
GetPriorityClass

VB 声明
Declare Function GetPriorityClass Lib "kernel32" Alias "GetPriorityClass" (ByVal hProcess As Long) As Long

说明
　获取特定进程的优先级别
返回值
Long，进程的优先级，零表示失败。会设置 GetLastError
参数表
参数类型及说明

hProcessLong，进程句柄

Top
GetProcessShutdownParameters
GetProcessShutdownParameters

VB 声明
Declare Function GetProcessShutdownParameters Lib "kernel32" Alias "GetProcessShutdownParameters" (lpdwLevel As Long, lpdwFlags As Long) As Long

说明
　调查系统关闭时一个指定的进程相对于其它进程的关闭早迟情况
返回值
Long，非零表示成功，零表示失败。
参数表
参数类型及说明
lpdwLevelLong，从 0 到&H4FF 的一个值。编号越高程序在系统关闭时越早关闭。
lpdwFlagsLong，指定 SHUTDOWN_NORETRY 标志。如关闭一个进程时，会出现一个重试对话框。通过设置这个标志，就可以禁止那个对话框出现，并直接关闭进程
适用平台
　Wnidows NT

Top
GetProcessTimes
GetProcessTimes

VB 声明
Declare Function GetProcessTimes Lib "kernel32" Alias "GetProcessTimes" (ByVal hProcess As Long, lpCreationTime As FILETIME, lpExitTime As FILETIME, lpKernelTime As FILETIME, lpUserTime As FILETIME) As Long

说明
　获取与一个进程的经过时间有关的信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hProcessLong，一个进程句柄
lpCreationTimeFILETIME，指定一个 FILETIME 结构，在其中装载进程的创建时间
lpExitTimeFILETIME，指定一个 FILETIME 结构，在其中装载进程的中止时间
lpKernelTimeFILETIME，指定一个 FILETIME 结构，在其中装载进程花在内核模式上的总时间
lpUserTimeFILETIME，指定一个 FILETIME 结构，在其中装载进程花在用户模式上的总时间

适用平台
Windows NT

Top
GetProcessWorkingSetSize
GetProcessWorkingSetSize

VB 声明
Declare Function GetProcessWorkingSetSize Lib "kernel32" Alias "GetProcessWorkingSetSize" (ByVal hProcess As Long, lpMinimumWorkingSetSize As Long, lpMaximumWorkingSetSize As Long) As Long

说明
　了解一个应用程序在运行过程中实际向它交付了多大容量的内存
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hProcessLong，指定一个进程的句柄
lpMinimumWorkingSetSizeLong，用于装载最小进程容量的一个变量
lpMaximumWorkingSetSizeLong，用于装载最大进程容量的一个变量
适用平台
Windows NT

Top
GetSartupInfo
GetSartupInfo

VB 声明
Declare Sub GetStartupInfo Lib "kernel32" Alias "GetStartupInfoA" (lpStartupInfo As STARTUPINFO)

说明
获取一个进程的启动信息
参数表
参数类型及说明
lpStartupInfoSTARTUPINFO，指定一个 STARTUPINFO 结构，用于最终载入进程的启动信息

Top
GetTheardTimes
GetTheardTimes

VB 声明
Declare Function GetThreadTimes Lib "kernel32" Alias "GetThreadTimes" (ByVal hThread As Long, lpCreationTime As FILETIME, lpExitTime As FILETIME, lpKernelTime As FILETIME, lpUserTime As FILETIME) As Long
说明
　获取与一个线程的经过时间有关的信息
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hThreadLong，一个线程句柄
lpCreationTimeFILETIME，指定一个 FILETIME 结构，在其中装载线程的创建时间
lpExitTimeFILETIME，指定一个 FILETIME 结构，在其中装载线程的中止时间
lpKernelTimeFILETIME，指定一个 FILETIME 结构，在其中装载线程花在内核模式上的总时间
lpUserTimeFILETIME，指定一个 FILETIME 结构，在其中装载线程花在用户模式上的总时间
适用平台
Windows NT

Top
GetThreadPriority
GetThreadPriority

VB 声明
Declare Function GetThreadPriority Lib "kernel32" Alias "GetThreadPriority" (ByVal hThread As Long) As Long
说明
获取特定线程的优先级别
返回值
Long，返回带有 THREAD_PRIORITY_??? 前缀的某个函数，它规定了线程的优级。THREAD_PRIORITY_ERROR_RETURN 表示出错
参数表
参数类型及说明
hThreadLong，线程句柄
注解
线程的优先级同进程的优先级类组合在一起就决定了线程的实际优先级

Top
GetWindowThreadProcessId
GetWindowThreadProcessId

VB 声明
Declare Function GetWindowThreadProcessId Lib "user32" Alias "GetWindowThreadProcessId"

(ByVal hwnd As Long, lpdwProcessId As Long) As Long

说明
获取与指定窗口关联在一起的一个进程和线程标识符
返回值
Long，拥有窗口的线程的标识符
参数表
参数类型及说明
lpdwProcessIdLong，指定一个变量，用于装载拥有那个窗口的一个进程的标识符
hwndLong，指定窗口句柄

Top
LoadLibrary
LoadLibrary

VB 声明
Declare Function LoadLibrary Lib "kernel32" Alias "LoadLibraryA" (ByVal lpLibFileName As String) As Long
说明
载入指定的动态链接库，并将它映射到当前进程使用的地址空间。一旦载入，即可访问库内保存的资源
返回值
Long，成功则返回库模块的句柄，零表示失败。会设置 GetLastError
参数表
参数类型及说明
lpLibFileNameString，指定要载入的动态链接库的名称。采用与 CreateProcess 函数的 lpCommandLine 参数指定的同样的搜索顺序
注解
一旦不需要，用 FreeLibrary 函数释放 DLL

Top
LoadLibraryEx
LoadLibraryEx

VB 声明
Declare Function LoadLibraryEx Lib "kernel32" Alias "LoadLibraryExA" (ByVal lpLibFileName As String, ByVal hFile As Long, ByVal dwFlags As Long) As Long

说明
装载指定的动态链接库，并为当前进程把它映射到地址空间。一旦载入，就可以访问库内保存的资源
返回值
Long，成功则返回库模块的句柄，零表示失败。会设置 GetLastError
参数表

参数类型及说明
lpLibFileNameString，指定要载入的动态链接库的名称。采用与 CreateProcess 函数的 lpCommandLine 参数指定的同样的搜索顺序
hFileLong，未用，设为零
dwFlagsLong，指定下述常数的一个或多个
DONT_RESOLVE_DLL_REFERENCES：不对 DLL 进行初始化，仅用于 NT
LOAD_LIBRARY_AS_DATAFILE：不准备 DLL 执行。如装载一个 DLL 只是为了访问它的资源，就可以改善一部分性能
LOAD_WITH_ALTERED_SEARCH_PATH：指定搜索的路径
注解
一旦不需要，用 FreeLibrary 函数释放 DLL

Top
LoadModule
LoadModule

VB 声明
Declare Function LoadModule Lib "kernel32" Alias "LoadModule" (ByVal lpModuleName As String, lpParameterBlock As Any) As Long
说明
　载入一个 windows 应用程序，并在指定的环境中运行
返回值
Long，大于 32 表示成功，请参考 FindExecutable 函数的返回值
参数表
参数类型及说明
lpModuleNameSring，要装载的可执行程序的文件名
lpParameterBlockAny，指定一个结构，用它定义装载新应用程序时使用的参数

Top
MsgWaitForMultipleObjects
MsgWaitForMultipleObjects

VB 声明
Declare Function MsgWaitForMultipleObjects Lib "user32" Alias "MsgWaitForMultipleObjects" (ByVal nCount As Long, pHandles As Long, ByVal fWaitAll As Long, ByVal dwMilliseconds As Long, ByVal dwWakeMask As Long) As Long
说明
等侯单个对象或一系列对象发出信号---标志着规定的超时已经过去，或特定类型的消息已抵达线程的输入队列。如返回条件已经满足，则立即返回
返回值
Long，如 fWaitAll 设为 TRUE，则下述任何一个常数都标志着成功：
WAIT_ABANDONED_0：所有对象都发出消息，而且其中一个或多个属于互斥体（一但拥有它门的进程中止，就会发出信号）。
WAIT_TIMEOUT：对象保持未发信号的状态，但规定的等待超时时间已经超过

WAIT_OBJECT_0：所有的对象都发出信号
WAIT_TO_COMPLETION（仅适用于 WaitForSingleObjectEx），由于一个 I/O 完成操作已准备好执行，从而造成了函数的返回。
返回 WAIT_FALIED 表示函数执行失败。会设置 GetLastError
如 fWaitAll 设为 FALSE，那返回结果与前面说的相似，只是可能还会返回相对于 WAIT_ABANDONED_0 或 WAIT_OBJECT_0 的一个正偏移量，指出哪个对象是被抛弃还是发出信号。
如果是由于 dwWakeMask 指定的，符合特殊标准的一条消息的到达而造成了函数的返回，则返回 WAIT_OBJECT_0 + nCount
参数表
参数类型及说明
nCountLong，指定列表中的句柄数量
pHandlesLong，指定对象句柄组合中的第一个元素
fWaitAllLong，如果为 TRUE，表示除非对象同时发出信号，否则就等待下去。如果为 FALSE，表示任何对象发出信号即可。
dwMillisecondsLong，指定要等待的毫秒数。
dwWakeMaskLong，带有 QS_？？前缀的一个或多个常数，用于标识特定的消息类型。
注解
如果函数是由于对象发出信号而返回，这个函数还会得到一些额外的效果：
信号机：递增信号机计数
互斥体：将互斥体的所有权限赋于发出调用的线程
自动重设事件：将事件发信状态设为 FALSE
自动重设可等待计时器：将计时器状态设为 FALSE

Top
SetPriorityClass
SetPriorityClass

VB 声明
Declare Function SetPriorityClass Lib "kernel32" Alias "SetPriorityClass" (ByVal hProcess As Long, ByVal dwPriorityClass As Long) As Long
说明
　设置一个进程的优先级别
返回值
Long，进程的优先级，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hProcessLong，指定一个进程句柄
dwPriorityClassLong，指定一个新优先级类的一个常数，请参考 CreateProcess 函数

Top
SetProcessShutdownParameters
SetProcessShutdownParameters

VB 声明
Declare Function SetProcessShutdownParameters Lib "kernel32" Alias "SetProcessShutdownParameters" (ByVal dwLevel As Long, ByVal dwFlags As Long) As Long
说明
　在系统关闭期间，为指定进程设置他相对于其它程序的关闭顺序
返回值
Long，非零表示成功，零表示失败。
参数表
参数类型及说明
lpdwLevelLong，从 0 到&H4FF 的一个值。编号越高程序在系统关闭时越早关闭。
lpdwFlagsLong，指定 SHUTDOWN_NORETRY 标志。如关闭一个进程时，会出现一个重试对话框。通过设置这个标志，就可以禁止那个对话框出现，并直接关闭进程
适用平台
　Wnidows NT

Top
SetProcessWorkingSetSize
SetProcessWorkingSetSize

VB 声明
Declare Function SetProcessWorkingSetSize Lib "kernel32" Alias "SetProcessWorkingSetSize" (ByVal hProcess As Long, ByVal dwMinimumWorkingSetSize As Long, ByVal dwMaximumWorkingSetSize As Long) As Long
说明
　设置操作系统实际划分给进程使用的内存容量
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hProcessLong，指定一个进程的句柄
lpMinimumWorkingSetSizeLong，用于装载最小进程容量的一个变量
lpMaximumWorkingSetSizeLong，用于装载最大进程容量的一个变量
适用平台
Windows NT

Top
SetThreadPriority
SetThreadPriority

VB 声明
Declare Function SetThreadPriority Lib "kernel32" Alias "SetThreadPriority" (ByVal hThread As Long, ByVal nPriority As Long) As Long
说明
　设定线程的优先级别

返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hThreadLong，线程句柄
nPriorityLong，返回带有 THREAD_PRIORITY_???前缀的某个函数，它定义了线程的优级。
注解
线程的优先级同进程的优先级类组合在一起就决定了线程的实际优先级

Top
ShellExecute
ShellExecute

VB 声明
Declare Function ShellExecute Lib "shell32.dll" Alias "ShellExecuteA" (ByVal hwnd As Long, ByVal lpOperation As String, ByVal lpFile As String, ByVal lpParameters As String, ByVal lpDirectory As String, ByVal nShowCmd As Long) As Long
说明
　查找与指定文件关联在一起的程序的文件名
返回值
Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hwndLong，指定一个窗口的句柄，有时候，windows 程序有必要在创建自己的主窗口前显示一个消息框
lpOperationString，指定字串“open”来打开 lpFlie 文档，或指定“Print”来打印它
lpFileString，想用关联程序打印或打开一个程序名或文件名
lpParametersString，如 lpszFlie 是可执行文件，则这个字串包含传递给执行程序的参数
lpDirectoryString，想使用的完整路径
nShowCmdLong，定义了如何显示启动程序的常数值。参考 ShowWindow 函数的 nCmdShow 参数

Top
TerminateProcess
TerminateProcess

VB 声明
Declare Function TerminateProcess Lib "kernel32" Alias "TerminateProcess" (ByVal hProcess As Long, ByVal uExitCode As Long) As Long
说明
结束一个进程
在 VB 里使用
可以使用，但尽量不用
返回值

Long，非零表示成功，零表示失败。会设置 GetLastError
参数表
参数类型及说明
hProcessLong，指定要中断的一个进程的句柄
uExitCodeLong，进程的一个退出代码

Top
WinExec
WinExec

VB 声明
Declare Function WinExec Lib "kernel32" Alias "WinExec" (ByVal lpCmdLine As String, ByVal nCmdShow As Long) As Long
说明
运行指定的程序
返回值
Long，大于 32 表示成功，请参考 FindExecutable 函数
参数表
参数类型及说明
lpCmdLineString，包含要执行的命令行
nCmdShowLong，定义了以怎样的形式启动程序的常数值。参考 ShowWindow 函数的 nCmdShow 参数
注解
请参考对 CreateProcess 函数的说明，了解在目录中查找指定文件的顺序

Top