1. API之网络函数    
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    
  
      2. API之消息函数    
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    
3. API之文件处理函数    
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    
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    
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    
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    
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    
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    
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    
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

4. API之打印函数   
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   
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   
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   
5. API之文本和字体函数   
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   
SetMapperFlags 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   
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

6. API之菜单函数   
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   
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   
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   
7. API之位图、图标和光栅运算函数   
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   
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   
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   
  
  
8. API之绘图函数   
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   
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   
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   
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   
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   
PaintDesk 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   
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   
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

9. API之设备场景函数   
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   
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   
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   
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   
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.

10. API之硬件与系统函数   
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   
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   
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   
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   
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

11. API之进程和线程函数   
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   
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   
  
  
  
  
12. API之控件与消息函数   
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   
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   
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   
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   
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

GetClientRect   
  
函数功能：该函数获取窗口客户区的坐标。客户区坐标指定客户区的左上角和右下角。由于客户区坐标是相对子窗口客户区的左上角而言的，因此左上角坐标为（0，0）   
函数原型：BOOL GetClientRect（HWND hWnd，LPRECT lpRect）；   
参数：   
GetLastError 函数。   
备注：Windows CE：命令条包含在客户区中。   
速查：Windows NT: 3.1以上版本：Windows：95以上版本： Windows CE：1.0以上版本：头文件：winuser.h；库文件：user32.lib   
  
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =   
  
GetDesktopWindow   
  
函数功能：该函数返回桌面窗口的句柄。桌面窗口覆盖整个屏幕。桌面窗口是一个要在其上绘制所有的图标和其他窗口的区域。   
函数原型：HWND GetDesktopWindow（VOID）   
参数：无。   
返回值：函数返回桌面窗口的句柄。   
速查：Windows NT：3.1以上版本；Windows：95以上版本：Windows CE：不支持；头文件：Winuser.h；库文件：user32.lib。   
  
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =   
  
GetForegroundwindow   
  
函数功能：该函数返回前台窗口（用户当前工作的窗口）。系统分配给产生前台窗口的线程一个稍高一点的优先级。   
函数原型：HWND GetForegroundwindow（VOID）   
参数：无。   
返回值：函数返回前台窗回的句柄。   
速查：Windows NT：3.1以上版本；Windows：95以上版本：Windows CE：1.0以上版本：头文件：Winuser.h；库文件：user32.lib。   
  
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =   
  
GetLastActivePopup   
  
函数功能：该函数确定指定窗口中的哪一个弹出式窗口是最近活动的窗口。   
函数原型：HWND GetLastActivePopup（HWND hWnd）；   
参数：   
hWnd:所有者窗口句柄。   
返回值：返回值标识了最近活动的弹出式窗口的句柄。如果满足下列任一条件，则返回值与参数hWnd   
相同：由hWnd指定的窗口是最近活动的：由hWnd指定的窗口不拥有任何弹出式窗口；由hWnd指定的窗口不是顶层窗口或它属于其他窗口。   
速查：Windows：3.1以上版本；Windows：95以上版本：Windows CE：不支持；头文件：winuser.h；   
库文件：user32.lib。   
  
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =   
  
GetNextWindow   
  
函数功能：该函数返回z序中的前一个或后一个窗口的句柄。下一窗口在指定窗口的下面，前一窗口则在上面。如果指定的窗口是顶端窗口，该函数返回下一个（或前一个）顶端窗口的句柄。如果指定的窗口是顶层窗口，函数返回下一个（或前一个）顶层窗口的句柄。如果函数是子窗口，则函数搜索下一个或前一个子窗口的句柄。   
函数原型：HWND GetNextWindow（HWND hWnd，UNIT wCmd）；   
参数：   
hWnd：一个窗口的句柄。窗口句柄在wCmd参数的基础上获得的相对于这个窗口的句柄。   
wCmd:指明窗口返回的是前一窗口的句柄还是后一窗口的句柄。该参数可以是下列两个值之一：   
GW HWNONEXT：返回在给定窗口的下面窗口的句柄。   
GW\_HWNDPREV：返回在给定窗口的上面窗口的句柄。   
返回值：如果函数成功，返回值是前一窗口（或后一窗口）的句柄。如果前后窗口不存在，则返回值为NULL。若想获得更多错误信息，请调用GetLastError函数。   
备注：在设定了GW\_HWNDNEXT或GW\_GETPREV标志时，调用该函数与调用GetWindow函数相同。   
速查：Windows NT：3.1以上版本；Windows：95以上版本：Windows CE：不支持：头文件：Winuer.h;库文件：user32.lib。   
  
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =   
  
GetParent   
  
函数功能：该函数获得一个指定子窗口的父窗口句柄。   
函数原型：HWND GetParent（HWND hWnd）；   
参数：   
hWnd:子窗口句柄，函数要获得该子窗口的父窗口句柄。   
返回值：如果函数成功，返回值为父窗口句柄。如果窗口无父窗口，则函数返回NULL。若想获得更多错误信息，请调用GetLastError函数。   
备注：WindowsCE：Windows CE1.0版本不支持除了对话框之外的所属子窗口。   
速查：Windows NT：3.1以上版本；Windows：95以上版本；Windows CE：1.0以上版本：头文件：Winuser.h：库文件：user32.lib。

GetTopWindow   
  
函数功能：该函数检查与特定父窗口相联的子窗口z序，并返回在z序顶部的子窗口的句柄。   
函数原型：HWND GetTopWindow（HWND hWnd）；   
参数：   
hWnd:被查序的父窗口的句柄。如果该参数为NULL，函数返回Z序顶部的窗口句柄。   
返回值；如果函数成功，返回值为在Z序顶部的子窗口句柄。如果指定的窗口无子窗口，返回值为NULL。   
若想获得更多错误信息，请调用GetLastError函数。   
速查：Windows NT：3.1以上版本；Windows：95以上版本：Windows CE：不支持；头文件：Winuser.h：库文件；user32.lib。   
  
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =   
  
GetWindow   
  
函数功能：该函数返回与指定窗口有特定关系（如Z序或所有者）的窗口句柄。   
函数原型：HWND GetWindow（HWND hWnd，UNIT nCmd）；   
参数：   
hWnd:窗口句柄。要获得的窗口句柄是依据nCmd参数值相对于这个窗口的句柄。   
nCmd:说明指定窗口与要获得句柄的窗口之间的关系。该参数值可以是下列之一：   
GW\_CHILD：如果指定窗口是父窗口，则获得的是在Z序顶端的子窗口的句柄，否则为NULL。函数仅检查指定父窗口的子窗口，不检查继承窗口。   
GW\_ENABLEDPOUP：（WindowsNT 5.0）返回的句柄标识了属于指定窗口的处于使能状态弹出式窗口（检索使用第一个由GW\_HWNDNEXT 查找到的满足前述条件的窗口）；如果无使能窗口，则获得的句柄与指定窗口相同。   
GW\_HWNDFIRST：返回的句柄标识了在Z序最高端的相同类型的窗口。如果指定窗口是最高端窗口，则该句柄标识了在Z序最高端的最高端窗口；如果指定窗口是顶层窗口，则该句柄标识了在z序最高端的顶层窗口：如果指定窗口是子窗口，则句柄标识了在Z序最高端的同属窗口。   
GW\_HWNDLAST:返回的句柄标识了在z序最低端的相同类型的窗口。如果指定窗口是最高端窗口，则该柄标识了在z序最低端的最高端窗口：如果指定窗口是顶层窗口，则该句柄标识了在z序最低端的顶层窗口；如果指定窗口是子窗口，则句柄标识了在Z序最低端的同属窗口。   
GW\_HWNDNEXT：返回的句柄标识了在Z序中指定窗口下的相同类型的窗口。如果指定窗口是最高端窗口，则该句柄标识了在指定窗口下的最高端窗口：如果指定窗口是顶层窗口，则该句柄标识了在指定窗口下的顶层窗口；如果指定窗口是子窗口，则句柄标识了在指定窗口下的同属窗口。   
GW HWNDPREV：返回的句柄标识了在Z序中指定窗口上的相同类型的窗口。如果指定窗口是最高端窗口，则该句柄标识了在指定窗口上的最高端窗口；如果指定窗口是顶层窗口，则该句柄标识了在指定窗口上的顶层窗口；如果指定窗口是子窗口，则句柄标识了在指定窗口上的同属窗口。   
GW\_OWNER：返回的句柄标识了指定窗口的所有者窗口（如果存在）。   
返回值：如果函数成功，返回值为窗口句柄；如果与指定窗口有特定关系的窗口不存在，则返回值为NULL。   
若想获得更多错误信息，请调用GetLastError函数。   
备注：在循环体中调用函数EnumChildWindow比调用GetWindow函数可靠。调用GetWindow函数实现该任务的应用程序可能会陷入死循环或退回一个已被销毁的窗口句柄。   
速查：Windows NT：3.1以上版本；Windows：95以上版本；Windows CE：1.0以上版本；头文件：winuser.h；库文件：user32.lib。   
  
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =   
  
GetWindowPlacement   
  
函数功能：该函数返回指定窗口的显示状态以及被恢复的、最大化的和最小化的窗口位置。   
函数原型：BOOL GetWindowPlacement（HWND hWnd，WINDOWPLACEMENT★lpwndpl）;   
参数：   
hWnd: 窗日句柄。   
lpwndpl：指向WINDOWPLACEMENT结构的指针，该结构存贮显示状态和位置信息。   
在调用GetWindowPlacement函数之前，将WINDOWPLACEMENT结构的长度设为   
sizeof（WIDNOWPLACEMENT）。如果lpwndpl－＞length设置不正确则函数GetWindowPlacement将失败。   
返回值；如果函数成功，返回值为非零；如果函数失败，返回值为零。若想获得更多错误信息，请调用GetlastError函数。   
备注：由该函数获得的WINDOWPLACEMENT结构的flag单元总为0。如果hWnd的窗口被最大化，则showCmd元为SHOWMZAXMIZED，如果窗口被最小化，则showCmd元为SHOWMINIMIZED，除此之外为SHOWNORMN，WINDOWPLACEMENT长度单元必须置为sizeOf（ WINDOWPLACEMENT），如果参数设置不正确，函数返回FALSE。查看设置窗口位置坐标的正确信息，参看WINDOWPLACEMENT。   
速查：Windows Nt：3.1以上版本：Windows：95以上版本；Windows CE：不支持；头文件：Winuser.h；库文件：User32.lib。   
  
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =   
  
GetWindowRect   
  
函数功能：该函数返回指定窗口的边框矩形的尺寸。该尺寸以相对于屏幕坐标左上角的屏幕坐标给出。   
函数原型：BOOL GetWindowRect（HWND hWnd，LPRECTlpRect）；   
参数：   
hWnd:窗口句柄。   
lpRect：指向一个RECT结构的指针，该结构接收窗口的左上角和右下角的屏幕坐标。   
返回值：如果函数成功，返回值为非零：如果函数失败，返回值为零。若想获得更多错误信息，请调用GetLastError函数。   
速查：Windows NT：3.1以上版本：Windows：95以上版本；Windows CE：1.0以上版本；头文件：Winuser.h；库文件：User32.lib。

GetWindowText   
  
函数功能：该函数将指定窗口的标题条文本（如果存在）拷贝到一个缓存区内。如果指定的窗口是一个控制，则拷贝控制的文本。但是，GetWindowTeXt不能接收在其他应用程序中的控制文本。   
函数原型：Int GetWindowText（HWND hWnd，LPTSTR lpString，Int nMaxCount）；   
参数：   
hWnd:带文本的窗口或控制的句柄。   
IpString：指向接收文本的缓冲区的指针。   
nMaxCount:指定要保存在缓冲区内的字符的最大个数，其中包含NULL字符。如果文本超过界限，它就被截断。   
返回值：如果函数成功，返回值是拷贝的字符串的字符个数，不包括中断的空字符；如果窗口无标题栏或文本，或标题栏为空，或窗口或控制的句柄无效，则返回值为零。若想获得更多错误信息，请调用GetLastError函数。   
函数不能返回在其他应用程序中的编辑控制的文本。   
备注：如果目标窗口属于当前进程，GetWindowText函数给指定的窗口或控制发送WM\_GETTEXT消息。如果目标窗口属于其他进程，并且有一个窗口标题，则GetWindowTeXt返回窗口的标题文本，如果窗口无标题，则函数返回空字符串。   
速查：Windows NT：3.1以上版本；Windows：95以上版本：Windows CE：1.0以上版本；头文件：Winuser.h;库文件：user32.lib：Unicode:在Windows NT上实现为Unicode和ANSI两种版本。   
  
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =   
  
IsChild   
  
函数功能：该函数测试一个窗口是否是指定父窗口的子窗口或后代窗口。如果该父窗口是在父窗口的链表上则子窗口是指定父窗口的直接后代。父窗口链表从原始层叠窗口或弹出窗口一直连到该子窗口。   
函数原型：BOOL IsChild（HWND hWndParant，HWND hWnd）；   
参数：   
hWndparant:父窗口句柄。   
hWnd：将被测试的窗口句柄。   
返回值：如果窗口是指定窗口的子窗口或后代窗口，则退回值为非零。如果窗口不是指定窗口的子窗口或后代窗口，则退回值为零。   
速查：Windows NT：3.1以上版本：Windows：95以上版本；Windows CE：1.0以上版本；头文件：winuser.h；库文件：user32.Iib。   
  
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =   
  
GetWindowTextLent   
  
函数功能：该函数返回指定窗口的标题文本（如果存在）的字符长度。如果指定窗口是一个控制，函数将返回控制内文本的长度。但是GetWindowTextLength函数不能返回在其他应用程序中的控制的文本长度。   
函数原型：nit GetWindowTextLent（HWND hWnd）；   
参数：   
hWnd：窗口或控制的句柄。   
返回值：如果函数成功，返回值为文本的字符长度。在一定的条件下，返回值可能比实际的文本长度大。请参看说明。如果窗口无文本，返回值为零。若想获得更多错误信息，请调用GetLastError函数。   
备注：如果目标窗口属于当前进程，GetWindowTextLength函数给指定的窗口或控制发送WM\_GETTEXT消息。   
在一定的条件下，函数GetWindowTextLength的返回值可能比实际的文本长度大。这是由于ANSI和Unlcode的混和使用以及系统允许DBCS字符在文本内存在的原因，但是函数返回值要至少与文本的实际长度相等，因此可以利用这一点指导缓存区的分配。在应用程序既使用ANSI函数又使用Unicode的普通对话框时就会有缓存分配的问题；同样，当应用程序在一个Unicode的窗口过程中使用了ANSI的GetWindowTextLength函数，或在一个ANSI的窗口过程中使用了Unicode的GetWindowTextLength函数的时候也有缓存分配的问题。查看ANSI和Vnicode函数，参考Wind32函数prototypes。   
要获得文本的实际长度，使用WM\_GETTEXT, LB\_GETTEXT或CB\_GETLBTBTEXT消息或GetWindowText函数。   
速查：Windows NT：3.1以上版本：Windows：95以上版本：Windows CE：1.0以上版本；头文件：Winuser.h；库文件：user32.lib;Unicode：在Windows NT上实现为山Unicode和ANSI两种版本。   
  
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =   
  
GetWindowThreadProcessld   
  
函数功能：该函数返回创建指定窗口线程的标识和创建窗口的进程的标识符，后一项是可选的。   
函数原型；DWORD GetWindowThreadProcessld（HWND hwnd，LPDWORD lpdwProcessld）；   
参数：   
hWnd:窗口句柄。   
lpdwProcessld:接收进程标识的32位值的地址。如果这个参数不为NULL，GetWindwThreadProcessld将进程标识拷贝到这个32位值中，否则不拷贝。   
返回值：返回值为创建窗口的线程标识。   
速查：Windows NT：3.1以上版本；Windows：95以上版本；Windows CE：1.0以上版本；头文件：winuser.h；库文件；user32.lib。   
  
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =   
  
IsIconic   
  
函数功能：该函数确定给定窗口是否是最小化（图标化）的窗口。   
函数原型：BOOL IsIconic（HWND hWnd）；   
参数：   
hWnd：被测试窗口的句柄。   
返回值：如果窗口已图标化，返回值为非零；如果窗口未图标化，返回值为零。   
速查：Windows NT：3.1以上版本；Windows：95以上版本；Windows CE：不支持；头文件：winuser.h；   
库文件：user32.Iib。   
  
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =   
  
isWindow   
  
函数功能：该函数确定给定的窗口句柄是否识别一个已存在的窗口。   
因数原型：BOOL isWindow（HWND hWnd）；   
参数：   
hWnd:被测试窗口的句柄。   
返回值：如果窗口句柄标识了一个已存在的窗口，返回值为非零；如果窗口句柄未标识一个已存在窗口，返回值为零。   
速查：Windows NT：3.1以上版本；Windows：95以上版本；Windows CE：1.0以上版本；头文件：winuserhs；库文件：User32.lib。

lswindowUnicode   
  
函数功能：该函数确定指定的窗口是否是一个本地Unicode窗口。   
函数原型： BOOL lswindowUnicode（HWND hwndJ；   
参数：   
hWnd:被测试窗口的句柄。   
返回值：如果窗口是一个本地Unicode窗口，返回值为非零；如果窗口不是一个本地Unicode窗口，返回值为零，同时说明窗口是一个ANSI窗口。   
备注；一个窗口的字符集是由函数RegosterClass决定的。如果窗口类是以ANSI版的RegisterClass （ RegjsterClassA）注册的，则窗口字符集是ANSI的；如果窗口类是以Unicode版的Registerclass（ RegisterClassW）注册的，则窗口字符集是Unicode。   
系统为窗口消息自动作Unicode和ANSI的双向翻译。例如，如果一个使用Unicode字符集的窗口测到一个ANSI窗口消息，则系统在调用窗口过程之前先将该消息转换为Unicode消息。系统调用lsWindowUnicode函数决定是否翻译消息。   
速查：Windows NT：3.1以上版本；Windows： 95以上版本；Windows CE：不支持；头文件：Winuser.h；库文件：user32.lib。   
  
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =   
  
IsWindowVisible   
  
函数功能：该函数获得给定窗口的可视状态。；   
函数原型：BOOL IsWindowVisible（HWND hWnd）；   
参数；   
hWnd：被测试窗口的句柄。   
返回值：如果指定的窗口及其父窗口具有WS\_VISIBLE风格，返回值为非零；如果指定的窗口及其父窗口不具有WS\_VISIBLE风格，返回值为零。由于返回值表明了窗口是否具有Ws\_VISIBLE风格，因此，即使该窗口被其他窗口遮盖，函数返回值也为非零。   
备注：窗口的可视状态由WS\_VISIBLE位指示。当设置了WS\_VISIBLE位，窗口就可显示，而且只要窗口具有WS\_VISIBLE风格，任何画在窗口的信息都将被显示。   
速查：Windows NT：3.1以上版本；Windows：95以上版本；Windows CE：1.0以上版本；头文件：winuser.h;库文件：user32.lib。   
  
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =   
  
IsZoomed   
  
函数功能：该函数确定窗口是否是最大化的窗口。   
函数原型：BOOL IsZoomed（HWND hWnd）；   
参数：   
hWnd:被测试窗口的句柄。   
返回值：如果窗口己最大化，则返回值为非零；如果窗口未最大化，则返回值为零。   
速查：Windows NT:3.1以上版本；Windows：95以上版本；Windows CE：不支持；头文件：Winuer.h;库文件：User32.Iib。   
  
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =   
  
MoveWindow   
  
函数功能：该函数改变指定窗口的位置和尺寸。对于顶层窗口，位置和尺寸是相对于屏幕的左上角的：对于子窗口，位置和尺寸是相对于父窗口客户区的左上角坐标的。   
函数原型：BOOL MoveWindow（HWND hWnd.int x.int y,int nWidth,int nHeight,BOOL BRePaint）；   
参数：   
hWnd：窗口句柄。   
x：指定窗口的新位置的左边界。   
Y：指定窗口的新位置的顶部边界。   
nWidth：指定窗口的新的宽度。   
nHaight：指定窗口的新的高度。   
bRepaint:确定窗口是否被刷新。如果该参数为TRUE，窗口接收一个WM\_PAINT消息；如果参数为FALSE，不发生任何刷新动作。它适用于客户区，非客户区（包括标题栏和滚动条），及由于移动子窗口而露出的父窗口的区域。如果参数为FALSE，应用程序就必须明确地使窗口无效或重画该窗口和需要刷新的父窗口。   
返回值：如果函数成功，返回值为非零；如果函数失败，返回值为零。若想获得更多错误信息，请调用GetLastError函数。   
备注：如果bRepaint为TRUE，系统在窗口移动后立即给窗口过程发送WM\_PAINT消息（即由MoveWindow函数调用UPdateWindow函数）。如果bRepaint 为FALSE，系统将WM\_PAINT消息放在该窗口的消息队列中。消息循环只有在派遣完消息队列中的其他消息时才派遣WM\_PAINT消息。   
MoveWindow给窗口发送WM\_WfNOWPOSCHANGING，WM\_WINDOWPOSCHANGED，WM\_MOVE，WM\_SIZE和WM\_NCCALCSIZE消息，   
速查：Windows NT：3.1以上版本：Windows：95以上版本；Windows CE：1.0以上版本：头文件：winuser.h；库文件：user32.lib。   
  
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =   
  
Openlcon   
  
函数功能：该函数将一个最小化窗口恢复到原来的位置和尺寸并且激活该窗口。   
函数原型：BOOL Openlcon（HWNDhWnd）；   
参数：   
hWnd:被恢复与激活的窗口的句柄。   
返回值：如果函数成功，返回值为非零；如果函数失败，返回值为零。若想获得更多错误信息，请调用GetLastError函数。   
备注：Openlcon向给出的窗口发送WM\_QUERYOPEN消息。   
速查：Windows NT:3.1以上版本；Windows:95以上版本；Windows CE：不支持：头文件：winuser.h;库文件：user32.lib。