返回总目录

目　　录

北京希望电子出版社版权所有

附录 A　常用的 Windows API 调用

本附录列出了 PowerBuilder 常用 Windows API 系统调用，同时给出了这些函数的功能、声明格式以及应用示例。下表首先给出常用 API 调用的名称和扼要功能，读者需要详细了解某个函数的声明格式和示例时，可通过库号在本节中找到相应详细说明。

表 A　常用的 Windows API 系统

序号	函数	功能
1	Arc()	在窗口上画一条弧线
2	Beep()	让计算机按指定的频率和周期发声
3	BringWindowToTop()	将窗口放置在最前面
4	Chord()	绘制弦图
5	CloseHandle()	释放打开对象的句柄
6	CloseWindow()	最小化窗口
7	CopyFileA()	复制文件
8	CreateDirectoryA()	创建目录
9	DeleteFileA()	删除文件
10	DeleteMenu()	删除指定菜单的菜单项
11	DestroyWindow()	关闭窗口
12	DllRegisterServer()	触发 OCX 控件完成自注册过程
13	Ellipse()	绘制椭圆
14	ExitWindowsEx()	通知 Windows 关闭操作系统
15	FatalExit()	立即退出应用程序
16	FindWindowA()	根据窗口标题查找窗口
17	FreeLibrary()	从活动内存中卸载一个 DLL
18	GetBkColor()	获得指定窗口的背景颜色
19	GetCapture()	获得鼠标所在窗口的句柄
20	GetComputerNameA()	获得执行应用程序的计算机的名称
21	GetClassNameA()	获得指定窗口或对象的类名
22	GetCurrentDirectoryA()	获得当前工作目录
23	GetCurrentThread()	获得当前线程的句柄
24	GetCursor()	获得光标的句柄
25	GetCursorPos()	获得光标的位置
26	GetDC()	获得指定窗口的设备上下文
27	GetKeyboardState()	获得键盘的状态
28	GetKeyState()	获得键盘上指定按键的状态
29	GetModuleHandleA()	获得活动内存中指定模块或动态链接库的句柄
30	GetPixel()	获得窗口上指定像素的颜色
31	GetSystemMenu()	获得系统菜单或窗口菜单的句柄
32	GetSystemTime()	获得系统时间，并存放到一个结构中

（续表）

序号	函数	功能
33	GetThreadPriority()	获得指定线程的优先级
34	GetSystemMetrics()	获得屏幕的分辨率，以像素为单位
35	GetUserNameA()	获得当前用户的登录名称
36	GetVolumnInformationA()	获得硬盘的参数信息
37	GetWindowsDirectoryA()	获得缺省的 Windows 目录
38	GlobalMemoryStatus()	获得内存的详细信息
39	LineTo()	从当前位置到指定位置绘制一条直线
40	LoadLibraryA()	将 32 位 DLL 加载到活动内存中
41	mciSendStringA()	控制和播放.AVI 文件
42	MessageBoxA()	显示一个消息对话框
43	Mouse_Event()	控制和操作鼠标
44	MoveToEx()	将光标移动到指定位置，同时保存移动前的光标位置
45	MoveWindow()	移动、放大或缩小窗口
46	Pie()	绘制饼图
47	Polygon()	绘制多边形
48	PostMessageA()	向创建指定窗口的线索发送一条消息，但不等待线索处理该消息
49	Rectangle()	绘制矩形
50	ReleaseCapture()	解锁鼠标
51	SendMessageA()	向创建指定窗口的线索发送一条消息，但等待线索处理该消息
52	SetCapture()	锁定鼠标
53	SetComputerNameA()	修改计算机的名称
54	SetCurrentDirectoryA()	设置当前目录
55	SetCursorPos()	设置光标的位置
56	SetFocus()	将输入焦点设置到指定对象或窗口上
57	SetKeyboardState()	设置键盘的状态
58	SetPixel()	设置窗口上指定像素的颜色
59	SetThreadPriority()	设置线程的优先级
60	Sleep()	睡眠指定的时间
61	SndPlaySoundA() WaveOutGetNumDevs()	播放.WAV 文件
62	SwapMouseButton()	交换鼠标的左右按钮
63	WinExec()	运行指定的可执行文件

使用这些函数时，首先将它们声明为外部全局函数，然后在脚本中调用（在下面的所有声明格式中，函数的声明都应该在一行内书写，由于排版原因，本书分写在多行上）。需要注意的是，所有引用类参数（由关键字 Ref 指明）在传递之前都必须分配足够的内存，否则可能引发 GPF 错误（参看函数 GetComputerNameA()的说明）。下面分别介绍这些函数。

北京希望电子出版社版权所有

1、Arc()

功　　能：在窗口上画一条弧线。

声明格式：Function boolean Arc(ulong hwnd,long r1,long r2,long r3,long r4,long a1, long a2,long a3,long a4) Library"Gdi32.dll"

示例

```
Boolean rtn
ulong l_handle, l_device
long lv[8]
l_handle = handle(w_main)            // w_main 是一个示例窗口
l_device = GetDC(l_handle)          //获得窗口的设备上下文
lv[ ]  = {10,40,300,220,0,0,180,0}  //数组赋值
rtn = Arc(l_device, lv[1], lv[2], lv[3], lv[4], lv[5], lv[6], lv[7], lv[8])
```

2、Beep()

功　　能：让计算机按指定的频率和周期发声。

声明格式：Function boolean Beep(long freq,long dur) Library"Kernel32.dll"

示例

```
Boolean rtn
Long ll_freq, ll_dur
ll_freq = 500
ll_dur = 20
rtn = Beep(ll_freq, ll_dur)
```

3、BringWindowToTop()

功　　能：将窗口放置在最前面。

声明格式：Function boolean BringWindowToTop(ulong w_handle) Library"User32.dll"

示例

```
Boolean rtn
ulong l_handle
l_handle = handle(w_win2)  //获得窗口的句柄
rtn = BringWindowToTop(l_handle)
```

4、Chord()

功　　能：绘制弦图。弦图由椭圆的一部分和一个线段组成。

声明格式：Function boolean Chord(ulong hwnd,long x1,long y1,long x2,long y2,long r1, long r2, long r3, long r4) Library"Gdi32.dll"

示例

```
boolean rtn
ulong l_handle, l_device
long lv[8]
l_handle = handle(w_main)
```

北京希望电子出版社版权所有

```
l_device = GetDC(l_handle)
l_device = GetDC(handle(w_main))
lv[ ] = {5,5,200,200,0,0,200,300}
rtn = Chord(l_device, lv[1], lv[2], lv[3], lv[4], lv[5], lv[6], lv[7],
lv[8])
```

5、CloseHandle()

功　　能：释放打开对象的句柄。

声明格式：Function boolean CloseHandle(ulong w_handle) Library"Kernel32.dll"

示例

```
boolean rtn
ulong l_handle
rtn = CloseHandle(l_handle)  //l_handle 为窗口或对象的句柄
```

6、CloseWindow()

功　　能：最小化窗口。

声明格式：Function boolean CloseWindow(ulong w_handle) Library"User32.dll"

示例

```
boolean rtn
ulong l_handle
string ls_wname
ls_wname = "<窗口标题>"  //请替换为要最小化的窗口的准确标题
l_handle = FindWindowA(0, ls_wname)  //由窗口标题找到窗口句柄
rtn = CloseWindow(l_handle)
```

7、CopyFileA()

功　　能：复制文件。源文件和目标文件的名称参数都采用引用方式。如果 flag 参数设置为 True，那么复制文件时不覆盖已有的文件；如果 flag 参数设置为 False，那么复制文件时覆盖已有的文件。

声明格式：Function boolean CopyFileA(ref string cfrom, ref string cto, boolean flag) Library"Kernel32.dll"

示例

```
string l_from, l_to
boolean l_flag, rtn
l_flag = false
l_from = c:\Windows\forest.bmp  //指定源文件
l_to = c:\test.bmp              //指定目标文件
rtn = CopyFileA(l_from, l_to, l_flag)  //复制
MessageBox("复制文件结果", string(rtn))
```

8、CreateDirectoryA()

功　　能：创建目录。第一个参数指明要创建的目录，第二个参数在 Windows NT 中使用，

在 Windows 95、Windows 98 中忽略该参数。下面的示例中，我们在 C 盘的根目录下创建目录 ZCF。

声明格式： Function boolean CreateDirectoryA(ref string pathname, int sa) Library "Kernel32.dll"

示例

```
boolean rtn
string l_dir
l_dir = "C:\ZCF"
rtn = CreateDirectoryA(l_dir, 0)
If rtn then
   MessageBox("成功创建新的目录", "新目录为 C:\ZCF")
else
   MessageBox("创建目录", "创建过程失败")
end if
```

9、DeleteFileA()

功　　能： 删除文件。下面的示例中，用户在单行编辑框中输入一个文件名后，代码将该文件删除。

声明格式： Function boolean DeleteFileA(ref string filename) Library"Kernel32.dll"

示例

```
string l_file
boolean rtn
l_file = string(sle_deletefile.text)
rtn = DeleteFileA(l_file)
MessageBox("删除文件", string(rtn))
```

10、DeleteMenu()

功　　能： 删除指定菜单的菜单项。如果该菜单项下还有子菜单，则函数删除指向子菜单的句柄，并释放子菜单所占用的内存。

声明格式： Function boolean DeleteMenu(ulong mhand, uint upos, uint flag) Library "User32.dll"

示例

```
ulong m_handle
boolean rtn
m_handle = GetSystemMenu(handle(w_main), false) // 首先获得系统菜单的句柄
rtn = DeleteMenu(m_handle, 1, 0)  // 第二个参数（这里为 1）指定要删除菜
                                  //单项在菜单中的位置
      Messagebox("菜单句柄", string(m_handle))
Messagebox("菜单删除结果", string(rtn))
```

11、DestroyWindow()

功　　能： 关闭窗口。该函数向指定窗口发送一条删除窗口的消息。

声明格式： Function boolean DestroyWindow(ulong w_handle) Library "User32.dll"

示例

```
boolean rtn
ulong l_handle
open(w_win2)  // 打开一个测试窗口
l_handle = handle(w_win2)   //获得窗口的句柄
rtn = DestroyWindow(l_handle)  //关闭窗口
```

12、DllRegisterServer()

功　　能： 触发 OCX 控件完成自注册过程。实际使用时，将声明格式中 ocxname 替换为真实的 OCX 的名称。

声明格式： Function long DllRegisterServer() Library"c:\windows\ocxname.ocx"

示例

```
Long ll_rtn
ll_rtn = DllRegisterServer()  //通常返回 0 表示 OCX 已经注册
```

13、Ellipse()

功　　能： 绘制椭圆。

声明格式： Function boolean Ellipse(ulong hwnd,long x1,long y1,long x2,long y2) Library "Gdi32.dll"

示例

```
Boolean rtn
ulong l_handle, l_device
long lv[4]
l_handle = handle(w_main)
l_device = GetDC(l_handle)
lv[ ] = {5,5,300,300}
rtn = Ellipse(l_device, lv[1], lv[2], lv[3], lv[4])
```

14、ExitWindowsEx()

功　　能： 通知 Windows 关闭操作系统。

声明格式： Function boolean ExitWindowsEx(uint dwReserved, uint uReserved) Library "User32.dll"

示例

```
boolean rtn
rtn = ExitWindowsEx(0,0)  // 两个参数 0 告诉 Windows 立即关闭系统
```

15、FatalExit()

功　　能： 立即退出应用程序。不做任何清理工作，各种对象依然保留在内存中。调用该函数时，通常会导致 GPF 错误。一般在调试应用程序时使用。

声明格式： SubRoutine FatalExit(int exitcode) Library"Kernel32.dll"

北京希望电子出版社版权所有

示例

```
int rtn
rtn = MessageBox("该 API 调用将产生 GPF 错误!","一定要做吗？", Exclamation!,
YesNo!,2)
If rtn = 1 Then
                MessageBox("再次警告!","执行该 API 调用后，必须重新启动系统！")
   FatalExit(1)
End If
```

16、FindWindowA()

功　　能：根据窗口标题查找窗口。

声明格式：Function ulong FindWindowA(ulong classname,string windowname) Library "User32.dll"

示例

```
ulong l_handle
string ls_wname
ls_wname = "<窗口标题>"  //比如，无标题 - 记事本
l_handle = FindWindowA(0, ls_wname)
```

17、FreeLibrary()

功　　能：从活动内存中卸载一个 DLL。警告，卸载正在使用的 DLL 时将引发 GPF。

声明格式：SubRoutine FreeLibrary(ulong libhandle) Library "Kernel32.dll"

示例

```
ulong modhandle
modhandle = LoadLibrary("<32 位 dll 文件名>") //通常在另一个事件过程中加载库
FreeLibrary(modhandle)
```

18、GetBkColor()

功　　能：获得指定窗口的背景颜色。

声明格式：Function ulong GetBkColor (ulong hwnd) Library"Gdi32.dll"

示例

```
ulong l_handle, l_device, l_color
l_handle = handle(w_main)
l_device = GetDC(l_handle)  //获得设备上下文
l_color = GetBkColor(l_device)  //获得背景颜色
```

19、GetCapture()

功　　能：获得鼠标所在窗口的句柄。

声明格式：Function ulong GetCapture( ) Library "User32.dll"

示例

```
ulong l_handle
```

```
l_handle = GetCapture( )
```

20、GetComputerNameA()

功　　能：获得执行应用程序的计算机的名称。需要注意的是，名称参数 cname 必须分配足够的内存，否则在退出应用程序时可能会引发 GPF 错误。实际上，所有外部函数的引用类参数都必须分配足够的内存。

声明格式：Function boolean GetComputerNameA(ref string cname,ref long nbuf) Library "Kernel32.dll"

示例

```
string ls_compname
long ll_buf
ll_buf = 25  //设置 cname 参数的长度
ls_compname = space(ll_buf)  //为 ls_compname 参数分配空间
GetComputerNameA(ls_compname, ll_buf)
MessageBox("计算机名称", ls_compname)
```

21、GetClassNameA()

功　　能：获得指定窗口或对象的类名。

声明格式：Function long GetClassNameA(ulong hwnd, ref string cname, int buf) Library "User32.dll"

示例

```
string l_class
long rtn
ulong l_handle
l_handle = handle(w_main)
l_class = space(50)
rtn = GetClassNameA(l_handle,l_class,50)
Messagebox("类名", l_class)
```

22、GetCurrentDirectoryA()

功　　能：获得当前工作目录。

声明格式：Function ulong GetCurrentDirectoryA(ulong BufferLen, ref string currentdir) Library"Kernel32.dll"

示例

```
string ls_curdir
ulong l_buf
l_buf = 100
ls_curdir = space(l_buf)  //为参数分配空间
GetCurrentDirectoryA(l_buf, ls_curdir)
MessageBox("当前目录", ls_curdir)
```

23、GetCurrentThread()

北京希望电子出版社版权所有

功　　能：获得当前线程的句柄。

声明格式：Function ulong GetCurrentThread() Library"Kernel32.dll"

示例

```
ulong rtn
rtn = GetCurrentThread()
MessageBox("当前线程句柄", string(rtn))
```

24、GetCursor()

功　　能：获得光标的句柄。

声明格式：Function ulong GetCursor( ) Library"User32.dll

示例

```
ulong l_cursor
l_cursor = GetCursor( )
```

25、GetCursorPos()

功　　能：获得光标的位置。参数 mousepos 是一个结构，它包括两个分量：long xpos, long ypos。

声明格式：Function boolean GetCursorPos(ref mousepos) Library"User32.dll"

示例

```
mousepos mouseloc
GetCursorPos(mouseloc)
Messagebox("光标位置",
          "X="+string(mouseloc.xpos)+"Y="+string(mouseloc.ypos))
```

26、GetDC()

功　　能：获得指定窗口的设备上下文。如果想执行图形外部函数调用，必须获得设备上下文。

声明格式：Function ulong GetDC(ulong hwnd) library "User32.dll"

示例

```
ulong l_handle, l_device
l_handle = handle(w_main)
l_device = GetDC(l_handle)
MessageBox("设备上下文", string(l_device))
```

27、GetKeyboardState()

功　　能：获得键盘的状态。该函数按照字符的 ASCII 码表示，将键盘上每个键的状态保存到包含 256 个整数元素的数组中。值 0 表示相应键没有被按下。

声明格式：Function boolean GetKeyboardState(ref integer kbarray[256])　Library "User32.dll"

北京希望电子出版社版权所有

示例

```
boolean rtn
integer ipkey[256]
rtn = GetKeyboardState(ipkey)
```

28、GetKeyState()

功　　能：获得键盘上指定按键的状态。值 0 表示相应键没有被按下。

声明格式：Function int GetKeyState(integer VirtualKeycode) Library "User32.dll"

示例

```
int rtn
rtn = GetKeyState(65)  // 65 = A
if rtn = 0 then
  MessageBox("按键状态","字母 A 没有按下！")
else
  MessageBox("按键状态","字母 A 被按下！")
end if
```

29、GetModuleHandleA()

功　　能：获得活动内存中指定模块或动态链接库的句柄。可以使用 FreeLibrary()函数释放该函数获得其句柄的模块或动态链接库。

声明格式：Function long GetModuleHandleA(string modname) Library "Kernel32.dll"

示例

```
ulong rtn
rtn = GetModuleHandleA("User32.dll")
MessageBox("返回代码", string(rtn))
```

30、GetPixel()

功　　能：获得窗口上指定像素的颜色。

声明格式：Function ulong GetPixel(ulong hwnd, long xpos, long ypos) Library "Gdi32.dll"

示例

请参看 SetPixel()函数。

31、GetSystemMenu()

功　　能：获得系统菜单或窗口菜单的句柄。

声明格式：Function boolean GetSystemMenu(ulong mhandle, boolean flag) Library "User32.dll"

示例

```
boolean flag
ulong l_handle, m_handle
```

北京希望电子出版社 版权所有

```
l_handle = handle(w_main)
flag = false
m_handle = GetSystemMenu(l_handle, flag)
Messagebox("返回值", string(m_handle))
```

32、GetSystemTime()

功　　能：获得系统时间，并存放到一个结构中。SystemTime 结构的分量为：uint year、uint month、 uint dayofweek、uint day、uint hour、uint minute、uint second、uint millisecond。

声明格式：SubRoutine GetSystemTime(ref systemtime systimeptr) Library "Kernel32.dll"

示例

```
systemtime s_systime
string l_day, l_date, l_time
GetSystemTime(s_systime)
l_date =  string(s_systime.year) +"/"+ string(s_systime.day) &
          +"/"+string(s_systime.month)  //日期：年/月/日
l_time = string(s_systime.hour) +":"+ string(s_systime.minute) &
    +":"+ string(s_systime.second) +":"+ string(s_systime.millisecond)
     //时：分：秒：毫秒
CHOOSE CASE s_systime.dayofweek
  CASE 1
      l_day = "星期日"
  CASE 2
      l_day = "星期一"
  CASE 3
      l_day = "星期二"
  CASE 4
      l_day = "星期三"
  CASE 5
      l_day = "星期四"
  CASE 6
      l_day = "星期五"
  CASE 7
      l_day = "星期六"
END CHOOSE
Messagebox("系统时间",l_date + "   " + l_day + "   " + l_time)
```

33、GetThreadPriority()

功　　能：获得指定线程的优先级。线程的缺省优先级为 0，大于 0 的值优先级更高，因而线程将获得更多的 CPU 时间。

声明格式：Function int GetThreadPriority(ulong hthread) Library"Kernel32.dll"

示例

```
ulong l_handle
```

北京希望电子出版社版权所有

```
integer rtn
l_handle = GetCurrentThread()
rtn = GetThreadPriority(l_handle)
MessageBox("当前线程的优先级", string(rtn))
```

34、GetSystemMetrics()

功　　能：获得屏幕的分辨率，以像素为单位。请注意，该函数区分大小写，函数名称要严格按照书上的式样书写。

声明格式：Function int GetSystemMetrics(int indexnum) Library"User32.dll"

示例

```
int l_xx, l_yy
l_xx = GetSystemMetrics(0)  //获得 X 方向分辨率
l_yy = GetSystemMetrics(1)  //获得 Y 方向分辨率
Messagebox("屏幕分辨率", string(l_xx) + " , " + string(l_yy))
```

35、GetUserNameA()

功　　能：获得当前用户的登录名称。

声明格式：Function boolean GetUserNameA(ref string uname, ref ulong slength) Library "ADVAPI32.DLL"

示例

```
string  ls_username
string  ls_var
ulong  lu_val
boolean rtn
lu_val = 255
ls_username = Space( 255 )  //分配足够的空间
rtn = GetUserNameA(ls_username, lu_val)
Messagebox("GetUserNameA", "用户名为: " + string(ls_username))
```

36、GetVolumnInformationA()

功　　能：获得硬盘的参数信息。

声明格式：Function boolean GetVolumeInformation(ref string lpRootPathName,ref string lpVolumeNameBuffer,ulong nVolumeNameSize,ref ulong lpVolumeSerialNumber, ref ulong lpMaximumComponentLength,ref ulong lpFileSystemFlags,ref string lpFileSystemNameBuffer, ulong nFileSystemNameSize)　Library "Kernel32.dll"

示例

```
boolean rtn
string lprootpathname = "c:"  //驱动器
string lpVolumeNameBuffer = space(256)  //分配足够的空间
ulong nVolumeNameSize = 256
ulong lpVolumeSerialNumber
```

北京希望电子出版社版权所有

```
ulong lpMaximumComponentLength
setnull(lpVolumeSerialNumber)
lpMaximumComponentLength = 256
ulong lpFileSystemFlags
setnull(lpFileSystemFlags)
string lpFileSystemNameBuffer = space(256)  //分配足够的空间
ulong nFileSystemNameSize = 256
rtn = GetVolumeInformationA(lpRootPathName, lpVolumeNameBuffer, &
nVolumeNameSize,lpVolumeSerialNumber, lpMaximumComponentLength, &
lpFileSystemFlags, lpFileSystemNameBuffer, nFileSystemNameSize)
sle_1.text = lprootpathname  //路径名
sle_2.text = lpVolumeNameBuffer
sle_3.text = string(nVolumeNameSize)
sle_4.text = string(lpVolumeSerialNumber)
sle_5.text = string(lpMaximumComponentLength)
sle_6.text = string(lpFileSystemFlags)
sle_7.text = string(lpFileSystemNameBuffer)
sle_8.text = string(nFileSystemNameSize)
```

37、GetWindowsDirectoryA()

功　　能：获得缺省的 Windows 目录。

声明格式：Function ulong GetWindowsDirectoryA(ref string wdir, ulong buf) Library "Kernel32.dll"

示例

```
ulong l_buf
string windir
l_buf = 144
windir = space(144)
GetWindowsDirectoryA(windir, l_buf)
MessageBox("当前目录", windir)
```

38、GlobalMemoryStatus()

功　　能：获得内存的详细信息。该函数使用的 Memory 结构包括下述分量：ulong m_length、ulong m_loaded、ulong m_totalphys、ulong m_availphys、ulong m_totalpagefile 、 ulong m_availpagefile 、 ulong m_totalvirtual 、 ulong m_availvirtual。

声明格式：SubRoutine GlobalMemoryStatus(ref memory mem2) Library"Kernel32.dll"

示例

```
memory sysmem
GlobalMemoryStatus(sysmem)
Messagebox("内存大小", string(sysmem.m_length))
Messagebox("已加载内存", string(sysmem.m_loaded))
Messagebox("总的物理内存", string(sysmem.m_totalphys))
```

北京希望电子出版社版权所有

```
Messagebox("总的可用内存", string(sysmem.m_availphys))
Messagebox("总的页面数", string(sysmem.m_totalpagefile))
Messagebox("可用页面数", string(sysmem.m_availpagefile))
Messagebox("总的虚拟内存", string(sysmem.m_totalvirtual))
Messagebox("可用虚拟内存", string(sysmem.m_availvirtual))
```

39、LineTo()

功　　能：从当前位置到指定位置绘制一条直线。

声明格式：Function boolean LineTo(ulong hwnd,long wx, long wy) Library"Gdi32.dll"

示例

请参看 MoveToEx()函数。

40、LoadLibraryA()

功　　能：将 32 位 DLL 加载到活动内存中。

声明格式：Function ulong LoadLibraryA(string modname) Library"Kernel32.dll"

示例

```
ulong modhandle
 //当 DLL 库在 DOS 搜索路径上时，可以不指定路径
modhandle = LoadLibraryA("c:\windows\mydll.dll")
If modhandle > 0 Then
   MessageBox("返回值", "加载成功。句柄 = " + string(modhandle))
else
   MessageBox("函数执行结果","不能加载指定模块")
end if
```

41、mciSendStringA()

功　　能：控制和播放.AVI 文件。

声明格式：Function long mciSendStringA(string cmd, REF string rtn, long size, long wnd) Library"winmm.dll"

示例

```
string s_errortext
string filename
filename ="c:\spin.avi"
mciSendStringA ("open "+Filename+" type AVIVideo alias test
wait",s_errortext, 0,0)
mciSendStringA ("Window test handle " + string(handle(w_main)) + "
wait",s_errortext, 0, 0)
mciSendStringA ("Put test destination wait",s_errortext, 0, 0)
mciSendStringA ("Play test wait", s_errortext, 0, 0)
mciSendStringA ("Close test", s_errortext, 0, 0)
```

42、MessageBoxA()

北京希望电子出版社版权所有

功　　能：显示一个消息对话框。

声明格式：Function long MessageBoxA(ulong hwnd, ref string text, ref string title, ulong style) Library"User32.dll"

示例

```
long rtn
ulong handle1, style1
string text1
string title1
handle1 = handle(parent)
text1 = "这是一个 API Messagebox"
title1 = "API MessageBox"
style1 = 0
rtn = MessageBoxA(handle1,text1,title1,style1)
```

43、Mouse_Event()

功　　能：控制和操作鼠标。该函数能够移动鼠标指针、按下鼠标按钮，并且能够完成用户使用鼠标能够完成的一切任务。下面的示例将鼠标指针左移 100 个像素，上移 70 个像素。

声明格式：SubRoutine Mouse_Event(ulong dwflag,ulong dx,ulong dy,ulong cbutton,ulong dwextra) Library"User32.dll"

示例

```
int lflag
lflag = 1  //1 = 移动鼠标指针, 7 = 按下左按钮, 25 = 按下右按钮
mouse_event(lflag,-80,-50,0,0)
```

44、MoveToEx()

功　　能：将光标移动到指定位置，同时保存移动前的光标位置。函数使用的 prepos 结构包含下述分量：long xpos, long ypos。

声明格式：Function boolean MoveToEx(ulong hwnd,long wx, long wy,ref prepos prepos2) Library"Gdi32.dll"

示例

```
ulong l_handle, l_device
prepos previouspos  //用于保存光标移动前的位置
l_handle = handle(w_main)
l_device = GetDC(l_handle)
MoveToEx(l_device,200,200,previouspos)  //将光标移动到（200,200）处
LineTo(l_device,300,300)  //在当前位置和（300,300）之间画一条直线
```

45、MoveWindow()

功　　能：移动、放大或缩小窗口。

声明格式：Function boolean MoveWindow(ulong whand,int wx,int wy,int ww,int wh,boolean

wflag) Library"User32.dll"

示例

```
boolean rtn
ulong l_handle, l_device
l_handle = handle(w_main)
//10-X 位置；20-Y 位置；100-宽度；200-高度
rtn = MoveWindow(l_handle,10,20,100,200,true)
MessageBox("返回值",string(rtn))
```

46、Pie()

功　　能：绘制饼图。

声明格式：Function boolean Pie(ulong hwnd,long x1,long y1,long x2,long y2,long x3,long y3,long x4,long y4) Library"Gdi32.dll"

示例

```
Boolean rtn
ulong l_handle,l_device
long lv[8]
lv[ ] = {10,50,290,220,0,0,80,0}
l_handle = handle(w_main)
l_device  = GetDC(l_handle)
rtn = Pie(l_device,lv[1],lv[2],lv[3],lv[4],lv[5],lv[6],lv[7],lv[8])
```

47、Polygon()

功　　能：绘制多边形。该函数使用的结构 poly 包含下述分量：long xpos[5], long ypos[5]。数组的大小与多边形边的条数相关（示例中边的条数为 5）。

声明格式：Function boolean Polygon(hdc, ref struct poly poly2, int cnt) Library "Gdi32.dll"

示例

```
ulong l_handle, l_device
int pcnt
l_handle = handle(w_main)
l_device = GetDC(l_handle)
pcnt = 5
poly poly3
poly3.xpos[ ] = {50,100,150,200,250}
poly3.ypos[ ] = {50,100,150,200,250}
Polygon(l_device,poly3,pcnt)
```

48、PostMessageA()

功　　能：向创建指定窗口的线索发送一条消息，但不等待线索处理该消息。

声明格式：Function boolean PostMessageA(ulong hwndle,UINT wmsg,ulong wParam,ulong lParam) Library "User32.dll"

北京清鸿电子出版社版权所有

示例

```
ulong l_handle
boolean rtn
l_handle = handle(w_main)
// 61472 = 最小化窗口; 61488 = 最大化窗口; 61728 = 正常窗口
rtn = PostMessageA(l_handle,274,61472,0)
```

49、Rectangle()

功　　能：绘制矩形。

声明格式：Function boolean Rectangle(ulong hwnd,long x1,long y1,long x2,long y2) Library"Gdi32.dll"

示例

```
Boolean rtn
ulong l_handle,l_device
long lv[4]
lv[ ] = { 10,10,275,215}  //矩形的四个顶点
l_handle = handle(w_main)
l_device  = GetDC(l_handle)
rtn = Rectangle(l_device,lv[1],lv[2],lv[3],lv[4])
```

50、ReleaseCapture()

功　　能：解锁鼠标。

声明格式：Function boolean ReleaseCapture( ) Library"User32.dll"

示例

请参看 SetCapture()函数。

51、SendMessageA()

功　　能：向创建指定窗口的线索发送一条消息，但等待线索处理该消息。

声明格式：Function long SendMessageA(ulong hwndle,UINT wmsg,ulong wParam,ulong lParam) Library "User32.dll"

示例

```
ulong l_handle
long rtn
l_handle = handle(w_main)
rtn = SendMessageA(l_handle,274,61728,0)
```

52、SetCapture()

功　　能：锁定鼠标。鼠标锁定后，使用 ReleaseCapture()解锁鼠标。

声明格式：Function ulong SetCapture(ulong a) Library"User32.dll"

示例

```
boolean rtn
```

```
ulong l_loop, u_test, u_long
u_test = handle(parent)
u_long = SetCapture(u_test)  //锁定鼠标
SetPointer(SizeNWSE!)
for l_loop = 1 to 150000  //循环等待
next
rtn = ReleaseCapture( )  //解锁鼠标
```

53、SetComputerNameA()

功　　能：修改计算机的名称。

声明格式：Function boolean SetComputerNameA(ref string cname)Library"Kernel32.dll"

示例

```
boolean rtn
string l_name
l_name = "ZCF 计算机"
rtn = SetComputerNameA(l_name)
if rtn then
   MessageBox("计算机的名称已经修改为“ZCF 计算机”",  &
       "重新启动计算机后该名称发挥作用！")
else
   MessageBox("设置计算机名称", "哎，失败了！")
end if
```

54、SetCurrentDirectoryA()

功　　能：设置当前目录。

声明格式：Function boolean SetCurrentDirectoryA(ref string cdir) Library"Kernel32.dll"

示例

```
boolean rtn
string l_dir
l_dir = "C:\My Documents"
rtn = SetCurrentDirectoryA(l_dir)
MessageBox("SetCurrentDirectory", string(rtn))
```

55、SetCursorPos()

功　　能：设置光标的位置。

声明格式：Function boolean SetCursorPos(int cx, int cy) Library "User32.dll"

示例

```
SetCursorPos(300,350)
Messagebox("光标位置","X =" +string(mouseloc.xpos) + "Y =" &
 + string(mouseloc.ypos))
```

56、SetFocus()

功　　能：将输入焦点设置到指定对象或窗口上.

北京希望电子出版社版权所有

声明格式： SubRoutine SetFocus(long objhandle) Library"User32.dll"

示例

```
SetFocus(handle(sle_1))  // handle（）函数获得单行编辑框控件 sle_1 的句柄
```

57、SetKeyboardState()

功　　能： 设置键盘的状态。值 0 表示相应键没有被按下。

声明格式： Function boolean SetKeyboardState(ref integer kbarray[256]) Library "User32.dll"

示例

```
boolean rtn
rtn = SetKeyboardState(ipkey)
if rtn = false then
  Messagebox("操作失败","加载键盘状态时出错！")
else
  Messagebox("操作成功","成功完成键盘状态加载工作。")
end if
```

58、SetPixel()

功　　能： 设置窗口上指定像素的颜色。

声明格式： Function ulong SetPixel(ulong hwnd, long xpos, long ypos, ulong pcol)Library "Gdi32.dll"

示例

```
long lx, ly
ulong rtn
ulong l_handle, l_device
lx = 200
ly = 200
l_handle = handle(w_main)
l_device = GetDC(l_handle)  //获得设备上下文
rtn = GetPixel(l_device, lx, ly)  //获得指定位置的像素颜色
MessageBox("颜色","位置 x="+string(lx)+",y="+ string(ly)+"的颜色值="+
          string(rtn))
SetPixel(l_device, lx, ly, 0)  //将指定位置像素的颜色设置为黑色（0）
```

59、SetThreadPriority()

功　　能： 设置线程的优先级。线程的缺省优先级为 0。大于 0 的优先级有用更高的优先权。优先级越大，获得的 CPU 时间越多。但是，不应该把优先级设置得过大，否则将导致鼠标停止工作。

声明格式： Function boolean SetThreadPriority(ulong hthread, int npriority) Library "Kernel32.dll"

北京希望电子出版社版权所有

示例

```
ulong l_handle
boolean rtn
l_handle = GetCurrentThread()  //获得当前的优先级
rtn = SetThreadPriority(l_handle, 2) // 将线程的优先级设置为 2
MessageBox("将线程的当前优先级修改为", string(rtn))
```

60、Sleep()

功　　能：睡眠指定的时间。该函数执行过程中，系统将不重绘屏幕。该函数的作用相当于在 PowerBuilder 执行了一个 For ... Next 循环。

声明格式：SubRoutine Sleep(ulong milli) Library"Kernel32.dll"

示例

```
ulong l_delay
l_delay = 5000  //5000 毫秒（5 秒）
Sleep(l_delay)  //睡眠 5 秒钟
```

61、SndPlaySoundA()和 WaveOutGetNumDevs()

功　　能：这两个函数结合起来使用，用于播放.WAV 格式的声音文件。

声明格式：Function boolean SndPlaySoundA(string wavfile, uint flag) Library "Winmm.dll"
Function uint WaveOutGetNumDevs() Library"Winmm.dll"

示例

```
uint lui_NumDevs, l_mode
string ls_file
l_mode = 0
ls_file = string(c:\windows\media\chimes.wav)
lui_NumDevs = WaveOutGetNumDevs()
IF lui_NumDevs > 0 THEN
   SndPlaySoundA(ls_file, l_mode)
END IF
```

62、SwapMouseButton()

功　　能：交换鼠标的左右按钮。让鼠标按钮返回正常状态时，需要再次调用该函数。

声明格式：Function boolean SwapMouseButton(boolean var) Library"User32.dll"

示例

```
boolean rtn, l_mouse
rtn = SwapMouseButton(l_mouse)
If l_mouse = true Then
  MessageBox("交换鼠标按钮","左边的按钮发挥右按钮的作用")
Else
  MessageBox("交换鼠标按钮","右边的按钮发挥右按钮的作用")
End If
```

北京希望电子出版社 版权所有

63、WinExec()

功　　能：运行指定的可执行文件。

声明格式：Function uint WinExec(ref string filename, uint wstyle) Library"Kernel32.dll"

示例

```
string ls_filename
uint rtn, wstyle
ls_filename = "c:\windows\calc.exe"  //程序名称
wstyle = 1
rtn = WinExec(ls_filename, wstyle)  //运行指定程序
Messagebox("返回值", string(rtn))
```

北京希望电子出版社版权所有