

**VBLib**  
release 1f  
December 1998

Freeware @1998  
All rights reserved

Programmed by Benoît Prézeau

The product is provided as is and by using it you accept the responsibility of what happens as a result. The OCX has been fully tested and should work as described. If you find any bug, please e-mail me with the pertinent information.

Although this program is Freeware, please take the time to e-mail me you're name, where you're from (country, province/state), what kind of program you're making and what VBLib properties you will be using. Thanks!

## **A general purpose visual basic OCX library**

### **Distribution**

The required files for distribution of your applications are VBLib.OCX and CpuInf32.DLL

### **Required Files**

In order for the demo to work, you need the SP3 (Service Pack 3) release of Visual Basic 5.

### **Changes since 1e**

Added CTRL\_ALT\_DEL  
Added FindFiles and FindFilesStop.  
Added CpuID.  
Added CpuCount.  
Added CPUDesc.  
Added CPUMMX.  
Added CPUFPU.  
Added CpuSpeed.  
Added CpuRSpeed.  
Changed status from Shareware to Freeware.

### **Changes since 1d**

Added PlayWav.  
Added get/set num lock, caps lock, scroll lock key states. (You can do a light show!)  
Added start Connect To dialog box.  
Added Drag & Drop files from Windows Explorer

### **Changes since 1c**

Ini files can now be accessed in a single line.  
Registry can now be accessed in a single line.  
DosEnvVar can now be accessed in a single line.  
FileNameShort can now be accessed in a single line.  
FilePathShort can now be accessed in a single line.  
FileType can now be accessed in a single line.  
FontTTName can now be accessed in a single line.

### **Changes since 1b**

Added Resolution change and Joystick informations.

## **Properties**

### **Associate**

#### Description:

Creates a document type association.

#### Parameter format:

extension,program name,program path and executable file name

#### Parameter type:

string

#### Example:

```
VBLib1.Associate = ".txt,Notepad,c:\windows\notepad.exe ' Associates the .txt extension with the notepad
```

### **CDDoor**

#### Description:

Open or closes the CD-Drive door.

#### Parameter type:

boolean

#### Example:

```
VBLib1.CDDoor = True ' Opens the CD-Drive door  
VBLib1.CDDoor = False ' Closes the CD-Drive door
```

### **ComputerName**

#### Description:

Sets or returns the name of the computer.

#### Parameter type:

string

#### Example:

```
VBLib1.ComputerName = "JohnDoe" ' Sets a new computer name  
V_CompName = VBLib1.ComputerName ' Gets the computer name
```

### **ConnectTo**

#### Description:

Starts the "Connect to" dialog box. Note that the character case of the connection name is not ignored.

#### Parameter type:

string

#### Example:

```
Call VBLib1.ConnectTo(VBLib1.RegistryRead(CURRENT_USER, "RemoteAccess",  
"Default"))  
' Starts the default connection  
Call VBLib1.ConnectTo("My Connection")  
' Starts My Connection
```

### **ControlPanel**

Description:

Activates a control panel window.

Parameter type:

integer

Example:

```
VBLib1.ControlPanel = Display_Appearance ' Shows the Display window with the  
Appearance tag selected
```

**CPUCount**

Description:

Returns the number of CPUs.

Parameter type:

integer

Example:

```
msgbox(str$(VBLib1.CPUCount()))
```

**CPUDesc**

Description:

Returns the CPU's Description. Either as full or short description. Set the parameter to True to get a full description.

Parameter format:

Boolean

Parameter type:

String

Example:

```
MsgBox (VBLib1.CPUDesc(True))
```

**CPUID**

Description:

Returns the CPU ID Support.

Parameter type:

Boolean

Example:

```
MsgBox (VBLib1.CPUID)
```

**CPUFPU**

Description:

Returns whether the CPU has floating point support

Parameter type:

Boolean

Example:

```
MsgBox (VBLib1.CPUFPU)
```

## **CPUMMX**

### Description:

Returns whether the CPU has MMX support

### Parameter type:

Boolean

### Example:

```
MsgBox (VBLib1.CPUMMX)
```

## **CPURSpeed**

### Description:

Returns the CPU's Raw Speed.

### Parameter type:

integer

### Example:

```
msgbox(str$(VBLib1.CPURSpeed()))
```

## **CPUSpeed**

### Description:

Returns the CPU's Normal Speed.

### Parameter type:

integer

### Example:

```
msgbox(str$(VBLib1.CPUSpeed()))
```

## **CTRL ALT DEL**

### Description:

Enables/Disables the CTRL-ALT-DEL keys detection.

### Parameter type:

boolean

### Example:

```
VBLib1.CTRL_ALT_DEL = False
```

## **CursorPos**

### Description:

Sets or returns the current mouse pointer position.

### Parameter format:

x,y

### Parameter type:

string

### Example:

```
v_OldPosition = VBLib1.CursorPos ' Saves the cursor position
```

`VBLib1.CursorPos = v_OldPosition` ' Retrieves the cursor position

### **DateFormat**

Description:

Regional setting date format.

Parameter type:

string

Example:

```
MsgBox (Format$(Now, VBLib1.DateFormat))
```

### **DateMask**

Description:

Regional setting date mask. Use with the VB MaskedTextBox control for a date input from the user.

Parameter type:

string

Example:

```
MaskedTextBox1.Mask = VBLib1.DateMask
```

### **DirectXVer**

Description:

Direct X version.

Parameter type:

string

Example:

```
MsgBox (VBLib1.DirectXVer)
```

### **Disk**

Description:

Select the disk to get the information from for the following properties.

Parameter type:

one character string

Example:

```
VBLib1.Disk="C"
```

### **DiskLabel**

Description:

Returns the disk label. The Disk property must have been set prior to using this property.

Parameter type:

string

Example:

```
m_DiskLabel = VBLib1.DiskLabel
```

### **DiskSerial**

Description:

Returns the disk serial number. The Disk property must have been set prior to using this property.

Parameter type:

long

Example:

```
m_DiskSerial = VBLib1.DiskSerial
```

**DiskSpace**

Description:

Returns the disk total capacity. The Disk property must have been set prior to using this property.

Parameter type:

long

Example:

```
m_DiskSpace = VBLib1.DiskSpace
```

**DiskUsed**

Description:

Returns the disk used space. The Disk property must have been set prior to using this property.

Parameter type:

long

Example:

```
m_DiskSpace = VBLib1.DiskSpace
```

**DocExecute**

Description:

Opens a document with the program associated to that document's extension.

Parameter format:

Document path and file name

Parameter type:

string

Example:

```
VBLib1.DocExecute = VBLib1.PathWindows + "circles.bmp" ' Opens the file  
circles.bmp located in the windows directory.
```

**DosEnvVar**

Description:

Returns the value of a DOS environment variable.

Parameter format:

Variable name

Parameter type:

string

**Example:**

```
msgbox (VBLib1.DOSEnvVar("Path"))
```

**DragFile**

**Description:**

Prepares an object to accept or stop accepting files dragged from Windows Explorer.

**Parameter format:**

Handle to the object (hwnd), Accept or Stop accept

**Example:**

```
Call VBLib1.DragFile(Form1.hWnd, True)
```

**DragFileCheck**

**Description:**

Check if a file has been dragged from Windows Explorer. If multiple files have been dragged, each file name is separated by a column ":".

**Parameter type:**

String

**Example:**

```
' Select form1 as the container object
Call VBLib1.DragFile(Form1.hWnd, True)

Do
    ' Check if a file has been dragged
    Temp$ = VBLib1.DragFileCheck

    If Temp$ <> "" Then MsgBox (Temp$)
    DoEvents
Loop Until Temp$ <> ""

' Stop accepting files
Call VBLib1.DragFile(Form1.hWnd, False)
```

**DriveTypes**

**Description:**

Returns the types of drives on the user's system.

**Parameter type:**

string

**Example:**

```
MsgBox (VBLib1.DriveTypes)
```

**ExtractIcon**

**Description:**

Extracts an icon from a file. For use with the following extract properties.

**Parameter format:**

File path and name of a file to extract the icon from

Parameter type:

string

Example:

```
VBLib1.ExtractIcon = "c:\command.com"
```

### **ExtractLarge**

Description:

Selects a picture to receive a large icon. Set this property prior to using ExtractIcon. The picture's AutoRedraw property must be set to false.

Parameter format:

Picture hDC

Parameter type:

long

Example:

```
VBLib1.ExtractLarge = Picture1.hDC
```

### **ExtractSmall**

Description:

Selects a picture to receive a small icon. Set this property prior to using ExtractIcon. The picture's AutoRedraw property must be set to false.

Parameter format:

Picture hDC

Parameter type:

long

Example:

```
VBLib1.ExtractSmall = Picture2.hDC
```

### **FileNameShort**

Description:

Returns the DOS short file name of a Windows long file name.

Parameter format:

file path and name

Parameter type:

string

Example:

```
msgbox(VBLib1.FileNameShort("c:\windows\carved stone.bmp"))
```

### **FilePathShort**

Description:

Returns the DOS short path name of a Windows long path name.

Parameter format:

directory name



Parameter type:

string

Example:

```
msgbox(VBLib1.FilePathShort("c:\windows\Start Menu"))
```

## **FileType**

Description:

Returns file type of a file (or File Not Found).

Parameter format:

file path and name

Parameter type:

string

Example:

```
msgbox(VBLib1.FileType ("c:\command.com"))
```

## **FindFiles**

Description:

Finds files matching a search pattern. The file names will cause the VBLib1\_FileFound event to be raised with a file name as the returned parameter.

Parameter format:

String

Parameter type:

string

Example:

For this example to work, you must have a listbox named List1 and two command buttons named Command1 and Command2. The list box will display a list of the found files, the command button1 starts the search and the command button2 stops a search if the user wishes it. The two commented lines are to be used if you want a really fast search. But the user won't be able to see the files as they are found.

```
Private Sub Command1_Click()  
    VBLib1.Disk = "C" ' Set the drive to search  
    List1.Clear  
    Command1.Enabled = False  
    ' List1.Visible = False  
    Call VBLib1.FindFiles("*.*.*) ' You can use any search pattern  
    Command1.Enabled = False  
    Label1.Caption = ""  
    ' List1.Visible = True  
End Sub  
  
Private Sub VBLib1_FileFound(ByVal FileFoundName As String)  
    List1.AddItem FileFoundName ' A file was found, add it to the list  
End Sub  
  
Private Sub VBLib1_FileSearchPath(ByVal SearchingPath As String)  
    Label1.Caption = SearchingPath ' Shows the current path being searched  
End Sub
```

```
Private Sub Command2_Click()  
    VBLib1.FindFilesStop = True ' Stop searching by user request  
End Sub
```

### **FindFilesStop**

Description:

Stops a file search by setting to false.

Parameter format:

Boolean

Example:

See the FindFiles example.

### **FontLoad**

Description:

Loads a font from a font file for the current Windows session.

Parameter format:

font file path and name

Parameter type:

string

Example:

```
VBLib1.FontLoad = "c:\fonts\cartoon3.ttf" ' Loads the cartoon font from a  
uninstalled fonts directory
```

### **FontTTName**

Description:

Returns a font's complete name from a font file (or Error).

Parameter format:

font file path and name

Parameter type:

string

Example:

```
msgbox(VBLib1.FontTTName ("c:\windows\fonts\times.ttf"))
```

### **FontUnLoad**

Description:

Unloads a font from a font file that was previously loaded with the FontLoad property.

Parameter format:

font file path and name

Parameter type:

string

Example:

```
VBLib1.FontUnLoad = "c:\fonts\cartoon3.ttf"
```

## **FormatDisk**

### Description:

Shows the explorer's format disk dialog box.

### Parameter type:

boolean

### Example:

```
VBLib1.FormatDisk = True
```

## **IniRead**

### Description:

Reads a key from an ini file.

### Parameter format:

File name, Section name, Key name, Default Value, Key Length

### Parameter type:

string

### Example:

```
msgbox(VBLib1.IniRead("test.ini","Section","Key","none",10))
```

## **IniWrite**

### Description:

Writes a key to an ini file. If the file path is not specified, the ini file will be located in the Windows directory.

### Parameter format:

File name, Section name, Key name, Key Value

### Parameter type:

string

### Example:

```
x% = vbLib1.IniWrite("test.ini","Section","Key","Value")
```

## **JoyAxes**

### Description:

Returns the number of axes.

### Parameter type:

Integer

### Example:

```
MsgBox (VBLib1.JoyAxes)
```

## **JoyButtons**

### Description:

Returns the number of buttons.

### Parameter type:

Integer

Example:

MsgBox (VBLib1.JoyButtons)

**JoyPName**

Description:

Returns the joystick's product name.

Parameter type:

String

Example:

MsgBox (VBLib1.JoyPName)

**JoyRKey**

Description:

Returns the joystick's registry key.

Parameter type:

String

Example:

MsgBox (VBLib1.JoyRKey)

**JoyStats**

Description:

Returns the joystick's minimum and maximum values for each axis.

Parameter type:

String

Example:

MsgBox (VBLib1.JoyStats) ' could return

XMin:0 XMax:65535 YMin:0 YMax:65535 ZMin:0 ZMax:65535 RMin:0 RMax:65535 UMin:0  
UMax:0 VMin:0 VMax:0

**JoyStatus**

Description:

Returns the joystick's status, either "Connected", "Not Connected" or "Error:x".

Parameter type:

String

Example:

MsgBox (VBLib1.JoyStatus)

**Joystick**

Description:

Returns the current Joystick positions and buttons pressed.

The following positions are possible.

X Y Z R U V POV (POV is also known as the Hat)

Parameter type:

String

Example:

```
MsgBox (VBLib1.Joystick) ' could return "X32000 Y32768 Z33280 R31744 POV65535  
B001B002"  
' Joystick buttons 1 and 2 are pressed
```

### **KeyCaps**

Description:

Returns or sets the caps lock key state.

Parameter type:

Boolean

Example:

```
VBLib1.KeyCaps = True  
KeyState = VBLib1.KeyCaps
```

### **KeyNum**

Description:

Returns or sets the num lock key state.

Parameter type:

Boolean

Example:

```
VBLib1.KeyNum = True  
KeyState = VBLib1.KeyNum
```

### **KeyScroll**

Description:

Returns or sets the scroll lock key state.

Parameter type:

Boolean

Example:

```
VBLib1.KeyScroll = True  
KeyState = VBLib1.KeyScroll
```

### **MemoryFree**

Description:

Returns the amount of free memory.

Parameter type:

long

Example:

```
MsgBox (Str$(VBLib1.MemoryFree))
```

### **MemoryTotal**

Description:

Returns the amount of total memory.

Parameter type:

long

Example:

```
MsgBox (Str$(VBLib1.MemoryTotal))
```

**MemoryVFree**

Description:

Returns the amount of free virtual memory.

Parameter type:

long

Example:

```
MsgBox (Str$(VBLib1.MemoryVFree))
```

**MemoryVTotal**

Description:

Returns the amount of total virtual memory.

Parameter type:

long

Example:

```
MsgBox (Str$(VBLib1.MemoryVTotal))
```

**MouseCursorVisible**

Description:

Shows or hides the mouse pointer.

Parameter type:

boolean

Example:

```
VBLib1.MouseCursorVisible = Not VBLib1.MouseCursorVisible
```

**OnTop**

Description:

Makes a form stay on top of the others. Before making a second form stay on top, make sure that the first form has been reset to "normal". Only one form can be made on top at once.

Parameter format:

form handle

Parameter type:

long

Example:

```
If VBLib1.OnTop = 0 Then ' Check if the form is already on top.  
    VBLib1.OnTop = Me.hWnd ' Puts form on top  
Else
```

```
VBLib1.OnTop = 0 ' Puts form as normal  
End If
```

### **PathSystem**

Description:

Returns the windows system path.

Parameter type:

string

Example:

```
MsgBox (VBLib1.PathSystem)
```

### **PathTemp**

Description:

Returns the windows temp path.

Parameter type:

string

Example:

```
MsgBox (VBLib1.PathTemp)
```

### **PathWindows**

Description:

Returns the path of windows.

Parameter type:

string

Example:

```
MsgBox (VBLib1.PathWindows)
```

### **PlayWav**

Description:

Plays a WAV file once or looping.

Parameter format:

File path and name, loop

Example:

```
Call VBLib1.PlayWav ("c:\windows\media\The Microsoft Sound.wav", True)  
'to stop a looping sound, use the following line  
Call VBLib1.PlayWav ("", False)
```

### **ProcessorType**

Description:

Returns the type of processor.

Parameter type:

long

Example:

```
MsgBox (STR$(VBLib1.ProcessorType))
```

### **RebootSystem**

Description:

Reboots the computer.

Parameter type:

boolean

Example:

```
VBLib1.RebootSystem = True  
' The following code will be executed if the reboot was canceled  
MsgBox ("Reboot canceled.")
```

### **Recycle**

Description:

Sends files to the recycle bin. If the user disabled the recycle bin, the files will be deleted.

Parameter format:

file path and name

Parameter type:

string

Example:

```
VBLib1.Recycle = "c:\*.txt" ' Be sure to get the files back!!!
```

### **RecycleName**

Description:

Sets or returns the name of the recycle bin.

Parameter type:

string

Example:

```
VBLib1.RecycleName = "Dumpster"  
MsgBox (VBLib1.RecycleName)
```

### **RegistryRead**

Description:

Reads a registry key.

Parameter format:

HKey, Path, Key

Example:

```
msgbox (VBLib1.RegistryRead (Current_User,"Control Panel\desktop","wallpaper"))
```

### **RegistryWrite**

Description:

Writes to a registry key.

Parameter format:



HKey, Path, Key, NewValue

Example:

```
call vbLib1.RegistryWrite (Current_User, "Control  
Panel\desktop", "wallpaper", "test.bmp")
```

**Resolution**

Description:

Changes the current screen resolution. The change will take effect only if the desired resolution is supported by the display devices and if the new resolution is available in the current screen depth.

Parameter type:

integer

Example:

```
VBLib1.Resolution = 0 'Sets the resolution to 640*480
```

**ScreenDepth**

Description:

Returns the bits per pixel of the current resolution.

Parameter type:

integer

Example:

```
MsgBox (STR$(VBLib1.ScreenDepth))
```

**ScreenSizeX**

Description:

Returns the screen number of pixels horizontally.

Parameter type:

integer

Example:

```
MsgBox (STR$(VBLib1.ScreenSizeX))
```

**ScreenSizeY**

Description:

Returns the screen number of pixels vertically.

Parameter type:

integer

Example:

```
MsgBox (STR$(VBLib1.ScreenSizeY))
```

**ScreenWorkX**

Description:

Returns the available region of the screen. The first value is the left, the second value is the right.

Parameter format:

left,right

Parameter type:

string

Example:

```
MsgBox (STR$(VBLib1.ScreenWorkX))
```

### **ScreenWorkY**

Description:

Returns the available region of the screen. The first value is the top, the second value is the bottom.

Parameter format:

top,bottom

Parameter type:

string

Example:

```
MsgBox (STR$(VBLib1.ScreenWorkY))
```

### **Shutdown**

Description:

Shuts down the computer.

Parameter type:

boolean

Example:

```
VBLib1.Shutdown = True  
' The following code will be executed if the shutdown was canceled  
MsgBox ("Shutdown canceled.")
```

### **TaskBarHwnd**

Description:

Sets the form to be used in the task bar.

This task bar icon property must be the last one to be set.

Parameter format:

form handle

Parameter type:

long

Example:

```
Me.Visible = False ' Hide the form  
VBLib1.TaskBarIcon = "c:\icons\icon.ico" ' Select an icon  
VBLib1.TaskBarHwnd = Me.hWnd  
  
' Create a menu on your form and name it MenuTray  
' Use the following code to show the menu when the user double clicks  
' with the left mouse button.  
  
Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As  
Single)  
    Static Freeze As Boolean
```

```

If Freeze = False Then
    Freeze = True

    Select Case X
        '516,7740 right button down
        '513,7695 left button down
        '517,7755 right button up
        '514,7710 left button up
        '515,7725 left button dbl-click
        '518,7770 right button dbl-click
    Case 515, 7725
        PopupMenu MenuTray, 2
    End Select

    Freeze = False
End If
End Sub

' Use the following code to remove your icon from the tray
' when your application terminates

Private Sub Form_Unload(Cancel As Integer)
    If VBLib1.TaskBarHwnd <> 0 Then VBLib1.TaskBarHwnd = 0
End
End Sub

```

## **TaskBarIcon**

### Description:

Sets the icon to be used in the task bar.

### Parameter format:

icon file path and name

### Parameter type:

string

### Example:

```

VBLib1.TaskBarIcon = "c:\icons\icon.ico" ' refers to the icon file
' to be used in the tray bar.

```

## **TaskBarVisible**

### Description:

Makes the task bar visible or not.

### Parameter type:

boolean

### Example:

```

VBLib1.TaskBarVisible = NOT VBLib1.TaskBarVisible

```

## **UserCompany**

### Description:

Returns the organization name of the user.

### Parameter type:

string

Example:

```
MsgBox (VBLib1.UserCompany)
```

**UserName**

Description:

Returns the name of the user.

Parameter type:

string

Example:

```
MsgBox (VBLib1.UserName)
```

**Wallpaper**

Description:

Returns or sets the name of the wallpaper file.

Parameter format:

wallpaper file name (must be in the windows directory)

Parameter type:

string

Example:

```
MsgBox (VBLib1.Wallpaper)  
VBLib1.Wallpaper = "circles.bmp"
```

**WinBuild, WinPlatform, WinSerial, WinServicePack, WinVersion**

Description:

Returns the corresponding parameters.

Parameter type:

string

Example:

```
MsgBox (VBLib1.WinBuild)  
MsgBox (VBLib1.WinPlatform)  
MsgBox (VBLib1.WinSerial)  
MsgBox (VBLib1.WinServicePack)  
MsgBox (VBLib1.WinVersion)
```