

MCS VB Functions

Welcome to the MCSVBFunctions.DLL help file. This file details the use of the various functions that are available through this ActiveX 32-bit Dynamic Link Library. See here for [Technical Support](#) or for see the section on [Registration](#) for details about this shareware.

For ease of use, the DLL has been designed with specific tools and functions separated into different Classes. The following list show the classes and their associated functions:

AdvDataAccess

- [ApplyACEFilter](#)
- [CloseACEIndex](#)
- [CloseACETable](#)
- [CloseAllACEConnections](#)
- [LocateACERecord](#)
- [OpenACEIndex](#)
- [OpenACETable](#)
- [SeekACERecord](#)

ASPFunctions

- [ComboListDirectory](#)
- [GetCounter](#)
- [GetFileDate](#)
- [ReadTextFile](#)

BitmapFunctions

- [TransparentBit](#)

DateFunctions

- [GetLastDayInMonth](#)
- [IsLeapYear](#)

EncryptFunctions

- [CryptFile](#)
- [DecryptString](#)
- [EncryptString](#)

FormFunctions

- [AddIconToSysTray](#)
- [CenterForm](#)
- [DeleteIconFromSysTray](#)
- [DisplayAbout](#)
- [GetCalDate](#)
- [GetUserDetails](#)

MCSRegisterFunctions

- [CheckValidRegistration](#)
- [GetNewSerialNumber](#)
- [GetValidKey](#)

MDIFunctions

- [DecreaseMIDIVolume](#)
- [DecreaseWavVolume](#)
- [EjectCD](#)
- [IncreaseMIDIVolume](#)

IncreaseWavVolume
NextCDTrack
PlayAVIFile
PlayCD
PlayMIDIFile
PlayWavFile
PrevCDTrack
StopAVI
StopCD
StopMIDI
StopWav

RegistryFunctions

DeleteRegSetting
GetAllRegSettings
GetRegSetting
SaveRegSetting

TextFileFunctions

AddDirSep
AddURLDirSep
CreateINIFile
CreateLogFile
CreateLogFileWithBackup
DirExists
ExtractDirectoryPath
ExtractExtension
ExtractFilename
ExtractFilenameNoExtension
FileExists
IsInLogFile
IsUNCName
ReadINIFile
ReadTextFileString
ReadTextFileVariant
SetFileDate
ShellTextFile
UpdateLogFile
WriteINIFile
WriteINIFileBlankLine

VariousFunctions

DataExists
ExtractDecimal
ExtractRubbish
Sleep

Win32Functions

CopyFiles
GetDiskSerialNumber
GetDiskVolumeFlags
GetDiskVolumeName
GetFileSystemName
GetFreeDiskSpace
GetMaxFilenameLength
GetTempFilename

GetTotalDiskSpace
GetWindowsDir
GetWindowsSysDir
GetWindowsTempDir
GetWindowsVersion
PutWindowOnTop
RemoveWallpaper
SetWallpaper
StripTerminator
WindowsExit

Copyright Management Control Systems Ltd 1998

ComboListDirectory

The ComboListDirectory function will provide an HTML string that lists any files found in the passed directory in the form of a drop down combo box.

```
ComboListDirectory(ByVal strComboName As String, ByVal strDirectory As String,  
Optional ByVal blnExtension As Variant) As Variant
```

Passed Arguments:

strComboName:	The name of the combo box that is returned
strDirectory:	The path of the directory that is to be listed
[blnExtension]:	If TRUE then the list will show the files with their extensions, otherwise just the filename.

Returns:

ComboListDirectory:	A string in HTML format that will provide a combo list box.
---------------------	---

Notes:

- If the function does not find any files then the returned string is empty, i.e. ""
- If an empty string is passed for the name, then the returned combo box is called 'Unnamed'
- The directory path can be passed with or without the final directory separator
- If the extension parameter is not passed then the function will return the list of files without the extension by default.

Example

GetCounter

The GetCounter function is used to return a numerical value from a text file that is used to store the number of hits on a web page.

```
GetCounter(ByVal strFilename As String, Optional ByVal blnIncrement As Variant) As Variant
```

Passed Arguments:

strFilename:	The filename (with directory path) of the counter file to be used
[blnIncrement]:	If TRUE then the counter is incremented by one and the new value output to the file

Returns:

GetCounter:	A variant containing the original value of the counter field found in the passed counter file
-------------	---

Notes:

- If the function does not find the specified counter file, one is created and a value of 1 is written to it
- If the increment parameter is not passed or set to FALSE then the counter value will not be incremented.

Example

GetFileDate

The GetFileDate function is used to return a string containing the file date and time information for the passed file.

```
GetFileDate (ByVal strFilePathAndName As String) As Variant
```

Passed Arguments:

strFilename: The filename (with directory path) of the file to be analysed

Returns:

GetFileDate: A variant containing the date and time information of the passed file

Notes:

- If the function does not find the specified file, the string "File does not exist..." is returned
- The format of the date and time returned is dictated by the settings on the host computer (long format)

Example

Copyright Management Control Systems Ltd 1998

ComboListDirectory Example

The following example is used within Server Side Active Server Pages and VB Script:

```
<%
    dim strCombo
    dim objASPFunctions
    set objASPFunctions=server.createobject("MCSVBFunctions.ASPFunctions")
    strCombo=objASPFunctions.ComboListDirectory ("Files","d:
\development\ibmtransfer\logs\teams")
    if strCombo="" then
%>

<p><font size="4">There are no error logs.</font></p>

<%
    else
%>
<p><font size="4">Please select the team you wish to view:</font></p>
<form method="post" action="display.asp">
<%
    response.write strCombo
%>
<input type="submit" value="Select Team">
</form>
<%
    end if
%>
```

GetCounter Example

The following example is used within Server Side Active Server Pages and VB Script in the Global.asa file:

```
Sub Application_OnStart

    ' declare the counter file...
    application("CounterFile")="c:\inetpub\wwwroot\is\counter.txt"

End Sub

Sub Session_OnStart
    ' create an instance of the ASPFunctions library
    dim objASPFunctions
    set objASPFunctions=server.createObject("ASPFunctions.Functions")
    ' call the GetCounter function
    session("Counter")= objASPFunctions.GetCounter(cstr(Application("CounterFile")),
true)
    set objASPFunctions=nothing
End Sub
```


GetFileDate Example

The following example is used within Server Side Active Server Pages and VB Script:

```
<%
    response.write "<tr>"
    dim strPathAndFilename
    dim objASPFunctions
    set objASPFunctions=server.createobject("ASPFunctions.Functions")
    strPathAndFilename=request.servervariables("Path_Translated")
    response.write "<td><p align='left'>This page was last updated on " &
objASPFunctions.GetFileDate(cstr(strPathAndFilename))
    set objASPFunctions=nothing
    response.write " and uses </td>"
%>
    <td><p align="center"></td>
```

ReadTextFile

The readTextFile function will provide an HTML string formatting the contents of a text file.

```
ReadTextFile(ByVal strFilename As String) As Variant
```

Passed Arguments:

strFilename: The filename (and directory) of the file to be read

Returns:

ReadTextFile: A variant string in HTML format that contains the text information

Notes:

- If the function does not find the file then "File Does Not Exist!!!" is returned
- The text file can contain HTML format codes that will be translated and returned in the string

Example

Copyright Management Control Systems Ltd 1998

ReadTextFile Example

The following example is used within Server Side Active Server Pages and VB Script:

```
<% dim objASPFunctions
    set objASPFunctions=server.createobject("ASPFunctions.Functions")
    response.write objaspfunctions.readtextfile _      ("c:
\inetpub\wwwroot\is\pcdev\demo\readtext\readtextfile.txt")
    set objaspfunctions=nothing
%>
```

CryptFile

The CryptFile function will encrypt the passed file (or, if the file is already encrypted, the function will decrypt the file).

```
CryptFile(ByVal strFileIn As String, ByVal strFileOut As String, ByVal strPassword As String, Optional ByVal blnReplace As Boolean) As Boolean
```

Passed Arguments:

strFileIn:	The filename (with directory path) of the file to be encrypted/ decrypted
strFileOut:	The filename (with directory path) of the file to be created with the encrypted/ decrypted information
strPassword:	The password used to encrypt/ decrypt the file
[blnReplace]:	Optional parameter. If set to TRUE then the original file is deleted and the new encrypted/ decrypted file is renamed to replace it

Returns:

CryptFile:	A boolean value indicating the success or failure of the function
------------	---

Notes:

- If the replace parameter is missing then the default will be used (which will not replace the original file)

Example

CryptFile Example

```
dim objCryptFile as new EncryptFunctions
dim blnSuccess as Boolean

    blnSuccess=objCryptFile.CryptFile ("c:\suspense\orig.txt","c:\suspense\crypt.txt",
-    "MyPassword", True)
    set objCryptFile = Nothing
```

This will encrypt the orig.txt file into crypt.txt (using the password 'MyPassword') then delete the orig.txt file and rename the crypt.txt file to orig.txt.

DecryptString

The DecryptString function will decrypt the passed string using the passed password..

```
DecryptString(ByVal strStringIn As String, ByVal strPassword As String) As String
```

Passed Arguments:

strStringIn:	The string to be decrypted
strPassword:	The password used to decrypt the string

Returns:

DecryptString:	A string containing the decrypted information
----------------	---

Example

Copyright Management Control Systems Ltd 1998

DecryptString Example

```
dim objDecryptString as new EncryptFunctions
dim strDecrypted as String

    strDecrypted=objCryptFile.DecryptString (strEncryptedString, "MyPassword")
    set objDecryptString = Nothing
```

EncryptString

The EncryptString function will encrypt the passed string using the passed password..

```
EncryptString(ByVal strStringIn As String, ByVal strPassword As String) As String
```

Passed Arguments:

strStringIn:	The string to be encrypted
strPassword:	The password used to encrypt the string

Returns:

EncryptString:	A string containing the encrypted information
----------------	---

Example

Copyright Management Control Systems Ltd 1998

EncryptString Example

```
dim objEncryptString as new EncryptFunctions
dim strEncrypted as String

    strEncrypted =objCryptFile.EncryptString  (strDecryptedString, "MyPassword")
set objEncryptString = Nothing
```

AddIconToSysTray

The AddIconToSysTray function will put an icon in the Windows 95 or Windows NT system tray for the application and minimise it.

```
AddIconToSysTray(ByVal frmForm As Object, ByVal strTooltip As String, ByVal lngIcon  
                  As Long) As Boolean
```

Passed Arguments:

frmForm:	The form that is to be added to the system tray and minimised
strTooltip:	The text that is to be shown when the mouse pointer is over the system tray icon
lngIcon:	The long value representing the icon that is to be displayed

Returns:

AddIconToSysTray:	A boolean value indicating the success/ failure of the function
-------------------	---

Notes:

To restore a form that has had it's icon moved to the system tray...you will need to include the following code in the forms code section:

```
Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As  
Single)  
    If Not Me.Visible Then  
        Select Case X  
            Case 7725  
                Me.Visible = True  
                Me.WindowState = vbNormal  
        End Select  
    End If  
End Sub
```

Example

Copyright Management Control Systems Ltd 1998

AddIconToSysTray Example

```
Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
' this code is needed to restore the form...
If Not Me.Visible Then
    Select Case X
        Case 7725
            Me.Visible = True
            Me.WindowState = vbNormal
    End Select
End If
End Sub

Private Sub cmdMinForm_Click()
Dim objF As New FormFunctions
    objF.AddIconToSysTray Me, "Test Application", Me.Icon
    Set objF = Nothing
End Sub
```

CenterForm

The CenterForm function will center the calling form on the users screen

```
CenterForm(ByVal frmForm As Object) As Boolean
```

Passed Arguments:

frmForm:	The form that is to be centered on the screen
----------	---

Returns:

CenterForm:	A boolean value indicating the success/ failure of the function
-------------	---

Example

Copyright Management Control Systems Ltd 1998

CenterForm Example

```
Sub Form_Load ()  
Dim objF as New FormFunctions  
    objF.CenterForm Me  
    set objF = Nothing  
End Sub
```

DeleteIconFromSysTray

The DeleteIconToSysTray function will delete an icon from the Windows 95 or Windows NT system tray .

```
DeleteIconFromSysTray(ByVal frmForm As Object, ByVal lngIcon As Long) As Boolean
```

Passed Arguments:

frmForm:	The form that is to be removed from the system tray
lngIcon:	The long value representing the icon that is to be removed

Returns:

DeleteIconToSysTray:	A boolean value indicating the success/ failure of the function
----------------------	---

Notes:

To restore a form that has had it's icon moved to the system tray...you will need to include the following code in the forms code section:

```
Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If Not Me.Visible Then
        Select Case X
            Case 7725
                Me.Visible = True
                Me.WindowState = vbNormal
        End Select
    End If
End Sub
```

Example

Copyright Management Control Systems Ltd 1998

DeleteIconFromSysTray Example

```
Private Sub cmdMinForm_Click()  
Dim objF As New FormFunctions  
    objF.AddIconToSysTray Me, "Test Application", Me.Icon  
    Set objF = Nothing  
End Sub  
  
Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As  
Single)  
    If Not Me.Visible Then  
        Select Case X  
            Case 7725  
                Me.Visible = True  
                Me.WindowState = vbNormal  
        End Select  
    End If  
End Sub  
  
Private Sub Form_Terminate()  
Dim objF As New FormFunctions  
    objF.DeleteIconFromSysTray Me, Me.Icon  
    Set objF = Nothing  
End Sub
```

CheckValidRegistration

The CheckValidRegistration function will check the passed parameters to see if the codes are applicable to the current MCS Registration routines.

```
CheckValidRegistration(ByVal strSerialNumber As String, ByVal strKey As String) As  
Boolean
```

Passed Arguments:

strSerialNumber:	The serial number issued to the client machine/ user
strKey:	The serial number key issued to the client machine/ user

Returns:

CheckValidRegistration:	A boolean value indicating the success/ failure of the key to serial number check
-------------------------	---

Example

Copyright Management Control Systems Ltd 1998

CheckValidRegistration Example

```
Function Registration(ByVal strPassword As String) As Boolean
Dim objR As New MCSRegisterFunctions
Dim strSerialNumber As String
Dim strKey As String
    strSerialNumber = objR.GetNewSerialNumber("1.01")
    strKey = objR.GetValidKey(strSerialNumber, strPassword)
    Registration = objR.CheckValidRegistration(strSerialNumber, strKey)
    Set objR = Nothing
End Function
```

GetNewSerialNumber

The GetNewSerialNumber function will return an MCS Registration valid serial number

```
GetNewSerialNumber(ByVal strVersion As String) As String
```

Passed Arguments:

strVersion: A string that will prefix the returned serial number string

Returns:

GetNewSerialNumber: A string containing an MCS valid serial number

Notes:

- The version parameter can be an empty string, i.e. ""

Example

Copyright Management Control Systems Ltd 1998

GetNewSerialNumber Example

```
Function Registration(ByVal strPassword As String) As Boolean
Dim objR As New MCSRegisterFunctions
Dim strSerialNumber As String
Dim strKey As String
    strSerialNumber = objR.GetNewSerialNumber("1.01")
    strKey = objR.GetValidKey(strSerialNumber, strPassword)
    Registration = objR.CheckValidRegistration(strSerialNumber, strKey)
    Set objR = Nothing
End Function
```

GetValidKey

The GetValidKey function will return an MCS Key based on the passed valid serial number

```
GetValidKey(ByVal strSerialNumber As String, ByVal strPassword As String) As String
```

Passed Arguments:

strSerialNumber:	A string containing a valid MCS Serial Number
strPassword:	The password used by MCS to issue valid keys <i>(for information about this password, please contact MCS via the Technical Support page)</i>

Returns:

GetValidKey:	A string containing an MCS key for the valid serial number
--------------	--

Example

Copyright Management Control Systems Ltd 1998

GetValidKey Example

```
Function Registration(ByVal strPassword As String) As Boolean
Dim objR As New MCSRegisterFunctions
Dim strSerialNumber As String
Dim strKey As String
    strSerialNumber = objR.GetNewSerialNumber("1.01")
    strKey = objR.GetValidKey(strSerialNumber, strPassword)
    Registration = objR.CheckValidRegistration(strSerialNumber, strKey)
    Set objR = Nothing
End Function
```

Technical Support

For additional information etc. please contact MCS via the following:

Snail Mail:

Management Control Systems Ltd
30 Lilac Grove
Biggleswade
Bedfordshire
SG18 8BP

E.Mail:

gibbsc@hotmail.com

DecreaseMIDIVolume

The DecreaseMIDIVolume function will decrease the client computers set MIDI volume by one.

```
DecreaseMIDIVolume() As Boolean
```

Returns:

DecreaseMIDIVolume: A boolean indicating the success/ failure of the function

Note:

- All MDIFunctions require that valid equipment and drivers are installed

Example

Copyright Management Control Systems Ltd 1998

DecreaseMIDIVolume Example

```
Dim objM as New MDIFunctions
Dim blnSuccess as Boolean
    blnSuccess=objM.DecreaseMIDIVolume
Set objM=Nothing
```


DecreaseWavVolume

The DecreaseWavVolume function will decrease the client computers set WAV volume by one.

```
DecreaseWavVolume () As Boolean
```

Returns:

DecreaseWavVolume: A boolean indicating the success/ failure of the function

Note:

- All MDIFunctions require that valid equipment and drivers are installed

Example

Copyright Management Control Systems Ltd 1998

DecreaseWavVolume Example

```
Dim objM as New MDIFunctions
Dim blnSuccess as Boolean
    blnSuccess=objM.DecreaseWavVolume
Set objM=Nothing
```

EjectCD

The EjectCD function will eject the CD compartment attached to the client computer.

`EjectCD() As Boolean`

Returns:

EjectCD: A boolean indicating the success/ failure of the function

Note:

- All MDIFunctions require that valid equipment and drivers are installed

Example

Copyright Management Control Systems Ltd 1998

EjectCD Example

```
Dim objM as New MDIFunctions
Dim blnSuccess as Boolean
    blnSuccess=objM.EjectCD
    Set objM=Nothing
```

IncreaseMIDIVolume

The IncreaseMIDIVolume function will increase the client computers set MIDI volume by one.

```
IncreaseMIDIVolume() As Boolean
```

Returns:

IncreaseMIDIVolume: A boolean indicating the success/ failure of the function

Note:

- All MDIFunctions require that valid equipment and drivers are installed

Example

Copyright Management Control Systems Ltd 1998

IncreaseMIDIVolume Example

```
Dim objM as New MDIFunctions
Dim blnSuccess as Boolean
    blnSuccess=objM.IncreaseMIDIVolume
Set objM=Nothing
```

IncreaseWavVolume

The IncreaseWavVolume function will increase the client computers set WAV volume by one.

```
IncreaseWavVolume() As Boolean
```

Returns:

IncreaseWavVolume: A boolean indicating the success/ failure of the function

Note:

- All MDIFunctions require that valid equipment and drivers are installed

Example

Copyright Management Control Systems Ltd 1998

IncreaseWavVolume Example

```
Dim objM as New MDIFunctions
Dim blnSuccess as Boolean
    blnSuccess=objM.IncreaseWavVolume
Set objM=Nothing
```


NextCDTrack

The NextCDTrack function will increment the current CD track playing by one (providing that the next track is available).

`NextCDTrack()` As Boolean

Returns:

NextCDTrack: A boolean indicating the success/ failure of the function

Note:

- All MDIFunctions require that valid equipment and drivers are installed

Example

Copyright Management Control Systems Ltd 1998

NextCDTrack Example

```
Dim objM as New MDIFunctions
Dim blnSuccess as Boolean
    blnSuccess=objM.NextCDTrack
    Set objM=Nothing
```

PlayAVIFile

The PlayAVIFile function will play the passed AVI movie in the passed frame.

```
PlayAVIFile(ByVal strFilename As String, ByVal fraFrame As Object) As Boolean
```

Passed Arguments:

strFilename:	The Filename (and directory path) of the AVI file
fraFrame:	The frame container in which to play the AVI file

Returns:

PlayAVIFile:	A boolean value indicating the success/ failure of the function
--------------	---

Notes:

- The frame will have to be manually sized to fit the AVI movie(s)

Example

Copyright Management Control Systems Ltd 1998

PlayAVIFile Example

This assumes that you have set up a frame object on the form called Frame1

```
Dim objM as New MDIFunctions
Dim blnSuccess as Boolean
    blnSuccess=objM.PlayAVIFile ("c:\test\test.avi",Frame1)
Set objM=Nothing
```

PlayCD

The PlayCD function will play the CD currently in the CD drive on the client computer.

`PlayCD() As Boolean`

Returns:

PlayCD: A boolean indicating the success/ failure of the function

Note:

- All MDIFunctions require that valid equipment and drivers are installed

Example

Copyright Management Control Systems Ltd 1998

PlayCD Example

```
Dim objM as New MDIFunctions
Dim blnSuccess as Boolean
    blnSuccess=objM.PlayCD
    Set objM=Nothing
```

PlayMIDIFile

The PlayMIDIFile function will play the passed MIDI file through the computers sound system.

```
PlayMIDIFile(ByVal strFilename As String) As Boolean
```

Passed Arguments:

strFilename:	The Filename (and directory path) of the MIDI file
--------------	--

Returns:

PlayMIDIFile:	A boolean value indicating the success/ failure of the function
---------------	---

Example

Copyright Management Control Systems Ltd 1998

PlayMIDIFile Example

```
Dim objM as New MDIFunctions
Dim blnSuccess as Boolean
    blnSuccess=objM.PlayMIDIFile ("c:\test\test.mid")
    Set objM=Nothing
```


PlayWavFile

The PlayWavFile function will play the passed Wav file through the computers sound system.

```
PlayWavFile(ByVal strFilename As String) As Boolean
```

Passed Arguments:

strFilename:	The Filename (and directory path) of the MIDI file
--------------	--

Returns:

PlayWavFile:	A boolean value indicating the success/ failure of the function
--------------	---

Example

Copyright Management Control Systems Ltd 1998

PlayWavFile Example

```
Dim objM as New MDIFunctions
Dim blnSuccess as Boolean
    blnSuccess=objM.PlayWavFile ("c:\test\test.wav")
    Set objM=Nothing
```

PrevCDTrack

The PrevCDTrack function will decrement the current CD track playing by one (providing that the previous track is available).

`PrevCDTrack()` As Boolean

Returns:

PrevCDTrack: A boolean indicating the success/ failure of the function

Note:

- All MDIFunctions require that valid equipment and drivers are installed

Example

Copyright Management Control Systems Ltd 1998

PrevCDTrack Example

```
Dim objM as New MDIFunctions
Dim blnSuccess as Boolean
    blnSuccess=objM.PrevCDTrack
    Set objM=Nothing
```

StopAVI

The StopAVI function will stop the playing of the AVI movie in the set frame

`StopAVI () As Boolean`

Returns:

StopAVI: A boolean indicating the success/ failure of the function

Note:

- All MDIFunctions require that valid equipment and drivers are installed

Example

Copyright Management Control Systems Ltd 1998

StopAVI Example

```
Dim objM as New MDIFunctions
Dim blnSuccess as Boolean
    blnSuccess=objM.StopAVI
    Set objM=Nothing
```

StopCD

The StopCD function will stop the playing of the computer's CD

`StopCD() As Boolean`

Returns:

StopCD: A boolean indicating the success/ failure of the function

Note:

- All MDIFunctions require that valid equipment and drivers are installed

Example

Copyright Management Control Systems Ltd 1998

StopCD Example

```
Dim objM as New MDIFunctions
Dim blnSuccess as Boolean
    blnSuccess=objM.StopCD
    Set objM=Nothing
```


StopMIDI

The StopMIDI function will stop the playing of the specified MIDI file

```
StopMIDI () As Boolean
```

Returns:

StopMIDI: A boolean indicating the success/ failure of the function

Note:

- All MDIFunctions require that valid equipment and drivers are installed

Example

Copyright Management Control Systems Ltd 1998

StopMIDI Example

```
Dim objM as New MDIFunctions
Dim blnSuccess as Boolean
    blnSuccess=objM.StopMIDI
    Set objM=Nothing
```

StopWav

The StopWav function will stop the playing of the Wav file

```
StopWav() As Boolean
```

Returns:

StopWav: A boolean indicating the success/ failure of the function

Note:

- All MDIFunctions require that valid equipment and drivers are installed

Example

Copyright Management Control Systems Ltd 1998

StopWav Example

```
Dim objM as New MDIFunctions
Dim blnSuccess as Boolean
    blnSuccess=objM.StopWav
Set objM=Nothing
```

DeleteRegSettings

The DeleteRegSettings function will delete either a single registry setting or all of the settings associated with a section.

```
DeleteRegSetting(ByVal strAppName As String, ByVal strSection As String, Optional  
ByVal strKey As String) As Boolean
```

Passed Arguments:

strAppName:	The application name under which all of the registry sections and keys are stored.
strSection:	The section under which the keys are stored.
[strKey] :	Optional. The key to be deleted.

Returns:

DeleteRegSettings:	A boolean value indicating the success/ failure of the function
--------------------	---

Notes:

- If the optional key parameter is not passed then all of the settings under the section will be deleted.

Example

DeleteRegSettings Example

```
Private Sub Form_Load()  
Dim objD As New RegistryFunctions  
Dim blnSuccess As Boolean  
    ' delete the start file associated with this application in the registry  
    blnSuccess = objD.DeleteRegSetting(App.EXENAME, "Start", "Start File")  
    Set objD = Nothing  
End Sub
```

GetAllRegSettings

The GetAllRegSettings function will retrieve all of the settings within a section into an array.

```
GetAllRegSettings (ByVal strAppName As String, ByVal strSection As String) As Variant
```

Passed Arguments:

strAppName:	The application name under which all of the registry sections and keys are stored.
strSection:	The section under which the keys are stored.

Returns:

GetAllRegSettings:	A two dimensional variant array containing the keys and their associated values
--------------------	---

Example

Copyright Management Control Systems Ltd 1998

GetAllRegSettings Example

```
Private Sub Form_Load()  
Dim objD As New RegistryFunctions  
Dim vntArray As Variant  
Dim intLoop As Integer  
    vntArray = objD.GetAllRegSettings("MCSVBFuctions", "Registration")  
    Set objD = Nothing  
    For intLoop = 0 To UBound(vntArray)  
        Debug.Print vntArray(intLoop, 0)  
        Debug.Print vntArray(intLoop, 1)  
    Next  
End Sub
```


GetRegSetting

The GetRegSetting function will retrieve the setting of a key within a section.

```
GetRegSetting (ByVal strAppName As String, ByVal strSection As String, ByVal strKey As String) As Variant
```

Passed Arguments:

strAppName:	The application name under which all of the registry sections and keys are stored.
strSection:	The section under which the keys are stored.
strKey:	The key value to be retrieved

Returns:

GetRegSetting:	A variant containing the associated values
----------------	--

Example

Copyright Management Control Systems Ltd 1998

GetRegSetting Example

```
Private Sub Form_Load()  
Dim objD As New RegistryFunctions  
Dim vntValue As Variant  
Dim intLoop As Integer  
    vntValue = objD.GetRegSetting("MCSVBFuctions", "Registration", "Serial Number")  
    Set objD = Nothing  
    Debug.Print vntValue  
End Sub
```

SaveRegSetting

The SaveRegSetting function will save the setting of a key within the specified section.

```
SaveRegSetting(ByVal strAppName As String, ByVal strSection As String, ByVal strKey  
               As String, ByVal strSetting As String) As Boolean
```

Passed Arguments:

strAppName:	The application name under which all of the registry sections and keys are stored.
strSection:	The section under which the keys are stored.
strKey:	The key value to be stored
strSetting:	The actual value to be stored

Returns:

SaveRegSetting:	A boolean indicating the success/ failure of the function
-----------------	---

Example

Copyright Management Control Systems Ltd 1998

SaveRegSetting Example

```
Private Sub Form_Load()  
Dim objD As New RegistryFunctions  
Dim strMyString As String  
Dim blnSuccess As Boolean  
    strMyString = "This is a test"  
    blnSuccess = objD.SaveRegSetting(App.EXENAME, "Startup", "Values", strMyString)  
    Set objD = Nothing  
End Sub
```

AddDirSep

The AddDirSep function will add a directory backslash ("\") to the passed directory string if it requires one.

```
AddDirSep(ByRef strPathName As String)
```

Passed Arguments:

strPathName:	The directory path (string) to be changed
--------------	---

Returns:

strPathName:	The passed string with the directory separator on the end.
--------------	--

Notes:

- If the string already has the separator then the string is returned unchanged.

Example

Copyright Management Control Systems Ltd 1998

AddDirSep Example

```
Dim objT As New TextFileFunctions
Dim strAppPath As String
    strAppPath = App.Path
    objT.AddDirSep strAppPath
Set objT = Nothing
```

AddURLDirSep

The AddURLDirSep function will add a URL directory backslash ("/") to the passed directory string if it requires one. This is useful if your code requires manipulation of URL addresses for ASP/ HTML coding.

```
AddURLDirSep(ByRef strPathName As String)
```

Passed Arguments:

strPathName:	The directory path (string) to be changed
--------------	---

Returns:

strPathName:	The passed string with the URL directory separator on the end.
--------------	--

Notes:

- If the string already has the separator then the string is returned unchanged.

Example

Copyright Management Control Systems Ltd 1998

AddURLDirSep Example

```
Dim objT As New TextFileFunctions
Dim strAppPath As String
    strAppPath = App.Path
    objT.AddURLDirSep strAppPath
Set objT = Nothing
```


CreateINIFile

The CreateINIFile function will create an INI file with the optional passed header string.

```
CreateINIFile(ByVal strFilename As String, Optional ByVal strHeader As Variant) As Boolean
```

Passed Arguments:

strFilename:	The filename (and directory) of the INI file to create
[strHeader]:	Optional. The header string that will be placed at the top of the INI file.

Returns:

CreateINIFile:	A boolean indicating the success/ failure of the function.
----------------	--

Notes:

- Using the WriteINIFile function will create an INI file if it does not exist, however this will not allow you to put in a specific header.

Example

Copyright Management Control Systems Ltd 1998

CreateINIFile Example

```
Dim objT As New TextFileFunctions
Dim blnSuccess As Boolean
    blnSuccess = objT.CreateINIFile(App.Path & "MyFile.ini", "INI FILE FOR " &
App.EXENAME)
    Set objT = Nothing
```

CreateLogFile

The CreateLogFile function will create a Log file with the optional passed header string.

```
CreateLogFile(ByVal strFilename As String, Optional ByVal strHeader As Variant) As Boolean
```

Passed Arguments:

strFilename:	The filename (and directory) of the INI file to create
[strHeader]:	Optional. The header string that will be placed at the top of the INI file.

Returns:

CreateLogFile:	A boolean indicating the success/ failure of the function.
----------------	--

Example

Copyright Management Control Systems Ltd 1998

CreateLogFile Example

```
Dim objT As New TextFileFunctions
Dim blnSuccess As Boolean
    blnSuccess = objT.CreateLogFile(App.Path & "MyFile.log", "LOG FILE FOR " &
App.EXENAME)
    Set objT = Nothing
```

CreateLogFileWithBackup

The CreateLogFileWithBackup function will create a Log file with the optional passed header string and backup any previous days files in the format "yyyymmdd.log". Backups can be kept for a specific number of days (for example one week) by passing the additional parameter of days (e.g. 7 days will delete any backup log files older than one week).

```
CreateLogFileWithBackup(ByVal strFilename As String, Optional ByVal strHeader As  
Variant, Optional ByVal intDays As Variant) As Boolean
```

Passed Arguments:

strFilename:	The filename (and directory) of the INI file to create
[strHeader]:	Optional. The header string that will be placed at the top of the INI file.
[intDays]:	Optional. The number of days to keep a backup

Returns:

CreateLogFileWithBackup: A boolean indicating the success/ failure of the function.

Example

Copyright Management Control Systems Ltd 1998

CreateLogFileWithBackup Example

```
Dim objT As New TextFileFunctions
Dim blnSuccess As Boolean
    blnSuccess = objT.CreateLogFileWithBackup(App.Path & "MyFile.log", "LOG FILE FOR
" & App.EXENAME, 7)
    Set objT = Nothing
```

DirExists

The DirExists function will check if the directory string passed exists on the client computer.

```
DirExists (ByVal strDirName As String) As Boolean
```

Passed Arguments:

strDirName:	The directory to check for
-------------	----------------------------

Returns:

DirExists:	A boolean indicating whether the directory exists or not
------------	--

Example

Copyright Management Control Systems Ltd 1998

DirExists Example

```
Dim objT As New TextFileFunctions
  If Not objT.DirExists ("c:\temp\myapp") Then
    Mkdir "c:\temp\myapp"
  End If
  Set objT = Nothing
```


ExtractDirectoryPath

The ExtractDirectoryPath function will extract the directory part of the passed string.

```
ExtractDirectoryPath(ByVal strInputFile As String) As String
```

Passed Arguments:

strInputFile:	The filename including the directory
---------------	--------------------------------------

Returns:

ExtractDirectoryPath:	A string containing the extracted directory path
-----------------------	--

Example

Copyright Management Control Systems Ltd 1998

ExtractDirectoryPath Example

```
Dim objT As New TextFileFunctions
Dim strPath As String
    strPath = objT.ExtractDirectoryPath("c:\temp\MyFile.ini")
Set objT = Nothing
MkDir strPath
```

ExtractExtension

The ExtractExtension function will extract the extension part of the passed string.

```
ExtractExtension(ByVal strInputFile As String) As String
```

Passed Arguments:

strInputFile:	The filename including the directory
---------------	--------------------------------------

Returns:

ExtractExtension:	A string containing the extracted extension of the file
-------------------	---

Example

Copyright Management Control Systems Ltd 1998

ExtractExtension Example

```
Dim objT As New TextFileFunctions
Dim strExt As String
    strExt = objT.ExtractExtension ("c:\temp\MyFile.ini")
Set objT = Nothing
```

ExtractFilename

The ExtractFilename function will extract the filename part of the passed string.

```
ExtractFilename(ByVal strInputFile As String) As String
```

Passed Arguments:

strInputFile:	The filename including the directory
---------------	--------------------------------------

Returns:

ExtractExtension:	A string containing the extracted filename of the file
-------------------	--

Example

Copyright Management Control Systems Ltd 1998

ExtractFilename Example

```
Dim objT As New TextFileFunctions
Dim strFilename As String
    strExt = objT.ExtractFilename ("c:\temp\MyFile.ini")
Set objT = Nothing
```

ExtractFilenameNoExtension

The ExtractEFilenameNoExtension function will extract the filename part of the passed string without the extension.

```
ExtractFilenameNoExtension (ByVal strInputFile As String) As String
```

Passed Arguments:

strInputFile:	The filename including the directory
---------------	--------------------------------------

Returns:

ExtractFilenameNoExtension:	A string containing the extracted filename without the extension
-----------------------------	--

Example

Copyright Management Control Systems Ltd 1998

ExtractFilenameNoExtension Example

```
Dim objT As New TextFileFunctions
Dim strExt As String
    strExt = objT.ExtractFilenameNoExtension ("c:\temp\MyFile.ini")
Set objT = Nothing
```


FileExists

The FileExists function will check if the file passed exists on the client computer.

```
FileExists (ByVal strFilename As String) As Boolean
```

Passed Arguments:

strFilename:	The file to check for
--------------	-----------------------

Returns:

FileExists:	A boolean indicating whether the file exists or not
-------------	---

Example

Copyright Management Control Systems Ltd 1998

FileExists Example

```
Dim objT As New TextFileFunctions
  If Not objT.FileExists ("c:\temp\myapp.ini") Then
    objT.CreateLogFile ("c:\temp\myapp.ini","File created on " & now)
  End If
  Set objT = Nothing
```

IsInLogFile

The IsInLogFile function will check a text file for the passed string

```
IsInLogFile(ByVal strLogFile As String, ByVal strSearch As String) As Boolean
```

Passed Arguments:

strLogFile:	The text file to check
strSearch:	The string to search for

Returns:

IsInLogFile:	A boolean indicating whether the search string exists or not
--------------	--

Example

Copyright Management Control Systems Ltd 1998

IsInLogFile Example

```
Dim objT As New TextFileFunctions
Dim strFile As String
Dim blnSuccess As Boolean
    strFile = "c:\temp\myapp.ini"
    blnSuccess = objT.IsInLogFile(strFile, "Error")
    If blnSuccess Then
        MsgBox "There has been an error..."
    End If
Set objT = Nothing
```

IsUNCName

The IsInLogFile function will check a text file for the passed string

```
IsUNCName (ByVal strPathName As String) As Boolean
```

Passed Arguments:

strPathName:	The string to check
--------------	---------------------

Returns:

IsUNCName:	A boolean indicating whether the string is a valid UNC name
------------	---

Example

Copyright Management Control Systems Ltd 1998

IsUNCName Example

```
Dim objT As New TextFileFunctions
Dim strString As String
Dim blnSuccess As Boolean
strString = "http://www.mcs.co.uk"
blnSuccess = objT.IsUNCName(strString)
If blnSuccess Then
    ' open IE4 and go to the site...
End If
Set objT = Nothing
```

ReadINIFile

The ReadINIFile function will read an INI file for the required key

```
ReadIniFile (ByVal strIniFile As String, ByVal strSection As String, ByVal strKey As  
String) As String
```

Passed Arguments:

strIniFile:	The INI file to check
strSection:	The section within the file to check
strKey:	The key within the section to check

Returns:

ReadIniFile:	A string containing the value retrieved from the INI file
--------------	---

Example

Copyright Management Control Systems Ltd 1998

ReadINIFile Example

```
Dim objT As New TextFileFunctions
Dim strString As String
    strString = objT.ReadIniFile("c:\test\app.ini", "Users", "JoBloggs")
    If strString <> "" Then
        ' the user exists...
    End If
Set objT = Nothing
```


ReadTextFileString

The ReadTextFileString function will read a text file into a string

```
ReadTextFileString (ByVal strFilename As String) As String
```

Passed Arguments:

strFilename:	The text file to read
--------------	-----------------------

Returns:

ReadTextFileString:	A string containing the text contained within the text file
---------------------	---

Example

Copyright Management Control Systems Ltd 1998

ReadTextFileString Example

```
Dim objT As New TextFileFunctions
Dim strString As String
strString = objT.ReadTextFileString("c:\test\app.log")
If Instr(strString, "Error") Then
    MsgBox "There was an error..."
End If
Set objT = Nothing
```

ReadTextFileVariant

```
Dim objT As New TextFileFunctions
```

```
Dim lngHandle As Long
```

```
    lngHandle = objT.ShellTextFile(Me, "c:\test.bmp")
```

```
    Set objT = Nothing
```

```
Dim objT As New TextFileFunctions
```

```
Dim lngHandle As Long
```

```
    lngHandle = objT.ShellTextFile(Me, "c:\test.bmp")
```

```
    Set objT = Nothing
```

```
Dim objT As New TextFileFunctions
```

```
Dim lngHandle As Long
```

```
    lngHandle = objT.ShellTextFile(Me, "c:\test.bmp")
```

```
    Set objT = Nothing
```

Copyright Management Control Systems Ltd 1998

ReadTextFileVariant Example

```
Dim objT As New TextFileFunctions
Dim vntArray As Variant
Dim intLoop As Integer
vntArray = objT.ReadTextFileVariant("c:\test\app.log")
If Not IsEmpty(vntArray) Then
    For intLoop = 0 To UBound(vntArray)
        ' perform the functions...
    Next
End If
Set objT = Nothing
```

ShellTextFile

The ShellTextFile function will open the passed file in the associated viewer for that file type.

```
ShellTextFile(ByVal frmForm As Object, ByVal strFilename As String) As Long
```

Passed Arguments:

frmForm:	The calling VB form
strFilename:	The filename to open

Returns:

ShellTextFile:	A long value representing the hWnd value of the shelled application
----------------	---

Example

Copyright Management Control Systems Ltd 1998

ShellTextFile Example

```
Dim objT As New TextFileFunctions
Dim lngHandle As Long
    lngHandle = objT.ShellTextFile(Me, "c:\test.bmp")
Set objT = Nothing
```

SetFileDate

The SetFileDate function will set the date and time of the passed file to that specified in the call

```
SetFileDate(ByVal strFilename As String, ByVal dteDate As Date) As Boolean
```

Passed Arguments:

strFilename:	The filename (with directory) to change the date and time
dteDate:	The date (and time) the file is to be set to

Returns:

SetFileDate:	A boolean value indicating the success or failure of the function
--------------	---

Example

Copyright Management Control Systems Ltd 1998

SetFileDate Example

```
Dim objT As New TextFileFunctions
Dim blnSuccess As Boolean
    blnSuccess = objT.SetFileDate("c:\test\test.fdd", #1/1/1998 12:00:00 PM#)
Set objT = Nothing
```


UpdateLogFile

The UpdateLogFile function will update the log file with the string passed.

```
UpdateLogFile(ByVal strLogFile As String, ByVal strDetails As String) As Boolean
```

Passed Arguments:

strLogFile: The log file to update (with directory)

strDetails: The string to append to the file

Returns:

UpdateLogFile: A boolean indicating the success/failure of the function

Example

Copyright Management Control Systems Ltd 1998

UpdateLogFile Example

```
Dim objT As New TextFileFunctions
Dim blnSuccess As Boolean
Dim strText As String
    strText = "Error 123: This occurred at " & Format(Now, "yyyymmdd")
    blnSuccess = objT.UpdateLogFile("c:\test\test.log", strText)
Set objT = Nothing
```

WriteINIFile

The WriteINIFile function will write a value to the specified section under the specified header in the INI file.

```
WriteINIFile(ByVal strIniFile As String, ByVal strHeader As String, ByVal strSection  
            As String, ByVal strValue As String) As Boolean
```

Passed Arguments:

strIniFile:	The INI filename (with directory)
strHeader:	The header within the INI file
strSection:	The section to which the value applies
strValue:	The value you wish to set

Returns:

WriteINIFile:	A boolean value indicating the success or failure of the function
---------------	---

Notes:

- If the INI file does not exist then the file is created.

Example

Copyright Management Control Systems Ltd 1998

WriteINIFile Example

```
Dim objT As New TextFileFunctions
Dim blnSuccess As Boolean
Dim strUser As String
    strUser = "JoBloggs"
    blnSuccess = objT.WriteINIFile("c:\test\test.ini", "Users", "Administrator",
strUser)
    Set objT = Nothing
```

WriteINIFileBlankLine

The WriteINIFileBlankLine function will write blank line in the specified INI file.

```
WriteINIFileBlankLine(ByVal strIniFile As String) As Boolean
```

Passed Arguments:

strIniFile:	The INI filename (with directory)
-------------	-----------------------------------

Returns:

WriteINIFileBlankLine:	A boolean value indicating the success or failure of the function
------------------------	---

Example

Copyright Management Control Systems Ltd 1998

WriteINIFileBlankLine Example

```
Dim objT As New TextFileFunctions
Dim blnSuccess As Boolean
    blnSuccess = objT.WriteINIFileBlankLine("c:\test\test.ini")
Set objT = Nothing
```

DataExists

The DataExists function returns a boolean value indicating whether the string passed contains data or not.

```
DataExists (ByVal strData As String) As Boolean
```

Passed Arguments:

strData:	The string to check
----------	---------------------

Returns:

DataExists:	A boolean value indicating whether data exists in the string
-------------	--

Example

Copyright Management Control Systems Ltd 1998

DataExists Example

```
Dim objT As New VariousFunctions
Dim blnSuccess As Boolean
    blnSuccess = objT.DataExists(strTestingString)
Set objT = Nothing
```


ExtractDecimal

The ExtractDecimal function returns a string containing the specified number of decimals.

```
ExtractDecimal(ByVal strIN As String, ByVal intDecimals As Integer) As String
```

Passed Arguments:

strIN:	A string containing the number to change
--------	--

intDecimals:	The number of decimals required
--------------	---------------------------------

Returns:

ExtractDecimal:	A string with the specified number of decimals
-----------------	--

Example

Copyright Management Control Systems Ltd 1998

ExtractDecimal Example

```
Dim objT As New VariousFunctions
Dim strVal As String
    strVal = "123.456"
    strVal = objT.ExtractDecimal(strVal, 2)
    ' will return 123.46
Set objT = Nothing
```

Sleep

The Sleep function pauses the computer system for the specified number of seconds.

```
Sleep(Seconds As Double) As Boolean
```

Passed Arguments:

Seconds:	The number of seconds required to 'sleep'
----------	---

Returns:

Sleep:	A boolean indicating the success or failure of the function
--------	---

Example

Copyright Management Control Systems Ltd 1998

Sleep Example

```
Dim objT As New VariousFunctions
Dim blnSuccess As Boolean
    ' pause the computer for 25 seconds...
    blnSuccess = objT.Sleep(25)
Set objT = Nothing
```

CopyFiles

The CopyFiles function copies files (and/or sub directories) to other directories or drives.

```
CopyFiles(srcPath As String, dstPath As String, IncludeSubDirs As Integer, FilePat As String) As Boolean
```

Passed Arguments:

srcPath:	The source directory
dstPath:	The destination directory
IncludeSubDirs:	A boolean indicating whether to copy sub directories or not
FilePat:	A file pattern e.g. "*.txt" will copy all files with an extension of txt

Returns:

CopyFiles:	A boolean indicating the success or failure of the function
------------	---

Example

Copyright Management Control Systems Ltd 1998

CopyFiles Example

```
Dim objW As New Win32Functions
Dim blnSuccess As Boolean
    ' copy all of the files from c:\temp to d:\temp
    blnSuccess = objW.CopyFiles("c:\temp", "d:\temp", True, "*.*")
Set objW = Nothing
```

Registration

This ActiveX 32-bit Dynamic Link Library is shareware and, as such, use of the unregistered DLL is approved for a period of 28 days.

All of the functions are available in the unregistered version although use of a function will result in a 'nag screen' appearing prompting for the registration code. Registration will disable this nag screen.

To register the DLL, please decide whether you will require a single licence or a development licence. Single licences are specific to the installation computer and therefore a licence will be required for each computer that the functions will be required to run. A development licence will provide unlimited use of the DLL on all installation computers.

Registration Details:

Single licence: **£50**

Development Licence: **£150.00**

Currently, MCS does not support on-line registration so please make your cheques payable to: **Management Control Systems Ltd** and send them to the address below:

*Management Control Systems Ltd
c/o 30 Lilac Grove
Biggleswade
Bedfordshire
SG18 8TP*

Depending on the licence agreement, you will be sent either a registration code by either post, E.Mail or Fax (depending on your specified method) or a full unlimited use installation package for the development DLL via post or E.Mail.

Also note that this library is dynamic and therefore additional functionality is constantly being added. Registered users will automatically receive updates via E.Mail or post.

Copyright Management Control Systems Ltd 1998

GetCalDate

The GetCalDate function provides a calendar control through which a date can be selected.

```
GetCalDate() As String
```

Returns:

GetCalDate: A string in containing the date selected.

Example

Copyright Management Control Systems Ltd 1998

GetCalDate Example

```
Dim objD As New DateFunctions
Command1.Caption = objD.GetCalDate
Set objD = Nothing
```

GetLastDayInMonth

The GetLastDayInMonth function will provide an integer specifying the last day in the month passed through the date value.

```
GetLastDayInMonth (ByVal dteDate As Date) As Integer
```

Passed Arguments:

dteDate: The date for which the month is to be checked

Returns:

GetLastDayInMonth: An integer specifying the last day in the month

Example

Copyright Management Control Systems Ltd 1998

GetLastDayInMonth Example

```
Dim objD As New DateFunctions
Dim intLastDay As Integer
    intLastDay = objD.GetLastDayInMonth(Now)
Set objD = Nothing
```

IsLeapYear

The IsLeapYear function will provide a boolean return specifying whether the sent year is a leap year or not.

```
IsLeapYear (ByVal intYear As Integer) As Boolean
```

Passed Arguments:

intYear:	The year to check
----------	-------------------

Returns:

IsLeapYear:	A boolean that is TRUE if the year is a leap year
-------------	---

Example

Copyright Management Control Systems Ltd 1998

IsLeapYear Example

```
Dim objD As New DateFunctions
Dim blnLeapYear As Boolean
    blnLeapYear = objD.IsLeapYear(2000)
Set objD = Nothing
```

GetDiskSerialNumber

The GetDiskSerialNumber function will provide the serial number for the specified drive.

```
GetDiskSerialNumber(ByVal strDrive As String) As String
```

Passed Arguments:

strDrive:	The drive to get the serial number for
-----------	--

Returns:

GetDiskSerialNumber:	A string containing the serial number
----------------------	---------------------------------------

Example

Copyright Management Control Systems Ltd 1998

GetDiskSerialNumber Example

```
Dim objW As New Win32Functions
Command1.Caption = objW.GetDiskSerialNumber("c:\")
Set objW = Nothing
```

GetDiskVolumeFlags

The GetDiskVolumeFlags function will provide a string containing all of the set flags for the passed drive.

```
GetDiskVolumeFlags (ByVal strDrive As String) As String
```

Passed Arguments:

strDrive: The drive to get the flags for

Returns:

GetDiskVolumeFlags: A string containing the volume flags (see notes)

Notes:

- The following flags will be contained within the string if they are applicable:
 - "FS_CASE_IS_PRESERVED"
 - "FS_CASE_SENSITIVE"
 - "FS_UNICODE_STORED_ON_DISK"
 - "FS_PERSISTENT_ACLS"
 - "FS_VOL_IS_COMPRESSED"
 - "FS_FILE_COMPRESSION"

Example

GetDiskVolumeFlags Example

```
Dim objW As New Win32Functions
Dim strFlags As String
    strFlags = objW.GetDiskVolumeFlags("c:\")
Set objW = Nothing
```

GetDiskVolumeName

The GetDiskVolumeName function will provide a string containing the volume name for the passed drive.

```
GetDiskVolumeName (ByVal strDrive As String) As String
```

Passed Arguments:

strDrive: The drive to get the flags for

Returns:

GetDiskVolumeName: A string containing the volume name

Example

Copyright Management Control Systems Ltd 1998

GetDiskVolumeName Example

```
Dim objW As New Win32Functions
Dim strName As String
    strNames = objW.GetDiskVolumeName("c:\")
Set objW = Nothing
```

GetFreeDiskSpace

The GetFreeDiskSpace function will provide an integer specifying how many Mb are left on the drive

```
GetFreeDiskSpace (ByVal strDrive As String) As Integer
```

Passed Arguments:

strDrive: The drive to get the flags for

Returns:

GetFreeDiskSpace: An integer of the remaining Mb free

Example

Copyright Management Control Systems Ltd 1998

GetFreeDiskSpace Example

```
Dim objW As New Win32Functions
Dim intFree As Integer
    intFree = objW.GetFreeDiskSpace("c:\")
Set objW = Nothing
```

GetFileSystemName

The GetFileSystemName function will provide a string indicating the file system in operation on the specified drive

```
GetFileSystemName (ByVal strDrive As String) As String
```

Passed Arguments:

strDrive:	The drive to get the flags for
-----------	--------------------------------

Returns:

GetFileSystemName:	A string specifying the file system of the drive
--------------------	--

Example

Copyright Management Control Systems Ltd 1998

GetFileSystemName Example

```
Dim objW As New Win32Functions
Dim strName As String
    strName = objW.GetFileSystemName("c:\")
Set objW = Nothing
```

GetMaxFilenameLength

The GetMaxFilenameLength function return the value of the maximum filename length for the specified drive.

```
GetMaxFilenameLength (ByVal strDrive As String) As Long
```

Passed Arguments:

strDrive:	The drive to check
-----------	--------------------

Returns:

GetMaxFilenameLength: A long value representing the maximum number of characters for a filename

Example

Copyright Management Control Systems Ltd 1998

GetMaxFilenameLength Example

```
Dim objW As New Win32Functions
Dim lngMax As Long
    lngMax = objW.GetMaxFilenameLength("c:\")
Set objW = Nothing
```

GetTempFilename

The GetTempFilename function will return a temporary filename either on its own or after the specified directory path.

```
GetTempFilename(Optional ByVal strPath As Variant) As String
```

Passed Arguments:

[strPath]:	Optional. A directory path in which the temporary file is to be located
------------	---

Returns:

GetTempFilename:	A string containing the filename and directory path (if specified)
------------------	--

Example

Copyright Management Control Systems Ltd 1998

GetTempFilename Example

```
Dim objW As New Win32Functions
Dim strFile As String
    strFile = objW.GetTempFilename(App.Path)
Set objW = Nothing
```

GetTotalDiskSpace

The GetTotalDiskSpace function returns the total amount of disk space on the specified drive (in Mb).

```
GetTotalDiskSpace(ByVal strDrive As String) As Integer
```

Passed Arguments:

strDrive:	The drive to check
-----------	--------------------

Returns:

GetTotalDiskSpace:	An integer specifying the total number of Mb on the drive
--------------------	---

Example

Copyright Management Control Systems Ltd 1998

GetTotalDiskSpace Example

```
Dim objW As New Win32Functions
Dim intSpace As Integer
    intSpace = objW.GetTotalDiskSpace("c:\")
Set objW = Nothing
```

GetWindowsDir

The GetWindowsDir function returns a string specifying the set Windows directory.

```
GetWindowsDir() As String
```

Returns:

GetWindowsDir: A string specifying the set Windows directory

Example

Copyright Management Control Systems Ltd 1998

GetWindowsDir Example

```
Dim objW As New Win32Functions
Dim strWinDir As String
    strWinDir = objW.GetWindowsDir
Set objW = Nothing
```

GetWindowsSysDir

The GetWindowsSysDir function returns a string specifying the set Windows System directory.

```
GetWindowsSysDir() As String
```

Returns:

GetWindowsSysDir: A string specifying the set Windows system directory

Example

Copyright Management Control Systems Ltd 1998

GetWindowsSysDir Example

```
Dim objW As New Win32Functions
Dim strWinSysDir As String
    strWinSysDir = objW.GetWindowsSysDir
Set objW = Nothing
```

GetWindowsTempDir

The GetWindowsTempDir function returns a string specifying the set Windows Temporary directory.

```
GetWindowsTempDir() As String
```

Returns:

GetWindowsTempDir: A string specifying the set Windows Temporary directory

Example

Copyright Management Control Systems Ltd 1998

GetWindowsTempDir Example

```
Dim objW As New Win32Functions
Dim strWinSysDir As String
    strWinDSysir = objW.GetWindowsSysDir
Set objW = Nothing
```

PutWindowOnTop

The PutWindowOnTop function will put the specified window handle on top of the window order.

```
PutWindowOnTop (ByVal lngFormhWnd As Long) As Boolean
```

Passed Arguments:

lngFormhWnd:	The handle of the form to put to the top
--------------	--

Returns:

PutWindowOnTop:	A boolean indicating the success/ failure of the function
-----------------	---

Example

Copyright Management Control Systems Ltd 1998

PutWindowOnTop Example

```
Dim objW As New Win32Functions
objW.PutWindowOnTop Me.hWnd
Set objW = Nothing
```

RemoveWallpaper

The RemoveWallpaper function will set the current Windows wallpaper selection to {NONE}.

```
RemoveWallpaper() As Long
```

Returns:

RemoveWallpaper: The long value returned from the Windows system call

Example

Copyright Management Control Systems Ltd 1998

RemoveWallpaper Example

```
Dim objW As New Win32Functions  
objW.RemoveWallpaper  
Set objW = Nothing
```

SetWallpaper

The SetWallpaper function will change the current windows wallpaper to that of the specified bitmap

```
SetWallpaper (ByVal strBitmap As String) As Long
```

Passed Arguments:

strBitmap:	The filename (including directory) of the bitmap
------------	--

Returns:

SetWallpaper:	A long value representing the return from the Windows call
---------------	--

Example

Copyright Management Control Systems Ltd 1998

SetWallpaper Example

```
Dim objW As New Win32Functions
  objW.SetWallpaper ("c:\windows\system\mybitmap.bmp")
Set objW = Nothing
```

StripTerminator

The StripTerminator function will remove any terminating zeros from the passed string- usually strings returned from Windows API calls

```
StripTerminator(ByVal strString As String) As String
```

Passed Arguments:

strString:	The string to remove any terminators
------------	--------------------------------------

Returns:

StripTerminator:	The string without zero terminators
------------------	-------------------------------------

Example

Copyright Management Control Systems Ltd 1998

StripTerminator Example

```
Dim objW As New Win32Functions
Dim strNew As String
    strNew = objW.StripTerminator(strWinRetString)
Set objW = Nothing
```

WindowsExit

The WindowsExit function will exit or logoff Windows depending on the value passed.

```
WindowsExit(ByVal intOption As Integer) As Boolean
```

Passed Arguments:

intOption:	The mode in which to exit (see notes)
------------	---------------------------------------

Returns:

WindowsExit:	A boolean value indicating the success/failure of the function
--------------	--

Notes:

- The following are valid options:

0 - Exit Windows
1 - Logoff

Example

Copyright Management Control Systems Ltd 1998

WindowsExit Example

```
Const intLogoff = 1
Dim objW As New Win32Functions
    objW.WindowsExit intLoggoff
Set objW = Nothing
```

TransparentBit

The TransparentBit function will transfer a picture object over another bitmap rendering the background of the source picture transparent.

```
TransparentBit(hDestDC As Long, nDestX As Long, nDestY As Long, nWidth As Long,  
nHeight As Long, hSourceDC As Long, nSourceX As Long, nSourceY As Long, lTransparentColor  
As Long) As Boolean
```

Passed Arguments:

hDestDC:	The handle of the destination picture
nDestX:	The x destination of the source picture in the destination picture
nDesyY:	The y destination of the source picture in the destination picture
nWidth:	The width of the source picture
nHeight:	The height of the source picture
hSourceDC:	The handle of the Source picture
nSourceX:	The source picture start x position
nSourceY:	The source picture start y position
lTransparentColor:	The background color

Returns:

TransparentBit:	A boolean indicating the success/ failure of the function.
-----------------	--

Notes:

- The calling form will have to do some calculations on the two pictures for the function to be successful.

Example

TransparentBit Example

```
Private Declare Function GetPixel Lib "gdi32" (ByVal hdc As Long, ByVal x As Long,
ByVal y As Long) As Long
```

```
Private Sub btnTest_Click()
Dim lTransparentColor As Long
Dim nWidest As Long
Dim nHighest As Long
Dim nDestX As Long
Dim nDestY As Long
Dim hSourceDC As Long
Dim hDestDC As Long
Dim nWidth As Long
Dim nHeight As Long
Dim objB As New BitmapFunctions
    lTransparentColor = GetPixel(picSource.hdc, 1, 1)
    '-- Generate a random coordinate within the destination
    picDest.ScaleMode = 3 '-- Pixels
    picSource.ScaleMode = 3
    nWidth = picSource.ScaleWidth
    nHeight = picSource.ScaleHeight
    nWidest = picDest.ScaleWidth - nWidth
    nHighest = picDest.ScaleHeight - nHeight
    nDestX = Int(nWidest * Rnd + 1)
    nDestY = Int(nHighest * Rnd + 1)
    hSourceDC = picSource.hdc
    hDestDC = picDest.hdc
    '-- Clear the old image and Display the image at the new coordinates
    picDest.Cls
    If objB.TransparentBit(hDestDC, nDestX, nDestY, nWidth, nHeight, hSourceDC, 0, 0,
lTransparentColor) = False Then
        MsgBox "TransBlt was unsuccessful"
    End If
End Sub
```

ApplyACEFilter

The ApplyACEFilter function apply a filter to an open Advantage Database Table.

```
ApplyACEFilter(ByVal lngTableHandle As Long, ByVal strFilter As String) As Boolean
```

Passed Arguments:

lngTableHandle:	The handle of the table to which the filter is to be applied
strFilter:	A valid filter string (according to the Advantage scripting engine).

Returns:

ApplyACEFilter:	A boolean indicating the success/failure of the function
-----------------	--

Notes:

- All ACE and Advantage functions require the Advantage 32 bit Dynamic Link Libraries to be installed on the computer.

Copyright Management Control Systems Ltd 1998

CloseACEIndex

The CloseACEIndex function closes an open Advantage database table index.

```
CloseACEIndex(ByRef lngIndexHandle As Long) As Boolean
```

Passed Arguments:

lngIndexHandle: The handle of the table index to be closed

Returns:

CloseACEIndex: A boolean indicating the success/failure of the function

Notes:

- All ACE and Advantage functions require the Advantage 32 bit Dynamic Link Libraries to be installed on the computer.

Copyright Management Control Systems Ltd 1998

CloseACETable

The CloseACETable function closes an open Advantage database table.

```
CloseACETable(ByRef lngTableHandle As Long) As Boolean
```

Passed Arguments:

lngIndexHandle: The handle of the table to be closed

Returns:

CloseACETable: A boolean indicating the success/failure of the function

Notes:

- All ACE and Advantage functions require the Advantage 32 bit Dynamic Link Libraries to be installed on the computer.

Copyright Management Control Systems Ltd 1998

CloseAllACEConnections

The CloseAllACEConnections function closes all open Advantage database connections.

```
CloseAllACEConnections() As Boolean
```

Returns:

CloseAllACEConnections: A boolean indicating the success/failure of the function

Notes:

- All ACE and Advantage functions require the Advantage 32 bit Dynamic Link Libraries to be installed on the computer.

Copyright Management Control Systems Ltd 1998

LocateACERecord

The LocateACERecord function locates a record in the open database table.

```
LocateACERecord(ByVal lngTableHandle As Long, ByVal strFilter As String, ByRef  
                blnLocateFilter As Boolean) As Boolean
```

Passed Arguments:

lngIndexHandle:	The handle of the table to be searched
strFilter:	An Advantage script engine specific filter string
blnLocateFilter:	A boolean value indicating whether the function is running for the first time or continuing the locate

Returns:

LocateACERecord:	A boolean indicating the success/failure of the locate
------------------	--

Notes:

- All ACE and Advantage functions require the Advantage 32 bit Dynamic Link Libraries to be installed on the computer.

OpenACEIndex

The OpenACEIndex function opens an Advantage database table index.

```
OpenACEIndex (ByVal lngTableHandle As Long, ByVal strIndex As String) As Long
```

Passed Arguments:

lngTableHandle:	The handle of the table
strIndex:	The path to the Index file

Returns:

OpenACEIndex:	A long value containing the handle to the table index
---------------	---

Notes:

- All ACE and Advantage functions require the Advantage 32 bit Dynamic Link Libraries to be installed on the computer.

Copyright Management Control Systems Ltd 1998

OpenACETable

The OpenACETable function opens an Advantage database table.

```
OpenACETable (ByVal strTable As String) As Long
```

Passed Arguments:

strTable:	The path to the table file
-----------	----------------------------

Returns:

OpenACETable:	A long value containing the handle to the table
---------------	---

Notes:

- All ACE and Advantage functions require the Advantage 32 bit Dynamic Link Libraries to be installed on the computer.

Copyright Management Control Systems Ltd 1998

SeekACERecord

The SeekACERecord function performs the fastest search through the records for the passed string in the table index.

```
SeekACERecord(ByVal lngIndexHandle As Long, ByVal strSeek As String) As Boolean
```

Passed Arguments:

lngIndexHandle:	The database table index handle
strSeek:	A string that will be contained within the index (note buffer sizes of the indexes)

Returns:

SeekACERecord:	A boolean indicating whether the record was found or not
----------------	--

Notes:

- All ACE and Advantage functions require the Advantage 32 bit Dynamic Link Libraries to be installed on the computer.

Copyright Management Control Systems Ltd 1998

DisplayAbout

The DisplayAbout function will display a simple about screen with the information passed to it.

```
DisplayAbout (ByVal strCompanyName As String, ByVal strAppName As String, ByVal  
              strVersion As String) As Boolean
```

Passed Arguments:

strCompanyName:	A string containing the company details
StrAppName:	A string containing the application name
strVersion:	A string containing the application version details

Returns:

DisplayAbout:	A boolean value indicating the success/ failure of the function
---------------	---

Example

Copyright Management Control Systems Ltd 1998

GetUserDetails

The GetUserDetails function will display a simple logon screen for a user to enter their details.

```
GetUserDetails(ByRef strUsername As String, ByRef strPassword As String) As Boolean
```

Passed Arguments:

strUsername:	A string to return the username entered
strPassword:	A string to return the password entered

Returns:

GetUserDetails:	A boolean. If false is returned then the Cancel button was pressed.
-----------------	---

Example

Copyright Management Control Systems Ltd 1998

DisplayAbout Example

```
Dim objF As New FormFunctions
    objF.DisplayAbout "MCS Ltd", App.EXENAME, App.Major & "." & App.Minor
Set objF = Nothing
```

GetUserDetails Example

```
Dim obj As New FormFunctions
Dim strUsername As String
Dim strPassword As String
Dim blnSuccess As String
    blnSuccess = obj.GetUserDetails(strUsername, strPassword)
    If blnSuccess then
        ....
    End If
Set obj = Nothing
```

GetWindowsVersion

The GetWindowsVersion function will return a 0 for Windows95 and a 1 for WindowsNT.

```
GetWindowsVersion() As Long
```

Returns:

GetWindowsVersion: A 0 for Windows 95 and a 1 for Windows NT

Example

Copyright Management Control Systems Ltd 1998

GetWindowsVersion Example

```
Dim objW As New Win32Functions
If objW.GetWindowsVersion=0 Then
' windows 96
Else
' windows NT
End If
Set objW = Nothing
```

ExtractRubbish

The ExtractRubbish function will remove all characters that are not numeric or alphabetic from the passed string.

```
ExtractRubbish(ByVal strIN As String) As String
```

Passed Arguments:

strIN:	The string to be changed
--------	--------------------------

Returns:

ExtractRubbish:	The passed string with all non-alphanumeric characters removed.
-----------------	---

Example

Copyright Management Control Systems Ltd 1998

ExtractRubbish Example

```
Dim objV As New VariousFunctions
Dim strTest As String
Dim strReturn As String
    strTest="Ghll" & vbNewLine & Chr$(10)
    strReturn=objV.ExtractRubbish(strTest)
Set objV = Nothing
```


