

Tipp : 1.

'Beispiel : Dialog ueber eine Form zentrieren.

'Hinweis : Verwendet Microsoft Common Dialog Control 6.0 (SP3)

```
Private Sub Command1_Click(Index As Integer)
    Select Case Index
    Case 0
        IHook = SetWindowsHookEx(WH_CALLWNDPROC, AddressOf fHook, _
            App.hInstance, App.ThreadID)
        CommonDialog1.ShowOpen
        If IHook Then UnhookWindowsHookEx IHook
    Case 1
        Unload Me
    End
    End Select
End Sub
```

```
Public Declare Function GetWindowRect Lib "user32" _
    (ByVal hwnd As Long, _
    lpRect As RECT) _
    As Long
```

```
Public Declare Function SetWindowLong Lib "user32" _
    Alias "SetWindowLongA" _
    (ByVal hwnd As Long, _
    ByVal nIndex As Long, _
    ByVal dwNewLong As Long) _
    As Long
```

```
Public Declare Function SetWindowPos Lib "user32" _
    (ByVal hwnd As Long, _
    ByVal hWndInsertAfter As Long, _
    ByVal x As Long, _
    ByVal y As Long, _
    ByVal cx As Long, _
    ByVal cy As Long, _
    ByVal wFlags As Long) _
    As Long
```

```
Public Declare Sub CopyMemory Lib "kernel32" _
    Alias "RtlMoveMemory" _
    (hpvDest As Any, _
    hpvSource As Any, _
    ByVal cbCopy As Long)
```

```
Public Declare Function SetWindowsHookEx Lib "user32" _
    Alias "SetWindowsHookExA" _
    (ByVal idHook As Long, _
    ByVal lpfn As Long, _
    ByVal hmod As Long, _
    ByVal dwThreadId As Long) _
    As Long
```

```
Public Declare Function UnhookWindowsHookEx Lib "user32" _
    (ByVal hHook As Long) _
    As Long
```

```
Public Declare Function CallNextHookEx Lib "user32" _
    (ByVal hHook As Long, _
    ByVal ncode As Long, _
    ByVal wParam As Long, _
    lParam As Any) _
    As Long
```

```
Public Declare Function CallWindowProc Lib "user32" _
    Alias "CallWindowProcA" _
    (ByVal lpPrevWndFunc As Long, _
    ByVal hwnd As Long, _
    ByVal Msg As Long, _
    ByVal wParam As Long, _
    ByVal lParam As Long) _
    As Long
```

```
Public Type RECT
    Left As Long
```

```

Top                As Long
Right              As Long
Bottom            As Long
End Type

```

```

Public Type CWPSTRUCT
  IParam          As Long
  wParam         As Long
  message        As Long
  hwnd           As Long
End Type

```

```

Public Const GWL_WNDPROC = (-4)
Public Const SWP_NOSIZE = &H1
Public Const SWP_NOACTIVATE = &H10
Public Const WH_CALLWNDPROC = 4
Public Const WM_INITDIALOG = &H110

```

```

Public IWndProc    As Long
Public IHook       As Long

```

```

Public ILeft       As Long
Public ITop        As Long

```

```

Public Function fHook(ByVal IHookId As Long, ByVal IwParameter As Long, _
    ByVal IParameter As Long) _
    As Long

```

```

    Dim CWS          As CWPSTRUCT

```

```

    CopyMemory CWS, ByVal IParameter, Len(CWS)

```

```

    Select Case CWS.message

```

```

        Case WM_INITDIALOG

```

```

            IWndProc = SetWindowLong(CWS.hwnd, GWL_WNDPROC, AddressOf fDialogProcess)

```

```

            fHook = CallNextHookEx(IHook, IHookId, IwParameter, ByVal IParameter)

```

```

            UnhookWindowsHookEx IHook

```

```

            IHook = 0

```

```

            Exit Function

```

```

        End Select

```

```

        fHook = CallNextHookEx(IHook, IHookId, IwParameter, ByVal IParameter)

```

```

    End Function

```

```

Public Function fDialogProcess(ByVal IHwnd As Long, _
    ByVal IMessage As Long, _
    ByVal IwParameter As Long, _
    ByVal IParameter As Long) _
    As Long

```

```

    Dim RC          As RECT

```

```

    Select Case IMessage

```

```

        Case WM_INITDIALOG

```

```

            GetWindowRect IHwnd, RC

```

```

            ILeft = (Form1.Left \ Screen.TwipsPerPixelX + _
                (Form1.Width \ Screen.TwipsPerPixelX - _
                (RC.Right - RC.Left)) \ 2)

```

```

            ITop = (Form1.Top \ Screen.TwipsPerPixelY + _
                (Form1.Height \ Screen.TwipsPerPixelY - _
                (RC.Bottom - RC.Top)) \ 2)

```

```

            SetWindowPos IHwnd, 0, ILeft, ITop, 0, 0, SWP_NOSIZE Or _
                SWP_NOACTIVATE

```

```

            SetWindowLong IHwnd, GWL_WNDPROC, IWndProc

```

```

        End Select

```

```

        fDialogProcess = CallWindowProc(IWndProc, IHwnd, IMessage, _
            IwParameter, IParameter)

```

```

    End Function

```

Tipp : 2.

'IN FORM :

'Beispiel: Verwendung der erstellten .DLL

'Hinweis : Erstellen Sie eine Referenz zu der im Beispiel mitgelieferten
' DLL.

'In der DLL sind folgende Sourcen vorhanden :

- '1. Bitmap (LOGO)
- '2. Icon (1)
- '3. Cursor (KREUZ)

- '4. String (100) = www.Visual-Basic5.de
- '5. String (101) = &Ende
- '6. String (102) = outa.space@t-online.de

```
Private Sub Command1_Click()  
    Unload Me  
    Set frmMain = Nothing  
End Sub
```

```
Private Sub Command1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)  
    With Command1  
        .MousePointer = 99  
        .MouseIcon = fLoadGrafik("KREUZ", vbResCursor)  
    End With  
End Sub
```

```
Private Sub Form_Load()  
    sLoadResData  
End Sub
```

```
Private Sub sLoadResData()  
    With Me  
        .Caption = fLoadString(102)  
        .Icon = fLoadGrafik(1, vbResIcon)  
        With Picture1  
            .Picture = fLoadGrafik("LOGO", vbResBitmap)  
        End With  
        With Label1  
            .Caption = fLoadString(100)  
        End With  
    End With  
    With Command1  
        .Caption = fLoadString(101)  
    End With  
    Image1.Picture = fLoadGrafik("KREUZ", vbResCursor)  
End Sub
```

'IN MODUL1 :

```
Public Function fLoadGrafik(vResId As Variant, _  
    IrcLoadPicture _  
    As LoadResConstants) _  
    As IPictureDisp  
  
    Dim objResource As Object  
  
    Set objResource = CreateObject("TestDLL.clsResource")  
    Set fLoadGrafik = objResource.fLoadResourceBitmap(vResId, IrcLoadPicture)  
    Set objResource = Nothing  
End Function
```

```
Public Function fLoadString(vResId As Variant) _  
    As String  
  
    Dim objResource As Object  
  
    Set objResource = CreateObject("TestDLL.clsResource")  
    fLoadString = objResource.fLoadResourceString(vResId)  
    Set objResource = Nothing  
End Function
```

'IN clsResource :

'Beispiel : Resource DLL herstellen.

```
Public Function fLoadResourceBitmap(vResId As Variant, _  
    IrcLoadPicture As LoadResConstants) _  
    As IPictureDisp  
  
    On Error GoTo ErrHandle  
    Set fLoadResourceBitmap = LoadResPicture(vResId, IrcLoadPicture)  
    Exit Function
```

```

ErrHandle:
  Set fLoadResourceBitmap = Nothing
  MsgBox "Fehler : ", , "Error..."
End Function

Public Function fLoadResourceString(vResId As Variant) _
  As String

  On Error GoTo ErrHandle
  fLoadResourceString = LoadResString(vResId)
Exit Function
ErrHandle:
  fLoadResourceString = ""
  MsgBox "Fehler : ", , "Error..."
End Function

```

'Herstellung der in clsResource.vbp verwendeten Resourcedatei :
 'Kompilierung erfolgt mit : rc /r clsResource.rc > clsResource.res
 'clsResource.rc

```

LOGO                BITMAP                DISCARDABLE                "bitmap.bmp"
1                   ICON                  DISCARDABLE                "icon.ico"
KREUZ               CURSOR                DISCARDABLE                "cursor.cur"
STRINGTABLE FIXED IMPURE
BEGIN
  100                "www.Visual-Basic5.de"
  101                "&Ende"
  102                "outa.space@t-online.de"
END

```

Tipp : 3.

'Beispiel : Systemverzeichnis ermitteln und Inhalt in einer
 ' ComboBox anzeigen.

```

Private Declare Function GetSystemDirectory Lib "kernel32" _
  Alias "GetSystemDirectoryA" _
  (ByVal lpBuffer As String, _
  ByVal nSize As Long) _
  As Long

Private Function fAddFilesSysDir()
  Dim sFiles As String
  Dim sBuffer As String

  Screen.MousePointer = vbHourglass
  sBuffer = Space$(256)
  GetSystemDirectory sBuffer, Len(sBuffer)
  sBuffer = Left$(sBuffer, InStr(sBuffer, Chr$(0)) - 1)

  If Right$(sBuffer, 1) <> "\" Then
    sBuffer = sBuffer & "\"
  End If

  sFiles = Dir$(sBuffer & "*. *")
  Do While Len(sFiles)
    Combo1.AddItem sFiles
    sFiles = Dir$
  Loop
  Combo1.ListIndex = 0
  Screen.MousePointer = vbDefault

End Function

Private Sub Command1_Click()
  Unload Me
End Sub

Private Sub Command2_Click()
  fAddFilesSysDir
End Sub

```

Tipp : 4.

'Beispiel : Verwendung der API Funktionen :

- ' 1. GetWindowTextLength
- ' 2. SetWindowText
- ' 3. GetWindowText
- ' 4. GetActiveWindow

```
Private Declare Function GetWindowTextLength Lib "user32" _
    Alias "GetWindowTextLengthA" _
    (ByVal hwnd As Long) _
    As Long
```

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

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

```
Private Declare Function GetActiveWindow Lib "user32" () _
    As Long
```

```
Private Sub Command1_Click()
    'Ermittlung der Anzahl der Titelzeilenzeichen des uebergebenen
    'Fensters.
    Dim IRet As Long
    Dim IActiveWnd As Long

    IActiveWnd = GetActiveWindow()
    IRet = GetWindowTextLength(IActiveWnd)
    Text1 = IRet
End Sub
```

```
Private Sub Command2_Click()
    'Aendern der Titelzeilenbeschriftung des uebergebenen Fensters.
    Dim IActiveWnd As Long
    Dim sNewTitle As String
    sNewTitle = "www.Visual-Basic5.de"

    IActiveWnd = GetActiveWindow()
    vRet = SetWindowText(IActiveWnd, sNewTitle)
End Sub
```

```
Private Sub Command3_Click()
    'Ermittlung des uebergebenen Fenstertitels.
    Dim IActiveWnd As Long
    Dim sBuffer As String
    Dim iChars As Integer

    sBuffer = Space(255)
    IActiveWnd = GetActiveWindow()
    iChars = GetWindowText(IActiveWnd, sBuffer, 255)
    Text1 = fConvertCResValue(sBuffer)
End Sub
```

```
'Zeichenkette konvertieren.
Private Function fConvertCResValue(sToConvert As String) As String
    fConvertCResValue = Left(sToConvert, Len(sToConvert) - 1)
End Function
```

Tipp : 5.

'Beispiel : ASCII-Tabelle ausgeben.

```
Private Sub sPrintAsciiCode(frm As Form)
    Dim n As Integer
    Dim k As Integer
    Me.Cls
    frm.Print
```

```

For n = 32 To 248 Step 8
  For k = 0 To 7
    frm.Print Format(n + k, "@@@@@@") & ": " & Chr(n + k);
  Next k
  Print
Next n
End Sub

```

```

Private Sub Command1_Click()
  Unload Me
End Sub

```

```

Private Sub Form_Paint()
  sPrintAsciiCode Form1
End Sub

```

Tipp : 6.

'Beispiel : Automatisches ausfuellen in einer ComboBox.

```

Private Declare Function SendMessage Lib "user32" _
  Alias "SendMessageA" _
  (ByVal hwnd As Long, _
  ByVal wParam As Long, _
  ByVal lParam As Any) _
  As Long

```

```

Private bEditMode As Boolean

```

```

Private Const CB_FINDSTRING = &H14C
Private Const CB_ERR = (-1)

```

```

Private Sub Command1_Click()
  Unload Me
  End
End Sub

```

```

Private Sub Form_Load()
  Combo1.AddItem "Dell Computers"
  Combo1.AddItem "TFC Computer"
  Combo1.AddItem "AXION Electronics"
  Combo1.AddItem "Wavemaster Soundsystem"
  Combo1.AddItem "Microsoft Visual Basic"
  Combo1.AddItem "Visual-Basic5.de"
  Combo1.AddItem "outa.space@t-online.de"
  Combo1.AddItem "www.vb-5.de"
  Combo1.AddItem "New York am 11.September 2001"
  Combo1.AddItem "Twin Tower"
End Sub

```

```

Private Sub Combo1_Change()
  Dim sInsertString As String
  Dim IRet As Long
  Dim ILength As Long
  Dim sResString As String

  If bEditMode Then
    bEditMode = False
    Exit Sub
  End If
  sInsertString = Combo1.Text
  IRet = SendMessage(Combo1.hwnd, CB_FINDSTRING, -1, _
    ByVal sInsertString)
  If IRet <> CB_ERR Then
    sResString = Combo1.List(IRet)
    ILength = Len(sResString) - Len(sInsertString)
    If ILength <> 0 Then
      bEditMode = True
      Combo1.Select = Right$(sResString, ILength)
      Combo1.SelectStart = Len(sInsertString)
      Combo1.SelectLength = ILength
    End If
  End If
End Sub

```

```

Private Sub Combo1_KeyDown(KeyCode As Integer, Shift As Integer)
    Select Case KeyCode
        Case vbKeyBack
            bEditMode = True
        Case vbKeyDelete
            bEditMode = True
    End Select
End Sub

```

Tipp : 7.

'Beispiel : Farbauswahl. Verwendung eines Arrays zur Farbbeschreibung unter der Nutzung der Basic Funktion QBColor.

```

Private Sub Form_Load()
    Dim n As Integer
    For n = 0 To 15
        Picture1(n).BackColor = QBColor(n)
    Next n
End Sub

Private Sub Picture1_MouseDown(Index As Integer, Button As Integer, _
    Shift As Integer, _
    X As Single, _
    Y As Single)

    Dim varrRes As Variant
    varrRes = Array(" Black", "Dark Blue", _
        "Dark Green", "Dark Cyan", _
        "Dark Red ", "Dark Magenta ", _
        "Dark Yellow", "Button Face", _
        "Dark Gray", "Light Blue", _
        "Light Green", "Light Cyan", _
        "Light Red", "Light Magenta", _
        "Light Yellow", "White")

    If Button = vbLeftButton Then
        Picture3.BackColor = Picture1(Index).BackColor
        Text1 = varrRes(Index)
        Text2 = " " & Index
    ElseIf Button = vbRightButton Then
        Picture2.BackColor = Picture1(Index).BackColor
        Text3 = varrRes(Index)
        Text4 = " " & Index
    End If
    Me.Refresh
End Sub

```

Tipp : 8.

'Beispiel : Access Datenbank : Passwort auslesen.
'Hinweis : Funktionsfaehig mit Access 97

```

Dim sPath As String

Private Sub Command1_Click()
    fGetAccessPassword Text1
End Sub

Private Function fGetAccessPassword(sFileName As String)
    Dim sTmpMid As String * 1
    Dim sTmpInp As String * 1
    Dim lCount As Long
    Dim sRes As String
    Dim sCharSet As String

    sPath = sFileName
    sCharSet = Chr(78) & Chr(134) & _
        Chr(251) & Chr(236) & _
        Chr(55) & Chr(93) & _
        Chr(68) & Chr(156) & _
        Chr(250) & Chr(198) & _
        Chr(94) & Chr(40) & _
        Chr(230) & Chr(19)

```

```

Debug.Print sCharSet

Open sPath For Binary As #1
Seek #1, &H42
For ICount = 1 To 14
    sTmpMid = Mid(sCharSet, ICount, 1)
    sTmpInp = Input(1, 1)
    If (Asc(sTmpMid) Xor Asc(sTmpInp)) <> 0 Then
        sRes = sRes & Chr(Asc(sTmpMid) Xor Asc(sTmpInp))
        Debug.Print sRes
    End If
Next
Close 1

MsgBox "Das Passwort lautet : " & sRes, , "Info..."
End Function

Private Sub Command2_Click()
    Unload Me
End Sub

Private Sub Form_Load()
    Text1 = App.Path & "\test.mdb"
End Sub

```

Tipp : 9.

'Beispiel : Digitale Uhr.

```

Dim vTime          As Variant
Dim sDate          As String
Dim ICount         As Long
Dim IHours         As Long
Dim IMinutes       As Long
Dim ISeconds       As Long

Private Sub sInit()
    Dim vMonths As Variant
    Dim vDayName As Variant

    vDayName = Array("Sonntag", "Montag", "Dienstag", "Mittwoch", "Donnerstag", "Freitag", "Samstag")
    vMonths = Array("Januar", "Februar", "Maerz", "April", "Mai", "Juni", "Juli", "August", "September", "Oktober", "November",
"Dezember")
    sDate = ""
    sDate = sDate & vDayName(WeekDay(Now) - 1) & " " & Day(Now) & " " & vMonths(Month(Now) - 1) & " " & Year(Now)
    Frame1.Caption = sDate
    For ICount = 0 To 59
        shpSeconds(ICount).BackColor = QBColor(1)
    Next ICount
    For ICount = 0 To 59
        shpMinutes(ICount).BackColor = QBColor(15)
    Next ICount
    For ICount = 0 To 23
        shpHours(ICount).BackColor = QBColor(1)
    Next ICount
    vTime = Time
    IMinutes = Minute(vTime)
    For ICount = 1 To IMinutes
        shpMinutes(ICount).BackColor = QBColor(7)
    Next ICount
    ISeconds = Second(vTime)
    For ICount = 1 To ISeconds
        shpSeconds(ICount).BackColor = QBColor(9)
    Next ICount
End Sub

Private Sub Form_Activate()
    For n = 0 To 23
        Label3(n) = n + 1
    Next n
    sInit
End Sub

Private Sub Timer1_Timer()
    vTime = Time

```



```

ISeconds = Second(vTime)
IMinutes = Minute(vTime)
IHours = Hour(vTime)
If ISeconds = 1 Then
    For ICount = 0 To 59
        shpSeconds(ICount).BackColor = QBColor(1)
    Next ICount
End If
shpSeconds(ISeconds).BackColor = QBColor(9)
If (IMinutes = 1) And (ISeconds = 1) Then
    For ICount = 0 To 59
        shpMinutes(ICount).BackColor = QBColor(15)
    Next ICount
End If
shpMinutes(IMinutes).BackColor = QBColor(7)
If (IMinutes = 0) And (ISeconds = 1) Then
    For ICount = 0 To 23
        shpHours(ICount).BackColor = QBColor(1)
    Next ICount
End If
shpHours(IHours).BackColor = QBColor(9)
Label1 = IMinutes & " Minuten"
Label2 = ISeconds & " Sekunden"
End Sub

```

Tipp : 10.

'In frmExeBlocker

'Beispiel : Exe-Dateien voruebergehend unbrauchbar machen und
' Funktionalitaet wieder herstellen.

```

Dim ByteOne As String * 1
Dim sFileName As String

```

```

Private Sub sSaveLastChangeBlock(sFile As String)

```

```

    Dim sValue As String
    Dim iFileNumber As Integer

```

```

    sValue = Date & " " & Time & " " & sFileName & " Locked"
    iFileNumber = FreeFile
    Open sFile For Append As #iFileNumber
    Print #iFileNumber, sValue
    Close #iFileNumber

```

```

End Sub

```

```

Private Sub sSaveLastChangeUnBlock(sFile As String)

```

```

    Dim sValue As String
    Dim iFileNumber As Integer

```

```

    sValue = Date & " " & Time & " " & sFileName & " Unlocked"
    iFileNumber = FreeFile
    Open sFile For Append As #iFileNumber
    Print #iFileNumber, sValue
    Close #iFileNumber

```

```

End Sub

```

```

Private Sub Command1_Click(Index As Integer)

```

```

    Select Case Index

```

```

        Case 0

```

```

            sBlockUnblockExe sFileName, "N", True

```

```

        Case 1

```

```

            sBlockUnblockExe sFileName, "M", False

```

```

    End Select

```

```

End Sub

```

```

Private Function sBlockUnblockExe(sFile As String, _

```

```

    sByte As String, _

```

```

    bBlock As Boolean)

```

```

    ByteOne = sByte

```

```

    If sFile = vbNullString And mnuGerman.Enabled = False Then

```

```

        MsgBox "Sie haben keine .exe Datei ausgewaehlt.", , "Fehler..."

```

```

        Exit Function

```

```

    End If

```

```

    If sFile = vbNullString And mnuEnglish.Enabled = False Then

```

```

        MsgBox "You don'tn have choose one .exe File.", , "Error..."

```

```

        Exit Function

```

```

    End If

    If blBlock Then
        Open sFile For Random As #1 Len = 1
        Put #1, 1, ByteOne
        Close #1
        sSaveLastChangeBlock sLogFile
    Else
        Open sFile For Random As #1 Len = 1
        Put #1, 1, ByteOne
        Close #1
        sSaveLastChangeUnBlock sLogFile
    End If
End Function

Private Sub Command2_Click()
    Dim frm As Form
    Set frm = frmViewLastSik
    frm.Show , Me
End Sub

Private Sub Command3_Click()
    If blDefaultLanguage Then
        sHelpGerman
    Else
        sHelpEnglish
    End If
End Sub

Private Sub sHelpGerman()
    Dim sTxt As String
    sTxt = "Hilfe" & vbCrLf
    sTxt = sTxt & "" & vbCrLf
    sTxt = sTxt & "Testen koennen Sie die Aenderung der" & vbCrLf
    sTxt = sTxt & ".exe indem Sie einen Klick mit der rechten" & vbCrLf
    sTxt = sTxt & "Mousetaste im rechten Listenfeld ausfuehren," & vbCrLf
    sTxt = sTxt & "nachdem Sie die zu testende Datei markiert haben." & vbCrLf
    sTxt = sTxt & "" & vbCrLf
    sTxt = sTxt & "Download by : www.Visual-Basic5.de" & vbCrLf
    sTxt = sTxt & "all rights reserved by prelle's basics | germany 2001"
    MsgBox sTxt, , "Hilfe..."
End Sub

Private Sub sHelpEnglish()
    Dim sTxt As String
    sTxt = "Help" & vbCrLf
    sTxt = sTxt & "" & vbCrLf
    sTxt = sTxt & "Testing the change with right mouseclick in" & vbCrLf
    sTxt = sTxt & "the right listbox, after your have marked the .exe" & vbCrLf
    sTxt = sTxt & "File." & vbCrLf
    sTxt = sTxt & "" & vbCrLf
    sTxt = sTxt & "Download by : www.Visual-Basic5.de" & vbCrLf
    sTxt = sTxt & "all rights reserved by prelle's basics | germany 2001"
    MsgBox sTxt, , "Help..."
End Sub

Private Sub File1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If Button = vbRightButton Then
        Call fCallFileOrProg(File1.filename)
    End If
End Sub

Private Sub Form_Load()
    If blDefaultLanguage Then
        fLoadResDataGerman
        mnuGerman.Enabled = False
    Else
        fLoadResDataEnglish
        mnuEnglish.Enabled = False
    End If
    With File1
        .Pattern = "*.exe"
    End With
    sLogFile = fCompletePath(App.Path) & "log.dat"
End Sub

Private Function fLoadResDataGerman()

```

```

With Me
    .Icon = fLoadResourceBitmap("germany", vbResIcon)
End With
mnuFile.Caption = fLoadResourceString(100)
mnuEnd.Caption = fLoadResourceString(101)
mnuLanguage.Caption = fLoadResourceString(102)
mnuEnglish.Caption = fLoadResourceString(103)
mnuGerman.Caption = fLoadResourceString(104)
frmExeBlocker.Caption = fLoadResourceString(105)
Command1(0).Caption = fLoadResourceString(106)
Command1(1).Caption = fLoadResourceString(107)
Command2.Caption = fLoadResourceString(108)
Frame1.Caption = fLoadResourceString(109)
mnuInfo.Caption = fLoadResourceString(108)
End Function

Private Function fLoadResDataEnglish()
    With Me
        .Icon = fLoadResourceBitmap("unkingdom", vbResIcon)
    End With
    mnuFile.Caption = fLoadResourceString(200)
    mnuEnd.Caption = fLoadResourceString(201)
    mnuLanguage.Caption = fLoadResourceString(202)
    mnuEnglish.Caption = fLoadResourceString(203)
    mnuGerman.Caption = fLoadResourceString(204)
    frmExeBlocker.Caption = fLoadResourceString(205)
    Command1(0).Caption = fLoadResourceString(206)
    Command1(1).Caption = fLoadResourceString(207)
    Command2.Caption = fLoadResourceString(208)
    Frame1.Caption = fLoadResourceString(209)
    mnuInfo.Caption = fLoadResourceString(208)
End Function

Private Sub Form_Unload(Cancel As Integer)
    Set frmExeBlocker = Nothing
    Unload Me
End Sub

Private Sub mnuEnd_Click()
    Unload Me
End Sub

Private Sub mnuEnglish_Click()
    fLoadResDataEnglish
    mnuEnglish.Enabled = False
    mnuGerman.Enabled = True
    blDefaultLanguage = False
End Sub

Private Sub mnuGerman_Click()
    fLoadResDataGerman
    mnuGerman.Enabled = False
    mnuEnglish.Enabled = True
    blDefaultLanguage = True
End Sub

Private Sub Dir1_Change()
    File1.Path = Dir1.Path
    Label1 = vbNullString
End Sub

Private Sub Drive1_Change()
    On Error Resume Next
    Dir1.Path = Drive1.Drive
    Drive1.Drive = Dir1.Path
    Label1 = vbNullString
End Sub

Private Sub File1_Click()
    sFileName = File1.Path & "\" & File1.filename
    If InStr(sFileName, "\\") Then _
        sFileName = File1.Path & File1.filename
    Label1 = "Lock " & "<" & File1.filename & "> ?..."
End Sub

```

```

Function fCallFileOrProg(sProgOrFile As String, _
    Optional IShowAs As Long = vbNormalNoFocus, _
    Optional ByVal vPath As Variant) As Long

    Dim iBackSlash As Integer
    Dim iBeginChar As Integer

    If Left(sProgOrFile, 1) = "" Then
        iBeginChar = InStr(2, sProgOrFile, "")
    If iBeginChar <> 0 Then
        sProgOrFile = Mid(sProgOrFile, 2, iBeginChar - 2) & _
            Mid(sProgOrFile, iBeginChar + 1)
        iBeginChar = iBeginChar - 1
    End If
    Else
        iBeginChar = InStr(sProgOrFile, " ")
    End If
    If iBeginChar = 0 Then iBeginChar = Len(sProgOrFile) + 1
    If IsMissing(vPath) Then
        For iBackSlash = iBeginChar - 1 To 1 Step -1
            If Mid(sProgOrFile, iBackSlash, 1) = "\" Then Exit For
        Next
    If iBackSlash = 0 Then
        vPath = CurDir
    ElseIf iBackSlash = 1 Or Mid(sProgOrFile, _
        iBackSlash - 1, 1) = ":" Then
        vPath = Left(sProgOrFile, iBackSlash)
    Else
        vPath = Left(sProgOrFile, iBackSlash - 1)
    End If
End If
fCallFileOrProg = ShellExecute(0, vbNullString, _
    Left(sProgOrFile, _
        iBeginChar - 1), _
    LTrim(Mid(sProgOrFile, _
        iBeginChar)), _
    vPath, IShowAs)
If fCallFileOrProg < 32 Then
    MsgBox "Magic Header ist blockiert : Magic Header is Locked.", , "File Locked..."
End If
End Function

Private Sub mnulInfo_Click()
    Call Command3_Click
End Sub

'In frmChooseLanguage
Private Sub Form_Load()
    sSaveInRegistry

    With Picture2
        .Width = Me.Width
        .Height = Me.Height
        .AutoRedraw = True
    End With

    sSetBackGround Picture2, frmChooseLanguage

    For n = 0 To 1
        Picture1(n).AutoSize = True
        Picture1(n).BorderStyle = 0
    Next n
    Picture1(0).Picture = fLoadResourceBitmap("germany", vbResIcon)
    Picture1(1).Picture = fLoadResourceBitmap("unkingdom", vbResIcon)
    fCenterForm Me
End Sub

Private Sub sSaveInRegistry()
    SaveSetting "Exe-Blocker", "PrellesBasic", "Registrierung", _
        "Free I-Net Version 1.1"
End Sub

Private Sub Picture1_Click(Index As Integer)
    Select Case Index
        Case 0
            blDefaultLanguage = True
        Case 1
    
```

```

        blDefaultLanguage = False
    End Select
    Dim frm As Form
    Set frm = frmExeBlocker
    frm.Show
    Unload Me
End Sub

'In frmViewLastSik
Dim lLeft As Long
Dim lTop As Long
Dim lWidth As Long
Dim lHeight As Long

Private Sub Command1_Click()
    Set frmViewLastSik = Nothing
    Unload Me
End Sub

Private Sub sMoveMe()
    lLeft = frmExeBlocker.Left
    lTop = frmExeBlocker.Top
    lWidth = Me.Width
    lHeight = frmExeBlocker.Height
End Sub

Private Sub Form_Load()
    sMoveMe
    With Me
        .Move lLeft, lTop, lWidth, lHeight
        .Icon = frmExeBlocker.Icon
    End With

    If blDefaultLanguage Then
        Me.Caption = fLoadResourceString(110)
        Command1.Caption = fLoadResourceString(111)
    Else
        Me.Caption = fLoadResourceString(210)
        Command1.Caption = fLoadResourceString(211)
    End If
    sOpenFile sLogFile
End Sub

Private Sub Form_Resize()
    Command1.Top = Me.Height - (Command1.Height * 2)
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Set frmViewLastSik = Nothing
    Unload Me
End Sub

Private Function sOpenFile(sLogName As String)
    Dim iFileNumber As Integer
    Dim sRetStr As String

    On Error GoTo ErrHandle
    iFileNumber = FreeFile
    Open sLogName For Input As #iFileNumber
    sRetStr = Input(LOF(iFileNumber), #iFileNumber)
    Text1 = sRetStr
    Close iFileNumber
ErrHandle:
    Exit Function
End Function

'In Modul1
Public Declare Function GetSystemMetrics Lib "user32" _
    (ByVal nIndex As Long) _
    As Long

Public Declare Function ShellExecute Lib "shell32.dll" _
    Alias "ShellExecuteA" _
    (ByVal hwnd As Long, _
    ByVal lpOperation As String, _
    ByVal lpFile As String, _
    ByVal lpParameters As String, _

```

```
ByVal IpDirectory As String, _  
ByVal nShowCmd As Long) _  
As Long
```

```
Public Declare Function BitBlt Lib "gdi32" _  
    (ByVal hDestDC As Long, _  
    ByVal X As Long, _  
    ByVal Y As Long, _  
    ByVal nWidth As Long, _  
    ByVal nHeight As Long, _  
    ByVal hSrcDC As Long, _  
    ByVal xSrc As Long, _  
    ByVal ySrc As Long, _  
    ByVal dwRop As Long) _  
    As Long
```

```
Public Const SRCCOPY = &HCC0020  
Public Const SM_CXFULLSCREEN = 16  
Public Const SM_CYFULLSCREEN = 17
```

```
Public blDefaultLanguage As Boolean  
Public sLogFile As String
```

```
Public Function fCenterForm(frm As Form) As Boolean
```

```
    Dim lLeft As Long  
    Dim lTop As Long
```

```
    On Error Resume Next
```

```
    If frm.ScaleMode <> vbTwips Then Exit Function
```

```
    lLeft = (Screen.TwipsPerPixelX * GetSystemMetrics(SM_CXFULLSCREEN)) / 2
```

```
    lLeft = lLeft - (frm.Width / 2)
```

```
    lTop = (Screen.TwipsPerPixelY * GetSystemMetrics(SM_CYFULLSCREEN)) / 2
```

```
    lTop = lTop - (frm.Height / 2)
```

```
    frm.Move lLeft, lTop
```

```
    fCenterForm = Err.Number = 0 And Err.LastDllError = 0
```

```
End Function
```

```
Public Function fLoadResourceString(vResId As Variant) _  
    As String
```

```
    On Error GoTo ErrHandle
```

```
    fLoadResourceString = LoadResString(vResId)
```

```
    Exit Function
```

```
ErrHandle:
```

```
    fLoadResourceString = ""
```

```
    MsgBox "Fehler beim laden der Resource : ", , "Error..."
```

```
End Function
```

```
Public Function fLoadResourceBitmap(vResId As Variant, _  
    lrcLoadPicture As LoadResConstants) _  
    As IPictureDisp
```

```
    On Error GoTo ErrHandle
```

```
    Set fLoadResourceBitmap = LoadResPicture(vResId, lrcLoadPicture)
```

```
    Exit Function
```

```
ErrHandle:
```

```
    Set fLoadResourceBitmap = Nothing
```

```
    MsgBox "Fehler : ", , "Error..."
```

```
End Function
```

```
Public Function fCompletePath(ByVal sPath As String) As String
```

```
    If Right$(sPath, 1) <> "\" Then
```

```
        sPath = sPath + "\"
```

```
    End If
```

```
    fCompletePath = sPath
```

```
End Function
```

```
Public Sub sSetBackGround(pb As PictureBox, frm As Form)
```

```
    Dim lTileWidth As Long
```

```
    Dim lTileHeight As Long
```

```
    Dim lResX As Long
```

```
    Dim lResY As Long
```

```
    Dim lRetX As Long
```

```
    Dim lRetY As Long
```

```
    Dim lRndX As Long
```

```

Dim IRndY                As Long
Dim n                    As Long

Randomize
IRndX = pb.ScaleWidth - 1
IRndY = pb.ScaleHeight - 1
pb.Cls
For n = 0 To 150
    IRetX = Int(IRndX * Rnd + 1)
    IRetY = Int(IRndY * Rnd + 1)
    pb.PSet (IRetX, IRetY), QBColor(8)
    pb.PSet (IRetX + 1, IRetY + 1), QBColor(15)
Next n

ITileWidth = 128: ITileHeight = 128
IResX = (frm.ScaleWidth \ ITileWidth) + 1
IResY = (frm.ScaleHeight \ ITileHeight) + 1
For IRetY = IResY To 0 Step -1
    For IRetX = 0 To IResX
        BitBlt frm.hDC, IRetX * ITileWidth, IRetY * _
            ITileHeight, ITileWidth, _
            ITileHeight, pb.hDC, 0, 0, _
            SRCCOPY
    Next IRetX
Next IRetY
End Sub

```

Tipp : 11.

'Beispiel : Text im Uhrzeigersinn automatisch rotieren.

```

Private Declare Function DeleteObject Lib "gdi32" _
    (ByVal hObject As Long) _
    As Long

Private Declare Function GetTickCount Lib "kernel32" _
    () As Long

Private Declare Function CreateFont Lib "gdi32" _
    Alias "CreateFontA" _
    (ByVal H As Long, _
    ByVal W As Long, _
    ByVal E As Long, _
    ByVal O As Long, _
    ByVal W As Long, _
    ByVal I As Long, _
    ByVal u As Long, _
    ByVal S As Long, _
    ByVal C As Long, _
    ByVal OP As Long, _
    ByVal CP As Long, _
    ByVal Q As Long, _
    ByVal PAF As Long, _
    ByVal F As String) _
    As Long

Private Declare Function SelectObject Lib "gdi32" _
    (ByVal hdc As Long, _
    ByVal hObject As Long) _
    As Long

Const TRUEATYPE_FONTTYPE As Long = &H4
Const FF_DONTCARE As Long = 0
Const ANSI_CHARSET As Long = 0
Const CLIP_LH_ANGLES As Long = 16
Const CLIP_DEFAULT_PRECIS As Long = 0
Const PROOF_QUALITY As Long = 2
Const OUT_TT_ONLY_PRECIS As Long = 7

```

'Verwenden Sie lcfHHeight um die Schrifthoehe zu aendern.

```
Const lcfHHeight As Long = 44
```

'Verwenden Sie lcfWWidth um die Schriftbreite zu aendern.

```
Const lcfWWidth As Long = 16
```

```
Const sOutputString As String = "www.Visual-Basic5.de"
```

'Verwenden Sie lAnimInterval zum veraendern der Rotationsgeschwindigkeit

'Niedriger Wert(z.B.10) = Schnell

'Hoher Wert(z.B.80) = Langsamer

```

Const IAnimInterval           As Long = 50
Const ILeftIndex             As Long = 1
Const ITopIndex              As Long = 2
'Verwenden Sie sFontType um die Schriftart zu aendern.
Const sFontType              As String = "Times New Roman"

Dim IarrCoordinates(1 To 2, 1 To 36) As Long
Dim IDrawRadian              As Long
Dim IXPosition               As Long
Dim IYPosition               As Long

Dim bEndRotation             As Boolean

Function fPi() As Double
    fPi = 4 * Atn(1)
End Function

Private Sub sInit()
    Dim ILastAnimInterval     As Long
    Dim IAngle                 As Long
    Me.Show
    IAngle = 1800

    Do
        If bEndRotation Then Exit Do
        If GetTickCount() - ILastAnimInterval > IAnimInterval Then
            IAngle = (IAngle Mod 3600) - 100
            Me.Cls
            sRotationAnimText IAngle
            ILastAnimInterval = GetTickCount()
        End If
        DoEvents
    Loop
End Sub

Private Sub sRotationAnimText(IAngle As Long)
    Static n                   As Long
    Dim IRetCreateFont         As Long
    Dim IRetOldFont            As Long

    IRetCreateFont = CreateFont(IcfHHeight, IcfWWidth, _
        IAngle, 0, FF_DONTCARE, 0, 0, 0, _
        ANSI_CHARSET, OUT_TT_ONLY_PRECIS, _
        CLIP_LH_ANGLES Or CLIP_DEFAULT_PRECIS, _
        PROOF_QUALITY, TRUETYPE_FONTTYPE, sFontType)
    IRetOldFont = SelectObject(Me.hdc, IRetCreateFont)

    n = (n Mod 36) + 1
    CurrentX = IXPosition + IarrCoordinates(1, n)
    CurrentY = IYPosition + IarrCoordinates(2, n)
    Print sOutputString
    IRetCreateFont = SelectObject(Me.hdc, IRetOldFont)
    DeleteObject IRetCreateFont
End Sub

Private Sub Command1_Click()
    bEndRotation = False
    sInit
End Sub

Private Sub Form_Load()
    With Form1
        .AutoRedraw = True
        .ForeColor = vbRed
        .ScaleMode = vbPixels
    End With

    IDrawRadian = ((Len(sOutputString) * IcfWWidth) / 2)
    sArrContents
    bEndRotation = True
    sInit
End Sub

Private Sub Form_Resize()
    With Form1
        IXPosition = .ScaleWidth / 2
        IYPosition = .ScaleHeight / 2
    End With
End Sub

```



```

End With
End Sub

Private Sub Form_Unload(Cancel As Integer)
    bEndRotation = True
End Sub

Private Sub sArrContents()
    Dim lAngle As Long
    Dim n As Long

    For n = LBound(larrCoordinates, 2) To UBound(larrCoordinates, 2)
        larrCoordinates(lLeftIndex, n) = _
            CLng(Cos((lAngle * fPi / 180)) * lDrawRadian)

        larrCoordinates(lTopIndex, n) = _
            CLng(Sin((lAngle * fPi / 180)) * lDrawRadian)

        lAngle = (lAngle Mod 360) + 10
    Next n
End Sub

```

Tipp : 12.

'Beispiel : Fremdanwendung starten. Doppelstart der Anwendung verhindern.

'Hinweis :

In diesem Beispiel wird der MediaPlayer verwendet.

'Einige Klassennamen :

```

' : SmartSketchMDIFrame = Flash 5
' : GenesisDLG = Cinema 4D
' : PoserWndClass = Poser 2
' : OMain = Access
' : EFrame = Internet Explorer
' : ThorBrowserWndClass = Outlook
' : OpusApp = MSWord
' : XLMAIN = MSEXcel
' : MSPhotoEditor32MainClasse = MSPhoto Editor
' : PP97FrameClass = PowerPoint

```

```

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

```

```

Private Function fStartApp() As Boolean
    fStartApp = True
    If FindWindow("SJE_CdPlayerClass", vbNullString) = 0 Then
        Shell "cdplayer.exe", vbNormalFocus
    Else
        fStartApp = True
    End If
End Function

```

```

Private Sub Command1_Click()
    fStartApp
End Sub

```

```

Private Sub Command2_Click()
    Unload Me
End Sub

```

Tipp : 13.

'Beispiel : Grafikinformationen ermitteln.

```

Private Sub Command1_Click()
    Dim spPicture As StdPicture
    Dim lWidth As Long
    Dim lHeight As Long

    Set spPicture = LoadPicture(App.Path & "\test1.bmp")

```

```

    sPictureWidthHeight spPicture, IHeight, IWidth
    Label1 = "Weite : " & CStr(IWidth) & " Pixel"
    Label2 = "Hoehe : " & CStr(IHeight) & " Pixel"
End Sub

Private Sub Command2_Click()
    Unload Me
End Sub

Private Sub sPictureWidthHeight(spPicture As StdPicture, _
    ByRef IHeight As Long, _
    ByRef IWidth As Long)
    Width = CLng(ScaleX(spPicture.Width, vbHimetric, vbPixels))
    IHeight = CLng(ScaleY(spPicture.Height, vbHimetric, vbPixels))
End Sub

```

Tipp : 14.

'Beispiel : Grafikinfos ermitteln : 2

```

Private Type BITMAPFILEHEADER
    bfType          As Integer
    bfSize          As Long
    bfReserved1    As Integer
    bfReserved2    As Integer
    bfOffBits      As Long
End Type

Private Type BITMAPINFOHEADER
    biSize          As Long
    biWidth         As Long
    biHeight        As Long
    biPlanes        As Integer
    biBitCount      As Integer
    biCompression  As Long
    biSizeImage     As Long
    biXPelsPerMeter As Long
    biYPelsPerMeter As Long
    biClrUsed       As Long
    biClrImportant  As Long
End Type

Private Sub Command1_Click()
    MsgBox "INET-Version : Keine Hilfe verfuegbar.", vbInformation, "Help..."
End Sub

Private Sub Command2_Click()
    End
End Sub

Private Sub Dir1_Change()
    File1.Path = Dir1.Path
End Sub

Private Sub Dir1_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = 13 Then
        Dir1.Path = Dir1.List(Dir1.ListIndex)
    End If
End Sub

Private Sub Dir1_MouseDown(Button As Integer, _
    Shift As Integer, _
    X As Single, _
    Y As Single)
    If Button = 1 Then
        Dir1.Path = Dir1.List(Dir1.ListIndex)
    End If
End Sub

Private Sub Drive1_Change()
    On Error Resume Next
    Dir1.Path = Drive1.Drive
End Sub

Private Sub File1_Click()
    Text1 = File1.filename

```

```

    fGetBitmapInfo
End Sub

Private Sub Form_Load()
    Drive1.Drive = "C:\"
    File1.Pattern = Text2
End Sub

Public Function fCompletePath(ByVal sPath As String) As String
    If Right$(sPath, 1) <> "\" Then
        sPath = sPath + "\"
    End If
    fCompletePath = sPath
End Function

Private Function fGetBitmapInfo()
    Dim sFileName          As String
    Dim iFileNumber        As Integer
    Dim BIH                As BITMAPINFOHEADER
    Dim BFH                As BITMAPFILEHEADER

    sFileName = fCompletePath(Dir1.Path) + File1.filename
    iFileNumber = FreeFile
    On Error GoTo ErrHandle
    Open sFileName For Binary Access Read As #iFileNumber
        Get #iFileNumber, , BFH
        Get #iFileNumber, , BIH
    Close #iFileNumber

ErrHandle:
    Label5(0) = "Grafik-Weite   : " & CStr(BIH.biWidth) & " Pixel."
    Label5(1) = "Grafik-Hoehe  : " & CStr(BIH.biHeight) & " Pixel."
    Label5(2) = "Groesse       : " & Format$(FileLen(sFileName), "###,###") & " Bytes."
    Label5(3) = "Anzahl Farben : " & (2 ^ BIH.biBitCount) & " Farben."
    Label5(4) = "Tiefe         : " & CStr(BIH.biBitCount) & " Bits."
End Function

```

Tipp : 15.

'Beispiel : Grafik Anti - Alias.

```

Private Declare Function LoadImage Lib "user32" _
    Alias "LoadImageA" _
    (ByVal hInst As Long, _
    ByVal lpsz As String, _
    ByVal ildc As Long, _
    ByVal cx As Long, _
    ByVal cy As Long, _
    ByVal fFlags As Long) _
    As Long

Private Declare Function DeleteObject Lib "gdi32" _
    (ByVal hObject As Long) _
    As Long

Private Declare Function SelectObject Lib "gdi32" _
    (ByVal hdc As Long, _
    ByVal hObject As Long) _
    As Long

Private Declare Function StretchBlt Lib "gdi32" _
    (ByVal hdc As Long, _
    ByVal x As Long, _
    ByVal y As Long, _
    ByVal nWidth As Long, _
    ByVal nHeight As Long, _
    ByVal hSrcDC As Long, _
    ByVal xSrc As Long, _
    ByVal ySrc As Long, _
    ByVal nSrcWidth As Long, _
    ByVal nSrcHeight As Long, _
    ByVal dwRop As Long) _
    As Long

Private Declare Function CreateCompatibleDC Lib "gdi32" _
    (ByVal hdc As Long) _

```

As Long

```
Private Declare Function DeleteDC Lib "gdi32" _
    (ByVal hdc As Long) _
    As Long
```

```
Private Const IMAGE_BITMAP = 0
Private Const LR_LOADFROMFILE = &H10
Private Const SRCCOPY = &HCC0020
```

```
Private Sub sLoadGraficalFile(sFileName As String, _
    bOnOff As Boolean)
    Dim IdcBuffer As Long
    Dim IImageBuffer As Long

    IdcBuffer = CreateCompatibleDC(0)
    If bOnOff Then
        IImageBuffer = LoadImage(0&, sFileName, IMAGE_BITMAP, 480, 120, _
            LR_LOADFROMFILE)
        Call SelectObject(IdcBuffer, IImageBuffer)
        With Picture2
            .AutoRedraw = True
            Call StretchBlt(.hdc, 0, 0, 480, 120, IdcBuffer, 0, 0, 480, 120, _
                SRCCOPY)
            .Refresh
            .AutoRedraw = False
        End With
    Else
        IImageBuffer = LoadImage(0&, sFileName, IMAGE_BITMAP, 0, 0, _
            LR_LOADFROMFILE)
        Call SelectObject(IdcBuffer, IImageBuffer)
        With Picture2
            .AutoRedraw = True
            Call StretchBlt(.hdc, 0, 0, 480, 120, IdcBuffer, 0, 0, 120, 30, _
                SRCCOPY)
            .Refresh
            .AutoRedraw = False
        End With
    End If
    Call DeleteDC(IdcBuffer)
    Call DeleteObject(IImageBuffer)
End Sub
```

```
Private Sub Command1_Click(Index As Integer)
    Select Case Index
        Case 0
            Call sLoadGraficalFile(App.Path & "\test1.bmp", True)
        Case 1
            Call sLoadGraficalFile(App.Path & "\test1.bmp", False)
        Case 2
            Unload Me
    End Select
End Sub
```

```
Private Sub Form_Load()
    On Error Resume Next
    Image1.Picture = LoadPicture(App.Path & "\test1.bmp")
    Picture2.ScaleMode = vbPixels
End Sub
```

Tipp : 16.

'Beispiel : Text Enabled / Disabled.

```
Private Declare Function DrawStateText Lib "user32" _
    Alias "DrawStateA" _
    (ByVal hDC As Long, _
    ByVal hBrush As Long, _
    ByVal lpDrawStateProc As Long, _
    ByVal lData As String, _
    ByVal wData As Long, _
    ByVal X As Long, _
    ByVal Y As Long, _
    ByVal cx As Long, _
    ByVal cy As Long, _
```

```
ByVal fFlags As Long) _  
As Long
```

```
Dim IXPos As Long  
Dim IYPos As Long  
Private Const TXT_DISABLED As Long = &H20&  
Private Const TXT_PREFIXTEXT As Long = &H2&  
Private Const TXT_TEXT As Long = &H1&
```

```
Private Sub Form_Load()  
With Form1  
.ScaleMode = vbPixels  
.AutoRedraw = True  
.FontSize = 16  
.FontBold = True  
End With  
sDrawTextEnabledDisabled 0  
Me.Show  
End Sub
```

```
Private Sub Command1_Click(Index As Integer)  
Select Case Index  
Case 0  
sDrawTextEnabledDisabled 0  
Case 1  
sDrawTextEnabledDisabled 1  
Case 2  
Unload Me  
End  
End Select  
End Sub
```

```
Private Sub sDrawTextEnabledDisabled(Index As Integer)  
Dim IDrawStateFlag As Long  
Dim sTxtRet As String  
  
If Index = 1 Then IDrawStateFlag = TXT_DISABLED  
Me.Cls  
sTxtRet = "http://www.Visual-Basic5.de"  
IXPos = (Me.ScaleWidth \ 2) - (Me.TextWidth(sTxtRet) \ 2)  
IYPos = (Me.ScaleHeight \ 2) - (Me.TextHeight(sTxtRet))  
Call DrawStateText(Me.hDC, 0&, 0&, sTxtRet, Len(sTxtRet), _  
IXPos, IYPos, 0&, 0&, TXT_TEXT Or IDrawStateFlag)  
Me.Refresh  
End Sub
```

Tipp : 17.

'Beispiel : Bestimmten Grafikbereich zoomen.

```
Private Const IMinWidth As Long = 20  
Private Const IMinHeight As Long = 20  
Dim ILeft As Long  
Dim ITop As Long  
Dim IHeight As Long  
Dim IWidth As Long  
Dim bRect As Boolean  
Dim bUpDown As Boolean  
Dim IPosX0 As Long  
Dim IPosY0 As Long  
Dim IPosX1 As Long  
Dim IPosY1 As Long  
Dim IPosX2 As Long  
Dim IPosY2 As Long  
Dim iZoomFactor As Integer
```

```
Private Sub Form_Load()  
On Error Resume Next  
Picture1.Picture = LoadPicture(App.Path & "\test.jpg")  
iZoomFactor = 1  
bUpDown = True  
With Picture1  
.DrawMode = vbNotXorPen  
.DrawStyle = vbDot  
With Picture2  
.AutoRedraw = True
```

```
End With
End With
End Sub
```

```
Private Sub Picture1_MouseDown(Button As Integer, _
    Shift As Integer, _
    X As Single, _
    Y As Single)
    If bRect = True Then Call sSetRectangle(Picture1)
    If bUpDown = True Then
        sFirstLine Picture1
        IPosX1 = X
        IPosY1 = Y
        IPosX2 = X
        IPosY2 = Y
        Picture1.Line (IPosX1, IPosY1)-(IPosX2, IPosY2), , B
        IPosX0 = IPosX1
        IPosY0 = IPosY1
        bUpDown = False
    End If
End Sub
```

```
Private Sub sFirstLine(obj As PictureBox)
    IPosX2 = IPosX0
    IPosY2 = IPosY0
    obj.Line (IPosX1, IPosY1)-(IPosX2, IPosY2), , B
End Sub
```

```
Private Sub Picture1_MouseMove(Button As Integer, _
    Shift As Integer, _
    X As Single, _
    Y As Single)
    If bUpDown = False Then
        sFirstLine Picture1
        IPosX2 = X
        IPosY2 = Y
        Picture1.Line (IPosX1, IPosY1)-(IPosX2, IPosY2), , B
        IPosX0 = IPosX2
        IPosY0 = IPosY2
    End If
End Sub
```

```
Private Sub Picture1_MouseUp(Button As Integer, _
    Shift As Integer, _
    X As Single, _
    Y As Single)
    bUpDown = True
    If IPosX2 < IPosX1 Then
        ILeft = IPosX2
    Else
        ILeft = IPosX1
    End If
    If IPosY2 < IPosY1 Then
        ITop = IPosY2
    Else
        ITop = IPosY1
    End If
    IWidth = Abs(IPosX2 - IPosX1)
    IHeight = Abs(IPosY2 - IPosY1)
    If ILeft < 0 Then ILeft = 0
    If ITop < 0 Then ITop = 0
    If IWidth = 0 Then IWidth = IMinWidth
    If IHeight = 0 Then IHeight = IMinHeight

    With Picture1
        If ILeft + IWidth > .Width Then IWidth = .Width - ILeft
        If ITop + IHeight > .Height Then IHeight = .Height - ITop
    End With

    With Picture2
        .Width = IWidth * iZoomFactor
        .Height = IHeight * iZoomFactor
        .Refresh
        .PaintPicture Picture1.Image, 0, 0, IWidth * iZoomFactor, IHeight * iZoomFactor, ILeft, ITop, IWidth, IHeight
    End With
    Call sSetRectangle(Picture1)
End Sub
```

```

Private Sub Command1_Click(Index As Integer)
    Select Case Index
        Case 0
            iZoomFactor = 1
        Case 1
            iZoomFactor = 2
        Case 2
            iZoomFactor = 3
    End Select
End Sub

```

```

Private Sub sSetRectangle(obj As PictureBox)
    obj.DrawStyle = vbSolid
    obj.Line (ILeft - 120, ITop - 120)-(ILeft - 40, ITop - 40), 0, BF
    obj.Line (ILeft - 120, ITop + IHeight + 40)-(ILeft - 40, ITop + IHeight + 120), 0, BF
    obj.Line (ILeft + IWidth + 40, ITop - 120)-(ILeft + IWidth + 120, ITop - 40), 0, BF
    obj.Line (ILeft + IWidth + 40, ITop + IHeight + 40)-(ILeft + IWidth + 120, ITop + IHeight + 120), 0, BF
    obj.DrawStyle = vbDot
    bRect = True
End Sub

```

Tipp : 18.

'Beispiel : Grafik von oben nach unten farbig ueberblenden.

```

Private Sub sBlendPictureTopToBottom(pic As PictureBox, _
    iBlendSpeed As Integer, _
    IBlendColor As Long)

    Dim iHeight As Integer
    Dim iWidth As Integer
    Dim k As Integer
    Dim n As Integer

    pic.ForeColor = IBlendColor
    iWidth = pic.ScaleWidth / iBlendSpeed
    iHeight = pic.ScaleHeight / iBlendSpeed
    For n = 0 To iBlendSpeed
        For k = 0 To iBlendSpeed
            pic.Line ((k * iWidth), (n * iHeight))-((k + 1) * _
                iWidth, (n + 1) * iHeight), , BF
            DoEvents
        Next k
        DoEvents
    Next n
End Sub

```

```

Private Sub Command1_Click()
    Picture1.Refresh
    sBlendPictureTopToBottom Picture1, 40, &HFF8080
    Command2.Enabled = True
    Command1.Enabled = False
End Sub

```

```

Private Sub Command2_Click()
    Picture1.Refresh
    Command2.Enabled = False
    Command1.Enabled = True
End Sub

```

```

Private Sub Command3_Click()
    Unload Me
End Sub

```

```

Private Sub Form_Load()
    Picture1.Picture = LoadPicture(App.Path & "\test1.jpg")
End Sub

```

Tipp : 19.

'Beispiel : Grafik von unten farbig ueberblenden.

```

Private Sub sBlendPictureBottomToTop(pic As PictureBox, _

```

```

        iBlendSpeed As Integer, _
        iBlendColor As Long)

Dim iHeight As Integer
Dim iWidth As Integer
Dim k As Integer
Dim n As Integer

pic.ForeColor = iBlendColor
iWidth = pic.ScaleWidth / iBlendSpeed
iHeight = pic.ScaleHeight / iBlendSpeed
For n = iBlendSpeed To 0 Step -1
    For k = 0 To iBlendSpeed
        pic.Line (((k - 1) * iWidth), ((n - 1) * iHeight))- _
            (k * iWidth, n * iHeight), , BF
        DoEvents
    Next k
    DoEvents
Next n
End Sub

Private Sub Command1_Click()
    Picture1.Refresh
    sBlendPictureBottomToTop Picture1, 50, &HC0C0&
    Command2.Enabled = True
    Command1.Enabled = False
End Sub

Private Sub Command2_Click()
    Picture1.Refresh
    Command2.Enabled = False
    Command1.Enabled = True
End Sub

Private Sub Command3_Click()
    Unload Me
End Sub

Private Sub Form_Load()
    Picture1.Picture = LoadPicture(App.Path & "\test1.jpg")
End Sub

```

Tipp : 20.

'Beispiel : Grafik von links nach rechts farbig ueberblenden.

```

Private Sub sBlendPictureLeftToRight(pic As PictureBox, _
    iBlendSpeed As Integer, _
    iBlendColor As Long)

Dim iHeight As Integer
Dim iWidth As Integer
Dim k As Integer
Dim n As Integer

pic.ForeColor = iBlendColor
iWidth = pic.ScaleWidth / iBlendSpeed
iHeight = pic.ScaleHeight / iBlendSpeed
For n = 0 To iBlendSpeed
    For k = 0 To iBlendSpeed
        pic.Line ((n * iWidth), (k * iHeight))- _
            ((n + 1) * iWidth, (k + 1) * iHeight), , BF
        DoEvents
    Next k
    DoEvents
Next n
End Sub

Private Sub Command1_Click()
    Picture1.Refresh
    sBlendPictureLeftToRight Picture1, 50, &HFF&
    Command2.Enabled = True
    Command1.Enabled = False
End Sub

Private Sub Command2_Click()

```



```

Picture1.Refresh
Command2.Enabled = False
Command1.Enabled = True
End Sub

```

```

Private Sub Command3_Click()
Unload Me
End Sub

```

```

Private Sub Form_Load()
Picture1.Picture = LoadPicture(App.Path & "\test1.jpg")
End Sub

```

Tipp : 21.

'Beispiel : Grafik von rechts nach links farbig ueberblenden.

```

Private Sub sBlendPictureRightToLeft(pic As PictureBox, _
iBlendSpeed As Integer, _
IBlendColor As Long)

Dim iHeight As Integer
Dim iWidth As Integer
Dim k As Integer
Dim n As Integer

pic.ForeColor = IBlendColor
iWidth = pic.ScaleWidth / iBlendSpeed
iHeight = pic.ScaleHeight / iBlendSpeed
For n = iBlendSpeed To 0 Step -1
For k = 0 To iBlendSpeed
pic.Line ((n * iWidth), (k * iHeight))- _
((n + 1) * iWidth, (k + 1) * iHeight), , BF
DoEvents
Next k
DoEvents
Next n
End Sub

```

```

Private Sub Command1_Click()
Picture1.Refresh
sBlendPictureRightToLeft Picture1, 50, &HFF&
Command2.Enabled = True
Command1.Enabled = False
End Sub

```

```

Private Sub Command2_Click()
Picture1.Refresh
Command2.Enabled = False
Command1.Enabled = True
End Sub

```

```

Private Sub Command3_Click()
Unload Me
End Sub

```

```

Private Sub Form_Load()
Picture1.Picture = LoadPicture(App.Path & "\test1.jpg")
End Sub

```

Tipp : 22.

'Beispiel : Grafik von innen nach aussen farbig ueberblenden.

```

Private Sub sBlendPictureInsideToOutside(pic As PictureBox, _
iBlendSpeed As Integer, _
IBlendColor As Long)

Dim iHeight As Integer
Dim iWidth As Integer
Dim k As Integer
Dim n As Integer

pic.ForeColor = IBlendColor

```

```

iWidth = pic.ScaleWidth / iBlendSpeed
iHeight = pic.ScaleHeight / iBlendSpeed
For n = (iBlendSpeed / 2) To 0 Step -1
    For k = iBlendSpeed To 0 Step -1
        DoEvents
    Next k
    pic.Line (n * iWidth, n * iHeight)-(((iBlendSpeed _
    - n) + 1) * iWidth, ((iBlendSpeed - n) + 1) * _
    iHeight), , BF
Next n
End Sub

Private Sub Command1_Click()
    Picture1.Refresh
    sBlendPictureInsideToOutside Picture1, 80, &HC00000
    Command2.Enabled = True
    Command1.Enabled = False
End Sub

Private Sub Command2_Click()
    Picture1.Refresh
    Command2.Enabled = False
    Command1.Enabled = True
End Sub

Private Sub Command3_Click()
    Unload Me
End Sub

Private Sub Form_Load()
    Picture1.Picture = LoadPicture(App.Path & "\test1.jpg")
End Sub

```

Tipp : 23.

'Beispiel : Grafik Zufallgesteuert ueberblenden.

```

Private Sub sBlendPictureRandomize(pic As PictureBox, _
    iBlendSpeed As Integer, _
    IBlendColor As Long, _
    IDrawWidth As Integer)

    Dim iHeight          As Integer
    Dim iWidth           As Integer
    Dim iTmpHeight       As Integer
    Dim iTmpWidth        As Integer
    Dim iCounter         As Integer
    Dim bArr()           As Byte

    ReDim bArr(iBlendSpeed, iBlendSpeed)

    pic.ForeColor = IBlendColor
    pic.DrawWidth = IDrawWidth
    iWidth = pic.ScaleWidth / iBlendSpeed
    iHeight = pic.ScaleHeight / iBlendSpeed
    Do
        Do
            iTmpWidth = Int(iBlendSpeed * Rnd)
            iTmpHeight = Int(iBlendSpeed * Rnd)
            If bArr(iTmpWidth, iTmpHeight) = 0 Then
                Exit Do
            End If
            iCounter = iCounter + 1
            If iCounter = iBlendSpeed * 5 Then Exit Do
        Loop
        If iCounter = iBlendSpeed * 5 Then Exit Do
        iCounter = 0
        bArr(iTmpWidth, iTmpHeight) = 1
        pic.Line ((iTmpWidth * iWidth), _
            (iTmpHeight * iHeight))-((iTmpWidth + 1) * _
            iWidth, (iTmpHeight + 1) * iHeight), , BF
        DoEvents
    Loop
    pic.Line (0, 0)-(pic.ScaleWidth, pic.ScaleHeight), , BF
End Sub

```

```

Private Sub Command1_Click()
    Picture1.Refresh
    sBlendPictureRandomize Picture1, 100, &HC00000, 3
    Command2.Enabled = True
    Command1.Enabled = False
End Sub

```

```

Private Sub Command2_Click()
    Picture1.Refresh
    Command2.Enabled = False
    Command1.Enabled = True
End Sub

```

```

Private Sub Command3_Click()
    Unload Me
End Sub

```

```

Private Sub Form_Load()
    Picture1.Picture = LoadPicture(App.Path & "\test1.jpg")
End Sub

```

Tipp : 24.

'Beispiel : Verwendung von Resource-Dateien. Icons Enabled / Disabled.

'Hinweis : In der IDE werden Icons aus User.exe geladen.
' Kompiliert werden Icons aus der Resource - Datei geladen.

```

Private Declare Function DestroyIcon Lib "user32" _
    (ByVal hIcon As Long) _
    As Long

```

```

Private Declare Function GetModuleHandle Lib "kernel32" _
    Alias "GetModuleHandleA" _
    (ByVal lpModuleName As String) _
    As Long

```

```

Private Declare Function DrawState Lib "user32" _
    Alias "DrawStateA" _
    (ByVal hDC As Long, _
    ByVal hBrush As Long, _
    ByVal lpDrawStateProc As Long, _
    ByVal IData As Long, _
    ByVal wData As Long, _
    ByVal X As Long, _
    ByVal Y As Long, _
    ByVal cx As Long, _
    ByVal cy As Long, _
    ByVal fFlags As Long) _
    As Long

```

```

Private Declare Function LoadImageBynum Lib "user32" _
    Alias "LoadImageA" _
    (ByVal hInst As Long, _
    ByVal lpszName As Long, _
    ByVal uType As Long, _
    ByVal cxDesired As Long, _
    ByVal cyDesired As Long, _
    ByVal fuLoad As Long) _
    As Long

```

```

Private Const IC_DISABLED           As Long = &H20&
Private Const IC_DESTINATION       As Long = &H3&
Private Const IC_IMAGE             As Long = 1&
Private Const IC_DEFCOLOR          As Long = &H0&

```

```

Private bRunIde                    As Boolean

```

```

Private Function flconEnabledDisabled(Index As Integer)
    Dim licCount                    As Long
    Dim lhModule                    As Long
    Dim lhIcon                      As Long
    Dim lDrawStateFlag              As Long
    Dim lIconXPos                   As Long
    Dim lIconYPos                   As Long
    Const SMALLSIZE                 As Long = 16&

```

```

Const LARGESIZE          As Long = 32&

If Index = 1 Then IDrawStateFlag = IC_DISABLED
Me.Cls
lIconXPos = 20
lIconYPos = 20
If bRunIde = True Then
    lhModule = GetModuleHandle("User.exe")
Else
    lhModule = App.hInstance
End If
For licCount = 100 To 105
    lhIcon = LoadImageByNum(lhModule, licCount, IC_IMAGE, LARGESIZE, LARGESIZE, IC_DEFCOLOR)
    Call DrawState(Me.hDC, 0&, 0&, lhIcon, 0&, lIconXPos, lIconYPos, 0&, 0&, IC_DESTINATION Or IDrawStateFlag)
    Call DestroyIcon(lhIcon)
    lhIcon = LoadImageByNum(lhModule, licCount, IC_IMAGE, SMALLSIZE, SMALLSIZE, IC_DEFCOLOR)
    lIconXPos = lIconXPos + 10
    lIconYPos = lIconYPos + 40
    Call DrawState(Me.hDC, 0&, 0&, lhIcon, 0&, lIconXPos, lIconYPos, 0&, 0&, IC_DESTINATION Or IDrawStateFlag)
    Call DestroyIcon(lhIcon)
    lIconXPos = lIconXPos + 30
    lIconYPos = lIconYPos - 40
Next licCount
Me.Refresh
End Function

Private Sub Command1_Click()
    Unload Me
End Sub

Private Sub Command2_Click()
    fIconEnabledDisabled 0
End Sub

Private Sub Command3_Click()
    fIconEnabledDisabled 1
End Sub

Private Sub Form_Load()
    Form1.ScaleMode = 3
    bRunIde = Not fExeOrIDE
    fIconEnabledDisabled 0
    Me.Show
    If bRunIde Then
        MsgBox "In der IDE werden die Icons nicht aus der Resource, " _
            & vbCrLf & "sondern aus USER32.EXE geladen.", , "Message..."
    End If
End Sub

Private Function fExeOrIDE() As Boolean
    On Error GoTo ErrHandle
    Debug.Print 10 \ 0
    fExeOrIDE = True
ErrHandle:
End Function

```

Tipp : 25.

'Beispiel : Grafik invertieren.

```

Private Declare Function GetClientRect Lib "user32" _
    (ByVal hwnd As Long, _
    lpRect As RECT) _
    As Long

Private Declare Function InvertRect Lib "user32" _
    (ByVal hdc As Long, _
    lpRect As RECT) _
    As Long

Private Type RECT
    Left    As Long
    Top     As Long
    Right   As Long
    Bottom  As Long

```

```

End Type

Private Sub sInvertGrafic(pb As PictureBox)
    Dim RC As RECT
    Call GetClientRect(pb.hwnd, RC)
    Call InvertRect(pb.hdc, RC)
End Sub

Private Sub Command1_Click()
    sInvertGrafic Picture1
End Sub

Private Sub Command2_Click()
    Unload Me
End Sub

Private Sub Command3_Click()
    Picture1.Refresh
End Sub

Private Sub Form_Load()
    On Error GoTo ErrHandle
    Picture1.Picture = LoadPicture(App.Path & "\test1.jpg")
    Exit Sub
ErrHandle:
    MsgBox Err.Description, , "Error..."
    End
End Sub

```

Tipp : 26.

'Beispiel : Bereich invertieren.

```

Private Declare Function InvertRect Lib "user32" _
    (ByVal hdc As Long, _
    lpRect As RECT) _
    As Long

```

```

Private Type RECT
    Left As Long
    Top As Long
    Right As Long
    Bottom As Long
End Type

```

```

Private Sub sInvertRegion(frm As Form)
    Dim RC As RECT

```

'Oberen Bereich invertieren.

```

RC.Left = 15
RC.Top = 40
RC.Right = 210
RC.Bottom = 18
Call InvertRect(frm.hdc, RC)

```

'Unteren Bereich invertieren.

```

RC.Left = 132
RC.Top = 60
RC.Right = 260
RC.Bottom = 30
Call InvertRect(frm.hdc, RC)
End Sub

```

```

Private Sub Command1_Click()
    sInvertRegion Form1
End Sub

```

```

Private Sub Command2_Click()
    Unload Me
End Sub

```

Tipp : 27.

'Beispiel : Grafik ueber eine Form bewegen.

'Hinweis : Auf eine Fehlerbehandlung beim laden der Grafiken
' wurde verzichtet.

```
Private Type CURPOSITION
    iLeft      As Integer
    iTop       As Integer
End Type

Private Type DIRECTION
    iLeftDir   As Integer
    iTopDir    As Integer
End Type

Private Function fMoveDirection(objDrawCon As Object, _
    pbColPic As PictureBox, _
    pbMaskPic As PictureBox, _
    pbMovePic As PictureBox, _
    iSpeed As Integer, _
    IBackCol As Long)

    Dim CP      As CURPOSITION
    Dim DI      As DIRECTION

    CP.iLeft = 0: CP.iTop = 0
    DI.iLeftDir = 1: DI.iTopDir = 1
    objDrawCon.BackColor = IBackCol

    If iSpeed > 150 Then iSpeed = 150: If iSpeed < 60 Then iSpeed = 60
    Do
        pbMovePic.Picture = pbMaskPic.Picture

        If CP.iLeft + 80 + pbMovePic.Width > objDrawCon.ScaleWidth Then
            DI.iLeftDir = 0
        ElseIf CP.iLeft - 80 < 0 Then
            DI.iLeftDir = 1
        End If
        If CP.iTop + 80 + pbMovePic.Height > objDrawCon.ScaleHeight Then
            DI.iTopDir = 0
        ElseIf CP.iTop - 80 < 0 Then
            DI.iTopDir = 1
        End If
        If DI.iLeftDir = 1 Then
            CP.iLeft = CP.iLeft + iSpeed
        ElseIf DI.iLeftDir = 0 Then
            CP.iLeft = CP.iLeft - iSpeed
        End If
        If DI.iTopDir = 1 Then
            CP.iTop = CP.iTop + iSpeed
        ElseIf DI.iTopDir = 0 Then
            CP.iTop = CP.iTop - iSpeed
        End If

        pbMovePic.PaintPicture pbMovePic.Picture, 0, 0, , , , , vbDstInvert
        pbMovePic.PaintPicture pbColPic.Picture, 0, 0, , , , , vbSrcAnd
        pbMovePic.Picture = pbMovePic.Image
        objDrawCon.AutoRedraw = True
        objDrawCon.Cls
        objDrawCon.PaintPicture pbMaskPic.Picture, CP.iLeft, CP.iTop, , , , , vbSrcAnd
        objDrawCon.PaintPicture pbMovePic.Picture, CP.iLeft, CP.iTop, , , , , vbSrcPaint
        objDrawCon.Refresh
        objDrawCon.AutoRedraw = False
        DoEvents
    Loop
End Function

Private Sub Form_Load()
    Picture1.Picture = LoadPicture(App.Path & "\1.bmp")
    Picture2.Picture = LoadPicture(App.Path & "\2.bmp")

    On Error Resume Next
    'Aendern Sie hier den Pfad fuer das Testhintergrundbild ab.
    Me.Picture = LoadPicture("c:\windows\wolken.bmp")
    Me.Show
    fMoveDirection Form1, Picture1, Picture2, Picture3, 150, &HFF8080
End Sub

Private Sub Form_Unload(Cancel As Integer)
```

```

    Set Form1 = Nothing
End
End Sub

```

Tipp : 28.

'Beispiel : Grafik aufrollen. Von links nach rechts.
 Dim bOnOff As Boolean

```

Function fPi() As Double
    fPi = 4 * Atn(1)
End Function

```

```

Private Sub Command2_Click(Index As Integer)
    Select Case Index
        Case 0
            Command2(0).Enabled = False
            Command2(1).Enabled = False
            bOnOff = True
            sSetCube 40, Picture1, Picture2
            Command2(0).Enabled = True
            Command2(1).Enabled = True
        Case 1
            Command2(1).Enabled = False
            Command2(0).Enabled = False
            bOnOff = True
            sSetCube 100, Picture1, Picture2
            Command2(1).Enabled = True
            Command2(0).Enabled = True
    End Select
End Sub

```

```

Private Function fCosinus(dXPos As Double) As Double
    dXPos = dXPos - 1
    If dXPos < 1 And dXPos > -1 Then
        fCosinus = Atn(-dXPos / Sqr(-dXPos * dXPos + 1)) + 2 * Atn(1)
    Else
        fCosinus = 0
    End If
End Function

```

```

Private Sub Command1_Click()
    Unload Me
End
End Sub

```

```

Private Sub Command3_Click()
    bOnOff = False
End Sub

```

```

Private Sub Form_Load()
    On Error Resume Next
    Picture1.Picture = LoadPicture(App.Path & "\test.jpg")
    With Picture2
        .Width = Picture1.Width
        .Height = Picture1.Height
    End With
End Sub

```

```

Private Sub sSetCube(ICubeWidth, pb1 As PictureBox, _
    pb2 As PictureBox)

```

```

    Dim lAddXPic As Long
    Dim dRet As Double
    Dim i As Long
    Dim n As Long

```

```

    pb2.Cls
    dRet = fPi * 2 / (ICubeWidth * 2)
    For n = 0 To pb1.ScaleWidth - 1

```

```

    If bOnOff = False Then Exit Sub

```

```

    If n - ICubeWidth >= 0 Then pb2.PaintPicture pb1.Picture, _
        n - ICubeWidth, 0, 1, pb1.ScaleHeight, n - ICubeWidth, _

```

```

0, 1, pb1.ScaleHeight

For i = 1 To ICubeWidth
    IAddXPic = fCosinus(i / (ICubeWidth / 2)) / dRet
    If n + IAddXPic < pb1.ScaleWidth Then

        pb2.PaintPicture pb1.Picture, n + i - ICubeWidth, _
            0, 1, pb1.ScaleHeight, n + IAddXPic, 0, 1, _
            pb1.ScaleHeight

    Else

        pb2.PaintPicture pb1.Picture, n + i - ICubeWidth, _
            0, 1, pb1.ScaleHeight, n + i - ICubeWidth, 0, 1, _
            pb1.ScaleHeight

    End If
    DoEvents
Next i
Next n
End Sub

```

Tipp : 29.

'Beispiel : Grafik aufrollen. Von rechts nach links.

```
Dim bOnOff As Boolean
```

```
Function fPi() As Double
    fPi = 4 * Atn(1)
End Function
```

```
Private Sub Command2_Click(Index As Integer)
    Select Case Index
        Case 0
            Command2(0).Enabled = False
            Command2(1).Enabled = False
            bOnOff = True
            sSetCube 40, Picture1, Picture2
            Command2(0).Enabled = True
            Command2(1).Enabled = True
        Case 1
            Command2(1).Enabled = False
            Command2(0).Enabled = False
            bOnOff = True
            sSetCube 100, Picture1, Picture2
            Command2(1).Enabled = True
            Command2(0).Enabled = True
    End Select
End Sub

```

```
Private Function fCosinus(dXPos As Double) As Double
    dXPos = dXPos - 1
    If dXPos < 1 And dXPos > -1 Then
        fCosinus = Atn(-dXPos / Sqr(-dXPos * dXPos + 1)) + 2 * Atn(1)
    Else
        fCosinus = 0
    End If
End Function

```

```
Private Sub Command1_Click()
    Unload Me
End Sub

```

```
Private Sub Command3_Click()
    bOnOff = False
End Sub

```

```
Private Sub Form_Load()
    On Error Resume Next
    Picture1.Picture = LoadPicture(App.Path & "test.jpg")
    With Picture2
        .Width = Picture1.Width
        .Height = Picture1.Height
    End With
End Sub

```



```

Private Sub sSetCube(ICubeWidth, pb1 As PictureBox, _
    pb2 As PictureBox)

    Dim IAddXPic As Long
    Dim dRet As Double
    Dim i As Long
    Dim n As Long

    pb2.Cls
    dRet = fPi * 2 / (ICubeWidth * 2)
    For n = pb1.ScaleWidth To 0 Step -1
    If bOnOff = False Then Exit Sub
    If n - ICubeWidth >= 0 Then pb2.PaintPicture pb1.Picture, _
        n - ICubeWidth, 0, 1, pb1.ScaleHeight, n - ICubeWidth, _
        0, 1, pb1.ScaleHeight

    For i = 1 To ICubeWidth
        IAddXPic = fCosinus(i / (ICubeWidth / 2)) / dRet
        If n + IAddXPic < pb1.ScaleWidth Then

            pb2.PaintPicture pb1.Picture, n + i - ICubeWidth, _
                0, 1, pb1.ScaleHeight, n + IAddXPic, 0, 1, _
                pb1.ScaleHeight

        Else

            pb2.PaintPicture pb1.Picture, n + i - ICubeWidth, _
                0, 1, pb1.ScaleHeight, n - i - ICubeWidth, 0, 1, _
                pb1.ScaleHeight

        End If
        DoEvents
    Next i
    Next n
End Sub

```

Tipp : 30.

'Beispiel : Grafik ueberblenden per API Funktionen.

```

Private Declare Function SetPixel Lib "gdi32" _
    (ByVal hdc As Long, _
    ByVal x As Long, _
    ByVal y As Long, _
    ByVal crColor As Long) _
    As Long

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

Private Sub Command1_Click()
    fBlendPicture
End Sub

Private Sub Command2_Click()
    Unload Me
End Sub

Private Sub Form_Load()
    sLoadFile Picture1, Picture2
    Picture1.ScaleMode = 3
    Picture2.ScaleMode = 3
    With Picture3
        .ScaleMode = 3
        .Width = Picture2.Width
    End With
End Sub

Private Sub sLoadFile(pbSource1 As PictureBox, pbSource2 As PictureBox)
    On Error GoTo ErrHandle
    pbSource1.Picture = LoadPicture(App.Path & "\schuhe.jpg")

```

```

    pbSource2.Picture = LoadPicture(App.Path & "\mallorca.jpg")
    Exit Sub
ErrHandle:
    MsgBox Err.Description, vbCritical, "Error..."
    End
End Sub

Private Function fBlendPicture()
    Dim n          As Long
    Dim k          As Long

    For k = 0 To Picture1.ScaleHeight
        For n = 0 To Picture1.ScaleWidth
            Call SetPixel(Picture3.hdc, n, k, _
                fColoratedBlending(GetPixel(Picture1.hdc, n, k), _
                GetPixel(Picture2.hdc, n, k), 70, 70, 70))
            DoEvents
        Next n
    Next k
End Function

```

```

Private Function fColoratedBlending(ByVal IPicColorsA As Long, _
    ByVal IPicColorsB As Long, _
    iRedValue As Integer, _
    iGreenValue As Integer, _
    iBlueValue As Integer) _
    As Long

    Dim IResRed          As Long
    Dim IResGreen        As Long
    Dim IResBlue         As Long

    IResRed = (((IPicColorsB And &HFF) * iRedValue) + _
        ((IPicColorsA And &HFF) * iRedValue)) \ 100

    IResGreen = (((IPicColorsB \ &H100) And &HFF) * _
        iGreenValue) + (((IPicColorsA \ &H100) And &HFF) * _
        iGreenValue) \ 100

    IResBlue = (((IPicColorsB \ &H10000) And &HFF) * _
        iBlueValue) + (((IPicColorsA \ &H10000) And &HFF) * _
        iBlueValue) \ 100

    fColoratedBlending = RGB(IResRed, IResGreen, IResBlue)
End Function

```

Tipp : 31.

'Beispiel : Grafik ueberblenden 2.

```

Private Sub sLoadFile(pbSource1 As PictureBox, pbSource2 As PictureBox)
    On Error GoTo ErrHandle
    pbSource1.Picture = LoadPicture(App.Path & "\schuhe.jpg")
    pbSource2.Picture = LoadPicture(App.Path & "\mallorca.jpg")
    Exit Sub
ErrHandle:
    MsgBox Err.Description, vbCritical, "Error..."
    End
End Sub

Private Function fBlendPicture(pbDim As PictureBox)
    Dim n          As Long
    Dim k          As Long

    For k = 0 To pbDim.ScaleHeight
        For n = 0 To pbDim.ScaleWidth
            Picture3.PSet (n, k), fColoratedBlending(pbDim.Point(n, k), _
                Picture2.Point(n, k), 90, 90, 90)
        Next n
    Next k
End Function

Private Function fColoratedBlending(ByVal IPicColorsA As Long, _
    ByVal IPicColorsB As Long, _
    iRedValue As Integer, _
    iGreenValue As Integer, _

```

```

        iBlueValue As Integer) _
        As Long

Dim IResRed          As Long
Dim IResGreen       As Long
Dim IResBlue        As Long

IResRed = (((IPicColorsB And &HFF) * iRedValue) + _
((IPicColorsA And &HFF) * iRedValue)) \ 100

IResGreen = (((IPicColorsB \ &H100) And &HFF) * _
iGreenValue) + (((IPicColorsA \ &H100) And &HFF) * _
iGreenValue) \ 100

IResBlue = (((IPicColorsB \ &H10000) And &HFF) * _
iBlueValue) + (((IPicColorsA \ &H10000) And &HFF) * _
iBlueValue) \ 100

fColoratedBlending = RGB(IResRed, IResGreen, IResBlue)
End Function

Private Sub Command1_Click()
    fBlendPicture Picture1
End Sub

Private Sub Command2_Click()
    Unload Me
End Sub

Private Sub Form_Load()
    sLoadFile Picture1, Picture2
    Picture1.ScaleMode = 3
    Picture2.ScaleMode = 3
    With Picture3
        .ScaleMode = 3
        .Width = Picture2.Width
    End With
End Sub

```

Tipp : 32.

'Beispiel : Online-Verbindung im Sekudentakt zaehlen.

```

Private Declare Function InternetGetConnectedState Lib "wininet.dll" _
    (ByRef lpdwFlags As Long, _
    ByVal dwReserved As Long) _
    As Long

Private Const sTrueMsg As String = "Online-Verbindung ist aktiv."
Private Const sFalseMsg As String = "Online-Verbindung ist inaktiv."
Private IRet As Long

Private Function fGetConnectionState() As Boolean
    fGetConnectionState = InternetGetConnectedState(0&, 0&)
End Function

Private Sub Command2_Click()
    Set Form1 = Nothing
    Unload Me
End Sub

Private Sub Timer1_Timer()
    If fGetConnectionState Then
        Timer2.Enabled = True
        Label2 = sTrueMsg
    Else
        Timer2.Enabled = False
        Label2 = sFalseMsg
    End If

    If IRet > 0 And Not fGetConnectionState Then
        Me.Caption = "Sie waren : " & CStr(IRet) & " Sekunden Online."
    End If
End Sub

```

```

Private Sub Timer2_Timer()
    Static i As Long
    i = i + 1000
    IRet = (i / 1000)
    Me.Caption = "Sie sind : " & CStr(IRet) & " Sekunden Online."
End Sub

```

Tipp : 33.

'Beispiel : Online-Zeit und Kosten aufzeichnen.

```

'IN MODUL1
Public Declare Function InternetGetConnectedState Lib "wininet.dll" _
    (ByRef lpdwFlags As Long, _
    ByVal dwReserved As Long) _
    As Long

Public Declare Function SetWindowPos Lib "user32" _
    (ByVal hWnd As Long, _
    ByVal hWndInsertAfter As Long, _
    ByVal x As Long, _
    ByVal y As Long, _
    ByVal cx As Long, _
    ByVal cy As Long, _
    ByVal wFlags As Long) _
    As Long

Public Declare Function SystemParametersInfo Lib "user32" _
    Alias "SystemParametersInfoA" _
    (ByVal uAction As Long, _
    ByVal uParam As Long, _
    aRect As Any, _
    ByVal fuWinIni As Long) _
    As Long

Public Declare Function InternetAutodial Lib "wininet.dll" _
    (ByVal dwFlags As Long, _
    ByVal dwReserved As Long) _
    As Long

Public Declare Function InternetAutodialHangup Lib "wininet.dll" _
    (ByVal dwReserved As Long) _
    As Long

Public Type RECT
    Left As Long
    Top As Long
    Right As Long
    Bottom As Long
End Type

Public Const SPI_GETWORKAREA = 48
Public Const INTERNET_AUTODIAL_FORCE_ONLINE = 1
Public Const wFlag = 33

Public sPath As String

Public Function fCompletePath(ByVal sPath As String) As String
    If Right$(sPath, 1) <> "\" Then
        sPath = sPath + "\"
    End If
    fCompletePath = sPath
End Function

Public Sub fFormCenter(frm As Form)
    Dim IRet As Long
    Dim IWidth As Long
    Dim IHeight As Long
    Dim RC As RECT

    IRet = SystemParametersInfo(SPI_GETWORKAREA, 0, RC, 0)
    If IRet Then
        With RC
            .Left = Screen.TwipsPerPixelX * .Left
            .Top = Screen.TwipsPerPixelY * .Top
            .Right = Screen.TwipsPerPixelX * .Right

```

```

        .Bottom = Screen.TwipsPerPixelY * .Bottom
        IWidth = .Right - .Left
        IHeight = .Bottom - .Top
        frm.Move .Left + (IWidth - frm.Width) \ 2, _
            .Top + (IHeight - frm.Height) \ 2
    End With
End If
End Sub

```

'In Form2

```

Private Sub Command1_Click()
    Unload Me
End Sub

```

```

Private Sub Form_Activate()
    sOpenFile sPath
    Me.Caption = "Logfile : " & fFileName(sPath)
End Sub

```

```

Private Function fFileName(sPath As String)
    Dim sTmpPath As String
    Dim iLen As Integer
    Dim iRet As Integer

    sTmpPath = sPath
    Do Until InStr(sTmpPath, "\") = 0
        iLen = Len(sTmpPath)
        iRet = InStr(sTmpPath, "\")
        sTmpPath = Right(sTmpPath, iLen - iRet)
    Loop
    fFileName = sTmpPath
End Function

```

```

Private Sub Form_Load()
    Set Me.Icon = Form1.Icon
    fFormCenter Me
End Sub

```

```

Private Function sOpenFile(sLogName As String)
    Dim iFileNumber As Integer
    Dim sRetStr As String

    On Error GoTo ErrHandle
    iFileNumber = FreeFile
    Open sLogName For Input As #iFileNumber
    sRetStr = Input(LOF(iFileNumber), #iFileNumber)
    Text1 = sRetStr
    Close iFileNumber
ErrHandle:
    Exit Function
End Function

```

```

Private Sub fPrint(sPath As String)
    Dim sPrintString As String
    Dim iFileNumber As Integer

    iFileNumber = FreeFile
    On Error GoTo ErrHandle
    Open sPath For Input As iFileNumber
    Do While Not EOF(iFileNumber)
        Line Input #iFileNumber, sPrintString
        Printer.Print sPrintString
    Loop
    Printer.EndDoc
    Close #iFileNumber
Exit Sub
ErrHandle:
    MsgBox "Allgemeiner Fehler : " & Err.Description, _
        vbInformation, "Fehlermeldung..."
End Sub

```

```

Private Sub Command2_Click()
    fPrint sPath
End Sub

```

```

Private Function fClipboard(obj As TextBox)
    On Error Resume Next

```

```
Clipboard.Clear
Clipboard.SetText obj
End Function
```

```
Private Sub Command4_Click()
    fClipboard.Text1
End Sub
```

'In Form1

```
Private Const sTrueMsg As String = "Online-Verbindung ist aktiv."
Private Const sFalseMsg As String = "Online-Verbindung ist inaktiv."
```

```
Private dRet As Double
Private dRes As Double
```

```
Private Sub sAlwaysOnTop(frm As Form, OnTop As Boolean)
    Dim lPosLeft As Long
    Dim lPosTop As Long

    lPosLeft = 0
    lPosTop = 0
    If OnTop = True Then
        SetWindowPos frm.hWnd, OnTop, lPosLeft, lPosTop, 0, 0, wFlag
    End If
End Sub
```

```
Private Function flsNumeric(tbTxt As TextBox, _
    ByVal sChar As String) _
    As Boolean
    flsNumeric = True
    Select Case Chr(sChar)
        Case "0" To "9"
            '
        Case ","
            If InStr(tbTxt.Text, ",") <> 0 Then
                flsNumeric = False
            End If
        Case Chr(vbKeyBack)
            '
        Case Else
            flsNumeric = False
    End Select
End Function
```

```
Private Sub sHelpText()
    Dim t As String
    t = "Hilfe." & vbCrLf
    t = t & "" & vbCrLf
    t = t & "Die einzelnen Verbindungen mit den Werten" & vbCrLf
    t = t & "werden in <time.dat> gespeichert." & vbCrLf
    t = t & "" & vbCrLf
    t = t & "Starten Sie eine DFUE - Verbindung um" & vbCrLf
    t = t & "das Beispiel zu testen."
    MsgBox t, vbInformation + vbOKOnly, "Help..."
End Sub
```

```
Private Sub Command1_Click()
    sHelpText
End Sub
```

```
Private Sub Command3_Click()
    Form2.Show 1
End Sub
```

```
Private Sub Command4_Click()
    Dim lRet As Long
    lRet = InternetAutodial(INTERNET_AUTODIAL_FORCE_ONLINE, 0)
    Command4.Enabled = False
End Sub
```

```
Private Sub Command5_Click()
    Dim lRet As Long
    lRet = InternetAutodialHangup(0)
    Command5.Enabled = False
End Sub
```

```
Private Sub Form_Resize()
```

```

    If WindowState = vbNormal Then
        sAlwaysOnTop Me, True
    End If
End Sub

Private Sub Text1_Click()
    With Text1
        .SelStart = 0
        .SelLength = Len(Text1)
    End With
End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer)
    If flsNumeric(Text1, KeyAscii) = False Then
        KeyAscii = 0
    End If
End Sub

Private Function fCostValue(obj As TextBox)
    If obj = vbNullString Then
        obj = "2,9"
    End If

    dRes = (dRet * CDbl(obj.Text / 60))
    Text2 = Round(dRes, 1)
End Function

Private Function Round(cCostVal As Double, iDecimalPoint As Integer) As Double
    Round = Int((cCostVal * 10 ^ iDecimalPoint) + 0.5) / 10 ^ iDecimalPoint
End Function

Private Sub sSaveTimeContent(sFile As String)
    Dim sValue As String
    Dim iFileNumber As Integer

    sValue = Date & " " & Time & " " & dRet & " " & Round(dRes, 2)
    iFileNumber = FreeFile
    Open sFile For Append As #iFileNumber
    Print #iFileNumber, sValue
    Close #iFileNumber
End Sub

Private Sub Form_Activate()
    sAlwaysOnTop Me, True
    With Text1
        .MaxLength = 3
    End With
End Sub

Private Sub Form_Load()
    sAppPrevInstance Me
    sPath = fCompletePath(App.Path) & "time.dat"
    fFileExists sPath
End Sub

Private Function fGetConnectionState() As Boolean
    fGetConnectionState = InternetGetConnectedState(0&, 0&)
End Function

Private Sub Command2_Click()
    Set Form1 = Nothing
    Unload Me
End Sub

Private Sub Form_Unload(Cancel As Integer)
    sSaveTimeContent sPath
End Sub

Private Sub Timer1_Timer()
    If fGetConnectionState Then
        Timer2.Enabled = True
        Label2 = sTrueMsg
        fCostValue Text1
        Command5.Enabled = True
    Else
        Timer2.Enabled = False
        Label2 = sFalseMsg
    End If
End Sub

```



```

        sFileName = Right(sFileName, Len(sFileName) - 1)
    End If
End If
n = n + 1
Loop

fReplaceNotLegalCharInFileName = sFileName
MsgBox sFileName, , "Message..."
End Function

Private Sub Command2_Click()
    Unload Me
End Sub

```

Tipp : 35.

'Beispiel : Grafik zoomen.
'Hinweis : Auf eine Fehlerbehandlung beim laden der Grafik
' wurde verzichtet.

```

Private Type SCALEFACTOR
    iFactor      As Integer
    iSubtract    As Integer
End Type

Private Sub Command1_Click(Index As Integer)
    Dim SF      As SCALEFACTOR
    SF.iSubtract = 5
    Select Case Index
        Case 0
            'ca. 120%
            SF.iFactor = 180 - SF.iSubtract
            Call sBitmapZoom(Picture2, SF.iFactor, Picture1)
        Case 1
            'ca. 150%
            SF.iFactor = 250 - SF.iSubtract
            Call sBitmapZoom(Picture2, SF.iFactor, Picture1)
        Case 2
            'ca. 80%
            SF.iFactor = 120
            Call sBitmapZoom(Picture2, SF.iFactor, Picture1)
        Case 3
            'ca. 50%
            SF.iFactor = 80
            Call sBitmapZoom(Picture2, SF.iFactor, Picture1)
    End Select
End Sub

Private Sub Form_Load()
    Me.ScaleMode = 3
    With Picture1
        .ScaleMode = 3
        .BackColor = &H8000000F
        .AutoRedraw = True
    End With
    With Picture2
        .Visible = False
        .Picture = LoadPicture(App.Path & "\test1.jpg")
    End With
End Sub

Private Sub sBitmapZoom(pbSource As PictureBox, _
    ByVal iScaleFactor As Integer, _
    pbPaint As PictureBox)

    Dim lSourceWidth      As Long
    Dim lSourceHeight     As Long
    Dim lTmpX             As Long
    Dim lTmpY             As Long

    Picture1.Cls
    lSourceWidth = pbSource.Width
    lSourceHeight = pbSource.Height
    lTmpX = lSourceWidth * iScaleFactor / 100
    lTmpY = lSourceHeight * iScaleFactor / 100

```

```

Label1 = "Weite : " & ITmpX & " Pixel"
Label2 = "Hoehe : " & ITmpY & " Pixel"
pbPaint.PaintPicture pbSource, 0, 0, ITmpX, ITmpY
End Sub

```

Tipp : 36.

'Beispiel : Inhalt einer obj mit Seitenbegrenzungen drucken.

'Hinweis : Verwendet Microsoft RichTextBox Control.

```

Private Declare Function GetDeviceCaps Lib "gdi32" _
    (ByVal hdc As Long, _
    ByVal nIndex As Long) _
    As Long

Private Declare Function SendMessage Lib "user32" _
    Alias "SendMessageA" _
    (ByVal hwnd As Long, _
    ByVal wParam As Long, _
    ByVal lParam As Any) _
    As Long

Private Type MARGINORIENTATION
    IMarginLeft           As Long
    IMarginTop            As Long
    IMarginRight          As Long
    IMarginBottom         As Long
End Type

Private Const WM_USER = &H400
Private Const WM_ARRANGE As Long = WM_USER + 57

Private Const PHYSICALOFFSETX As Long = 112
Private Const PHYSICALOFFSETY As Long = 113

Private Type RECT
    Left           As Long
    Top            As Long
    Right          As Long
    Bottom         As Long
End Type

Private Type CHARORIENTATION
    IMin           As Long
    IMax           As Long
End Type

Private Type FORMATORIENTATION
    Ihdc           As Long
    IPrinterHdc   As Long
    RC             As RECT
    RCPRINT       As RECT
    C_OR          As CHARORIENTATION
End Type

Public Function fPrintTextWithMargin(obj As RichTextBox, _
    IMarginWidthLeft As Long, _
    IMarginHeightTop As Long, _
    IMarginWidthRight As Long, _
    IMarginHeightBottom As Long) _
    As Boolean

    On Error GoTo ErrHandle

    Dim RCPAINT As RECT
    Dim RCPRINT As RECT
    Dim M_OR As MARGINORIENTATION
    Dim F_OR As FORMATORIENTATION
    Dim IRet As Long
    Dim IOffsetX As Long
    Dim IOffsetY As Long
    Dim IGetCharPos As Long
    Dim ITextLen As Long

    Screen.MousePointer = vbHourglass

```

```

Printer.ScaleMode = vbTwips
IOffsetX = Printer.ScaleX(GetDeviceCaps(Printer.hdc, _
    PHYSICALOFFSETX), _
    vbPixels, _
    vbTwips)

IOffsetY = Printer.ScaleY(GetDeviceCaps(Printer.hdc, _
    PHYSICALOFFSETY), _
    vbPixels, _
    vbTwips)

M_OR.IMarginLeft = IMarginWidthLeft - IOffsetX
M_OR.IMarginTop = IMarginHeightTop - IOffsetY
M_OR.IMarginRight = (Printer.Width - IMarginWidthRight) - IOffsetX
M_OR.IMarginBottom = (Printer.Height - IMarginHeightBottom) - IOffsetY
RCPRINT.Left = 0
RCPRINT.Top = 0
RCPRINT.Right = Printer.ScaleWidth
RCPRINT.Bottom = Printer.ScaleHeight
RCPAINT.Left = M_OR.IMarginLeft
RCPAINT.Top = M_OR.IMarginTop
RCPAINT.Right = M_OR.IMarginRight
RCPAINT.Bottom = M_OR.IMarginBottom
F_OR.lhdc = Printer.hdc
F_OR.IPrinterHdc = Printer.hdc
F_OR.RC = RCPAINT
F_OR.RCPRINT = RCPRINT
F_OR.C_OR.IMin = 0
F_OR.C_OR.IMax = -1
ITextLen = Len(obj.Text)
Do
    F_OR.lhdc = Printer.hdc
    F_OR.IPrinterHdc = Printer.hdc
    IGetCharPos = SendMessage(obj.hwnd, WM_ARRANGE, True, F_OR)
    If IGetCharPos >= ITextLen Then Exit Do
    F_OR.C_OR.IMin = IGetCharPos
    Printer.NewPage
Loop
Printer.EndDoc
IRet = SendMessage(obj.hwnd, WM_ARRANGE, False, ByVal CLng(0))
fPrintTextWithMargin = True
Screen.MousePointer = vbNormal
Exit Function
ErrHandle:
    fPrintTextWithMargin = False
End Function

Private Sub Command1_Click()
    fPrintTextWithMargin RichTextBox1, 3000, 3000, 3000, 3000
End Sub

Private Sub Command2_Click()
    Unload Me
End
End Sub

```

Tipp : 37.

'Beispiel : [Online-Verbindungen auflisten und beenden.](#)

```

Private Declare Function RasGetConnectStatus Lib "RasAPI32.dll" _
    Alias "RasGetConnectStatusA" _
    (ByVal hRasConnection As Long, _
    ByRef state As RASCONNSTATUS) _
    As Long

Private Declare Sub Sleep Lib "kernel32.dll" _
    (ByVal dwMilliseconds As Long)

Private Declare Function timeGetTime Lib "winmm.dll" () _
    As Long

Private Declare Function RasGetErrorString Lib "RasAPI32.dll" _
    Alias "RasGetErrorStringA" _
    (ByVal errorNum As Long, _
    ByVal errorString As String, _

```

```
ByVal lenString As Long) _  
As Long
```

```
Private Declare Function RasEnumConnections Lib "RasAPI32.dll" _  
Alias "RasEnumConnectionsA" _  
(ByRef lpRasCon As RASCONN, _  
ByRef lpcb As Long, _  
ByRef lpNumConnections As Long) _  
As Long
```

```
Private Declare Function RasHangUp Lib "RasAPI32.dll" _  
Alias "RasHangUpA" _  
(ByVal hRasConnection As Long) _  
As Long
```

```
Private Type RASCONN  
dwSize As Long  
hRasConn As Long  
szEntryName(256) As Byte  
szDeviceType(16) As Byte  
szDeviceName(129) As Byte  
End Type
```

```
Private Type RASCONNSTATUS  
dwSize As Long  
rasState As RASCONNSTATE  
dwError As Long  
szDeviceType(16) As Byte  
szDeviceName(130) As Byte  
End Type
```

```
Private Const RASCS_DONE As Long = &H2000  
Private Const RASCS_PAUSED As Long = &H1000
```

```
Private Const ERROR_INVALID_HANDLE As Long = 6  
Private bCancel As Boolean
```

```
Public Enum RASCONNSTATE  
RASCS_OpenPort = 0  
RASCS_PortOpened = 1  
RASCS_ConnectDevice = 2  
RASCS_DeviceConnected = 3  
RASCS_AllDevicesConnected = 4  
RASCS_Authenticate = 5  
RASCS_AuthNotify = 6  
RASCS_AuthRetry = 7  
RASCS_AuthCallback = 8  
RASCS_AuthChangePassword = 9  
RASCS_AuthProject = 10  
RASCS_AuthLinkSpeed = 11  
RASCS_AuthAck = 12  
RASCS_ReAuthenticate = 13  
RASCS_Authenticated = 14  
RASCS_PrepareForCallback = 15  
RASCS_WaitForModemReset = 16  
RASCS_WaitForCallback = 17  
RASCS_Projected = 18  
RASCS_StartAuthentication = 19  
RASCS_CallbackComplete = 20  
RASCS_LogonNetwork = 21  
RASCS_SubEntryConnected = 22  
RASCS_SubEntryDisconnected = 23  
  
RASCS_Interactive = RASCS_PAUSED  
RASCS_RetryAuthentication = RASCS_PAUSED + 1  
RASCS_CallbackSetByCaller = RASCS_PAUSED + 2  
RASCS_PasswordExpired = RASCS_PAUSED + 3  
  
RASCS_Connected = RASCS_DONE  
RASCS_Disconnected = RASCS_DONE + 1  
End Enum
```

```
Private Sub Command3_Click()  
Unload Me  
End  
End Sub
```

```

Private Sub Form_Load()
    Command2.Enabled = False
End Sub

Private Sub Command2_Click()
    bCancel = True
End Sub

Private Sub sEnumerateConnect(Ist As ListBox)
    Dim RC(12) As RASCONN
    Dim RCStatus As RASCONNSTATUS
    Dim RCState As RASCONNSTATE
    Dim IConnectionCount As Long
    Dim IEndCount As Long
    Dim IRECRet As Long
    Dim IBufferSize As Long
    Dim sErrMsg As String

    Command1.Enabled = False
    Command2.Enabled = True
    bCancel = False
    RC(0).dwSize = Len(RC(0))

    IBufferSize = RC(0).dwSize * 12
    IRECRet = RasEnumConnections(RC(0), IBufferSize, IConnectionCount)
    If IRECRet <> 0 Then
        sErrMsg = Space$(128)
        Call RasGetErrorString(IRECRet, sErrMsg, 128)
        MsgBox "Keine Online-Verbindung vorhanden : " & sErrMsg
    Else
        If IConnectionCount < 1 Then
            Ist.Clear
            Ist.AddItem "Keine Aktive Online - Verbindung vorhanden."
            Else
            With Ist
                .Clear
                .AddItem "Folgende Online-Verbindung war aktiv :"
            For IEndCount = 0 To IConnectionCount - 1
                .AddItem String(55, "-")
                .AddItem "Verbindung :" & Space(12) & IEndCount + 1
                .AddItem "Provider :" & Space(17) & fConvertByteToString(RC(IEndCount).szEntryName)
                .AddItem "Geraete-Type :" & Space(8) & fConvertByteToString(RC(IEndCount).szDeviceType)
                .AddItem "Geraete-Name :" & Space(7) & fConvertByteToString(RC(IEndCount).szDeviceName)
                .AddItem String(55, "-")
                IEndCount = IEndCount + 1
            Next
            .AddItem "Programm wird in 4 Sekunden beendet."
            For IEndCount = 3 To 1 Step -1
                sWaitFor 1000
                If bCancel Then
                    .AddItem "Verbindungsabbruch wurde unterbrochen."
                    GoTo BTControl
                End If
            Next
            Refresh
            For IEndCount = 0 To IConnectionCount - 1
                .AddItem "Verbindung wurde beendet... "
                Call RasHangUp(RC(IEndCount).hRasConn)
                RCState = -1
            Do
                RCStatus.dwSize = LenB(RCStatus)
                IRECRet = RasGetConnectStatus(RC(IEndCount).hRasConn, RCStatus)
                If IRECRet <> 0 Then
                    If IRECRet = ERROR_INVALID_HANDLE Then
                        .AddItem "Ende"
                        Exit Do
                    Else
                        sErrMsg = Space$(128)
                        Call RasGetErrorString(IRECRet, sErrMsg, 128)
                        MsgBox "Verbindungsfehler " & sErrMsg, , "Error..."
                        .AddItem "Fehlernummer : " & IRECRet
                    End Do
                End If
            Else
                If RCState <> RCStatus.rasState Then
                    RCState = RCStatus.rasState
                End If
            End If
        End Sub

```

```

        .AddItem "Verbindungsstatus : " & RCState
        .Refresh
    End If
    Call Sleep(0)
    End If
    Loop
    Next
    End With
End If
End If
BTControl:
    Command1.Enabled = True
    Command2.Enabled = False
End Sub

Private Sub Command1_Click()
    sEnumerateConnect List1
End Sub

Private Function fConvertByteToString(ByRef btarrByte() As Byte) As String
    Dim sResChar As String
    Dim n As Long

    On Error GoTo ErrHandle
    For n = 0 To UBound(btarrByte)
        sResChar = (Chr(btarrByte(n)))
        If sResChar <> vbNullChar Then
            fConvertByteToString = fConvertByteToString & sResChar
        Else
            Exit For
        End If
    Next
    Exit Function
ErrHandle:
    fConvertByteToString = ""
End Function

Private Sub sWaitFor(ByVal lSeconds As Long, _
    Optional bFreeSys As Boolean = True)
    Dim lBeginCount As Long
    lBeginCount = timeGetTime()
    Do While timeGetTime < lBeginCount + lSeconds
        If bFreeSys Then
            DoEvents
        End If
    Loop
End Sub

Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
    If Command2.Enabled = True Then
        Cancel = True
    End If
End Sub

```

Tipp : 38.

'Beispiel : Screen animiert Grau einfaerben.

```

Private Declare Function GetDesktopWindow Lib "user32" _
    () As Long

Private Declare Function GetDC Lib "user32" _
    (ByVal hwnd As Long) _
    As Long

Private Declare Function ReleaseDC Lib "user32" _
    (ByVal hwnd As Long, _
    ByVal hdc As Long) _
    As Long

Private Declare Function SetWindowPos Lib "user32" _
    (ByVal hwnd As Long, _
    ByVal hwndInsertAfter As Long, _
    ByVal X As Long, _
    ByVal Y As Long, _
    ByVal CX As Long, _

```

```
ByVal CY As Long, _  
ByVal wFlags As Long) _  
As Long
```

```
Private Declare Function BitBlt Lib "gdi32" _  
    (ByVal hDestDC As Long, _  
    ByVal X As Long, _  
    ByVal Y As Long, _  
    ByVal nWidth As Long, _  
    ByVal nHeight As Long, _  
    ByVal hSrcDC As Long, _  
    ByVal XSrc As Long, _  
    ByVal YSrc As Long, _  
    ByVal dwRop As Long) _  
    As Long
```

```
Dim n As Long  
Dim k As Long  
Dim pPicture As Picture  
Dim IRed As Long  
Dim IGreen As Long  
Dim IBlue As Long  
Dim IPoint As Long
```

```
Private Const COLORVALUE = 16777216  
Private Const SRCCOPY = &HCC0020
```

```
Function fColors(IColorPoints As Long, IR As Long, IG As Long, IB As Long)  
    IR = IColorPoints And &HFF  
    IG = (IColorPoints \ &H100) And &HFF  
    IB = (IColorPoints \ &H10000) And &HFF  
End Function
```

```
Private Sub Command1_Click()  
    End  
End Sub
```

```
Private Sub Form_Activate()  
    For n = 0 To Picture1.ScaleWidth - 1  
        For k = 0 To Picture1.ScaleHeight - 1  
            IPoint = Picture1.Point(n, k)  
            fColors IPoint, IRed, IGreen, IBlue  
            Picture1.PSet (n, k), RGB((IRed + IGreen + IBlue) \ 3, (IRed + IGreen + IBlue) \ 3, (IRed + IGreen + IBlue) \ 3)  
        Next k  
        DoEvents  
    Next n  
SysFree:  
    For n = 0 To Picture1.ScaleWidth - 1  
        For k = 0 To Picture1.ScaleHeight - 1  
            Picture1.PSet (n, k), COLORVALUE - Picture1.Point(n, k)  
        Next k  
        DoEvents  
    Next n  
    GoTo SysFree  
End Sub
```

```
Private Sub Form_Load()  
    Command1.Move 0, 0  
    flnit Form1, Picture1  
End Sub
```

```
Private Function flnit(frm As Form, pb As PictureBox)  
    Dim IHwndSource As Long  
    Dim IGetDC As Long  
    Dim IRet As Long  
  
    With frm  
        .Width = Screen.Width  
        .Height = Screen.Height  
        .ScaleMode = vbPixels  
        .AutoRedraw = True  
        .Move 0, 0, Screen.Width + 1, Screen.Height + 1  
    End With
```

```
IHwndSource = GetDesktopWindow()  
IGetDC = GetDC(IHwndSource)  
IRet = BitBlt(hdc, 0, 0, ScaleWidth, _
```

```
ScaleHeight, _  
IGetDC, 0, 0, _  
SRCCOPY)
```

```
IRet = ReleaseDC(IHwndSource, IGetDC)  
frm.Show  
Set pPicture = Image  
frm.WindowState = vbMaximized
```

```
With pb  
    .Width = Screen.Width \ Screen.TwipsPerPixelX  
    .Height = Screen.Height \ Screen.TwipsPerPixelY  
    Set .Picture = pPicture  
    .Enabled = False  
End With
```

End Function

Tipp : 39.

'Beispiel : Internet-Verbindung herstellen.

```
Private Declare Function InternetAutodial Lib "wininet.dll" _  
    (ByVal dwFlags As Long, _  
    ByVal dwReserved As Long) _  
    As Long  
  
Private Const INTERNET_AUTODIAL_FORCE_ONLINE = 1  
  
Private Sub Command1_Click()  
    Dim IRet As Long  
    IRet = InternetAutodial(INTERNET_AUTODIAL_FORCE_ONLINE, 0)  
End Sub  
  
Private Sub Command2_Click()  
    Unload Me  
End Sub
```

Tipp : 40.

'Beispiel : Internet-Verbindung herstellen und beenden.

```
Private Declare Function InternetAutodial Lib "wininet.dll" _  
    (ByVal dwFlags As Long, _  
    ByVal dwReserved As Long) _  
    As Long  
  
Private Declare Function InternetAutodialHangup Lib "wininet.dll" _  
    (ByVal dwReserved As Long) _  
    As Long  
  
Private Const INTERNET_AUTODIAL_FORCE_ONLINE = 1  
  
Private Sub Command1_Click()  
    Dim IRet As Long  
    IRet = InternetAutodial(INTERNET_AUTODIAL_FORCE_ONLINE, 0)  
End Sub  
  
Private Sub Command2_Click()  
    Unload Me  
End Sub  
  
Private Sub Command3_Click()  
    Dim IRet As Long  
    IRet = InternetAutodialHangup(0)  
End Sub
```

Tipp : 41.

'Beispiel : Vorhandene DFUE - Verbindungen ermitteln.

```
Private Declare Function RasEnumEntriesA Lib "RasApi32.DLL" _  
    (ByVal reserved As String, _
```



```

ByVal IpszPhonebook As String, _
lprasentryname As Any, _
lpcb As Long, _
lpcEntries As Long) _
As Long

```

```

Private Const RASMAXENTRYNAME = 256
Private Const RASMAXDWSIZE = 264

```

```

Private Type RASENTRYNAME95
    dwSize As Long
    szEntryname(RASMAXENTRYNAME) As Byte
End Type

```

```

Public Sub sGetProviderContact(arrContact() As String)
    Dim lLen As Long
    Dim sConvertToUnicode As String
    Dim lSize As Long
    Dim REN(RASMAXENTRYNAME - 1) As RASENTRYNAME95
    Dim n As Long

```

```

    REN(0).dwSize = RASMAXDWSIZE
    lSize = RASMAXENTRYNAME * REN(0).dwSize

```

```

    Call RasEnumEntriesA(vbNullString, vbNullString, REN(0), lSize, lLen)
    lLen = lLen - 1

```

```

    ReDim arrContact(lLen)
    For n = 0 To lLen
        sConvertToUnicode = StrConv(REN(n).szEntryname(), vbUnicode)
        arrContact(n) = Left$(sConvertToUnicode, InStr(sConvertToUnicode, vbNullChar) - 1)
    Next n

```

```

End Sub

```

```

Private Sub Command1_Click()
    Dim n As Integer
    Dim arrContact() As String

```

```

    sGetProviderContact arrContact
    For n = 0 To UBound(arrContact)
        List1.AddItem arrContact(n)
    Next
    List1.ListIndex = 0

```

```

End Sub

```

```

Private Sub Command2_Click()
    Unload Me
End Sub

```

Tipp : 42.

'Beispiel : Test ob Online-Verbindung besteht. Teil 2

```

Private Declare Function RasEnumConnections Lib "RasApi32.dll" _
    Alias "RasEnumConnectionsA" _
    (lpRasCon As Any, _
    lpcb As Long, _
    lpcConnections As Long) _
    As Long

```

```

Private Declare Function RasGetConnectStatus Lib "RasApi32.dll" _
    Alias "RasGetConnectStatusA" _
    (ByVal hRasCon As Long, _
    lpStatus As Any) _
    As Long

```

```

Private Const RAS95_MaxEntryName = 256
Private Const RAS95_MaxDeviceType = 16
Private Const RAS95_MaxDeviceName = 32
Private Const RASCS_DONE = &H2000
Private Const RASCS_CONNECTED = RASCS_DONE

```

```

Private Type RASCONN95
    dwSize As Long
    hRasCon As Long
    szEntryName(RAS95_MaxEntryName) As Byte
    szDeviceType(RAS95_MaxDeviceType) As Byte
    szDeviceName(RAS95_MaxDeviceName) As Byte

```

End Type

```
Private Type RASCONNSTATUS95
    dwSize As Long
    RasConnState As Long
    dwError As Long
    szDeviceType(RAS95_MaxDeviceType) As Byte
    szDeviceName(RAS95_MaxDeviceName) As Byte
End Type
```

```
Private Function fTestingOnlineConnect() As Boolean
    Dim RC(255) As RASCONN95
    Dim RCS As RASCONNSTATUS95
    Dim ldwSize As Long
    Dim lpcCon As Long
    Dim lRet As Long
```

```
    RC(0).dwSize = 412
    ldwSize = 256 * RC(0).dwSize
    lRet = RasEnumConnections(RC(0), ldwSize, lpcCon)
```

```
    If lRet <> 0 Then
        Exit Function
    End If
```

```
    RCS.dwSize = 160
    lRet = RasGetConnectStatus(RC(0).hRasCon, RCS)
    If RCS.RasConnState = RASCS_CONNECTED Then ' &H2000 = 9192
        fTestingOnlineConnect = True
        Label1 = fTestingOnlineConnect & " : " & " Es besteht eine Online-Verbindung."
    Else
        fTestingOnlineConnect = False
        Label1 = fTestingOnlineConnect & " : " & " Es besteht keine Online-Verbindung."
    End If
```

End Function

```
Private Sub Command1_Click()
    fTestingOnlineConnect
End Sub
```

```
Private Sub Command2_Click()
    Unload Me
End Sub
```

Tipp : 43.

Beispiel : Online Provider Auflisten und Starten.

```
Private Declare Function RasEnumEntries Lib "RasApi32.DLL" _
    Alias "RasEnumEntriesA" _
    (ByVal reserved As String, _
    ByVal lpszPhonebook As String, _
    lprasentryname As Any, _
    lpcb As Long, _
    lpcEntries As Long) _
    As Long
```

```
Private Const RAS95_MaxEntryName = 256
```

```
Private Type RASENTRYNAME95
    dwSize As Long
    szEntryName(RAS95_MaxEntryName) As Byte
End Type
```

```
Private Sub Command1_Click()
    Dim sRet As String
    sRet = "rundll maui.dll,RnaDial " & Me.List1.List(Me.List1.ListIndex)
    Shell sRet, vbNormalFocus
End Sub
```

```
Private Sub Command2_Click()
    Unload Me
End Sub
```

```

Private Sub Form_Load()
    ReDim REN(255) As RASENTRYNAME95
    Dim sConvertUni           As String
    Dim ldwSize              As Long
    Dim lRet                 As Long
    Dim lEntries             As Long

    REN(0).dwSize = 264
    ldwSize = 256 * REN(0).dwSize
    lRet = RasEnumEntries(vbNullString, vbNullString, REN(0), ldwSize, lEntries)
    For lRet = 0 To lEntries - 1
        sConvertUni = StrConv(REN(lRet).szEntryName(), vbUnicode)
        Me.List1.AddItem Left$(sConvertUni, InStr(sConvertUni, Chr$(0)) - 1)
    Next
    On Local Error Resume Next
    Me.List1.ListIndex = 0
End Sub

```

Tipp : 44.

'Beispiel : Online Provider Auflisten und Starten und Verbindung beenden.

```

Private Declare Function RasEnumEntries Lib "RasApi32.dll" _
    Alias "RasEnumEntriesA" _
    (ByVal reserved As String, _
    ByVal lpszPhonebook As String, _
    lprasentryname As Any, _
    lpcb As Long, _
    lpcEntries As Long) _
    As Long

Private Declare Function RasHangUp Lib "RasApi32.dll" _
    Alias "RasHangUpA" _
    (ByVal hRasConn As Long) _
    As Long

Private Declare Function RasEnumConnections Lib "RasApi32.dll" _
    Alias "RasEnumConnectionsA" _
    (lpRasCon As Any, _
    lpcb As Long, _
    lpcConnections As Long) _
    As Long

Private Declare Function RasGetConnectStatus Lib "RasApi32.dll" _
    Alias "RasGetConnectStatusA" _
    (ByVal hRasCon As Long, _
    lpStatus As Any) _
    As Long

Private Const RAS95_MaxEntryName = 256
Private Const RAS95_MaxDeviceType = 16
Private Const RAS95_MaxDeviceName = 32
Private Const RASCS_DONE = &H2000
Private Const RASCS_CONNECTED = RASCS_DONE

Private Type RASCONN95
    dwSize           As Long
    hRasCon          As Long
    szEntryName(RAS95_MaxEntryName) As Byte
    szDeviceType(RAS95_MaxDeviceType) As Byte
    szDeviceName(RAS95_MaxDeviceName) As Byte
End Type

Private Type RASCONNSTATUS95
    dwSize           As Long
    RasConnState     As Long
    dwError          As Long
    szDeviceType(RAS95_MaxDeviceType) As Byte
    szDeviceName(RAS95_MaxDeviceName) As Byte
End Type

Private Type RASENTRYNAME95
    dwSize           As Long
    szEntryName(RAS95_MaxEntryName) As Byte
End Type

```

```

Private IIsConnect As Long

Private Function fTestingOnlineConnect() As Boolean
    Dim RC(255)           As RASCONN95
    Dim RCS               As RASCONNSTATUS95
    Dim IdwSize           As Long
    Dim IpcCon            As Long
    Dim IRet              As Long

    RC(0).dwSize = 412
    IdwSize = 256 * RC(0).dwSize
    IRet = RasEnumConnections(RC(0), IdwSize, IpcCon)

    If IRet <> 0 Then
        Exit Function
    End If

    RCS.dwSize = 160
    IRet = RasGetConnectStatus(RC(0).hRasCon, RCS)
    If RCS.RasConnState = RASCS_CONNECTED Then '&H2000 = 9192
        fTestingOnlineConnect = True
        Frame1 = fTestingOnlineConnect & " : " & " Es besteht eine Online-Verbindung."
        IIsConnect = RC(0).hRasCon
        Command3.Enabled = True
        Command1.Enabled = False
    Else
        fTestingOnlineConnect = False
        Frame1 = fTestingOnlineConnect & " : " & " Es besteht keine Online-Verbindung."
        Command3.Enabled = False
        Command1.Enabled = True
    End If
End Function

Private Sub Command1_Click()
    Dim sRet As String
    sRet = "rundll rnaui.dll,RnaDial " & Me.List1.List(Me.List1.ListIndex)
    Shell sRet, vbNormalFocus
End Sub

Private Sub Command2_Click()
    Unload Me
End Sub

Private Sub Form_Load()
    fAddInstalledProvider
End Sub

Private Function fAddInstalledProvider()
    ReDim REN(255)           As RASENTRYNAME95
    Dim sConvertUni         As String
    Dim IdwSize             As Long
    Dim IRet                As Long
    Dim IEntries            As Long

    REN(0).dwSize = 264
    IdwSize = 256 * REN(0).dwSize
    IRet = RasEnumEntries(vbNullString, vbNullString, REN(0), IdwSize, IEntries)
    For IRet = 0 To IEntries - 1
        sConvertUni = StrConv(REN(IRet).szEntryName(), vbUnicode)
        Me.List1.AddItem Left$(sConvertUni, InStr(sConvertUni, Chr$(0)) - 1)
    Next
    On Local Error Resume Next
    Me.List1.ListIndex = 0
End Function

Private Sub Timer1_Timer()
    fTestingOnlineConnect
End Sub

Private Function fCloseConnect() As Integer
    Dim IRet                As Long

    On Error GoTo ErrHandle
    fCloseConnect = False
    IRet = RasHangUp(IIsConnect)

```

```

    If IRet = 0 Then
        fCloseConnect = True
    End If
    Exit Function
ErrHandle:
    fCloseConnect = False
    Exit Function
End Function

```

```

Private Sub Command3_Click()
    fCloseConnect
End Sub

```

Tipp : 45.

'Beispiel : Standard-Browser und Email Client aufrufen.

```

Private Declare Function ShellExecute Lib "shell32.dll" _
    Alias "ShellExecuteA" _
    (ByVal hwnd As Long, _
    ByVal lpOperation As String, _
    ByVal lpFile As String, _
    ByVal lpParameters As String, _
    ByVal lpDirectory As String, _
    ByVal nShowCmd As Long) _
    As Long

Private Sub Command1_Click()
    Dim IRet As Long
    IRet = ShellExecute(hwnd, "Open", "www.Visual-Basic5.de", "", _
    App.Path, 1)
End Sub

Private Sub Command2_Click()
    Dim IRet As Long
    IRet = ShellExecute(hwnd, "Open", "mailto:outa.space@t-online.de", "", _
    App.Path, 1)
End Sub

Private Sub Command3_Click()
    Unload Me
    End
End Sub

```

Tipp : 46.

'Beispiel : Zwei Listenfelder synchronisieren.

```

Private Sub Command3_Click()
    Dim n As Integer
    n = 0
    Do While n < List1.ListCount
        If List1.Selected(n) Then
            List1.RemoveItem n
            List2.RemoveItem n
        Else
            n = n + 1
        End If
    Loop
    List1.Refresh
    List2.Refresh
End Sub

Private Sub Command4_Click()
    Dim sCopyValue As String
    Dim n As Integer
    Dim k As Integer

    Clipboard.Clear
    sCopyValue = vbNullString
    For n = 0 To List1.ListCount - 1
        For k = 0 To List2.ListCount - 1
            If List1.Selected(n) Then
                If List1.Selected(k) Then

```

```

        sCopyValue = sCopyValue & List1.List(n) & " " & List2.List(k) + vbCrLf
    End If
End If
Next
Next
Clipboard.SetText sCopyValue
End Sub

Private Sub Form_Load()
    With List1
        .AddItem "Mogel@Packung.de"
        .AddItem "Gauner@Stück.de"
        .AddItem "Fernseh@Sender.de"
    End With
    With List2
        .AddItem "Elka Laden"
        .AddItem "Jack Ripper"
        .AddItem "Zwei Sat"
    End With
End With
End Sub

Private Sub List1_Click()
    sSynchronizeControls List1
End Sub

Private Sub List2_Click()
    sSynchronizeControls List2
End Sub

Private Sub sSynchronizeControls(LB As Control)
    Static iListPrevious As Integer
    Dim iListAct As Integer

    iListAct = LB.TopIndex
    If iListAct <> iListPrevious Then
        List1.TopIndex = iListAct
        List2.TopIndex = iListAct
        iListPrevious = iListAct
    End If
    List1.ListIndex = LB.ListIndex
    List2.ListIndex = LB.ListIndex
End Sub

Private Sub Text1_Click()
    Text1.SelStart = 0
    Text1.SelLength = Len(Text1)
End Sub

Private Sub Text2_Click()
    Text2.SelStart = 0
    Text2.SelLength = Len(Text2)
End Sub

Private Sub Command1_Click()
    If Text1 <> vbNullString And Text2 <> vbNullString Then
        List1.AddItem Text1
        List2.AddItem Text2
    Else
        MsgBox "Fuellen Sie bitte die Eingabefelder korrekt aus.", , "Eingabefehler..."
    End If
    Exit Sub
End Sub

Private Sub Command2_Click()
    Unload Me
End Sub

```

Tipp : 47.

'Beispiel : Verwendung der verschiedenen Resource-Datei Inhalten.

```

Private Declare Function GetTempPath Lib "kernel32" _
    Alias "GetTempPathA" _
    (ByVal nBufferLength As Long, _
    ByVal lpBuffer As String) _

```

As Long

```
Private Declare Function GetTempFilename Lib "kernel32" _
    Alias "GetTempFileNameA" _
    (ByVal lpszPath As String, _
    ByVal lpPrefixString As String, _
    ByVal wUnique As Long, _
    ByVal lpTempFilename As String) _
    As Long
```

```
Private Sub Command1_Click(Index As Integer)
    Select Case Index
        Case 0
            'Resource ID 1001 [.jpg] laden.
            Set Picture1.Picture = fLoadResPicture(1001, "CUSTOM")
        Case 1
            'Resource ID 1002 [.gif] laden.
            Set Picture1.Picture = fLoadResPicture(1002, "CUSTOM")
        Case 2
            'Resource ID 1003 [.bmp] laden.
            Set Picture1.Picture = fLoadResPicture(1003, "CUSTOM")
        Case 3
            Unload Me
            End
    End Select
End Sub
```

```
Private Function fLoadResPicture(ByVal IResId As Long, _
    ByVal IResType As String, _
    Optional vTmp As Variant) _
    As Picture
    'Rueckgabe eines Picture Objekts aus der Resource - Datei.
    Dim sFileName As String

    If IsMissing(vTmp) Then
        Call fTraceTmpFile("", "~VB", 0, sFileName)
    Else
        sFileName = vTmp
    End If
    If fSaveltem(IResId, IResType, sFileName) = 0 Then
        Set fLoadResPicture = LoadPicture(sFileName)
        'Testen Sie hier die Anlage der temporaeren
        'Datei indem Sie <Kill> auskommentieren,
        'die temporaeren Grafikdateien im Temp-Pfad
        'umbenennen und in einem Grafikeditor oeffnen.
        Kill sFileName
    End If
End Function
```

```
Private Function fSaveltem(ByVal iResNumber As Integer, _
    ByVal IResType As String, _
    ByVal sDestination As String) _
    As Long

    Dim iFileNumber As Integer
    Dim arrRes() As Byte

    On Error GoTo ErrHandle
    arrRes = LoadResData(iResNumber, IResType)
    iFileNumber = FreeFile
    Open sDestination For Binary Access Write As #iFileNumber
    Put #iFileNumber, , arrRes
    Close #iFileNumber
    fSaveltem = 0
    Exit Function
ErrHandle:
    fSaveltem = Err.Number
End Function
```

```
Private Function fTraceTmpFile(ByVal sPath As String, _
    ByVal sPrefix As String, _
    ByVal iUnique As Integer, _
    sTmpFile As String) _
    As Boolean

    If Len(sPath) = 0 Then
        sPath = String$(255, vbNullChar)
```

```

    If GetTempPath(255, sPath) = 0 Then
        fTraceTmpFile = False
        Exit Function
    End If
End If
sTmpFile = String$(255, vbNullChar)
fTraceTmpFile = GetTempFilename(sPath, _
    sPrefix, _
    iUnique, _
    sTmpFile) > 0
sTmpFile = fNullTerminator(sTmpFile)
End Function

```

```

Private Function fNullTerminator(ByVal sValue As String) As String
    Dim iNullPosition As Integer

    iNullPosition = InStr(sValue, Chr$(0))
    If iNullPosition > 0 Then
        fNullTerminator = Left$(sValue, iNullPosition - 1)
    Else
        fNullTerminator = sValue
    End If
End Function

```

Tipp : 48.

'Beispiel : Transparenten Bereich unter Verwendung von MSIMG32.DLL bestimmen.

'Hinweis : Auf eine Fehlerbehandlung beim Laden der Grafik wurde verzichtet.

```

Private Declare Function TransparentBlt Lib "msimg32" _
    (ByVal hdcDest As Long, _
    ByVal nXOriginDest As Long, _
    ByVal nYOriginDest As Long, _
    ByVal nWidthDest As Long, _
    ByVal nHeightDest As Long, _
    ByVal hdcSrc As Long, _
    ByVal nXOriginSrc As Long, _
    ByVal nYOriginSrc As Long, _
    ByVal nWidthSrc As Long, _
    ByVal nHeightSrc As Long, _
    ByVal crTransparent As Long) _
    As Long

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

Private Function fMakeTransparentRegion(pbSource As PictureBox, _
    pbDestination As PictureBox)

    Dim lTransparentColor As Long

    lTransparentColor = GetPixel(pbSource.hdc, 0, 0)
    Call TransparentBlt(pbDestination.hdc, 0, 0, pbSource.ScaleWidth, _
        pbSource.ScaleHeight, _
        pbSource.hdc, 0, 0, _
        pbSource.ScaleWidth, _
        pbSource.ScaleHeight, _
        lTransparentColor)

    Picture2.Refresh
End Function

Private Sub Command1_Click()
    fMakeTransparentRegion Picture1, Picture2
End Sub

Private Sub Command2_Click()
    Unload Me
End Sub

Private Sub Command3_Click()
    Picture2.Cls
End Sub

```



```

Private Sub Form_Load()
    Picture1.Picture = LoadPicture(App.Path & "\test.bmp")

    With Picture1
        .AutoRedraw = True
        .ScaleMode = 3
    With Picture2
        .AutoRedraw = True
        .ScaleMode = 3
        .Width = Picture1.Width
        .Height = Picture1.Height
    End With
    End With
End Sub

```

Tipp : 49.

'Beispiel : Fenster-Caption ermitteln.

```

Private Declare Function GetWindowText Lib "user32" _
    Alias "GetWindowTextA" _
    (ByVal hwnd As Long, _
    ByVal lpString As String, _
    ByVal cch As Long) _
    As Long

Private Declare Function GetWindowTextLength Lib "user32" _
    Alias "GetWindowTextLengthA" _
    (ByVal hwnd As Long) _
    As Long

Private Sub Command1_Click()
    Dim lhWnd As Long
    lhWnd = Me.hwnd
    Label1 = fGetWndCaption(lhWnd)
End Sub

Private Sub Command2_Click()
    Unload Me
End Sub

Private Function fGetWndCaption(lhWnd As Long) As String
    Dim lWndCapLen As Long
    Dim sRes As String

    lWndCapLen = GetWindowTextLength(lhWnd)
    sRes = String(lWndCapLen, 0&)
    Call GetWindowText(lhWnd, sRes, lWndCapLen + 1)
    fGetWndCaption = sRes
End Function

```

Tipp : 50.

'Beispiel : Caption eingeben und Handle ermitteln.

```

Private Declare Function GetWindow Lib "user32" _
    (ByVal hwnd As Long, _
    ByVal wCmd As Long) _
    As Long

Private Declare Function SendMessage Lib "user32" _
    Alias "SendMessageA" _
    (ByVal hwnd As Long, _
    ByVal wParam As Long, _
    ByVal lParam As Any) _
    As Long

Private Declare Function GetDesktopWindow Lib "user32" () _
    As Long

Const GW_CHILD = 5
Const WM_GETTEXT = &HD

```

Const GW_HWNDNEXT = 2

Private Function fSplitString(sHandingTxt As String, Optional vTmp As Variant) As Variant
If IsMissing(vTmp) Then vTmp = ""

Dim ICount As Long
Dim IResInStr As Long
Dim sResHanding As String
ReDim vCount(0) As Variant

sResHanding = sHandingTxt
IResInStr = InStr(sResHanding, vTmp)
Do While IResInStr
If ICount > UBound(vCount) Then ReDim Preserve vCount(0 To ICount) As Variant
vCount(ICount) = Left\$(sResHanding, IResInStr - 1)
sResHanding = Mid\$(sResHanding, IResInStr + Len(vTmp))
IResInStr = InStr(sResHanding, vTmp)
ICount = ICount + 1
Loop
If Len(sResHanding) Then
If ICount > UBound(vCount) Then ReDim Preserve vCount(0 To ICount) As Variant
vCount(ICount) = sResHanding
End If
fSplitString = vCount
End Function

Private Function fGetWndHandle(tb As TextBox)

Dim vSplitRes As Variant
Dim IGetWndRes As Long
Dim sReserved As String
Dim IRes As Long
Dim n As Long
Dim blRes As Boolean

vSplitRes = fSplitString(Text2.Text, vbCrLf)

IGetWndRes = GetWindow(GetDesktopWindow(), GW_CHILD)

tb.Text = ""

Do While IGetWndRes

sReserved = Space\$(128)

IRes = SendMessage(IGetWndRes, WM_GETTEXT, 128, ByVal sReserved)

If IRes Then

blRes = False

sReserved = Left\$(sReserved, IRes)

For n = LBound(vSplitRes) To UBound(vSplitRes)

If Left\$(vSplitRes(n), 1) = "*" Then

If Right\$(vSplitRes(n), 1) = "*" Then

If InStr(sReserved, Mid\$(vSplitRes(n), 2, Len(vSplitRes(n)) - 2)) Then

blRes = True

Exit For

End If

Elseif Right\$(sReserved, Len(vSplitRes(n)) - 1) = Mid\$(vSplitRes(n), 2) Then

blRes = True

Exit For

End If

Elseif Right\$(vSplitRes(n), 1) = "*" Then

If Left\$(sReserved, Len(vSplitRes(n)) - 1) = Left\$(vSplitRes(n), Len(vSplitRes(n)) - 1) Then

blRes = True

Exit For

End If

Elseif sReserved = vSplitRes(n) Then

blRes = True

Exit For

End If

Next

If blRes Then

tb.SetStart = Len(tb.Text)

tb.SetText = "Wnd Handle : " & IGetWndRes & vbCrLf & " " & sReserved & " " & vbCrLf

End If

End If

IGetWndRes = GetWindow(IGetWndRes, GW_HWNDNEXT)

Loop

If tb.Text = "" Then

tb.Text = "Das gesuchte Fenster wurde nicht gefunden."

End If

End Function

```
Private Sub Command1_Click()  
    fGetWndHandle Text1  
End Sub
```

```
Private Sub Command2_Click()  
    Unload Me  
End  
End Sub
```