# Neuer Datensatz in einem Formular und Courser springt auf ein Feld

DoCmd.GoToRecord , "", acNewRec

 DoCmd.GoToControl "Feld"

# Öffnungsritual für Formulare

DoCmd.Maximize

DoCmd.GoToRecord , "", acNewRec

DoCmd.GoToControl "Feld"

# Ein Formular schließen und ein anderes öffnen

 Dim stDocName As String

 Dim stLinkCriteria As String

 DoCmd.Close

 stDocName = "Formular"

 DoCmd.OpenForm stDocName, , , stLinkCriteria

# Schließt ein bestimmtes Formular, ohne dass dieses aktiv sein muss

 DoCmd.Close acForm, "Formular"

# Löst eine Abfrage aus und aktualisiert die Daten

DoCmd.OpenQuery "Abfrage", acNormal, acEdit

DoCmd.Requery ""

# Löst 2 Makros aus

DoCmd.RunMacro "Makro ? ", , ""

 DoCmd.RunMacro "Makro ?", , ""

# Aktualisiert die Formulardaten

DoCmd.DoMenuItem acFormBar, acRecordsMenu, 5, , acMenuVer70

# Schließt ein Formular

DoCmd.Close

# Nachfragefenster bevor Datensatz gelöscht wird

Dim antwort

antwort = MsgBox(prompt:="Möchten Sie wirklich löschen ?", buttons:=vbYesNo, title:=" Löschen?")

If antwort = vbYes Then

DoCmd.DoMenuItem acFormBar, acEditMenu, 8, , acMenuVer70

 DoCmd.DoMenuItem acFormBar, acEditMenu, 6, , acMenuVer70

Else

 DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord, , acMenuVer70

End If

# Setzt einem Hyperlink " Mailto" voraus, um Outlook zu öffnen

Private Sub Mail\_AfterUpdate()

Dim strAnzeige

strAnzeige = HyperlinkPart(Me."Feldname mit Mailadresse", acDisplayedValue)

Me. "Feldname mit Mailadresse" = strAnzeige & "#mailto:" & strAnzeige & "#"

End Sub

# Löst einen gesetzten Filter in einem Formular, incl. Meldung

DoCmd.ShowAllRecords

MsgBox " es können wieder alle Datensätze aufgerufen werden "

# Ein externes Dokument erstellen

Erstellt auf C:\ ein Worddokument mit dem Namen Brief und schreibt in die erste Zeile "erste Überschrift

Set fs = CreateObject("Scripting.FileSystemObject")

Set a = fs.CreateTextFile("c:\Brief.doc", True)

a.WriteLine ("erste Ueberschrift.")

a.Close

Zeigt den Shell-Dialog 'Ordner auswählen' an und gibt den ausgewählten Ordner zurück. Wird der Dialog abgebrochen ist die Rückgabe ein Nullstring.

BrowseForFolder([Title As String], [StartFolder As String], [RootFolder As ShellSpecialFolderConstants], [EditBox As Boolean], [ShowFiles as Boolean], [Form As Form]) As String

Dim strPath As String
    strPath = BrowseForFolder("C:\", &H1, &H11)
    If strPath = vbNullString Then
        'canceled
        MsgBox "Dialog canceled"
    Else
        MsgBox strPath
    End If

# Erstellt auf Laufwerk C: einen Ordner mit ausgewähltem Namen, schaut ob es den schon gibt und erstellt dann ein Unterverzeichnis mit weiterem Feldnamen

If Dir("C:\" & Me!Textfeld, vbDirectory) = "" Then
        MkDir "C:\" & Me!Feldname
    End If
    MkDir "C:\" & Me!Textfeld & "\" & Me!anderer\_Feldname

Die Zeile nach End If kann wahlweise geschrieben werden, wenn ich noch einen Unterordner will. Wichtig ist alles was über End If steht

# ??? Bild in einen Bildpositionsrahmen namens picBild laden

Private Sub IDSteuergerät\_AfterUpdate()

On Error GoTo Err\_Form\_L1

 Me!picBild.Picture = "c:\A\_DatenbankSteuergeräte\Fremdmuster\" & Me!Hersteller & "\" & Me!Bild\_Pfad & "\Bilder\" & Me!Bild\_Name

 Exit Sub

Err\_Form\_L1:

On Error GoTo Err\_Form\_L2

 Me!picBild.Picture = "\\rtcl04g1\rtdfs\aeel\aeel2\wettbewerber\Steuergeräte\" & Me!Bild\_Pfad & "\Bilder\" & Me!Bild\_Name

 Exit Sub

Err\_Form\_L2:

 Exit Sub

End Sub

# Objekt ein oder ausblenden

If <Bedingung> Then

   Me!<Steuerelement1>.Visible = True

Else

   Me!<Steuerelement2>.Visible = False

End If

# Zeigt ein Feld nur wenn was drin steht

If isnull(Me.Feldname) then

Me.Feldname.Visible = false

else

Me.Feldname.Visible = true

End if

oder die Kurzfassung:

With Me.Feldname
.Visible = Not IsNull(.Value)
End With

noch eine Idee

if Bedingung then Me![Feldname].visible = False

# Öffnet einen bestimmten Ordner, dessen Pfad sich in einem Feldnamen befindet, ohne Feldname ist der statt & ... der Pfad anzugeben ( c:\Ordner\Ordner\ )

Shell "explorer.exe " & [Feldname], vbNormalFocus