

Bei den folgenden Erläuterungen wird immer davon ausgegangen, dass ein Formular standardmäßig eine Auswahlliste (z.B. Liste aller Mitglieder) enthält. Ein Klick in die Auswahlliste führt zur Anzeige des angeklickten Datensatzes (z.B. der Mitgliederdaten wie Name, Geburtstag, ... usw.) in entsprechenden Text-, Listen- und Kombinationsfeldern auf dem Formular.

Die Beispiele in der Spalte "Wie?" stammen - wenn nicht anders genannt - aus der Beispieldatenbank "Verein".

lfd. Nr.	Was?	Wie?
1	Formular öffnen	
2	Ist der in den Feldern des Formulars angezeigte Datensatz in der Auswahlliste durch eine farbige Hinterlegung markiert?	Wählen Sie gleich beim Öffnen des Formulars die erste Zeile in der Auswahlliste aus: <pre>Private Sub Form_Open(Cancel As Integer) ... If Nz(lstMannschaften.ListCount) > 0 Then Me!lstMannschaften = Me!lstMannschaften.ItemData(0) Call lstMannschaften_AfterUpdate End If</pre>
3	Wenn das Formular ein oder mehrere Unterformulare hat: Zeigen die Unterformulare die Daten an, die zu dem Datensatz gehören, der im Formular angezeigt wird? (z.B. die Liste der Aufträge des angezeigten Kunden) Wechselt der Inhalt des Unterformulare, wenn ein anderer Datensatz in der Auswahlliste angeklickt wird?	Das wird gewährleistet durch die Verknüpfung zwischen Formular und Unterformular auf dem Eigenschaftenblatt "Daten" des Unterformulars: z.B.: frmMannschaften / frmMannschaften_ufoMitglieder: Verknüpfen nach: man_id Verknüpfen von: man_id_f
4	Damit der Benutzer nicht erst ein Handbuch lesen oder herumprobieren muss, können Sie einzelne Bedienungshinweise gleich auf dem Formular unterbringen - z.B. bei Bemerkungsfeldern "Zeilenumbruch mit Strg-Enter".	Legen Sie einfach ein Bezeichnungsfeld an und schreiben Sie den Text hinein.
5	Textfelder werden von Access standardmäßig ohne vertikale Scrollbalken angelegt. Bei Bemerkungsfeldern sollten Sie diese aber einschalten.	Eigenschaftenblatt "Format" des Textfeldes: "Bildlaufleisten = Vertikal"
6	Wenn Sie mit der Maus einige Sekunden über einem Steuerelement verharren, erscheint ein sog. "Steuerelement-Tipp". Was dann dort steht, können Sie selber bestimmen.	Eigenschaftenblatt "Andere" des Steuerelements: "SteuerelementTip-Text"

lfd. Nr.	Was?	Wie?
7	Sie können den Standard-Schließen-Button (ganz rechts oben) deaktivieren und stattdessen einen eigenen Schließen-Button zu programmieren. Dann können Sie selber bestimmen, was beim Schließen des Formulars passieren soll.	<p>deaktivieren des Standard-Schließen-Buttons: auf dem Eigenschaftenblatt "Format" des Formulars "Schließen-Schaltfläche = Nein"</p> <p>eigener Schließen-Button:</p> <pre>Private Sub cmdSchliessen_Click() ' eigene Befehle, die vor dem Schließen ausgeführt werden sollen DoCmd.Close End Sub</pre>
8	Beim Klick auf den "Neu"-Button werden der "Neu"- und der "Löschen"-Button deaktiviert. Sie sollten beim Öffnen des Formulars sicherheitshalber aktiviert werden.	<pre>Private Sub Form_Open(Cancel As Integer) ... cmdLoeschen.Enabled = True cmdNeu.Enabled = True End Sub</pre>
9	An vielen Stellen bietet es sich an, mit einem Doppelclick in ein Textfeld zu einem anderen Formular zu wechseln - z.B. Feld "Kunde" im Auftragsformular: Ein Doppelclick bringt Sie ins Kundenformular.	<p>Schreiben Sie für das Ereignis "Doppelclick" eine solche Prozedur:</p> <pre>Private Sub txtName_DblClick(Cancel As Integer) DoCmd.OpenForm "frmMitglieder", , , , , Me!mit_id End Sub</pre> <p>Im aufgerufenen Formular (im Beispiel: frmMitglieder) müssen Sie dann dafür sorgen, dass der angeklickte Datensatz dort auch angezeigt wird:</p> <pre>Private Sub Form_Open(Cancel As Integer) If IsNull(OpenArgs) Then ' Anzeige der ersten Zeile der Auswahlliste (siehe oben) Else Me!lstMitglieder = OpenArgs Call lstMitglieder_AfterUpdate End If End Sub</pre>
10	Klick in die Auswahlliste	

lfd. Nr.	Was?	Wie?
11	Wird nach einem Klick in die Auswahlliste in den Text-, Listen- und Kombinationsfeldern des Formulars der richtige Datensatz angezeigt?	<p>Schreiben Sie für das Ereignis "AfterUpdate" eine solche Prozedur:</p> <pre>Private Sub lstMitglieder_AfterUpdate() If Not IsNull(lstMitglieder) Then _ Me.Recordset.FindFirst "mit_id=" & Me!lstMitglieder End Sub</pre> <p>In Access 2003 wird eine ähnliche VBA-Prozedur automatisch durch den Listefeld-Assistenten erzeugt. Ab Access 2007 müssen Sie das automatisch erzeugte "Eingebettete Makro" durch diese Prozedur ersetzen!</p>
12	Werden nach einem Klick in die Auswahlliste nicht nur im Hauptformular, sondern auch <u>in den Unterformularen</u> die richtigen Daten angezeigt?	<p>Das wird gewährleistet durch die Verknüpfung zwischen Formular und Unterformular auf dem Eigenschaftenblatt "Daten" des Unterformulars:</p> <p>z.B.: frmMannschaften / frmMannschaften_ufoMitglieder</p> <p>Verknüpfen nach: man_id Verknüpfen von: man_id_f</p>
13	Nach dem Klick auf "Neuer Datensatz" sollten der Neu- und der Löschbutton deaktiviert werden. Werden sie nach dem Klick in die Auswahlliste wieder aktiviert?	<pre>Private Sub lstMitglieder_AfterUpdate() ... cmdLoeschen.Enabled = True cmdNeu.Enabled = True ...</pre>
14 Datensatz bearbeiten		
15	Sie können sich mit der Tabulatortaste im Formular von Steuerelement zu Steuerelement bewegen, ohne die Maus benutzen zu müssen. Werden die Steuerelemente (i.a. Textfelder) dabei in der richtigen (und logischen!) Reihenfolge angesprungen?	<p>Zum Einstellen der Tabulatorreihenfolge klicken Sie in der Entwurfsansicht ein beliebiges Steuerelement an und öffnen das Eigenschaftenblatt "Andere". Dort können Sie unter "In Reihenfolge" einstellen, ob das Steuerelement überhaupt angesprungen werden soll. Wenn Sie in das Feld rechts neben "Reihenfolgeposition" klicken, erscheint ein Button mit drei Punkten, der das Fenster öffnet, mit dem Sie die Reihenfolgepositionen aller Steuerlemente einstellen können.</p> <p>Außerdem sollten Sie unter "Andere / Zyklus" einstellen: "Aktueller Datensatz". Tut man das nicht, so passiert folgendes: Wenn Sie im letzten Eingabefeld des Formulars stehen und dann auf die Tabulatortaste drücken, springt die Einfügemarke nicht (wie erwartet) wieder zurück zum ersten Eingabefeld, sondern zum nächsten (noch leeren!) Datensatz, d.h. zur Verblüffung des Benutzers sind plötzlich alle Eingabefelder leer!</p>

lfd. Nr.	Was?	Wie?
16	Manchmal ist es sinnvoll, Felder oder Listen zu sperren, so dass sie nicht mehr bearbeitet werden können (z.B. darf nicht mehr in die Auswahlliste geklickt werden, wenn der Benutzer auf "Neu" geklickt hat). Sind die richtigen Felder gesperrt und können sie wirklich nicht bearbeitet werden?	z.B.: <pre>Private Sub cmdNeu_Click() ... lstMitglieder.Enabled = False</pre>
17	Für die Dateneingabe benutzt man häufig auch Kombinationsfelder, die eine Liste von möglichen Eingabedaten anzeigen (z.B. Namen der Trainer für eine Mannschaft). Zeigen die Kombinationsfelder des Formulars die richtigen Daten an?	Einstellung der Datenquelle des Kombinationsfeldes (1) entweder über das Eigenschaftenblatt "Daten": "Datensatzherkunft" (2) oder per VBA-Code - z.B. in frmTraining: <pre>Private Sub cboPlatzFilter_AfterUpdate() strSQL = "SELECT ... " Me!lstTrainings.RowSourceType = "Table/Query" Me!lstTrainings.RowSource = strSQL</pre>
18	Manchmal möchte man den Benutzer zwingen, Daten in einer bestimmten Reihenfolge einzugeben, indem man z.B. ein Textfeld oder eine Liste erst aktiviert, nachdem in einem anderen Textfeld Daten eingegeben wurden. Erfolgt die Aktivierung der Steuerelemente in der richtigen Reihenfolge?	z.B.: <pre>Private Sub cmdSpeichern_Click() ... lstMitglieder.Enabled = True</pre>
19	Sie können ein Kombinationsfeld doppelt nutzen: Zur Anzeige und zur Eingabe von Daten (siehe Stichwort "Einzelzuordnung" im Buch). Dabei kann es schnell mal passieren, dass man auf dem Feld herumklickt und die Auswahl unbeabsichtigt ändert. Das muss verhindert werden.	Programmieren Sie die folgende Sicherheitsabfrage (Beispiel aus frmMitglieder): <pre>Private Sub cboTrainer_Dirty(Cancel As Integer) If IsNull(cboTrainer.OldValue) Then Exit Sub If MsgBox("Wollen Sie den Trainer wirklich ändern?", _ vbYesNo + vbDefaultButton2) = vbNo Then Cancel = True ' Abbruch der Bearbeitung SendKeys ("{ESC}") ' Schließen der Combobox End If End Sub</pre> <p>"Dirty" ist die engl. Bezeichnung für das Ereignis "Bei Geändert". Das "If IsNull ..." dient dazu, die Nachfrage zu verhindern, wenn es sich um die erste Auswahl eines Wertes handelt (OldValue = Null). Nur wenn im Kombinationsfeld schon ein Wert stand, der jetzt geändert werden soll, kommt die Sicherheitsabfrage.</p>

lfd. Nr.	Was?	Wie?
20	Der Benutzer sollte in einem Kombinationsfeld auf möglichst einfache Art auch <u>nichts</u> auswählen können!	Ersetzen Sie die Beschriftung des Kombinationsfeldes durch einen Button, der das Kombinationsfeld leert. Ein entsprechendes Beispiel finden Sie in der Prozedur <code>Private Sub cmdTrainer_Click() in frmMitglieder.</code>
21	Neuen Datensatz einfügen	
22	Sind nach dem Druck auf den "Neu"-Button alle Eingabe-Textfelder leer, so dass neue Daten eingegeben werden können?	In gebundenen Feldern erfolgt das automatisch; bei ungebundenen Feldern müssen Sie selber dafür sorgen: <code>Private Sub cmdNeu_Click() ... txtMitgliedsnummer = ""</code>
23	Sind auch die Kombinationsfelder geleert, d.h. zeigen sie nicht evtl. noch den Wert des vorher angezeigten Datensatzes an?	<code>Private Sub cmdNeu_Click() ... cboTrainer = Null</code>
24	Sind nach dem Klick auf den "Neu"-Button auch alle Eingabe-Textfelder <u>in den Unterformularen</u> leer?	In gebundenen Feldern erfolgt das automatisch; bei ungebundenen Feldern müssen Sie selber dafür sorgen: z.B.: <code>Forms!frmMannschaften!frmMannschaften_ufoMitglieder!txtMitname = ""</code>
25	Nach dem Klick auf den "Neu"-Button darf in der Auswahlliste keine Zeile markiert (=farbig hinterlegt) sein, denn es soll ja ein neuer Datensatz eingegeben werden, der in der Auswahlliste noch gar nicht existiert.	z.B.: <code>Private Sub cmdNeu_Click() ... lstMitglieder = Null</code>
26	Wenn der Benutzer kundgetan hat, dass er einen neuen Datensatz einfügen will, dann darf er sofort danach weder den Neu- noch den Löschenbutton betätigen. Darum sollten diese jetzt deaktiviert werden!	z.B.: <code>Private Sub cmdNeu_Click() ... cmdNeu.Enabled = False cmdLoeschen.Enabled = False</code>
27	Der Benutzer sollte sofort nach dem Klick auf den "Neu"-Button mit der Dateneingabe beginnen können. Dafür muss die Einfügemarke im ersten Mussfeld stehen.	z.B.: <code>Private Sub cmdNeu_Click() ... txtMitname.SetFocus</code>

lfd. Nr.	Was?	Wie?
28	Manchmal muss ein Datensatz erst gespeichert werden, bevor weitere Daten eingegeben werden können (z.B. muss ein neues Vereinsmitglied erst gespeichert sein, bevor ihm per Kombinationsfeld ein Trainer zugewiesen werden kann). Sind die entspr. Text- bzw. Kombinationsfelder deaktiviert?	z.B.: <pre>Private Sub cmdNeu_Click() ... cboTrainer.Enabled = False</pre>
29	Wenn der Benutzer kundgetan hat, dass er einen neuen Datensatz einfügen will, dann darf er nicht sofort danach in die Auswahlliste klicken. Darum sollte diese jetzt deaktiviert sein!	z.B.: <pre>Private Sub cmdNeu_Click() ... lstMitglieder.Enabled = False</pre>
30	Datensatz explizit speichern	
31	Wenn Muss-Daten fehlen (z.B. Name des Mitgliedes), darf der Datensatz nicht gespeichert werden.	z.B.: <pre>Private Sub cmdSpeichern_Click() ... If Nz(txtName) = "" Then MsgBox "Bitte geben Sie einen Namen ein !" txtName.SetFocus Exit Sub End If</pre>
32	Der Benutzer sollte in für ihn verständlicher Form darüber informiert werden, WELCHE Muss-Daten fehlen.	Das erfolgt mit Hilfe der MsgBox (s.o.)
33	Da die Auswahlliste beim Klick auf den "Neu"-Button deaktiviert wird, muss sie jetzt wieder aktiviert werden.	z.B.: <pre>Private Sub cmdSpeichern_Click() ... DoCmd.RunCommand acCmdSaveRecord lstMitglieder.Enabled = True lstMitglieder.Requery ...</pre>

lfd. Nr.	Was?	Wie?
34	Nach erfolgreicher Speicherung eines Datensatzes, muss die entspr. Zeile in der Auswahlliste markiert (=farbig hinterlegt) sein. Für die Änderung eines bereits existierenden Datensatzes ist das selbstverständlich - es muss aber auch nach der Speicherung eines neuen Datensatzes funktionieren!	<pre>Private Sub cmdSpeichern_Click() Dim lngMitId As Long ... DoCmd.RunCommand (acCmdSaveRecord) lngMitId = Me!mit_id ' Speichern des Primärschlüssels lstMitarbeiter.Requery ' Refresh der Auswahlliste lstMitarbeiter.SetFocus lstMitarbeiter = lngMitId ' Anzeige des gespeicherten Datensatzes Call lstMitarbeiter_AfterUpdate ' Click-Simulation</pre> <p>Der Aufruf der Prozedur lstMitarbeiter_AfterUpdate simuliert einen Klick in die Auswahlliste. Dadurch werden alle dort programmierten Befehle ausgeführt.</p>
35	Manchmal wirkt sich das Einfügen neuer Daten bzw. die Änderung vorhandener Daten auf andere Formulare aus - z.B. muss ein im Formular "Plätze" neu eingefügter Platztyp im Kombinationsfeld für die Platzauswahl im Formular "Training" erscheinen.	<p>Das ist ganz einfach: Immer wenn das Steuerelement (im Beispiel ein Kombinationsfeld) angeklickt wird ("den Focus erhält"), wird seine Datenquelle aktualisiert:</p> <p>z.B.:</p> <pre>Private Sub cboPlatz_GotFocus() cboPlatz.Requery End Sub</pre>
36	Manchmal muss ein Datensatz erst gespeichert werden, bevor weitere Daten eingegeben werden können (z.B. muss ein neues Vereinsmitglied erst gespeichert sein, bevor ihm per Kombinationsfeld ein Trainer zugewiesen werden kann). Sind die entspr. Text- bzw. Kombinationsfelder für die Eingabe weiterer Daten jetzt aktiviert?	<p>z.B.:</p> <pre>Private Sub cmdSpeichern_Click() ... cboTrainer.Enabled = True</pre>
37	Bei sehr großen Datenmengen (oder langsamen Rechnerkrücken ;-) kann die Speicherung von Daten schon mal eine Weile dauern. Dass sollte durch die Anzeige der Sanduhr angezeigt werden, damit der Benutzer sieht, dass der Rechner noch aktiv ist!	<p>z.B.:</p> <pre>Private Sub cmdSpeichern_Click() ... DoCmd.Hourglass True DoCmd.RunCommand acCmdSaveRecord DoCmd.Hourglass False</pre>
38	Datensatz implizit speichern	

lfd. Nr.	Was?	Wie?
39	Leider kann der Benutzer Daten nicht nur explizit durch Klick auf den entspr. Button speichern, sondern auch implizit - z.B. durch Klick in die Auswahlliste oder durch Schließen des Formulars unmittelbar nach der Dateneingabe. Wurde das verhindert? (gelingt leider nicht immer komplett - man muss da auch ein wenig an die Intelligenz des Benutzers appellieren;-)	implizites Speichern durch (1) Klick in die Auswahlliste: Auswahlliste deaktivieren (2) Klick auf "Speichern" oder "Löschen": Buttons deaktivieren (3) Schließen des Formulars: Eigenschaftenblatt "Format" des Formulars - "Schließen Schaltfläche" = "Nein" und eigenen Schließen-Button programmieren, in dessen VBA-Code das ordnungsgemäße Speichern selber organisiert wird
40	Wenn dann schon implizit gespeichert wurde - sind dann wenigstens die richtigen Daten gespeichert worden?	Formular schließen, erneut öffnen und Daten kontrollieren; wenn nicht, dann implizites Speichern verhindern (s.o.)
41	Ganz schlimm wird es, wenn der Nutzer erst auf den Button "Neuer Datensatz" klickt, dann aber nichts eingibt und implizit speichert (z.B. indem er in die Auswahlliste klickt oder das Formular einfach schließt). Dann kann es passieren, dass "Geisterdaten" gespeichert werden. Sie erkennen sie, wenn Sie das Formular schließen und erneut öffnen. Dann erscheint in der Auswahlliste eine Leerzeile. Das muss unbedingt verhindert werden.	Sie können diesen unangenehmen Effekt am besten verhindern, indem Sie mindestens eines der Datenfelder zum Muss-Feld machen - und zwar schon ganz früh beim Tabellenentwurf ("Eingabe erforderlich = ja"). Wenn Sie es dort vergessen haben, können Sie es jetzt immer noch nachholen. Evtl. kommt dann aber eine Warnung, weil in einigen Zeilen der Tabelle das Feld, das Sie jetzt zum Muss-Feld machen wollen, noch leer ist. Beispiel: Der Mitgliedsname (mit_name) war bisher kein Mussfeld. Dadurch erscheinen nach dem impliziten Speichern "Geistemitglieder" ohne Namen in der Mitgliederliste. Sie wollen also nachträglich mit_name zum Muss-Feld machen - das gelingt aber nicht, weil Sie bereits einige Mitglieder ohne Namen gespeichert haben. Diese müssen Sie entweder vorher in der Tabelle löschen oder Sie müssen ihnen in der Tabelle (nicht im Formular) einen Namen geben. Erst wenn ALLE Mitglieder einen Namen haben, können Sie mit_name zum Muss-Feld machen. Anschließend wird im Formular nicht mehr implizit gespeichert, weil Muss-Daten fehlen!
42	Datensatz löschen	

lfd. Nr.	Was?	Wie?
43	Access zeigt auch automatisch einen Warnhinweis an, bevor gelöscht wird. Dieser Hinweis liest sich aber etwas unverständlich. Darum sollten Sie selber einen Hinweis programmieren!	<p>Sie haben wieder mindestens zwei Möglichkeiten:</p> <p>(1) Sie schreiben die Sicherheitsabfrage in die Form_Delete-Prozedur, die VOR der Löschung ausgeführt wird. Indem Sie den darin enthaltenen Parameter <code>Cancel</code> auf <code>True</code> setzen, können Sie die Löschung noch im letzten Augenblick verhindern:</p> <pre>Private Sub Form_Delete(Cancel As Integer) If MsgBox("Wollen Sie das Mitglied wirklich löschen?", _ vbYesNo + vbDefaultButton2) = vbNo Then Cancel = True End Sub</pre> <p>(2) Sie erstellen einen eigenen Löschbutton mit einer Sicherheitsabfrage :</p> <pre>Private Sub cmdLoeschen_Click() ... If MsgBox("Wollen Sie das Mitglied wirklich löschen?", _ vbYesNo + vbDefaultButton2) = vbNo Then Exit Sub ...</pre>
44	Häufig hat die Löschung von Daten zu Folge, dass dann auch weitere Daten gelöscht werden (z.B. werden mit der Löschung eines Kunden auch seine Aufträge gelöscht). Der Benutzer sollte darauf hingewiesen werden und sich ggf. gegen die Löschung entscheiden können.	<p>z.B.:</p> <pre>Private Sub Form_Delete(Cancel As Integer) If MsgBox("Wollen Sie den Platz wirklich löschen?" & _ vbCrLf & vbCrLf & "ACHTUNG:" & vbCrLf & _ "Dann werden auch alle Trainings auf diesem Platz gelöscht!", _ vbYesNo + vbDefaultButton2) = vbNo Then Cancel = True End Sub</pre>
45	Der Standardbutton (also derjenige der durch die Enter-Taste betätigt wird) bei der Frage "Wollen Sie wirklich löschen?" sollte der "Nein"-Button sein.	schreiben Sie im MsgBox-Befehl " <code>vbYesNo + vbDefaultButton2</code> " statt nur " <code>vbYesNo</code> "!
46	Evtl. wirkt sich die Löschung von Daten auch auf die Anzeige von Daten in anderen Formularen aus - z.B. muss ein im Formular "Plätze" gelöschter Platztyp auch im Kombinationsfeld für die Platzauswahl im Formular "Training" verschwinden.	<p>Das passiert automatisch durch einen Requery-Befehl in der GotFocus-Prozedur des Kombinationsfeldes:</p> <p>z.B.:</p> <pre>Private Sub cboPlatz_GotFocus() cboPlatz.Requery End Sub</pre>

lfd. Nr.	Was?	Wie?
47	In bestimmten Fällen darf die Wahl "Löschen ja/nein?" nicht dem Benutzer überlassen werden (z.B. darf kein Kunde gelöscht werden, wenn es von ihm noch unbezahlte Rechnungen gibt). In den Fall muss der Benutzer entspr. informiert werden ("Kann nicht gelöscht werden, weil ...")	<p>Indem Sie zählen, wie oft der Primärschlüssel ("_id") des zu löschenden Datensatzes noch als Fremdschlüssel ("_id_f") in ANDEREN Tabellen vorkommt, können Sie herauskriegen, ob der Datensatz gelöscht werden darf:</p> <p>z.B.:</p> <pre>Private Sub cmdLoeschen_Click(Cancel As Integer) ... If DCount("mit_id","tblMitglied","mtyp_id_f=" & Str(Me!mtyp_id)) > 0 Then MsgBox "Dieser Mitgliedstyp kann nicht gelöscht werden," & _ vbCrLf & "weil es noch Mitglieder dieses Typs gibt!" Cancel = True End If End Sub</pre>
48	In Formularen vom Typ "Endlosformular" brauchen Sie nicht unbedingt einen eigenen Löschbutton zu programmieren. Der Benutzer kann auch auf den Datensatzmarkierer am linken Rand klicken und dann die Lösch taste drücken.	<p>(1) Programmierung einer eigenen Sicherheitsabfrage:</p> <pre>Private Sub Form_Delete(Cancel As Integer) If MsgBox("Wollen Sie den Platztyp wirklich löschen?" , _ vbYesNo + vbDefaultButton2) = vbNo Then Cancel = True ... End Sub</pre> <p>(2) Unterdrückung der Standardwarnung "Sie sind dabei, 1 Datensätze zu löschen ...":</p> <pre>Private Sub Form_BeforeDelConfirm(Cancel As Integer, Response As Integer) Response = acDataErrContinue End Sub</pre>
49	Nachdem ein Datensatz gelöscht wurde soll der erste Datensatz der Auswahlliste angezeigt werden. Er muss auch in der Auswahlliste markiert sein.	<p>z.B.:</p> <pre>Private Sub cmdLoeschen_Click() ... lstWettkaempfe.Requery lstWettkaempfe.SetFocus If Nz(lstWettkaempfe.ListCount) > 0 Then ' Anzeige der ersten Zeile Me!lstWettkaempfe = Me!lstWettkaempfe.ItemData(0) Call lstWettkaempfe_AfterUpdate End If</pre>