




1. Formulare

Formulare dienen der Interaktion des Benutzers; sie sind die **Benutzerschnittstelle**. In Formularen werden Daten eingegeben und ausgewertet. Ein Benutzer sollte möglichst nie Tabellen oder Abfragen zu sehen bekommen, sondern immer nur Formulare.

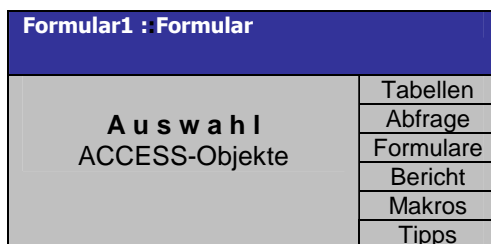
Tabellen müssen aus Sicht einer effektiven Datenhaltung streng systematisch geplant werden. Im Gegensatz dazu sollten **Formulare** einen Benutzer einfach und intuitiv leiten.

Obwohl man natürlich schon vorher wissen sollte, wofür man ein Formular entwirft, ist das Erstellen eines Formulars eine Sache, bei der man viel gestalterisch ausprobieren kann. Ein wesentlicher Unterschied zum Entwerfen von Tabellen!

2. Anlegen eines Formulars

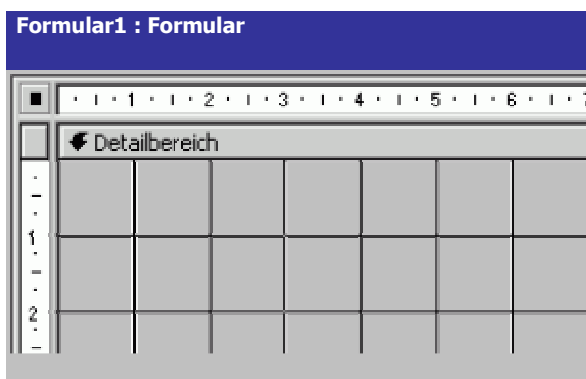
Wie alle Datenbankobjekte, werden auch neue Formulare mit  **Neu** angelegt. Wie üblich, machen wir das über die Entwurfsansicht. Bei vorhandenen Formularen kommt man mit  **Entwurf** in die Entwurfsansicht und mit  **Öffnen** in die Formularansicht.

Beim Anlegen eines neuen Formulars kann man wählen, aus welcher Tabelle oder Abfrage die Daten für das Formular stammen sollen. Diese Datenherkunft kann man später noch ändern. In der Praxis bekommt meistens jedes Formular eine eigene **Abfrage** als Datenherkunft. Formulare mit einer **Tabelle** als Datenherkunft sind eher selten: Beim Erstellen eines Formulars zeigt sich oft, dass das Formular Kriterien benötigt, die am besten mit einer Abfrage verwirklicht werden.



Ein Formular kann auch ohne Datenherkunft auskommen - ein so genanntes **ungebundenes Formular**. Typisches Beispiel ist eine Benutzersteuerung, in der der User wählt, was er machen will, z.B. andere Formulare öffnen.

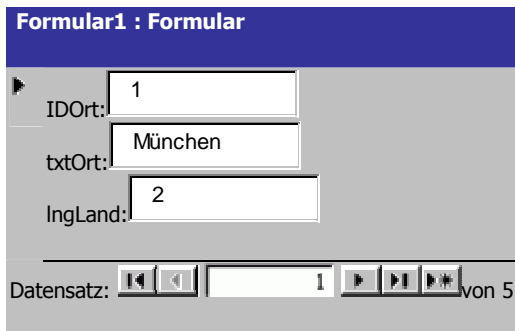
Wie bei **Visual Basic** (vgl. *VBA = Visual Basic for Applications*) bekommt man im Formularentwurf zunächst ein leeres Formular zu sehen, das nur ein paar Hilfslinien und Lineale enthält, anhand derer man die einzufügenden Steuerelemente (s.u.) graphisch ausrichten bzw. punktgenau anordnen kann.





Bei einem gebundenen Formular (also ein Formular mit Datenherkunft) kann man mittels **Feldliste** die Felder der Datenherkunft einblenden. Das Aussehen der Feldliste ist bereits von den Beziehungen und Abfragen he bekannt.

Zieht man mit der Maus einen oder mehrere Feldnamen aus der Feldliste auf den Formularentwurf, werden die Felder dort als Textfelder angezeigt. Schaltet man nun von der Entwurfs- in die **Formularansicht**, können die Daten der zugrunde liegenden Tabellen geändert oder neue Datensätze erfasst werden.



3. Steuerelemente

ACCESS-Steuerelemente entsprechen im Wesentlichen denen der Skriptsprache **Visual Basic**:

Formulare enthalten meist verschiedene Arten von Steuerelementtypen. Diese wählt man mit der **Toolbox** aus. Klickt man in der Toolbox einen Steuerelementtyp an und zieht danach ein Feld aus der Feldliste ins Formular, wird dieser Steuerelementtyp in das Formular integriert. Hier die wichtigsten Steuerelementtypen:

Textfeld

In ein Textfeld können beliebige Texte und Zahlen geschrieben werden.

Kontrollkästchen

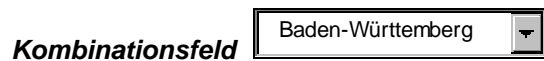
Kontrollkästchen können per Mausclick mit einem Häkchen versehen bzw. das Häkchen wieder entfernt werden. Gedacht sind sie für Ja/Nein-Werte (Boolean). Das ist aber nicht zwingend so. Liegt einem Kontrollkästchen z.B. ein Feld vom Typ 'Zahl' zugrunde, so wird der Feldwert auf '0' gesetzt, wenn das Kästchen geleert, und auf '-1', wenn ein Häkchen gesetzt wird. '0' steht also für 'Nein' und '-1' für 'Ja'.

Optionsfeld, Optionsgruppe

z.B.: **Anrede**

- Herr**
- Frau
- Firma

Optionsfelder werden in Optionsgruppen eingesetzt. Eine Optionsgruppe ist ein Rahmen, der die darin befindlichen Optionsfelder als zusammengehörig definiert. Außerdem ist die Optionsgruppe mit dem zugrunde liegenden Datenfeld verknüpft. Optionsfelder sind Knöpfe, aus denen ein Benutzer immer nur genau einen auswählen kann. Der Wert des aktivierten Optionsfelds wird in das der Optionsgruppe zugeordnete Datenfeld geschrieben.



Kombinationsfelder zeigen Werte aus einer frei definierbaren Liste oder einer Tabelle/Abfrage an. Der Benutzer kann einen davon auswählen und ein damit in Zusammenhang stehender Wert wird eingetragen.

Bezeichnungsfeld


Bezeichnungsfelder enthalten freien Text zur Information des Benutzers. Man kann beliebige Bezeichnungsfelder einfügen.

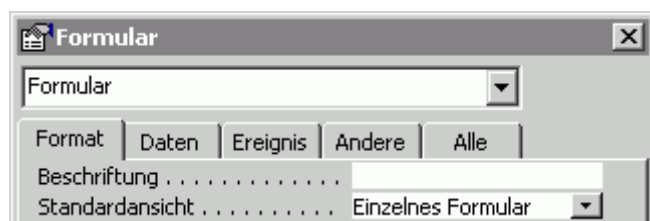
Außerdem wird beim Einfügen anderer Steuerelemente stets auch ein Bezeichnungsfeld mit eingefügt, das den Namen des zugrunde liegenden Datenfeldes anzeigt. Diese Bezeichnungsfelder sind dann immer automatisch mit dem Steuerelement assoziiert, mit dem sie eingefügt wurden. Klickt der Benutzer auf ein assoziiertes Bezeichnungsfeld, erhält das zugehörige Steuerelement den Focus. Dies ähnelt einem `<label>`-Tag in HTML.

Meistens sollte man assoziierte Bezeichnungsfelder noch umbenennen: Fügt man in der Entwurfsansicht in ein Bezeichnungsfeld ein '&' ein, wird das darauf folgende Zeichen des Bezeichnungsfeldes in der Formularansicht unterstrichen dargestellt. Damit bekommt das assoziierte Steuerelement ein entsprechendes Tastaturkürzel und kann mit `[ALT]-[Taste]` angesprochen werden.

4. Eigenschaften

Auch so genannte ACCESS-„Eigenschaften“ entsprechen im Wesentlichen denen von Visual Basic:

Alle Steuerelemente - und auch das Formular selbst - können über das  **Eigenschaftenfenster** angepasst werden. Natürlich sind die Eigenschaften nicht für alle Objekte gleich, aber sie sind sich doch recht ähnlich. Zur besseren Orientierung ist das Eigenschaftenfenster in Karteiregister unterteilt.



Unter **Format** finden sich Eigenschaften, die das Aussehen des Elements bestimmen, z.B. Text- und Hintergrundfarbe oder den Rahmen.

Unter **Daten** wird insbesondere die Datenherkunft des Objekts festgelegt.

Ereignis ist im Zusammenhang mit Automatisierung (Visual Basic) interessant. Es kann z.B. bestimmt werden, dass beim Klicken oder beim Verlassen eines Elements bestimmte Aktionen passieren, wie z.B. das Ausdrucken eines Berichts.

Unter **Andere** finden sich weitere Eigenschaften. Wichtig ist z.B. der Name eines Steuerelements, aber auch die Reihenfolge der Steuerelemente wird hier festgelegt. Wem es optisch mehr zusagt, findet unter **Alle** alle Eigenschaften zugleich.

5. Beispielformular

Als Beispiel soll wieder die Erfassung von Orten samt Bundesländern. Es bestehen also bereits zwei Tabellen:



Mit **Neu** erstellen wir ein neues Formular und wählen `tabOrte` als Datenherkunft. In den Formulareigenschaften (Register 'Daten', Eigenschaft 'Datenherkunft') kann man jede vorhandene Tabelle oder Abfrage als Datenquelle auswählen. Man könnte sogar eine neue Abfrage kreieren, die nicht als Abfrage, sondern im Formular selbst gespeichert wird. Davon ist aber abzuraten! So eine Abfrage ist stets langsamer als eine Abfrage, die auch als Abfrage gespeichert ist.

In der Entwurfsansicht sollten **Feldliste**, **Eigenschaftenfenster** und **Toolbox** eingeschaltet sein. Der **Steuerelement-Assistent** in der Toolbox sollte ausgeschaltet sein. Das ist zwar u.U. eine nützliche Hilfe, aber für den Lerneffekt kontraproduktiv.

Zuerst ziehen wir `txtOrt` aus der Feldliste in den Formularentwurf. Wir sehen dort jetzt ein Textfeld, in dem `txtOrt` steht, und davor ein Bezeichnungsfeld, das mit `txtOrt:` beschriftet ist. Die Beschriftung ändern wir jetzt. Dazu klicken wir auf das Bezeichnungsfeld, um es auszuwählen, und wählen im Eigenschaftenfenster das Register 'Format'. Dort findet sich die Eigenschaft 'Beschriftung', die wir von `txtOrt:` auf `&Ort:` (mit einem '&' am Anfang) ändern.

Kleine Anmerkung: Im Tabellenentwurf von `tabOrte` hätten wir im unteren Bereich des Feldes `txtOrt` unter 'Beschriftung' schon `&Ort:` eingeben können. Dann wäre beim Einfügen des Feldes ins Formular gleich diese Bezeichnung eingefügt worden. Ob man das nutzt, ist Geschmackssache.

Nun klicken wir in der Toolbox auf **Kombinationsfeld**, und ziehen aus der Feldliste `lngLand` ins Formular. Auch hier ändern wir die Bezeichnung, nämlich auf `&Land:`. Außerdem ändern wir den Namen des Kombinationsfeldes im Register 'Andere' auf `cboLand`. Dann legen wir fest, was das Kombinationsfeld eigentlich anzeigen soll. Dazu wählen wir im Eigenschaftenfenster das Register 'Daten'. Der Steuerelementinhalt `lngLand` stimmt ja schon (das haben wir schließlich ins Formular gezogen), die Eigenschaft 'Herkunftstyp' muss auf `Tabelle/Abfrage` stehen, die Datensatzherkunft auf `tabLaender`, und 'gebundene Spalte' auf 1. Jetzt schalten wir probeweise in die Formularansicht.

Jetzt ist der Buchstabe 'O' im Wort 'Ort' und das 'L' im Wort 'Land' unterstrichen. Wir können also mit den Tastenkombinationen `[ALT]-O` bzw. `[ALT]-L` zwischen den beiden Steuerelementen hin- und herspringen.

Für das Kombinationsfeld `cboLand` stehen die Zahlen von 1 bis 16 zur Verfügung. Das sind die Nummern der Bundesländer aus `tabLaender`. Das ist natürlich unbefriedigend; man erwartet ja deren Namen. Daher gehen wir wieder in die Entwurfsansicht, wählen das Kombinationsfeld aus und im Eigenschaftenfenster das Register 'Format'. Dort stellen wir 'Spaltenanzahl' auf 2 und geben in 'Spaltenbreiten' `1cm; 3cm` ein. Außerdem sollte 'Breite' auf mindestens `4cm` stehen. Dann gehen wir zurück in die Formularansicht.

Auf den ersten Blick hat sich nichts geändert. Öffnen wir jetzt aber das Kombinationsfeld, sehen wir neben den Nummern der Bundesländer auch deren Namen. Wir haben nämlich festgelegt, dass das Kombinationsfeld die ersten zwei Spalten der Tabelle `tabLaender` anzeigen soll. Gebunden ist die Erste (es wird also der Wert der ersten Spalte eingetragen).

Meistens ist es sogar sinnvoll, die Eigenschaft 'Spaltenbreiten' auf Werte wie `0cm; 3cm` zu stellen. So wird die Nummer ausgeblendet, zu sehen ist nur noch der Name des Bundeslandes.

Was das Formular nicht anzeigt, ist der ID des Ortes. Aber muss ein Benutzer den ID wirklich sehen? Das Feld ist vom Typ Autowert und füllt sich selbst. Normalerweise kann auf derartige Angaben in Formularen getrost verzichtet werden. Unser Formular ist also fertig, und wir speichern es unter `frmOrte` ab.

Das soll aber niemanden davon abhalten, noch etwas mit dem Formular zu spielen. Bei Formularen führen meistens viele Wege zum Ziel. So kann man z.B. testen, ob man für die Bundesländer eine Optionsgruppe statt eines Kombinationsfeldes nutzen möchte.

6. Endlosformular

Mit dem oben erstellten Formular `frmOrte` beschäftigen wir uns noch eingehender: Das Formular hat nur zwei Steuerelemente. In der Praxis werden darüber wohl ziemlich viele Datensätze eingegeben, von denen man aber immer nur einen sieht. Es wäre übersichtlicher, eine Liste zu haben.

Kein Problem für Access: Zuerst verschieben wir die beiden Steuerelemente so, dass sie nebeneinander liegen. Beide sollten ziemlich weit oben, knapp unter dem Wort 'Detailbereich' liegen. Die Größe des Detailbereichs selbst kann man ebenfalls mit der Maus anpassen, wir sorgen für eine geringe Höhe, z.B. 1cm oder weniger. Dann wählen wir im Eigenschaftsfenster das Formular aus und stellen im Register 'Format' die Eigenschaft 'Standardansicht' auf `Endlosformular`. Zurück in der Formularansicht haben wir unser Ziel schon erreicht!

Es ist unnötig, in jedem Datensatz die Beschriftungen 'Ort' und 'Land' zu sehen. Deswegen löschen wir in der Entwurfsansicht die Bezeichnungsfelder und wählen im Menü `Ansicht - Formulkopf/-fuß`. Die Entwurfsansicht ist jetzt unterteilt in `Formularkopf`, `Detailbereich`, `Formularfuß`. Im `Formularkopf` fügen wir zwei Bezeichnungsfelder ein, von denen wir eines mit `Ort` beschriften und über dem Textfeld `txtOrt` im `Detailbereich` ausrichten. Ein weiteres Bezeichnungsfeld

beschriften wir mit `Land` und richten es über `cboLand` aus. Der Formularfuß erhält abschließend noch eine Höhe von 0. Nun haben wir eine 'Überschriftenzeile', die auch beim Scrollen stehen bleibt.

7. Unterformular

Das Endlosformular `frmOrte` zeigt nun alle Orte aus `tabOrte` an. Es wäre aber praktisch, immer man nur die Orte eines bestimmten Bundeslandes zu sehen. Dazu erstellen wir ein neues Formular, das wir `frmLaender` nennen und das auf `tabLaender` zugreift. Für unser Beispiel genügt es, wenn ein einfaches Textfeld darin enthalten ist, das `txtLand` anzeigt.

Nun fügen wir aus der Toolbox ein  **Unterformular** ein. Diesem neuen Steuerelement weisen wir im Eigenschaftsfenster als Herkunftsobjekt (in der Registerkarte 'Daten') unser Endlosformular `frmOrte` zu. Und falls das nicht automatisch passieren sollte, stellen wir die Eigenschaften 'Verknüpfen von' sowie 'Verknüpfen nach' auf `lngLand` bzw. `IDLand`.

Damit wird in `frmLaender` in einem 'Fensterchen' `frmOrte` angezeigt - aber nur noch mit den Orten, die im gerade angezeigten Bundesland liegen (Man kann also in `frmOrte` auch die Anzeige der Bundesländer löschen). Geht man jetzt im Hauptformular zu einem anderen Datensatz, werden auch im Unterformular die dazu passenden Datensätze angezeigt.

6. Tipps & Tricks zu Formularen

Beim Entwerfen von Formularen sollten folgende drei Gesichtspunkte beachtet werden::

1. **Intuition des Benutzers**

Was erwartet ein Benutzer vom Formular, und wie kommt er mit den Elementen des Formulars zurecht? Technisch sind zahllose Möglichkeiten vorhanden, und es lassen sich faszinierende Lösungen im Internet finden. Aber in der Praxis ist meistens das einfache Formular dem technisch verliebten überlegen. Schon alleine, weil es nicht lange erklärt werden muss.

2. **Übersichtlichkeit des Formulars**

Überfrachtet man ein Formular mit zu vielen Informationen, sieht der Benutzer den Wald vor lauter Bäumen nicht mehr. Dann sollte man entweder (oft nicht so gut) die Daten auf mehrere Register verteilen, oder (oft besser) mehrere Formulare verwenden.

3. **Wiedererkennungseffekte**

Darunter kann man Verschiedenes verstehen: Wenn die Nutzer z.B. vorher schon ein anderes System hatten - egal, ob Papier oder EDV - erleichtert es die Einarbeitung, wenn das Formular ähnlich aussieht. Wenn die Anwendung in mehreren Formularen z.T. gleiche Informationen zeigt, sollten diese Teile gleich aussehen. Wenn es an verschiedenen Stellen Steuerelemente mit gleichen oder sehr ähnlichen Zwecken gibt, sollten sie gleich formatiert werden (z.B. nicht einmal einen Button mit 'Drucken' beschriften, und einmal mit einem Bild).

Die Datenherkunft eines Formulars kann eine Tabelle oder eine Abfrage sein. Außerdem ist es möglich, direkt im Formular eine Abfrage in der Eigenschaft 'Datenherkunft' zu speichern. Davon ist abzuraten (siehe hierzu auch [Tipps & Tricks zu Abfragen](#)).