Wenn Sie den Benutzern Ihrer Datenbank höchstmöglichen Komfort bieten möchten, speichern Sie die Einstellungen, die er an der Datenblattansicht in Unterformularen tätigt.

Die Datenblattansicht ist sehr flexibel: Der Benutzer kann die Spaltenbreiten und die Anordnung ändern, er kann Spalten ein- und ausblenden und er kann sogar Spalten fixieren, sodass die linken x Spalten stehen bleiben, während man nach rechts scrollt, um weiter Spalten zu betrachten.

Das einzige Problem ist, dass diese Einstellungen nicht zuverlässig gespeichert werden - und unter der Runtime-Version schon gar nicht. Also müssen Sie Abhilfe schaffen und die Einstellungen selbst speichern.

Datenblatteinstellungen speichern

Datenblatteinstellungen speichern Sie mit unserem **mossDatasheetLayouter**. Um ihn einzusetzen, brauchen Sie lediglich eine einzige Klasse in die gewünschte Daten zu importieren und drei bis vier Zeilen Code zum betroffenen Unterformular hinzuzufügen. Diese sehen so aus:

```
 Dim objDataSheetForm As clsMossDatasheetLayouter  Private Sub Form_Load()
   Set objDataSheetForm = New clsMossDatasheetLayouter
   Set objDataSheetForm.DataSheetForm = Me.Form
   Set objDataSheetForm.SetStandardConfigButton = Me.Parent!cmdStandardkonfiguration
End Sub
```

Die Deklarationszeile macht die neue Klasse im Klassenmodul des Unterformulars bekannt. Die **Form_Load**-Methode erledigt dann diese zwei beziehungsweise drei Schritte:

1. Instanzierung der Klasse
2. Zuweisen eines Verweises auf das aktuelle Formular, damit die Klasse weiß, zu welchem Unterformular es die Datenblatteinstellungen speichern soll
3. Falls gewünscht, angabe einer Schaltfläche im Hauptformular, mit welcher der Benutzer die ursprüngliche Konfiguration wiederherstellen kann.

Download

Wenn Sie sich ansehen wollen, wie der mossDatasheetLayouter arbeitet, können Sie sich unsere Beispiel-Datei im .mde-Format herunterladen:

Download Beispieldatenbank (mde)

Sollte Ihnen die Lösung gefallen, finden Sie hier den kostenpflichtigen Download der Klasse **cls MossDataSheetLayouter**
:

Rechnungsart auswählen
mit Rechnung im PDF-Format, unsigniert €5,95 mit Rechnung per Post €8,95

Zu teuer?

Vielleicht. Aber nur, wenn Sie die Klasse aus dem Download in weit weniger als einer halben Stunde selbst programmieren können oder eine gleichwertige Lösung in der entsprechenden Zeit im Web finden. Ansonsten erhalten Sie hier eine gut getestete Klasse, die Sie überall einsetzen können, wo Sie Daten in der Datenblattansicht anzeigen. Und das durch Hinzufügen einer Klasse und ein paar Zeilen Code, über die Sie noch nicht einmal nachdenken müssen.

Detaillierte Anleitung

Wenn Sie die kostenpflichtige Software heruntergeladen und in ein Verzeichnis Ihrer Wahl entpackt haben, gehen Sie wie folgt vor, um die Klasse zum VBA-Projekt Ihrer Anwendung hinzuzufügen:

- Öffnen Sie die Datenbank, die Sie mit dem **DatasheetLayouter** ausstatten möchten.
- Aktivieren Sie den VBA-Editor (Alt + F11).
- Wählen Sie den Menübefehl **Datei|Datei importieren** aus.
- Selektieren Sie im nun erscheinenden Dialog die Datei **clsMossDatasheetLayouter.cls**.

Nun statten Sie das Formular, dass Daten in der Datenblattansicht anzeigen soll, mit der in **cls MossDatasheetLayouter**
enthaltenen Funktionalität aus:

- Öffnen Sie das gewünschte Formular in der Entwurfsansicht.
- Aktivieren Sie das Eigenschaftsfenster (F4).
- Wählen Sie für die Eigenschaft **Beim Laden** den Eintrag **[Ereignisprozedur]** aus, sofern diese Eigenschaft noch leer ist.

- Klicken Sie auf die Schaltfläche mit den drei Punkten.
- Der VBA-Editor wird aktiviert und zeigt, sofern Sie zuvor noch keine Ereignisprozedur für das Ereignis **Beim Laden** des Formulars angelegt haben, eine leere Ereignisprozedur namens **Form_Load** an.
- Fügen Sie im Kopf des Moduls die folgende Deklarationszeile ein:

```
Dim objDataSheetForm As clsMossDatasheetLayouter
```

- Ergänzen Sie die Prozedur wie folgt:

```
Private Sub Form_Load()
  Set objDataSheetForm = New clsMossDatasheetLayouter
  Set objDataSheetForm.DataSheetForm = Me.Form
End Sub
```

Dies sorgt dafür, dass die Klasse beim Öffnen des Formulars (beziehungsweise Unterformulars) instanziert und mit einem Verweis auf das aktuelle Formular gefüllt wird. Im gleichen Zuge prüft die Klasse, ob die Datenbank bereits eine Tabelle zum Speichern der Konfigurationsdaten des Datasheets enthält. Falls nicht, fragt sie nach, ob die Tabelle angelegt werden soll. Dem sollten Sie zustimmen, da die Lösung sonst nicht funktioniert. Beim nächsten Schließen speichert der **DatasheetLayouter** die aktuelle Konfiguration des Datasheets und stellt sie beim nächsten Öffnen wieder her.

Standardkonfiguration speichern und wiederherstellen

Beim ersten Schließen des Formulars wird die aktuelle Konfiguration nicht nur als aktuelle, sondern auch als Standardkonfiguration gespeichert. Diese können Sie über eine Schaltfläche im übergeordneten Formular, also dem Hauptformular des Unterformulars, wieder herstellen. Dazu legen Sie eine entsprechende Schaltfläche an, die Sie etwa **cmdStandardkonfiguration** nennen, und fügen der
Form_Load
-Prozedur des Unterformulars noch eine weitere Zeile hinzu. Die Prozedur sieht nun so aus (neue Zeile fettgedruckt):

```
Private Sub Form_Load()
```

```
    Set objDataSheetForm = New clsMossDatasheetLayouter
    Set objDataSheetForm.DataSheetForm = Me.Form
    Set objDataSheetForm.SetStandardConfigButton =
Me.Parent!cmdStandardkonfiguration
End Sub
```

Sie brauchen dieser Schaltfläche keine Ereignisprozedur zuweisen, dies geschieht alles automatisch beim Zuweisen der Schaltfläche zur Eigenschaft **SetStandardConfigButton** der Klasse.

Einsatz in mehreren Formularen

Sie können den **DatasheetLayouter** problemlos in mehreren Formularen der gleichen Anwendung einsetzen. Die Konfigurationsdaten werden unter Angabe des Formular- und Steuerelementnamens gespeichert. Dazu brauchen Sie einfach nur die oben angegebenen Codezeilen zur **Beim Laden**-Ereignisprozedur zuzuweisen.

Bei Problemen ...

... schreiben Sie uns einfach einen Kommentar mit der Fehlerbeschreibung. Gern können Sie uns auch Verbesserungsvorschläge unterbreiten. Kommentare werden nur veröffentlicht, wenn diese von allgemeinem Interesse sind.