

18.10 ScrollView

Wenn das Layout für eine Programmoberfläche statisch ist, kommen Sie bei zu geringer Fenstergröße in die Situation, dass einige Komponenten nicht mehr vollständig sichtbar und andere völlig unsichtbar sind. Ein Ausweg bietet sich mit dem Einsatz einer ScrollView an, die auf das Formular gelegt wird und bei dem die Eigenschaft *Form.Arrangement* auf *Arrange.Fill* gesetzt ist. Nun erreichen Sie alle Komponenten sicher mit Hilfe der Scrollbalken.

Die Klasse *ScrollView* implementiert einen Container, in dem man durch seinen Inhalt scrollen kann. Bildlaufleisten werden automatisch angezeigt, wenn Komponenten in der ScrollView nicht oder nur teilweise sichtbar sind.

18.10.1 Eigenschaften ScrollView

Eigenschaft	Datentyp	Default	Beschreibung
Arrangement	Integer	0	Ermittelt oder legt fest, wie die Komponenten in der ScrollView angeordnet werden sollen. In der Klasse <i>Arrange</i> → Kapitel 20.7.2 finden Sie eine Liste von Konstanten für diese Eigenschaft.
Border	Integer	0	Setzt den anzuzeigenden Typ des Randes oder gibt diesen Typ zurück. In der Klasse <i>Border</i> → Kapitel 23.2.1 finden Sie eine Liste von Konstanten für diese Eigenschaft.
Indent	Boolean	False	Wenn wahr, dann wird das erste Element eingerückt. Die Einrücktiefe beträgt <i>Desktop.Scale</i> (Pixel). Die Richtung wird durch den Wert von <i>.Invert</i> festgelegt.
Invert	Boolean	False	Wenn wahr, dann wird der Inhalt im Container nicht von links nach rechts sondern umgekehrt angeordnet.
Margin	Boolean	False	Wenn wahr, dann wird ein Zwischenraum zwischen dem Containerrand und den Elementen im Container eingefügt. Dieser Abstand beträgt <i>Desktop.Scale</i> (Pixel).
Padding	Integer	0	Es wird ein Abstand von k Pixel zwischen dem Containerrand und den Elementen im Container eingefügt.
Spacing	Boolean	False	Wenn wahr, dann wird ein (horizontaler) Zwischenraum zwischen die Elemente im Container eingefügt. Dieser Abstand beträgt <i>Desktop.Scale</i> (Pixel).
Scrollbar	Integer	0	Über die Konstanten <i>None</i> (0), <i>Horizontal</i> (1), <i>Vertical</i> (2) und <i>Both</i> (3) geben Sie an, ob keine, je eine oder je nach Bedarf eine oder zwei Scroll-Balken angezeigt werden sollen. Sie können die Anzahl und Art der angezeigten Scroll-Balken auch auslesen.
ScrollX	Integer	0	Liest die horizontale Scroll-Position aus oder setzt sie.
ScrollY	Integer	0	Liest die vertikale Scroll-Position aus oder setzt sie.

Tabelle 18.10.1.1: Übersicht Eigenschaften ScrollView

18.10.2 Methoden ScrollView

Mit den folgenden 2 speziellen Methoden einer ScrollView steuern Sie die sichtbaren Bereiche in einer ScrollView:

- **Scroll (X As Integer, Y As Integer)** : Scrollt den Inhalt an die vorgegebene Position in der ScrollView.
- **EnsureVisible (X As Integer, Y As Integer, Width As Integer, Height As Integer)** : Sorgt dafür, dass ein bestimmter Bereich der ScrollView sichtbar wird.

18.10.3 Ereignisse ScrollView

Die ScrollView besitzt nur ein spezifisches Ereignis:

Das Ereignis *ScrollView_Scroll()* wird ausgelöst, wenn durch den Inhalt der ScrollView gescrollt wird.

18.10.4 Projekt Datenbankmanager mit ScrollView

Ein Einsatzgebiet für eine ScrollView im vorgestellten Projekt ist ein Formular, dem dynamisch Komponenten hinzugefügt werden, wobei sich die Anzahl dieser Komponenten zur Programm-Laufzeit ständig verändern kann. Im Projekt wird der Inhalt einer Datenbank-Tabelle (MySQL, PostgreSQL oder SQLite3) ausgelesen und angezeigt.

Für jedes Feld des Tabellen-Schemas der in einem *Dialog* ausgewählten Datenbank-Tabelle wird ein Komponenten-Paar (Label,DB-Komponente) in einer ScrollView erzeugt. Intern ist jedes Komponenten-Paar an die Komponenten *DataSource* und *DataBrowser* gebunden werden, so dass die Feldinhalte auch editiert werden können! Das Label zeigt den Feldnamen an. Für die Feldtypen Boolean und Date werden als DataControl ComboBoxen respektive DateBoxen bereitgestellt. Die Daten werden somit kontrolliert eingegeben. Für die anderen Typen führt Gambas – in Abhängigkeit vom Datenbank-Datentyp – automatisch eine Prüfung durch.

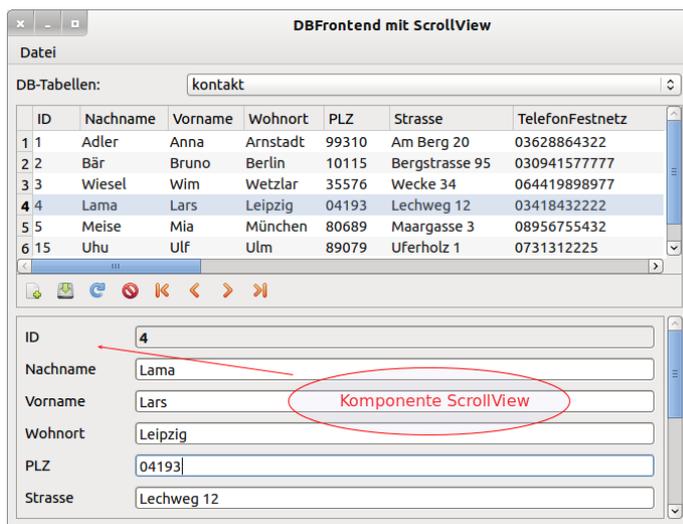


Abbildung 18.10.4.1: Universeller DB-Manager

In der Abbildung 18.10.4.1 sehen Sie, wie Sie einerseits durch die Tabelle blättern können und andererseits auch durch den Inhalt der ScrollView. In den Mittelpunkt der Betrachtungen rückt jetzt die Prozedur, in der die o.a. Komponenten-Paare in der ScrollView (*svwControl*) generiert werden:

```

Private Sub GenerateControls()
    Dim hField As Field
    Dim hLabel As Label
    Dim hControl As DataControl

    For Each hField In $hConn.Tables[srcData.Table].Fields
        ' LABEL
        hLabel = New Label(svwControl)
        hLabel.Text = hField.Name
        hLabel.Move(0, 0, 100, 24)

        ' DB-KOMPONENTE
        hControl = New DataControl(svwControl)
        hControl.Move(100, 0, 100, 24)
        hControl.Expand = True
        hControl.Field = hField.Name
    Next hField

    svwControl.Visible = True
End Sub GenerateControls()

```

Das vollständige Projekt des universellen Datenbankmanagers von Tobias Boege finden Sie im Download-Bereich. Es enthält im Ordner *DBMuster* eine SQLite3-Datenbank *Kontakte.sqlite*, deren Datenbanktabelle *kontakt* Sie bei der Erprobung des universellen DB-Managers verwenden können.