

17.7.2 Ausgewählte Gittereigenschaften GridView

Das sind die wesentlichen Gittereigenschaften, die Sie setzen oder auslesen können:

Eigenschaft	Beschreibung
GridView.Header	Die Eigenschaft bestimmt, ob eine Kopfzeile angezeigt wird oder nicht (0=None) oder in welcher Ausrichtung sie angezeigt wird: 3=Both, 1=Horizontal oder 2=Vertical

Tabelle: 17.7.2.1 Beschreibung Gittereigenschaft .Header

Die Eigenschaften können für alle Spalten oder nur für ausgewählte gelten:

Eigenschaft	Read/Write	Beschreibung
GridView.Columns.Count	RW	Anzahl der Spalten in der GridView
GridView.Columns.Width	RW	Breite aller Spalten in Pixeln
GridView.Columns[k].Width	RW	Breite der Spalte k in Pixeln
GridView.Resizable GridView.Columns.Resizable	RW	Resizable legt mit dem Wert TRUE oder FALSE fest, ob alle Spalten in der Breite verändert werden können oder nicht und muss vor allen anderen Eigenschaften gesetzt werden.
GridView.Columns[k].Resizable	RW	Resizable legt mit dem Wert TRUE oder FALSE fest, ob die Spaltenbreite der Spalte k verändert werden kann oder nicht und muss <u>nach</u> GridView.Columns.Count gesetzt werden.
GridView.Columns.Height	R	Die Höhe der Kopfzeile in Pixel wird ausgelesen

Tabelle: 17.7.2.2 Beschreibung ausgewählter Gittereigenschaften – Spalten

Für die Zeilen in der GridView gilt Analoges:

Eigenschaft	Read/Write	Beschreibung
GridView.Rows.Count	RW	Anzahl der Gitter-Zeilen. Wenn die Anzahl 0 ist, dann wird <i>kein</i> Daten-Gitter angezeigt.
GridView.Rows.Width	R	Höhe der Spaltenüberschrift in Pixeln
GridView.Rows[k].Height	RW	Höhe der Zeile k
GridView.Rows.Resizable	RW	Resizable legt mit dem Wert TRUE oder FALSE fest, ob die Zeilenhöhe verändert werden kann oder nicht.
GridView.Rows.Height	RW	Höhe aller Daten-Zeilen in Pixeln

Tabelle: 17.7.2.3 Beschreibung ausgewählter Gittereigenschaften – Zeilen

Es folgt die Beschreibung weiterer Eigenschaften der GridView:

Eigenschaft	Read/Write	Beschreibung
GridView.Column	RW	Setzt oder liest die aktuelle Spalte
GridView.Row	RW	Setzt oder liest die aktuelle Zeile
GridView.Current	R	Gibt die Eigenschaften einer Zelle zurück; Typ <i>GridViewCell</i> : Alignment, BackColor, Background, Font, ForeColor, Foreground, Height, Left, Padding, Picture, RichText, Text, Top, Width, X, Y
GridView.Grid	W	Mit den Werten TRUE oder FALSE wird bestimmt, ob Gitter-Linien gezeichnet werden oder nicht.
GridView.Mode	RW	Der Selektions-Modus für Zeilen hat die 3 Modi: Select.None, Select.Single und Select.Multiple. Nur in den Modi Select.Single und Select.Multiple werden die ausgewählten Zeilen farbig markiert, sonst u.a. der Spaltenkopf in Fettschrift.
GridView[z,s].Text	W	Text in der Zelle [zeile,spalte]
GridView[z,s].Alignment	RW	Text-Ausrichtung in der Zelle [z,s] mit 13 verschiedenen Werten. Vergleichen Sie mit den Angaben im Kapitel 20.7 zur Klasse <i>Align</i> .
GridView[z,s].Picture	W	Sie können auch multimediale Objekte – zum Beispiel ein Bild – mit Picture["image.png"] in eine Zelle einfügen.
GridView.Data	RW	Die virtuelle Klasse gilt nur in Verbindung mit dem GridView.Data-Event und hat folgende Eigenschaften: Alignment, BackColor, Background, Font, ForeColor, Foreground, Padding, Picture, RichText und Text für den Gitter-Inhalt.

Tabelle: 17.7.2.4 Beschreibung ausgewählter Gittereigenschaften

Interessant sind vor allem diese beiden Methoden:

Methode	Beschreibung
GridView.Clear	Das Daten-Gitter wird gelöscht. Wenn zum Beispiel mit <i>GridView.Header = 1</i> eine Kopfzeile mit Spaltenüberschriften angezeigt wird, dann bleibt diese erhalten.
GridView.MoveTo(z,s)	Setzt die aktuelle Zelle auf Zeile=z und Spalte=s

Tabelle: 17.7.2.5 Beschreibung ausgewählter Methoden

Es hat sich als vorteilhaft erwiesen, das Setzen der Gittereigenschaften in einer eigenen Prozedur zu realisieren, wie es die folgenden Auszüge aus zwei verschiedenen Quelltexten zeigen. Im ersten Ausschnitt ist die Anzahl der Spalten bekannt:

```
PRIVATE SUB SetGridProperty()
    GridView1.Resizable = TRUE ' Muss vor allen anderen Eigenschaften gesetzt werden
    GridView1.Mode = Select.Single
    GridView1.Columns.Count = 4
    GridView1.Header = GridView1.Both
    GridView1.Columns[0].Width = 110
    GridView1.Columns[0].Title = "Zahl"
    GridView1.Columns[1].Width = 120
    GridView1.Columns[1].Title = "Wahrheitswert"
    ...
END ' SetGridProperty()
```

Im zweiten Quelltext-Auszug werden die Anzahl der Spalten ermittelt sowie die Spaltenüberschriften dynamisch generiert:

```
PUBLIC SUB SetGridViewProperty()
    DIM iDatensatznummer, iSpaltennummer AS Integer

    GridView.Header = 1 ' GridView.Horizontal
    GridView.Columns.Count = rDBResult.Fields.Count
    GridView.Columns[0].Width = 25
    GridView.Columns[0].Resizable = FALSE ' Nur diese Feldbreite ist fix
    ' GridView-Spalten-Bezeichner ermitteln und eintragen
    FOR iSpaltennummer = 0 TO rDBResult.Fields.Count - 1
        GridView.Columns[iSpaltennummer].Title = rDBResult.Fields[iSpaltennummer].Name
    NEXT ' Feldname
END ' SetGridViewProperty
```

In den folgenden Abschnitten werden Ihnen Projekte vorgestellt, die eine GridView zur Anzeige von Daten aus unterschiedlichen Quellen nutzen und verschiedene Möglichkeiten zum Einfügen der Daten bieten:

- GridView mit fester Anzahl der Spalten; Einfügen der Daten ohne Zwischenspeicherung
- GridView mit fester Anzahl der Spalten; Speichern der Daten in einer Matrix und Einfügen der Daten mit dem Ereignis *GridView_Data(Row AS Integer, Column AS Integer)* – verbunden mit Möglichkeiten, zum Beispiel die Gitterzeilen oder Gitterzellen in besonderer Weise farblich auszuzeichnen.
- Anzeige von Daten aus einer Datenbank-Tabelle, wobei die Anzahl der Zeilen und Spalten – abhängig von der SQL-Anfrage – beim Programmstart nicht bekannt ist.
- Daten-Export aus einer GridView in eine csv-Datei.
- Daten-Import aus einer csv-Datei und Anzeige der Daten in einer GridView.

Die nächsten zwei Projekte machen nur dann Sinn, wenn Sie die manipulierten Daten in der Gitteransicht zum Beispiel in einer csv-Datei speichern oder aus dieser wieder einlesen können:

- Änderung von Inhalten ausgewählter Zellen in einer GridView,
- Änderung ausgewählter Zellen in einer GridView unter Verwendung einer speziellen Klasse.

Das abschließende Projekt benötigt Gambas 3 – mindestens in der Revision 4715:

- Sortierung von Daten in einer GridView

Der Quelltext wird nur für jene Prozeduren und Funktionen vollständig angegeben und erläutert, die für das Verständnis der verwendeten Eigenschaften und der eingesetzten Methoden der GridView notwendig sind.