

### 17.4.0 ListView

Das Steuerelement ListView in gb.qt4 (→ gb.gui.base) implementiert eine Liste mit auswählbaren Elementen.

- Die einzelnen Elemente der Liste werden durch einen eindeutigen Schlüssel indiziert. Sie zeigen einen Text und optional ein Bild vor jedem Text.
- Das Steuerelement verfügt über einen internen Zeiger. Verwenden Sie die unterschiedlichen Move-Methoden, um den internen Zeiger zu bewegen und die Current-Eigenschaft, um auf das aktuelle Element zuzugreifen.
- Mit den verschiedenen Move-Methoden können Sie den internen Zeiger verschieben. Wenn Sie den internen Zeiger auf ein bestimmtes Element setzen wollen, dann müssen Sie die Methode MoveTo(Key) benutzen. Beachten Sie, dass diese Move-Methoden den Wert *True* zurück geben, wenn die Bewegung des internen Zeigers *nicht* ausgeführt werden kann.
- Um den sichtbaren Zeiger – als Auswahl eines bestimmten Elements der ListView – zu setzen, weisen Sie der Key-Eigenschaft eines Elements den Schlüssel des zu markierenden Elements zu.

#### 17.4.0.1 Eigenschaften ListView

Die Klasse *ListView* verfügt u.a. über diese Eigenschaften:

Eigenschaft	Datentyp	Beschreibung
Current	_TreeView_Item	Gibt das aktuelle Element zurück. Im Allgemeinen ist es das Element, das den Fokus hat oder das markiert ist (Single-Selection-Modus).
Item	_TreeView_Item	Gibt das Element zurück, auf das der interne Zeiger zeigt oder NULL, wenn der interne Zeiger nicht erreichbar ist.
Count	Integer	Gibt die Anzahl aller Elemente in der ListView zurück.
Editable	Boolean	Gibt an, ob die Elemente standardmäßig editierbar sind. Das heißt, ob sie vom Benutzer umbenannt werden können, wenn er auf sie klickt.
Key	String	Gibt den Key des aktuellen Elements zurück.
Mode	Integer	Setzt den Auswahlmodus oder gibt den Auswahlmodus zurück. Die verwendeten Konstanten sind: Single, None oder Multiple.
Selection	String[ ]	Zurückgegeben werden die Keys aller markierten Elemente in einem String-Array (Multi-Selection-Modus).
Sorted	Boolean	Zeigt an oder legt fest, ob die Elemente in der ListView sortiert <i>angezeigt</i> werden.

Tabelle 17.4.0.1.1 : Ausgewählte Eigenschaften der Klasse ListView

#### 17.4.0.2 Methoden ListView

Die Klasse *ListView* besitzt diese ausgewählten Methoden:

Methode	Rückgabotyp	Beschreibung
Exist ( Key As String )	Boolean	Zurückgegeben wird der Wert True, wenn das Element mit dem angegebenen Key existiert.
Add ( Key As String, Text As String [ , Picture As Picture, After As String ] )	_TreeView_Item	Fügt ein Element in die ListView ein.
MoveBack()	Boolean	Wenn eine der anderen Move-Methoden den Wert True zurück gibt, so können Sie diese Methode benutzen, um den internen Zeiger an seine ursprüngliche Position zu bewegen. Es wird True zurück gegeben, wenn der interne Zeiger nicht auf einer gültigen Position war.
MoveBelow()	Boolean	Bewegt den internen Zeiger auf ein sichtbares Element unterhalb des aktuellen. Es wird <i>True</i> zurückgegeben, wenn die Ver-

Methode	Rückgabotyp	Beschreibung
		schiebung <u>nicht</u> ausgeführt werden kann.
MoveTo ([ Key As String ])	Boolean	Setzt den internen Cursor auf das Element in der ListView mit dem angegebenen Key.
MoveBelow()	Boolean	Bewegt den internen Zeiger auf ein sichtbares Element unterhalb des aktuellen. Es wird <i>True</i> zurückgegeben, wenn die Verschiebung <u>nicht</u> ausgeführt werden kann.
MoveCurrent()	Boolean	Setzt den internen Cursor auf das markierte Element in der ListView. Es wird <i>True</i> zurückgegeben, wenn die Verschiebung <u>nicht</u> ausgeführt werden kann.
MoveFirst()	Boolean	Setzt den internen Cursor auf das erste Element in der ListView. Es wird <i>True</i> zurückgegeben, wenn die Liste leer ist.
MoveLast()	Boolean	Setzt den internen Cursor auf das letzte Element in der ListView. Es wird <i>True</i> zurückgegeben, wenn die Liste leer ist.
Rename()	Boolean	Startet das Umbenennen des markierten Elements.

Tabelle 17.4.0.2.1 : Ausgewählte Methoden der Klasse ListView

Die Add(...) - Methode fügt ein neues Element in die ListView ein:

```
Function Add ( Key As String, Text As String [ , Picture As Picture, After As String ] ) As _TreeView_Item
```

- *Key* ist der Schlüssel für das neue Element.
- *Text* ist der angezeigte Text auf dem neuen Element.
- *Picture* ist das Bild, das vor dem Text angezeigt wird. Standardmäßig wird kein Bild angezeigt.
- *After* ist der Schlüssel von dem Element, nach dem das neue Element eingefügt wird. Ohne diesen (optionalen) Schlüssel wird das neue Element standardmäßig als letztes Element eingefügt.

Im folgenden Beispiel aus dem Projekt *ListViewE* werden die einzelnen Elemente in der ListView aus dem Inhalt einer Datei generiert:

```
Private Sub ListViewImport(Path As String, hListView As ListView)

    Dim cJSONCollection As JSONCollection
    Dim cCollection As JSONCollection
    Dim sKey, sText As String
    Dim picPicture As Picture

    hListView.Clear
    cJSONCollection = JSON.Decode(File.Load(Path), True)
    For Each cCollection In cJSONCollection
        sKey = cCollection["KEY"]
        sText = cCollection["TEXT"]
        picPicture = Picture[cCollection["PICTURE"]]
        hListView.Add(sKey, sText, picPicture, Null) ' Null, weil die Reihenfolge feststeht
    Next
End ' ListViewExport(...)
```

### 17.4.0.3 Ereignisse ListView

Die Klasse *ListView* besitzt unter anderem die folgenden Ereignisse:

Ereignis	Beschreibung
Click()	Das Ereignis wird ausgelöst, wenn auf ein Element in der ListView geklickt wird.
Select()	Das Ereignis wird ausgelöst, wenn sich die Auswahl in der ListView ändert.
Compare ( Key As String, OtherKey As String )	Dieses Ereignis wird ausgelöst, wenn zwei Elemente in der ListView verglichen werden. Das Ergebnis des Vergleichs wird in der <i>Compare</i> -Eigenschaft gespeichert.

Tabelle 17.4.0.3.1 : Ausgewählte Ereignisse der Klasse ListView

Im nächsten Abschnitt werden Ihnen wichtige Eigenschaften und Methoden der virtuellen Klasse *\_TreeView\_Item* vorgestellt, da die Eigenschaften *Current* und *Item* vom Typ *\_TreeView\_Item* sind.

17.4.0.4 Eigenschaften und Methoden der Klasse `_TreeView_Item`

Die virtuelle Klasse `_TreeView_Item` repräsentiert einen Eintrag in einer ListView.

17.4.0.5 Eigenschaften der Klasse `_TreeView_Item`

Eigenschaft	Datentyp	Beschreibung
Count	Integer	Gibt die Anzahl der Elemente zurück.
Editable	Boolean	Gibt an, ob dieses Element editierbar ist. Das bedeutet, ob es vom Anwender umbenannt werden kann, wenn er darauf klickt.
Key	String	Gibt den Key zurück.
Picture	Picture	Gibt das Bild zurück oder setzt das Bild, das neben dem Element angezeigt wird.
RichText	String	Gibt den RichText zurück oder setzt den RichText.
Selected	Boolean	Gibt an, ob das Element ausgewählt ist.
Text	String	Gibt den angezeigten Text auf dem Element zurück oder setzt den anzuzeigenden Text.

Tabelle 17.4.0.5.1 : Ausgewählte Eigenschaften der virtuellen Klasse `_ListView_Item`

Seine Stärke als Steuerelement zur Anzeige von Daten zeigt die ListView erst dann so richtig, wenn Sie den Text als RichText erzeugen und zusammen mit einem Bild (Datentyp Picture) vor dem Text verwenden:



Abbildung 17.4.0.5.1: Verwendung von RichText in einer ListView

17.4.0.6 Methoden der Klasse `_TreeView_Item`

Methode	Beschreibung
Delete()	Löscht das (aktuelle) Element aus der ListView.
Ensurevisible()	Sichert die (sichtbare) Anzeige des Elements in der ListView (Scroll-Effekt).
Rename()	Startet das Umbenennen des markierten Elements.

Tabelle 17.4.0.6.1 : Ausgewählte Methoden der virtuellen Klasse `_TreeView_Item`

Die Verwendung unterschiedlicher Methoden zeigt der folgende, ausführlich kommentierte Beispiel-Quelltext zum Löschen eines (markierten) Elements:

```
Public Sub btnDeleteElement_Click()  
    ' Nur ein markiertes Element darf gelöscht werden  
    If Not ListView1.Key Then Return  
    ' Zuerst den internen Cursor auf das markierte Element setzen (Start-Position)  
    ListView1.MoveCurrent()  
    ' Dann den internen Cursor auf das nächste *sichtbare* - oberhalb liegenden - Element setzen  
    ListView1.MoveAbove() ' ***  
    ' Das markierte Element wird gelöscht  
    ListView1.Remove(ListView1.Key)  
    ' Der interne Cursor zeigt nach (***) auf das nach dem Löschen auszuwählende Element.  
    ' Gibt es nach dem Löschen noch ein oberes Element, dann wird der externe Cursor darauf  
    ' gesetzt → es wird markiert.  
    If ListView1.Item Then  
        ListView1.Key = ListView1.Item.Key  
        ' Sonst wird der interne Cursor auf den ersten Eintrag gesetzt,  
        ' wenn er existiert und dann das oberste Element markiert.  
    Else  
        If Not ListView1.MoveFirst() Then ListView1.Key = ListView1.Item.Key  
    Endif  
End Sub btnDeleteElement_Click()
```