

17.4.1 Projekt – Exkurs ListView

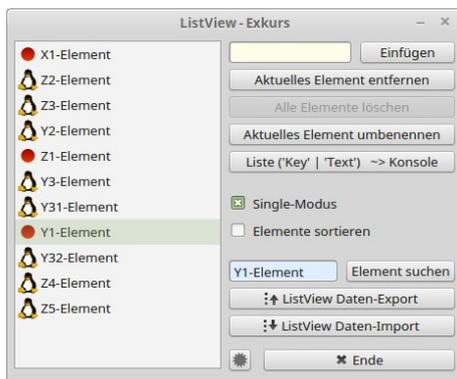


Abbildung 17.4.1.1: ListView mit markiertem Eintrag (nach einer Suche)

Das Projekt demonstriert, wie Sie mit einer ListView arbeiten können:

- Ein Element einfügen – Text auch im RichText-Format,
- ein Element markieren,
- den Selektionsmodus festlegen (single oder multiple),
- das aktuelle Element löschen,
- das aktuelle Element umbenennen,
- ein Element suchen,
- formatierte Ausgabe aller Elemente einer ListView (Konsole),
- Export aller Elemente in der ListView im JSON-Format → Kapitel 24.9.0 JSON. Die Daten der ListView werden inklusive ihrer Hierarchie in einer Datei *export.dat* im Projekt-Verzeichnis gespeichert,
- Import aller Elemente (JSON-Format) und Anzeige der importierten Elemente in der ListView in der ursprünglichen Hierarchie.
- Markierung für alle Elemente entfernen,
- die Sortierung der Elemente in der ListView ein- und ausschalten.
- Bei einem Klick auf ein Element in der ListView werden ausgewählte Informationen zum markierten Element in der Konsole der Gambas-IDE ausgegeben:

```
Key = KX1 | Text = X1-Element | Bild-Pfad = icon:/22/record
Key = KY1 | Text = Y1-Element | Bild-Pfad = icon:/22/record
Key = KY31 | Text = Y31-Element | Bild-Pfad = icon:/22/linux
...
```

Den recht umfangreichen Quelltext finden Sie im Projekt 'ListViewE' im Download-Bereich. Der Quelltext ist hinreichend kommentiert. Mit dem folgenden Quelltext-Abschnitt werden zum Beispiel neue Elemente in die ListView eingefügt:

```
Public Sub btnInsertElement_Click()

  If txbElement.Text <> Null Then
    If Not ListView1.Exist("K" & txbElement.Text) Then
      ListView1.MoveCurrent()
      ListView1.Add("K" & txbElement.Text, , Picture["icon:/22/info"], ListView1.Key)
      ListView1.Item.EnsureVisible()
    Else
      Message.Warning("Das Element '" & txbElement.Text & "' existiert bereits!")
      txbElementName.Clear()
      Return
    Endif
    txbElementName.Clear()
    ' Das neue Element wird markiert
    ListView1.Key = ListView1.Item.Key
  Endif

End Sub btnInsertElement_Click()
```

Sie sollten sich rechtzeitig entscheiden, ob die Elemente nach dem Einfügen sortiert angezeigt werden sollen oder nicht. Die Eigenschaft 'Sorted' können Sie *nicht* als Umschalter zwischen sortierter Anzeige und nicht-sortierter Anzeige verwenden. So können Sie mit dem Anklicken der CheckBox mit dem

Bezeichner 'Elemente sortieren' die Anzeige einmalig auf sortierte Anzeige umstellen. Jede weitere Änderung des Wertes (True oder False) der CheckBox – interpretiert als 'sortieren' oder 'nicht sortieren' – hat dann folgende Wirkung:

- Wert 'sortieren': Jedes neue Element wird so eingefügt, wie es der Sortierreihenfolge entspricht.
- Wert 'nicht sortieren': Jedes neue Element wird *nach* dem markierten Element eingefügt.

Für eigene Projekte müssen Sie sich die Passagen aus dem Quelltext heraussuchen, die Sie benötigen.

Hinweis: Die Eigenschaft *Current* repräsentiert den vom Benutzer markierten Eintrag in einer ListView, während die Eigenschaft *Item* vom Typ `_TreeView_Item` ein interner, unsichtbarer und von der Eigenschaft *Current* unabhängiger Zeiger ist. *Current* verwendet man zur Auswertung von Benutzereingaben in einer ListView und *Item* für alle Algorithmen, welche eine ListView durchlaufen.