

18.0 Container

Im Kapitel 18 werden Ihnen Container als die Komponenten vorgestellt, welche andere Komponenten aufnehmen. Zu den Containern gehören:

- Frame,
- Expander,
- Embedder,
- Panel, SidePanel, HPanel, VPanel,
- ToolPanel,
- ToolBar,
- HSplit, VSplit, HBox, VBox,
- ScrollView,
- DrawingArea,
- ListContainer.

Es werden die wichtigsten Eigenschaften und Methoden der verschiedenen Container beschrieben, wobei dem wichtigsten Container – dem Formular – ein eigenes Kapitel 12 gewidmet ist.

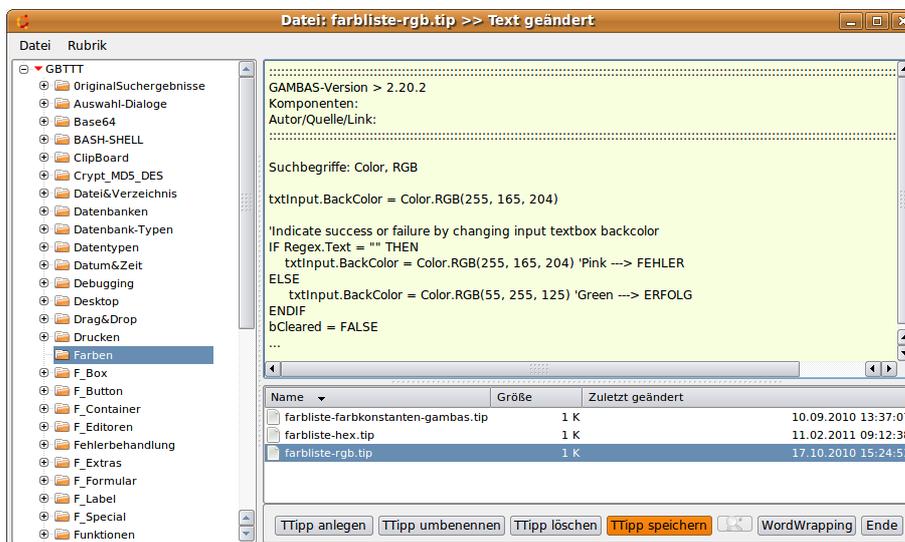


Abbildung 18.0.1: Programmoberfläche für das Projekt GBTT mit den Containern HSplit, VSplit und VBox

18.0.1 Eigenschaften Container

Die folgenden Eigenschaften finden Sie bei allen Containern:

Eigenschaft	Datentyp	Beschreibung
.AutoSize	Boolean	Wenn wahr, dann passt sich die Größe der HBox an den Inhalt im Container an.
.Indent	Boolean	Wenn wahr, dann wird das erste Element eingerückt. Die Einrücktiefe beträgt Desktop.Scale (Pixel). Die Richtung wird durch den Wert von <i>.Invert</i> festgelegt.
.Invert	Boolean	Wenn wahr, dann wird der Inhalt im Container nicht von links nach rechts sondern umgekehrt angeordnet. Nicht alle Container besitzen diese Eigenschaft.
.Margin *	Boolean	Wenn wahr, dann wird ein Zwischenraum zwischen dem Containertrand und den Elementen im Container eingefügt. Dieser Abstand beträgt Desktop.Scale (Pixel).
.Padding *	Integer	Es wird ein Abstand von k Pixel (Default 0) zwischen dem Containertrand und den Elementen im Container eingefügt.
.Spacing *	Boolean	Wenn wahr, dann wird ein (horizontaler) Zwischenraum zwischen die Elemente im Container eingefügt. Dieser Abstand beträgt Desktop.Scale (Pixel).

Tabelle 18.0.1.1: Übersicht zu den Eigenschaften von Containern

18.0.2 Hinweise zum Zusammenhang der Eigenschaften Margin, Spacing und Padding *

Wenn Sie die Eigenschaft `Margin` auf `True` setzen und zusätzlich auch die Eigenschaft `Padding` auf `k` (Pixel), dann wird die Eigenschaft `Margin` ignoriert! Als Empfehlung gilt: Wählen Sie die Eigenschaft `Margin`, für einen festen Abstand mit der konstanten Distanz `Desktop.Scale` zwischen Container-Rand und den Elementen im Container und die Eigenschaft `Padding`, um einen variablen Abstand mit der Distanz `k` ($k \geq 1, k \in \text{Integer}$) Pixel festzulegen.

Die Eigenschaft `Padding` definiert den Abstand und somit gilt für einen Container `C`:

- Wenn `C.Margin = True` und `C.Padding = 0`, dann ist `C.Margin = Desktop.Scale`
- Wenn `C.Margin = True` und `C.Padding = 5`, dann ist `C.Margin = 5`
- Wenn `C.Spacing = True` und `C.Padding = 0`, dann ist `C.Spacing = Desktop.Scale`
- Wenn `C.Spacing = True` und `C.Padding = 5`, dann ist `C.Spacing = 5`

18.0.3 Beispiele

Im diesem Abschnitt werden nur Beispiele für die Wirkungen der Container-Eigenschaften `Margin`, `Spacing`, `Indent` und `Invert` vorgestellt und ein weiterer Container eingefügt – dargestellt am Container `HBox`:



Abbildung 18.0.3.1: Zwei identische Container vom Typ `HBox` mit jeweils 4 Button; ungeordnet

- Ziehen Sie zwei Container vom Typ `HBox` (Hintergrundfarbe grau) mit einer Höhe von 56 Pixel sowie mit 4 Button (40x40px) wie in der Abbildung 18.0.3.1 in der IDE auf ein Formular.
- Setzen Sie ausgewählte Container-Eigenschaften (`Margin`, `Spacing`, `Indent` und `Invert`) auf die in den einzelnen Beispielen angegebenen Werte.
- Präsentieren und vergleichen Sie die Ergebnisse in 1 bis 5.

18.0.3.1 Einstellungen Beispiel 1

```
HBox1.Margin = False HBox1.Spacing = False HBox1.Indent = False HBox1.Invert = False
HBox2.Margin = True HBox2.Spacing = False HBox2.Indent = False HBox2.Invert = False
```



Abbildung 18.0.3.1.1: Ergebnis Beispiel 1 – Inhalt von `HBox2` mit konstantem Abstand zum Rand

18.0.3.2 Einstellungen Beispiel 2

```
HBox1.Margin = False HBox1.Spacing = False HBox1.Indent = False HBox1.Invert = False
HBox2.Margin = True HBox2.Spacing = True HBox2.Indent = False HBox2.Invert = False
```



Abbildung 18.0.3.2.1: Ergebnis Beispiel 2 – Inhalt von `HBox2` zusätzlich mit konstantem Abstand zwischen den 4 Button

18.0.3.3 Einstellungen Beispiel 3

```
HBox1.Margin = True HBox1.Spacing = True HBox1.Indent = False HBox1.Invert = False
HBox2.Margin = True HBox2.Spacing = True HBox2.Indent = False HBox2.Invert = True
```



Abbildung 18.0.3.3.1: Ergebnis Beispiel 3 – Ausrichtung im Container `HBox2` invertiert

18.0.3.4 Einstellungen Beispiel 4

```
HBox1.H = 40      HBox2.H = 40
HBox1.Margin = False  HBox1.Spacing = False  HBox1.Indent = False  HBox1.Invert = False
HBox2.Margin = False  HBox2.Spacing = False  HBox2.Indent = True   HBox2.Invert = False
```



Abbildung 18.0.3.4.1: Ergebnis Beispiel 4 – Einrückung um Desktop.Scale Pixel nach rechts im Container HBox2

18.0.3.5 Einstellungen Beispiel 5



Abbildung 18.0.3.5.1: Beispiel 5 – Höhe der Container 56px

- Setzen Sie die Höhe der beiden Container HBox auf 56px.
- Fügen Sie in den Container HBox2 ein Panel ein → Abbildung 18.0.3.5.1.

```
HBox1.H = 56
HBox2.H = 56
panSpace.Expand = True
HBox1.Margin = False  HBox1.Spacing = False  HBox1.Indent = False  HBox1.Invert = False
HBox2.Margin = True   HBox2.Spacing = True   HBox2.Indent = True   HBox2.Invert = False
```



Abbildung 18.0.3.5.2: Beispiel 5 – Höhe der Container 56px

Das Panel 'schiebt' den letzten Button an den rechten Rand, weil das Panel auf Grund der gesetzten Eigenschaft *panSpace.Expand = True* den gesamten verfügbaren Raum zwischen dem 3. und dem 4. Button einnimmt.

Platzieren Sie zum Beispiel – bei sonst gleichen Eigenschaftswerten – das Panel mit dem Namen *panSpace* zwischen den zweiten und dritten Button und beobachten Sie das Ergebnis!

Im Projekt GeoMap → Kapitel 17.15.1 werden die Intentionen zum Beispiel 5 praktisch umgesetzt:



Abbildung 18.0.3.5.3: Projekt GeoMap