

20.7 Klassen Align und Arrange (gb.qt4)

Diese beiden Klassen werden Sie für die Anordnung (Arrangement) von Komponenten in einem Container oder für die Ausrichtung (Alignment) – zum Beispiel von Text in einer TextBox oder eines Bildes in einer PictureBox – einsetzen.

20.7.1 Klasse Align

Diese statische Klasse stellt Konstanten und 6 Methoden bereit, die für die Festlegung der Ausrichtung von Objekten oder Text benutzt werden, sofern diese über die Eigenschaft *Alignment* verfügen. Auch die *Draw.Text*-Methode nutzt die Konstanten der Klasse *Align*.

20.7.1.1 Ausrichtung von sichtbaren Objekten

Um die Ausrichtung von sichtbaren Objekten festzulegen, ist die Kenntnis der Richtungsmatrix genauso notwendig wie die dazugehörigen Konstanten für die ausgewiesenen Richtungen:

Align	Konstante	Symbol	Beschreibung
.BottomLeft	33	↙	Ausrichtung horizontal links und vertikal unten
.Bottom	35	↓	Ausrichtung horizontal zentriert und vertikal unten
.BottomRight	34	↘	Ausrichtung horizontal rechts und vertikal unten
.Left	1	←	Ausrichtung horizontal links und vertikal zentriert
.Center	3	※	Ausrichtung horizontal und vertikal zentriert
.Right	2	→	Ausrichtung horizontal rechts und vertikal zentriert
.TopLeft	17	↖	Ausrichtung horizontal links und vertikal oben
.Top	19	↑	Ausrichtung horizontal zentriert und vertikal oben
.TopRight	18	↗	Ausrichtung horizontal rechts und vertikal oben

Tabelle 20.7.1.1.1: Richtungsmatrix – Konstanten für die Eigenschaft Alignment

20.7.1.2 Ausrichtung von Text

Verwenden Sie die Ausrichtung von Text zum Beispiel beim Editor *TextEdit* zur Festlegung der Ausrichtung von Text, so beachten Sie die Abhängigkeit von der Schreibrichtung der verwendeten Sprache:

Align	Konstante	Beschreibung
.Normal	0	Ausrichtung vertikal zentriert und horizontal nach der Schreibrichtung
.TopNormal	16	Ausrichtung vertikal oben und horizontal nach der Schreibrichtung
.BottomNormal	32	Ausrichtung vertikal unten und horizontal nach der Schreibrichtung
.Justify	4	Ausrichtung horizontal und vertikal zentriert

Tabelle 20.7.1.2.1: Ausrichtung von Text nach der Schreibrichtung

20.7.1.3 Setzen und Auslesen der Werte der Eigenschaft Alignment

Es können alle Eigenschaftswerte aus den oberen Tabellen gesetzt werden oder mit den 6 Methoden *IsBottom*, *IsCenter*, *IsLeft*, *IsMiddle*, *IsRight* und *IsTop* ausgelesen werden, wie das Beispiel 1 zeigt:

```
Public Sub btnSetCenter_Click()
    If Not Align.IsCenter(PictureBox1.Alignment) Then
        PictureBox1.Alignment = Align.Center
    Endif ' Not Align.IsCenter(..)
End ' SetCenter
```

Beispiel 2

So setzen Sie die Text-Ausrichtung einer Textbox – analog auch einer MaskBox – und einer Textarea:

```

...
  TextBox1.Text = "Ausrichtung"
  TextBox1.Alignment = Align.Right
  TextAreal.Alignment = Align.Normal
...

```

Wenn Sie dagegen im Editor *TextEdit* den aktuellen Absatz oder den markierten Text ausrichten möchten, dann setzen Sie die Eigenschaft *TextEdit.Format.Alignment* mit den 4 Ausrichtungen Normal, Left, Center und Right:

```

Public Sub btnSetProperty_Click()
  TextEdit1.Format.Alignment = Align.Center
  TextEdit1.Format.Color = Color.Red
  TextEdit1.Format.Background = Color.LightGray
  TextEdit1.Format.Font = Font["FreeMono,14"]
End ' SetProperty

```

20.7.2 Klasse Arrange

Diese Klasse ist statisch. Sie stellt Konstanten bereit, die für die Festlegung der *Arrangement*-Eigenschaft vieler Komponenten benutzt werden können, falls diese über die Eigenschaft *Arrangement* verfügen. Nur Komponenten, die als Container für andere Komponenten wie zum Beispiel Form, Panel, Expander, DrawingArea1 oder TabStrip fungieren, verfügen über die Eigenschaft *Arrangement*. Die Angabe zur Anordnung von Komponenten ist immer relativ zum Container vorzunehmen.

Arrange	Konstante	Beschreibung
.None	0	Standardwert. Die Anordnung der Komponenten im Client-Bereich des Containers wird nicht beeinflusst.
.Horizontal	1	Die Komponenten im Container werden nebeneinander angeordnet und erhalten die <i>Höhe des Containers</i> . Ihre Breite wird nicht beeinflusst. Die Komponente im Container, deren Wert <i>Komponent.X</i> am kleinsten ist, wird links angezeigt. Sind diese Werte gleich, entscheidet die <i>Position in der Hierarchie</i> .
.Vertical	2	Die Komponenten im Container werden untereinander angeordnet und erhalten die Breite des Containers. Ihre Höhe wird nicht beeinflusst. Die Komponente im Container, deren Wert <i>Komponent.Y</i> am kleinsten ist, wird oben angezeigt. Sind diese Werte gleich, entscheidet die <i>Position in der Hierarchie</i> .
.Column .TopBottom	4	Die Komponenten werden im Container in einer Spalte <i>gestapelt</i> , sofern die Höhe ausreicht. Sonst wird eine weitere Spalte angelegt.
.Row .LeftRight	3	Die Komponenten werden im Container in einer Zeile angeordnet, sofern die Breite ausreicht. Sonst wird eine weitere Zeile unter der bestehenden angelegt.
.Fill	5	Eine Komponente im Container füllt den gesamten Client-Bereich im Container aus. Mehrere Komponenten machen keinen Sinn. Befinden sich dennoch mehrere Komponenten im Container, dann wird jene Komponente angezeigt, die in der <i>Hierarchie</i> – bezogen auf den Container – an <i>letzter</i> Stelle steht.

Tabelle 20.7.2.1: Konstanten für die Eigenschaft Arrangement

20.7.2.1 Setzen und Auslesen der Werte der Eigenschaft Arrangement

Sie können die in der Tabelle 20.7.2.1 angegebenen Werte für die Anordnung von Komponenten in einem Container verwenden. Bei der Anordnung von Komponenten in einem Container können Sie über die Container-Eigenschaften *Spacing* und *Padding* weitere Anpassungen vornehmen:

- Container.Padding
Der Integer-Wert (Pixel) bestimmt den Abstand einer Komponente vom Rand des Containers.
- Container.Spacing
Der Integer-Wert (Pixel) bestimmt den Abstand zwischen einzelnen Komponenten im Container.

Beispiel 1

Einem Panel1 werden drei Button hinzugefügt. Folgende Eigenschaften werden gesetzt:

- Panel1.W = 280
- Panel1.Spacing = 14
- Panel1.Padding = 7
- **Panel1. Arrangement = 1**
- Panel1.Border = 1 ' Plain



Panel1 zur Entwurfszeit

Abbildung 20.7.2.1.1: Design



Panel1. Arrangement = 1

Abbildung 20.7.2.1.2: Horizontal



Panel1. Arrangement = 3 (Row)

Abbildung 20.7.2.1.3: Zeile

Wenn man statt eines Panels einen anderen Container mit den drei Button verkleinert – sehr geeignet ist ein Fenster – dann kann man den Effekt der Anlage einer neuen Spalte gut sehen. Mit der Zuweisung `Panel1. Arrangement = Arrange.Row` erhält man die Anordnung in der Abbildung 20.7.1.3.

Beispiel 2

Auf dem Formular FMain wird eine HSplit-Komponente platziert. Mit der Anweisung

```
FMain.Arrangement = Arrange.Fill
```

füllt die Komponente HSplit die Fläche im Container Formular vollständig aus.