

# GizmoRBControls Hilfe

30. Juni 2013



Abbildung 1: **GizmoRBControls** Version 2.0.2

*Entschuldigen Sie mich für meine schlechte Beherrschung der deutschen Sprache.*

## Inhaltsverzeichnis

|   |          |
|---|----------|
| <b>1 Was ist die Verwendung von GizmoRBControls ?</b> | <b>3</b> |
| 1.1 Neu in Version 2.0.2                              | 5        |
| 1.2 Beschränkungen                                    | 5        |
| 1.3 Installation.                                     | 5        |
| 1.4 Verwendung  | 5        |
| 1.4.1 Localization                                    | 6        |
| 1.4.2 Wartung   | 6        |
| <b>2 Das PushButton TPB</b>                           | <b>6</b> |
| 2.1 Nützlichkeit und Umsetzung.                       | 6        |
| 2.2 Beispiel Usage:                                   | 6        |
| <b>3 Das PopUpMenu TPUMenu</b>                        | <b>7</b> |
| 3.1 Nützlichkeit und Umsetzung.                       | 7        |
| 3.2 Erholung der Wert.                                | 7        |
| <b>4 Das Label STText</b>                             | <b>7</b> |
| 4.1 Nützlichkeit und Umsetzung.                       | 7        |
| 4.2 Beispiel Usage:                                   | 8        |
| <b>5 Klass TFont</b>                                  | <b>8</b> |
| 5.1 Nützlichkeit und Umsetzung.                       | 8        |
| 5.2 Beispiel Usage:                                   | 8        |
| <b>6 Das Label STTextO</b>                            | <b>8</b> |
| 6.1 Nützlichkeit und Umsetzung.                       | 8        |
| 6.2 Beispiel Usage:                                   | 9        |
| <b>7 Container CEditNum.</b>                          | <b>9</b> |
| 7.1 Nützlichkeit und Umsetzung.                       | 9        |
| 7.2 Beispiel Usage:                                   | 10       |
| 7.3 Erholung der Wert:                                | 10       |

|           |                                  |           |
|-----------|----------------------------------|-----------|
| <b>8</b>  | <b>Container CombiEditNumStd</b> | <b>10</b> |
| 8.1       | Nützlichkeit und Umsetzung.      | 10        |
| 8.2       | Beispiel Usage:                  | 10        |
| 8.3       | Erholung der Wert:               | 11        |
| <b>9</b>  | <b>Container CombiEditNumUD</b>  | <b>12</b> |
| 9.1       | Nützlichkeit und Umsetzung.      | 12        |
| <b>10</b> | <b>Container CombiEditNumLR</b>  | <b>12</b> |
| <b>11</b> | <b>Container RoundButton</b>     | <b>13</b> |
| 11.1      | Nützlichkeit und Umsetzung.      | 13        |
| 11.2      | Beispiel Usage:                  | 13        |
| 11.3      | Erholung der Wert:               | 14        |
| <b>12</b> | <b>Container CSliderH</b>        | <b>14</b> |
| 12.1      | Nützlichkeit und Umsetzung.      | 14        |
| 12.2      | Beispiel Usage:                  | 15        |
| 12.3      | Erholung der Wert:               | 16        |
| <b>13</b> | <b>Container CSliderV</b>        | <b>16</b> |
| 13.1      | Nützlichkeit und Umsetzung.      | 16        |
| 13.2      | Beispiel Usage:                  | 17        |
| 13.3      | Erholung der Wert:               | 18        |
| <b>14</b> | <b>Dialog DlgDate</b>            | <b>18</b> |
| 14.1      | Nützlichkeit und Umsetzung.      | 18        |
| 14.2      | Erholung der Wert:               | 19        |
| 14.3      | Beispiel Usage:                  | 19        |

## Abbildungsverzeichnis

|    |  |    |
|----|--|----|
| 1  | <b>GizmoRBControls</b> Version 2.0.2   | 1  |
| 2  | <b>GizmoRBControls</b> mit Mac OS X  | 3  |
| 3  | <b>GizmoRBControls</b> mit Windows   | 3  |
| 4  | <b>GizmoRBControls</b> mit Linux   | 4  |
| 5  | Der Dialog Einstellungen können Sie wählen, die Sprache der Benutzeroberfläche: <b>Auto</b> →<br>platform Sprache. <b>Deutsch</b> , <b>Englisch</b> oder <b>Französisch</b> : Sprache aufbürend.   | 4  |
| 6  | ExportGizmoCRB 2.0.2 File Contents   | 5  |
| 7  | Instanz von PushButton abgeleitet von TPB (Mac, Windows, Linux).   | 6  |
| 8  | Instanz von PopUpMenu abgeleitet von TPUMenu (Mac, Windows, Linux).  | 7  |
| 9  | Instanz von Label abgeleitet von STText (Mac, Windows, Linux).   | 7  |
| 10 | CombiEditNum Standard mit und ohne Rahmen.   | 10 |
| 11 | CombiEditNumUD (UpDown geändert)   | 12 |
| 12 | CombiEditNumLR (LeftRight geändert) mit und ohne Rand.   | 13 |
| 13 | Container RoundButton. Sie können die Winkel mit der Kontrolle UpDownArrow, die<br>runde Taste mit der Maus oder durch Anklicken der Schaltfläche. Der Aspekt der Kontrolle<br>können eingestellt werden und der Benutzer kann eigene Logos erstellen. | 13 |
| 14 | Container <b>CSliderH</b> . Sie können statt der Indikator-Wert (1000 \$ oder 100,0 mm) oben<br>(kGradUp) oder unten (kGradDown) des Cursor.   | 14 |
| 15 | Container <b>CSliderV</b> . Sie können statt der Indikator-Wert (1000 \$ oder 100,0 mm) nach<br>links (kGradLeft) oder rechts (kGradRight) des Cursor.   | 16 |
| 16 | Dialog DlgDate. Beachten Sie die beiden Schaltflächen am unteren Rand des Dialogs, um<br>eine <b>Kopie</b> oder eine <b>Paste</b> machen.  | 18 |

# 1 Was ist die Verwendung von GizmoRBControls ?

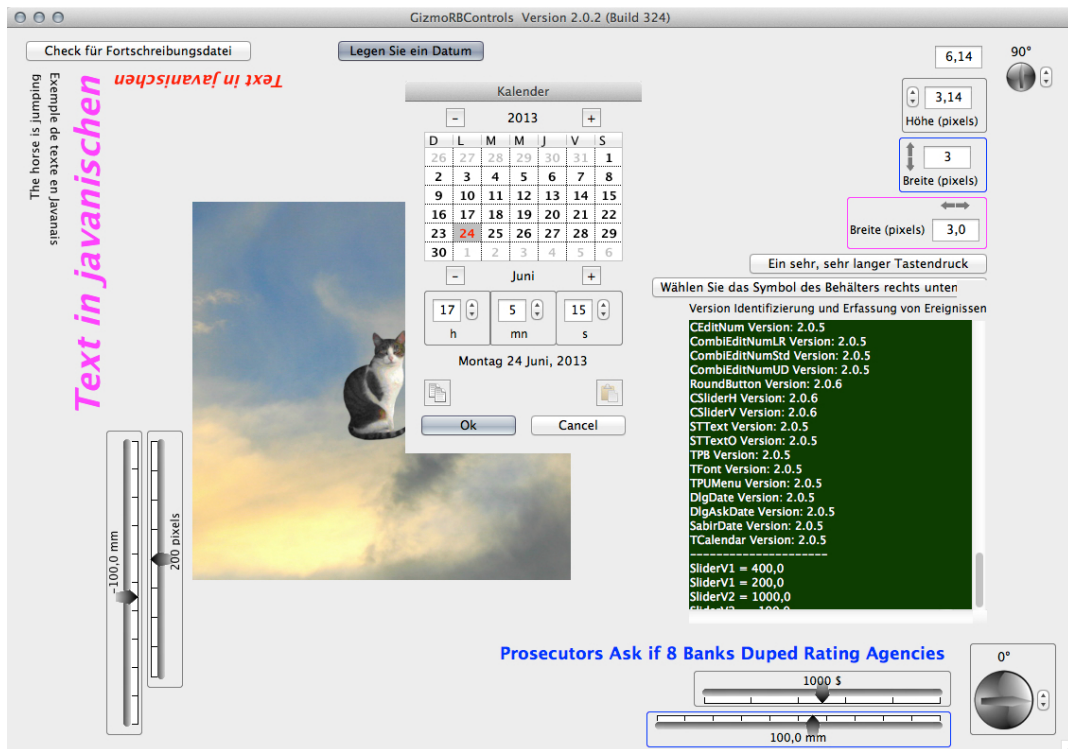


Abbildung 2: GizmoRBControls mit Mac OS X

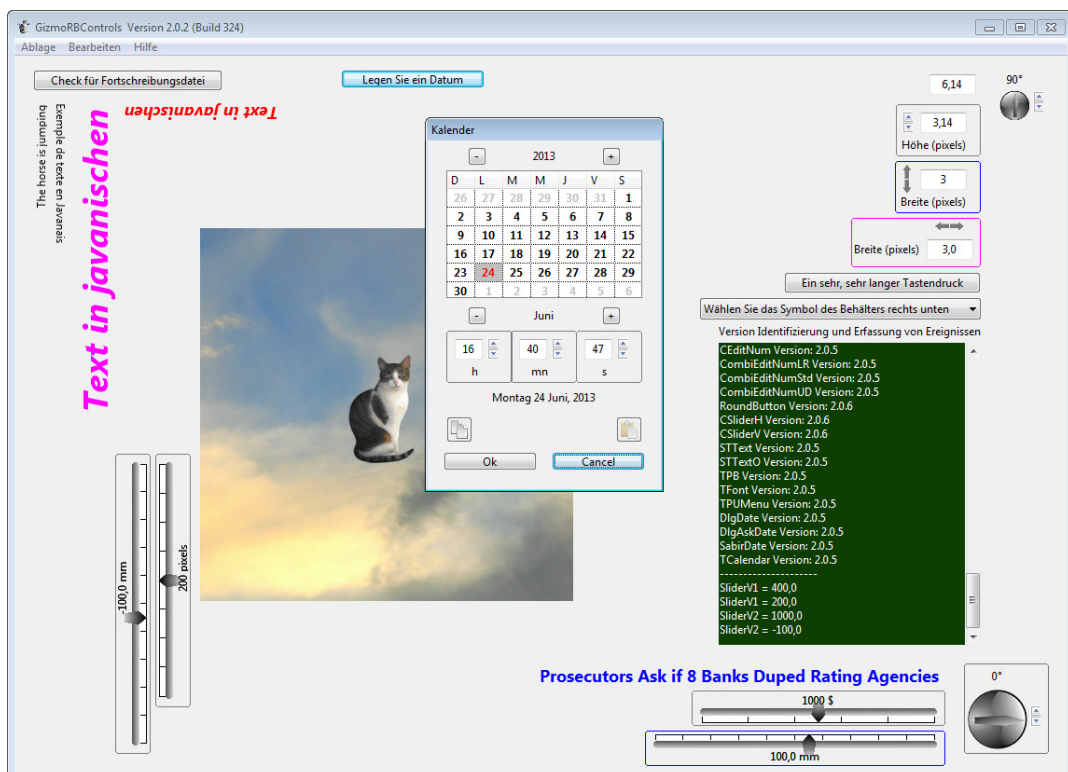


Abbildung 3: GizmoRBControls mit Windows

Das Programm beschreibt die Verwendung einer Reihe von **Klassen**, **Kontrolle** und **Container** anwenbar mit **RealBasic/Xojo**. Diese Komponenten sind verfügbar für **Macintosh**, **Windows** und **Linux** wie in den Abbildungen [2,3,4] gezeigt.

**GizmoRBControls** ist ein Freeware und ist open source. Sie haben Quellen der Kontrolle und Container (befindet sich im Ordner **ExportGizmoCRB 2.0.2** abbildung [6]) dass Sie frei sind zu ändern.

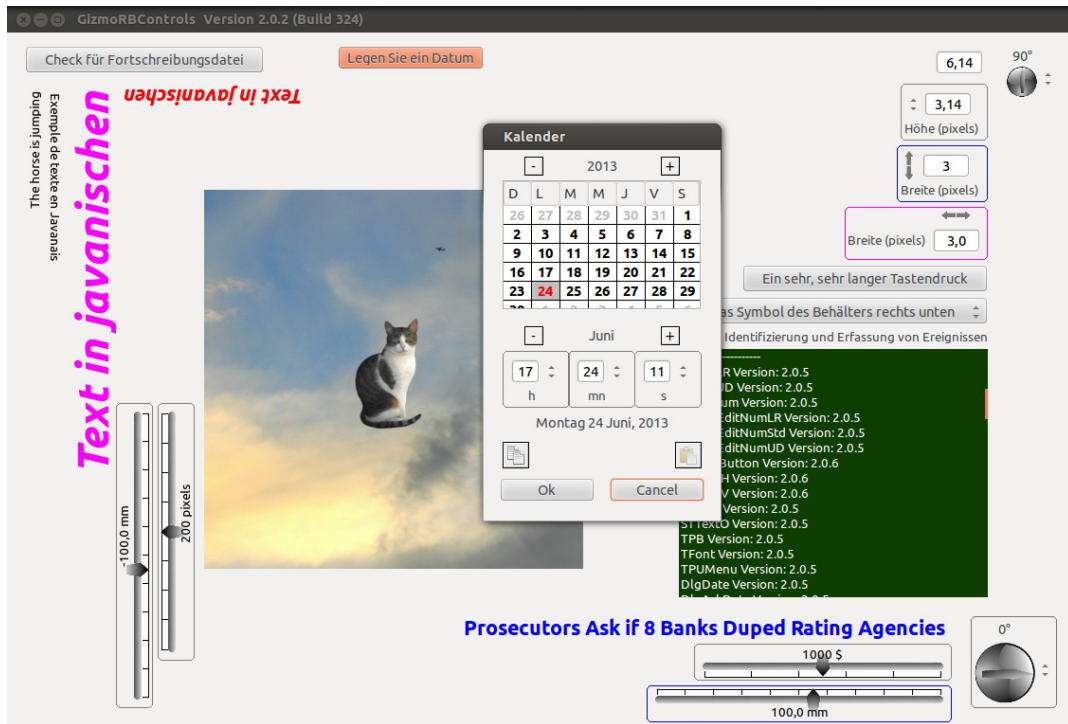


Abbildung 4: **GizmoRBControls** mit Linux

Die Kontrollen in den Interface von **GizmoRBControls** sind aktiv. Sie können ihnen Verhalten nachprüfen in das Textfeld ein . Mit Eine vertikale Gleitfläche können Sie das Bild von Gizmo die Katze in ein Canvas bewegen.



Abbildung 5: Der Dialog Einstellungen können Sie wählen, die Sprache der Benutzeroberfläche: **Auto**→  
platform Sprache. **Deutsch** , **Englisch** oder **Französisch**: Sprache aufbürdend.

## 1.1 Neu in Version 2.0.2

Dieses Release enthält einen neuen Dialog (**DlgDate**) für die Auswahl eines Datums.

Es hat viele Fehler korrigiert.

Wir vereinfachte Verwaltung der Sprache und deutscher Sprache hinzugefügt.

Da die Elemente von **ExportGizmoCRB 2.0.2** vom Benutzer geändert werden, führten wir eine Basisversion Kontrolle (siehe [1.4.2]).

Dieses Programm wurde mit der neuen Version **Xojo** von REALbasic getestet. Die vorgesehenen Kontrollen sind **kompatiblen PowerPC, Intel, Macintosh/Carbon, MacIntosh/Cocoa, Windows XP, Windows 8, Linux (Ubuntu Version von 8 bis 12.X)**.

## 1.2 Beschränkungen

Die Klassen und Kontrolle sind von allen Versionen von REALbasic nutzbaren. Im Gegensatz dazu sind nutzbar die Container *nur in den Pro-Versionen von REALbasic*.

## 1.3 Installation.

Auf **Macintosh** enthält die Disk-Image **GizmoRBControls.dmg** das Programms **GizmoRBControls.app** (in den **Programme-Ordner** platziert werden) und Datei **ExportGizmoCRB 2.0.2** (platziert werden, zum Beispiel, die in der Dokumentation Ordner), deren Inhalt is gezeigt Abbildung [figa]. Die darin enthaltenen Dateien sind entweder Klassen oder Kontrollen (extension **.rbo**) oder Container (**.rbw**) oder Bilder (**.png**).

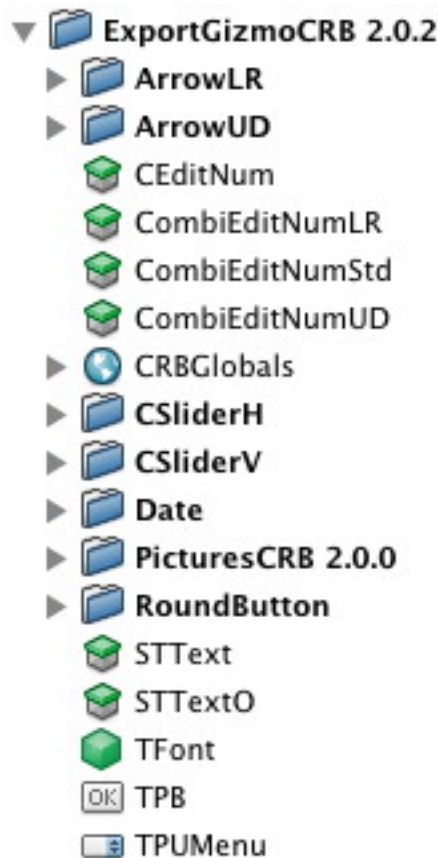


Abbildung 6: ExportGizmoCRB 2.0.2 File Contents

## 1.4 Verwendung

So verwenden Sie die Elemente in den **ExportGizmoCRB 2.0.2**-Ordner (siehe Abbildung [6]), einfach per Drag & Drop diese Artikel in der Komponente **Entwurf EDI** (Environments Integrierte Entwicklungsumgebung) Ihres Programms **RealBasic/Xojo**. Die folgende Dokumentation erfahren Sie, was Sie nach dem Ziel zu importieren.

### 1.4.1 Localization

Einige Module Informationen an oder hängen von der verwendeten Sprache (mit den Elementen **CEditNum** und **SabirDate**). Es ist wichtig, diese Informationen zu finden. Zu diesem Zweck ist es ratsam, dass Sie das Modul definiert **SabirDate** drei Konstanten verwenden:

- English = 0
- French = 1
- German = 2

und eine variable (*Langage* As Integer).

Dann einfach, bevor sie eine Verwendung von Steuerungen, Container oder Dialog zu **GizmoRB-Controls** *Language = Englisch, Französisch oder Deutsch* zu definieren. Es ist möglich, andere Sprachen erstellen, indem Sie das Modul **SabirDate** *SetSabirDate* des Verfahrens einzuführen oder *Setlangage* Verfahren von **CEditNum**.

### 1.4.2 Wartung

Um die Wartung zu erleichtern **CRBGlobals** Modul enthält alle Versionsnummern der Elemente. Jedes Element enthält auch eigene Versionsnummer. Debug-Modus werden diese Zahlen im Vergleich zu Ihnen zu einem Widerspruch aufmerksam: Wenn die Versionsnummer des Elements ist kleiner als die Zahl in **CRBGlobals** enthalten, benachrichtigt Sie das Programm, dass das Element verwendet veraltet ist.

Es ist dein Job zu halten diese Versionsnummern, um sicherzustellen, (indem sie immer die aktuellste Version Ihres Programms **CRBGlobals** in Entwicklungsländern) die Konsistenz mit **ExportGizmo-CRB 2.0.2**.

## 2 Das PushButton TPB

### 2.1 Nützlichkeit und Umsetzung.



Abbildung 7: Instanz von PushButton abgeleitet von TPB (Mac, Windows, Linux).

Der Wert dieser Art von Button ist, dass seine Breite und Höhe automatisch an die Breite und Höhe des Textes anzupassen. Es ist eine nützliche Eigenschaft für die Tasten mit verschiedenen Standorten oder in verschiedenen Systemen verwendet.

Per Drag & Drop die Datei TBP.rbo, von dem **ExportGizmoCRB 2.0.2**-Ordner (siehe Abbildung [6] in den Entwurf Ihres Programms.

Lassen TPB1 eine Instanz der **TPB**. Es hat zwei Funktionen:

- **TPB1.Open**: Diese Funktion passt automatisch die Größe der Schaltfläche unter dem Text in der Kontrollgruppe zur Design-Zeit gesetzt.
- **TPB1.SetText(S As String)**: équivalent zu `TPB1.Caption = S`, aber die Größe der Schaltfläche unterhalb der Textlänge. Die Eigenschaften Textfont, TextSize, Fett, Kursiv, Unterstrichen und LockRight Bühnenbild sind erhalten.

### 2.2 Beispiel Usage:

```
' TPB -----  
TPB1.SetText("A very very long button")  
' Positionierung -----  
TPB1.Top = CombiEditNumLR1.Top + CombiEditNumLR1.Height + 5  
TPB1.Left = Width - 20 - TPB1.Width
```

### 3 Das PopUpMenu TPUMenu

#### 3.1 Nützlichkeit und Umsetzung.



Abbildung 8: Instanz von PopUpMenu abgeleitet von TPUMenu (Mac, Windows, Linux).

Das Interesse an dieser Art von PopUpMenu ist die Breite und Höhe automatisch an die Breite und Höhe des Textes anzupassen. Es ist eine nützliche Eigenschaft für PopUpMenus mit verschiedenen Standorten oder in verschiedenen Systemen verwendet.

Per Drag & Drop die Datei TPUMenu.rbo, von dem **ExportGizmoCRB 2.0.2**-Ordner (siehe Abbildung [6]) in den Entwurf Ihres Programms.

Lassen TPUMenu1 eine Instanz der **TPUMenu**. Es hat die Funktion:

- **TPUMenu1. SetPopup(StrList As String, Index As Integer)**: équivallent zu

```
for i = 1 to CountFields(StrList, ",")
  TPUMenu1.AddRow = NthField(StrList, ',', ', ', i)
next i
```

aber die Größe des Menü entlang der Länge der Zeilen Text. Die Eigenschaften Textfont, TextSize, Fett, Kursiv, Unterstrichen und LockRight Bühnenbild sind erhalten.

```
'TPUMenu -----
TPUMenu1.SetPopup("Choix de l'icone du Container en bas à droite,RButtonYingYang,
RButtonBlue,RButtonBlueButterfly,RButtonWithArrow,Mythomaniac,Psychopathe",0)
' Positionierung -----
TPUMenu1.Top = TPB1.Top + TPB1.Height + 5
TPUMenu1.Left = Width - 20 - TPUMenu1.Width
```

#### 3.2 Erholung der Wert.

Nutzen Sie einfach die Veranstaltung **Change** die Instanz TPUMenu1 von **TPUMenu**.

### 4 Das Label STText

#### 4.1 Nützlichkeit und Umsetzung.



Abbildung 9: Instanz von Label abgeleitet von STText (Mac, Windows, Linux).

Das Interesse an dieser Art von Label ist die Breite und Höhe automatisch an die Breite und Höhe des Textes anzupassen. Es ist eine nützliche Eigenschaft für PopUpMenus mit verschiedenen Standorten oder in verschiedenen Systemen verwendet.

Per Drag & Drop die Datei STText.rbw, von dem **ExportGizmoCRB 2.0.2**-Ordner (siehe Abbildung [6]) in den Entwurf Ihres Programms.

Lassen **STText1** eine Instanz der **STText**. Es hat die Funktion:

- `STText1.SetText(S As String)` Größen der Label basierend auf dem Inhalt von `S`.

## 4.2 Beispiel Usage:

```
'STText -----
  STText1.SetText("Prosecutors Ask if 8 Banks Duped Rating Agencies")
' Positionierung -----
  STText1.Left = RoundButton2.Left - STText1.Width-20
  STText1.Top = RoundButton2.Top
```

## 5 Klass TFont

### 5.1 Nützlichkeit und Umsetzung.

Diese Klasse hat den Vorteil, dass das Vorbringen von Schriftart mehr kompakt weiterzugeben. Es ist in den lenkbaren Label `STTextO` verwendet.

Per Drag & Drop die Datei `TFont.rbo`, von dem **ExportGizmoCRB 2.0.2**-Ordner (siehe Abbildung [6] in den Entwurf Ihres Programms.

Klass-Eigenschaften:

- **Name As String**: den Namen der Schriftart.
- **Size As Integer**: Schriftgröße.
- **Bold As Boolean**.
- **Italic As Boolean**.
- **Underline As Boolean**.

Klasse-Funktion `Init`:

```
Sub Init
  Name = "System"
  Size = 0
  Bold = false
  Italic = false
  Underline = false
```

### 5.2 Beispiel Usage:

```
'TFont -----
  Dim Fonte As TFont

  Fonte = new TFont
  Fonte.Init
  ...
```

## 6 Das Label STTextO

### 6.1 Nützlichkeit und Umsetzung.

Das Interesse an dieser Art von Label ist die Breite und Höhe automatisch an die Breite und Höhe des Textes anzupassen. Es ist eine nützliche Eigenschaft für PopupMenus mit verschiedenen Standorten oder in verschiedenen Systemen verwendet.

Auch das Layout von Text kann gedreht werden, um  $0^0$ ,  $90^0$ ,  $180^0$  oder  $-90^0$ . Siehe Abbildungen [??,3,4].

Per Drag & Drop die Datei `STTextO.rbw`, von dem **ExportGizmoCRB 2.0.2**-Ordner (siehe Abbildung [6] in den Entwurf Ihres Programms.

Lassen `STTextO1` eine Instanz der `STTextO`. Es hat die Funktionen:



- **STText.SetText(S As String,Angle As Integer)** Label skaliert auf die Inhalte der **S** und die mit der Richtung der durch den Winkel **Angle**. **Angle** ist als in Grad angerechnet, positiv (gegen den Uhrzeigersinn drehen). Nur die Winkel  $0^0$ ,  $90^0$ ,  $180^0$  und  $-90^0$  sind willkommen.
- **STText.SetText(S As String,Angle As Integer, AColor As Color)** Label skaliert auf die Inhalte der **S** und die mit der Richtung der durch den Winkel **Angle** und mit der Farbe von **AColor** gegeben.
- **STText.SetText(S As String,Angle As Integer, AColor As Color, Fonte As TFont)** Label skaliert auf die Inhalte der **S** und die mit der Richtung der durch den Winkel **Angle** und, mit der Farbe von **AColor** gegeben und SchriftArt gegeben durch **Fonte**(see Klass **TFont** [5]).

## 6.2 Beispiel Usage:

```
'STText0 -----
  STText01.SetText("The horse is jumping",90)
' Positionierung -----
  STText01.Left = 20
  STText01.Top = PBCheckUpdate.Top + PBCheckUpdate.Height + 10
' Ein weiteres Beispiel -----
  Dim Fonte As TFont

  Fonte = new TFont
  Fonte.Init
  Fonte.Size = 36
  Fonte.Bold = true
  Fonte.Italic = true
  STText03.SetText("Text in javanischen",90,&cFF0000,Fonte)
' Positionierung -----
  STText03.Left = STText02.Left + STText02.Width + 5
  STText03.Top = STText01.Top
```

## 7 Container CEditNum.

### 7.1 Nützlichkeit und Umsetzung.

Mit dieser Container können Sie Integer oder reale Zahlen ausfiltern. Sie können auch einen Mindest- und / oder einen maximalen Wert von Zahlen eingeben.

Per Drag & Drop die Datei **CEditNum.rbw**, von dem **ExportGizmoCRB 2.0.2**-Ordner (siehe Abbildung [6] in den Entwurf Ihres Programms.

So initialisieren Sie es, nutzen Sie einfach die Funktion **Resize**, die seiner Größe initialisiert und Funktion **SetValue** initialisiert seine Content:

- **CEditNum1.Resize(Marge As Integer, TextFieldWidth As Integer)**. **Marge** ist die Seitenrand rund der **TextField** und **TextFieldWidth** die Breite des **TextField**.
- **CEditNum1.SetValue(X)**. Wenn **X** ist ein Integer, können wir nur ganze Zahlen eintreten. Wenn **X** ist double, können wir nur integer oder real Zahlen eintreten.
- Im Fall der reellen Zahlen, des Formats kann durch Angabe eingestellt werden **AFormat** : **CEditNum.AFormat = ###.##"** (zum Beispiel).

Es ist auch möglich, eine minimale oder maximale Zahl eingeben gesetzt werden:

- **CEditNum1.SetMinValue(5.2)** auf den minimalen Wert angeben.
- **CEditNum1.SetMaxValue(15.0)** auf den maximalen Wert angeben.
- **CEditNum1.SetLangage** um die Sprache der Warnmeldungen im Falle einer Überschreitung der max min oder angeben (Sichen [8.2]).

## 7.2 Beispiel Usage:

```
' CEditNum -----  
CEditNum1.Resize(3,50) ' Marge As Integer, TextFieldWidth As Integer  
CEditNum1.SetValue(6.14)  
CEditNum1.SetMinValue(5.2)  
CEditNum1.SetMaxValue(15.0)  
CEditNum1.SetLangage  
' Positionierung -----  
CEditNum1.Top = RoundButton1.Top + RoundButton1.Height + 5  
CEditNum1.Left = Width - 20 - CEditNum1.width
```

## 7.3 Erholung der Wert:

Nutzen Sie einfach die Veranstaltung **ValueChanged** die Instanz **CEditNum1** von **CEditNum** oder bei die Funktion **GetValue**.

- **CEditNum1.GetValue As double**: Gibt den numerischen Wert des TextField.
- **CEditNum1.ValueChanged(X As double)**: X st der numerische Wert des TextField.

## 8 Container CombiEditNumStd

### 8.1 Nützlichkeit und Umsetzung.

Mit dieser Container können Sie Integer oder reale Zahlen ausfiltern. Sie können auch einen Mindest- und / oder einen maximalen Wert von Zahlen eingegeben. Darüber hinaus hat es Pfeile, um die aktive Eingabe der Nummer zu ändern (siehe Abbildung voir figure [10]). Die Steuerung kann wahlweise mit einem Rahmen umgeben sein.

Der Container enthält:

- Ein Container **CEditNum**.
- Ein Kontrolle **STText** Für Legende.
- Ein Kontrolle **UpDownArrow**.

Per Drag & Drop die Datei **CombiEditNumStd.rbw**, von dem **ExportGizmoCRB 2.0.2**-Ordner (siehe Abbildung [6] in den Entwurf Ihres Programms.

Wie in Abbildung 10 Container **CombiEditNumStd** gezeigt, verbindet eine **CEditNum**, ein **Bildunterschrift** und ein Kontrolle **UpDownArrow** Standard.



Abbildung 10: CombiEditNum Standard mit und ohne Rahmen.

### 8.2 Beispiel Usage:

```
' CombiEditNumStd -----  
CombiEditNumStd1.DispoLeg = CombiEditNumStd1.kLegBottom  
CombiEditNumStd1.DispoArrow = CombiEditNumStd1.kArrowLeft  
CombiEditNumStd1.SetText("Hauteur relative (pixels)")  
  
CombiEditNumStd1.Resize(3,50,&c7F7F7F)  
CombiEditNumStd1.SetValue(6.0)  
CombiEditNumStd1.SetMinValue(5.2)
```

```

CombiEditNumStd1.SetMaxValue(15.0)
CombiEditNumStd1.SetLangage
CombiEditNumStd1.SetStep(1.0)
' Positionierung -----
CombiEditNumStd1.Top = CEditNum1.Top + CEditNum1.Height + 5
CombiEditNumStd1.Left = Width - 20 - CombiEditNumStd1.Width

```

Die Variablen und DispoLeg und DispoArrow fix das Layout der Elemente:

- **CombiEditNumStd1.kLegTop**: Bildunterschrift über dem Textfield.
- **CombiEditNumStd1.kLegLeft**: Bildunterschrift zu dem der TextField links.
- **CombiEditNumStd1.kLegRight**: Bildunterschrift auf der rechten Seite des TextField.
- **CombiEditNumStd1.kLegBottom**: Bildunterschrift unter dem Textfield.
- **CombiEditNumStd1.kArrowLeft**: contrôle UpDownArrow zu dem der TextField links.
- **CombiEditNumStd1.kArrowRight**: contrôle UpDownArrow auf der rechten Seite des TextField.

Beachten Sie, dass dies die Position der Legende, die Priorität hat.  
Die Funktionen:

- **CombiEditNumStd1.SetText("Hauteur relative (pixels)")** es sichert die Bildunterschrift.
- **CombiEditNumStd1.Resize(Marge As Integer, TextFieldWidth As Integer, BorderColor As Color)** Marge ist die Seitenrand rund der TextField, TextFieldWidth die Breite des TextField und Farbe des Rahmens um die Kontrolle. **Wenn die Farbe weggelassen wird, ist die Grenze nicht gezogen.**
- **CombiEditNumStd1.SetValue(Value As Integer/double)** Es legt den Anfangswert der TextField.
- **CombiEditNumStd1.SetValue(Value As Integer/double, AFormat As String)** Es legt den Anfangswert der TextField. Das FehlWert Format für real ist `###.0##`.
- **CombiEditNumStd1.SetMinValue(MinVal As Integer/double)** (fakultativ) auf den minimalen Wert angeben.
- **CombiEditNumStd1.SetMaxValue(MaxVal As Integer/double)** (fakultativ) auf den maximalen Wert angeben.
- **CombiEditNumStd1.SetLangage** um die Sprache der Warnmeldungen im Falle einer Überschreitung der max min oder angeben. Die Funktion SetLangage hängt (bis jetzt) 3 globalen Konstanten: **English** = 0, **French** = 1 und **German** = 2 und die globale Variable **Langage**, die eine diese drei Werte übernimmt.
- **CombiEditNumStd1.SetStep(Step As Integer/double)** Es setzt den Wert von der Steuerung UpDownArrow verwendet, um die TextField variieren.

Beachten Sie, dass Anrufe auf **SetMinValue** und **SetMaxValue** sollte nach **SetValue** platziert werden, SetValue bricht die min und max.

### 8.3 Erholung der Wert:

Nutzen Sie einfach die Veranstaltung **ValueChanged** die Instanz CombiEditNumStd1 von **CombiEditNumStd** oder bei die Funktion **GetValue**.

- **CombiEditNumStd1.GetValue As double**: Gibt den numerischen Wert des TextField.
- **CombiEditNumStd1.ValueChanged(X As double)**: X st der numerische Wert des TextField.

## 9 Container CombiEditNumUD

### 9.1 Nützlichkeit und Umsetzung.

Abgesehen von der Grafik UpDown Pfeile, den Betrieb dieser Steuerung ist identisch mit dem Container **CombiEditNumStd** [8].

Dieser Container enthält (siehe Abbildung [11]):

- Ein container **CEditNum**.
- Ein Kontrolle **STText** pour la légende.
- Ein container **ArrowUD**.
- Ein Bild mit einer Maske (type PNG): **ArrowUpN12x15**.
- Ein Bild mit einer Maske (type PNG): **ArrowUpG12x15**.
- Ein Bild mit einer Maske (type PNG): **ArrowDownN12x15**.
- Ein Bild mit einer Maske (type PNG): **ArrowDownG12x15**.

Per Drag & Drop die Datei CombiEditNumUD.rbw and die Bilder oben, von dem **ExportGizmoCRB 2.0.2**-Ordner (siehe Abbildung [6] in den Entwurf Ihres Programms.



Abbildung 11: CombiEditNumUD (UpDown geändert)

## 10 Container CombiEditNumLR

Abgesehen von der Grafik UpDown Pfeile, den Betrieb dieser Steuerung ist identisch mit dem Container **CombiEditNumStd** [8].

Dieser Container enthält (siehe Abbildung [12]):

- Ein container **CEditNum**.
- Ein Kontrolle **STText** pour la légende.
- Ein container **ArrowLR**.
- Ein Bild mit einer Maske (type PNG): **ArrowLeftN12x15**.
- Ein Bild mit einer Maske (type PNG): **ArrowLeftG12x15**.
- Ein Bild mit einer Maske (type PNG): **ArrowRightN12x15**.
- Ein Bild mit einer Maske (type PNG): **ArrowRightG12x15**.

Per Drag & Drop die Datei CombiEditNumLR.rbw and die Bilder oben, von dem **ExportGizmoCRB 2.0.2**-Ordner (siehe Abbildung [6] in den Entwurf Ihres Programms.

Les variables DispoLeg et DispoArrow fixent la disposition des éléments:

- **CombiEditNumStd1.kLegTop**: Bildunterschrift über dem Textfield.
- **CombiEditNumStd1.kLegLeft**: Bildunterschrift zu dem der TextField links.
- **CombiEditNumStd1.kLegRight**: Bildunterschrift auf der rechten Seite des TextField.
- **CombiEditNumStd1.kLegBottom**: Bildunterschrift unter dem Textfield.



Abbildung 12: CombiEditNumLR (LeftRight geändert) mit und ohne Rand.

- **CombiEditNumStd1.kArrowTop**: Kontrolle LeftRightArrow über dem Textfield.
- **CombiEditNumStd1.kArrowBottom**: Kontrolle LeftRightArrow unter dem Textfield.

Beachten Sie, dass dies die Position der Legende, die Priorität hat.

## 11 Container RoundButton

### 11.1 Nützlichkeit und Umsetzung.

Diese Container können wählen, einen Winkel zwischen  $0^0$  und  $360^0$ . Es setzt den Wert entweder mit der Kontrolle UpDownArrow oder durch Klicken auf die Schaltfläche Symbol entlang der Wert. Der Button-Symbol können modifiziert werden, wie in Abbildung [13] dargestellt werden.

Per Drag & Drop die Position der **RoundButton.rbw ExportGizmoCRB 2.0.2** (siehe Abbildung [6]) in das Projekt des Programms, sowie Bilder *RButtonYingYang.png*, *RButtonBlue.png*, *RButtonBlueButterfly.png* et *RButtonWithArrow.png*.

Dieser Container enthält [13]:

- Ein Bild mit einer Maske (type PNG): *RButtonYingYang.png*, *RButtonBlue.png*, *RButtonBlueButterfly.png* or *RButtonWithArrow.png*.
- Ein canvas: **Cnvs**.
- Ein Kontrolle UpDownArrow: **UDRot**.
- Ein Label: **STDegre**.



Abbildung 13: Container RoundButton. Sie können die Winkel mit der Kontrolle UpDownArrow, die runde Taste mit der Maus oder durch Anklicken der Schaltfläche. Der Aspekt der Kontrolle können eingestellt werden und der Benutzer kann eigene Logos erstellen.

### 11.2 Beispiel Usage:

```
' RoundButton -----
RoundButton1.Resize(48,3,0)
' Positionierung -----
RoundButton1.Top = 14
RoundButton1.Left = Width - 20 - RoundButton1.Width
```

On dispose de 3 fonctions pour l'initialisation:

- **RoundButton1.Resize(ButtonSize As Integer,Margin As Integer,InitialAngle As Integer)** ButtonSize bestimmt die Größe der Schaltfläche (32 bis 128 Pixel), Margin die Marge um die Container und InitialAngle die anfängliche Winkelwert.

- **RoundButton1.Resize(ButtonSize As Integer,Margin As Integer,InitialAngle As Integer, PictureName As String)** ButtonSize bestimmt die Größe der Schaltfläche (32 bis 128 Pixel), Margin die Marge um die Container, InitialAngle die anfängliche Winkel und PictureName die Wahl der Schaltfläche Bild. Standardmäßig **PictureName** ist **RButtonYingYang**). Es hat die folgenden Werte:
  - "RButtonYingYang"
  - "RButtonBlue"
  - "RButtonBlueButterfly"
  - "RButtonWithArrow"

- **RoundButton1.Resize(ButtonSize As Integer,Margin As Integer,InitialAngle As Integer, PictureName As String, AColor As Color)** ButtonSize bestimmt die Größe der Schaltfläche (32 bis 128 Pixel), Margin die Marge um die Container, InitialAngle die anfängliche Winkel; PictureName die Wahl der Schaltfläche Bild und AColors, Rahmen um die Kontrolle mit der Wahl der Farbe.

### 11.3 Erholung der Wert:

Nutzen Sie einfach die Veranstaltung **ValueChanged** die Instanz RoundButton1 von **RoundButton** oder bei die Funktion **GetValue**.

- **RoundButton1.GetValue As double:** Gibt den numerischen Wert des Winkels im Bogenmaß. Die Winkel sind entgegen dem Uhrzeigersinn gezählt.
- **RoundButton1.ValueChanged(X As double):** X ist der numerische Wert des Winkels im Bogenmaß

## 12 Container CSliderH

### 12.1 Nützlichkeit und Umsetzung.

Diese Container können Sie einen Wert zwischen einem Minimalwert und Maximalwert, indem Sie den Cursor entlang der **horizontalen** Lineal mit der Maus auswählen[14]. Es kann auch, wenn der Behälter zu konzentrieren, erhöhen Sie den Wert durch tippen der **Schaltfläche +** und dem Rückgang bei tippen der **Schaltfläche -**. Der Cursor wird blau, wenn *der Container den Fokus hat*.

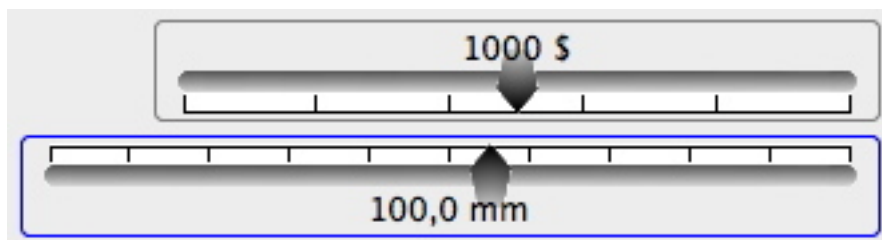


Abbildung 14: Container **CSliderH**. Sie können statt der Indikator-Wert (1000 \$ oder 100,0 mm) oben (kGradUp) oder unten (kGradDown) des Cursor.

Dieser Container enthält:

- Ein Bild mit einer Maske (type PNG): **LineLeft**.
- Ein Bild mit einer Maske (type PNG): **LineHorM**.
- Ein Bild mit einer Maske (type PNG): **LineRight**.
- Ein Bild mit einer Maske (type PNG): **PointeurUp**.
- Ein Bild mit einer Maske (type PNG): **PointeurUpOn**.
- Ein Bild mit einer Maske (type PNG): **PointeurDown**.
- Ein Bild mit einer Maske (type PNG): **PointeurDownOn**.

- Ein canvas: **Cnvs**.

Per Drag & Drop die Position der `CSliderH.rbw ExportGizmoCRB 2.0.2` (siehe Abbildung [6]) in das Projekt des Programms, und die Bilder oben gezeigt.

## 12.2 Beispiel Usage:

```
' CSliderH -----
CSliderH1.Resize(250,500,1500,"$",5,CSliderH1.kGradUp,&c7F7F7F)
' Arguments: Länge der Schubstange,Mini, Maxi,Einheit,Anzahl der Ticks,Position der Indikatorwert,Fo
' Startwert -----
CSliderH1.SetValue(1000)
' Positionierung -----
CSliderH1.Left = RoundButton2.Left - CSliderH1.Width - 20
CSliderH1.Top = RoundButton2.Top + (RoundButton2.Height - CSliderH1.Height)/2
```

Es verfügt über 6 Funktionen zur Initialisierung, [3 für reelle Werte \(double\)](#):

- **CSliderH1.Resize(L As Integer, MinR As double, MaxR As double, Unity As String, NTics As Integer, PosLeg As Integer)**. Zu definieren einen Horizontal SchiebeRegler.
  - **L** = Länge der Schubstange (in Pixeln).
  - **MinR** = Mindestwert (double).
  - **MaxR** = maximal zulässige Wert (double).
  - **Unity** = Werte der Einheit (zB mm, \$, etc..).
  - **NTics** = Anzahl der Ticks: das Intervall MinR, MaxR in NTics unterteilt.
  - **PosLeg** = Position der Indikatorwert. Siehe unten für Werte PosLeg.
- **CSliderH1.Resize(L As Integer, MinR As double, MaxR As double, Unity As String, NTics As Integer, PosLeg As Integer, AFormat As String)** Zu definieren einen Horizontal SchiebeRegler. Argumente identisch mit denen der oben genannten Funktionen, plus ein neues Argument:
  - **AFormat** = Zahlenformat für die Indikator-Wert.  
Standardmäßig, AFormat = -###.0##.
- **CSliderH1.Resize(L As Integer, MinR As double, MaxR As double, Unity As String, NTics As Integer, PosLeg As Integer, AFormat As String, AColor As Color)** Zu definieren einen Horizontal SchiebeRegler. Argumente identisch mit denen der oben genannten Funktionen, plus ein neues Argument:
  - **AColor**: Wenn dieses Argument vorhanden ist, wird ein Rahmen um den Container gezogen.

und [2 für Integer-Werte](#):

- **CSliderH1.Resize(L As Integer, MinI As Integer, MaxI As Integer, Unity As String, NTics As Integer, PosLeg As Integer)**
- **CSliderH1.Resize(L As Integer, MinI As Integer, MaxI As Integer, Unity As String, NTics As Integer, PosLeg As Integer, AColor As Color)**

*Der einzige Unterschied bei den Funktionen für reelle Werte ist, dass hier die Argumente [MinI et MaxI](#) sind Integer Zahlen und das Argument [AFormat](#) ist gegangen.*

### Werte der Konstanten **PosLeg (Horizontal SchiebeRegler)** :

- **CSliderH1.kGradUp**: Indikator-Wert über den Schieberegler.
- **CSliderH1.kGradDown**: Indikator-Wert unterhalb des Schiebereglers.

Auch können Sie den Startwert mit der Funktion **SetValue** gesetzt :

- **CSliderH1.SetValue(X As Integer/double)**: use **nach** der Funktion **Resize(...)**.

### 12.3 Erholung der Wert:

Nutzen Sie einfach die Veranstaltung **ValueChanged** die Instanz **CSliderH1** von **CSliderH** oder bei die Funktion **GetValue**.

- **CSliderH1.GetValue As double**: Gibt den Wert des Wert Indikator.
- **CSliderH1.ValueChanged(X As double)**:X ist des Wert des Indikator Wert.

## 13 Container CSliderV

### 13.1 Nützlichkeit und Umsetzung.

Diese Container können Sie einen Wert zwischen einem Minimalwert und Maximalwert, indem Sie den Cursor entlang der **vertikalen** Lineal mit der Maus auswählen[15]. Es kann auch, wenn der Behälter zu konzentrieren, erhöhen Sie den Wert durch tippen der **Schaltfläche +** und dem Rückgang bei tippen der **Schaltfläche -**. Der Cursor wird blau, wenn *der Container den Fokus hat*.

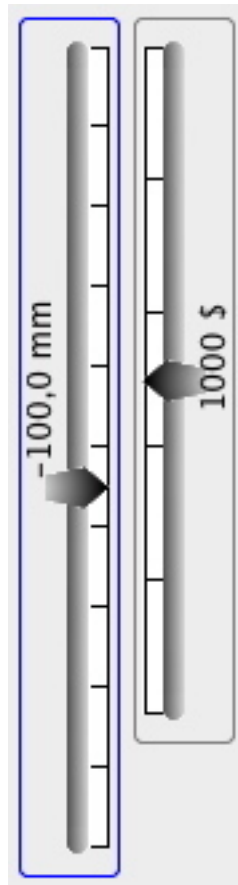


Abbildung 15: Container **CSliderV**. Sie können statt der Indikator-Wert (1000 \$ oder 100,0 mm) nach links (**kGradLeft**) oder rechts (**kGradRight**) des Cursor.

Ce container contient:

- Ein Bild mit einer Maske (type PNG): **LineTop**.
- Ein Bild mit einer Maske (type PNG): **LineVerM**.
- Ein Bild mit einer Maske (type PNG): **LineBottom**.
- Ein Bild mit einer Maske (type PNG): **PointeurLeft**.
- Ein Bild mit einer Maske (type PNG): **PointeurLeftOn**.
- Ein Bild mit einer Maske (type PNG): **PointeurRight**.
- Ein Bild mit einer Maske (type PNG): **PointeurRightOn**.



- Ein canvas: **Cnvs**.

Per Drag & Drop die Position der **CSliderV**.rbw **ExportGizmoCRB 2.0.2** (siehe Abbildung [6]) in das Projekt des Programms, und die Bilder oben gezeigt.

Die Nutzung dieser Container ist ähnlich wie **CSliderH**[12].

## 13.2 Beispiel Usage:

```
' CSliderV-----
CSliderV1.Resize(250,500,1500,"$",5,CSliderV1.kGradLeft,&c7F7F7F)
' Arguments: Länge der Schubstange,Mini, Maxi,Einheit,Anzahl der Ticks,Position der Indikatorwert,Fo
' Startwert -----
CSliderV1.SetValue(1000)
' Positionierung -----
CSliderV1.Left = CSliderV2.Left + CSliderV2.Width + 5
CSliderV1.Top = CSliderV2.Top
```

Es verfügt über 6 Funktionen zur Initialisierung, **3 für reelle Werte (double)**:

- **CSliderV1.Resize(L As Integer, MinR As double, MaxR As double, Unity As String, NTics As Integer, PosLeg As Integer)**. Zu definieren einen Horizontal SchiebeRegler.
  - **L** = Länge der Schubstange (in Pixeln).
  - **MinR** = Mindestwert (double).
  - **MaxR** = maximal zulässige Wert (double).
  - **Unity** = Werte der Einheit (zB mm, \$, etc..).
  - **NTics** = Anzahl der Ticks: das Intervall MinR, MaxR in NTics unterteilt.
  - **PosLeg** = Position der Indikatorwert. Siehe unten für Werte PosLeg.
- **CSliderV1.Resize(L As Integer, MinR As double, MaxR As double, Unity As String, NTics As Integer, PosLeg As Integer, AFormat As String)** Zu definieren einen Horizontal SchiebeRegler. Argumente identisch mit denen der oben genannten Funktionen, plus ein neues Argument:
  - **AFormat** = Zahlenformat für die Indikator-Wert.  
Standardmäßig, AFormat = -###.0##.
- **CSliderV1.Resize(L As Integer, MinR As double, MaxR As double, Unity As String, NTics As Integer, PosLeg As Integer, AFormat As String, AColor As Color)** Zu definieren einen Horizontal SchiebeRegler. Argumente identisch mit denen der oben genannten Funktionen, plus ein neues Argument:
  - **AColor**: Wenn dieses Argument vorhanden ist, wird ein Rahmen um den Container gezogen.

und **2 für Integer-Werte**:

- **CSliderV1.Resize(L As Integer, MinI As Integer, MaxI As Integer, Unity As String, NTics As Integer, PosLeg As Integer)**
- **CSliderV1.Resize(L As Integer, MinI As Integer, MaxI As Integer, Unity As String, NTics As Integer, PosLeg As Integer, AColor As Color)**

*Der einzige Unterschied bei den Funktionen für reelle Werte ist, dass hier die Argumente **MinI** et **MaxI** sind Integer Zahlen und das Argument **AFormat** ist gegangen.*

### **Valeurs de la constante PosLeg (Slider vertical) :**

- **CSliderV1.kGradLeft**: Indikator-Wert, die der Schieberegler links.
- **CSliderV1.kGradRight**: Indikator-Wert auf der rechten Seite der Schieberegler.

Auch können Sie den Startwert mit der Funktion **SetValue** gesetzt :

- **CSliderV1.SetValue(X As Integer/double)**: use **nach** der Funktion **Resize(...)**.

### 13.3 Erholung der Wert:

Nutzen Sie einfach die Veranstaltung **ValueChanged** die Instanz **CSliderV1** von **CSliderV** oder bei die Funktion **GetValue**.

- **CSliderV1.GetValue As double**: Gibt den Wert des Wert Indikator.
- **CSliderV1.ValueChanged(X As double)**:X ist des Wert des Indikator Wert.

## 14 Dialog DlgDate

### 14.1 Nützlichkeit und Umsetzung.

Mit diesem Dialog können Sie bequem auswählen Datum und Uhrzeit (siehe Abbildung [16]). Die Tasten + und - nach den Monaten und Jahren zu navigieren. Mit einem Klick auf das Jahr oder Monat Sie können wählen eine arbitrary datum. Mit einem Klick auf das Raster wählen Sie das Datum. Sie können die Zeit wählen mit der Zeit-Felder.



Abbildung 16: Dialog DlgDate. Beachten Sie die beiden Schaltflächen am unteren Rand des Dialogs, um eine **Kopie** oder eine **Paste** machen.

Dieser Dialog enthält:

- Dialog **DlgDate**.
- Klasse **TCalendar**.
- Container **CombiEditNumStd**.
- Dialog **DlgAskDate**.
- Label **STText**.
- Container **CEditNum**.
- Modul **SabirDate**.

Es ist genügend drag and drop oder importieren diese Elemente von Ordner **ExportGizmoCRB 2.0.2** (siehe Abbildung [6]) in Ihrem Projekt.

## 14.2 Erholung der Wert:

Bei die Funktion `GetValue As double`.

## 14.3 Beispiel Usage:

```
DlgDate.TotalSeconds = CurrentDate.TotalSeconds
DlgDate.ShowModal
if DlgDate.IsOk then
    CurrentDate.TotalSeconds = DlgDate.GetValue
end if
```

wo `CurrentDate` was ist eine Instanz der `Date`.

Denken Sie daran, den Wert von `Langage` zu setzen vor dem Aufruf zum Dialog[\[1.4.1\]](#).