

Das Package Multis

Installation

TMultiPanel

TMultiButton

TMultiButtonStyleManager

TMultiplexSlider

TMultiSeperator

TMultiLayer

TMultiRadioGroup

TMultiCheckGroup

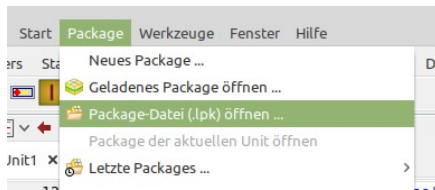
TMultiSwitch

Siehe auch <https://www.lazarusforum.de/viewtopic.php?f=29&t=14033>
oder die Demo Projekte im Package.

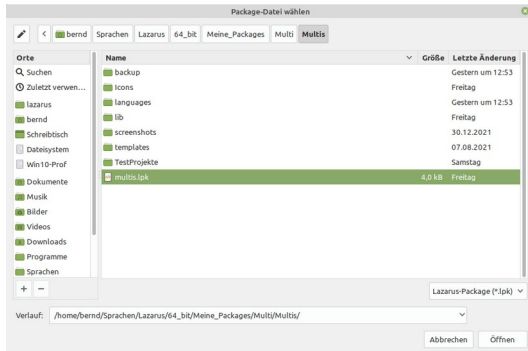
Installation

Das Package befindet sich in folgendem Githubkonto: <https://github.com/wennerer/Multis>

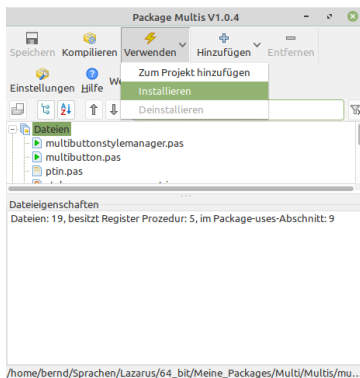
Nachdem das Package geklont oder heruntergeladen wurde kann es in Lazarus installiert werden. Dazu Lazarus öffnen, dort unter Package den Punkt Package-Datei (.lpk) öffnen... anklicken.



Jetzt in den Ordner Multis navigieren und dort die Datei multis.lpk auswählen.

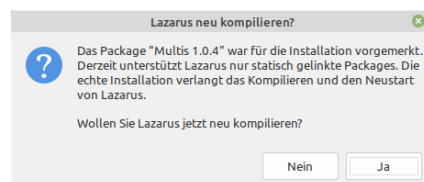


Es öffnet sich folgendes Fenster:



Hier auf Verwenden und dann Installieren.

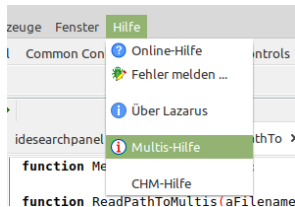
Diesen Dialog mit Ja bestätigen:



Zwischendurch erscheint eine Abfrage ob für diese Hilfe ein Menüeintrag erstellt werden soll. Möchte man dies sollte man einfach mit Ja bestätigen.



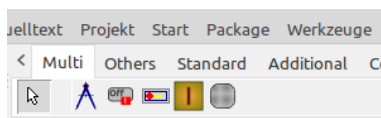
Im Hilfemenü befindet sich nun ein neuer Eintrag Multis-Hilfe.



Möchte man warum auch immer den Installationsdialog erneut aufrufen löscht man im Lazarus Config-Ordner die Datei multis.xml. Der Dialog wird dann erneut angezeigt.

Tipp: Ist der Hilfe Menüeintrag vorhanden aber es wird die Hilfedatei nicht gefunden kann es daran liegen das noch ungültige Package-Links eingetragen sind. Dazu bitte unter Package, Package-Links nach Multis suchen und alle ungültigen Links entfernen.

Nach erfolgreicher Installation befindet sich ein neuer Reiter Multi in der Palettenauswahl.



TMultiPanel

Eigenschaften

Action	: Die dem Steuerelement zugeordnete Aktion
Align	: Gibt die Platzierung des Steuerelements innerhalb seines übergeordneten Elements an
Anchors	: Der Satz von Ankerdefinitionen für dieses Steuerelement
AnimationSpeed	: Geschwindigkeit für Appear bzw. Disappear (default 0,05) (nur zur Laufzeit!)
Appear	: läßt das Panel erscheinen (nur zur Laufzeit!)
Autosize	: Ermöglicht die automatische Anpassung der Größe des Steuerelements an seinen Inhalt
BidiMode	: Anpassung (von Textsteuerelementen) in bidirektionalen Leseumgebungen
BorderSettings	: Die Eigenschaften des Randes
BorderSettings.Between	: Der Raum zwischen Innen- und Außenrand
BorderSettings.InnerColor	: Die Farbe des inneren Randes
BorderSettings.InnerWidth	: Die Dicke des inneren Randes
BorderSettings.OuterColor	: Die Farbe des äußeren Randes
BorderSettings.OuterWidth	: Die Dicke des äußeren Randes
BorderSpacing	: Bestimmt den inneren und äußeren Rahmenabstand für dieses Steuerelement
Caption	: Der Text den der Benutzer in das Panel schreibt
CaptionAlignment	: Ausrichtung des Textes in der Caption (Links,Mitte,Rechts)
CaptionHorMargin	: Der horizontale Abstand des Textes im Textrechteck (nur wirksam mit taLeftJustify)
CaptionLayout	: Ausrichtung des Textes in der Caption (Oben,Mitte,Unten)
CaptionVerMargin	: Der vertikale Abstand des Textes im Textrechteck (nur wirksam mit tlTop)
CaptionWordbreak	: Ermöglicht einen Zeilenumbruch in der Caption
ColorEnd	: Die Endfarbe des Panels (für Farbverlauf)
ColorGradient	: Die Richtung des Farbverlaufs
ColorStart	: Die Startfarbe des Panels (für Farbverlauf)
Constraints	: Die minimale und maximale Breite und Höhe für das Steuerelement
Cursor	: Die Form des Mauszeigers, wenn sich die Maus über diesem Steuerelement befindet
Disappear	: läßt das Panel verschwinden (nur zur Laufzeit!)
DoubleBuffered	: Ermöglicht das Reduzieren des Flimmerns in der Lackierung des Steuerelements
DragCursor	: Die Cursorform, die angezeigt wird, während das Steuerelement gezogen wird
DragKind	: Der Vorgang, wenn das Steuerelement gezogen wird - Drag or Dock
DragMode	: Ermöglicht dem Benutzer, das Steuerelement zu ziehen
DrawACustomPanel	: Öffnet einen Editor in dem man ein Panel zeichnen kann
DropDownMenu	: Die Eigenschaften der DropDownfunktion
DropDownMenu.Active	: Aktiviert die DropDown-Funktion
DropDownMenu.Compressed	: Eigenschaften des komprimierten Panels
DropDownMenu.Compressed.Active	: Macht die Auswahl zum Startwert
DropDownMenu.Compressed.Height	: Die Höhe des komprimierten Panels
DropDownMenu.Compressed.Width	: Die Breite des komprimierten Panels

<code>DropDownMenu.Direction</code>	: Die Ausklapprichtung
<code>DropDownMenu.Hotspot</code>	: Legt den Bereich fest in dem ein Klick wirkt, nur aktive mit <code>DropDownMenu.Active</code> und <code>trPinned</code> (nur zur Laufzeit!)
<code>DropDownMenu.HotspotCursor</code>	: Die Form des Mauszeigers wenn die Maus über dem Hotspot ist (nur zur Laufzeit!)
<code>DropDownMenu.Speed</code>	: Die Zeichengeschwindigkeit (Timer Intervall)
<code>DropDownMenu.Step</code>	: Der Zeichenschritt (in Pixel)
<code>DropDownMenu.Stretched</code>	: Eigenschaften des gedehnten Panels
<code>DropDownMenu.Stretched.Active</code>	: Macht die Auswahl zum Startwert
<code>DropDownMenu.Stretched.Height</code>	: Die Höhe des gedehnten Panels
<code>DropDownMenu.Stretched.Width</code>	: Die Breite des gedehnten Panels
<code>DropDownMenu.Trigger</code>	: Der Auslöser
Font	: Die Schrift die für die Textanzeige in diesem Panel verwendet werden soll
Height	: Die vertikale Größe des Steuerelements
HelpContext	: Die ID für die kontextbezogene Hilfe zu diesem Steuerelement
HelpKeyword	: Das Schlüsselwort für die kontextbezogene Hilfe zu diesem Steuerelement
HelpType	: Legt fest, ob die kontextsensitive Hilfe nach numerischer ID oder Schlüsselwort ausgewählt wird
Hint	: Der Text, der im Hinweifenster für das Steuerelement angezeigt werden soll
ImageIndex	: Der Index eines Bildes in einer <code>ImageList</code>
ImageLeft	: Die Koordinate der linken Ecke des Bildes
Images	: Eine Liste zum Einfügen von Bildern
ImageTop	: Die Koordinate der oberen Ecke des Bildes
ImageWidth	: Die einmalige Breite aller Bilder in der Liste
<code>Left</code>	: Die Clientkoordinate des linken Rands des Steuerelements
<code>RndRctRadius</code>	: Eckendurchmesser wenn geometrische Form ist <code>RoundRect</code>
<code>ParentAsBkgrd</code>	: Hintergrund des Panels nimmt die Farbe des Parent an (nur zur Laufzeit!)
<code>Style</code>	: Die geometrische Form des Panels
<code>Top</code>	: Die Clientkoordinate des oberen Rands des Steuerelements
Visible	: Ermöglicht das Ein- oder Ausblenden des Steuerelements und aller seiner untergeordneten Elemente
Width	: Die horizontale Ausdehnung des Steuerelements

Ereignisse

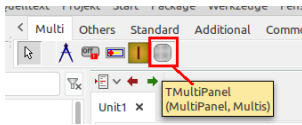
OnChangeBounds	: Handler für eine Änderung der Grenzen des Steuerelements
OnClick	: Benachrichtigungshandler für Mausklicks
OnCompressed	: Handler wenn das Panel komprimiert ist, nur aktive wenn
DropDownMenu.Active	
OnDragDrop	: Dieser Handler bestimmt die Aktion beim Ablegen auf diesem Steuerelement in einem Drag-Drop-Vorgang
OnDragOver	: Ereignishandler für ein Steuerelement, das über dieses Steuerelement gezogen wird
OnEndDrag	: Benachrichtigungshandler für das Ende eines Ziehvorgangs
OnEnter	: Handler für die Steuerung, die den Fokus erhält
OnExit	: Handler für die Steuerung, die den Fokus verliert;
OnKeyDown	: Handler für gedrückte Tastaturtaste
OnKeyPress	: Handler für ein vom Benutzer eingegebenes Zeichen
OnKeyUp	: Handler für Tastaturtaste freigegeben
OnMouseDown	: Ereignishandler für das Drücken der Maustaste
OnMouseEnter	: Ereignishandler für das Betreten des Bereichs des Steuerelements mit der Maus
OnMouseLeave	: Ereignishandler für Maus, die den Bereich des Steuerelements verlässt
OnMouseMove	: Ereignishandler für die Mausbewegung innerhalb des Steuerelements
OnMouseUp	: Ereignishandler für das Loslassen der Maustaste
OnStartDrag	: Ereignishandler für den Start eines Ziehvorgangs
OnStretched	: Handler wenn das Panel ausgeklappt ist, nur aktive wenn
DropDownMenu.Active	
OnVisible	: Wird ausgelöst wenn das Panel das erste Mal sichtbar wird

Öffentliche Prozeduren

```
procedure MouseMove({%H-}Shift: TShiftState; X, Y: Integer);override;
procedure MouseDown({%H-}Button: TMouseButton;{%H-}Shift: TShiftState; X, Y: Integer);override;
procedure MouseUp({%H-}Button: TMouseButton; {%H-}Shift: TShiftState; {%H-}X, {%H-}Y: Integer);override;
procedure LoadFromFile(aFileName: string);
procedure InvalidateBackground;
procedure ParentInputHandler({%H-}Sender: TObject; Msg: Cardinal);
procedure Notification(AComponent: TComponent;Operation: TOperation); override;
constructor Create(AOwner: TComponent); override;
destructor Destroy; override;
procedure MouseEnter; override;
procedure MouseLeave; override;
procedure Paint; override;
```

Beschreibung

Das MultiPanel findest du im Reiter Multis

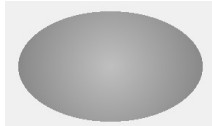


Die Form des MultiPanels lässt sich mit der Eigenschaft [Style](#) beeinflussen.

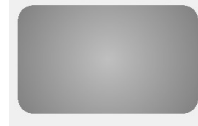
mpsRect:



mpsEllipse:



mpsRoundRect:

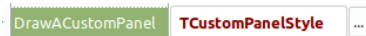


Der Eckenradius lässt sich mit [RndRctRadius](#) einstellen.
Standardeinstellung ist 40

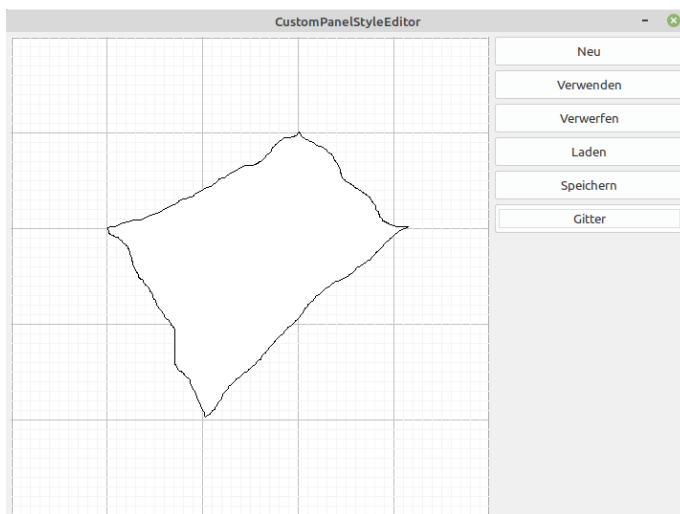
mpsCustom:



Bei mpsCustom ist standardmäßig ein Dreieck hinterlegt. Um ein benutzerdefiniertes Panel zu zeichnen muss man auf auf die 3 Punkte hinter [DrawACustomPanel](#) klicken.



Es öffnet sich ein Eigenschaftseditor:



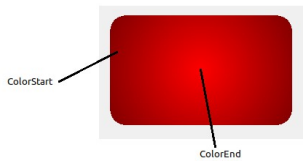
Klickt man jetzt auf Neu kann man einfach mit der Maus die Form des MultiPanels zeichnen. Klickt man auf Verwenden wird die MultiPanel-Form übernommen. Mit Verwerfen wird die MultiPanel-Form nicht übernommen und der Editor geschlossen. Mit Speichern kann man eine gezeichnete Form abspeichern und mit Laden wieder holen. Gitter blendet ein Hilfgitter ein das einem eventuell beim Zeichnen helfen kann.

Bedenke es muss mpsCustom eingestellt sein!

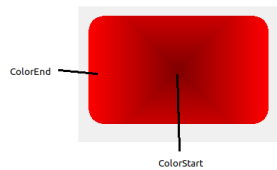
Zur Laufzeit lassen sich auch mit [LoadFromFile](#) vorab gespeicherte MultiPanel's laden.

Um die Farbe des MultiPanels zuverändern benötigst du die Eigenschaften [ColorStart](#), [ColorEnd](#) und [ColorGradient](#). Um ein einfarbiges MultiPanel zu bekommen muss ColorStart und ColorEnd gleich sein. Ansonsten bestimmt die Zusammensetzung aus den drei Eigenschaften das Aussehen.

gcSpread:



gcRadiant:

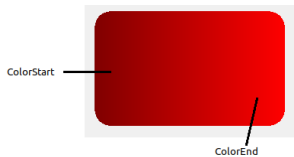


gcAlternate:

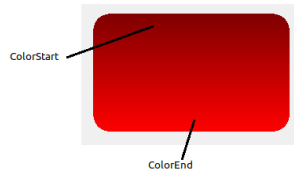


Setzt abwechselnd ein Pixel je auf Start- und EndColor.

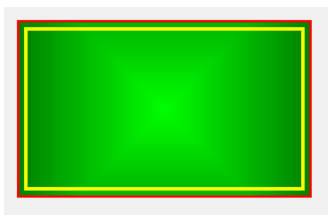
gcHorizontal:



gcVertical:



Wenn man den Rand hervorheben möchte stehen die Eigenschaften des [BorderSettings](#) zur Verfügung.



BorderSettings	
Between	7
InnerColor	clYellow
InnerWidth	3
OuterColor	clRed
OuterWidth	3

Um einen Rand zu erzeugen muss man einfach eine Farbe auswählen. Will man keinen Rand verwendet man clNone.

[BorderSettings.Between](#)

: Der Raum zwischen Innen- und Außenrand

[BorderSettings.InnerColor](#)

: Die Farbe des inneren Randes

[BorderSettings.InnerWidth](#)

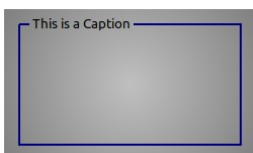
: Die Dicke des inneren Randes

[BorderSettings.OuterColor](#)

: Die Farbe des äußeren Randes

[BorderSettings.OuterWidth](#)

: Die Dicke des äußeren Randes



Möchte man eine GroupBox imitieren macht man die Innere Border sichtbar und schiebt diese mit BorderSettings.Between an die gewünschte Stelle. Damit die Border im Bereich der Caption unterbrochen wird muss für die Caption taLeftJustify und tlTop ausgewählt sein!

Die Eigenschaften [Appear](#), [Disappear](#) und [AnimationSpeed](#) können nur zur Laufzeit gesetzt werden!

Um ein unsichtbares MultiPanel erscheinen zu lassen benutzt man die Eigenschaft [Appear](#).

Beispiel-Code:

```
procedure TForm1.MultiButton1Click(Sender: TObject);
begin
    MultiPanel1.Appear:= true;
end;
```

Um ein sichtbares MultiPanel verschwinden zu lassen benutzt man die Eigenschaft [Disappear](#).

Beispiel-Code:

```
procedure TForm1.MultiButton2Click(Sender: TObject);
begin
    MultiPanel1.Disappear:= true;
end;
```

Mit der Eigenschaft [AnimationSpeed](#) lässt sich die Geschwindigkeit des Erscheinen bzw. Verschwindens beeinflussen.

Standartwert ist 0,05. Je kleiner der Wert desto langsamer geht die Animation vonstatten. Bei einem Wert von 0,001 geht es schon sehr langsam.

Beispiel-Code:

```
MultiPanel1.AnimationSpeed:= 0.001;
```

Wählt man als geometrische Form (Eigenschaft Style) etwas anderes als `mpsRect` so wird ein Teil des Hintergrundes des `MultiPanels` sichtbar. Diese nun sichtbaren Ecken nehmen die im Parent gesetzte Farbe an. Befinden sich im Parent zum Beispiel selbst gezeichnete Linien werden diese auch gezeigt. Dies geschieht da standardmäßig die Eigenschaft `ParentAsBkgrd` auf `true` gesetzt ist.

`ParentAsBkgrd := true`



`ParentAsBkgrd := false`



Diese Einstellung macht vor allem Sinn wenn der Parent seine Größe ändert. Denn dann wird zum Beispiel hier die gezeichnete Linie nicht richtig skaliert.

Besitzt der Parent einen Farbverlauf besteht die Möglichkeit die Skalierungsproblematik durch Aufruf der Prozedur `InvalidateBackground` auszugleichen.

Beispiel-Code:

```
procedure TForm1.FormChangeBounds(Sender: TObject);
begin
  MultiPanel1.InvalidateBackground;
end;
```



Um ein **DropDown** Menü (Hamburgermenü) zu erzeugen stellt man zuerst die Eigenschaft **DropDownMenu.Active** auf **true**.

DrawACustomPanel	TCustomPanelStyle
DropDownMenu	(TDDMenu)
Active	<input checked="" type="checkbox"/> (True)
Compressed	(TComp)
Direction	LeftTop_RightBottom
Speed	3
Step	2
Stretched	(TStre)
Trigger	trHover

Das MultiPanel zeigt nun den Komprimierten Zustand. Die Positionierung kann mit der Maus oder den Eigenschaften **Left** bzw. **Top** gemacht werden (natürlich können auch Anker gesetzt werden). Die Größe kann man einfach mit der Maus ziehen oder mit den Eigenschaften **DropDownMenu.Compressed.Height** bzw. **DropDownMenu.Compressed.Width** zuweisen. Passt die Größe schaltet man mit den Eigenschaften **DropDownMenu.Stretched.Active** bzw.

DropDownMenu.Compressed.Active auf den ausgedehnten Zustand. Nun kann ebenfalls die gewünschte Größe durch ziehen mit der Maus oder den Eigenschaften **DropDownMenu.Stretched.Height** bzw.

DropDownMenu.Stretched.Width eingestellt werden. Es empfiehlt sich in diesen

Zustand die gewünschten Kinder-Controls (Button's etc.) zu platzieren. Die Richtung nach der das MultiPanel ausklappt wird durch die Eigenschaft **DropDownMenu.Direction** bestimmt. Zur Auswahl stehen folgende Möglichkeiten:

TDirection = (LeftTop_RightBottom,RightTop_LeftBottom,LeftBottom_RightTop,RightBottom_LeftTop)

Die Geschwindigkeit des Ausklappens kann man durch die Eigenschaften **DropDownMenu.Speed** und **DropDownMenu.Step** beeinflussen. Hinter Speed versteckt sich das Timer-Intervall mit dem das Ausklappen aufgerufen wird. Zum Verlangsamen erhöht man dieses Wert bis zur gewünschten Geschwindigkeit. Mit Step lässt sich die Anzahl der Pixel einstellen die je Intervall zusätzlich gezeichnet werden. Will man also schneller Ausklappen erhöht man also den Wert bei Step.

Mit **DropDownMenu.Trigger** bestimmt man den Auslöser für das Ausklappen. Es stehen folgende Möglichkeiten bereit:

TTrigger = (trClick,trHover,trPinned)

Bei trClick muss in das Panel geklickt werden, bei trHover reicht es mit der Maus darüber zu fahren. Bei trPinned muss in einem festlegbaren Hotspot (**DropDownMenu.Hotspot**) geklickt werden. Befindet sich die Maus über diesem Hotspot kann der Mauszeiger mit **DropDownMenu.HotspotCursor** angepasst werden. Standardmäßig ändert er sich auf crHandPoint.

TMultiButton

Eigenschaften

Action	: Die dem Steuerelement zugeordnete Aktion
Align	: Gibt die Platzierung des Steuerelements innerhalb seines übergeordnetenElements an
AllowsUp	: Erlaubt eine gedrückte Schaltfläche auf nicht gedrückt zu setzen
Anchors	: Der Satz von Ankerdefinitionen für dieses Steuerelement
AutoSize	: Ermöglicht die automatische Anpassung der Größe des Steuerelements an seinen Inhalt
BidiMode	: Anpassung (von Textsteuerelementen) in bidirektionalen Leseumgebungen
BorderColor	: Die Farbe des Rahmens
BorderSpacing	: Bestimmt den inneren und äußeren Rahmenabstand für dieses Steuerelement
BorderWidth	: Die Dicke des Rahmens
Caption	: Der Text den der Benutzer in den Button schreibt
CaptionAlignment	: Ausrichtung des Textes in der Caption (Links,Mitte,Rechts)
CaptionHorMargin	: Der horizontale Abstand des Textes im Textrechteck (nur wirksam mit taLeftJustify)
CaptionLayout	: Ausrichtung des Textes in der Caption (Oben,Mitte,Unten)
CaptionVerMargin	: Der vertikale Abstand des Textes im Textrechteck (nur wirksam mit tlTop)
CaptionWordbreak	: Ermöglicht einen Zeilenumbruch in der Caption
ColorEnd	: Die Endfarbe des Buttons (für Farbverlauf)
ColorGradient	: Die Richtung des Farbverlaufs
ColorStart	: Die Startfarbe des Buttons (für Farbverlauf)
Constraints	: Die minimale und maximale Breite und Höhe für das Steuerelement
DisabledAlphaBValue	: Der Blendwert bei Enable:=false, nur zur Laufzeit!
DisabledColor	: Die Farbe bei Enable:=false, nur zur Laufzeit!
Down	: Der Button wird in den gedrückt Status versetzt
DragCursor	: Die Cursorform, die angezeigt wird, während das Steuerelement gezogen wird
DragKind	: Der Vorgang, wenn das Steuerelement gezogen wird - Drag or Dock
DragMode	: Ermöglicht dem Benutzer, das Steuerelement zu ziehen
Enable	: Legt fest, ob das Steuerelement auf Maus- oder Tastatureingaben reagiert
FocusAlphaBValue	: Wie transparent der Fokusrahmen ist (0=transparent, 255=undurchsichtig)
FocusColor	: Die Farbe des Fokusrahmens/Foregroundfocus wenn das Control den Fokus hat
FocusFrameOn	: Schaltet den Fokusrahmen ein und aus
FocusFrameWidth	: Die Dicke des Fokus-Rahmens
Font	: Die Schrift die für die Textanzeige in diesem Button verwendet werden soll
ForegroundFocusOn	: Zeigt an wenn der Button den Fokus besitzt
GroupIndex	: Der Index der Gruppe zu der der MultiButton gehört
Height	: Die Höhe des Controls. Die Höhe des MultiButtons ist minus HoverFrameWidth
HelpContext	: Die ID für die kontextbezogene Hilfe zu diesem Steuerelement
HelpKeyword	: Das Schlüsselwort für die kontextbezogene Hilfe zu diesem Steuerelement

HelpType	: Legt fest, ob die kontextsensitive Hilfe nach numerischer ID oder Schlüsselwort ausgewählt wird
Hint	: Der Text, der im Hinweifenster für das Steuerelement angezeigt werden soll
HoverEndColor	: Die Endfarbe eines Hoverereignisses
HoverFontColor	: Die Farbe der Caption während eines Hoverereignisses
HoverImageIndex	: Der Index eines Bildes in einer ImageList während eines Hoverereignisses
HoverOn	: Ermöglicht das Ein- oder Ausblenden eines Hoverereignisses
HoverStartColor	: Die Startfarbe eines Hoverereignisses
ImageIndex	: Der Index eines Bildes in einer ImageList
ImageLeft	: Die Koordinate der linken Ecke des Bildes
Images	: Eine Liste zum Einfügen von Bildern
ImageTop	: Die Koordinate der oberen Ecke des Bildes
ImageWidth	: Die einmalige Breite aller Bilder in der Liste
Left	: Die Clientkoordinate des linken Rands des Steuerelements
MessageButton	: Ein Message Button um Infos anzuzeigen bzw. um einen zweiten integrierten Button bereit zustellen
MessageButton.Alignment	: Die Position des MessageButtons
MessageButton.BorderColor	: Die Farbe des Rahmens
MessageButtonBorderWidth	: Die Dicke des Rahmens
MessageButton.CalculateAlthoughInvisible	: Wird benötigt wenn der MessageButton erst zur Laufzeit sichtbar wird
MessageButton.Caption	: Der Text den der Benutzer in den MessageButton schreibt
MessageButton.CaptionAlignment	: Ausrichtung des Textes in der Caption (Links,Mitte,Rechts)
MessageButton.CaptionHorMargin	: Der horizontale Abstand des Textes im Textrechteck (nur wirksam mit taLeftJustify)
MessageButton.CaptionLayout	: Ausrichtung des Textes in der Caption (Oben,Mitte,Unten)
MessageButton.CaptionVerMargin	: Der vertikale Abstand des Textes im Textrechteck (nur wirksam mit tlTop)
MessageButton.ColorEnd	: Die Endfarbe des MessageButtons (für Farbverlauf)
MessageButton.ColorGradient	: Die Richtung des Farbverlaufs
MessageButton.ColorStart	: Die Startfarbe des MessageButtons (für Farbverlauf)
MessageButton.Font	: Die Schrift die für die Textanzeige in diesem Button verwendet werden soll
MessageButton.Height	: Die vertikale Ausdehnung des MessageButtons
MessageButton.HoverColor	: Die Farbe eines Hoverereignisses
MessageButton.HoverOn	: Ermöglicht das Ein- oder Ausblenden eines Hoverereignisses
MessageButton.ImageIndex	: Der Index eines Bildes in einer ImageList
MessageButton.ImageLeft	: Die Koordinate der linken Ecke des Bildes
MessageButton.Images	: Eine Liste zum Einfügen von Bildern
MessageButton.ImageTop	: Die Koordinate der oberen Ecke des Bildes
MessageButton.ImageWidth	: Die einmalige Breite aller Bilder in der Liste
MessageButton.PositionFactor	: Positionsfaktor, nur aktive wenn alSE,alSW,alNW,alNE,alW,alE,alN,alS,alRightIn,alLeftIn,alTopIn,alBottomIn
MessageButton.PresdColBlendVal	: Wie transparent die PressedColor ist (0=transparent, 255=undurchsichtig)
MessageButton.PressedColor	: Die Farbe des MessageButtons wenn er gedrückt wird
MessageButton.ShowBorder	: Ermöglicht das Ein- oder Ausblenden eines Rahmens
MessageButton.ShowPressed	: Ermöglicht das Ein- oder Ausblenden der Gedrücktoption
MessageButton.Style	: Die geometrische Form des MessageButtons

<code>MessageButton.Visible</code>	: Ermöglicht das Ein- oder Ausblenden des Steuerelements und aller seiner untergeordneten Elemente
<code>MessageButton.Width</code>	: Die horizontale Ausdehnung des MessageButtons
<code>MultiButton_StyleManager</code>	: Vereinfacht das Gestalten des MultiButtons
<code>PopupMenu</code>	: Ein Menü das angezeigt wird, wenn Sie mit der rechten Maustaste auf dieses Steuerelement klicken
<code>PressedEndColor</code>	: Die Endfarbe des Buttons wenn er gedrückt wird (für Farbverlauf)
<code>PressedFontColor</code>	: Die Farbe des Textes der Caption wenn der Button gedrückt wird
<code>PressedImageIndex</code>	: Der Index eines Bildes in einer ImageList wenn der Button gedrückt ist
<code>PressedStartColor</code>	: Die Startfarbe des Buttons wenn er gedrückt wird (für Farbverlauf)
<code>RndRctRadius</code>	: Eckendurchmesser wenn geometrische Form ist RoundRect
<code>ShowBorder</code>	: Ermöglicht das Ein- oder Ausblenden eines Rahmens
<code>ShowHint</code>	: Wenn True, wird der Hinweistext angezeigt, wenn sich die Maus über dem Steuerelement befindet
<code>ShowMsgButtonInGroup</code>	: Zeigt den Message Button auf einem MultiButton in einer Gruppe
<code>ShowTurnedOn</code>	: Macht einen sichtbaren MessageButton farbig wenn der Button Down ist
<code>Style</code>	: Die geometrische Form des Buttons
<code>TabOrder</code>	: Bestimmt die Reihenfolge der Steuerelementnavigation, wenn der Benutzer die Tabulatortaste drückt
<code>TabStop</code>	: Ermöglicht dem Benutzer das Navigieren zu diesem Steuerelement durch Drücken der Tabulatortaste
<code>Top</code>	: Die Clientkoordinate des oberen Rands des Steuerelements
<code>Visible</code>	: Ermöglicht das Ein- oder Ausblenden des Steuerelements und aller seiner untergeordneten Elemente
<code>Width</code>	: Die Breite des Controls. Die Breite des MultiButtons ist minus <code>HoverFrameWidth</code>

Öffentliche Prozeduren

procedure `SetStyleManager`(AValue: TMultiButtonStyleManager);

Öffentliche Variablen

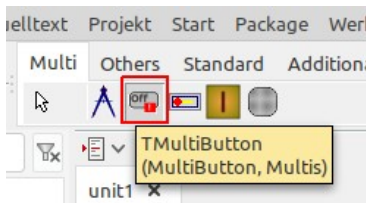
`MouseButton` : kann im `OnClick` verwendet werden um festzustellen mit welcher Maustaste der MultiButton geklickt wurde.

Ereignisse

OnClick	: Benachrichtigungshandler für Mausklicks
OnDragDrop	: Dieser Handler bestimmt die Aktion beim Ablegen auf diesem Steuerelement in einem Drag-Drop-Vorgang
OnDragOver	: Ereignishandler für ein Steuerelement, das über dieses Steuerelement gezogen wird
OnEndDrag	: Benachrichtigungshandler für das Ende eines Ziehvorgangs
OnEnter	: Handler für die Steuerung, die den Fokus erhält
OnExit	: Handler für die Steuerung, die den Fokus verliert;
OnKeyDown	: Handler für gedrückte Tastaturtaste
OnKeyPress	: Handler für ein vom Benutzer eingegebenes Zeichen
OnKeyUp	: Handler für Tastaturtaste freigegeben
OnMouseDown	: Ereignishandler für das Drücken der Maustaste
OnMouseEnter	: Ereignishandler für das Betreten des Bereichs des Steuerelements mit der Maus
OnMouseLeave	: Ereignishandler für Maus, die den Bereich des Steuerelements verlässt
OnMouseMove	: Ereignishandler für die Mausbewegung innerhalb des Steuerelements
OnMouseUp	: Ereignishandler für das Loslassen der Maustaste
OnStartDrag	: Ereignishandler für den Start eines Ziehvorgangs
MessageButton.OnClick	: Benachrichtigungshandler für Mausklicks
MessageButton.OnMouseMove	: Ereignishandler für die Mausbewegung innerhalb des Steuerelements

Beschreibung

Den MultiButton findest du im Reiter Multi:



Es ist wichtig zu wissen das der MultiButton von einem Fokusrahmen umgeben ist. Wie man hier sieht besitzt der fokussierte MultiButton einen olivgrünen Rahmen. Das bedeutet im Umkehrschluss das der eigentliche Button um den Rahmen kleiner ist.



Mit der Eigenschaft [FocusColor](#) kann man die Farbe des Fokusrahmens einstellen. Mit [FocusAlphaValue](#) kann die Transparenz des Fokusrahmens geregelt werden. Der Wert 0 bedeutet durchsichtig und 255 undurchsichtig. [FocusFrameWidth](#) bestimmt die Dicke des Rahmens.

Wert 50:



Wert 200:



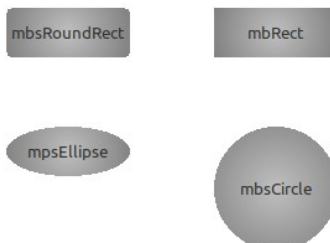
Setzt man [FocusFrameOn](#) auf false dann bleibt der Rand erhalten jedoch wird die Fokussierung nicht farblich angezeigt.

Mit [ForegroundFocusOn](#) besitzt der fokussierte MultiButton ein gepunktetes Rechteck. Die Farbe des Rechteckes kann man mit [FocusColor](#) beeinflussen. Es kann hier Sinnvoll sein [FocusFrameWidth](#) auf 0 und [FocusFrameOn](#) auf false zu setzen damit die Ecken nicht sichtbar werden!



Mit der Eigenschaft [Style](#) stellt man die gewünschte geometrische Form des MultiButtons ein.

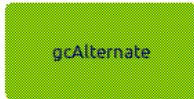
Ist [mbsRoundRect](#) eingestellt kann man mit der Eigenschaft [RndRectRadius](#) den Durchmesser der Eckenrundung einstellen.



Möchte man den MultiButton mit einem farbigen Rand ausstatten stellt man [ShowBorder](#) auf true. Die Farbe des Randes wird mit [BorderColor](#) und die Breite mit [BorderWidth](#) eingestellt.



Um die Farbe des MultiButtons zu verändern benötigst du die Eigenschaften [ColorStart](#), [ColorEnd](#) und [ColorGradient](#). Um einen einfarbigen MultiButton zu bekommen muss ColorStart und ColorEnd gleich sein. Ansonsten bestimmt die Zusammensetzung aus den drei Eigenschaften das Aussehen.



Alternate setzt abwechselnd ein Pixel je auf Start- und EndColor.



Hier ist StartColor clGreen und EndColor clYellow.

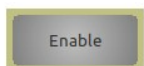
Standard mäßig steht [HoverOn](#) auf true. Das bedeutet das sich bei einem Hoverereignis (die Maus bewegt sich über dem MultiButton) mit [HoverStartColor](#), [HoverEndColor](#), [HoverFontColor](#) und [HoverImageIndex](#) das Aussehen nach Wunsch verändern lässt. Möchte man dies nicht setzt man HoverOn auf false.

HoverEndColor	: Die Endfarbe eines Hoverereignisses
HoverFontColor	: Die Farbe der Caption während eines Hoverereignisses
HoverImageIndex	: Der Index eines Bildes in einer ImageList während eines Hoverereignisses
HoverOn	: Ermöglicht das Ein- oder Ausblenden eines Hoverereignisses
HoverStartColor	: Die Startfarbe eines Hoverereignisses

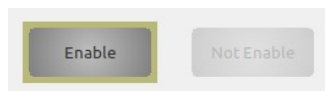
Wird der MultiButton gedrückt beeinflussen die Eigenschaften [PressedStartColor](#), [PressedEndColor](#), [PressedFontColor](#) und [PressedImageIndex](#) das Aussehen. Möchte man keine Änderung wenn der Button gedrückt wird bleibt nur die gleichen Einstellungen wie ColorStart etc. einzustellen.

PressedEndColor	: Die Endfarbe des Buttons wenn er gedrückt wird (für Farbverlauf)
PressedFontColor	: Die Farbe des Textes der Caption wenn der Button gedrückt wird
PressedImageIndex	: Der Index eines Bildes in einer ImageList wenn der Button gedrückt ist
PressedStartColor	: Die Startfarbe des Buttons wenn er gedrückt wird (für Farbverlauf)

Mit der Eigenschaft [Enable](#) legt man fest ob das Steuerelement auf Maus- oder Tastatureingaben reagiert. Das Aussehen bei nicht Enable kann zur Laufzeit mit [DisabledAlphaValue](#) und [DisabledColor](#) beeinflusst werden.



Möchte man keinen Rahmen sehen DisabledColor auf die gleiche Farbe wie den Parent stellen.

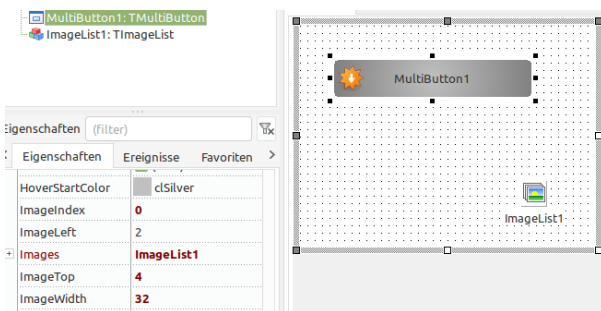


Verwende wenn nötig dazu nicht clDefault sondern `MultiButton2.DisabledColor:=GetColorResolvingParent;`

Möchte man ein Bild einfügen muss man zuerst eine ImageList auf die Form ziehen. Man weist dann dieser ImageList die gewünschten Bilder in den gewünschten Größen zu. Die Bedienung des BildListenEditors ist hier sehr gut beschrieben: <https://www.lazarusforum.de/viewtopic.php?f=18&t=13170>



Bei **Images** trägt man die auf der Form befindliche ImageList ein. Mit **ImageIndex** kann man das gewünschte Bild aus der ImageList wählen, wobei -1 kein Bild bedeutet. Mit **ImageLeft** und **ImageTop** bestimmt man die Position des Bildes. Mit **ImageWidth** kann man die Größe des Bildes skalieren. Es empfiehlt sich nur kleiner zu skalieren.



Tipp:

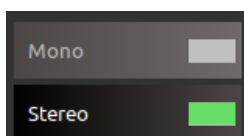
Verwendet man unter Windows HighDPI werden die Images und der MultiButton skaliert. Damit es zur Laufzeit funktioniert musste ich in den Projekteinstellungen noch bei DPI Anpassung Vista-8:an,8.1+:pro Monitor(True/PM) auswählen.

Mit der Eigenschaft **AllowsUp** verwandelt man den Button in eine Art Schalter. Dies bedeutet wenn man den Button drückt bleibt er gedrückt bis er wieder gedrückt wird. Möchte man das der MultiButton zum Beginn des Programmes gedrückt erscheint macht man dies mit der Eigenschaft **Down**.



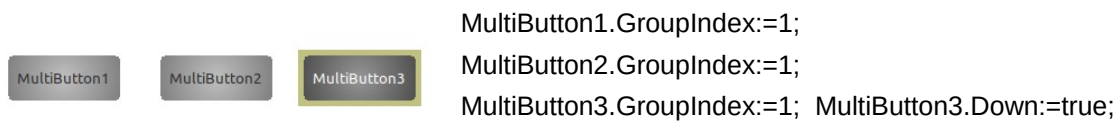
Ist der Button gedrückt wird er mit den Eigenschaften Pressed.... dargestellt!

Möchte man ein zusätzliches Merkmal haben das der Button in gedrückt Stellung ist kann man einen MessageButton sichtbar machen und setzt die Eigenschaft **ShowTurnedOn** auf true. Der MessageButton bekommt dann in gedrückter Stellung die Farbe die unter MessageButton.PressedColor festgelegt ist.



Siehe auch [GroupIndex](#)

Wenn es benötigt wird kann der MultiButton einer Gruppe angehören. Dies erreicht man mit der Eigenschaft [GroupIndex](#). Besitzt also ein MultiButton einen anderen Wert als 0 gehört er zu der Gruppe mit dem gleichen Wert. In einer Gruppe kann nur ein MultiButton gedrückt sein. Soll ein Button in der Gruppe beim Starten des Programmes bereits gedrückt sein kann man dies mit der Eigenschaft [Down](#) erreichen.



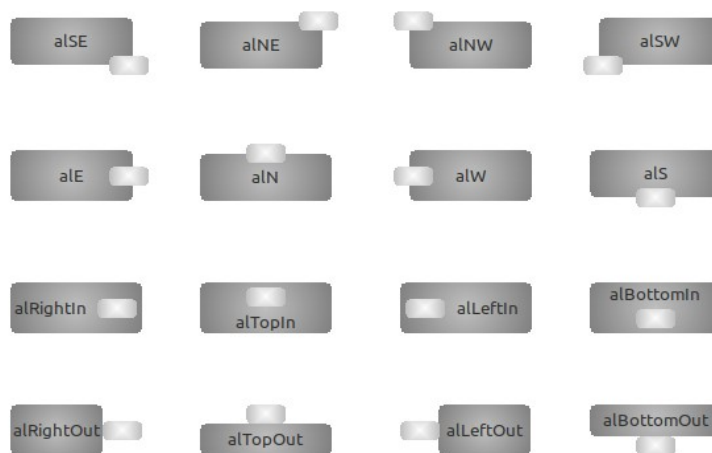
Will man den gedrückten Button optisch etwas deutlicher kenntlich machen bleibt einem noch die Eigenschaft [ShowMsgButtonInGroup](#). Der zuletzt gedrückte Button bekommt einen [MessageButton](#).



Der [MessageButton](#)



Um den integrierten [MessageButton](#) zu nutzen muss man zuerst [MessageButton.Visible](#) auf true stellen. Sodann kann man mit [MessageButton.Alignment](#) die Position des [MessageButtons](#) festlegen.



Mit der Eigenschaft [MessageButton.PositionFactor](#) kann die Position des [MessageButtons](#) etwas beeinflusst werden. Allerdings nur bei alSE,alSW,alNW,alNE,alW,alE,alN,alS,alRightIn,alLeftIn,alTopIn,alBottomIn



Möchte man die Form des [MessageButtons](#) ändern geschieht dies mit der Eigenschaft [MessageButton.Style](#).



Setzt man mehrere MultiButtons in eine Reihe und der MessageButton ist nicht bei allen sichtbar so haben die MultiButtons verschiedene Größen. Hier erscheint Nr. 2 größer:



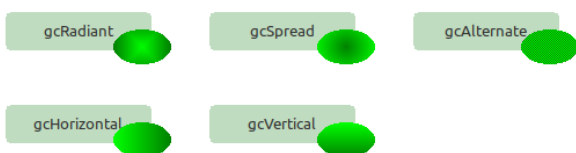
Um dies zu umgehen gibt es die Eigenschaft [MessageButton.CalculateAlthoughInvisible](#). Stellt man bei Nr. 2 diese Eigenschaft auf true sieht es so aus:



Möchte man den MessageButton mit einem farbigen Rand ausstatten stellt man [MessageButton.ShowBorder](#) auf true. Die Farbe des Randes wird mit [MessageButton.BorderColor](#) und die Breite mit [MessageButton.BorderWidth](#) eingestellt.



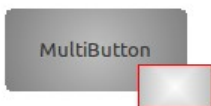
Um die Farbe des MessageButtons zu verändern benötigst du die Eigenschaften [MessageButton.ColorStart](#), [MessageButton.ColorEnd](#) und [MessageButton.ColorGradient](#). Um einen einfarbigen MessageButton zu bekommen muss ColorStart und ColorEnd gleich sein. Ansonsten bestimmt die Zusammensetzung aus den drei Eigenschaften das Aussehen.



Alternate setzt abwechselnd ein Pixel je auf Start- und EndColor.

Hier ist StartColor clLime und EndColor clGreen.

Standard mäßig steht [MessageButton.HoverOn](#) auf true. Das bedeutet das bei einem Hoverereignis (die Maus bewegt sich über dem MessageButton) um den MessageButton ein Rand gezeichnet wird. Die Farbe des Randes kann man mit [MessageButton.HoverColor](#) einstellen. Möchte man dies nicht setzt man HoverOn auf false.



MessageButton. BorderWidth beeinflusst die Dicke des HoverRandes!

Wird der MessageButton gedrückt und [MessageButton.ShowPressed](#) ist true dann wird die bei

[MessageButton.PressedColor](#) eingestellte Farbe mit dem bei [MessageButton.PressedColBlendVal](#) hinterlegtem Wert über den MessageButton geblendet. Wobei 0 transparent bedeutet und 255 undurchsichtig.



Möchte man im OnClick feststellen mit welcher Maustaste das Ereignis ausgelöst wurde kann man dies mit der öffentlichen Variablen [MouseButton](#). Der Wert in MouseButton bleibt erhalten bis er in der Procedure der MouseDown der Komponente erneut überschrieben wird.

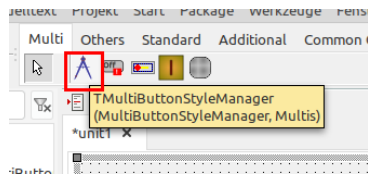
Beispiel:

```
procedure TForm1.MultiButton1Click(Sender: TObject);
begin
  if MultiButton1.MouseButton = mbLeft then showmessage('Left');
  if MultiButton1.MouseButton = mbRight then showmessage('Right');
  if MultiButton1.MouseButton = mbMiddle then showmessage('Middle');
end;
```

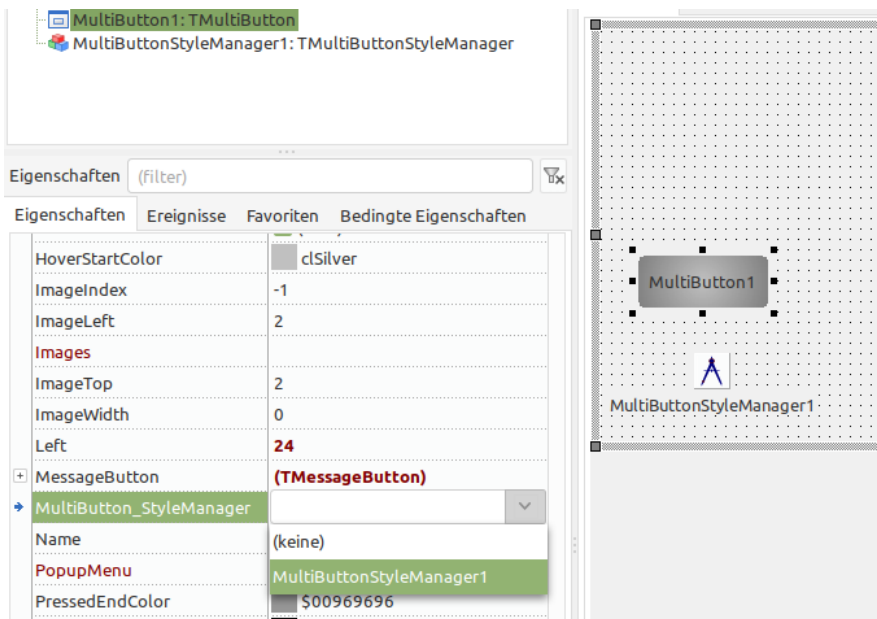
TMultiButtonStyleManager

Baut man ein Formular in dem viele MultiButtons vorkommen die optisch ähnlich aussehen sollen, so empfiehlt sich die Verwendung des MultiButtonStyleManagers.

Zu finden ist er hier:



Wie jede Komponente zieht man ihn einfach auf die Form. Um ihn mit einem MultiButton zu verbinden muss man nun bei der Eigenschaft **MultiButton_StyleManager** (des Buttons) den StyleManager auswählen.



Die im Objektinspektor unter MultiButtonStylmanager angezeigten Eigenschaften wirken sich nun auf alle verbundenen MultiButtons gleichzeitig aus.



Hier sind alle sechs Buttons verbunden. Durch umstellen des Styles auf mbsEllipse (im OI unter MBStyleManager) ändern sich alle MultiButtons auf einmal!

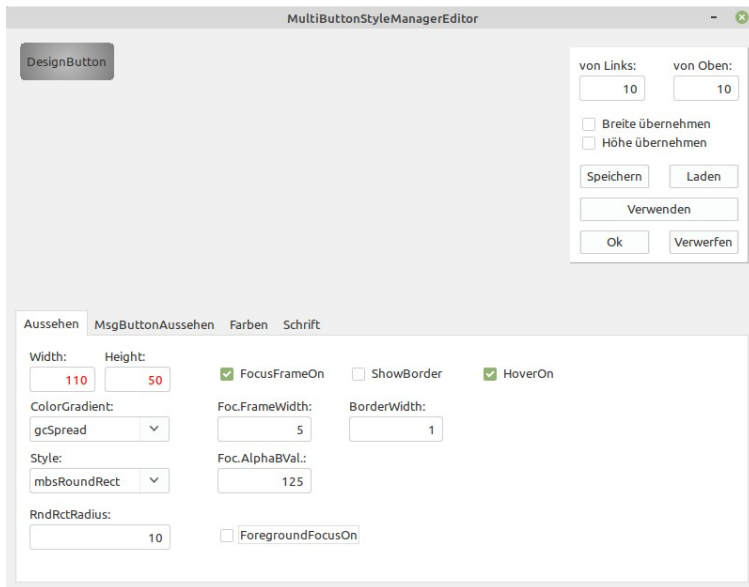
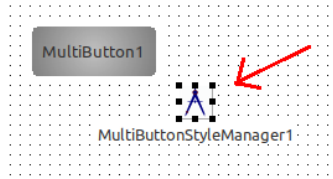
Achtung! Versucht man eine Eigenschaft eines verbundenen MultiButtons im OI Reiter des Buttons (nicht des Managers) zu ändern so schlägt das fehl!

Für mich hat sich die Vorgehensweise als gut herausgestellt das ich mit dem Stylemanager erst alle gewünschten Eigenschaften der MultiButtons einstelle und danach die Verbindung wieder heraus nehme. Das ist aber sicher Geschmackssache.

Eine besondere Situation ergibt sich bei den Eigenschaften Width und Height. Diese beiden Eigenschaften lassen sich über den Stylemanager nur ändern wenn die Eigenschaften **OffsetHeight** und **OffsetWidth** des Stylmanagers bewusst auf true gesetzt werden (Default = false).

MultiButtonStylemanagerEditor

Weiterhin bietet der MultiButtonStylemanager die Möglichkeit alle Einstellungen in einem MultiButtonStylemanagerEditor vorzunehmen. Öffnen lässt sich dieser durch einen Doppelklick auf das Komponentensymbol auf der Form.

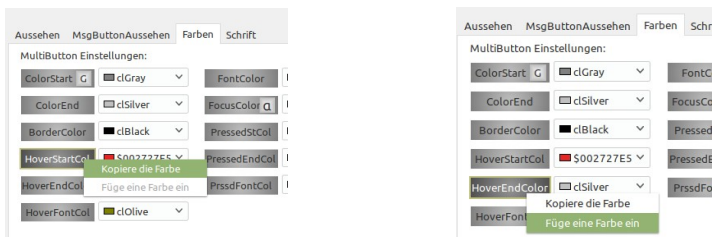


Überfährt man die verschiedenen Einstellmöglichkeiten mit der Maus werden deren Funktionen als Hinweise dargestellt.

Der Button im linken oberen Eck dient als Muster. Er kann mit *von Links* und *von Oben* im Editor bewegt werden.

Hat man ein Design erstellt das man eventuell in einem anderen Projekt wieder verwenden möchte lässt sich dieses mit Speichern in einer Datei sichern. Mit Laden kann man es von dort wieder holen.

Möchte man eine Farbe von einer Auswahl in eine Andere kopieren klickt man mit der rechten Maustaste in den Auswahlbutton, es öffnet sich ein Popup, dort wählt man kopieren bzw. einfügen.



Hat man mehrere MultiButtons mit einem Stylmanager verbunden und möchte die Buttons via Stylemanager zur Laufzeit verändern bedient man sicher der öffentlichen Procedure [SetStyleManager](#).

Das geht zum Beispiel so:

```
procedure TForm1.AdjustTheMultiButtons(Sender : TObject);
var lv : integer;
begin
  MultiButtonStyleManager1.ColorStart := clLime;
  MultiButtonStyleManager1.ColorEnd  := clRed;
  for lv := 0 to pred(ComponentCount) do
    if (Components[lv] is TMultiButton) then
      if TMultiButton(Components[lv]).MultiButton_StyleManager = MultiButtonStyleManager1 then
        TMultiButton(Components[lv]).SetStyleManager(MultiButtonStylemanager1);
  end;
```



TMultiplexSlider

Eigenschaften

Action	: Die dem Steuerelement zugeordnete Aktion
Align	: Gibt die Platzierung des Steuerelements innerhalb seines übergeordnetenElements an
Anchors	: Der Satz von Ankerdefinitionen für dieses Steuerelement
AutoRangeNegative	: Wird Min erreicht erhöht sich Min um AutoRangeValue
AutoRangePositive	: Wird Max erreicht erhöht sich Max um AutoRangeValue
AutoRangeValue	: Nur in Verbindung mit AutoRangePositive and AutoRangeNegative
AutoSize	: Nur aktive in Verbindung mit TextLabel
BorderColor	: Die Farbe des Rahmens (clNone macht unsichtbar)
BorderSpacing	: Bestimmt den inneren und äußeren Rahmenabstand für dieses Steuerelement
BorderWidth	: Die Dicke des Rahmens
ColorEnd	: Die Endfarbe des Sliders (für Farbverlauf)
ColorGradient	: Die Richtung des Farbverlaufs
ColorStart	: Die Startfarbe des Sliders (für Farbverlauf)
Constraints	: Die minimale und maximale Breite und Höhe für das Steuerelement
DragCursor	: Die Cursorform, die angezeigt wird, während das Steuerelement gezogen wird
DragKind	: Der Vorgang, wenn das Steuerelement gezogen wird - Drag or Dock
DragMode	: Ermöglicht dem Benutzer, das Steuerelement zu ziehen
Enabled	: Legt fest, ob das Steuerelement auf Maus- oder Tastatureingaben reagiert
FocusAlphaBValue	: Wie transparent der Fokusrahmen ist (0=transparent, 255=undurchsichtig)
FocusColor	: Die Farbe des Fokusrahmens/Foregroundfocus wenn das Control den Fokus hat
FocusFrameOn	: Schaltet den Fokusrahmen ein und aus
FocusFrameWidth	: Die Dicke des Fokus-Rahmens
ForegroundFocusOn	: Zeigt an wenn der Slider den Fokus besitzt
Height	: Die Höhe des Controls.
HelpContext	: Die ID für die kontextbezogene Hilfe zu diesem Steuerelement
HelpKeyword	: Das Schlüsselwort für die kontextbezogene Hilfe zu diesem Steuerelement
HelpType	: Legt fest, ob die kontextsensitive Hilfe nach numerischer ID oder Schlüsselwort ausgewählt wird
Hint	: Der Text, der im Hinweisenster für das Steuerelement angezeigt werden soll
JumpToPosition	: Wenn aktive springt der Knopf bei Klick in den Track auf Position
Knob1Settings	: Die Eigenschaften der Schiebeknöpfe

<code>Knob1Settings.ColorGradient</code>	: Die Richtung des Farbverlaufs
<code>Knob1Settings.Design</code>	: Das Aussehen des Knopfes
<code>Knob1Settings.DesignColor</code>	: Die Farbe der Border bzw. Zeichen im Knopf
<code>Knob1Settings HoverEndColor</code>	: Die Endfarbe eines Hoverereignisses
<code>Knob1Settings HoverOn</code>	: Ermöglicht das Ein- oder Ausblenden eines Hoverereignisses
<code>Knob1Settings HoverStartColor</code>	: Die Startfarbe eines Hoverereignisses
<code>Knob1Settings.KnobColorEnd</code>	: Die Endfarbe des Knopfes (für Farbverlauf)
<code>Knob1Settings.KnobColorStart</code>	: Die Startfarbe des Knopfes (für Farbverlauf)
<code>Knob1Settings.KnobPosition</code>	: Die Position des Knopfes im Slider
<code>Knob1Settings.KnobStyle</code>	: Die Form des Knopfes
<code>Knob1Settings.Visible</code>	: zeigt den Knopf
<code>Knob2Settings</code>	: Die Eigenschaften der Schiebeknöpfe
<code>Knob2Settings.ColorGradient</code>	: Die Richtung des Farbverlaufs
<code>Knob2Settings.Design</code>	: Das Aussehen des Knopfes
<code>Knob2Settings.DesignColor</code>	: Die Farbe der Border bzw. Zeichen im Knopf
<code>Knob2Settings HoverEndColor</code>	: Die Endfarbe eines Hoverereignisses
<code>Knob2Settings HoverOn</code>	: Ermöglicht das Ein- oder Ausblenden eines Hoverereignisses
<code>Knob2Settings HoverStartColor</code>	: Die Startfarbe eines Hoverereignisses
<code>Knob2Settings.KnobColorEnd</code>	: Die Endfarbe des Knopfes (für Farbverlauf)
<code>Knob2Settings.KnobColorStart</code>	: Die Startfarbe des Knopfes (für Farbverlauf)
<code>Knob2Settings.KnobPosition</code>	: Die Position des Knopfes im Slider
<code>Knob2Settings.KnobStyle</code>	: Die Form des Knopfes
<code>Knob2Settings.Visible</code>	: zeigt den Knopf
<code>Knob3Settings</code>	: Die Eigenschaften der Schiebeknöpfe
<code>Knob3Settings.ColorGradient</code>	: Die Richtung des Farbverlaufs
<code>Knob3Settings.Design</code>	: Das Aussehen des Knopfes
<code>Knob3Settings.DesignColor</code>	: Die Farbe der Border bzw. Zeichen im Knopf
<code>Knob3Settings HoverEndColor</code>	: Die Endfarbe eines Hoverereignisses
<code>Knob3Settings HoverOn</code>	: Ermöglicht das Ein- oder Ausblenden eines Hoverereignisses
<code>Knob3Settings HoverStartColor</code>	: Die Startfarbe eines Hoverereignisses
<code>Knob3Settings.KnobColorEnd</code>	: Die Endfarbe des Knopfes (für Farbverlauf)
<code>Knob3Settings.KnobColorStart</code>	: Die Startfarbe des Knopfes (für Farbverlauf)
<code>Knob3Settings.KnobPosition</code>	: Die Position des Knopfes im Slider
<code>Knob3Settings.KnobStyle</code>	: Die Form des Knopfes
<code>Knob3Settings.Visible</code>	: zeigt den Knopf
<code>Left</code>	: Die Clientkoordinate des linken Rands des Steuerelements
<code>Max</code>	: Der höchste Wert im Wertebereich
<code>Min</code>	: Der niedrigste Wert im Wertebereich

Orientation	: Die Orientierung des Sliders
PopupMenu	: Ein Menü das angezeigt wird, wenn Sie mit der rechten Maustaste auf dieses Steuerelement klicken
Reversed	: Max und Min werden vertauscht
RndRctRadius	: Eckendurchmesser wenn geometrische Form ist RoundRect
Scale1Settings	: Die Eigenschaften der ersten Skala
Scale1Settings.BigMarkColor	: Die Farbe der grossen Markierungen
Scale1Settings.BigMarkInterval	: Der Abstand der grossen Markierungen
Scale1Settings.BigMarksVisible	: Zeigt grosse Markierungen
Scale1Settings.LineColor	: Die Farbe der Linien in der Skala
Scale1Settings.LineWidth	: Die Dicke der Linien in der Skala
Scale1Settings.ScaleStyle	: Das Aussehen der Markierungen (ssNone macht unsichtbar)
Scale1Settings.SmallMarkColor	: Die Farbe der Markierungen in der Skala
Mit Scale1Settings.SmallMarkInterval	: Der Abstand der Markierungen in der Skala
Scale2Settings	: Die Eigenschaften der zweiten Skala
Scale2Settings.BigMarkColor	: Die Farbe der grossen Markierungen
Scale2Settings.BigMarkInterval	: Der Abstand der grossen Markierungen
Scale2Settings.BigMarksVisible	: Zeigt grosse Markierungen
Scale2Settings.LineColor	: Die Farbe der Linien in der Skala
Scale2Settings.LineWidth	: Die Dicke der Linien in der Skala
Scale2Settings.ScaleStyle	: Das Aussehen der Markierungen (ssNone macht unsichtbar)
Scale2Settings.SmallMarkColor	: Die Farbe der Markierungen in der Skala
Scale2Settings.SmallMarkInterval	: Der Abstand der Markierungen in der Skala
ShowHint	: Wenn True, wird der Hinweistext angezeigt, wenn sich die Maus über dem Steuerelement befindet
Style	: Die geometrische Form des Sliders
TabOrder	: Bestimmt die Reihenfolge der Steuerelementnavigation, wenn der Benutzer die Tabulatortaste drückt
TabStop	: Ermöglicht dem Benutzer das Navigieren zu diesem Steuerelement durch Drücken der Tabulatortaste
TextSettings	: Die Eigenschaften des Textlabels
TextSettings.AdInPercent	: Zeigt den Wert des Sliders im TextLabel in Prozent
TextSettings.AutoAd	: Zeigt den Wert des Sliders im TextLabel
TextSettings.BackgrdColor	: Die Hintergrundfarbe des Textlabels (clNone für keine Farbe)
TextSettings.BorderColor	: Die Farbe des Rahmens (clNone für unsichtbar)
TextSettings.BorderWidth	: Die Dicke des Rahmens
TextSettings.CaptionAlignment	: Ausrichtung des Textes in der Caption (Links,Mitte,Rechts)
TextSettings.CaptionBelow	: Schreibt die Buchstaben untereinander (nur aktive mit poLeft und poRight)

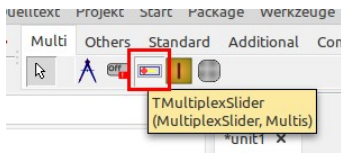
<code>TextSettings.CaptionHorMargin</code>	: Der horizontale Abstand des Textes im Textrechteck (nur wirksam mit <code>taLeftJustify</code>)
<code>TextSettings.CaptionLayout</code>	: Ausrichtung des Textes in der Caption (Oben,Mitte,Unten)
<code>TextSettings.CaptionVerMargin</code>	: Der vertikale Abstand des Textes im Textrechteck (nur wirksam mit <code>tlTop</code>)
<code>TextSettings.Font</code>	: Die Schrift die für die Textanzeige verwendet werden soll
<code>TextSettings.Height</code>	: Die Höhe des TextLabels (nur wirksam mit <code>poTop/poBottom</code>)
<code>TextSettings.Position</code>	: Die Position des Textlabels (<code>poNone</code> für unsichtbar)
<code>TextSettings.PostCaption</code>	: Die Text hinter dem Wert im Textlabel
<code>TextSettings.PreCaption</code>	: Die Text vor dem Wert im Textlabel
<code>TextSettings.Style</code>	: Die geometrische Form des Textlabels
<code>TextSettings.Width</code>	: Die Breite des TextLabels (nur wirksam mit <code>poLeft/poRight</code>)
<code>Top</code>	: Die Clientkoordinate des oberen Rands des Steuerelements
<code>TrackSettings</code>	: Die Eigenschaften des Schiebeschlitzes
<code>TrackSettings.ExtraColor</code>	: Die Farbe der zusätzlichen Farbe (<code>clNone</code> für unsichtbar)
<code>TrackSettings.ExtraRangeMax</code>	: Maximalwert der zusätzlichen Farbe
<code>TrackSettings.ExtraRangeMin</code>	: Minimalwert der zusätzlichen Farbe
<code>TrackSettings.SelRangeColor</code>	: Die Farbe des selektierten Bereichs (<code>clNone</code> für unsichtbar)
<code>TrackSettings.TrackColor</code>	: Die Farbe des Schlitzes
<code>ValueDisplaySettings</code>	: Die Eigenschaften der Wertanzeige
<code>ValueDisplaySettings.BorderColor</code>	: Die Farbe des Rahmens (<code>clNone</code> für unsichtbar)
<code>ValueDisplaySettings.BorderWidth</code>	: Die Dicke des Rahmens
<code>ValueDisplaySettings.ColorEnd</code>	: Die Endfarbe der Anzeige (für Farbverlauf)
<code>ValueDisplaySettings.ColorGradient</code>	: Die Richtung des Farbverlaufs
<code>ValueDisplaySettings.ColorStart</code>	: Die Startfarbe der Anzeige (für Farbverlauf)
<code>ValueDisplaySettings.Font</code>	: Die Schrift die für der Anzeige verwendet werden soll
<code>ValueDisplaySettings.InPercent</code>	: Zeigt den Wert in Prozent
<code>ValueDisplaySettings.Position</code>	: Die Position des Displays im Slider, <code>vdsNone</code> für Unsichtbar
<code>ValueDisplaySettings.Style</code>	: Die geometrische Form der Anzeige, <code>vdsNone</code> macht keine Form
<code>ValueDisplaySettings.X</code>	: Beeinflußt die Position, nur mit <code>vdpXY,vdpAboveRight,vdpBelowLeft</code>
<code>ValueDisplaySettings.Y</code>	: Beeinflußt die Position, nur mit <code>vdpXY,vdpAboveRight,vdpBelowLeft</code>
<code>Visible</code>	: Ermöglicht das Ein- oder Ausblenden des Steuerelements und aller seiner untergeordneten Elemente
<code>Width</code>	: Die Breite des Controls.

Ereignisse

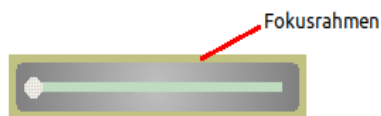
OnChange	: Liefert den Wert von Knob1 (als integer)
OnChangeStr	: Liefert den Wert von Knob1 als String
OnChange3x	: Liefert die Werte von Knob1,2,3 (als integer)
OnChange3xPlus	: Liefert die Werte von Knob1,2,3 (als integer) plus den KnobIndex
OnChangeStr3x	: Liefert die Werte von Knob1,2,3 als String
OnClick	: Benachrichtigungshandler für Mausklicks
OnDragDrop	: Dieser Handler bestimmt die Aktion beim Ablegen auf diesem Steuerelement in einem Drag-Drop-Vorgang
OnDragOver	: Ereignishandler für ein Steuerelement, das über dieses Steuerelement gezogen wird
OnEndDrag	: Benachrichtigungshandler für das Ende eines Ziehvorgangs
OnEnter	: Handler für die Steuerung, die den Fokus erhält
OnExit	: Handler für die Steuerung, die den Fokus verliert
OnKeyDown	: Handler für gedrückte Tastaturtaste
OnKeyPress	: Handler für ein vom Benutzer eingegebenes Zeichen
OnKeyUp	: Handler für Tastaturtaste freigegeben
OnMouseDown	: Ereignishandler für das Drücken der Maustaste
OnMouseEnter	: Ereignishandler für das Betreten des Bereichs des Steuerelements mit der Maus
OnMouseLeave	: Ereignishandler für Maus, die den Bereich des Steuerelements verlässt
OnMouseMove	: Ereignishandler für die Mausbewegung innerhalb des Steuerelements
OnMouseUp	: Ereignishandler für das Loslassen der Maustaste
OnMouseUp3xPlus	: Liefert die Werte von Knob1,2,3 (als integer) plus den KnobIndex
OnMouseWheelDown	: Ereignisbehandler für die Abwärtsbewegung des Mausekkrads
OnMouseWheelUp	: Ereignisbehandler für die Aufwärtsbewegung des Mausekkrads
OnStartDrag	: Ereignishandler für den Start eines Ziehvorgangs

Beschreibung

Den MultiplexSlider findest du im Reiter Multi:

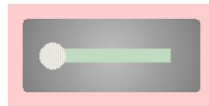


Es ist wichtig zu wissen das der MultiplexSlider von einem Fokusrahmen umgeben ist. Wie man hier sieht besitzt der fokussierte MultiplexSlider einen olivgrünen Rahmen. Das bedeutet im Umkehrschluss das der eigentliche Slider um den Rahmen kleiner ist.

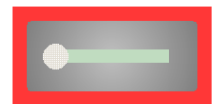


Mit der Eigenschaft [FocusColor](#) kann man die Farbe des Fokusrahmens einstellen. Mit [FocusAlphaBValue](#) kann die Transparenz des Fokusrahmens geregelt werden. Der Wert 0 bedeutet durchsichtig und 255 undurchsichtig.

Wert 50:



Wert 200:



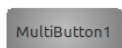
[FocusFrameWidth](#) bestimmt die Dicke des Rahmens.

Setzt man [FocusFrameOn](#) auf false dann bleibt der Rand erhalten jedoch wird die Fokussierung nicht farblich angezeigt.

Mit [ForegroundFocusOn](#) besitzt der fokussierte MultiplexSlider eine farbige Border. Die Farbe der Border kann man mit [FocusColor](#) beeinflussen. Diese Einstellung macht nur Sinn wenn [FocusFrameOn](#) auf false gesetzt wurde!



Hier zeigt die rote Border das der Slider den Fokus besitzt.



Um die Hintergrundfarbe des MultiplexSliders zu verändern benötigst du die Eigenschaften [ColorStart](#), [ColorEnd](#) und [ColorGradient](#). Um einen einfarbigen MultiplexSlider zu bekommen muss ColorStart und ColorEnd gleich sein. Ansonsten bestimmt die Zusammensetzung aus den drei Eigenschaften das farbliche Aussehen.

Hier ist StartColor clBlack und EndColor clFuchsia.

gcSpread



gcRadiant



gcAlternate



gcHorizontal



gcVertical

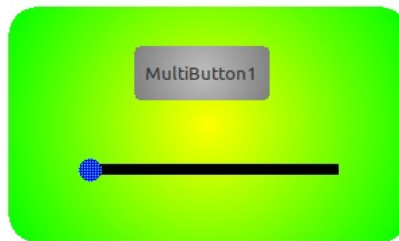


Alternate setzt abwechselnd ein Pixel je auf Start- und EndColor.

Tipp: Setzt man ColorStart und ColorEnd auf clNone wird bekommt der Hintergrund die Farbe des Parent.



Sitzt der Slider auf einem MultiPanel nimmt er dessen Farbe an (bei clNone).



Mit der Eigenschaft [Style](#) stellt man die gewünschte geometrische Form des MultiplexSliders ein.

mssRect



Ist mssRoundRect eingestellt kann man mit der Eigenschaft [RndRectRadius](#) den Durchmesser der Eckenrundung einstellen.

mssRoundRect



RndRectRadius := 10;

mssRoundRect



RndRectRadius := 40;

Möchte man den MultiplexSlider mit einem farbigen Rand ausstatten wählt man bei [BorderColor](#) die Farbe des Randes. Mit clNone wird kein Rand gezeichnet. Die Breite des Randes wird mit [BorderWidth](#) eingestellt.

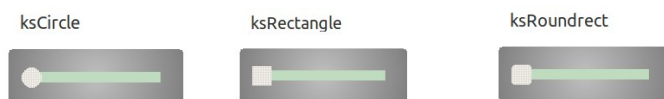


Die Eigenschaften der Schiebeknöpfe findet man unter [Knob1Settings](#), [Knob2Settings](#) und [Knob3Settings](#).

Während der 1. Knob standardmäßig sichtbar ist sind der Zweite und der Dritte unsichtbar. Das Sichtbar bzw. Unsichtbar machen geschieht mit [Knob1Settings.Visible](#), [Knob2Settings.Visible](#) bzw. [Knob3Settings.Visible](#).



Mit der Eigenschaft [Knob1Settings.KnobStyle](#), [Knob2Settings.KnobStyle](#) bzw. [Knob3Settings.KnobStyle](#) stellt man die gewünschte geometrische Form des Knobs ein.



Um die Hintergrundfarbe des Knobs zu verändern benötigst du die Eigenschaften

[Knob1Settings.KnobColorStart](#), [Knob2Settings.KnobColorStart](#) bzw. [Knob3Settings.KnobColorStart](#)
[Knob1Settings.KnobColorEnd](#), [Knob2Settings.KnobColorEnd](#) bzw. [Knob3Settings.KnobColorEnd](#) und
[Knob1Settings.ColorGradient](#), [Knob2Settings.ColorGradient](#) bzw. [Knob3Settings.ColorGradient](#)

Um einen einfarbigen Knob zu bekommen muss ColorStart und ColorEnd gleich sein. Ansonsten bestimmt die Zusammensetzung aus den drei Eigenschaften das farbliche Aussehen.

Hier ist StartColor cBlack und EndColor cFuchsia.



Alternate setzt abwechselnd ein Pixel je auf Start- und EndColor.



Standard mäßig steht [Knob1Settings.HoverOn](#), [Knob2Settings.HoverOn](#) und [Knob3Settings.HoverOn](#) auf true. Das bedeutet das sich bei einem Hoverereigniss (die Maus bewegt sich über dem Knob) mit [Knob1Settings.HoverStartColor](#), [Knob2Settings.HoverStartColor](#) bzw. [Knob3Settings.HoverStartColor](#) und [Knob1Settings.HoverEndColor](#), [Knob2Settings.HoverEndColor](#) bzw. [Knob3Settings.HoverEndColor](#) das Aussehen nach Wunsch verändern lässt. Möchte man dies nicht setzt man HoverOn auf false.

[Knob1-3Settings.HoverEndColor](#) : Die Endfarbe eines Hoverereignisses

[Knob1-3Settings.HoverOn](#) : Ermöglicht das Ein- oder Ausblenden eines Hoverereignisses

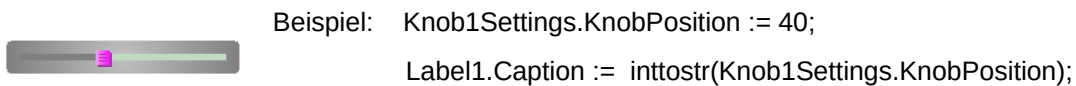
[Knob1-3Settings.HoverStartColor](#) : Die Startfarbe eines Hoverereignisses

Eine weitere Möglichkeit den Knobs visuelle Effekte hinzuzufügen bieten die Eigenschaften [Knob1Settings.Design](#), [Knob2Settings.Design](#) bzw. [Knob3Settings.Design](#) und [Knob1Settings.DesignColor](#), [Knob2Settings.DesignColor](#) bzw. [Knob3Settings.DesignColor](#).

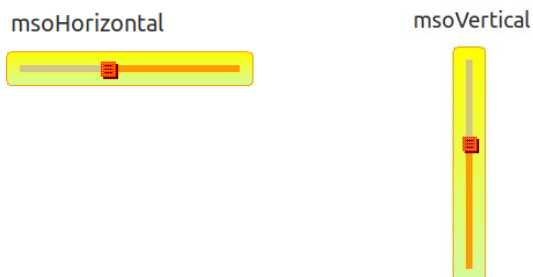
Bei den Beispielen ist DesignColor cYellow. Bei kdDefault wird DesignColor nicht verwendet.



Mit der Eigenschaft [Knob1Settings.KnobPosition](#), [Knob2Settings.KnobPosition](#) bzw. [Knob3Settings.KnobPosition](#) kann die Position des Knobs innerhalb des Sliders gesetzt oder auch abgefragt werden.

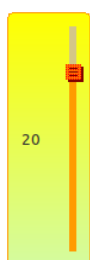


Die Ausrichtung des MultiplexSliders wird mittels [Orientation](#) erreicht.

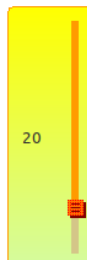
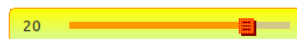


Nutzt man die Eigenschaft [Reversed](#) werden der max. Wert und der min. Wert vertauscht

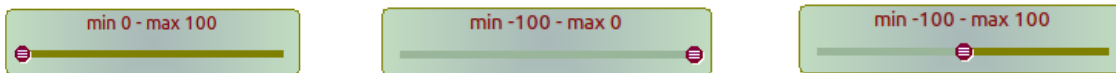
Reversed := false;



Reversed := true;



Die Eigenschaft **Max** stellt den höchsten Wert im Wertebereich des MultiplexSliders dar und **Min** den Niedrigsten. Hier mit Knob Position 0.



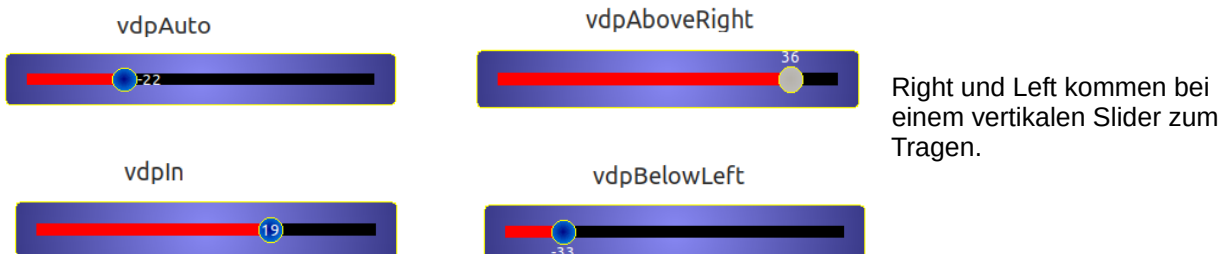
Ist man sich über den Wertebereich des MultiplexSliders noch nicht im klaren oder benötigt man einen dynamischen Wertebereich bedient man sich der Eigenschaften **AutoRangeNegative** bzw. **AutoRangePositive** und **AutoRangeValue**.

Möchte man zum Beispiel das sich der max. Wert beim Erreichen um 20 Einheiten erhöht stellt man folgendes ein:

AutoRangeNegative	<input type="checkbox"/> (False)
AutoRangePositive	<input checked="" type="checkbox"/> (True)
AutoRangeValue	20

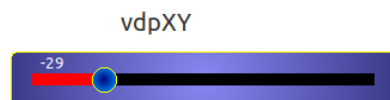
Erreicht nun der Knob im Slider den max. Wert wird dieser um 20 erhöht. Bei einem erneuten Erreichen von Max erfolgt wieder eine Erhöhung. Bei Unterschreiten vom originalen Maxwert wird der ursprüngliche Wertebereich wieder hergestellt.

Möchte man den eingestellten Wert des Sliders anzeigen kann man sich der Eigenschaften der **ValueDisplaySettings** bedienen. Um die Anzeige zu aktivieren muss bei **ValueDisplaySettings.Position** ein anderer Typ als **vdpNone** gewählt werden.



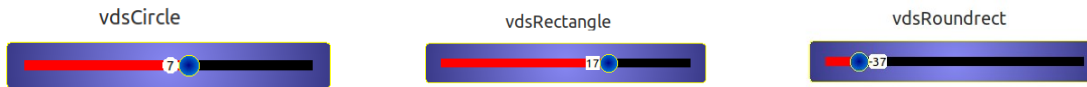
Mit den zusätzlichen Eigenschaften **ValueDisplaySettings.X** und **ValueDisplaySettings.Y** lässt sich die Position der Anzeige bei **vdpAboveRight**, **vdpBelowLeft** beeinflussen.

Bei **vdpXY** wird die Position der Anzeige mit diesen Eigenschaften frei im Slider gewählt.

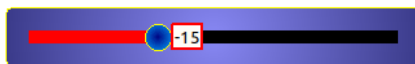


Soll die Anzeige einen farbigen Hintergrund bekommen realisiert man dies mit den Eigenschaften `ValueDisplaySettings.Style`, `ValueDisplaySettings.ColorStart`, `ValueDisplaySettings.ColorEnd` und `ValueDisplaySettings.ColorGradient`.

Die Form des Hintergrundes wird mit `Style` eingestellt. Wählt man `vdsNone` wird kein Hintergrund gezeichnet.



`ColorStart`, `ColorEnd` und `ColorGradient` funktionieren analog wie das Einstellen des Hintergrundes. Siehe 31. Einen farbigen Rand um die Anzeige erstellt man mit `BorderColor` (`clNone` für unsichtbar) und `Border.Width`.

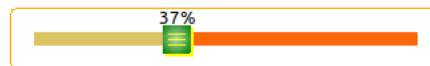


`ValueDisplaySettings.BorderColor := clRed;`

`ValueDisplaySettings.BorderWidth := 2;`

Mit `ValueDisplaySettings.Font` kann man wie üblich alle Attribute der Schrift wie zum Beispiel Farbe und Größe etc. einstellen.

Wer die Wertanzeige in Prozent benötigt kann einfach `ValueDisplaySettings.InPercent` auf `true` stellen.



Alle verfügbaren Eigenschaften des Tracks (Schiebeschlitzes) lassen sich in den `TrackSettings` einstellen.



`TrackSettings.TrackColor`

: Die Farbe des Schlitzes

`TrackSettings.SelRangeColor`

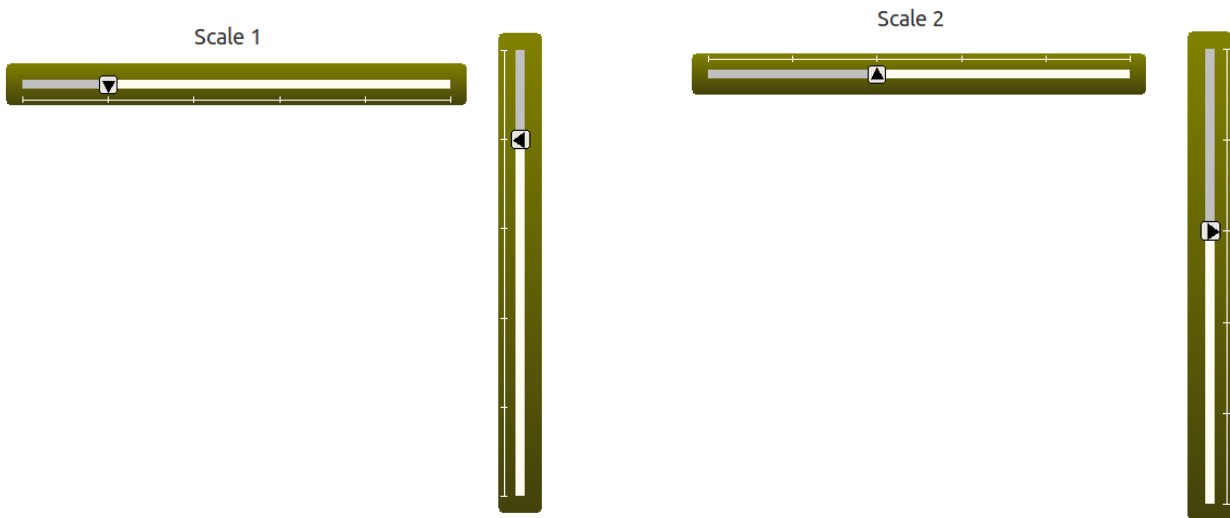
: Die Farbe des
selektierten Bereichs
(`clNone` für unsichtbar)

Möchte man noch eine zusätzliche Farbe darstellen kann man dies indem man der Eigenschaft `TrackSettings.ExtraColor` eine Farbe zuweist (`clNone` für unsichtbar). Mit den Eigenschaften `TrackSettings.ExtraRangeMax` und `TrackSettings.ExtraRangeMin` bestimmt man die Grenzen dieser Farbe.



Dieser Slider besitzt einen Wertebereich von 0-100. `ExtraRangeMin` ist hier auf 80 und `Max` auf 100 eingestellt. `ExtraColor` ist `clRed`.

Der MultiplexSlider besitzt zwei integrierte Skalen. Die Einstellungen hierfür findet man in den Eigenschaften von [Scale1Settings](#) und [Scale2Settings](#). Stellt man [Scale1Settings.ScaleStyle](#) bzw. [Scale2Settings.ScaleStyle](#) auf `ssDash` oder `ssCircle` wird die jeweilige Skala sichtbar (`ssNone` macht unsichtbar).



[Scale1Settings.ScaleStyle](#) bzw. [Scale2Settings.ScaleStyle](#):



[Scale1Settings.LineColor](#), [Scale2Settings.LineColor](#) : hier `clRed`
[Scale1Settings.LineWidth](#), [Scale2Settings.LineWidth](#) : Die Dicke der Linien der Skala, hier 2
[Scale1Settings.SmallMarkColor](#), [Scale2Settings.SmallMarkColor](#) : hier `clLime`

Mit [Scale1Settings.SmallMarkInterval](#) bzw. [Scale2Settings.SmallMarkInterval](#) lässt sich der Abstand der kleinen Markierungen zueinander einstellen.



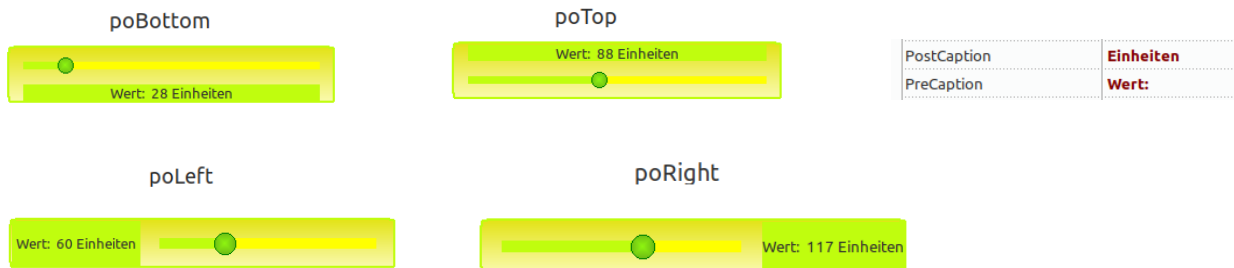
Hier Wertebereich 0-100. Interval 10.

Zur besseren Kenntlichkeit können zusätzlich Dicke Markierungen eingeblendet werden. Dies geschieht mit [Scale1Settings.BigMarksVisible](#) bzw. [Scale2Settings.BigMarksVisible](#) (auf `true`).

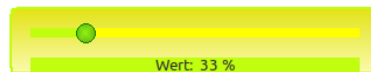
Hier ist [Scale1Settings.BigMarkColor](#) bzw. [Scale2Settings.BigMarkColor](#) `clYellow` und [Scale1Settings.BigMarkInterval](#) bzw. [Scale2Settings.BigMarkInterval](#) 20.



Möchte man das eingebaute Textlabel benutzen so muss man unter [TextSettings](#) die Eigenschaft [TextSettings.Position](#) auf einen anderen Wert als poNone stellen (poNone macht unsichtbar). Stellt man [TextSettings.AutoAd](#) auf true wird der Wert automatisch im Textlabel angezeigt. Will man weiteren Text ins Label einfügen geschieht dies mit den Eigenschaften [TextSettings.PreCaption](#) und [TextSettings.PostCaption](#). Was bei PreCaption eingegeben wird erscheint vor dem Wert, das bei PostCaption dahinter.



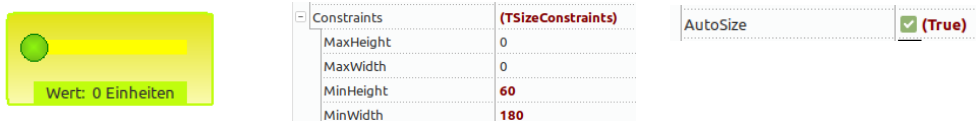
Mit der Eigenschaft [TextSettings.AdInPercent](#) wird der Wert des Sliders in Prozent angezeigt.



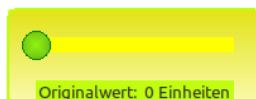
Sollte das Textlabel zu kurz sein lässt sich die Breite mit [TextSettings.Width](#) anpassen (nur wirksam mit poLeft/poRight).

Das gleiche gilt für die Höhe dort muss man [TextSettings.Height](#) verwenden (nur wirksam mit poTop/poBottom).

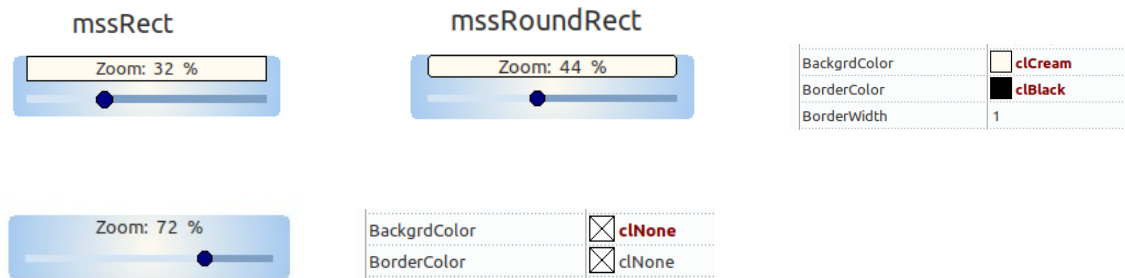
Die Eigenschaft [AutoSize](#) verlängert oder kürzt den gesamten Slider auf die notwendige Größe des TextLabels. Wird der Slider verkürzt macht die Verwendung von [Constraints](#) Sinn.



Verlängert sich aus irgendeinem Grund der Text im Textlabel passt sich hier die Länge an:



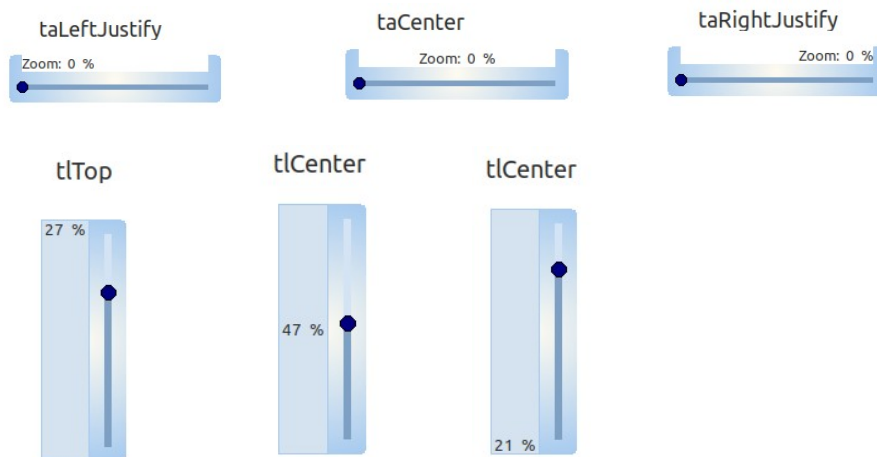
Der Hintergrund des Textlabels kann mit den Eigenschaften [TextSettings.BackgrdColor](#) (clNone für keine Farbe), [TextSettings.BorderColor](#) (clNone für unsichtbar), [TextSettings.BorderWidth](#) und [TextSettings.Style](#) eingestellt werden.



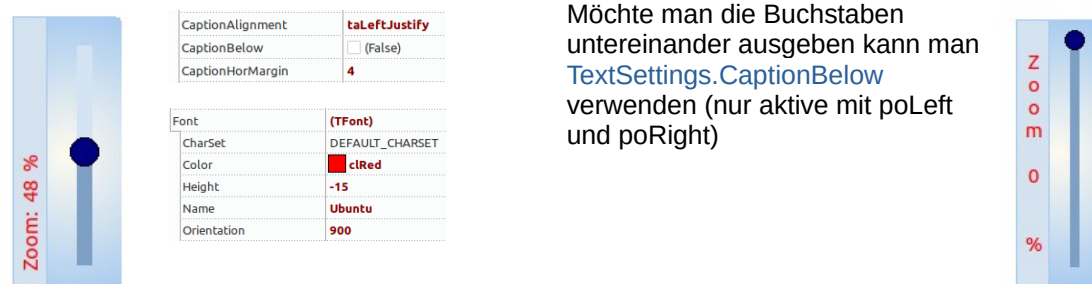
Die Position des Textes innerhalb des TextLabels wird durch die Eigenschaften

[TextSettings.CaptionAlignment](#) : Ausrichtung des Textes in der Caption (Links,Mitte,Rechts)
[TextSettings.CaptionHorMargin](#) : Der horizontale Abstand des Textes im Textrechteck (nur wirksam mit taLeftJustify)
[TextSettings.CaptionLayout](#) : Ausrichtung des Textes in der Caption (Oben,Mitte,Unten)
[TextSettings.CaptionVerMargin](#) : Der vertikale Abstand des Textes im Textrechteck (nur wirksam mit tlTop)

bestimmt.

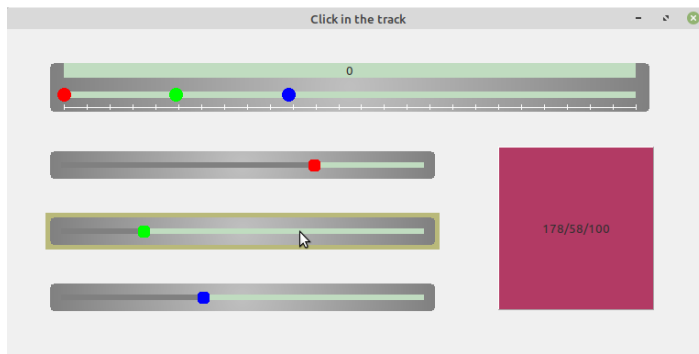


Möchte man in einem vertikalen Slider den Text von unten nach oben schreiben und die Schrift verändern geschieht dies mit [TextSettings.Font](#)



Möchte man die Buchstaben untereinander ausgeben kann man [TextSettings.CaptionBelow](#) verwenden (nur aktive mit poLeft und poRight)

Ist die Eigenschaft `JumpToPosition` true springt der Schiebeknopf bei einem Klick in den Track an diese Position. Sind zwei oder drei Knöpfe sichtbar springt der zuletzt bewegte Knopf.



Der Knopf springt bei Klick hier an die Stelle des Mauszeigers.

TMultiSeperator

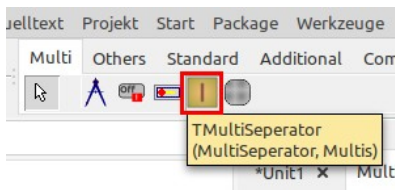
Eigenschaften

Action	: Die dem Steuerelement zugeordnete Aktion
Align	: Gibt die Platzierung des Steuerelements innerhalb seines übergeordnetenElements an
Anchors	: Der Satz von Ankerdefinitionen für dieses Steuerelement
BackgrdImage	: Beeinhaltet das Bild das im Control ausgegeben wird, nur bei gcBitmap
BorderColor	: Die Farbe des Rahmens, clNone macht unsichtbar
BorderSpacing	: Bestimmt den inneren und äußeren Rahmenabstand für dieses Steuerelement
BorderWidth	: Die Dicke des Rahmens
ColorEnd	: Die Endfarbe des Hintergrundes (für Farbverlauf),clNone macht unsichtbar
ColorGradient	: Die Richtung des Farbverlaufs
ColorStart	: Die Startfarbe des Hintergrundes (für Farbverlauf),clNone macht unsichtbar
Constraints	: Die minimale und maximale Breite und Höhe für das Steuerelement
Cursor	: Die geometrische Form des Mauszeigers wenn die Maus über diesem Steuerelement ist
Enabled	: Legt fest, ob das Steuerelement auf Maus- oder Tastatureingaben reagiert
Height	: Die Höhe des Controls
HelpContext	: Die ID für die kontextbezogene Hilfe zu diesem Steuerelement
HelpKeyword	: Das Schlüsselwort für die kontextbezogene Hilfe zu diesem Steuerelement
HelpType	: Legt fest, ob die kontextsensitive Hilfe nach numerischer ID oder Schlüsselwort ausgewählt wird
Hint	: Der Text, der im Hinweifenster für das Steuerelement angezeigt werden soll
Left	: Die Clientkoordinate des linken Rands des Steuerelements
LineSettings	: Die Einstellungen der Linie im Seperator
LineSettings.Color	: Die Farbe der Linie
LineSettings.Design	: Die Anzahl der Linien
LineSettings.EndCap	: Die Form der Linienenden
LineSettings.LinesLength	: Die Länge der Linien bei Strich-, Strichpunkt ...
LineSettings.LinesSpace	: Die Länge der Zwischenräume bei Strich-, Strichpunkt ...
LineSettings.Margin	: Der Abstand der Linie zur Border

LineSettings.PenWidth	: Die Dicke der Linie
LineSettings.Style	: Der Stil der Linie, cpsNull macht unsichtbar
Orientation	: Die Orientierung des Separators
RndRctRadius	: Eckendurchmesser wenn geometrische Form ist RoundRect
ShowHint	: Wenn True, wird der Hinweistext angezeigt, wenn sich die Maus über dem Steuerelement befindet
Style	: Die geometrische Form des Separators
Top	: Die Clientkoordinate des oberen Rands des Steuerelements
Visible	: Ermöglicht das Ein- oder Ausblenden des Steuerelements und aller seiner untergeordneten Elemente
Width	: Die Breite des Controls

Beschreibung

Den MultiSeparator findest du im Reiter Multi:



Der MultiSeparator ist eine Komponente die zur optischen Gestaltung gedacht ist. Gerne auch als Trenner zwischen anderen Controls.

Mit der Eigenschaft [Orientation](#) kann die Lage des Separators zwischen Horizontal und Vertikal umgestellt werden.

mSPVertical



mSPHorizontal



Um die Hintergrundfarbe des MultiSeparators zu verändern benötigst du die Eigenschaften [ColorStart](#), [ColorEnd](#) und [ColorGradient](#). Um einen einfarbigen MultiSeparator zu bekommen muss ColorStart und ColorEnd gleich sein. Ansonsten bestimmt die Zusammensetzung aus den drei Eigenschaften das farbliche Aussehen.

Hier ist StartColor clMaroon und EndColor \$0000B5FF.

gcSpread



gcRadiant



gcAlternate



Alternate setzt abwechselnd ein Pixel je auf Start- und EndColor.

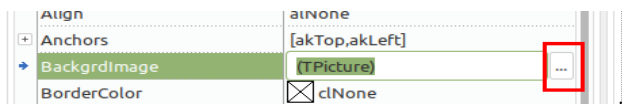
gcHorizontal



gcVertical



Um den Typ gcBitmap nutzen zu können muss vorab eine Bitmap mit der Eigenschaft [BackgrdImage](#) geladen werden. Dazu im OI bei BackgrdImage in dem Kästchen mit den drei Punkten klicken.



Es öffnet sich der Bild Laden Dialog. Dort auf Laden und eine Bitmap (es funktioniert auch mit png) auswählen. Zuletzt bei ColorGradient gcBitmap auswählen und die Bitmap wird als Hintergrund verwendet.



Mit der Eigenschaft [Style](#) stellt man die gewünschte geometrische Form des MultiSeperators ein.



Ist mspRoundRect eingestellt kann man mit der Eigenschaft [RndRctRadius](#) den Durchmesser der Eckenrundung einstellen.

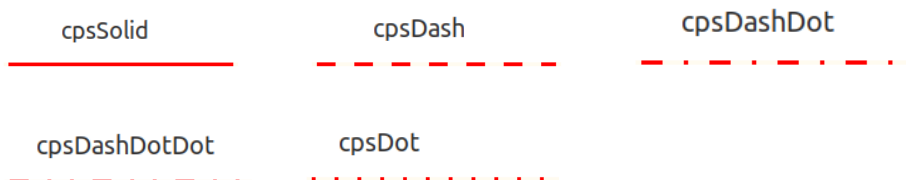
Stellt man [BorderColor](#) ungleich clNone erscheint eine Border. Mit [BorderWidth](#) kann dessen Dicke eingestellt werden.



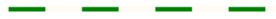
Mit den Einstellungen unter [LineSettings](#) kann in den Seperator eine Linie gezeichnet werden.



Durch Einstellen eines [LineSettings.Style](#) (cpsNull macht unsichtbar) wird die Linie sichtbar.



Mit `LineSettings.Color` kann die Farbe der Linie beeinflusst werden. `LineSettings.PenWidth` stellt die Dicke der Linie ein.



Die Länge der Unterbrechung zwischen den Linien kann mit `LineSettings.LinesSpace` und die Länge der Linien mit `LineSettings.LinesLength` eingestellt werden.



Mit `LineSettings.EndCap` lässt sich das Linienende anpassen:



Zum besseren Verständniss der Linienenden:



Möchte man den Abstand der Linie links und rechts zum Rand gleich einstellen geschieht dies mit `LineSettings.Margin`.



Mit `LineSettings.Design` kann eine Doppellinien erzeugt werden.



TMultiLayer

Eigenschaften

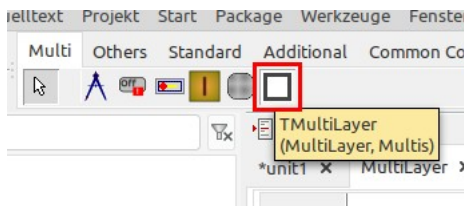
Align	: Gibt die Platzierung des Steuerelements innerhalb seines übergeordnetenElements an
Anchors	: Der Satz von Ankerdefinitionen für dieses Steuerelement
BorderSpacing	: Bestimmt den inneren und äußeren Rahmenabstand für dieses Steuerelement
Color	: Die Hintergrundfarbe des Controls
Constraints	: Die minimale und maximale Breite und Höhe für das Steuerelement
Cursor	: Die Form des Mauszeigers, wenn sich die Maus über diesem Steuerelement befindet
GroupIndex	: Der Index der Gruppe zu der der MultiLayer gehört
Height	: Die Höhe des Controls
HelpContext	: Die ID für die kontextbezogene Hilfe zu diesem Steuerelement
HelpKeyword	: Das Schlüsselwort für die kontextbezogene Hilfe zu diesem Steuerelement
HelpType	: Legt fest, ob die kontextsensitive Hilfe nach numerischer ID oder Schlüsselwort ausgewählt wird
Hint	: Der Text, der im Hinweisfenster für das Steuerelement angezeigt werden soll
Left	: Die Clientkoordinate des linken Rands des Steuerelements
Top	: Die Clientkoordinate des oberen Rands des Steuerelements
Visible	: Ermöglicht das Ein- oder Ausblenden des Steuerelements und aller seiner untergeordneten Elemente auch zur Designzeit
Width	: Die Breite des Controls.

Öffentliche Prozeduren

procedure **InvalidateMultiPanel**;

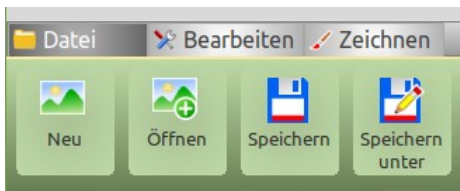
Beschreibung

Den MultiLayer findest du im Reiter Multi:



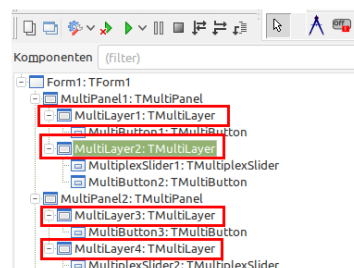
Der MultiLayer ist für ein vereinfachtes designing beim Zusammenklicken von Formularen gedacht. Alle Kind-Controls von TMultiLayer werden unsichtbar wenn TMultiLayer auf **Visible** := false gesetzt wird. Sitzt der MultiLayer auf einem MultiPanel wird dessen Farbverlauf als Hintergrund angenommen.

Tipp: Setzt man einen MultiLayer auf einen MultiLayer wird als Parent das erste dahinter liegende TWinControl genommen!



Hier sitzt auf einem MultiPanel für jeden Reiter ein TMultiLayer. Durch umschalten von Visible := true auf false lässt sich zur Designzeit jeder Reiter nach Wunsch gestalten.

Eine weitere Möglichkeit die Layer während der Designzeit zu schalten besteht darin im Treeview des Objektinspektors auf den MultiLayer **doppelt** zu klicken der sichtbar werden soll.



Besitzen MultiLayer den gleichen Parent und sollen aber unabhängig von einander geschaltet werden so muss der **GroupIndex** auf verschiedene Werte gestellt werden.

Sollte bei irgendeiner Konstellation der Hintergrund des MultiPanel bzw. weiterer MultiControls nicht neu gezeichnet werden kann mit **InvalidateMultiPanel** ein Neuzeichnen angestoßen werden.

TMultiRadioGroup

Eigenschaften

Action	: Die dem Steuerelement zugeordnete Aktion
Align	: Gibt die Platzierung des Steuerelements innerhalb seines übergeordnetenElements an
Anchor	: Der Satz von Ankerdefinitionen für dieses Steuerelement
AutoSize	: Ermöglicht die automatische Anpassung der Größe der Kontrolle an ihren Inhalt
BidiMode	: Anpassung (von Textsteuerelementen) in bidirektionalen Leseumgebungen
BorderColor	: Die Farbe des Rahmens
BorderMargin	: Der Abstand des Rahmens zur Außenkante
BorderWidth	: Die Dicke des Rahmens
BorderSpacing	: Bestimmt den inneren und äußeren Rahmenabstand für dieses Steuerelement
Caption	: Die Überschrift der Radiogroup
CaptionLeft	: Die Koordinate des linken Randes der Beschriftung
CaptionTop	: Die Koordinate des oberen Randes der Beschriftung
ColorEnd	: Die Endfarbe der RadioGroup (für Farbverlauf)
ColorGradient	: Die Richtung des Farbverlaufs
ColorStart	: Die Startfarbe der RadioGroup (für Farbverlauf)
Constraints	: Die minimale und maximale Breite und Höhe für das Steuerelement
Cursor	: Die geometrische Form des Mauszeigers wenn die Maus überdiesem Steuerelement ist
DragCursor	: Die Cursorform, die angezeigt wird, während das Steuerelement gezogen wird
DragKind	: Der Vorgang, wenn das Steuerelement gezogen wird - Drag or Dock
DragMode	: Ermöglicht dem Benutzer, das Steuerelement zu ziehen
Enabled	: Legt fest, ob das Steuerelement auf Maus- oder Tastatureingaben reagiert
FocusAlphaBValue	: Wie transparent der Fokusrahmen ist (0=transparent, 255=undurchsichtig)
FocusColor	: Die Farbe des Fokusrahmens/Foregroundfocus wenn das Control den Fokus hat
FocusFrameOn	: Schaltet den Fokusrahmen ein und aus

<code>FocusFrameWidth</code>	: Die Dicke des Fokus-Rahmens
<code>Font</code>	: Die Schrift die für die Textanzeige der Caption verwendet werden soll
<code>ForegroundFocusOn</code>	: Zeigt an wenn der Slider den Fokus besitzt, schaltet ein aus
<code>GroupIndex</code>	: Der Index der Gruppe zu der die MultiRadioGroup gehört
<code>Height</code>	: Die Höhe des Controls
<code>HelpContext</code>	: Die ID für die kontextbezogene Hilfe zu diesem Steuerelement
<code>HelpKeyword</code>	: Das Schlüsselwort für die kontextbezogene Hilfe zu diesem Steuerelement
<code>HelpType</code>	: Legt fest, ob die kontextsensitive Hilfe nach numerischer ID oder Schlüsselwort ausgewählt wird
<code>Hint</code>	: Der Text, der im Hinweisfenster für das Steuerelement angezeigt werden soll
<code>Left</code>	: Die Clientkoordinate des linken Rands des Steuerelements
<code>RadioButtons</code>	: Öffnet den Editor um Radiobuttons hinzuzufügen
<code>RadioButtons.Item[Index].ButtonColor</code>	: Die Farbe des Radiobuttons
<code>RadioButtons.Item[Index].ButtonSelColor</code>	: Die Farbe des selektierten Radiobuttons
<code>RadioButtons.Item[Index].Caption</code>	: Der Text den der Benutzer in den Radiobutton schreibt
<code>RadioButtons.Item[Index].CaptionAlignment</code>	: Ausrichtung des Textes in der Caption (Links,Mitte,Rechts) taLeftJustify;
<code>RadioButtons.Item[Index].CaptionHorMargin</code>	: Der horizontale Abstand des Textes im Textrechteck (nur wirksam mit taLeftJustify)
<code>RadioButtons.Item[Index].CaptionLayout</code>	: Ausrichtung des Textes in der Caption (Oben,Mitte,Unten)
<code>RadioButtons.Item[Index].CaptionVerMargin</code>	: Der vertikale Abstand des Textes im Textrechteck (nur wirksam mit tlTop)
<code>RadioButtons.Item[Index].CaptionWordbreak</code>	: Ermöglicht einen Zeilenumbruch in der Caption
<code>RadioButtons.Item[Index].Color</code>	: Die Hintergrundfarbe des RadioButtons
<code>RadioButtons.Items[Index].DisplayName</code>	: Der Name der im TreeView des Objektinspektors angezeigt wird
<code>RadioButtons.Items[Index].Enabled</code>	: Legt fest, ob das Steuerelement auf Maus- oder Tastatureingaben reagiert
<code>RadioButtons.Items[Index].Font</code>	: Die Schrift die für die Textanzeige der Caption verwendet werden soll
<code>RadioButtons.Item[Index].HoverColor</code>	: Die Farbe eines Hoverereignisses
<code>RadioButtons.Item[Index].HoverStyle</code>	: Ob ein Hoverereignis nur als Rahmen oder vollflächig gezeichnet wird
<code>RadioButtons.Item[Index].ImageIndex</code>	: Der Index eines Bildes in einer ImageList
<code>RadioButtons.Item[Index].ImageLeft</code>	: Die Koordinate der linken Ecke des Bildes
<code>RadioButtons.Item[Index].Images</code>	: Eine Liste zum Einfügen von Bildern

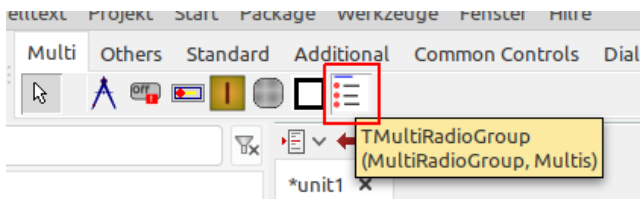
<code>RadioButtons.Item[Index].ImageTop</code>	: Die Koordinate der oberen Ecke des Bildes
<code>RadioButtons.Items[Index].ImageWidth</code>	: Die einmalige Breite aller Bilder in der Liste
<code>RadioButtons.Items[Index].ParentFont</code>	: Verwendet die Schriftart aus dem Parent, wenn aktiviert
<code>RadioButtons.Item[Index].Selected</code>	: Bestimmt ob ein Radiobutton ausgewählt ist
<code>RadioButtons.Items[Index].Tag</code>	: Kann verwendet werden, um einen ganzzahligen Wert in der Komponente zu speichern
<code>RndRctRadius</code>	: Eckendurchmesser wenn geometrische Form ist RoundRect
<code>Rows</code>	: Anzahl der Zeilen wenn Wordbreak aktive
<code>Style</code>	: Die geometrische Form der RadioGroup
<code>TabOrder</code>	: Bestimmt die Reihenfolge der Steuerelementnavigation, wenn der Benutzer die Tabulatortaste drückt
<code>TabStop</code>	: Ermöglicht dem Benutzer das Navigieren zu diesem Steuerelement durch Drücken der Tabulatortaste
<code>Tag</code>	: Kann verwendet werden, um einen ganzzahligen Wert in der Komponente zu speichern
<code>Top</code>	: Die Clientkoordinate des oberen Rands des Steuerelements
<code>Visible</code>	: Ermöglicht das Ein- oder Ausblenden des Steuerelements und aller seiner untergeordneten Elemente
<code>Width</code>	: Die Breite des Controls

Ereignisse

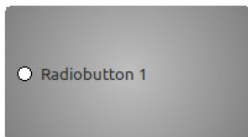
OnChange	: Liefert den Index des Radiobuttons
OnClick	: Benachrichtigungshandler für Mausklicks
OnDragDrop	: Dieser Handler bestimmt die Aktion beim Ablegen auf diesem Steuerelement in einem Drag-Drop-Vorgang
OnDragOver	: Ereignishandler für ein Steuerelement, das über dieses Steuerelement gezogen wird
OnEndDrag	: Benachrichtigungshandler für das Ende eines Ziehvorgangs
OnEnter	: Handler für die Steuerung, die den Fokus erhält
OnExit	: Handler für die Steuerung, die den Fokus verliert
OnKeyDown	: Handler für gedrückte Tastaturtaste
OnKeyPress	: Handler für ein vom Benutzer eingegebenes Zeichen
OnKeyUp	: Handler für Tastaturtaste freigegeben
OnMouseDown	: Ereignishandler für das Drücken der Maustaste
OnMouseEnter	: Ereignishandler für das Betreten des Bereichs des Steuerelements mit der Maus
OnMouseLeave	: Ereignishandler für Maus, die den Bereich des Steuerelements verlässt
OnMouseMove	: Ereignishandler für die Mausbewegung innerhalb des Steuerelements
OnMouseUp	: Ereignishandler für das Loslassen der Maustaste
OnStartDrag	: Ereignishandler für den Start eines Ziehvorgangs

Beschreibung

Die MultiRadioGroup findest du im Reiter Multi:

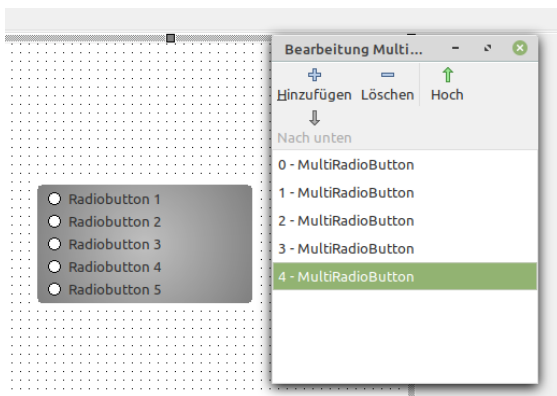


Wenn du die MultiRadioGroup auf die Form setzt besitzt sie nur einen RadioButton.



Dieser eine Radiobutton lässt sich auch nicht löschen!

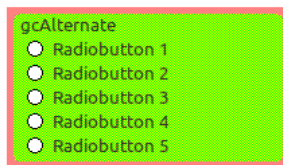
Um weiter Radiobuttons hinzuzufügen klickt man im Objektinspektor auf [RadioButtons](#). Jetzt öffnet sich ein Editor in dem weitere Radiobuttons angelegt werden können.



In der TreeView des Objektinspektors sieht man nun unter der MultiRadioGroup die einzelnen MultiRadioButton.



Es ist wichtig zu wissen das die MultiRadioGroup von einem Fokusrahmen umgeben ist. Wie man hier sieht besitzt die fokussierte MultiRadioGroup einen roten Rahmen. Das bedeutet im Umkehrschluss das die eigentliche RadioGroup um den Rahmen kleiner ist.

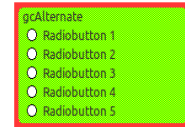


Mit der Eigenschaft [FocusColor](#) kann man die Farbe des Fokusrahmens einstellen. Mit [FocusAlphaBValue](#) kann die Transparenz des Fokusrahmens geregelt werden. Der Wert 0 bedeutet durchsichtig und 255 undurchsichtig. [FocusFrameWidth](#) bestimmt die Dicke des Rahmens.

Wert 50:

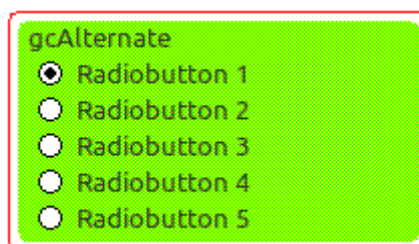


Wert 200:

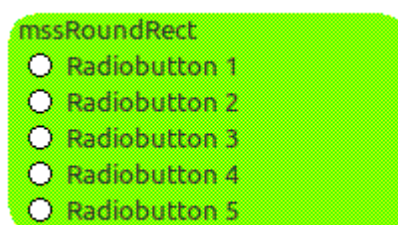
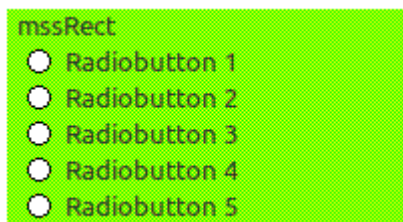


Setzt man [FocusFrameOn](#) auf false dann bleibt der Rand erhalten jedoch wird die Fokussierung nicht farblich angezeigt.

Mit [ForegroundFocusOn](#) besitzt die fokussierte MultiRadioGroup eine Border. Die Farbe der Border kann man mit [FocusColor](#) beeinflussen. Mit [FocusFrameWidth](#) kann der Abstand zur eigentlichen RadioGroup eingestellt werden. Es kann sinnvoll sein [FocusFrameOn](#) auf false zu setzen.

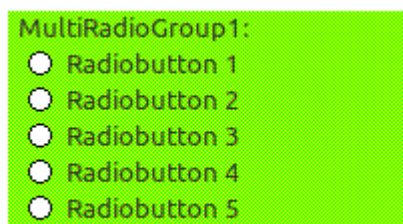


Mit der Eigenschaft [Style](#) stellt man die gewünschte geometrische Form der MultiRadioGroup ein.



Ist mssRoundRect eingestellt kann man mit der Eigenschaft [RndRctRadius](#) den Durchmesser der Eckenrundung einstellen.

Mit [Caption](#) kann man der MultiRadioGroup eine Überschrift hinzufügen. Siehe auch [57](#)



Der Hintergrund kann mit den Eigenschaften [ColorStart](#), [ColorEnd](#) und [ColorGradient](#) beeinflusst werden.

Hier ist ColorStart cILime und ColorEnd cIYellow:

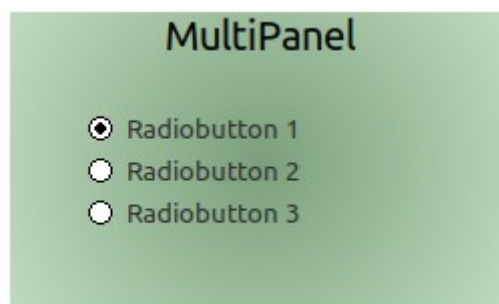


Alternate setzt abwechselnd ein Pixel auf Start bzw. EndColor.

Setzt man entweder ColorStart oder ColorEnd auf cINone wird der Hintergrund durchsichtig.

MultiRadioGroup1:

- ☐ Radiobutton 1
- ☐ Radiobutton 2
- ☐ Radiobutton 3
- ☐ Radiobutton 4
- ☐ Radiobutton 5



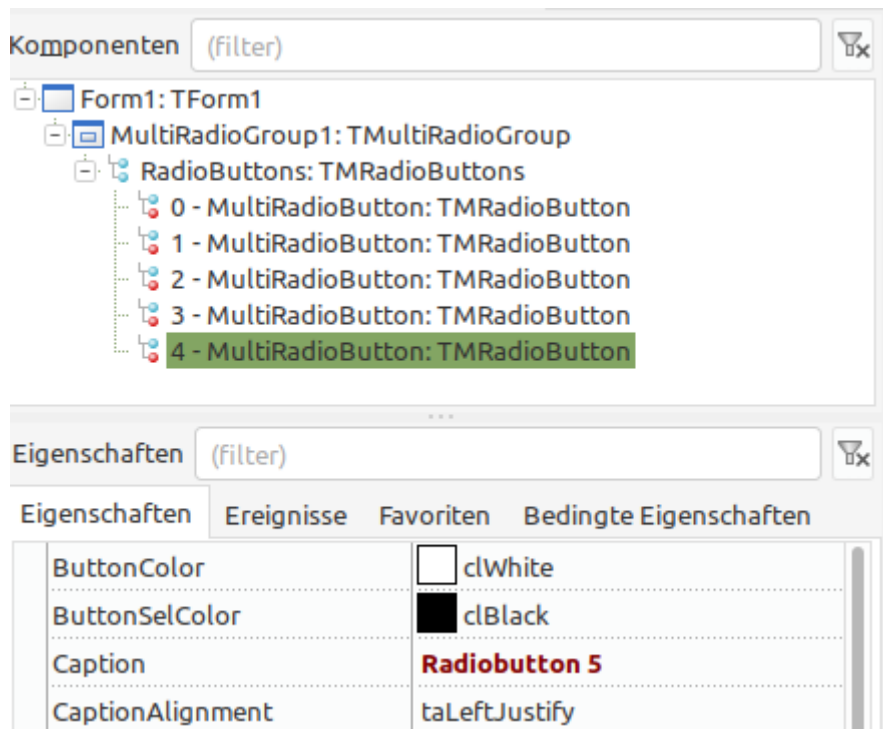
Sitz die MultiRadioGroup auf einem MultiPanel wird dieses sichtbar.

Gibt man der Eigenschaft [GroupIndex](#) einen anderen Wert als 0 lassen sich mehrere MultiRadioGroups die sich auf dem gleichen Parent befinden und den gleichen Index besitzen zusammen schalten.



Damit lassen sich beliebige Anordnungen verwirklichen. Navigieren kann man mit den Pfeiltasten, selektieren mit der Leertaste.

Die Einstellungen für die einzelnen MultiRadioButtons können separat für jeden Button vorgenommen werden. Einzige Ausnahme ist die Schriftgröße. Wird diese bei einem Button verändert so wirkt sich dies auf alle anderen Radiobuttons aus. Um einfach in die Ansicht der RadioButtons zu gelangen sollte der Komponentenbaum im Objektinspektor sichtbar sein. Ist dieser unsichtbar klickt man mit der rechten Maustaste in den OI und macht bei Komponentenbaum anzeigen einen Haken. Nun kann man die einzelnen MultiRadioButtons einfach anwählen.



Mit den `RadioButtons.Items[Index].Font` Einstellungen können (außer die Schriftgröße) für alle Button verschiedene Einstellungen getroffen werden.



Mit der Eigenschaft `RadioButtons.Item[Index].ButtonColor` wird der Hintergrund des RadioButtons gefüllt.



Hier Orange.

Mit `RadioButtons.Item[Index].Selected` wird bestimmt ob ein RadioButton angewählt ist. Mit `RadioButtons.Item[Index].ButtonSelColor` kann die Farbe der Selektierung bestimmt werden.



Hier Gelb.

Mit der Eigenschaft `RadioButtons.Item[Index].Caption` kann der Text hinter dem RadioButton gesetzt werden. Mit `RadioButtons.Item[Index].CaptionAlignment`, `RadioButtons.Item[Index].CaptionHorMargin` kann die Ausrichtung des Textes in der Horizontale beeinflusst werden.



Mit `RadioButtons.Item[Index].CaptionLayout` und `RadioButtons.Item[Index].CaptionVerMargin` kann der Text vertikal ausgerichtet werden.

Mit `RadioButtons.Item[Index].CaptionWordbreak` wird der Text bei Bedarf umgebrochen.



Allerdings muss mit `Rows` die Anzahl der benötigten Zeilen angegeben werden.

Mit `RadioButtons.Item[Index].Color` wird der Hintergrund des RadioButtons eingestellt, `clNone` macht durchsichtig.



Mit `RadioButtons.Item[Index].HoverColor` wird die Farbe des Hintergrundes des RadioButtons verändert wenn man mit der Maus darüber fährt, `clNone` macht durchsichtig.



Mit `RadioButtons.Item[Index].HoverStyle` kann gewählt werden ob das Hoverereignis vollausgefüllt oder nur als Rahmen dargestellt wird.



Mit Hilfe einer Imagelist lassen sich auch Bilder einfügen. Dazu bei `RadioButtons.Item[Index].Images` die ImageList auswählen. Bei `RadioButtons.Item[Index].ImageIndex` das gewünschte Bild auswählen und mit `RadioButtons.Item[Index].ImageLeft` und `RadioButtons.Item[Index].ImageTop` die Position des Bildes bestimmen.

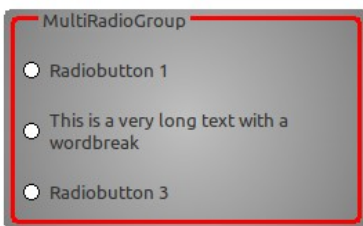


Möchte man der RadioGroup eine Border geben so ändert man [BorderColor](#) auf die gewünschte Farbe. Das bedeutet `clNone` macht die Border unsichtbar. Mit [BorderWidth](#) lässt sich die Dicke der Border einstellen.



Möchte man die Border nach innen verschieben kann man das mit [BorderMargin](#). Schiebt man dann eine Caption über die Border entsteht das Aussehen einer GroupBox.

Das verschieben der Caption geschieht mit [CaptionLeft](#) und [CaptionTop](#).



TMultiCheckGroup

Eigenschaften

Action	: Die dem Steuerelement zugeordnete Aktion
Align	: Gibt die Platzierung des Steuerelements innerhalb seines übergeordnetenElements an
Anchors	: Der Satz von Ankerdefinitionen für dieses Steuerelement
AutoSize	: Ermöglicht die automatische Anpassung der Größe der Kontrolle an ihren Inhalt
BidiMode	: Anpassung (von Textsteuerelementen) in bidirektionalen Leseumgebungen
BorderColor	: Die Farbe des Rahmens
BorderMargin	: Der Abstand des Rahmens zur Außenkante
BorderWidth	: Die Dicke des Rahmens
BorderSpacing	: Bestimmt den inneren und äußeren Rahmenabstand für dieses Steuerelement
Caption	: Die Überschrift der Checkgroup
CaptionLeft	: Die Koordinate des linken Randes der Beschriftung
CaptionTop	: Die Koordinate des oberen Randes der Beschriftung
Checkboxes	: Öffnet den Editor um Checkboxes hinzuzufügen
CheckBoxes.Item[Index].ButtonColor	: Die Farbe der Checkbox
CheckBoxes.Item[Index].ButtonSelBackColor	: Die Farbe der selektierten Checkbox
CheckBoxes.Item[Index].ButtonSelColor	: Die Farbe des Zeichens der selektierten Checkbox
CheckBoxes.Item[Index].Caption	: Der Text den der Benutzer in die Checkbox schreibt
CheckBoxes.Item[Index].CaptionAlignment	: Ausrichtung des Textes in der Caption (Links,Mitte,Rechts) taLeftJustify;
CheckBoxes.Item[Index].CaptionHorMargin	: Der horizontale Abstand des Textes im Textrechteck (nur wirksam mit taLeftJustify)
CheckBoxes.Item[Index].CaptionLayout	: Ausrichtung des Textes in der Caption (Oben,Mitte,Unten)
CheckBoxes.Item[Index].CaptionVerMargin	: Der vertikale Abstand des Textes im Textrechteck (nur wirksam mit tlTop)
CheckBoxes.Item[Index].CaptionWordbreak	: Ermöglicht einen Zeilenumbruch in der Caption
CheckBoxes.Item[Index].Color	: Die Hintergrundfarbe der Checkbox
Checkboxes.Items[Index].DisplayName	: Der Name der im TreeView des Objektinspektors angezeigt wird

<code>CheckBoxes.Item[Index].Enabled</code>	: Legt fest, ob das Steuerelement auf Maus- oder Tastatureingaben reagiert
<code>CheckBoxes.Items[Index].Font</code>	: Die Schrift die für die Textanzeige der Caption verwendet werden soll
<code>CheckBoxes.Item[Index].HoverColor</code>	: Die Farbe eines Hoverereignisses
<code>CheckBoxes.Item[Index].HoverStyle</code>	: Ob ein Hoverereignis nur als Rahmen oder vollflächig gezeichnet wird
<code>CheckBoxes.Item[Index].ImageIndex</code>	: Der Index eines Bildes in einer ImageList
<code>CheckBoxes.Item[Index].ImageLeft</code>	: Die Koordinate der linken Ecke des Bildes
<code>CheckBoxes.Item[Index].Images</code>	: Eine Liste zum Einfügen von Bildern
<code>CheckBoxes.Item[Index].ImageTop</code>	: Die Koordinate der oberen Ecke des Bildes
<code>CheckBoxes.Item[Index].ImageWidth</code>	: Die einmalige Breite aller Bilder in der Liste
<code>CheckBoxes.Items[Index].ParentFont</code>	: Verwendet die Schriftart aus dem Parent, wenn aktiviert
<code>CheckBoxes.Item[Index].Selected</code>	: Bestimmt ob eine Checkbox ausgewählt ist
<code>CheckBoxes.Item[Index].SelectedStyle</code>	: Bestimmt das Zeichen wenn selected
<code>CheckBoxes.Items[Index].Tag</code>	: Kann verwendet werden, um einen ganzzahligen Wert in der Komponente zu speichern
<code>ColorEnd</code>	: Die Endfarbe der CheckGroup (für Farbverlauf)
<code>ColorGradient</code>	: Die Richtung des Farbverlaufs
<code>ColorStart</code>	: Die Startfarbe der CheckGroup (für Farbverlauf)
<code>Constraints</code>	: Die minimale und maximale Breite und Höhe für das Steuerelement
<code>Cursor</code>	: Die geometrische Form des Mauszeigers wenn die Maus über diesem Steuerelement ist
<code>DragCursor</code>	: Die Cursorform, die angezeigt wird, während das Steuerelement gezogen wird
<code>DragKind</code>	: Der Vorgang, wenn das Steuerelement gezogen wird - Drag or Dock
<code>DragMode</code>	: Ermöglicht dem Benutzer, das Steuerelement zu ziehen
<code>Enabled</code>	: Legt fest, ob das Steuerelement auf Maus- oder Tastatureingaben reagiert
<code>FocusAlphaBValue</code>	: Wie transparent der Fokusrahmen ist (0=transparent, 255=undurchsichtig)
<code>FocusColor</code>	: Die Farbe des Fokusrahmens/Foregroundfocus wenn das Control den Fokus hat
<code>FocusFrameOn</code>	: Schaltet den Fokusrahmen ein und aus
<code>FocusFrameWidth</code>	: Die Dicke des Fokus-Rahmens
<code>Font</code>	: Die Schrift die für die Textanzeige der Caption verwendet werden soll
<code>ForegroundFocusOn</code>	: Zeigt an wenn der Slider den Fokus besitzt, schaltet ein aus

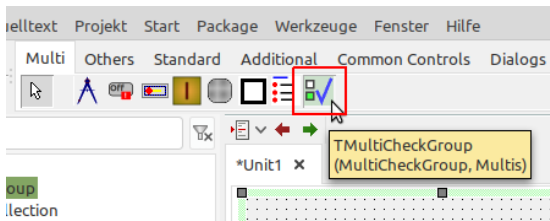
GroupIndex	: Der Index der Gruppe zu der die MultiCheckGroup gehört
Height	: Die Höhe des Controls
HelpContext	: Die ID für die kontextbezogene Hilfe zu diesem Steuerelement
HelpKeyword	: Das Schlüsselwort für die kontextbezogene Hilfe zu diesem Steuerelement
HelpType	: Legt fest, ob die kontextsensitive Hilfe nach numerischer ID oder Schlüsselwort ausgewählt wird
Hint	: Der Text, der im Hinweifenster für das Steuerelement angezeigt werden soll
Left	: Die Clientkoordinate des linken Rands des Steuerelements
RndRctRadius	: Eckendurchmesser wenn geometrische Form ist RoundRect
Rows	: Anzahl der Zeilen wenn Wordbreak aktive
Style	: Die geometrische Form der CheckGroup
TabOrder	: Bestimmt die Reihenfolge der Steuerelementnavigation, wenn der Benutzer die Tabulatortaste drückt
TabStop	: Ermöglicht dem Benutzer das Navigieren zu diesem Steuerelement durch Drücken der Tabulatortaste
Tag	: Kann verwendet werden, um einen ganzzahligen Wert in der Komponente zu speichern
Top	: Die Clientkoordinate des oberen Rands des Steuerelements
Visible	: Ermöglicht das Ein- oder Ausblenden des Steuerelements und aller seiner untergeordneten Elemente
Width	: Die Breite des Controls

Ereignisse

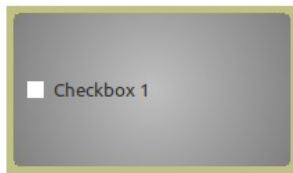
OnChange	: Liefert den Index der Checkbox
OnClick	: Benachrichtigungshandler für Mausklicks
OnDragDrop	: Dieser Handler bestimmt die Aktion beim Ablegen auf diesem Steuerelement in einem Drag-Drop-Vorgang
OnDragOver	: Ereignishandler für ein Steuerelement, das über dieses Steuerelement gezogen wird
OnEndDrag	: Benachrichtigungshandler für das Ende eines Ziehvorgangs
OnEnter	: Handler für die Steuerung, die den Fokus erhält
OnExit	: Handler für die Steuerung, die den Fokus verliert
OnKeyDown	: Handler für gedrückte Tastaturtaste
OnKeyPress	: Handler für ein vom Benutzer eingegebenes Zeichen
OnKeyUp	: Handler für Tastaturtaste freigegeben
OnMouseDown	: Ereignishandler für das Drücken der Maustaste
OnMouseEnter	: Ereignishandler für das Betreten des Bereichs des Steuerelements mit der Maus
OnMouseLeave	: Ereignishandler für Maus, die den Bereich des Steuerelements verlässt
OnMouseMove	: Ereignishandler für die Mausbewegung innerhalb des Steuerelements
OnMouseUp	: Ereignishandler für das Loslassen der Maustaste
OnStartDrag	: Ereignishandler für den Start eines Ziehvorgangs

Beschreibung

Die MultiCheckGroup findest du im Reiter Multi:

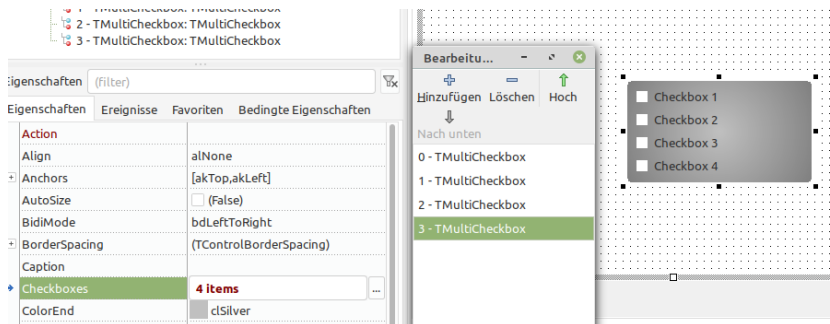


Wenn du die MultiCheckGroup auf die Form setzt besitzt sie nur eine CheckBox.

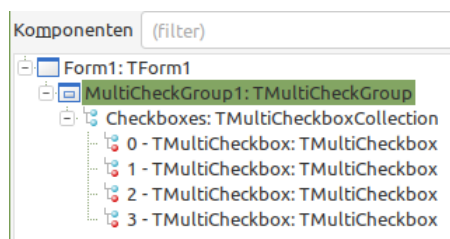


Diese eine CheckBox lässt sich auch nicht löschen!

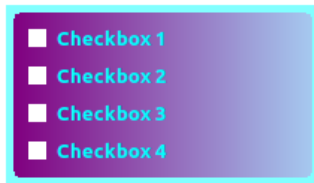
Um weitere CheckBoxen hinzuzufügen klickt man im Objektspektor auf [Checkboxes](#). Jetzt öffnet sich ein Editor in dem weitere CheckBoxen angelegt werden können.



In der TreeView des Objektspektors sieht man nun unter der MultiCheckGroup die einzelnen MultiCheckBoxen.

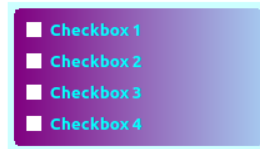


Es ist wichtig zu wissen das die MultiCheckGroup von einem Fokusrahmen umgeben ist. Wie man hier sieht besitzt die fokussierte MultiCheckGroup einen hellblauen Rahmen. Das bedeutet im Umkehrschluss das die eigentliche CheckGroup um den Rahmen kleiner ist.

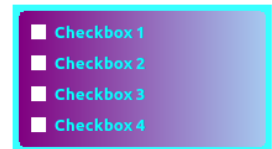


Mit der Eigenschaft `FocusColor` kann man die Farbe des Fokusrahmens einstellen. Mit `FocusAlphaBValue` kann die Transparenz des Fokusrahmens geregelt werden. Der Wert 0 bedeutet durchsichtig und 255 undurchsichtig. `FocusFrameWidth` bestimmt die Dicke des Rahmens.

Wert 50:

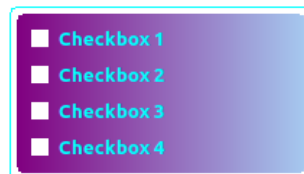


Wert 200:

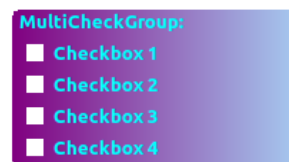


Setzt man `FocusFrameOn` auf false dann bleibt der Rand erhalten jedoch wird die Fokussierung nicht farblich angezeigt.

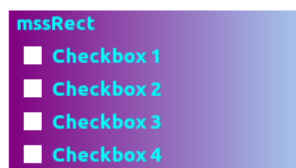
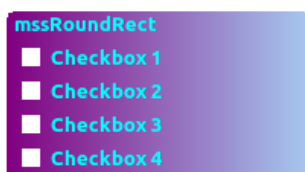
Mit `ForegroundFocusOn` besitzt die fokussierte MultiCheckGroup eine Border. Die Farbe der Border kann man mit `FocusColor` beeinflussen. Mit `FocusFrameWidth` kann der Abstand zur eigentlichen CheckGroup eingestellt werden. Es kann sinnvoll sein `FocusFrameOn` auf false zu setzen.



Mit `Caption` kann man der MultiCheckGroup eine Überschrift hinzufügen. Siehe auch 68

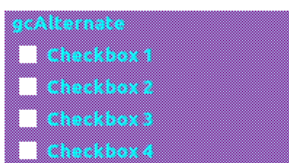
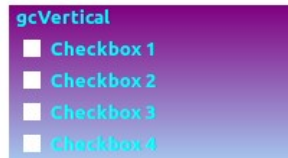
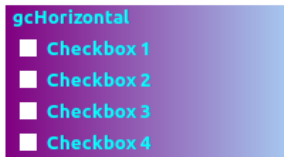


Mit der Eigenschaft `Style` stellt man die gewünschte geometrische Form der MultiCheckGroup ein.



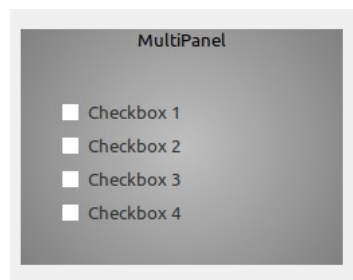
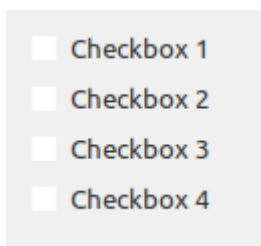
Ist `mssRoundRect` eingestellt kann man mit der Eigenschaft `RndRctRadius` den Durchmesser der Eckenrundung einstellen.

Der Hintergrund kann mit den Eigenschaften [ColorStart](#), [ColorEnd](#) und [ColorGradient](#) beeinflusst werden.
Hier ist ColorStart clPurple und ColorEnd clSkyBlue:



Alternate setzt abwechselnd ein Pixel auf Start bzw. EndColor.

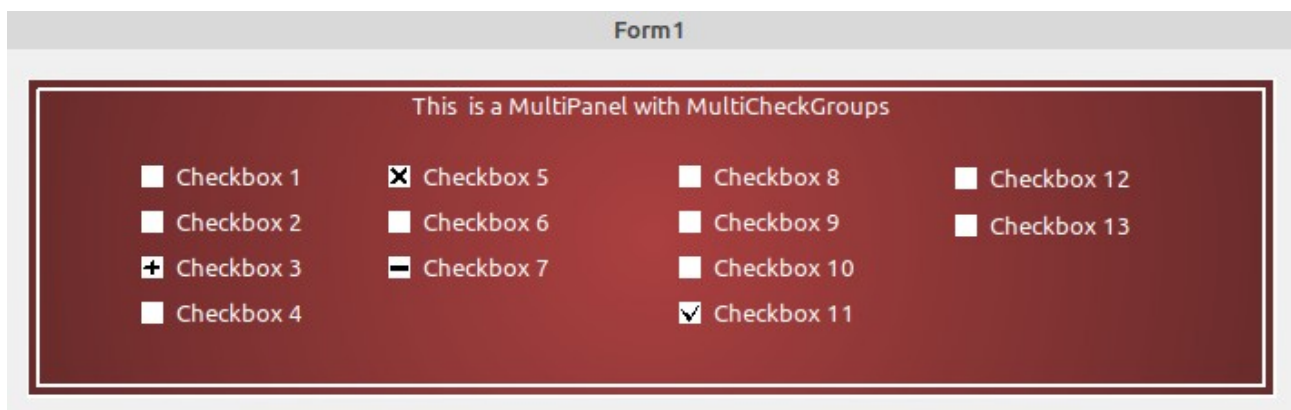
Setzt man entweder ColorStart oder ColorEnd auf clNone wird der Hintergrund durchsichtig.



Sitz die MultiRadioGroup auf einem MultiPanel wird dieses sichtbar.

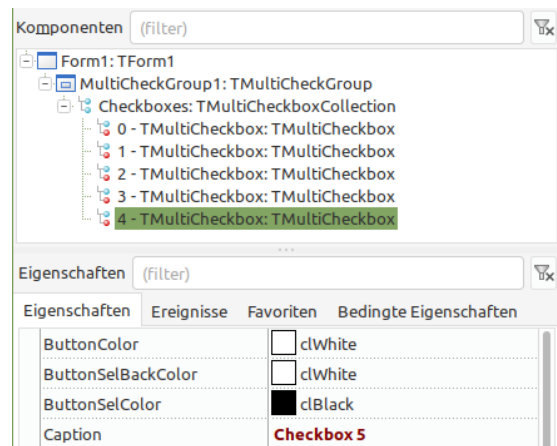
FocusFrameOn sollte besser auf false gesetzt werden.

Gibt man der Eigenschaft [GroupIndex](#) einen anderen Wert als 0 lassen sich mehrere MultiCheckGroups die sich auf dem gleichen Parent befinden und den gleichen Index besitzen zusammen schalten.



Damit lassen sich beliebige Anordnungen verwirklichen. Navigieren kann man mit den Pfeiltasten, selektieren mit der Leertaste.

Die Einstellungen für die einzelnen MultiCheckBoxes können separat für jede Checkbox vorgenommen werden. Einzige Ausnahme ist die Schriftgröße. Wird diese bei einer Checkbox verändert so wirkt sich dies auf alle anderen Checkboxes aus. Um einfach in die Ansicht der Checkboxes zu gelangen sollte der Komponentenbaum im Objektinspektor sichtbar sein. Ist dieser unsichtbar klickt man mit der rechten Maustaste in den OI und macht bei Komponentenbaum anzeigen einen Haken. Nun kann man die einzelnen Checkboxes einfach anwählen.



Mit den `Checkboxes.Items[Index].Font` Einstellungen können (außer die Schriftgröße) für alle Button verschiedene Einstellungen getroffen werden.



Mit der Eigenschaft `CheckBoxes.Item[Index].ButtonColor` wird der Hintergrund der CheckBox gefüllt.



Hier `clLime`.

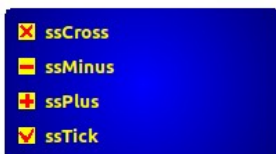
Mit `CheckBoxes.Item[Index].Selected` wird bestimmt ob eine CheckBox angewählt ist. Mit `CheckBoxes.Item[Index].ButtonSelColor` kann die Farbe der Selektierung bestimmt werden und mit `CheckBoxes.Item[Index].ButtonSelBackColor` der Hintergrund einer selektierten CheckBox.



`ButtonSelColor` ist hier `clRed`.

`ButtonSelBackColor` ist hier `clYellow`.

`CheckBoxes.Item[Index].SelectedStyle` bestimmt welches Zeichen in eine selektierte CheckBox gezeichnet wird.

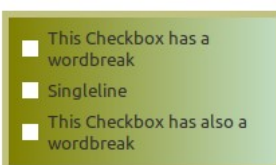


Mit der Eigenschaft `CheckBoxes.Item[Index].Caption` kann der Text hinter der CheckBox gesetzt werden. Mit `CheckBoxes.Item[Index].CaptionAlignment`, `CheckBoxes.Item[Index].CaptionHorMargin` kann die Ausrichtung des Textes in der Horizontale beeinflusst werden.

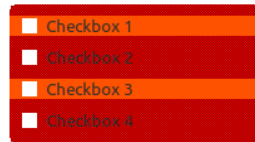


Mit `CheckBoxes.Item[Index].CaptionLayout` und `CheckBoxes.Item[Index].CaptionVerMargin` kann der Text vertikal ausgerichtet werden.

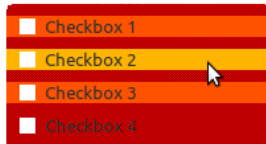
Mit `CheckBoxes.Item[Index].CaptionWordbreak` wird der Text bei Bedarf umgebrochen. Allerdings muss mit `Rows` die Anzahl der benötigten Zeilen angegeben werden.



Mit `CheckBoxes.Item[Index].Color` wird der Hintergrund der Checkbox incl. der Caption eingestellt, `clNone` macht durchsichtig.



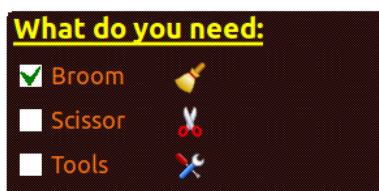
Mit `CheckBoxes.Item[Index].HoverColor` wird die Farbe des Hintergrundes des RadioButtons verändert wenn man mit der Maus darüber fährt, `clNone` macht durchsichtig.



Mit `CheckBoxes.Item[Index].HoverStyle` kann gewählt werden ob das Hoverereignis vollausgefüllt oder nur als Rahmen dargestellt wird.

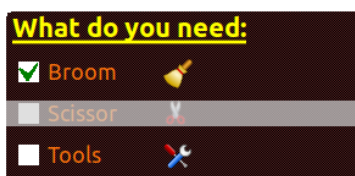


Mit Hilfe einer Imagelist lassen sich auch Bilder einfügen. Dazu bei `CheckBoxes.Item[Index].Images` die ImageList auswählen. Bei `CheckBoxes.Item[Index].ImageIndex` das gewünschte Bild auswählen und mit `CheckBoxes.Item[Index].ImageLeft` und `CheckBoxes.Item[Index].ImageTop` die Position des Bildes bestimmen.



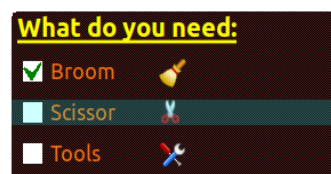
Mit `CheckBoxes.Item[Index].ImageWidth` kann die Größe des Bildes verändert werden. Es empfiehlt sich nur kleiner zu skalieren!

Setzt man `CheckBoxes.Item[Index].Enabled` auf false kann eine Auswahl deaktiviert werden.

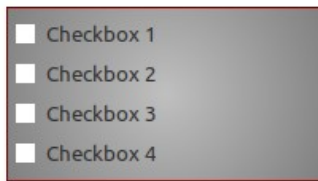


Zur Laufzeit kann die Farbe und die Transparenz des Disabledbalkens eingestellt werden.

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  MultiCheckGroup1.Checkboxes.Items[1].DisabledColor:= clAqua;
  MultiCheckGroup1.Checkboxes.Items[1].DisabledAlphaValue:= 50;
end;
```

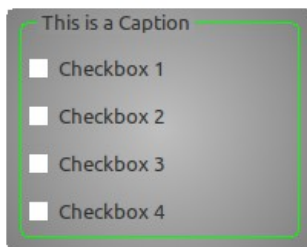


Möchte man der CheckGroup eine Border geben so ändert man [BorderColor](#) auf die gewünschte Farbe. Das bedeutet `clNone` macht die Border unsichtbar. Mit [BorderWidth](#) lässt sich die Dicke der Border einstellen.



Möchte man die Border nach innen verschieben kann man das mit [BorderMargin](#). Schiebt man dann eine Caption über die Border entsteht das Aussehen einer GroupBox.

Das verschieben der Caption geschieht mit [CaptionLeft](#) und [CaptionTop](#).



TMultiSwitch

Eigenschaften

Action	: Die dem Steuerelement zugeordnete Aktion
Align	: Gibt die Platzierung des Steuerelements innerhalb seines übergeordneten Elements an
Anchors	: Der Satz von Ankerdefinitionen für dieses Steuerelement
BestTextHeight	: Passt die Texthöhe automatisch der Größe des Controlls an
BidiMode	: Anpassung (von Textsteuerelementen) in bidirektionalen Leseumgebungen
BorderColor	: Farbe des Randes (clNone = keinRand)
BorderSpacing	: Bestimmt den inneren und äußeren Rahmenabstand für dieses Steuerelement
ButtonColor	: Die Farbe des Buttons, Rollimages sind nur sichtbar wenn ButtonColor = clNone
CaptionAlignment	: Ausrichtung des Textes in der Caption (Links,Mitte,Rechts)
CaptionHorMargin	: Der horizontale Abstand des Textes im Textrechteck (nur wirksam mit taLeftJustify)
CaptionLayout	: Ausrichtung des Textes in der Caption (Oben,Mitte,Unten)
CaptionVerMargin	: Der vertikale Abstand des Textes im Textrechteck (nur wirksam mit tlTop)
Constraints	: Die minimale und maximale Breite und Höhe für das Steuerelement
Cursor	: Die Form des Mauszeigers, wenn sich die Maus über diesem Steuerelement befindet
Direction	: Gibt an ob der Button beim Start rechts oder links ist
DisabledColor	: Die Farbe des Controlls wenn enable := false. Nur zur Laufzeit!
DragCursor	: Die Cursorform, die angezeigt wird, während das Steuerelement gezogen wird
DragKind	: Der Vorgang, wenn das Steuerelement gezogen wird - Drag or Dock
DragMode	: Ermöglicht dem Benutzer, das Steuerelement zu ziehen
Enable	: Legt fest, ob das Steuerelement auf Maus- oder Tastatureingaben reagiert.
EnabledBlendFaktor	: Wie transparent die DisabledColor ist (1=undurchsichtig, 0=durchsichtig). Nur zur Laufzeit!
FocusAlphaBValue	: Wie transparent der Fokusrahmen ist (0=transparent, 255=undurchsichtig).
FocusedBlendFaktor	: Wie transparent die Fokusfarbe ist (1=undurchsichtig, 0=durchsichtig). Nur zur Laufzeit!
FocusColor	: Die Farbe wenn das Control den Fokus hat (clNone = keine Fokus-Anzeige)
FocusFrameOn	: Schaltet den Fokusrahmen ein und aus
FocusFrameWidth	: Die Dicke des Fokus-Rahmens
Font	: Die Schrift die für die Textanzeige in diesem Schalter verwendet werden soll.
ForegroundFocusOn	: Zeigt an wenn der Button den Fokus besitzt
GroupIndex	: Der Index der Gruppe zu der der MultiSwitch gehört
Height	: Die vertikale Größe des Steuerelements
HelpContext	: Die ID für die kontextbezogene Hilfe zu diesem Steuerelement

HelpKeyword	: Das Schlüsselwort für die kontextbezogene Hilfe zu diesem Steuerelement
HelpType	: Legt fest, ob die kontextsensitive Hilfe nach numerischer ID oder Schlüsselwort ausgewählt wird
Hint	: Der Text, der im Hinweifenster für das Steuerelement angezeigt werden soll
HoverBlendFaktor	: Wie transparent die Hoverfarbe ist (1=undurchsichtig, 0=durchsichtig). Nur Zur Laufzeit!
HoverColor	: Die Farbe eines Hoverereignisses (clNone = kein Hover)
ImgSizeFactor	: Zum Ausgleich wenn mit LoadFromFile Images mit <>64px geladen werden. Nur zur Laufzeit.
Left	: Die Clientkoordinate des linken Rands des Steuerelements
LeftBgrdColor	: Die linke Hintergrundfarbe
LeftCaption	: Die Caption die angezeigt wird wenn der Button links ist
LeftImageIndex	: Der Index des linken geladenen Bildes
NewRollImage	: Startet den Eigenschaftseditor um geladene Images auszuwählen
PopupMenu	: Ein Menü das angezeigt wird, wenn Sie mit der rechten Maustaste auf dieses Steuerelement klicken
RightBgrdColor	: Die rechte Hintergrundfarbe
RightCaption	: Die Caption die angezeigt wird wenn der Button rechts ist
RightImageIndex	: Der Index des rechten geladenen Bildes
Roll	: Bestimmt ob sich der RollButton (Image) dreht
Rotation	: Die Schritte beim Drehen
Speed	: Die Geschwindigkeit mit der sich der Button bewegt
SwitchMode	: Der Modus mit dem der Schalter betätigt wird, klicken oder schieben
TabOrder	: Bestimmt die Reihenfolge der Steuerelementnavigation, wenn der Benutzer die Tabulatortaste drückt
TabStop	: Ermöglicht dem Benutzer das Navigieren zu diesem Steuerelement durch Drücken der Tabulatortaste
Top	: Die Clientkoordinate des oberen Rands des Steuerelements
Visible	: Ermöglicht das Ein- oder Ausblenden des Steuerelements und aller seiner untergeordneten Elemente
Width	: Die Breite des Controls.

Öffentliche Prozeduren

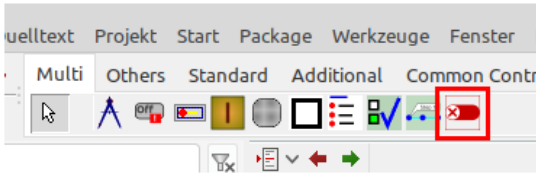
procedure **LoadImagesFromFile**(LeftFilename,RightFilename: string);

Ereignisse

OnChange	: Liefert den Index der Checkbox
OnClick	: Benachrichtigungshandler für Mausklicks
OnDirection	: Liefert die Richtung des Schalters, aLeft und aRight geben true oder false zurück.
OnDragDrop	: Dieser Handler bestimmt die Aktion beim Ablegen auf diesem Steuerelement in einem Drag-Drop-Vorgang
OnDragOver	: Ereignishandler für ein Steuerelement, das über dieses Steuerelement gezogen wird
OnEndDrag	: Benachrichtigungshandler für das Ende eines Ziehvorgangs
OnEnter	: Handler für die Steuerung, die den Fokus erhält
OnExit	: Handler für die Steuerung, die den Fokus verliert
OnKeyDown	: Handler für gedrückte Tastaturtaste
OnKeyPress	: Handler für ein vom Benutzer eingegebenes Zeichen
OnKeyUp	: Handler für Tastaturtaste freigegeben
OnLeft	: Liefert einen true Wert wenn der Button auf der linken Seite ankommt.
OnMouseDown	: Ereignishandler für das Drücken der Maustaste
OnMouseEnter	: Ereignishandler für das Betreten des Bereichs des Steuerelements mit der Maus
OnMouseLeave	: Ereignishandler für Maus, die den Bereich des Steuerelements verlässt
OnMouseMove	: Ereignishandler für die Mausbewegung innerhalb des Steuerelements
OnMouseUp	: Ereignishandler für das Loslassen der Maustaste
OnRight	: Liefert einen true Wert wenn der Button auf der rechten Seite ankommt.
OnStartDrag	: Ereignishandler für den Start eines Ziehvorgangs

Beschreibung

Den MultiSwitch findest du im Reiter Multi:



Die Idee zu diesem Schalter ist in einem Beitrag des Deutschen Lazarus Forums entstanden:

<https://www.lazarusforum.de/viewtopic.php?p=137567#p137567>

Wenn du den MultiSwitch auf die Form setzt sieht er so aus:



Er besitzt eine Breite von 60px und eine Höhe von 26px. Dieses Seitenverhältnis wird beim Skalieren immer beibehalten. Die max. Größe beträgt 175x76px. Alle geladenen Images im Rollbutton (hier das graue Kreuz) besitzen eine Größe von 64px. Die von mir in die Resource geladenen Images stammen von Roland Hahn (aka „Ally“). Vielen Dank an Roland!

<https://www.lazarusforum.de/viewtopic.php?f=1&t=14263>

Es ist wichtig zu wissen das der MultiSwitch von einem Fokusrahmen umgeben ist. Wie man hier sieht besitzt der fokussierte MultiSwitch einen orangenen Rahmen. Das bedeutet im Umkehrschluss das der eigentliche Switch um den Rahmen kleiner ist.



Mit der Eigenschaft `FocusColor` kann man die Farbe des Fokusrahmens einstellen. Mit `FocusAlphaBValue` kann die Transparenz des Fokusrahmens geregelt werden. Der Wert 0 bedeutet durchsichtig und 255 undurchsichtig. `FocusFrameWidth` bestimmt die Dicke des Rahmens.

Wert 50:



Wert 200:



Setzt man `FocusFrameOn` auf false dann bleibt der Rand erhalten jedoch wird die Fokussierung nicht farblich angezeigt.

Mit `ForegroundFocusOn` verändert der fokussierte MultiSwitch seine Farbe. Beeinflusst wird dies mit `FocusColor` und `FocusedBlendFaktor` (zwischen 0=durchsichtig und 1=undurchsichtig).

Nicht fokussiert:



Fokussiert 0.2 Blendfaktor:



Fokussiert 0.8 Blendfaktor:



Mit der Eigenschaft [Direction](#) kann man einstellen ob der Rollbutton zu Beginn links oder rechts ist.

msLeft:



msRight:



Mit [SwitchMode](#) wechselt man zwischen Klicken und Schieben um den Schalter zu schalten.

Wählt man msClick genügt ein Klick in den Schalter um den Zustand zu wechseln. Ist msSlide eingestellt muss man den Rollbutton mit der Maus auf die andere Seite ziehen. Man erkennt an einem Hand-Cursor das msSlide aktiv ist.



Besitzt der Switch den Fokus kann er mit der Entertaste geschaltet werden!

Beim Wechsel des Rollbuttons von einer zur anderen Seite beginnt sich der Button um sich selbst zudrehen. Möchte man dies nicht kann man die Eigenschaft [Roll](#) auf false stellen.



Mit [Speed](#) kann man die Drehgeschwindigkeit in ms einstellen. Mit [Rotation](#) den Winkelgrad um den sich der Rollbutton je Schritt weiter dreht.

Mit [LeftCaption](#) wird der Text gesetzt der angezeigt wird wenn der Rollbutton links ist:



Mit [RightCaption](#) wird der Text gesetzt der angezeigt wird wenn der Rollbutton rechts ist:



Ist die Eigenschaft [BestTextHeight](#) true wird versucht den Text möglichst hoch darzustellen. Wird der Text damit etwas zu lange kann man BestTextHeight auf false stellen und die Schriftgröße unter [Font](#) einstellen. Dort lässt sich ebenfalls die Farbe der Schrift verändern.



Mit den Eigenschaften [LeftBgndColor](#) und [RightBgndColor](#) lässt sich die Hintergrundfarbe des Controls verändern. Left und Right bezieht sich dabei immer auf die Lage des Rollbuttons.



Ändert man [BorderColor](#) auf $\langle \rangle$ clNone wird eine Border in der gewählten Farbe gezeichnet.



Möchte man kein Rollimage verwenden sondern einen einfarbigen Button haben muss man [ButtonColor](#) auf $\langle \rangle$ clNone stellen. Der Button wird dann in der gewählten Farbe gezeichnet.



Um einen Hover-Effekt zu bekommen stellt man [HoverColor](#) auf $\langle \rangle$ clNone. Die Transparenz der Hoverfarbe kann man mit [HoverBlendFaktor](#) verändern. Wobei 0 durchsichtig und 1 undurchsichtig entspricht.

Ohne Hover



Blendfaktor 0.2



Blendfaktor 0.5



Stellt man [Enable](#) auf false, so das der Switch nicht mehr auf Eingaben reagiert, wird dies durch eine veränderte Farbe kenntlich gemacht. Die Farbe kann man mit [DisabledColor](#) und die Transparenz mit [EnabledBlendFaktor](#) einstellen. Beide Properties stehen nur zur Laufzeit zur Verfügung!

Enabled:

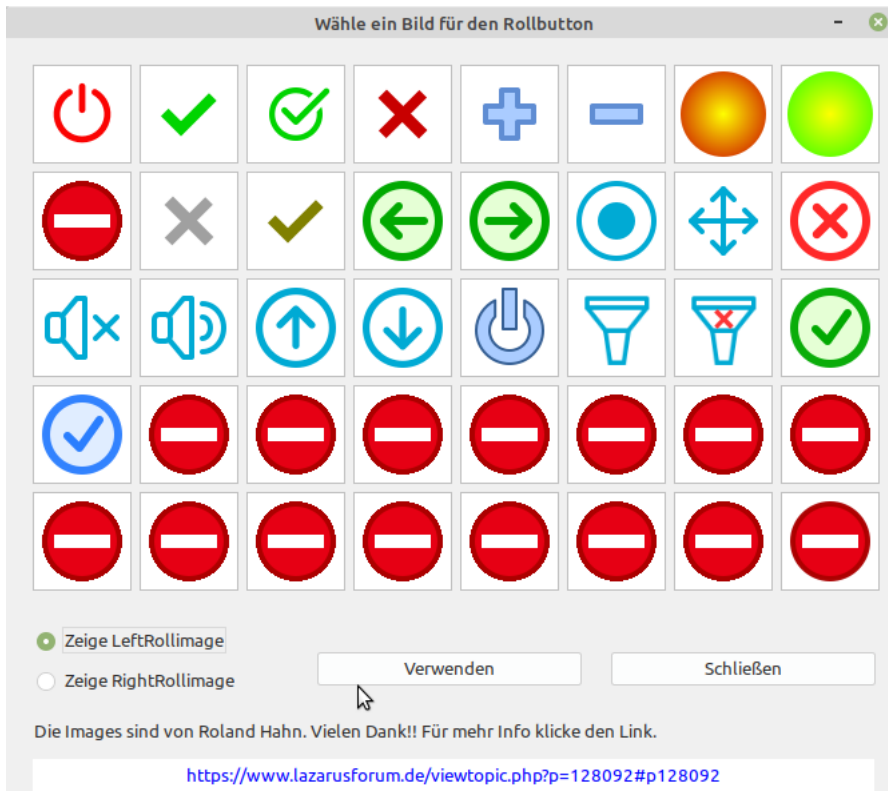


Disabled:



Rollimages kann man mittels Eigenschaftseditor laden. Dazu im Objektinspector auf [NewRollImage](#) klicken.

Es öffnet sich der Editor:



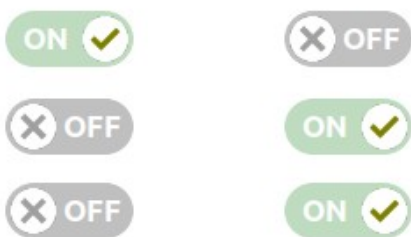
Als erstes wählt man unten links für welche Seite man ein Image wählen möchte. Mit verwenden wird das Image in den MultiSwitch übernommen und gleich angezeigt.

Wie zu sehen sind noch einige Plätze frei um eigene Images nach zu laden. Die Resource Datei befindet sich im Multis Verzeichnis.

Natürlich kann man auch im OI bei [LeftImageIndex](#) bzw. [RightImageIndex](#) den betreffenden Index des gewünschten Images einfach eingeben (sofern man ihn kennt).

Mit [LoadImagesFromFile](#)(LeftFilename,RightFilename: string) lassen sich Images aus einem Verzeichnis laden. Wichtig ist das beide Images die gleiche Größe besitzen müssen! Bevorzugt sollte man Images mit 64px verwenden. Um die Größe der geladenen Images etwas anpassen zu können gibt es die Eigenschaft [ImgSizeFactor](#) (nur zur Laufzeit).

Will man eine Gruppe von MultiSwitches von denen immer nur einer geschalten sein soll so muss man den [GroupIndex](#) auf $\neq 0$ stellen.



Der MultiSwitch der initial alleine steht bestimmt die Schaltrichtung.

Neben den üblichen Ereignissen gibt es noch drei spezielle MultiSwitch Events.

Das Ereignis [OnLeft](#) gibt einen Truwert zurück wenn der Rollbutton auf der linken Seite des Controlls ankommt.

Das Ereignis `OnRight` gibt einen Truewert zurück wenn der Rollbutton auf der rechten Seite des Controls ankommt.

Das Ereignis `OnDirection` liefert die Direction (also `msRight` oder `msLeft`) und je einen Booleanwert für `aLeft` und `aRight`.