

Das Package Multis

Installation

TMultiPanel

TMultiButton

TMultiButtonStyleManager

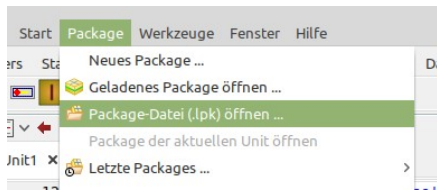
TMultiplexSlider

TMultiSeperator

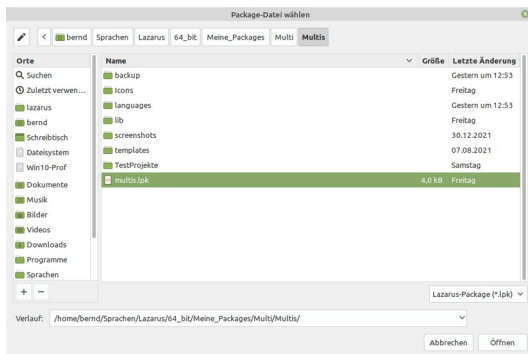
Installation

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

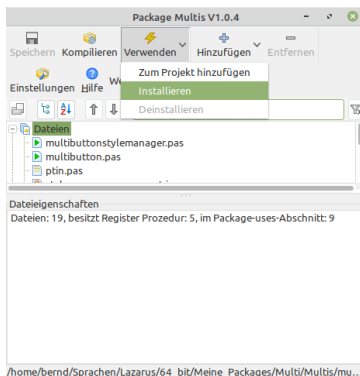
Nachdem das Package geklont oder heruntergeladen wurde kann es in Lazarus installiert werden. Dazu Lazarus öffnen, dort unter Package den Punkt Package-Datei (.lpk) öffnen... anklicken.



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

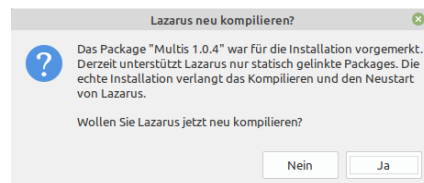


Es öffnet sich folgendes Fenster:

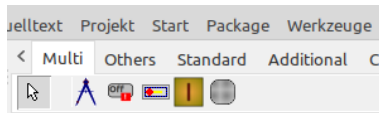


Hier auf Verwenden und dann Installieren.

Diesen Dialog mit Ja bestätigen:



Nun 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
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
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
DropDownMenu.Direction	: Die Ausklapprichtung
DropDownMenu.Hotspot	: Legt den Bereich fest in dem ein Klick wirkt, nur aktive mit DropDownMenu.Active und trPinned (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 Hinweisenster für das Steuerelement angezeigt werden soll
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
<code>Left</code>	: Die Clientkoordinate des linken Rands des Steuerelements
<code>RndRctRadius</code>	: Eckendurchmesser wenn geometrische Form ist RoundRect
<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
<code>Appear</code>	: läßt das Panel erscheinen (nur zur Laufzeit!)
<code>Disappear</code>	: läßt das Panel verschwinden (nur zur Laufzeit!)
<code>AnimationSpeed</code>	: Geschwindigkeit für Appear bzw. Disappear (default 0,05) (nur zur Laufzeit!)
<code>ParentAsBkgrd</code>	: Hintergrund des Panels nimmt die Farbe des Parent an (nur zur Laufzeit!)

Ereignisse

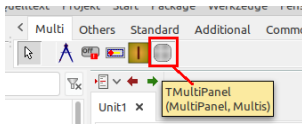
OnChangeBounds	: Handler für eine Änderung der Grenzen des Steuerelements
OnClick	: Benachrichtigungshandler für Mausklicks
OnCompressed	: Handler wenn das Panel komprimiert ist, nur aktive wenn
DropDownMenu.Active	
OnDragDrop	: Dieser Handler bestimmt die Aktion beim Ablegen auf diesem Steuerelement 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

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

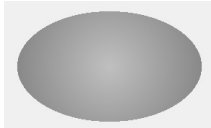


Die Form des MultiPanels lässt sich mit der Eigenschaft [Style](#) beeinflussen.

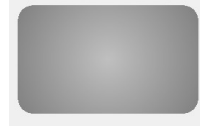
mpsRect:



mpsEllipse:



mpsRoundRect:

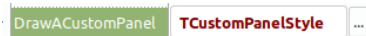


Der Eckenradius lässt sich mit [RndRctRadius](#) einstellen.
Standardeinstellung ist 40

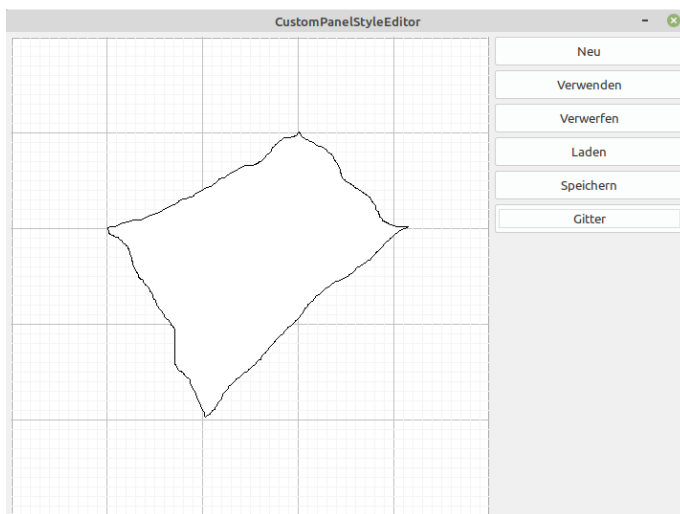
mpsCustom:



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:



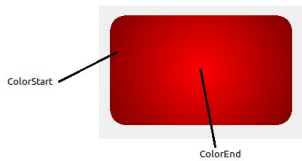
Klickt man jetzt auf Neu kann man einfach mit der Maus die Form des MultiPanels zeichnen. Klickt man auf Verwenden wird die MultiPanel-Form übernommen. Mit Verwerfen wird die MultiPanel-Form nicht übernommen und der Editor geschlossen. Mit Speichern kann man eine gezeichnete Form abspeichern und mit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

Bedenke es muss mpsCustom eingestellt sein!

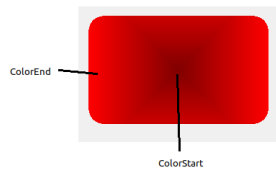
Zur Laufzeit lassen sich auch mit [LoadFromFile](#) vorab gespeicherte MultiPanel's laden.

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:

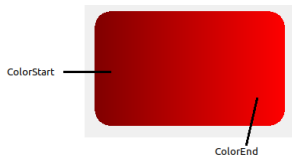


gcAlternate:

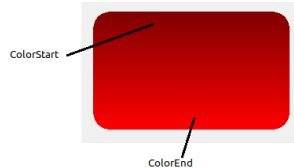


Setzt abwechselnd ein Pixel je auf Start- und EndColor.

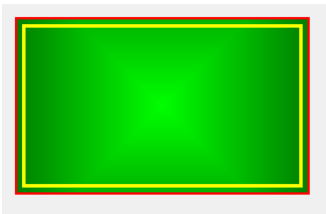
gcHorizontal:



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

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.

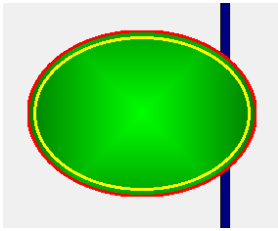
Standartwert ist 0,05. Je kleiner der Wert desto langsamer geht die Animation vonstatten. Bei einem Wert von 0,001 geht es schon sehr langsam.

Beispiel-Code:

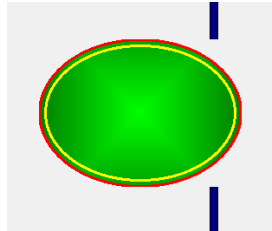
```
MultiPanel1.AnimationSpeed:= 0.001;
```


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`

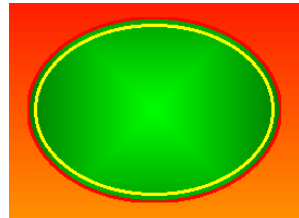


Diese Einstellung macht vor allem Sinn wenn der Parent seine Größe ändert. Denn dann wird zum Beispiel hier die gezeichnete Linie nicht richtig skaliert.

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 das MultiPanel geklickt werden, das Einklappen erfolgt jedoch nur wenn in einem festlegbaren Hotspot (**DropDownMenu.Hotspot**) geklickt wird.

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	: Zeigt an wenn der Button den Fokus besitzt
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>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

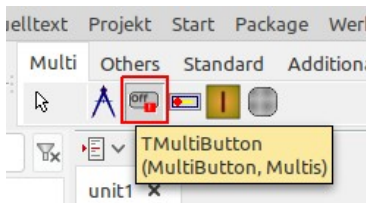
procedure `SetStyleManager`(AValue: TMultiButtonStyleManager);

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:



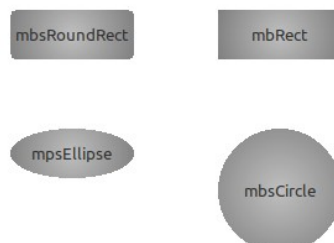
Setzt man [FocusFrameOn](#) auf false dann bleibt der Rand erhalten jedoch wird die Fokussierung nicht farblich angezeigt.

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 [RndRctRadius](#) 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

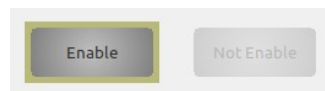
Wird der MultiButton gedrückt beeinflussen die Eigenschaften [PressedStartColor](#), [PressedEndColor](#), [PressedFontColor](#) und [PressedImageIndex](#) das Aussehen. Möchte man keine Änderung wenn der Button gedrückt wird bleibt nur die gleichen Einstellungen wie ColorStart etc. einzustellen.

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.

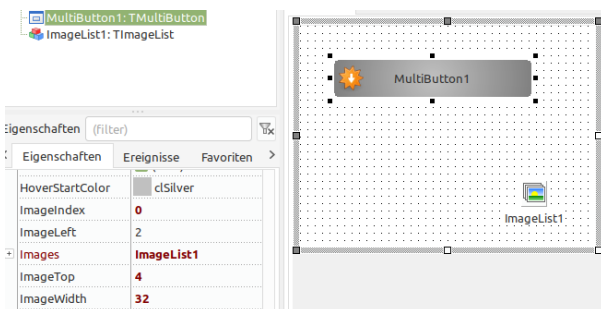


Verwende wenn nötig dazu nicht clDefault sondern `MultiButton2.DisabledColor:=GetColorResolvingParent;`

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:

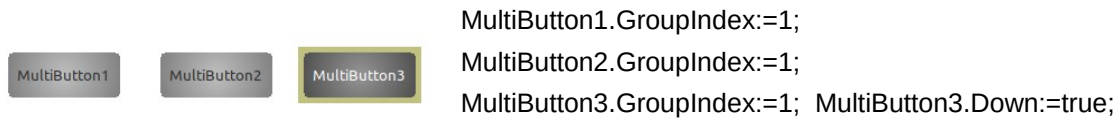
Verwendet man unter Windows HighDPI werden die Images und der MultiButton skaliert. Damit es zur Laufzeit funktioniert musste ich in den Projekteinstellungen noch bei DPI Anpassung Vista-8;an,8.1+;pro Monitor(True/PM) auswählen.

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!



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:



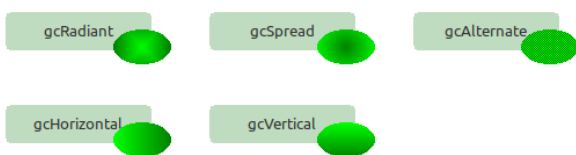
Um dies zu umgehen gibt es die Eigenschaft [MessageButton.CalculateAlthoughInvisible](#). Stellt man bei Nr. 2 diese Eigenschaft auf true sieht es so aus:



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.



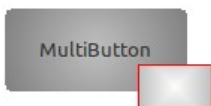
Um die Farbe des MessageButtons zu verändern benötigst du die Eigenschaften [MessageButton.ColorStart](#), [MessageButton.ColorEnd](#) und [MessageButton.ColorGradient](#). Um einen einfarbigen MessageButton zu bekommen muss ColorStart und ColorEnd gleich sein. Ansonsten bestimmt die Zusammensetzung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



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

Hier ist StartColor clLime und EndColor clGreen.

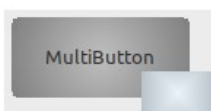
Standard mäßig steht [MessageButton.HoverOn](#) auf true. Das bedeutet das bei einem Hoverereignis (die Maus bewegt sich über dem MessageButton) um den MessageButton ein Rand gezeichnet wird. Die Farbe des Randes kann man mit [MessageButton.HoverColor](#) einstellen. Möchte man dies nicht setzt man HoverOn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



[MessageButton.BorderWidth](#) beeinflusst die Dicke des HoverRandes!

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

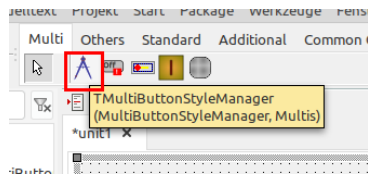
[MessageButton.PressedColor](#) eingestellte Farbe mit dem bei [MessageButton.PressedColBlendVal](#) hinterlegtem Wert über den MessageButton geblendet. Wobei 0 transparent bedeutet und 255 undurchsichtig.



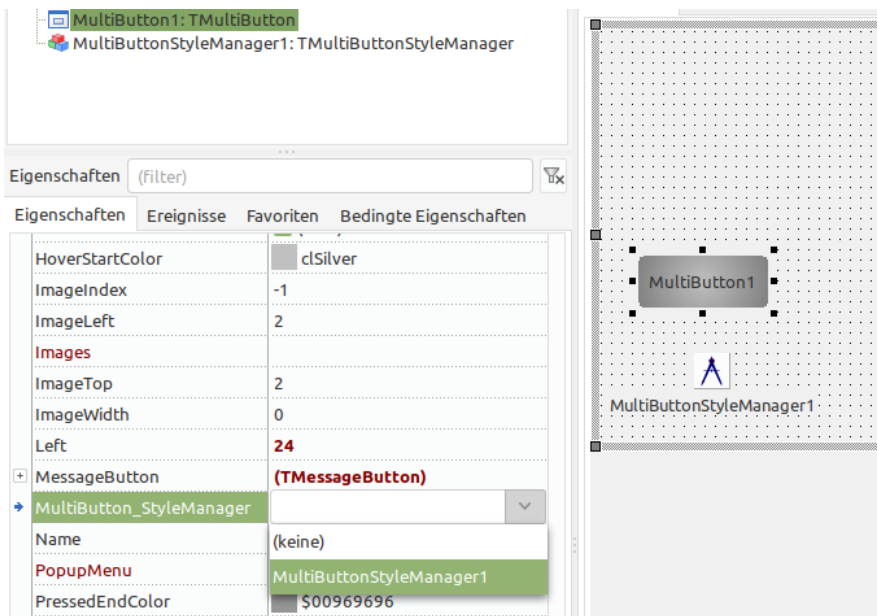
TMultiButtonStyleManager

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!

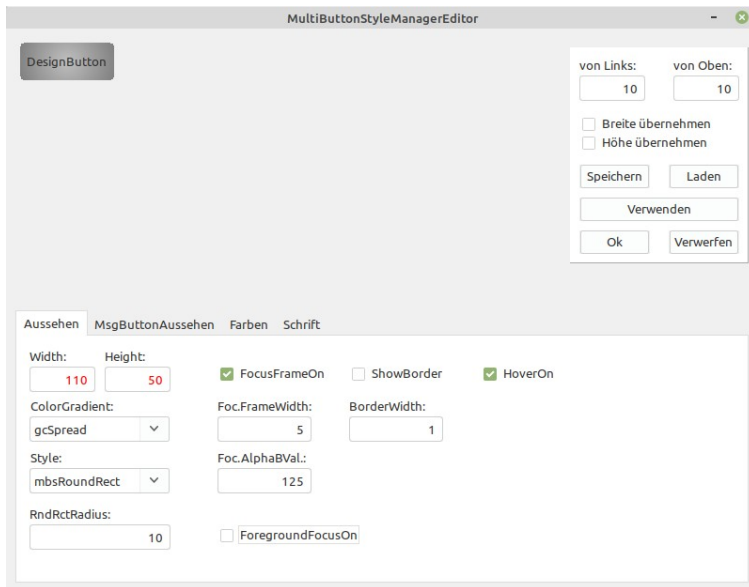
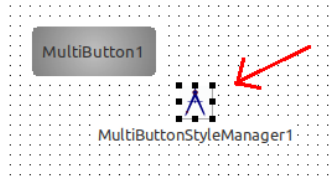
Achtung! Versucht man eine Eigenschaft eines verbundenen MultiButtons im OI Reiter des Buttons (nicht des Managers) zu ändern so schlägt das fehl!

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.

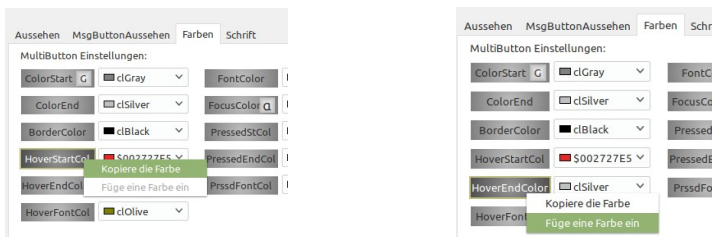


Überfährt man die verschiedenen Einstellmöglichkeiten mit der Maus werden deren Funktionen als Hinweise dargestellt.

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	: Zeigt an wenn der Slider den Fokus besitzt
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
Knob1Settings	: Die Eigenschaften der Schiebeknöpfe
Knob1Settings.ColorGradient	: Die Richtung des Farbverlaufs

Knob1Settings.Design	: Das Aussehen des Knopfes
Knob1Settings.DesignColor	: Die Farbe der Border bzw. Zeichen im Knopf
Knob1Settings HoverEndColor	: Die Endfarbe eines Hoverereignisses
Knob1Settings HoverOn	: Ermöglicht das Ein- oder Ausblenden eines Hoverereignisses
Knob1Settings HoverStartColor	: Die Startfarbe eines Hoverereignisses
Knob1Settings.KnobColorEnd	: Die Endfarbe des Knopfes (für Farbverlauf)
Knob1Settings.KnobColorStart	: Die Startfarbe des Knopfes (für Farbverlauf)
Knob1Settings.KnobPosition	: Die Position des Knopfes im Slider
Knob1Settings.KnobStyle	: Die Form des Knopfes
Knob1Settings.Visible	: zeigt den Knopf
Knob2Settings	: Die Eigenschaften der Schiebeknöpfe
Knob2Settings.ColorGradient	: Die Richtung des Farbverlaufs
Knob2Settings.Design	: Das Aussehen des Knopfes
Knob2Settings.DesignColor	: Die Farbe der Border bzw. Zeichen im Knopf
Knob2Settings HoverEndColor	: Die Endfarbe eines Hoverereignisses
Knob2Settings HoverOn	: Ermöglicht das Ein- oder Ausblenden eines Hoverereignisses
Knob2Settings HoverStartColor	: Die Startfarbe eines Hoverereignisses
Knob2Settings.KnobColorEnd	: Die Endfarbe des Knopfes (für Farbverlauf)
Knob2Settings.KnobColorStart	: Die Startfarbe des Knopfes (für Farbverlauf)
Knob2Settings.KnobPosition	: Die Position des Knopfes im Slider
Knob2Settings.KnobStyle	: Die Form des Knopfes
Knob2Settings.Visible	: zeigt den Knopf
Knob3Settings	: Die Eigenschaften der Schiebeknöpfe
Knob3Settings.ColorGradient	: Die Richtung des Farbverlaufs
Knob3Settings.Design	: Das Aussehen des Knopfes
Knob3Settings.DesignColor	: Die Farbe der Border bzw. Zeichen im Knopf
Knob3Settings HoverEndColor	: Die Endfarbe eines Hoverereignisses
Knob3Settings HoverOn	: Ermöglicht das Ein- oder Ausblenden eines Hoverereignisses
Knob3Settings HoverStartColor	: Die Startfarbe eines Hoverereignisses
Knob3Settings.KnobColorEnd	: Die Endfarbe des Knopfes (für Farbverlauf)
Knob3Settings.KnobColorStart	: Die Startfarbe des Knopfes (für Farbverlauf)
Knob3Settings.KnobPosition	: Die Position des Knopfes im Slider
Knob3Settings.KnobStyle	: Die Form des Knopfes
Knob3Settings.Visible	: zeigt den Knopf
Left	: Die Clientkoordinate des linken Rands des Steuerelements
Max	: Der höchste Wert im Wertebereich
Min	: 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
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)
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)
TextSettings.Font	: Die Schrift die für die Textanzeige verwendet werden soll
TextSettings.Height	: Die Höhe des TextLabels (nur wirksam mit poTop/poBottom)
TextSettings.Position	: Die Position des Textlabels (poNone für unsichtbar)
TextSettings.PostCaption	: Die Text hinter dem Wert im Textlabel
TextSettings.PreCaption	: Die Text vor dem Wert im Textlabel
TextSettings.Style	: Die geometrische Form des Textlabels
TextSettings.Width	: Die Breite des TextLabels (nur wirksam mit poLeft/poRight)
Top	: Die Clientkoordinate des oberen Rands des Steuerelements
TrackSettings	: Die Eigenschaften des Schiebeschlitzes
TrackSettings.ExtraColor	: Die Farbe der zusätzlichen Farbe (clNone für unsichtbar)
TrackSettings.ExtraRangeMax	: Maximalwert der zusätzlichen Farbe
TrackSettings.ExtraRangeMin	: Minimalwert der zusätzlichen Farbe
TrackSettings.SelRangeColor	: Die Farbe des selektierten Bereichs (clNone für unsichtbar)
TrackSettings.TrackColor	: Die Farbe des Schlitzes
ValueDisplaySettings	: Die Eigenschaften der Wertanzeige
ValueDisplaySettings.BorderColor	: Die Farbe des Rahmens (clNone für unsichtbar)
ValueDisplaySettings.BorderWidth	: Die Dicke des Rahmens
ValueDisplaySettings.ColorEnd	: Die Endfarbe der Anzeige (für Farbverlauf)
ValueDisplaySettings.ColorGradient	: Die Richtung des Farbverlaufs
ValueDisplaySettings.ColorStart	: Die Startfarbe der Anzeige (für Farbverlauf)
ValueDisplaySettings.Font	: Die Schrift die für der Anzeige verwendet werden soll
ValueDisplaySettings.InPercent	: Zeigt den Wert in Prozent
ValueDisplaySettings.Position	: Die Position des Displays im Slider, vdsNone für Unsichtbar
ValueDisplaySettings.Style	: Die geometrische Form der Anzeige, vdsNone macht keine Form
ValueDisplaySettings.X	: Beeinflußt die Position, nur mit vdpXY,vdpAboveRight,vdpBelowLeft
ValueDisplaySettings.Y	: Beeinflußt die Position, nur mit vdpXY,vdpAboveRight,vdpBelowLeft
Visible	: Ermöglicht das Ein- oder Ausblenden des Steuerelements und aller seiner untergeordneten Elemente
Width	: 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)
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
OnMouseWheelDown	: Ereignisbehandler für die Abwärtsbewegung des Mausekkrads
OnMouseWheelUp	: Ereignisbehandler für die Aufwärtsbewegung des Mausekkrads
OnStartDrag	: Ereignishandler für den Start eines Ziehvorgangs