Das Package Multis

Installation

TMultiPanel

TMultiButton

TMultiButtonStylemanager

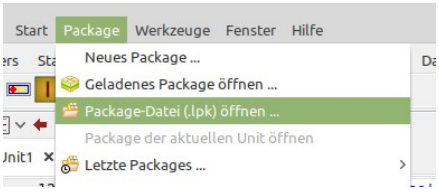
TMultiplexSlider

TMultiSeperator

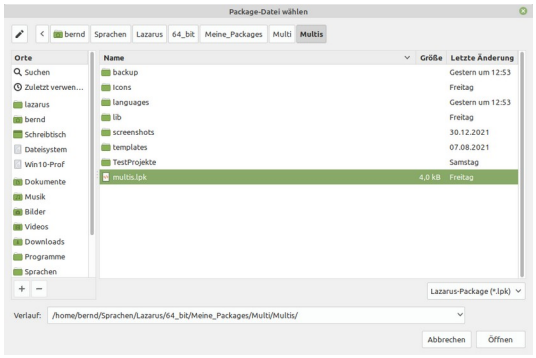
Installation

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

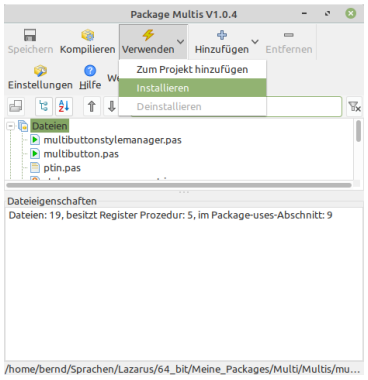
Nachdem das Package geklont oder heruntergeladen wurde kann es in Lazarus installiert werden. Dazu Lazarus öffnen, dort unter Package den Punkt Package-Datei (.lpk) öffnen… anklicken.



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

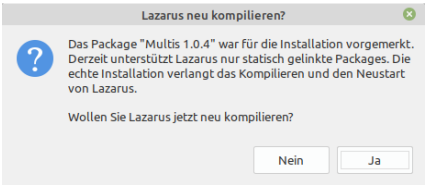


Es öffnet sich folgendes Fenster:

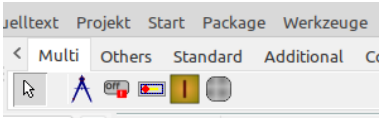


Hier auf Verwenden und dann Installieren.

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!)

DropDownMenu.Speed	: Die Zeichengeschwindigkeit (Timer Intervall)
DropDownMenu.Step	: Der Zeichenschritt (in Pixel)
DropDownMenu.Stretched	: Eigenschaften des gedehnten Panels
DropDownMenu.Stretched.Active	: Macht die Auswahl zum Startwert
DropDownMenu.Stretched.Height	: Die Höhe des gedehnten Panels
DropDownMenu.Stretched.Width	: Die Breite des gedehnten Panels
DropDownMenu.Trigger	: 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 Hinweisfenster 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
Left	: Die Clientkoordinate des linken Rands des Steuerelements
RndRctRadius	: Eckendurchmesser wenn geometrische Form ist RoundRect
Style	: Die geometrische Form des Panels
Top	: 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
Appear	: läßt das Panel erscheinen (nur zur Laufzeit!)
Disappear	: läßt das Panel verschwinden (nur zur Laufzeit!)
AnimationSpeed	: Geschwindigkeit für Appear bzw. Disappear (default 0,05) (nur zur Laufzeit!)
ParentAsBkgrd	: Hinterggrund 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 DropDownMenu.Active	: Handler wenn das Panel komprimiert ist, nur aktive wenn
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
OnStreched	: 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;

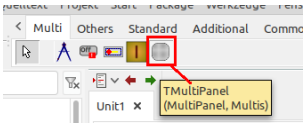
procedure MouseEnter; override;

procedure MouseLeave; override;

procedure Paint; override;

Beschreibung

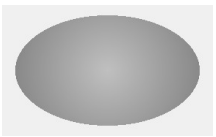
Das MultiPanel findest du im Reiter Multis



Die Form des MultiPanels lässt sich mit der Eigenschaft Style beeinflussen.

mpsRect:

mpsEllipse:



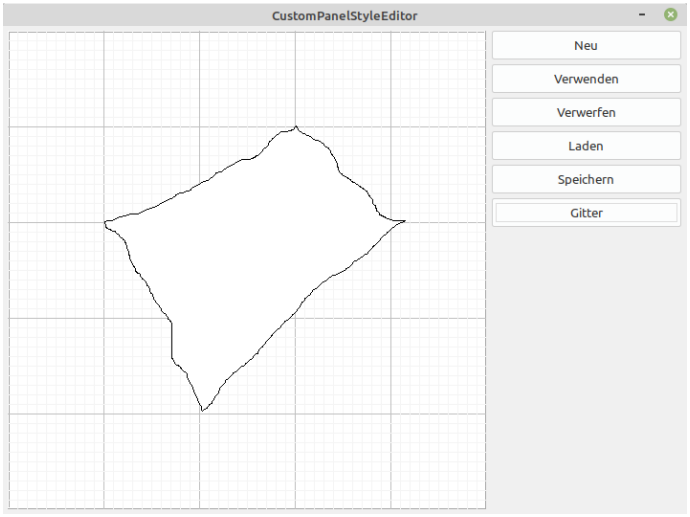
mpsRoundRect:

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:



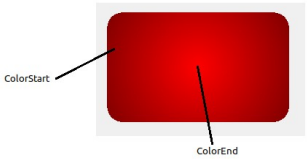
Klickt man jetzt auf Neu kann man einfach mit der Maus die Form des MultiPanels zeichnen. Klickt man auf Verwenden wird die MultiPanel-Form übernommen. Mit Verwerfen wird die MultiPanel-Form nicht übernommen und der Editor geschlossen. Mit Speichern kann man eine gezeichnete Form abspeichern und mit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

Bedenke es muss mpsCustom eingestellt sein!

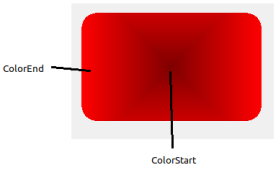
Zur Laufzeit lassen sich auch mit LoadFromFile vorab gespeicherte MultiPanel‘s laden.

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:



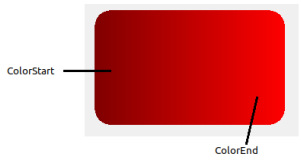
gcRadiant:



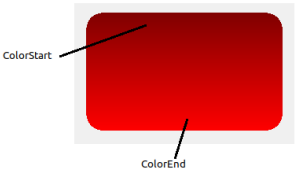
gcAlternate:

Setzt abwechselt ein Pixel je auf Start- und EndColor.

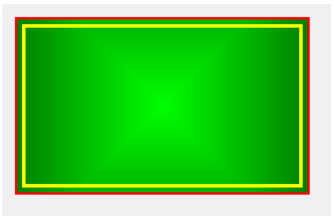
gcHorizontal:

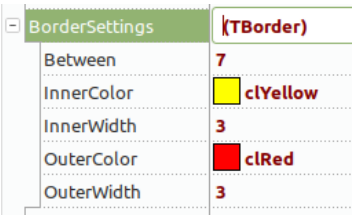


gcVertical:



Wenn man den Rand hervorheben möchte stehen die Eigenschaften des BorderSettings zur Verfügung.





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.

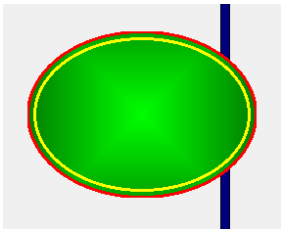
Standartwert ist 0,05. Je kleiner der Wert desto langsamer geht die Animation vonstatten. Bei einem Wert von 0,001 geht es schon sehr langsam.

Beispiel-Code:

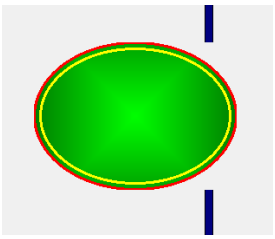
```
MultiPanel1.AnimationSpeed:= 0.001;
```

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

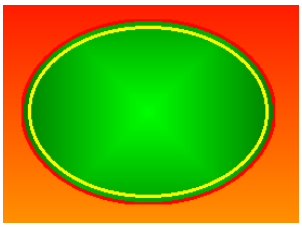


Diese Einstellung macht vor allem Sinn wenn der Parent seine Größe ändert. Denn dann wird zum Beispiel hier die gezeichnete Linie nicht richtig skaliert.

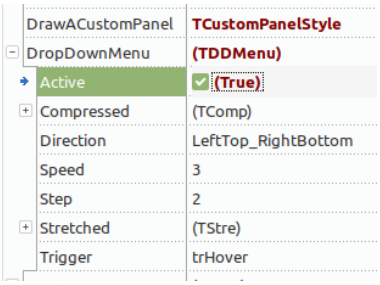
Besitzt der Parent einen Farbverlauf besteht die Möglichkeit die Skalierungsproblematik durch Aufruf der Prozedure InvalidateBackground auszugleichen.

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.



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-Controlls (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 Schaltfäche 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 Hinweisfenster 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

MessageButtonBorderWith : Die Dicke des Rahmens

MessageButton.CalculateAlthoughInvisible : Wird benötigt wenn der MessagButton 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

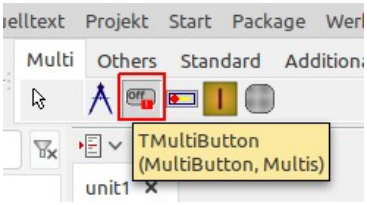
MessageButton.Visible	: Ermöglicht das Ein- oder Ausblenden des Steuerelements und aller seiner untergeordneten Elemente
MessageButton.Width	: Die horizontale Ausdehnung des MessageButtons
MultiButton_StyleManager	: Vereinfacht das Gestalten des MultiButtons
PopupMenu	: Ein Menü das angezeigt wird, wenn Sie mit der rechten Maustaste auf dieses Steuerelement klicken
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)
RndRctRadius	: Eckendurchmesser wenn geometrische Form ist RoundRect
ShowBorder	: Ermöglicht das Ein- oder Ausblenden eines Rahmens
ShowHint	: Wenn True, wird der Hinweistext angezeigt, wenn sich die Maus über dem Steuerelement befindet
ShowMsgButtonInGroup	: Zeigt den Message Button auf einem MultiButton in einer Gruppe
Style	: Die geometrische Form des Buttons
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. Die Breite des MultiButtons ist minus HoverFrameWidth

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 FocusAlphaBValue kann die Transparenz des Fokusrahmens geregelt werden. Der Wert 0 bedeutet durchsichtig und 255 undurchsichtig. FocusFrameWidth bestimmt die Dicke des Rahmens.

Wert 50:

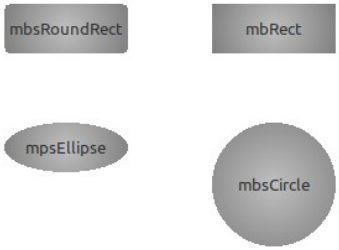
Wert 200:

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

Mit ForegroundFocusOn besitzt 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!

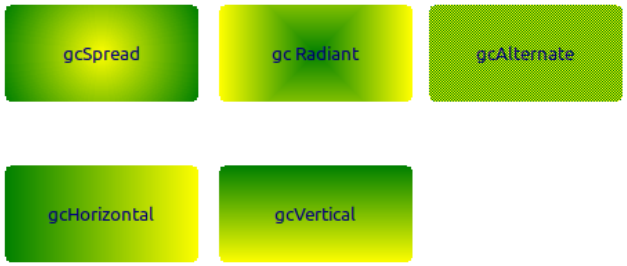
Mit der Eigenschaft Style stellt man die gewünschte geometrische Form des MultiButtons ein.

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 einfarbige MultiButton zu bekommen muss ColorStart und ColorEnd gleich sein. Ansonsten bestimmt die Zusammensetzung aus den drei Eigenschaften das Aussehen.



Alternate setzt abwechselt 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.

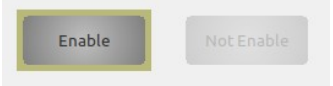
HoverEndColor	: Die Endfarbe eines Hoverereignisses
HoverFontColor	: Die Farbe der Caption während eines Hoverereignisses
HoverImageIndex	: Der Index eines Bildes in einer ImageList während eines Hoverereignisses
HoverOn	: Ermöglicht das Ein- oder Ausblenden eines Hoverereignisses
HoverStartColor	: Die 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 DisabledAlphaBValue und DisabledColor beeinflusst werden.

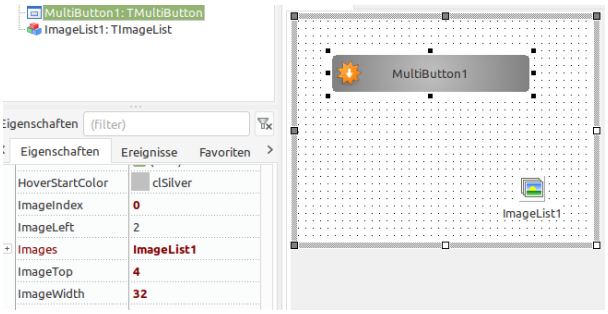
Möchte man keinen Rahmen sehen DisabledColor auf die gleiche Farbe wie den Parent stellen.



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!

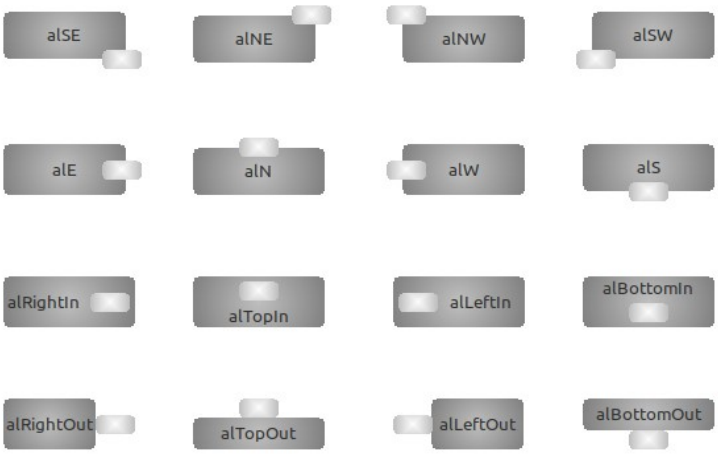
Wenn es benötigt wird kann der MultiButton einer Gruppe angehören. Dies erreicht man mit der Eigenschaft GroupIndex. Besitzt also ein MultiButton einen anderen Wert als 0 gehört er zu der Gruppe mit dem gleichen Wert. In einer Gruppe kann nur ein MultiButton gedrückt sein. Soll ein Button in der Gruppe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


```
MultiButton1.GroupIndex:=1;
MultiButton2.GroupIndex:=1;
MultiButton3.GroupIndex:=1;  MultiButton3.Down:=true;
```

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

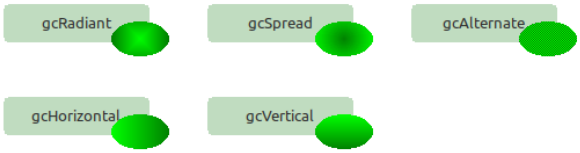
Möchte man die Form des MessageButtons ändern geschieht dies mit der Eigenschaft MessageButton.Style.

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 MessageButtonBorderWith 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 abwechselt 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 Hoverereigniss (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.PresdColBlendVal hinterlegtem Wert über den MessageButton geblendet. Wobei 0 transparent bedeutet und 255 undurchsichtig.