

TFlexiSwitch

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)
Checked	: Checked entspricht der Direction Right, aus Kompatibilität zu Radiobuttons eingefügt. Nur zur Laufzeit verfügbar!
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 Fokusfarbe ist (1=undurchsichtig, 0=durchsichtig). Nur zur Laufzeit!
FocusColor	: Die Farbe wenn das Control den Fokus hat (clNone = keine Fokus-Anzeige)
FocusedBlendFaktor	: Wie transparent die Fokusfarbe ist (1=undurchsichtig,0=durchsichtig)
Font	: Die Schrift die für die Textanzeige in diesem Schalter verwendet werden soll.
GroupIndex	: Der Index der Gruppe zu der der FlexiSwitch 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 Hinweisfenster 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

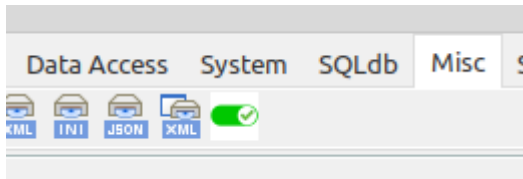
procedure **LoadImagesFromFile**(LeftFilename,RightFilename: string);

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 FlexiSwitch findest du im Reiter Misc:



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 FlexiSwitch 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 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>

Mit der Eigenschaft `FocusColor` kann man die Farbe des Fokusrahmens einstellen. Mit `FocusedBlendFactor` kann die Transparenz des Fokusrahmens geregelt werden. Der Wert 0 bedeutet durchsichtig und 1 undurchsichtig.

WERT 0:



Wert 0,5:



Wert 1:



Mit der Eigenschaft `Direction` kann man einstellen ob der Rollbutton zu Beginn links oder rechts ist.

fsLeft:



fsRight:



Mit `SwitchMode` wechselt man zwischen Klicken und Schieben um den Schalter zu schalten.

Wählt man fsClick genügt ein Klick in den Schalter um den Zustand zu wechseln. Ist fsSlide eingestellt muss man den Rollbutton mit der Maus auf die andere Seite ziehen. Man erkennt an einem Hand-Cursor das fsSlide 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 `LeftBgColor` und `RightBgColor` lässt sich die Hintergrundfarbe des Controlls verändern. Left und Right bezieht sich dabei immer auf die Lage des Rollbuttons.



Ändert man `BorderColor` auf `<> cNone` wird eine Border in der gewählten Farbe gezeichnet.



Möchte man kein Rollimage verwenden sondern einen einfarbigen Button haben muss man **ButtonColor** auf `<> clNone` stellen. Der Button wird dann in der gewählten Farbe gezeichnet.



Um einen Hover-Effekt zu bekommen stellt man **HoverColor** auf `<> clNone`. Die Transparenz der Hoverfarbe kann man mit **HoverBlendFactor** verändern. Wobei 0 durchsichtig und 1 undurchsichtig entspricht.

Ohne Hover



Blendfaktor 0.2



Blendfaktor 0.5

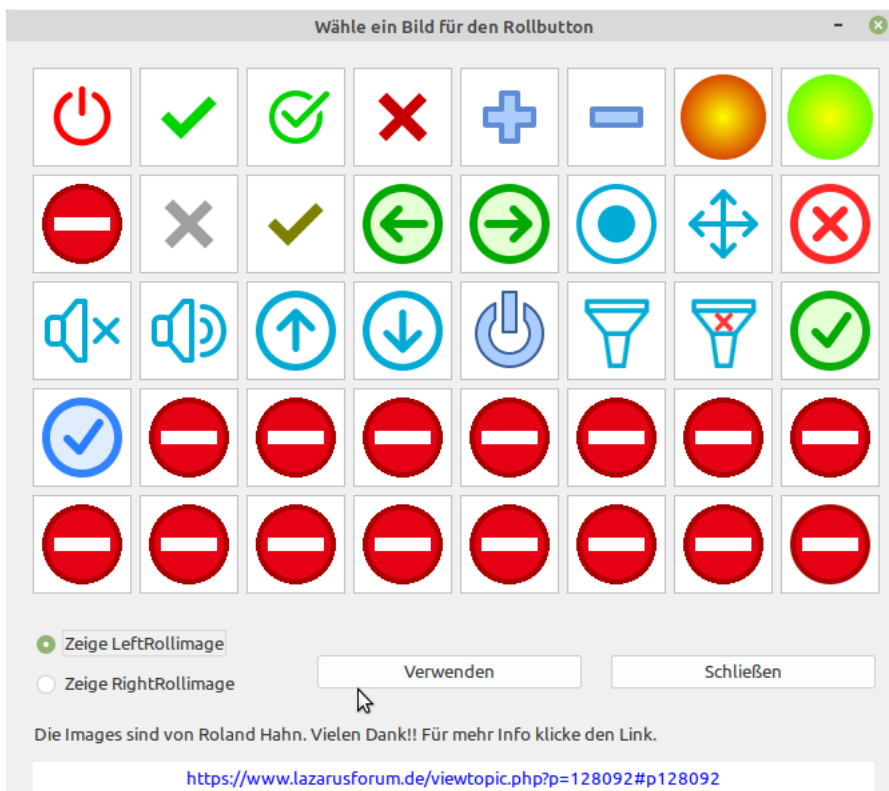


Stellt man **Enable** auf false, so das der Switch nicht mehr auf Eingaben reagiert, wird dies durch eine veränderte Farbe kenntlich gemacht. Die Farbe kann man mit **DisabledColor** und die Transparenz mit **EnabledBlendFactor** einstellen. Beide Properties stehen nur zur Laufzeit zur Verfügung!

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 FlexiSwitch übernommen und gleich angezeigt.

Wie zu sehen sind noch einige Plätze frei um eigene Images nach zu laden. Die Resource Datei befindet sich im Flexi 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 FlexiSwitches von denen immer nur einer geschaltet sein soll so muss man den [GroupIndex](#) auf $\neq 0$ stellen.



Der FlexiSwitch der initial alleine steht bestimmt die Schaltrichtung.



Um eine Kompatibilität mit Radiobuttons zu erzielen gibt es zur Laufzeit die Eigenschaft [Checked](#).

Checked entspricht fsRight, not checked fsLeft.

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

Das Ereignis [OnLeft](#) gibt einen Truewert 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 Controlls ankommt.

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