

# Delphi-Komponente "TAlphaPanel" in Unit "ALPHABAR"

TObject  
TPersistent  
TControl  
TWinControl  
TCustomPanel  
**TAlphaPanel**



Die Komponente TAlphaPanel ist eine Panel-Komponente mit Speedbuttons. TAlphaPanel ist als Index-Leiste konzipiert: Dem Anwender soll die Möglichkeit geboten werden, mit einfachem Klick auf einen Buchstaben-Schalter, den nächsten Index-Punkt zu erreichen. Dies könnte z. B. in einer indizierten Datentabelle sein. Dann müßte nach einem Klick auf das 'K' der erste Datensatz, dessen Indexwert mit 'K' beginnt, angezeigt werden. Man kann TAlphaPanel jedoch auch als "Ersatz-Tastatur" verwenden, z. B. um schnell Sonderzeichen eingeben zu können.

TAlphaPanel löst bei einem Klick auf die Leiste lediglich das Ereignis "OnValueChanged" aus, auf das die Anwendung dann zu reagieren hat.

Die Anordnung, Größe und Titel der Schalter in der Leiste kann über zahlreiche Eigenschaften konfiguriert werden:  $\Rightarrow$  "AllowAllUp", "AlphaButtons", "BlankXSize", "BlankYSize", "ButtonFont", "ButtonHeight", "ButtonLeftMargin", "ButtonTopMargin", "ButtonWidth", "ButtonXSpacing", "ButtonYSpacing", "CatchButtons".

## Eigenschaften

### geerbt

**Folgende geerbte Eigenschaften entsprechen denen von TPanel:**

Align, BevelInner, BevelOuter, BevelWidth, BorderStyle, Color, Ctl3D, Cursor, Enabled, Locked, ParentColor, ParentCtl3D, ParentShowHint, PopupMenu, ShowHint, Visible.

### ActiveButton

**property ActiveButton :char default #0**

Dient zum Auslesen des zuletzt gedrückten Schalters. Der Wert ist <#0>, wenn noch kein Schalter in der aktuellen Buttonanordnung gedrückt wurde. Wenn die Eigenschaft  $\Rightarrow$  "CatchButtons" <TRUE> ist, kann durch Setzen dieser Eigenschaft der niedergedrückte Schalter bestimmt werden. Das Setzen von "ActiveButton" löst das Ereignis  $\Rightarrow$  "OnValueChanged" aus.

### AllowAllUp

**property AllowAllUp :boolean default false**

Bestimmt das Verhalten der Schalter, wenn  $\Rightarrow$  "CatchButtons" auf <TRUE> gesetzt ist und der aktive Schalter nochmals angeklickt wird. Ist "AllowAllUp" <FALSE>, dann bleibt ein Schalter eingerastet, wenn er nochmals angeklickt wird; es wird dann auch nochmals das Ereignis  $\Rightarrow$  "OnValueChanged" ausgelöst. Ist "AllowAllUp" <TRUE>, dann wird das Einrasten des aktiven Schalters aufgehoben und  $\Rightarrow$  "ActiveButton" auf <#0> gesetzt, was auch ein "OnValueChanged"-Ereignis auslöst.

### AlphaButtons

**property AlphaButtons :tStringList**

Über diese Eigenschaft kann die Button-Belegung der Leiste ausgelesen und eingestellt werden. Voreinstellung ist ein leeres Panel. Jede neue Zeile in der Stringliste wird auch als neue Zeile in der Buttonleiste interpretiert. Eine

## Delphi-Komponente "TAlphaPanel" in Unit "ALPHABAR"

Leerstelle innerhalb einer Zeile erzeugt einen Abstand zum nächsten Schalter (⇒"BlankXSize"). Jedes einzelne Zeichen ist auch der Titel eines einzelnen Schalters! ⇒"ActiveButton" wird auf <#0> gesetzt, wenn ein neues Schalterlayout definiert wurde. Eine leere Zeile erzeugt einen Zeilenabstand (⇒"BlankYSize").

<b>BlankXSize</b>	<b>property BlankXSize :word default 9</b> Diese Eigenschaft bestimmt wieviele Bildpunkte ein horizontaler Abstand groß ist (⇒"AlphaButtons", "ButtonXSpacing").
<b>BlankYSize</b>	<b>property BlankYSize :word default 9</b> Diese Eigenschaft bestimmt wieviele Bildpunkte ein vertikaler Abstand groß ist (⇒"AlphaButtons", "ButtonYSpacing").
<b>ButtonFont</b>	<b>property ButtonFont :tFont</b> Diese Eigenschaft bestimmt den Font, der für die Schaltertitel verwendet wird.
<b>ButtonHeight</b>	<b>property ButtonHeight :word default 18</b> Diese Eigenschaft bestimmt die Höhe eines jeden Schalters in der Leiste.
<b>ButtonLeftMargin</b>	<b>property ButtonLeftMargin :word default 2</b> Bestimmt die Größe des linken Randes, der beim Einfügen der Schalter beachtet wird.
<b>ButtonTopMargin</b>	<b>property ButtonTopMargin :word default 2</b> Bestimmt die Größe des oberen Randes, der beim Einfügen der Schalter beachtet wird.
<b>ButtonWidth</b>	<b>property ButtonWidth :word default 18</b> Diese Eigenschaft bestimmt die Breite eines jeden Schalters in der Leiste. Ein manueller Abstand zwischen den Buttons entspricht ⇒"BlankXSize"
<b>ButtonXSpacing</b>	<b>property ButtonXSpacing :integer default -1</b> Die Schalter werden standardmäßig direkt neben- bzw. untereinander dargestellt. Diese Eigenschaft sorgt dafür, daß die Schalter zusätzlich noch nach links (negativer Wert) oder rechts (positiver Wert) eingerückt werden.
<b>ButtonYSpacing</b>	<b>property ButtonYSpacing :integer default -1</b> Die Schalter werden standardmäßig direkt neben- bzw. untereinander dargestellt. Diese Eigenschaft sorgt dafür, daß die Schalter einer neuen Zeile zusätzlich noch nach oben (negativer Wert) oder unten (positiver Wert) eingerückt werden.
<b>CatchButtons</b>	<b>property CatchButtons :boolean</b> Hiermit kann gesteuert werden, ob der zuletzt gedrückte Schalter einrastet (<TRUE>) oder nicht (<FALSE>, Standard). Ist "CatchButtons" <TRUE>, wird auch die Eigenschaft ⇒"AllowAllUp" beachtet. Damit ist es einfach, die Leiste als Auswahlwechsler für Filter zu verwenden, z. B. "zeige nur Datensätze an, die mit 'F' beginnen", wenn der Schalter 'F' eingerastet ist.
<b>Height</b>	<b>property Height default 22</b> Diese geerbte Eigenschaft wird per Voreinstellung auf 22 gesetzt und ist

## Delphi-Komponente "TAlphaPanel" in Unit "ALPHABAR"

damit hoch genug für eine Zeile Schalter in der Standardhöhe.

### Width

#### **property Width default 448**

Diese geerbte Eigenschaft wird per Voreinstellung auf 448 gesetzt und damit breit genug für eine Leiste mit allen Buchstaben von 'A' bis 'Z' in der Standardbreite.

## Delphi-Komponente "TAlphaPanel" in Unit "ALPHABAR"

### Methoden - public

<b>Create</b>	<b>constructor Create(aOwner :TComponent); override</b> Initialisiert die Komponente. Standardmäßig wird ein leeres Panel erstellt.
<b>Destroy</b>	<b>destructor Destroy; virtual</b> Gibt den von der Komponente belegten Speicherplatz frei.
<b>GetButton</b>	<b>function GetButton(value :char) :TSpeedButton</b> Diese Funktion gibt einen Zeiger auf den Schalter mit dem <tag>-Wert <Ord(value)> zurück oder <NIL>, wenn kein solcher Schalter existiert. Sollte es mehrere Schalter mit dem gleichen Wert in <tag> geben, wird der erste dieser Schalter zurückgegeben. Mit dieser Funktion hat man direkten Zugriff auf die einzelnen Schalter. ⇒"AddLetterButtons"

### Methoden - protected

<b>AddLetterButtons</b>	<b>procedure AddLetterButtons; virtual</b> Diese Methode fügt in das leere TAlphaPanel die Schalter gemäß der Definitionen in ⇒"AlphaButtons" ein. Die Schalter werden von Delphi automatisch benannt. Der Ordinalwert der Schalteraufschrift wird in das Feld <tag> des TSpeedButtons eingetragen. Nur der Wert in <tag> dient zur weiteren Identifikation der Schalter! ⇒"GetButton"
<b>DestroyButtons</b>	<b>procedure DestroyButtons</b> Diese Methode löscht alle Schalter, die im TAlphaPanel vorhanden sind, aus dem Speicher.
<b>Loaded</b>	<b>procedure Loaded; override</b> Die Schalter-Komponenten werden nicht in die Formulardatei geschrieben, sondern lediglich die Stringliste ⇒"AlphaButtons". Deshalb wurde die Methode "Loaded" so angepaßt, daß nach dem erfolgreichen Laden der Eigenschaftswerte die Schalter erzeugt werden.

### Ereignisse

<b>OnValueChange</b>	<b>property OnValueChange :TNotifyEvent</b> Dieses Ereignis wird immer dann ausgelöst, wenn im TAlphaPanel ein Schalter angeklickt wurde. Welcher Schalter dies war, kann aus ⇒"ActiveButton" gelesen werden. ⇒ "AllowAllUp", "AlphaButtons", "Create".
----------------------	--

## VCL-Komponente "TAlphaPanel" - Version 1.0

© 1995 by  
**Ingo Humann**  
**Mühlstraße 3**

**Delphi-Komponente "TAlphaPanel" in Unit "ALPHABAR"**

**67105 Schifferstadt**

**GERMANY**

**CIS: 100116,3354**

**Internet: 100116.3354@compuserve.com**