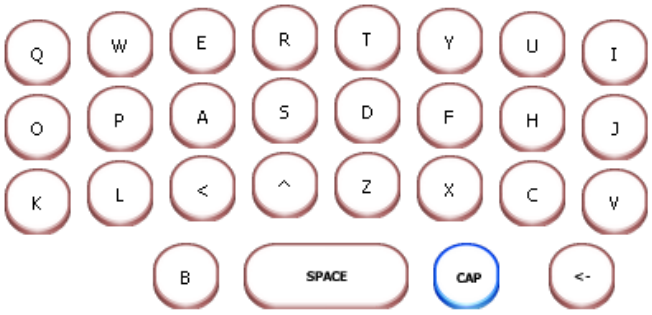
TADVTOUCHKEYBOARD
DEVELOPERS GUIDE

Documentation : Dec, 2016

Web: http://www.tmssoftware.com
Email : info@tmssoftware.com

Overview



TMS touchscreen keyboard components offer TAdvTouchkeyboard & TAdvPopupTouchkeyboard. TAdvTouchkeyboard and TAdvPopupTouchkeyboard are components to display a configurable onscreen keyboard for touch screen applications. The keyboard has built-in settings for QWERTY, AZERTY, DVORAK, NUMERIC, CELLPHONE keyboards but also supports fully customizable keyboard layouts.

Usage

Using TAdvTouchkeyboard is simple. The built-in keyboard layouts can be selected with the property AdvTouchkeyboard.KeyboardType property. Drop the component on the form and when the property AutoPostKey = true, it will automatically post the keyvalue for the keyboard key pressed to the active focused control. Alternatively, if it is not desirable to have TAdvTouchkeyboard automatically send keyboard keys, key presses can be handled through the event OnKeyDown or OnKeyClick event. The OnKeyDown event returns the key code and shift state. The OnKeyClick event returns the index of the key clicked in the TAdvTouchkeyboard.Keys collection.
Special keys such as Shift, Alt, Ctrl, Alt Gr are sticky keys. This means that the first click on the key puts the key in down state and the second click puts the key back in normal state. The sticky behaviour makes it possible to access all key states with single clicks.

TAdvPopupTouchkeyboard works very similar except that it is shown in a popup window. To use TAdvPopupTouchkeyboard, drop it on a form and call TAdvPopupTouchkeyboard.Show to show at default position or TAdvPopupTouchkeyboard.ShowAtXY(x,y) to show at x,y screen coordinates. TAdvPopupTouchkeyboard has the additional capability that its position can track the edit or memo control that has focus and can automatically display and hide when an edit/memo control has focus or not. To do this, set TAdvPopupTouchkeyboard.AutoFollowFocus and TAdvPopupTouchkeyboard.AutoHide to true.

Appearance

TAdvTouchkeyboard has many options to control the appearance. The background color is set through the TAdvTouchkeyboard.Color property. By default, keys are rounded flat rectangles with a color/bordercolor for normal state and color/bordercolor for down state, set for each key with TAdvTouchKeyItem.Color, TAdvTouchKeyItem.ColorDown, TAdvTouchKeyItem.BorderColor, TAdvTouchKeyItem.BorderColorDown.

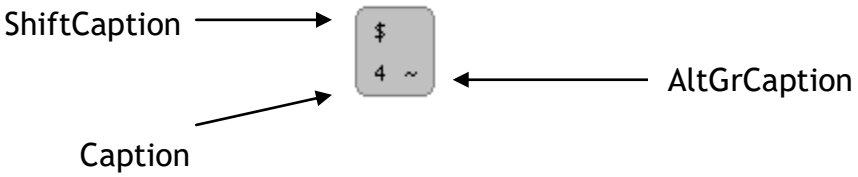
A key can also use a bitmap as background for normal state and down state. This can be set through TAdvTouchkeyboard.PictureDownState, TAdvTouchkeyboard.PictureNormalState. To use the same background picture sizing for different key sizes, a special bitmap stretching technique is applied. The borders & corners are preserved and the inner of the bitmap is stretched only. Unlike normal physical keyboards, the software touchscreen keyboard can also display keys different for Shift &

Alt-Gr states. When AutoCapsDisplay is true, the normals key caption displays either in uppercase or lowercase. When HighlightCaps or HighlightAltGr is a color different from clNone, keys with multiple values depending on Shift or Alt-Gr key state will show the caption in the appropriate color. For example, if the key has the value of Euro currency (€), when Alt-Gr is pressed, the key can show € in a special color set by HighlightAltGr.

Customization

Keyboard layouts can be fully customized with TAdvTouchkeyboard. To allow this, the TAdvTouchkeyboard has a collection of TAdvTouchkeyItem objects that control position, appearance and behaviour of each key on the keyboard. A TAdvTouchKeyItem has following properties:

AltGrCaption : text on the key to display for Alt-Gr combination (bottom right of key)
AltGrKeyValue : value of key when pressed in combination with Alt-Gr key, default -1
BorderColor : color of key border in normal state
BorderColorDown : color of key border in down state
Caption : text on the key (centered in key)
Color : color of the key in normal state
ColorDown : color of the key in down state
Height : height of the key in pixels
ImageIndex : index of image in associated TAdvTouchkeyboard imagelist to display on key
KeyValue : value of key when pressed in normal state, default -1
PictureDownState : background bitmap of key in down state
PictureNormalState : background bitmap of key in normal state
ShiftCaption : text on the key to display for Shift combination (top of key)
ShiftKeyValue : value of key when pressed in shift state, default -1
SpecialKey : sets key as special key : Caps, Ctrl, Alt, Alt-Gr, Shift, Enter, Tab, Spacebar, Return, Multiply, Divide, Delete, Subtract, Add, App, Win, Scroll, Num
TextColor : color of text on key in normal state
TextColorDown : color of text on key in down state
Width : width of the key
X : X position of the key on the keyboard
Y : Y position of the key on the keyboard



When SpecialKey is skNone, a key can send values to the keyboard for normal state, shift state and in combination with Alt-Gr key. When KeyValue, ShiftKeyValue, AltGrKeyValue are default -1, the key that is sent is the first letter of the text for the key state, ie. the first letter of Caption, ShiftCaption, AltGrCaption. When the KeyValue property is different from -1, the keyvalue specified is sent. For keys where SpecialKey is different from skNone, the special key value is automatically sent.

Saving & loading keyboard layouts

Keyboard layouts can be saved & loaded at any time. For this, TAdvTouchkeyboard exposes two methods:

SaveKeybLayout(FileName: string);
LoadKeybLayout(FileName: string);

Tips & FAQ

How to increase the size of the buttons

Use the function AdvTouchKeyboard.Zoom() , for example:

```
advtouchkeyboard1.Zoom(1.5,1.5);
```

How to calculate precisely where the TAdvPopUpTouchKeyBoard will appear

You can see the code how the coordinates are automatically calculated in the source code of ADVTOUCHKEYBOARD.PAS under procedure TAdvPopupTouchKeyBoard.TimerProc(Sender: TObject); The function GetWindowRect(wnd,r); gets the coordinates of the active edit control and this is used to position the keyboard.