

##\$SKK **CapCtrl Unit**

See Also

The CapCtrl unit declares the TCaptionControl component. This component can add a gradient fill to a form's caption, add caption buttons in addition to the default Close/Minimize/Maximize/Help buttons, and activate a popup menu in response to right-mouse-button clicks.

The following items are declared in the CapCtrl unit:

Classes

[TCaptionButtonsList](#)

[TCaptionButtonsListProperty](#)

Components

[TCaptionControl](#)

Types

[ECaptionError](#)

[TCaptionButton](#)

[TCaptionButtonClickEvent](#)

[TCaptionButtonDrawEvent](#)

[TCaptionButtonKind](#)

[TCaptionGradient](#)

[TDrawCaptionEvent](#)

[TDirection](#)

CapCtrl

§ CapCtrl Unit

κ CapCtrl;Components;Objects;Types

##\$KKB{bml capedith.bmp }TCaptionButtonsList

[Properties](#) [Methods](#)

Unit

[CapCtrl](#)

Description

TCaptionButtonsList is a list of TCaptionButton objects. The Buttons property is the default property of the class (meaning you can reference a button on the list by using MyBtnList.Buttons[i] or MyBtnList[i]). The Count property is a read only property indicating the number of buttons on the list.

TCaptionButtonsList defines several routines to manage the list. Add adds a TCaptionButton to the list. AddButton adds a button, but does not require a previously created TCaptionButton object, but creates it on the fly. Insert inserts a TCaptionButton at the specified location. Clear deletes all buttons from the list and destroys the related objects. Delete deletes a single button from the list and destroys it.

```
# TCaptionButtonsList
$ TCaptionButtonsList
κ TCaptionButtonsList;Components
B class_TCaptionButtonsList
```

See also

[TCaptionButtonsListProperty](#)

[TCaptionButtonsListProperty_Tasks_See_also](#)
§ [TCaptionButtonsListProperty Tasks See also](#)

##Properties

Buttons Count

TCaptionButtonsList_Properties

###\$KBB **Buttons Property**

[See Also](#)

Applies To

[TCaptionButtonsList](#)

Declaration

```
property Buttons[Index: Integer]: TCaptionButton;
```

Description

The Buttons property is the default property of TCaptionButtonsList. It lists the TCaptionButton objects and allows access to them. The array is zero-based, that is, the first button is Buttons[0].

```
# TCaptionButtonsList_Properties_Buttons  
$ Buttons Property  
κ Buttons;Properties;  
B prop_TCaptionButtonsListButtons
```

##\$**Properties**

Caption Kind Visible
Enabled Pushed

TCaptionButton_Properties
§ TCaptionButton Properties

##\$KBB **Count Property**

Applies To

[TCaptionButtonsList](#)

Declaration

```
property Count: Integer;
```

Description

Count indicates the number of buttons on the list. If Count is zero, there are no buttons. Otherwise, there are Count buttons, from Buttons[0] to Buttons[Count-1].

```
# TCaptionButtonsList_Properties_Count  
$ Count Property  
κ Count;Properties;  
B prop_TCaptionButtonsListCount
```

##\$ **Methods**

GetBtnKindStr Draw SetBtnKindStr

TCaptionButton_Methods
§ TCaptionButton Methods

##Methods

Add Clear Insert
AddButton Delete

TCaptionButtonsList_Methods

##\$SKK **Add Method**

Applies To

[TCaptionButtonsList](#)

Declaration

```
procedure Add(Button: TCaptionButton);
```

Description

The Add methods adds a button to the list. The actual instance passed in the Button parameter is added to the list. Destroying that instance without removing it from the list would cause the list to include a pointer to a non-existing object.

```
# TCaptionButtonsList_Methods_Add  
$ Add Method  
κ Add;Methods;
```

##\$KK **AddButton Method**

Applies To

[TCaptionButtonsList](#)

Declaration

```
procedure AddButton(Caption: String; Enabled, Visible, Pushed: Boolean;Kind:
TCaptionButtonKind);
```

Description

AddButton creates a new TCaptionButton objects with the properties defined in the procedure's parameters and adds it to the list.

```
# TCaptionButtonsList_Methods_AddButton
$ AddButton Method
κ AddButton;Methods;
```

##\$KK **Clear Method**

Applies To

[TCaptionButtonsList](#)

Declaration

```
procedure Clear;
```

Description

Clear deletes all buttons from the list. When a button is deleted, it is also destroyed and its memory is freed.

```
# TCaptionButtonsList_Methods_Clear  
$ Clear Method  
κ Clear;Methods;
```

##\$SKK **Delete Method**

Applies To

[TCaptionButtonsList](#)

Declaration

```
procedure Delete(Index: Integer);
```

Description

Delete deletes and frees an object from the list.

It is important to note that the object is freed. This is unlike the standard list classes, which remove an object from the list but do not free it.

```
# TCaptionButtonsList_Methods_Delete  
$ Delete Method  
κ Delete;Methods;
```

##\$KK**Insert Method**

Applies To

[TCaptionButtonsList](#)

Declaration

```
procedure Insert(Index: Integer; Button: TCaptionButton);
```

Description

Insert inserts a button to the list at the specified index. Index must be valid.

```
# TCaptionButtonsList_Methods_Insert  
$ Insert Method  
κ Insert;Methods;
```

##\$KKB{bml capedith.bmp }**TCaptionButtonsListProperty Component**

[Methods](#) [Tasks](#)

Unit

[CapCtrl](#)

Description

TCaptinButtonsListProperty is the custom property editor for the TCaptionButtonsList type. A custom property editor is required since the type cannot be edited by any of the standard property editors.

```
# TCaptionButtonsListProperty
$ TCaptionButtonsListProperty Component
κ TCaptionButtonsListProperty;Components
B class_TCaptionButtonsListProperty
```

##\$SKK{bml capedith.bmp }**Using the TCaptionButtonsListProperty Component**

[See Also](#)

Purpose

TCaptionButtonsListProperty is the property editor for the TCaptionButtonsList type. It is not intended to be used by any application. It is documented here to provide some information on its implementation and usage.

General

The property editor overrides three methods of TClassProperty. The GetValue method is used to get the text that is displayed by Delphi's Object Inspector. GetAttributes returns information about the property editor, and Edit edits the property and sets the Form Designer's modified state to inform Delphi of any changes.

TCaptionButtonsListProperty_Tasks
§ Using the TCaptionButtonsListProperty Component
κ TCaptionButtonsListProperty

##Methods

Edit GetAttributes: GetValue:

TCaptionButtonsListProperty_Methods

##\$K Edit Method

Applies To

[TCaptionButtonsListProperty](#)

Declaration

```
procedure Edit; override;
```

Description

The Edit methods creates the property editor dialog. It retrieves actual property data (a TCaptionButtonsList object) by calling GetOrdValue:

```
if EditCaptionButtons (TCaptionButtonsList (GetOrdValue) ,  
    TCaptionControl (GetComponent(0))) then Modified;
```

EditCaptionButtons is the dialog function. It returns True if the data was modified. The return value from GetOrdValue is typecast to TCaptionButtonsList and points to the actual property being edited.

GetComponent(0) returns the component containing the property (the TCaptionControl). If

EditCaptionButtons indicates the the property has changed the Modified method is called, notifying Delphi that data has changed and should be saved.

```
# TCaptionButtonsListProperty_Methods_Edit  
$ Edit Method  
κ Edit;Methods;
```

##\$KK **GetAttributes Method**

Applies To

[TCaptionButtonsListProperty](#)

Declaration

```
function GetAttributes: TPropertyAttributes; override;
```

Description

GetAttributes returns [paDialog, paReadOnly]. This indicates to Delphi that in order to edit the property, the user should double-click the "..." button for that property in the Object Inspector, and that Delphi should call the Edit method.

```
# TCaptionButtonsListProperty_Methods_GetAttributes:  
§ GetAttributes Method  
κ GetAttributes;Methods;
```

##\$KK **GetValue Method**

Applies To

[TCaptionButtonsListProperty](#)

Declaration

```
function GetValue: String; override;
```

Description

GetValue returns '(Buttons)'. This text will be displayed by Delphi in the Object Inspector window.

```
# TCaptionButtonsListProperty_Methods_GetValue:  
$ GetValue Method  
κ GetValue;Methods;
```

##\$KKB{bml capctrlh.bmp } **TCaptionControl Component**

[Properties](#) [Methods](#) [Events](#) [Tasks](#)

Unit

[CapCtrl](#)

Description

TCaptionControl controls and modifies a form's caption bar.

Use the CaptionGradient property to set a gradient fill for the caption. Use the Buttons property to add and manage new buttons on the forms caption bar. Set PopupMenu to allow a menu to be displayed when the caption bar is clicked with the right mouse button, instead of the default Windows system menu. Set CaptionDirection to align the caption text and WindowDirection to align the window under the Middle-East platform (Arabic and Hebrew Windows).

```
# TCaptionControl  
$ TCaptionControl Component  
κ TCaptionControl;Components  
B class_TCaptionControl
```

##\$SKK{bml capctrlh.bmp }**Using the TCaptionControl Component**

Purpose

TCaptionControl controls and modifies a form's caption bar. It can add a gradient fill and buttons, set a context popup menu, control caption text display and support for Windows 95 Middle Eastern (Arabic/Hebrew) platform with right-aligned text and window direction.

TCaptionControl_Tasks
\$ Using the TCaptionControl Component
κ TCaptionControl

##Properties

<u>Buttons</u>	<u>ColorBands</u>	<u>ShowButtons</u>
<u>ButtonsDirection</u>	<u>Enabled</u>	<u>WindowDirection</u>
<u>CaptionDirection</u>	<u>PopupMenu</u>	<u>CaptionGradient</u>
<u>RtlReading</u>		

TCaptionControl_Properties

##\$KBB **Buttons Property**

[See Also](#)

Applies To

[TCaptionControl](#)

Declaration

```
property Buttons: TCaptionButtonsList;
```

Description

The Buttons property lists the buttons that will be displayed on the form's caption at run-time. A button's appearance is determined by setting its Kind property, providing an event handler for TCaptionControl's OnCaptionButtonDraw event creating a descendant of TCaptionButton and adding it to the list at run-time.

```
# TCaptionControl_Properties_Buttons  
$ Buttons Property  
K Buttons;Properties;  
B prop_TCaptionControlButtons
```


See also

[TCaptionButton](#)

[TCaptionButtonsList_Properties_Buttons_See_also](#)
§ [TCaptionButtonsList_Properties_Buttons_See_also](#)

##\$KKB **ButtonsDirection Property**

[See Also](#)

Applies To

[TCaptionControl](#)

Declaration

```
property ButtonsDirection: TDirection;
```

Description

ButtonsDirection controls the alignment of the buttons on the caption bar. The buttons are always drawn left-to-right (the first button on the list is the leftmost button). The default value for ButtonsDirection is dirRight, which draws the buttons on the right side of the caption (next to the standard buttons). Setting this property to dirLeft causes the buttons to be drawn on the left side of the caption (next to the system menu icon).

```
# TCaptionControl_Properties_ButtonsDirection  
$ ButtonsDirection Property  
κ ButtonsDirection;Properties;  
B prop_TCaptionControlButtonsDirection
```

See also

[TCaptionButton](#)

[TCaptionButtonsList](#)

TCaptionControl_Properties_Buttons_See_also

§ TCaptionControl Properties Buttons See also

##\$KBB **CaptionDirection Property**

[See Also](#)

Applies To

[TCaptionControl](#)

Declaration

```
property CaptionDirection: TDirection;
```

Description

CaptionDirection controls the alignment of the caption text. The default is dirLeft (Windows 95 default). Setting the property to dirRight causes the text to be displayed on the right side of the caption. If the Enabled property is set to False, Windows performs the actual drawing of the caption. In that case, if the property is set to dirRight, the text will be left-aligned on all Windows platforms except the Middle-East platform.

```
# TCaptionControl_Properties_CaptionDirection  
$ CaptionDirection Property  
κ CaptionDirection;Properties;  
B prop_TCaptionControlCaptionDirection
```

See also

[TDirection](#)

TCaptionControl_Properties_ButtonsDirection_See_also
§ TCaptionControl_Properties_ButtonsDirection_See_also

##\$KKB **CaptionGradient Property**

[See Also](#)

Applies To

[TCaptionControl](#)

Declaration

```
property CaptionGradient: TCaptionGradient;
```

Description

CaptionGradient determines the appearance of the gradient fill, if any. Setting this property to cgNone prevents the gradient fill from showing, making the caption look like any other window. Setting the property to cgActive causes a gradient fill to be displayed only when the form is active. Setting it to cgAlways displays a gradient fill both when the form is active and inactive. The gradient color is based on the system active/inactive caption color.

```
# TCaptionControl_Properties_CaptionGradient  
$ CaptionGradient Property  
κ CaptionGradient;Properties;  
B prop_TCaptionControlCaptionGradient
```

See also

[TDirection](#)

TCaptionControl_Properties_CaptionDirection_See_also
§ TCaptionControl_Properties_CaptionDirection_See_also

##\$KBB **Enabled Property**

Applies To

[TCaptionControl](#)

Declaration

```
property Enabled: Boolean;
```

Description

The Enabled property determines the enable state of the control. When set to True, all features are activated (the display of caption buttons also depends on the setting of the ShowButtons property). When set to False, all caption bar related events are handled by Windows.

```
# TCaptionControl_Properties_Enabled  
$ Enabled Property  
κ Enabled;Properties;  
B prop_TCaptionControlEnabled
```


See also

[TCaptionGradient](#)

[TCaptionControl_Properties_CaptionGradient_See_also](#)
§ [TCaptionControl_Properties_CaptionGradient_See_also](#)

##\$KKBB **PopupMenu Property**

Applies To

[TCaptionControl](#)

Declaration

```
property PopupMenu: TPopupMenu;
```

Description

PopupMenu points to a TPopupMenu to be displayed when the users right-clicks the caption bar. If no popup-menu is assigned or the popup-menu's AutoPopup property is set to False, the popup-menu will not be displayed.

```
# TCaptionControl_Properties_PopupMenu  
$ PopupMenu Property  
κ PopupMenu;Properties;  
B prop_TCaptionControlPopupMenu
```

See also

[TDirection](#)

TCaptionControl_Properties_WindowDirection_See_also
§ TCaptionControl_Properties_WindowDirection_See_also

##\$KBB **RtlReading Property**

Applies To

[TCaptionControl](#)

Declaration

```
property RtlReading: Boolean;
```

Description

RtlReading sets the caption text's reading order when running under the Middle-East platform. Setting it to True causes Arabic and Hebrew text to be displayed properly. This property ignores the setting of the Enabled property, allowing to control caption text without using any of the other features by using the ME platform's supported styles.

Note: RtlReading is ignored under non-ME platforms. Borland has released an arabic version of Delphi 2.0, which supports some of the ME platform's capabilities. The TCaptionControl component has not been tested under the Arabic version of Delphi.

```
# TCaptionControl_Properties_RtlReading  
$ RtlReading Property  
κ RtlReading;Properties;  
B prop_TCaptionControlRtlReading
```

##\$KK **Caption Property**

Applies To

[TCaptionButton](#)

Declaration

```
property Caption: String;
```

Description

The button's caption, displayed if the button is of kind `cbkCustom` and the `OnButtonDraw` event does not handle its drawing. If the caption is set to '-', the button becomes a separator. It is not displayed as a button, but rather as a small gap between buttons, like the gap between the Close and Maximize buttons.

```
# TCaptionButtonI_Properties_Caption  
$ Caption Property  
κ Caption;Properties;
```

##\$KBB **ShowButtons Property**

Applies To

[TCaptionControl](#)

Declaration

```
property ShowButtons: Boolean;
```

Description

The ShowButtons property determines whether the caption buttons are displayed.

```
# TCaptionControl_Properties_ShowButtons  
$ ShowButtons Property  
κ ShowButtons;Properties;  
B prop_TCaptionControlShowButtons
```

##\$KK **Enabled Property**

Applies To

[TCaptionButton](#)

Declaration

```
property Enabled: Boolean;
```

Description

Determines whether the button is enabled. When False, the button is grayed and does not respond to mouse events.

```
# TCaptionButton_Properties_Enabled
```

```
§ Enabled Property
```

```
κ Enabled;Properties;
```

##\$KKBB **WindowDirection Property**

[See Also](#)

Applies To

[TCaptionControl](#)

Declaration

```
property WindowDirection: TDirection;
```

Description

The WindowDirection property is relevant only in the Middle-East platform. It sets the window direction to general left- or right-aligned, controlling caption text alignment for default Windows processing and window scroll-bar location. It is independent of the Enabled property.

```
# TCaptionControl_Properties_WindowDirection  
$ WindowDirection Property  
κ WindowDirection;Properties;  
B prop_TCaptionControlWindowDirection
```


##\$KK **Kind Property**

[See Also](#)

Applies To

[TCaptionButton](#)

Declaration

```
property Kind: TCaptionButtonKind;
```

Description

Determines the button kind. The drawing on the button's surface is based on the button's kind.

```
# TCaptionButton_Properties_Kind
```

```
$ Kind Property
```

```
κ Kind;Properties;
```

##Methods

Refresh

TCaptionControl_Methods

##\$KK Refresh Method

Applies To

[TCaptionControl](#)

Declaration

```
procedure Refresh;
```

Description

Refresh causes the caption bar to redraw itself. This is necessary, for example, after changing buttons settings (adding/deleting/modifying a button).

```
# TCaptionControl_Methods_Refresh  
$ Refresh Method  
κ Refresh;Methods;
```

##Events

OnClick OnDraw OnCaption

TCaptionControl_Events

##\$KBB **OnButtonClick Event**

[See Also](#)

Applies To

[TCaptionControl](#)

Declaration

```
property OnButtonClick: TCaptionButtonClickEvent;
```

Description

Triggered when the users clicks a caption button. This event is triggered only for internal buttons, not for the standard buttons.

```
# TCaptionControl_Events_OnButtonClick  
$ OnButtonClick Event  
κ OnButtonClick;Events;  
B event_TCaptionControlOnButtonClick
```

##\$KBB **OnButtonDraw Event**

[See Also](#)

Applies To

[TCaptionControl](#)

Declaration

```
property OnButtonDraw: TCaptionButtonDrawEvent;
```

Description

The OnButtonDraw event is activated whenever a button needs drawing. An application can control a button's appearance by handling this event.

```
# TCaptionControl_Events_OnButtonDraw  
$ OnButtonDraw Event  
κ OnButtonDraw;Events;  
B event_TCaptionControlOnButtonDraw
```

##\$KBB **OnDrawCaption Event**

[See Also](#)

Applies To

[TCaptionControl](#)

Declaration

```
property OnDrawCaption: TDrawCaptionEvent;
```

Description

The OnDrawCaption event is activated when the caption text needs to be drawn. By handling this event, the application can either change the caption text without modifying the form's Caption property or provide complete drawing of the caption text.

```
# TCaptionControl_Events_OnDrawCaption  
$ OnDrawCaption Event  
κ OnDrawCaption;Events;  
B event_TCaptionControlOnDrawCaption
```

##\$SKK **ECaptionError Type**

Unit

[CapCtrl](#)

Declaration

```
ECaptionError = class(Exception);
```

Description

The ECaptionError exception is raised when creating the component if an error occurs. The exception will be raised if the component's owner is not a form or the operating system version is incompatible with the effects and appearance of TCaptionControl.

```
# ECaptionError  
$ ECaptionError Type  
κ ECaptionError;Types
```


##\$SKK **TCaptionButton Type**

[Properties](#) [Methods](#)

Unit

[CapCtrl](#)

Declaration

```
TCaptionButton = class;
```

Description

The TCaptionButton type defines a button to add to a form's caption bar. TCaptionButton is not a stand-alone component but represents a single item in a TCaptionControl's Buttons property.

TCaptionButton defines several methods and properties. Caption is the button's caption, displayed if the button is not graphic. Enabled denotes the button's enabled state. Visible determines whether the button is visible. Pushed allows displaying a button as pushed. Kind sets the button kind to one of the predefined button kinds. To display the caption inside the button or to override default drawing, use the cbkCustom kind.

If the caption is set to '-', the button becomes a separator. It is not displayed as a button, but rather as a small gap between buttons, like the gap between the Close and Maximize buttons.

```
# TCaptionButton  
$ TCaptionButton Type  
κ TCaptionButton;Types
```

See also

[TCaptionButtonKind](#)

[TCaptionButton_Properties_Kind_See_also](#)
§ [TCaptionButton Properties Kind See also](#)

##\$SKK **Pushed Property**

Applies To

[TCaptionButton](#)

Declaration

```
property Pushed: boolean;
```

Description

Determines whether the button is in "pushed" state. A button can be in pushed state even if it isn't being pushed by the user.

```
# TCaptionButton_Properties_Pushed
```

```
§ Pushed Property
```

```
κ Pushed;Properties;
```

##\$SKK **TCaptionButtonKind Type**

Unit

[CapCtrl](#)

Declaration

```
TCaptionButtonKind = (cbkCustom, cbkOk, cbkRollUp, cbkRollDown,  
cbkDocument,cbkFolder, cbkWindow, cbkMail, cbkDownArrow, cbkUpArrow,  
cbkLeftArrow,cbkRightArrow, cbkMore, cbkFlag, cbkCool);
```

Description

TCaptionButtonKind determines a caption button's appearance. If the Kind property of a TCaptionButton is set to cbkCustom, the button's caption will be displayed in it's center. Otherwise, default drawing is supplied for the preefined types.

```
# TCaptionButtonKind  
$ TCaptionButtonKind Type  
κ TCaptionButtonKind;Types
```

##\$SKK **TCaptionGradient Type**

Unit

[CapCtrl](#)

Declaration

```
TCaptionGradient = (cgNone, cgActive, cgAlways);
```

Description

TCaptionGradient has three possible values used to determine the appearance of a form's caption's gradient fill:

cgNone - No gradient fill (default caption bar appearance).

cgActive - display gradient only when form is active.

cgAlways - always display a gradient.

TCaptionGradient

\$ TCaptionGradient Type

κ TCaptionGradient;Types

##\$SKK **TDirection Type**

Unit

[CapCtrl](#)

Declaration

```
TDirection = (dirLeft, dirRight);
```

Description

TDirection defines two directions (left-to-right and right-to-left). It is used by TCaptionControl to determine alignment and direction of displayed objects, and supports only two options, as opposed to the standard TAlignment type.

```
# TDirection  
$ TDirection Type  
κ TDirection;Types
```

##\$SKK **Visible Property**

Applies To

[TCaptionButton](#)

Declaration

```
property Visible: boolean;
```

Description

Determines whether the button is visible. If set to False, the button will not be displayed. Instead, the caption beneath where the button would have been is shown. The other buttons remain in their locations.

```
# TCaptionButton_Properties_Visible  
$ Visible Property  
κ Visible;Properties;
```

##\$KK **GetBtnKindStr Method**

Applies To

[TCaptionButton](#)

Declaration

```
function GetBtnKindStr: String;
```

Description

Returns a string based on the button kind. For example, a button whose Kind property is set to cbkCustom would return the string 'cbkCustom'.

```
# TCaptionButton_Methods_GetBtnKindStr  
$ GetBtnKindStr Method  
κ GetBtnKindStr;Methods;
```


##\$KK **SetBtnKindStr Method**

Applies To

[TCaptionButton](#)

Declaration

```
procedure SetBtnKindStr(KindStr: String);
```

Description

Sets the button kind based on a string and not a TCaptionButtonKind value. For example, calling SetBtnKindStr('cbkCustom') will change the button's Kind property to cbkCustom.

```
# TCaptionButton_Methods_SetBtnKindStr  
$ SetBtnKindStr Method  
κ SetBtnKindStr;Methods;
```

##\$SKK **Draw Method**

Applies To

[TCaptionButton](#)

Declaration

```
function Draw(DC: HDC; Rect: TRect): Boolean; virtual;
```

Description

The Draw method draws the appropriate graphic on the button surface. TCaptionControl calls the Draw method of each button in its Buttons list, unless the OnButtonDraw event handler informed the TCaptionControl that the button is already drawn.

TCaptionControl is responsible for drawing the actual button. It calls Draw to complete the drawing. The DC parameter is the device context to draw on. The Rect parameter specifies the bounds in which to draw the graphic. Any descendant of Draw should draw only within these bounds. If a descendant of draw cannot draw a button, it must call its inherited method.

```
# TCaptionButton_Methods_Draw  
$ Draw Method  
κ Draw;Methods;
```

##\$**See also**

[TCaptionButtonClickEvent](#)

TCaptionControl_Events_OnButtonClick_See_also
§ TCaptionControl Events OnButtonClick See also

##\$**See also**

[TCaptionButtonDrawEvent](#)

TCaptionControl_Events_OnButtonDraw_See_also
§ TCaptionControl Events OnButtonDraw See also

See also

[TDrawCaptionEvent](#)

TCaptionControl_Events_OnDrawCaption_See_also
§ TCaptionControl Events OnDrawCaption See also

##\$SKK **TDrawCaptionEvent Type**

Unit

[CapCtrl](#)

Declaration

```
TDrawCaptionEvent = procedure(Sender: TObject; var CaptionText: String;  
    DC: HDC; Rect: TRect; var Drawn: boolean);
```

Description

TDrawCaptionEvent is the event type for the OnDrawCaption event of TCaptionControl.

The CaptionText specifies the text to draw. Changing it will cause the default event handler to draw text other than the form's caption. The DC parameter specifies the device context to draw on. The Rect parameter specifies the bounds of the caption text. If an event handler draws the caption text, it should set Drawn to True, indicating no further drawing is necessary.

TDrawCaptionEvent

§ TDrawCaptionEvent Type

κ TDrawCaptionEvent;Types;

##\$SKK **TCaptionButtonDrawEvent Type**

Unit

[CapCtrl](#)

Declaration

```
TCaptionButtonDrawEvent = procedure(Sender: TObject; ButtonIndex: Integer;  
    DC: HDC; Rect: TRect; var Drawn: boolean) of object;
```

Description

TCaptionButtonDrawEvent is the event type of OnButtonDraw. It is called for each button, allowing an application-defined event handler to draw the button. ButtonIndex is the index of the button to be drawn. If the event handler draws the button, it should set the Drawn variable to True. The drawing should occur on the device context supplied by DC in the bounding rectangle defined by Rect.

```
# TCaptionButtonDrawEvent  
$ TCaptionButtonDrawEvent Type  
κ TCaptinButtonDrawEvent;Types;
```

##\$SKK **TCaptionButtonClickEvent Type**

Unit

[CapCtrl](#)

Declaration

```
TCaptionButtonClickEvent = procedure(Sender: TObject; ButtonIndex: Integer;  
    var Pushed: Boolean) of object;
```

Description

TCaptionButtonClickEvent is the event type of OnButtonClick. ButtonIndex is the index of the clicked button. Pushed is the new pushed state of the button. The event handler can change the pushed state by modifying the Pushed parameter. This is used to prevent displaying the button unpushed and pushed back.

```
# TCaptionButtonClickEvent  
§ TCaptionButtonClickEvent Type  
κ TCaptionButtonClickEvent;Types;
```


##\$KKB ColorBands Property

Applies To

[TCaptionControl](#)

Declaration

```
property ColorBands: Integer;
```

Description

ColorBands determines the number of bands in the gradient bar. Possible values range from 8 to 255, and the default value is 64, which is a reasonable compromise between speed and appearance. On systems with an 8-bit or lower color depth (bits per pixel), there is little difference between the default value and higher order values.

```
# TCaptionControl_Properties_ColorBands  
$ Color Bands Property  
κ ColorBands;Properties;  
B prop_TCaptionControlColorBands
```

See also

[Contact Information](#)

[Distribution and Copyrights](#)

[Known Problems](#)

[Version Information](#)

CapCtrl_See_also

§ CapCtrl See also

Contact Information

TCaptionControl is written by Yorai Aminov. The author can be contacted through the following means:

Internet

yaminov@trendline.co.il

CompuServe

100274,720

Mail

Yorai Aminov

54/b Tshernichovsky St.

Jerusalem 92587

Israel

Telephone

Work: 972-2-6795353

Home: 972-2-6790142

ContactInformaiont

§ Contact Information

Distribution and Copyrights

Distribution

TCaptionControl is written by Yorai Aminov. The component is Freeware – it can be redistributed, modified, and used in any application, commercial or noncommercial, under the following terms:

1. When redistributed in source form, all the files in the archive must be supplied (ReadMe file, source code, compiled code, example application, help source, compiled help and this file).
2. When used in an application, the application must bear a valid copyright notice – either the author's or the following: "portions Copyright © Yorai Aminov, 1996, 1997".
3. There is no charge for the source code, except for nominal handling fees.
4. The "About" dialog box, in module CapAbout, is not modified in any way except for additional items in the version history, and the property editor dialog, in CapEdit, must contain a caption button that displays the "About" dialog.

Copyrights

TCaptionControl is partly based on other publicly available classes and components.

Part of the code is based on TGradientForm, by Brad Stowers (bstowers@pobox.com or 72733,3374 on CompuServe). TGradientForm is a Delphi 2.0 form that adds a gradient caption bar. The TGradientForm is copyrighted by Brad Stowers and is freely available, reusable and redistributable. Most of TCaptionControl's gradient code is based on that form, with some corrections.

Some of the code is also based on TWinRoller, by Sean Hsieh (sean@mail.linkease.com.tw). TWinRoller is a small component that adds a single button to the caption bar that, when clicked, hides all of the form except the caption, and was released to the public domain by its author.

The icon drawing code was contributed by Rolf Frei. Rolf can be contacted via CompuServe at 100603,2154.

The TCaptionControl code itself is copyrighted by Yorai Aminov, 1996, 1997.

Distribution_and_Copyrights

§ Distribution and Copyrights

Version Information

This is version 1.0 of TCaptionControl. TCaptionControl had two public beta versions. The following is a version history of the component:

0.10 (12/02/96)

First public beta. Supported graded caption bar, caption buttons. Demonstrated property editing and storing.

0.20 (12/10/96)

Second public beta. Small appearance fixes, added features. Supplied workaround for Windows 95 bug when updating caption.

1.00 (03/21/97)

Release version. Added ColorBands property, changed icon display code, disallow multiple components on single form, displays gradient in design mode.

VersionInformation

§ Version Information

Known Problems

Although most problems have been worked out during beta testing, some problems remain. This topic details all known problems and workarounds, either applied in TCaptionControl's code or to be applied by users of the component.

Caption button size

At least in some versions of Windows 95, calling GetSystemMetrics(SM_CXSMSIZE) and GetSystemMetrics(SM_CYSMSIZE) returns 15, which is the default caption button size, no matter what the real size is.

Workaround

TCaptionControl gets around this by querying the system for the system menu icon size (SM_CXSMICON), which is apparently the caption button size + 1.

Caption bar dragging

When clicking on a separator (a button with caption '-' appears as a small gap between buttons) and dragging, the window is moved as if a normal part of the caption bar is dragged. This is not the Windows 95 default behavior – it does not occur when trying to drag the gap between the "Close" and "Maximize" buttons.

Workaround

No workaround is available at this time.

Unnotified caption repaint

When processing the WM_SETTEXT message (e.g. when the form's Caption property is modified), Windows 95 (and possibly NT 4.0) repaints the caption without notifying the window. TCaptionControl's window procedure does not receive the WM_SETTEXT messages.

Workaround

Supply a message handling procedure to handle the WM_SETTEXT message which will call TCaptionControl.Refresh. If the caption changes rarely and only on specific occasions, the Refresh method could be called explicitly after the caption change.

Disabled "Close" button

TCaptionControl does not recognize window states where the "Close" button is disabled, as this feature is not supported directly by Delphi. This might change in future versions.

Workaround

No workaround is available at this time. If this is a necessary feature, it could be incorporated easily into TCaptionControl's system buttons drawing code.

Windows NT support

TCaptionControl has been tested thoroughly under Windows 95. Several tests were run under Windows NT Server, version 4.0. No problems were encountered, however a single user reported problems displaying the caption or caption text. No additional reports of such problems were received by the author.

Additional problems and bug reports should be made known to the author by the means specified in the topic "Contact Information". Any suggestions, bug fixes and additional code are happily welcomed.

KnownProblems

§ Known Problems

