Table of Contents

1 TCaptionButton - Overview

Overview

The CaptionButton component applies an additional custom button to the title bars of your forms + special menu item associated with this caption button to the system menu.

Additional caption buttons is completely customizable and stand in close integration with the SystemMenu and, as any standard button on the title bar have built-in tooltips (Hint property). However, unlike the standard buttons which built-in to any window, CaptionButton have a lot of incredible additional features! See description and screenshoots below.

Flexible Customization

The order of button on the titlebar specified by BtnOrder property, but you can also separate buttons by any width of blank space specified in SeparatorWidth property.

The image on caption button can be either Sign (you may specify the SignFont) or Glyph bitmap image. You can also specify the glyph image for button in pressed state (GlyphDown property).

If you would like to allow user press the caption button - make AllowPress property True. If you would like to assign to the caption button any popup menu - welcome!, the required menu item will be available also in system menu at run-time.

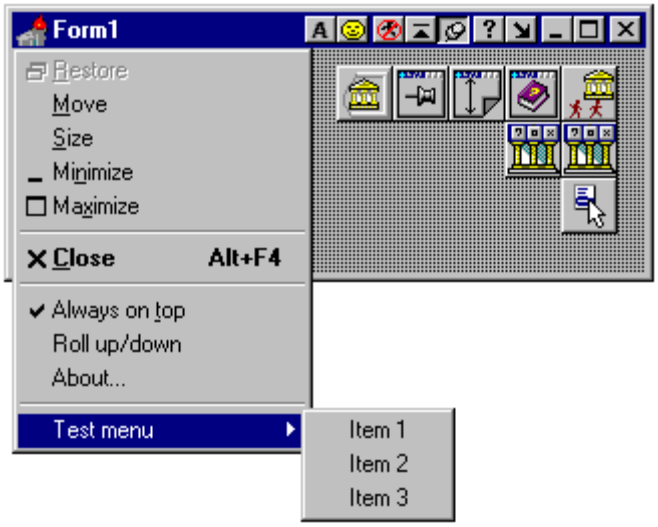
Any button can be in enabled (usual state) or disabled (user could not press it) state - see Enabled property, or invisible on the title bar at all - see Visible property.

All these and much more cool features could be specified at design-time and does not require ANY line of additional code!

See also CaptionButton's events for handling them.

Screenshoot 1 *(tooltips)*:

Screenshoot 2 *(integration with system menu)*:



Screenshoot 3 *(buttons with popup menu)*:

2 Installation Instructions

without source code

to Delphi 2

1. Unzip files from "Delphi2" directory to your "Delphi 2\Lib" directory.
2. Start Delphi 2 IDE.
3. Select "Component \ Install..." menu item.
4. Press "Add" button and select "CaptionButton.dcu" file.
5. Rebuild library.

to Delphi 3

1. Unzip files from "Delphi3" directory and copy them to "Delphi 3\Lib".
2. Start Delphi 3 IDE.
3. Open "CaptionButtonD3.dpk" file.
4. Install package to the components palette ("Install" button).

to Delphi 4

1. Unzip files from "Delphi4" directory and copy them to "Delphi 4\Lib".
2. Start Delphi 4 IDE.
3. Open "CaptionButtonD4.dpk" file.
4. Install package to the components palette ("Install" button).

to Delphi 5

1. Unzip files from "Delphi5" directory and copy them to "Delphi 5\Lib".
2. Start Delphi 5 IDE.
3. Open "CaptionButtonD5.dpk" file.
4. Install package to the components palette ("Install" button).

to Delphi 6

1. Unzip files from "Delphi6" directory and copy them to "Delphi 6\Lib".
2. Start Delphi 6 IDE.
3. Open "CaptionButtonD6.dpk" file.
4. Install package to the components palette ("Install" button).

to C++ Builder 1

1. Unzip files from "BCB1" directory to your "CBuilder\Lib" directory.
2. Start C++ Builder IDE.
3. Select "Component \ Install..." menu item.
4. Press "Add" button and select "CaptionButton.dcu" file.
5. Rebuild library.

to C++ Builder 3

1. Unzip files from "BCB3" directory and copy them to "CBuilder3\Lib".
2. Start C++ Builder 3 IDE.
3. Open "CaptionButtonCB3.bpk" file.
6. Select "Project \ Make CaptionButtonCB3" menu item.
7. Select "Component \ InstallPackages" menu item.
8. Press "Add" button and select "CaptionButtonCB3.bpl" file.

to C++ Builder 4

1. Unzip files from "BCB4" directory and copy them to "CBuilder4\Lib".
2. Start C++ Builder 4 IDE.
3. Open "CaptionButtonCB4.bpk" file.
4. Install package to the components palette ("Install" button).

to C++ Builder 5

1. Unzip files from "BCB5" directory and copy them to "CBuilder5\Lib".
2. Start C++ Builder 5 IDE.
3. Open "CaptionButtonCB5.bpk" file.
4. Install package to the components palette ("Install" button).

Source code

1. Uninstall / delete all previous (trial) instances of CaptionButton.
2. Unzip files from "Sources" directory and copy them to "..\Lib" directory.
3. Run Delphi or ++ Builder IDE.
4. Select "Component \ Install..." menu item.
5. Press "Add" button and select "CaptionButton.pas" file.
6. Rebuild library.

3 Registration Information

CaptionButton component is SHAREWARE. This means that you can try it out for free, but if you like it and want to use it you have to register it with the author. Before continue read and accept license agreement please.

The only difference between the unregistered and registered versions is that the registered one has not message box with remind to register and works without Delphi (C++ Builder) running. You can also purchase the source code, if you would like to have it, and be able to compile or modify the CaptionButton on any 32bit version of Delphi or C++ Builder.

If you would like to use the CaptionButton and receive full, unrestricted version, priority support or even source code — you have to purchase proper license.

All prices in US dollars. Registering entitles you to unlimited support via E-Mail, minor version updates indefinitely and major version updates for 6 month from date of purchase.

Registration types:

Full, unrestricted version without source code:

Single user license:

- **https://secure.element5.com/register.html?productid=140741 - $17,95**

Site license:

- https://secure.element5.com/register.html?productid=140742 **- $99,95**

Full version including 100% Source Code:

Single user license:

- **https://secure.element5.com/register.html?productid=140743 - $37,95**

Site license:

- https://secure.element5.com/register.html?productid=140744 **- $179,95**

Comments

1. ***Site license*** covers a single organisation in one location (building complex). If you buy a site license, you may use the software in unlimited number of your company's computers withing this area. Site license is very cost-effective if you have many computers (many software developers).

See license agreement for more details.

4 License Agreement

Copyright

The FormHelp component (software) is Copyright © 1998-2001, by Utilmind Solutions® (Utilmind). All rights reserved.
The authors - Utilmind Solutions® and Aleksey Kuznetsov (founder of Utilmind), exclusively own all copyrights to the Advanced Application Controls (AppControls) and all other products distributed by Utilmind Solutions®.

Liability disclaimer

THIS SOFTWARE IS DISTRIBUTED "AS IS" AND WITHOUT WARRANTIES AS TO PERFORMANCE OF MERCHANTABILITY OR ANY OTHER WARRANTIES WHETHER EXPRESSED OR IMPLIED. YOU USE IT AT YOUR OWN RISK. THE AUTHOR WILL NOT BE LIABLE FOR DATA LOSS, DAMAGES, LOSS OF PROFITS OR ANY OTHER KIND OF LOSS WHILE USING OR MISUSING THIS SOFTWARE.

Restrictions

You may not attempt to reverse compile, modify, translate or disassemble the software in whole or in part. You may not remove or modify any copyright notice or the method by which it may be invoked.

Operating license

Unregistered version

You may distribute the unregistered version of software freely, provided that all files are included and remain unmodified and that no extra files have been added to the package. You may not ask any money for the distribution. You may use the unregistered version of software free of charge for testing purposes, but if you want to use it for other purposes than testing - you have to register it with the author.

Registered version (single user license)

Once you have registered, you will receive a personal registered copy via email and login information to access your personal area at AppControls.com. This copy may not be copied or lend. You have the non-exclusive right to use registered version of the software only by a single person, on a single computer at a time. You may physically transfer the software from one computer to another, provided that the software is used only by a single person, on a single computer at a time. In group projects where multiple persons will use the software, you must purchase an individual license for each member of the group or purchase site license. Use over a "local area network" (within the same locale) is permitted provided that the software is used only by a single person, on a single computer at a time. Use over a "wide area network" (outside the same locale) is strictly prohibited under any and all circumstances.

Registered version (site/team license)

Once you have registered, you will receive a personal registered copy via email and login information to access your personal area at AppControls.com. This copy may not be copied or lend. You have the non-exclusive right to use and transfer registered version of software on any number of computers by your company or your team only in one location (building complex). If you purchase a site license, you may use the program in an unlimited number of your company's computers within this area.

Registered version (Educational site license)

Once you have registered, you will receive a personal registered copy via email and login information to access your personal area at AppControls.com. This copy may not be copied or lend.

You have the non-exclusive right to use and transfer registered version of software on any number of computers by your educational organisation (school/college/university etc) in one location (building complex). If you buy a educational site license, you may use the program in an unlimited number of your edicational organisation's computers within this area.

Registered version (World-wide license)
Once you have registered, you will receive a personal registered copy via email and login information to access your personal area at AppControls.com. This copy may not be copied or lend. You have the non-exclusive right to use and transfer registered version of software on any number of computers by your company or your team world-wide. If your company has many branches even with thouthands of computers, world wide license covers them all.

Notes (clarification)
"Single-user license" means "single-developer license". "Site license" means that it can be used by any number of software developers within your company.
You can use purchased components in ANY number of your projects and deploy the "end-user" software to ANY number of your users/customers without any additional royalty fees. However you are not permitted to distribute the component itself (the source code or .dcu files of components).

Back-up and transfer
You may make one copy of the software solely for "back-up" purposes, as prescribed by international copyright laws. You must reproduce and include the copyright notice on the back-up copy.

Terms
This license is effective until terminated. You may terminate it by destroying the program, the documentation and copies thereof. This license will also terminate if you fail to comply with any terms or conditions of this agreement. You agree upon such termination to destroy all copies of the program and of the documentation, or return them to author.

Other rights and restrictions
All other rights and restrictions not specifically granted in this license are reserved by authors.

5 Properties

5.1 AllowPress

Applies to
CaptionButton component.

Declaration
```
property AllowPress: Boolean;
```

Description
The AllowPress property controls whether CaptionButton could be pressed (if button pressed, Down property will be True).

If AllowPress is True, the button will have two states - pressed and unpressed (up and down state), which can be controlled by Down property. Also two additional events will occurs - OnPressed on button pressing and OnReleased on releasing.

See also
Down property and OnClick, OnPressed, OnReleased events.

5.2 BtnOrder

Applies to
CaptionButton component.

Declaration
```
type
 TBtnOrder = -1..32767;

property BtnOrder: TBtnOrder;
```

Description
The BtnOrder property indicates position of caption button, the order in which buttons will be displayed. BtnOrder calculates from right side of title bar to left side (see screenshoot below).

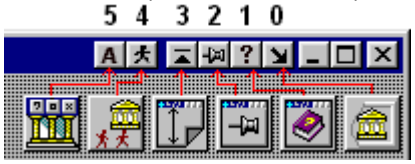
Initially, the button order is always the order in which the components were added to the form, but you can change this by changing the BtnOrder property. The value of the BtnOrder property is unique for each ancestor of the CaptionButton on the form (like TabOrder property of ancestors of TWinControl). The first caption button added to the form has a BtnOrder value of 0, the second is 1, the third is 2, and so on. These values determine where a control is on the title bar.

If you attempt to give a button a BtnOrder value greater than the number of caption buttons on the form minus one (because numbering starts with 0), AppControls won't accept the new value, but will enter the value that assures the component will be the last in the button order.

The caption button with the BtnOrder value of 0 is the custom button which placed on right side of title bar, after regular system buttons.

To separate the buttons by additional empty space — use SeparatorWidth property. For example screenshoot below shows that acAppAutoRun component from the AppControls pack have either BtnOrder and SeparatorWidth = 4 pixels.

Screenshoot (orders of buttons)



See also
SeparatorWidth property.

5.3 Cursor

Applies to
CaptionButton component.

Declaration
```
property Cursor: TCursor;
```

Description
The Cursor property controls the mouse cursor shape used when the mouse moves over

CaptionButton in released state.

See also
CursorDown property.

5.4 CursorDown

Applies to
CaptionButton component.

Declaration
```
property CursorDown: TCursor;
```

Description
The CursorDown property controls the mouse cursor shape used when the mouse moves over CaptionButton in pressed state (Down is True).

See also
Cursor, Down and AllowPress properties.

5.5 Down

Applies to
CaptionButton component.

Declaration
```
property Down: Boolean;
```

Description
The Down property controls whether the button is actually pressed (if True) or unpressed (False).

However, if CaptionButton does not allow pressing, you will unable to change this property, and Down will always False.

See also
AllowPress property.
OnClick, OnPressed, OnReleased events.

5.6 Enabled

Applies to
CaptionButton component.

Declaration
```
property Enabled: Boolean;
```

Description
The Enabled property controls whether the button on the title bar responds to mouse and keyboard messages. If Enabled is True, button responds normally. If Enabled is False, button become disabled and user can not press that button.

See also
Visible property.

5.7 Glyph

Applies to
CaptionButton component.

Declaration
```
property Glyph: TBitmap; // 12x11 pixels optionally
```

Description
The Glyph property specifies the bitmap that appears on the selected CaptionButton.

If Glyph image is not specified (bitmap is empty), then CaptionButton will show the character, specified in Sign property to show it as button image.

Use the Open dialog box that appears as an editor in the Object Inspector to choose a bitmap file (with a .BMP extension) to use on the button, or specify a bitmap image at run-time.

Note
Usually the bitmap size does not matter, but optionally, please use bitmap images with **12x11** pixels resolution.

See also
GlyphDown property.

5.8 GlyphDown

Applies to
CaptionButton component.

Declaration
```
property GlyphDown: TBitmap; // 12x11 pixels optionally
```

Description
The GlyphDown property specifies the bitmap that appears on the selected CaptionButton in pressed state (when Down property is True or when the user clicks this caption button).

If GlyphDown property is not specified (bitmap is empty), Glyph image will used for both button states (pressed and released). If Glyph image is not specified too, then will show the character, specified in Sign property to show it as button image.

Use the Open dialog box that appears as an editor in the Object Inspector to choose a bitmap file (with a .BMP extension) to use on the button, or specify a bitmap image at run-time.

Note
Usually the bitmap size does not matter, but optionally, please use bitmap images with **12x11** pixels resolution.

See also
AllowPress, Down, Glyph properties.
OnPressed, OnReleased events.

5.9 GlyphTransparent

Applies to
CaptionButton component.

Declaration
```
property GlyphTransparent: Boolean;
```

Description
The GlyphTransparent property controls whether the background of the Glyph images should be transparent (if True). Set GlyphTransparent to False if you would like to show the Glyph with original colors, including the background.

See also
Glyph and GlyphDown properties.

5.10 Hint

Applies to
CaptionButton component.

Declaration
```
property Hint: String;
```

Description
The Hint property is the text string that can appear when mouse pointer moves over current CaptionButton.

If the CaptionButton's ShowHint property is False, the Help Hint won't appear, but the other hints still will.

Screenshoot

See also
ShowHint property.

5.11 PopupMenu

Applies to
CaptionButton component.

Declaration
```
property PopupMenu: TPopupMenu;
```

Description
The PopupMenu property identifies the pop-up menu that appears when the user press the button on the form's title bar. When PopupMenu specified, same menu items will appears in the form's system menu.

If PopupMenu is not specified, button will behave as usual button without popup menu.

Screenshoot *(button with popup menu)*

See also
SystemMenu properties.

5.12 SeparatorWidth

Applies to
CaptionButton component.

Declaration
```
property SeparatorWidth: Byte; // in screen pixels
```

Description
The SeparatorWidth property controls the space between current and previous by order caption button in screen pixels.

See also
BtnOrder property.

5.13 ShowHint

Applies to
CaptionButton component.

Declaration
```
property ShowHint: Boolean;
```

Description
The ShowHint property controls whether CaptionButton can show the Hint string when mouse pointer moves over button. If ShowHint is True, Hint will appears.

See also
Hint property.

5.14 Sign

Applies to
CaptionButton component.

Declaration
```
property Sign: Char;
```

Description
The Sign property contains the character that will be shown as button image. You may choose custom font for this sign (SignFont property) and ever shift the sign accordingly to SignLeft and SignTop values.

Note

If any bitmap image assigned to Glyph property, the CaptionButton will show the Glyph image instead of sign. In this case, values assigned to Sign, SignFont, SignLeft and SignTop properties will not be used.

See also
SignFont, SignLeft, SignTop properties.

5.15 SignFont

Applies to
CaptionButton component.

Declaration
```
property SignFont: TFont;
```

Description
The SignFont property is a font object that controls the attributes of the button's Sign, if you don't using glyph images.

See also
Sign property.

5.16 SignLeft

Applies to
CaptionButton component.

Declaration
```
property SignLeft: Integer;
```

Description
The SignLeft property controls the horizontal shifting for the Sign character, relatively to the button center. For example, SignLeft = **1** means that Sign character will be shifted to the right side of button at one pixel. SignLeft = **-1** means that Sign will be shifted at one pixel to the left side.

See also
SignTop, Sign and SignFont properties.

5.17 SignTop

Applies to
CaptionButton component.

Declaration
```
property SignTop: Integer;
```

Description
The SignTop property controls the vertical shifting for the Sign character, relatively to the button center. For example, SignTop = **1** means that Sign character will be shifted to the right side of button at one pixel. SignTop = **-1** means that Sign will be shifted at one pixel to the left side.

See also
SignLeft, Sign and SignFont properties.

5.18 SystemMenu

Applies to
CaptionButton component.

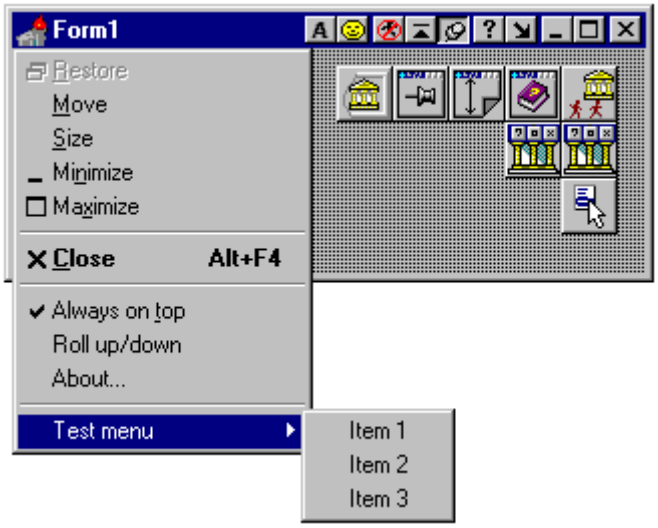
Declaration
```
property SystemMenu: TSystemMenu;
```

Description
The System Menu is the popup menu that appears when you click on the program icon on the title bar. The SystemMenu property is the list of properties that manages the menu item associated with current button on the form's title bar.

Properties

ApplyToMenu	applies or removes the menu item associated with current button from the system menu;
Caption	text for the title of menu item;
Position	position of current menu item in the system menu;
Separators	specifies the separators for menu item, to separatate it from previous and / or next menu item of the system menu.

Screenshoot



5.18.1 ApplyToMenu

Applies to
CaptionButton component.

Declaration
```
property ApplyToMenu: Boolean;
```

Description
The ApplyToMenu property controls whether the CaptionButton currently have the menu item assiciated with this button. Set ApplyToMenu property to True to add according menu item (with text specified by Caption property, in position specified by Position property) to the form's system menu and False to remove.

See also
Caption property.

5.18.2 Caption

Applies to
CaptionButton component.

Declaration
```
property Caption: String;
```

Description
The Caption property specifies the text for menu item in the form's system menu, associated with current caption button on the title bar. If Caption is not specified, text from Hint property will be taken as item title.

See also
Hint property.

5.18.3 Position

Applies to
CaptionButton component.

Declaration
```
property Position: Word;
```

Description
The Position property determinees the position for menu item in the system menu associated with current caption button on the form's title bar. Change the Position value to move current menu item by system menu.

Maximum value for Position property is the current maximum number of items in the system menu (since orders for menu items starts with 0).

See also
Separators property.

5.18.4 Separators

Applies to
CaptionButton component.

Declaration
```
type
  TMenuSeparators = set of (seBefore, seAfter);

property Separators: TMenuSeparators;
```

Description
The Separators property specifies the separators for current menu item to separate it from previous and / or next menu items of the system menu. To set separator before menu item — set `seBefore` to True. To set separator after menu item — set `seAfter` to True.

Specify Separators to separate current menu item in the system menu from another menu items.

See also
Position property.

5.19 Visible

Applies to
CaptionButton component.

Declaration
```
property Visible: Boolean;
```

Description
The Visible property determines whether the caption button currently visible on the title bar. Make Visible property True if you would like to show the button on title bar, or False if you would like to hide the button.


When you changing the Visible property, placements of others button will automacially recalculated and other buttons with higher BtnOrder will be shifted to the right side of the title bar, filling the empty space.

See also
Enabled property.

6 Methods

6.1 RefreshButton

Applies to
CaptionButton component.

Declaration
```
property RefreshButton;
```

Description
The RefreshButton method repaints the custom caption button on the title bar. For example, RefreshButton method automatically executes when caption text changes.

See also
RefreshNCArea method.

6.2 RefreshNCArea

Applies to
CaptionButton component.

Declaration
```
property RefreshNCArea;
```

Description
The RefreshNCArea method repaints all non-client parts of window (i.e. frames, caption buttons etc). RefreshNCArea method automatically executes when buttons need to recalculate their placement after changing of BtnOrder or SeparatorWidth values.

See also

RefreshButton method.

7 Events

7.1 OnClick

Applies to
CaptionButton component.

Declaration
```
property OnClick: TNotifyEvent;
```

Description
The OnClick event occurs when the user clicks the CaptionButton on title bar or selects in the SystemMenu the menu item associated with this button.

See also
OnPressed and OnReleased events.

7.2 OnPressed

Applies to
CaptionButton component.

Declaration
```
property OnPressed: TNotifyEvent;
```

Description
The OnPressed event occurs when the user clicks the the unpressed CaptionButton on title bar or selects in the SystemMenu the unchecked menu item associated with this button.

When user clicks the button which could be pressed (this controlled by AllowPress property), the button state will changed to down, Down property will be True and associated menu item in the system menu will be checked (if SystemMenu.ApplyToMenu is True).

See also
AllowPress, Down, SystemMenu properties.
OnClick and OnReleased events.

7.3 OnReleased

Applies to
CaptionButton component.

Declaration
```
property OnReleased: TNotifyEvent;
```

Description
The OnReleased event occurs when the user clicks the the pressed CaptionButton on title bar or selects in the SystemMenu the checked menu item associated with this button.

When user clicks (unpress) the button which could be pressed (this controlled by AllowPress property), the button state will changed to up, Down property will be False and associated menu item in the system menu will be unchecked (if SystemMenu.ApplyToMenu is True).

See also

AllowPress, Down, SystemMenu properties.
OnClick and OnPressed events.

Index

- I -

- L -

- O -

- R -

- T -