The Perseus VCP (visual component TSlider)

This is a shareware product. This allows you to try the software before you buy it. If after evaluating this control, you decide to continue using it, you are required to register this component by remitting the registration fee of $16 to us.
See How to register for information on registering this component.

You should the read the How to install section first and if you wonder about the name read the About Perseus section!

Description
A really goodlooking range control. With this component you don't have to use those old scrollbars which look like "smashed by a bus" if they don't have the default size! And you can adjust nearly everything: any of the colors, the styles of the different frames (if you use one), the style of the thumb (rect, square, pointer and ownerdrawn!), the size of the thumb, the groove and the scale, the number of ticks of the scale, SnapToScale mode, tick mark connection (inner, outer or none) and much more!

Properties

Frame
FrameSize
FrameStyle

Groove
GrooveColor
GrooveFill
GrooveSize
GrooveStyle

Orientation

Position
PositionMax
PositionMin
PositionStep

Scale
ScaleColor
ScaleConnect
ScaleDistance
ScaleHeight
ScaleOrig
ScalePosition
ScaleTicks
ScaleTicksEnlargeEvery

Thumb
ThumbSize
ThumbStyle
ThumbFill
ThumbColor

FocusAction
FocusColor
DisabledColor

Events

OnSelect
OnTrack
OnDrawThumb

See also:

How to register
About Perseus

Property Frame (TSlider)

Declaration

property Frame: TFrame;

Default: Lowered

Description
Frame specifies the kind of frame (around the whole component).

See also
The TSlider component

TFrame (TSlider)

Declaration

TFrame = (Raised, Lowered, Flat);

See also

Frame
Groove

The TSlider component

Property FrameStyle (TSlider)

Declaration

property FrameStyle: TFaceStyle;

Default: fsWin95

Description
FrameStyle specifies the style of the frame (around the whole component).

See also
The TSlider component

TFaceStyle

Declaration

TFaceStyle = (fsWin95, fsWin31, fsSpeedButton);

Description
Used to define different kinds of frames. **fsWin95** for Windows 95 (i.e. Ctl3D) frames, **fsWin31** for the frames of a standard Windows 3.x button and **fsSpeedButton** for Borlands SpeedButton frame.

See also
FrameStyle
GrooveStyle
ThumbStyle

The TSlider component

Property FrameSize (TSlider)

Declaration

property FrameSize: Byte;

Default: 7

Description
FrameSize is the distance between the ends of the groove and the Slider's frame. This property is normally used to deflate the FocusRect.

See also
The TSlider component

Property Orientation (TSlider)

Declaration

property Orientation: TOrientation;

Default: Horizontal

Description
This property defines the orientation of the slider.

See also
  The TSlider component

TOrientation (TSlider)

Declaration

type

```
  TOrientation = (Horizontal, Vertical);
```

See also

Orientation

The TSlider component

Property Groove (TSlider)

Declaration

property Groove: TFrame;

Default: Raised

Description
Groove specifies the style of the groove. This property is used for turning off the groove too (with setting Groove to None), but keep in mind that GrooveSize is then still used for the distance of the scale.

See also
The TSlider component

Property GrooveStyle (TSlider)

Declaration

property GrooveStyle: TFaceStyle;

Default: fsWin95

Description
GrooveStyle is used to specify the style of the groove's frame.

See also
The TSlider component

Property GrooveSize (TSlider)

Declaration

property GrooveSize: Byte;

Default: 6

Description
GrooveSize spezifies the size of groove. Note that the position of the scale depends not only on ScaleDistance, but on GrooveSize too.

See also
The TSlider component

Property GrooveFill (TSlider)

Declaration

property GrooveFill: Boolean;

Default: True

Description
GrooveFill defines if the groove should be filled with GrooveColor (True) or not (False).

See also
The TSlider component

Property GrooveColor (TSlider)

Declaration

property GrooveColor: TColor;

Default: clBtnFace

Description
GrooveColor is the color of the groove if GrooveFill is set to True.

See also
The TSlider component

Property Thumb (TSlider)

Declaration

property Thumb: TThumbStyle;

Default: PointerDown

Description
Thumb defines which kind of Thumb the Slider should use.

See also
The TSlider component

TThumbStyle (TSlider)

Declaration

TThumbStyle = (PointerUp, PointerDown, Square, Rect, OwnerDraw);

Description
OwnerDraw is used for a non-default outfit. Then the event OnOwnerDraw will be sent.

See also
Thumb

The TSlider component

Property ThumbSize (TSlider)

Declaration

property ThumbSize: Byte;

Default: 30

Description

ThumbSize specifies:

- the height of the thumb if Orientation is horizontal and Thumb is PointerUp, PointerDown or Rect. The other size-property depends on the kind of the used pointer and is set internally.
- the width of thumb if Orientation is vertical and Thumb is PointerUp, PointerDown or Rect. The other size-property depends on the kind of the used pointer and is set internally.
- height and width if and Thumb is Square or OwnerDraw.

See also

The TSlider component

Property ThumbStyle (TSlider)

Declaration

property ThumbStyle: TFaceStyle;

Default: fsWin95

Description
ThumbStyle defines the style of the thumb's frame.

See also
    The TSlider component

Property ThumbFill (TSlider)

Declaration

property ThumbFill: Boolean;

Default: True

Description
ThumbFill determines if the groove is filled with ThumbColor (True) or not (False).

See also
The TSlider component

Property ThumbColor (TSlider)

Declaration

property ThumbColor: TColor;

Default: clBtnFace

Description

ThumbColor is the color of the thumb if ThumbFill is set to True.

See also

The TSlider component

Property PositionMax (TSlider)

Declaration

property PositionMax: Integer;

Default: 100

Description

PositionMax specifies the maximum value of Position. Of course PositionMax has to be bigger than PositionMin.

See also

The TSlider component

Property PositionMin (TSlider)

Declaration

property PositionMin: Integer;

Default: 0

Description
PositionMin specifies the minimum value of Position. Of course Min has to be smaller than PositionMax.

See also
    The TSlider component

Property Position (TSlider)

Declaration

property Position: Integer;

Default: 0

Description
Position is the currently selected value. Position is never smaller than PositionMin or bigger then PositionMax. If you try to set a value out of this range, the maximum or minimum will be set. If the value doesnt fit into PositionStep (i.e. Position:=n*PositionStep) it will be rounded.

See also
    The TSlider component

Property PositionStep (TSlider)

Declaration

property PositionStep: Integer;

Default: 10

Description
Position step is used to make sure that the user can only select special kinds of values. An example: If you set PositionStep to 4, only values which can be divided by 4 can be selected (0,4,8,12...). Width this property you can adjust the SnapToScale modeand much more!

See also
The TSlider component

Property Scale (TSlider)

Declaration

property Scale: Boolean;

Default: True

Description
The scale of the Slider will be only displayed if this property is TRUE.

See also
The TSlider component

Property ScaleTicks (TSlider)

Declaration

property ScaleTicks: Byte;

Default: 10

Description
ScaleTicks is the number of ticks of the scale if Scale is TRUE. One scale tick will be displayed in addition, because PositionMin has to have a tick (e. g. PositionMax=100; PositionMin=0; ScaleTicks=10 means that a tick will be at 0, 10, 20,... 100).

See also
The TSlider component

Property ScaleTicksEnlargeEvery (TSlider)

Declaration

property ScaleEnlargeEvery: Byte;

Default: 5

Description
ScaleEnlargeEvery defines how many ScaleTicks it takes from an enlarged tick to the next enlarged. The height of an enlarged tick is ScaleHeight. For the others the height is ScaleHeight div 2.

See also
The TSlider component

Property ScaleHeight (TSlider)

Declaration

property ScaleHeight: Byte;

Default: 10

Description
With ScaleHeight you can define the height of the scale. For having ticks with different heights, see the ScaleTicksEnlargeEvery property

See also
The TSlider component

Property ScaleConnect (TSlider)

Declaration

property ScaleConnect: Boolean;

Default: True

Description
Only if ScaleConnect is True the ticks of the scale are connected with each other by a line.
Not that ScaleOrig specifies on which side the connection will be drawn.

See also
 The TSlider component

Property ScaleColor (TSlider)

Declaration

property ScaleColor: TColor;

Default: clBlack

Description
ScaleColor defines the color of the scale.

See also
  The TSlider component

Property ScaleOrig (TSlider)

Declaration

property ScaleOrig: TScaleOrig;

Default: Outer

Description
This property defines the origin of the scale. The effect of this property is only visible if ScaleEnlargeEvery isn't 0 or 1 and ScaleConnect is True

See also
The TSlider component

TScaleOrig (TSlider)

Declaration

type

```
  TScaleOrig = (Inner, Outer);
```

See also

ScaleOrig

The TSlider component

Property ScalePosition (TSlider)

Declaration

property ScalePosition: TScalePosition;

Default: BelowOrRight

Description
With ScalePosition you can specify on which side of the groove the scale should be displayed (if Scale is TRUE).

See also
The TSlider component

TScalePosition (TSlider)

Declaration

type
  TScalePosition = (AboveOrLeft, BelowOrRight, Both);

See also
  ScalePosition

  The TSlider component

Property ScaleDistance (TSlider)

Declaration

property ScaleDistance: Byte;

Default: 13

Description
ScaleDistance determines the distance between the scale and the groove.

See also
The TSlider component

Property FocusAction (TSlider)

Declaration

property FocusAction: TFocusAction;

Default: faFocusRect

Description
With FocusAction you can control the behaviour of the slider if it is focused:

faNone	the outfit won't change
faFocusRect	standard pointed rectangle
faFocusColor	if focused the thumb's color changes to FocusColor
faBoth	faFocusColor and faFocusRect combined

See also
TFocusAction
The TSlider component

TFocusAction (TSlider)

Declaration

type
TFocusAction = (faNone, faFocusRect, faFocusColor, faBoth);

Description

faNone	the outfit won't change
faFocusRect	standard pointed rectangle
faFocusColo r	if focused the thumb's color changes to FocusColor
faBoth	faFocusColor and faFocusRect combined

See also:

FocusAction

The TSlider component

Property FocusColor (TSlider)

Declaration

property FocusColor: TColor;

Default: clActiveCaption

Description

FocusColor defines the color the thumb will get if the slider has got the focus and ThumbFill is TRUE. Of course therefore the value of FocusAction has to be faFocusColor or faBoth.

See also

The TSlider component

Property DisabledColor (TSlider)

Declaration

property DisabledColor: TColor;

Default: clGrayText

Description

DisabledColor is the color of the thumb if the slider is disabled.

See also

The TSlider component

Event OnTrack (TSlider)

Declaration

property OnTrack: TNotifyEvent;

Description
The OnTrack event is send while the user changes the Position (e.g. while dragging the pointer with the mouse).

See also
The TSlider component

Event OnSelect (TSlider)

Declaration

property OnSelect: TNotifyEvent;

Description
The OnSelect event is send when the user has finished changing the Position (e.g. when he has stopped dragging).

See also
The TSlider component

Event OnOwnerDraw (TSlider)

Declaration

property OnOwnerDraw: TThumbDrawEvent;

Description
The OnOwnerDraw event is send when the thumb has to be painted but only if ThumbStyle is set to OwnerDraw.

See also
The TSlider component

TThumbDrawEvent (TSlider)

Declaration

```
type
  TThumbDrawEvent = procedure (Sender: TObject; Canvas: TCanvas) of object;
```

See also

TThumbDrawEvent

The TSlider component

About Perseus

Why Perseus and who was Perseus?

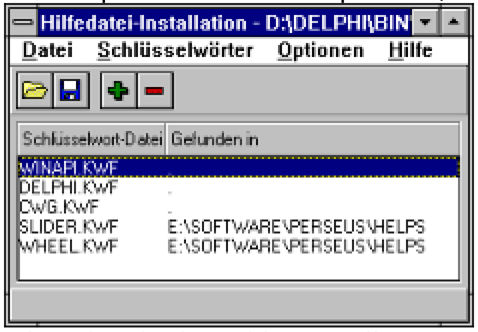
We've choosen this name because Delphi with its name started to introduce all programmers into the Greek Mythology and other (esp. Delphi-)products with mystic names followed and will follow.
But we want to inform you about the man whose name we have borrowed:

In Greek mythology, Perseus {pur'-see-uhs}, the son of Zeus and Danae, was the hero of many exciting adventures. Danae's father, King Acrisius, fearing the fulfillment of an oracle that he would die at the hands of his grandson, tried to dispose of mother and child by enclosing them in a chest and throwing them into the sea. The chest floated to safety on the island of Seriphus, where Perseus grew to manhood. Its ruler, King Polydectes, fell in love with Danae and schemed to get rid of Perseus. He set Perseus the almost impossible task of killing the Gorgon Medusa. With the aid of a cap that made him invisible, Perseus succeeded in slaying the monster (see picture).
In Ethiopia, before reaching home, he encountered Andromeda, whom he saved from a fearful sea monster and then took as a bride. So Perseus won the fights against two monsters (and one could say that this could be the reason for choosing this name <grin>).
After returning to Seriphus, he rescued his mother from Polydectes by turning him to stone with Medusa's head and then gave the head to the goddess Athena. Later, while participating in athletic games, Perseus threw a discus far beyond its target, hitting and killing Acrisius, who was a spectator. Thus the prophecy was fulfilled. As king of Tiryns, Perseus founded the city of Mycenae, and through his son Perses became the legendary founder of the Persian empire. After his death, according to a later version of the myth, Zeus immortalized Perseus by placing him and Andromeda in the sky as constellations.

How to install (TSlider)

Before you can use the TSlider component you have to install the component into Delphi (of course). Therefore choose the item "Install components" in the "options" menu. Then press "Add" and enter the directory of **SLIDER.DCU** or choose "Browse". After finishing with pressing "OK" Delphi will compile Slider into the component-library.

For integrating this helpfile totally into the Delphi-IDE you have to put all the files in the ZIP file in one directory. Then run HelpInst in the Delphi group of the ProgramManager or HELPINST.EXE from your \delphi\bin directory. Open the file DELPHI.HDX (directory \delphi\bin). Add **SLIDER.KWF** via the speedbar or the menuitem "Add keywordfile" from the "Keyword" menu. Save the new delphi.hdx (this will take some seconds because HelpInst "recompiles" the new delphi.hdx).



Schlüsselwort-Datei	Gefunden in
WINAPI.KWF	.
DELPHI.KWF	.
CWG.KWF	.
SLIDER.KWF	E:\SOFTWARE\PERSEUS\HELPS
WHEEL.KWF	E:\SOFTWARE\PERSEUS\HELPS

After that it should be possible to call this help from the ObjectInspector, from a form in designmode and from the GlobalSearch dialog.
If this still doesn't work, copy this helpfile into the delphi\bin directory or add a item for this file in the winhelp.ini!

How to register the TSlider component

The Perseus Visual Component Pack was created by:

Sebastian Modersohn Softwaredevelopment
Heinestrasse 20
D 15370 Petershagen, Germany
Phone: +49 33439 7462
CompuServe ID: 100340,1474

This is a shareware product. This allows you to try the software before you buy it. If after evaluating this component, you decide to continue using it, you are required to register this component by remitting the registration fee of $16 to us.

In general there are 2 possibilities to register the components you want:

Registering via CompuServes SWREG service:
If you want to register via CompuServe "GO SWREG" inside CompuServe and register this component (TSlider) with the database ID **6062**. (for the SWREG IDs of the other components see Other components)
After CompuServe has send us your User ID and your address, we will send the registered version via E-Mail as quickly as possible to you (normally after one or two days).
If you order at least 2 copies of one component you'll also get the full source code.

Registering via postal mail and cheque:
Now it is also possible to register via mail & cheque. Therefore you have to send in an non-formal order to our mailing address (see above) where you specify which components you want to register and a cheque drawn on a German bank (i.e. the banks address on the cheque is in Germany) or an Eurocheque for the sum of the registration fees in DM **plus 14,- DM** for the additional costs (charges for the cheque, mailing, diskette, receipt). Because of the additional cost we suggest that you first check all components and then register all components you want with one order.
Sorry but we cant accept cheques in other currencies or from foreign banks!
After we have checked everything, we will send you a diskette with the registered versions, a printed receipt and some goodies! This registration method will take at least one week.
If you order at least 2 copies of one component you'll also get the full source code.

See also:

Other components
Other procucts

Other components

Also available from

Sebastian Modersohn Softwaredevelopment
Heinestrasse 20
D 15370 Petershagen, Germany
Phone: +49 33439 7462
CompuServe ID: 100340,1474

are the following components as part of the Perseus VCP:

TVListBox

Did you ever try to add more than 10,000 items to a listbox? Do you remember how long it took until the list was displayed the first time correctly? Now you don't have to stress your customers with such listboxes! Here's the real virtual listbox which can have more than 2,000,000,000 items and displaying doesn't take any longer than displaying 10 items even if the list contains 2Giga items. Unlike other "virtual" listboxes this one is not a descendant of TListBox: it was written totally new! So you don't have to care about how to save the information which items are currently selected or any other stuff, all this is done automatically!
TVListBox uses event handlers to supply the items, its heights and protection states, so the data isnt stored twice in memory like TListBox does!
Of course our listbox provides the features known from the standard listbox, like multiple selections, always visible scrollbars, ownerdrawing, non-integral height, "searching" items with keys (CharToItem) and Drag & Drop.
But we have added some very interesting features: multiple rows with integrated and fully configurable header (much better than the VCL-THeader!) and of course horizontal scrolling (the header scrolls too)!
Registration details:
Not yet finalized. But the TVListBox is in final beta state, so give us one or two more weeks!

TSlider

A really goodlooking range control. With this component you don't have to use those old scrollbars which look like "smashed by a bus" if they don't have the default size! And you can adjust nearly everything: any of the colors, the styles of the different frames (if you use one), the style of the thumb (rect, square, pointer and ownerdrawn!), the size of the thumb, the groove and the scale, the number of ticks of the scale, SnapToScale mode, tick mark connection (inner, outer or none) and much more!
Registration details:
The free trial-run version is available in CompuServe's **DELPHI** forum, library **3rd Party**

Products[20], file **SLIDER.ZIP** and also in the **WINSHARE** forum, library **Programming / Dev Tools[20]**. You can register this component for $16 via the SWREG service, database ID **6062**.

TWheel

Another range control, better known as Volume control. Just turn the wheel to select a value. Supports mouse and keyboard input, configurable scale (with SnapToScale option!) and so on! The colors and frames are configurable too (of course!).
Registration details:
The free trial-run version is available in CompuServe's **DELPHI** forum, library **3rd Party Products[20]**, file **PSWHEEL.ZIP** and also in the **WINSHARE** forum, library **Programming / Dev Tools[20]**. You can register this component for $13 via the SWREG service, database ID **6063**.

TPSLabel

The first Label component for Borlands Delphi which can display block justified text! Also it is possible to display rotated Text, if the font is a True Type! Of course this Label has all possibilities of the standard Label, such as optional WordWrap and optional accelerator display (property ShowAccelChar).
Registration details:
The free trial-run version is available in CompuServe's **DELPHI** forum, library **3rd Party Products[20]**, file **PSLABEL.ZIP** and also in the **WINSHARE** forum, library **Programming / Dev Tools[20]**. You can register this component for $10 via the SWREG service, database ID **6086**.

TPSMaskEdit

A language-independent TMaskEdit descendant (uses the national masks for date, time and currency which differ from nation to nation), auto-fill and auto-extend options, comfortable mask editor and all the inherited TMaskEdit-features.
Registration details:
Not yet finalized.

See also:

How to register
Other products

Other products

Also available from

Sebastian Modersohn Softwaredevelopment
Heinestrasse 20
D 15370 Petershagen, Germany
Phone: +49 33439 7462
CompuServe ID: 100340,1474

are the following products:

The SMWCC custom control pack (for Borland Pascal, Borland C++ and other IDEs)
The Sebastian Modersohn's Windows Custom Controls are distributed as Shareware too. Every control can be installed into the Resource Workshop, so every IDE which can use the Resource Workshop can work with these controls (excepting the controls which need an interface, because these interfaces are only available in BP). For a complete list of all controls click here !

The ClipControl (only for Borland Pascal or TPW)
An BP7-tool for adding buttons to the caption bar of **every** kind of window only by adding one or two lines of code and some bitmaps for the outfit.
Features:
- supports Windows, MDI-Windows, MDI-Childs (with buttons in the menu bar if maximized), JanusWindows (from JanusW by Peter Sawatzki (necessary code included), see JANUSW.ZIP in DELPHI for more info), Dialogs and tDlgWindows
- supports Ctl3D-dialogs
- TOTALLY style-guide conform
- supports tips (little yellow text windows, known as hints too)
- demo code for all windows, if registered source code too
The unregistered version is available via CompuServe, forum BPASCAL, library Windows Tools[10], file CLIPCON.ZIP. You can register The Clip Controls for $10 via CompuServes SWREG service. Therefore "GO SWREG" inside the CIS and register The Clip Controls with the database ID **4195**.

See also:
How to register
Delphi components

Other controls of the SMWCC pack

Also available from

Sebastian Modersohn Softwaredevelopment

Heinestrasse 20
D 15370 Petershagen, Germany
Phone: +49 33439 7462
CompuServe ID: 100340,1474

is the SMWCC pack which consists of the following controls:

Name	**Description**
GraphicBtn	A button with optional graphic (bitmap or icon) and with optionally text in all fonts, using all styles, colors etc. Lots of features, good-looking. Makes design easier and language independent! The unregistered version is available via CompuServe, forum BPASCAL/DELPHI, Library 3rd Party Products[20], file GBUTTON.ZIP. You can register The GraphicButton for $15 in the forum SWREG. The database ID of The GraphicButton is **5197**.
Ctl3D Ctl	The Ctl3D control allows you to have the Ctl3D outfit for your dialogs by simply placing it into a dialog (while editing the dialog in Resource Workshop too!) The unregistered version is available via CompuServe, forum BPASCAL or DELPHI, Library 3rd Party Products, file 3DCTL.ZIP. You can register The Ctl3D Control for $10 in the forum SWREG. The database ID of the Ctl3D-control is **4672**.
Font Control	The Font control allows you to set a non-bold font for all controls specified by an ID-range. The unregistered version is available via CompuServe, forum BPASCAL or DELPHI, Library 3rd Party Products, file FNTCTL.ZIP. You can register The Font Control for $10 in the forum SWREG. The database ID of Font is **4673**.
Wheel	This is something like a scrollbar as a circle or a volume control. It has optional customizable scale with optional snap-mode; balance mode and all that in 3D-look! The unregistered version is available via CompuServe, forum BPASCAL or DELPHI, Library 3rd Party Products, file Wheel.ZIP. You can register The Font Control for $10 in the forum SWREG. The database ID of Font is **5025**.
HifiButton	This is a button which looks like a button of a CD-Player or something like that (just have a look!) with optional on/off LED. The unregistered version is available via CompuServe, forum BPASCAL or DELPHI, Library 3rd Party Products, file HIFIBTN.ZIP. You can register The HifiButton for $10 in the forum SWREG. The database ID of the HifiButton is **4675**.

PercentBar	This control is a "simply to use" percent bar. -supports Ctl3D frame -customizable filled color The unregistered version is available via CompuServe, forum BPASCAL or DELPHI, Library 3rd Party Products, file PBAR.ZIP. You can register The Percent Bar control for $10 in the forum SWREG. The database ID of The percent bar is **4677**.
MicroScroll	The MicroScroll control is a mini-scrollbar which is normally used to increase or decrease the value of the corresponding edit control. It can be partially disabled and supports units (for the edit values). The unregistered version is available via CompuServe, forum BPASCAL or DELPHI, Library 3rd Party Products, file MSCROLL.ZIP. You can register the MicroScroll for $10 in the forum SWREG. The database ID of the MicroScroll is **4674**.
Structo &	This is a control for simply drawing everything you want inside a dialog, and scroll your drawing (We used it to display metafiles and to write structural charts in dialogs). The unregistered version is available via CompuServe, forum BPASCAL or DELPHI, Library 3rd Party Products, file MBTN_STR.ZIP. **You can register the control only in connection with the MenuButton.**
MenuButton	This is a button which displays a local, customizable menu, if it was pressed. The unregistered version is available via CompuServe, forum BPASCAL or DELPHI, Library 3rd Party Products, file MBTN_STR.ZIP. You can register the Structo control in connection with the MenuButton for $10 in the forum SWREG. The database ID of this pack is **4676**.
Text control	This is a control for simply placing all kinds of text in a dialog (all fonts, styles, justifications, colors and effects possible!). The unregistered version is available via CompuServe, forum BPASCAL or DELPHI, Library 3rd Party Products, file TEXT.ZIP. You can register the Text control for $10 in the forum SWREG. The database ID of the Text control is **4678**.
Color Control	This is a control to make color-choosing easier than to call an extra common-dialog! -it supports the 16 base-colors -You can choose 1 or 2 (foreground/background) colors -supports keyboard and mouse selection The unregistered version is available via CompuServe, forum BPASCAL or DELPHI, Library 3rd Party Products, file COLOR.ZIP. You can register The Color Control for $10 in the forum SWREG. The database ID of the Color control is **4671**.

We have planned to make BC++ versions of every control, i.e. we will "tranlsate" the import units. We want to try to make version for **both OWL versions**, but we will start with the OWL2 version. If you're interested have a look in the **BCPPWIN** Forum or **just contact us** for further information.

Note that we've changed to Library **3rd Party Products [20],** in the **BPASCAL / DELPHI** Forum. All products from us will also be uploaded to the **WINSHARE** Forum, Library **Programming / Dev Tools[20]**. The newest versions will be available **earlier** in the WINSHARE Forum, because the Librarian of the DELPHI forum obviously needs more time.

See also:

Other procucts
Delphi components
Contents