SMWCC 2.0 (Color control) 

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 the control by remitting the registration fee of $10 to us.
See How to register for information on registering this control.

You should read the How to install section before you "install" the control!

This is a control to make color-choosing easier than to call an extra common-dialog. It supports 16 colors. You can either choose 2 colors (normally foreground and background) with the left and the right mouse-button or only one color. It is also possible to choose the color(s) by keyboard if the control has got the focus: Use the left, right, up and down keys for choosing the foreground color and press shift and the same keys to choose the background color. If the control is set to choose only one color the methods normally used to choose the second color change the first and only color.

Methods:

- wm_ColorChanged
- SetFGColor
- SetBKColor
- GetFGColor
- GetBKColor

See also:

- How to install
- How to compile
- How to register

- Other controls
- Other procucts

- What's new?

wm_ColorChanged (TColor Message)

Syntax:
ID = wParam /* ID of the Color-Control */
lParam = 0L /* unused, always 0L */

Description:
The Color-Control sends this message to its parent for notification of a selection-change.

Parameters:
ID The ID of the Color-Control that has sent this message

See also:
The Color-Control

TColor.SetFGColor (Method)

Syntax:
function SetFGColor(Color: LongInt): LongInt;

Description:
Sets the foreground color of the control to the color specified by the Color param, if it is a base color.

Returns:
This function returns 0 if it was succesful, otherwise it returns -1.

Comments:
Note that you can't use **TColorRef** types with an index of the actual palette! Use the RGB function to create usable colors ($00BBGGRR).

See also:
SetBKColor
GetFGColor
GetBKColor

The Color-Control

TColor.SetBKColor (Method)

Syntax:
function SetBGColor(Color: LongInt): LongInt;

Description:
Sets the background color of the control to the color specified by the Color param, if it is a base color and if the control supports 2 colors.

Returns:
This function returns 0 if it was succesful, otherwise it returns -1.

Comments:
Note that you can't use **TColorRef** types with an index of the actual palette! Use the RGB function to create usable colors ($00BBGGRR).

See also:
SetFGColor
GetFGColor
GetBKColor

The Color-Control

TColor.GetBKColor (Method)

Syntax:
function GetBKColor: LongInt;

Description:
Returns the actual selected background color.

Returns:
If the control doesn't support 2 colors, this function returns -1, otherwise it is the selected color in the $00BBGGRR format.

See also:
SetFGColor
SetBKColor
GetFGColor

The Color-Control

TColor.GetFGColor (Method)

Syntax:
function GetFGColor: LongInt;

Description:
Returns the actual selected foreground color in the $00BBGGRR format.

See also:
SetFGColor
SetBKColor
GetBKColor

The Color-Control

How to register

All SMWCC controls were 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 control, you decide to continue using it, you are required to register the software by remitting the registration fee of $10 to us.

To register "GO SWREG" inside CompuServe and register this control (Color) with the database ID 4671.
After CompuServe has send us your User ID we will send the source code of the control via E-Mail as quickly as possible to you.

See also:

- Other controls
- Other procucts

How to compile

This topic gives important information on How to (re)compile parts of the custom control.

1. **Compiling the DLL:**
   Therefore you only have to load **colors.pas** into the IDE and compile it with Ctrl+F9. You will get a "Cant execute units and DLLs!"-error but with Ctrl+F9 you compile the units which are needed automatically too!

2. **Compiling the redistributable DLL:**
   Just the same as in 1. but define a global symbol **REDIST** (use the Options|Compiler dialog!) and compile **colors.pas** using the **Compiler|Compile project new** feature, to make sure that all used unit are recompiled too!

3. **Compiling the interface unit:**
   If you compile the interface unit **color.pas** with the defined symbol **AUTOLOAD** (use the Options|Compiler dialog!) the interface unit will automatically load and free the DLL. Therefore the DLL has to be in the windows\system directory or in a directory which is specified in the Path environment variable (e.g. windows). If you don't want to store the DLL in these directories you have to load and free the DLL manually or you have to change the code in the interface unit!

4. **Compiling a unit-version of the control:**
   Now it is possible to include the control in your program, that means you don't need to distribute the DLL! Therefore you have to compile the interface unit (color.pas) with the global defined symbol **NoDLL** (use the Options|Compiler dialog!). Make sure that wncolors.pas will be recompiled too! With including "**color**" in the uses clause of your program the control will be automatically available. But for editing the control in RW you have to install the DLL of course.

5. **Compiling a DLL which includes various controls:**
   Therefore the file **smwcc.pas** is included in the zip. Load this file into the IDE and make sure that the compiler will find the wnXXXXXX file of each control (use Options|Directories...). If you want to redistribute the SMWCC.DLL, define a global symbol **REDIST** (use the Options|Compiler dialog!). First you should change the **NoOfCtls** constant to the number of controls you want to include. Then you should delete all bracket pairs *{ }* of the code parts which begin with the name of a control you want to include (e.g. *{Color:} {some code}* to *{Color:}* some code). After that you should be able to compile the DLL (may be you have to add/delete some commas or semicolons).

See also:
The Color-Control

How to install

You can copy the source files of the control in every directory you want. But the DLL (and this helpfile if you want to use the help in the Resource Workshop) has / have to be in a directory which is mentioned in your PATH statement or in your windows\system directory.

To install the custom control for the Resource Workshop chosse the menu item **Install Library...** from the menu **Options** while you are editing a dialog. Then a file dialog occurs there you can select the DLL you want to install.
If you want to remove a custom control or want to copy the DLL in an other directory you'll have to change the **workshop.ini** file in the **WINDOWS** directory (you'll see how and where).
With the RW of BC 4.x it's nearly the same but more comfortable.

To "install" a custom control for your program you should make sure that windows will find the library file. The most simply solution is to copy the DLL-files into the **WINDOWS\SYSTEM** directory and to use the autoloading feature of the interface unit (see How to compile).
If you do not want this, copy the DLL in a directory which is in your **PATH** statement or give the full path string to the LoadLibrary statement in the app's Init constructor, something like that:

```
...
DLL:=LoadLibrary('c:\myprog\dlls\name.dll');
...
```

See the demo code and the SDK help for more information about loading librarys.

See also:

- Other controls
- Other procucts

Other controls of the SMWCC pack

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 / BDELPHI** Forum. All products from us will also be uploaded to the **WINSHARE** Forum, Library **Programming Related**. The newest versions will be available **earlier** in the WINSHARE Forum, because the Librarian of BPASCAL obviously needs more time.

Also available from

Sebastian Modersohn Softwaredevelopment

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

are 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/BDELPHI, 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 BDELPHI, 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 BDELPHI, 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 BDELPHI, 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 BDELPHI, 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 BDELPHI, 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 BDELPHI, 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 BDELPHI, 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 BDELPHI, 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 BDELPHI, 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 BDELPHI, 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**.

See also:
Other procucts

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:

Name	**Description**
Clip Controls	An BP7-tool for adding buttons to the caption bar of **every** kind of window by only 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 BPASCAL 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 in the forum SWREG. The database ID of The Clip Controls is **4195**.

See also:

Other controls

What's new?

You can now compile different versions of this control:

1. a normal DLL
2. a DLL for redistribution
3. a unit version of the control to minimize the number of files of your product
4. an interface unit which automatically loads the DLL it needs

See How to compile for more information on this topic!

Installation is now more simple:

No longer concrete paths and resource strings! This helpfile and the DLL now have to be in a directory that is mentioned in your PATH statement or in you windows\system directory.

See How to install for more information on this topic!