

Lesson 39 — TColorBox

Additional Tab

Project: Lesson37_TColorBox | Units: Unit1.pas, Unit2.pas

1. What Is TColorBox?

TColorBox is a drop-down combo box that shows a list of colours with a coloured rectangle beside each name. The user selects a colour from the list. It is found on the Additional tab of the Component Palette.

Aspect	Detail
Palette tab	Additional
Unit	ColorBox
Base class	TCustomComboBox
Key property	Selected (TColor) — the chosen colour
Key event	OnChange — fires when the user picks a different colour

2. Key Properties

Property	Description
Selected	The currently selected TColor value. Read or write in code.
Style	A set of TColorBoxStyles flags. Controls which colours appear in the list. Default: [cbStandardColors, cbExtendedColors, cbSystemColors].
ColorRectWidth	Width of the coloured rectangle in pixels. Default 14.
ColorRectOffset	Space between the left edge and the colour rectangle. Default 2.
DefaultColorColor	The actual colour displayed for the clDefault entry.
NoneColorColor	The actual colour displayed for the clNone entry.
ColorDialog	An optional TColorDialog used when cbCustomColor is in Style.

3. Style Flags

The Style property is a set of TColorBoxStyles values. Add or remove flags at runtime:

```
cbBox.Style := cbBox.Style + [cbPrettyNames];  
cbBox.Style := cbBox.Style - [cbSystemColors];
```

Flag	Effect
cbStandardColors	16 standard VGA colours (default on)
cbExtendedColors	Additional named colours (default on)
cbSystemColors	System UI colours: clBtnFace, clHighlight etc. (default on)
cbIncludeNone	Adds clNone at the top of the list
cbIncludeDefault	Adds clDefault at the top of the list
cbCustomColor	Adds a "Custom..." entry that opens the ColorDialog when clicked
cbPrettyNames	Shows "Red" instead of "clRed" for nicer display
cbCustomColors	Fires OnGetColors so you can add your own colour entries

4. Key Events

Event	Description
OnChange	Fires when the selected colour changes. Read Selected here.
OnGetColors	Fires when cbCustomColors is in Style. Add custom entries to Items.

```
procedure TForm1.cbBasicChange(Sender: TObject);
begin
    shapePreview.Brush.Color := cbBasic.Selected;
end;

procedure TForm1.cbCustomGetColors(Sender: TCustomColorBox; Items: TStrings);
begin
    Items.Add("Coral=$0080505F");
    Items.Add("Mint=$0090EE90");
end;
```

5. Reading RGB Components

Use RedGreenBlue to split a TColor into its R, G and B byte values:

```
var R, G, B : Byte;
RedGreenBlue(cbBasic.Selected, R, G, B);
Labell1.Caption := "#" + IntToHex(R,2) + IntToHex(G,2) + IntToHex(B,2);
```

6. Panel Layout

Panel	Content
Panel 1 - What is TColorBox?	Explanation memo with key properties and code snippets
Panel 2 - Basic demo	TColorBox with OnChange applying selected colour to a TShape fill and/or border
Panel 3 - Style flags	Seven checkboxes toggling each TColorBoxStyles flag live
Panel 4 - Screen eyedropper	Pick any colour from anywhere on screen (see Section 7)
Panel 5 - Custom colours	TColorBox with cbCustomColors and OnGetColors adding 5 custom named colours
Panel 6 - Properties	ColorRectWidth, ColorRectOffset and cbPrettyNames applied to all boxes
Panel 7 - Event log	Timestamped log of all OnChange and OnGetColors events

7. Screen Eyedropper (Panel 4)

Panel 4 demonstrates two eyedropper approaches:

- Button 1 - "Pick from screen": activates an inline eyedropper. The cursor changes to a crosshair. A TTimer fires every 50ms, reads the pixel under the cursor using GetDC(0) + GetPixel (Windows) or BitBlt (Linux/macOS), and shows a live colour preview. Clicking the left mouse button captures the colour and sets cbBasic.Selected.
- Button 2 - "Open eyedropper window": opens the standalone TEyeDropperForm (Unit2). This form shows a 12x zoom magnifier around the cursor, a colour swatch, hex and RGB values, a colour grid, and a TColorBox showing the captured result.

Windows note: *GetDC(0) + GetPixel works reliably on Windows. On Linux and macOS, GetDC(0) + BitBlt into a 1x1 bitmap is used instead. In practice the cross-platform approach does not work reliably for reading screen pixels outside the form window in LCL. The eyedropper is most useful on Windows.*

8. Standalone Eyedropper Form (Unit2)

Unit2 contains TEyeDropperForm, a self-contained floating window with:

- A 132x132 pixel magnifier (TImage) showing a 11x11 screen area zoomed 12x with a crosshair overlay
- A colour swatch panel showing the colour under the cursor
- Hex (#RRGGBB) and RGB(r,g,b) labels updated live
- Screen coordinates (X, Y) of the cursor
- A TColorBox showing the captured result
- A 8x3 colour grid (TPaintBox) for quick reference

The magnifier is built entirely in code using TBitmap and BitBlt. No image files are needed. The crosshair is drawn with Pen lines in white then black for contrast against any background.

```
{ Key magnifier constants in Unit2 }
ZOOM    = 12;    { each screen pixel becomes a 12x12 square }
RADIUS  = 5;     { capture 11x11 pixels around cursor }
SIZE    = 132;   { output bitmap: 11 * 12 = 132 pixels }
```

9. Cross-Platform Notes: Windows, Linux and macOS

Aspect	Notes
TColorBox component	Works identically on all platforms
OnChange event	Works on all platforms
Style flags	Work on all platforms
Screen pixel reading	GetDC(0)+GetPixel is Windows only. BitBlt approach used for Linux/macOS but may not work reliably outside the form window in LCL.
Eyedropper	Most reliable on Windows. Linux/macOS support is limited by LCL screen access.

10. TScrollBox - Portability Note

All panels are inside a TScrollBox (Align = alClient, AutoScroll = True, BorderStyle = bsNone). This is the standard pattern used throughout this series to ensure the layout works on all screen sizes, resolutions and DPI settings across Windows, Linux and macOS.

11. Project Files

File	Purpose
Unit1.pas	Main form: TColorBox demos, style flags, inline eyedropper, custom colours, properties
Unit1.lfm	Main form layout
Unit2.pas	Standalone eyedropper form (TEyeDropperForm) with magnifier
Unit2.lfm	Eyedropper form layout
Lesson37_TColorBox.lpr	Program entry point
Lesson37_TColorBox.lpi	Lazarus project file

TColorBox Demo — Lazarus Pascal Beginners Series

What is TColorBox?

TColorBox is a drop-down list showing named colours with a coloured rectangle beside each name.

Key properties:

- Selected — the chosen TColor value
- Style — set of flags controlling which colours appear
- ColorRectWidth — width of the colour rectangle (default 14px)
- cbPrettyNames — show "Red" instead of "clRed"


Demo 1: Basic colour selection — select a colour and apply to a shape

Read `cbBasic.Selected` to get the TColor value. Set `cbBasic.Selected := clRed` to select in code.

Sky Blue

Selected: #A6CAF0 RGB(166, 202, 240)

☒ Apply to shape fill
☐ Apply to shape border



Preview shape

Demo 2: Style flags — toggle each flag to see which colours appear

Black

☒ cbStandardColors
☐ cbIncludeNone
☐ cbPrettyNames

☒ cbExtendedColors
☐ cbIncludeDefault

☒ cbSystemColors
☐ cbCustomColor

`cbCustomColor` adds a "Custom..." entry. Clicking it opens `ColorDialog1`. Assign `ColorDialog` to the TColorBox for this to work.

Demo 4: Screen eyedropper — pick any colour from anywhere on screen

Uses `GetDC(0)+GetPixel` from `LCLIntf` — works on Windows, Linux (GTK2/GTK3) and macOS (Cocoa).

Pick from screen

Open standalone eyedropper...

Click "Pick from screen", then click any pixel on screen.

Demo 3: Custom colours via OnGetColors event

`cbCustomColors` in Style triggers `OnGetColors`. Add items as "Name=\$00RRGGBB" to Items in the event handler.

Black

Selected: ---

Properties (applied to all colour boxes)

ColorRectWidth: 14
ColorRectOffset: 3

Apply

☒ cbPrettyNames

Event log:

```

[13:18:35] OnGetColors fired — added 5 custom colours
[13:18:35] cbBasic.Selected = clSkyBlue (#F0CAA6)
[13:18:35] cbPrettyNames = True
[13:18:35] TColorBox lesson ready.
[13:18:35] cbBasic.Selected = clSkyBlue
[13:18:35] cbBasic item count = 20

```

Clear log

What is TColorBox?

TColorBox is a drop-down list showing named colours with a coloured rectangle beside each name.

Key properties:

- Selected — the chosen TColor value
- Style — set of flags controlling which colours appear
- ColorRectWidth — width of the colour rectangle (default 14px)
- cbPrettyNames — show "Red" instead of "clRed"

Demo 1: Basic colour selection — select a colour and apply to a shape

Read `cbBasic.Selected` to get the TColor value. Set `cbBasic.Selected := clRed` to select in code.

Selected: #CCE4B0 RGB(204, 228, 176)

☒ Apply to shape fill ☐ Apply to shape border



Preview shape

Demo 2: Style flags — toggle each flag to see which colours appear

- ☒ cbStandardColors ☐ cbIncludeNone ☐ cbPrettyNames
- ☒ cbExtendedColors ☐ cbIncludeDefault
- ☒ cbSystemColors ☐ cbCustomColor

`cbCustomColor` adds a "Custom..." entry. Clicking it opens `ColorDialog1`. Assign `ColorDialog` to the TColorBox for this to work.

Demo 4: Screen eyedropper — pick any colour from anywhere on screen

Uses `GetDC(0)+GetPixel` from LCLIntf — works on Windows, Linux (GTK2/GTK3) and macOS (Cocoa).

Pick from screen

Open standalone eyedropper...

#CCE4B0 RGB(204, 228, 176)

Colour captured and set in `cbBasic` above!

Demo 3: Custom colours via OnGetColors event

`cbCustomColors` in Style triggers `OnGetColors`. Add items as "Name=\$00

Selected: ---

Properties (applied to all colour boxes)

ColorRectWidth: 14 ColorRectOffset: 3 ☒

Event log:

```
[13:19:28] Eyedropper captured: #E0EEF9 at screen pos (989,682)
[13:19:28] Eyedropper active — click anywhere on screen.
[13:19:32] cbBasic.Selected = $00B0E4CC (#B0E4CC)
[13:19:32] Eyedropper captured: #CCE4B0 at screen pos (1165,688)
[13:19:32] Standalone eyedropper opened.
```

