

How to create components with a transparent background

To do that kind of stuff right, create a new control derived for example from *TGroupbox*, and override its *CreateParams* method like this:

```
protected {in control declaration}

procedure CreateParams(var params: TCreateParams); override;

procedure TTransparentGroupbox.CreateParams(var params: TCreateParams);
begin
    inherited CreateParams(params);
    params.ExStyle := params.ExStyle or WS_EX_TRANSPARENT;
end;

{Add a handler for the WM_ERASEBKGD message}
procedure WMEraseBkGnd(var msg: TWMEraseBkGnd); message WM_ERASEBKGD;

procedure TTransparentGroupbox.WMEraseBkGnd(var msg: TWMEraseBkGnd);
begin
    SetBkMode (msg.DC, TRANSPARENT);
    msg.result := 1;
end;
```

That is the basic frame for a *TWinControl* descendent. For a *TGraphicsControl* you would drop the *CreateParams* (since only WinControls have that method) and override the create constructor. After calling the inherited constructor you modify the *ControlStyle* of the control:

```
ControlStyle := ControlStyle - [csOpaque];
```

Transparency actually works better for *TGraphicControls* than for *TWinControls*. The latter have problems if the control is moved or the background needs to change. Delphi container controls (like form or panel) are always created with the *WS_CLIPCHILDREN* style, which automatically excludes the area under child controls from updates, so the background will not be updated if required. Removing the *WS_CLIPCHILDREN* style from a controls parent is possible with:

```
SetWindowLong(
    parent.handle, GWL_STYLE,
    GetWindowLong(parent.handle, GWL_STYLE) and not WS_CLIPCHILDREN
);
```

But that may lead to excessive flicker on screen updates.

Original resource:	The Delphi Pool
Author:	Unknown
Added:	2009-08-12
Last updated:	2010-03-16

Copyright © Peter Johnson (DelphiDabbler) 2002-2018