

Как придать Вашему меню стиль XP или Ваш собственный стиль.

Этот код моделирует меню с видом **Office XP** и все функциональные особенности, включая свойство **RightToLeft**.

Вы можете изменить код, чтобы придать Вашему меню другой цвет и шрифт.

Обратите внимание: Удостоверьтесь, что свойства **OwnerRedraw** и **ParentBidiMode** установлены в

*True*

```
unit fMenu; interface uses Windows, SysUtils, Classes, Graphics, Controls,
Forms, Menus, ImgList; type TForm1 = class(TForm) ImageList1: TImageList;
MainMenu1: TMainMenu; FileMenu: TMenuItem; procedure DrawItem(Sender: TObject;
ACanvas: TCanvas; ARect: TRect; Selected: Boolean); procedure
FormCreate(Sender: TObject); private procedure MenuDrawItem(xMenu: TMenuItem);
public end; procedure MenuDrawItem(Sender: TObject; ACanvas: TCanvas; ARect:
TRect; Selected: Boolean); var Form1: TForm1; implementation {$R *.DFM}
procedure TForm1.FormCreate(Sender: TObject); begin MenuDrawItem(Menu); end;
procedure TForm1.DrawItem(Sender: TObject; ACanvas: TCanvas; ARect: TRect;
Selected: Boolean); begin MenuDrawItem(Sender, ACanvas, ARect, Selected); end;
procedure TForm1.MenuDrawItem(xMenu: TMenuItem); var i: integer; B: TBitmap;
FMenuItem: TMenuItem; begin B := TBitmap.Create; B.Width := 1; B.Height := 1;
for i := 0 to ComponentCount - 1 do if Components[i] is TMenuItem then begin
FMenuItem := TMenuItem(Components[i]); FMenuItem.OnDrawItem := DrawItem; if
(FMenuItem.ImageIndex = -1) and (FMenuItem.Bitmap.Width = 0) and (xMenu nil) then
if FMenuItem.GetParentComponent.Name = xMenu.Name then
FMenuItem.Bitmap.Assign(b); end; B.Free; DrawMenuBar(handle); end;
procedure MenuDrawItem(Sender: TObject; ACanvas: TCanvas; ARect: TRect;
Selected: Boolean); var txt: string; B: TBitmap; IConRect, TextRect: TRect;
FBackColor, FIconBackColor, FSelectedBkColor, FFontColor, FSelectedFontColor,
FDisabledFontColor, FSeparatorColor, FCheckedColor: TColor; i, X1, X2: integer;
TextFormat: integer; HasImgLstBitmap: boolean; FMenuItem: TMenuItem; FMenu:
TMenuItem; begin FMenuItem := TMenuItem(Sender); FMenu :=
FMenuItem.Parent.GetParentMenu; FBackColor := $00E1E1E1; FIconBackColor :=
$00D1D1D1; FSelectedBkColor := $00DCCFC7; FFontColor := clBlack;
FSelectedFontColor := clNavy; FDisabledFontColor := clGray; FSeparatorColor :=
```

```

$00D1D1D1; FCheckedColor := clGray; if FMenu.IsRightToLeft then begin X1 :=
ARect.Right - 20; X2 := ARect.Right; end else begin X1 := ARect.Left;
X2 := ARect.Left + 20; end; IConRect := Rect(X1, ARect.Top, X2, ARect.Bottom);
TextRect := ARect; txt := '' + FMenuItem.Caption; B := TBitmap.Create;
B.Transparent := True; B.TransparentMode := tmAuto; HasImgLstBitmap := false; if
(FMenuItem.Parent.GetParentMenu.Images nil) or (FMenuItem.Parent.SubMenuImages
nil) then begin if FMenuItem.ImageIndex = -1 then HasImgLstBitmap := true
else HasImgLstBitmap := false; end; if HasImgLstBitmap then begin if
FMenuItem.Parent.SubMenuImages nil then
FMenuItem.Parent.SubMenuImages.GetBitmap(FMenuItem.ImageIndex, B) else
FMenuItem.Parent.GetParentMenu.Images.GetBitmap(FMenuItem.ImageIndex, B)
end else if FMenuItem.Bitmap.Width > 0 then
B.Assign(TBitmap(FMenuItem.Bitmap)); if FMenu.IsRightToLeft then begin X1 :=
ARect.Left; X2 := ARect.Right - 20; end else begin X1 := ARect.Left + 20;
X2 := ARect.Right; end; TextRect := Rect(X1, ARect.Top, X2, ARect.Bottom);
ACanvas.brush.color := FBackColor; ACanvas.FillRect(TextRect); if FMenu is
TMainMenu then for i := 0 to FMenuItem.GetParentMenu.Items.Count - 1 do if
FMenuItem.GetParentMenu.Items[i] = FMenuItem then begin
ACanvas.brush.color := FIconBackColor; ACanvas.FillRect(ARect); if
(FMenuItem.ImageIndex = -1) and (FMenuItem.Bitmap.Width = 0) then begin
TextRect := ARect; break; end; end; ACanvas.brush.color
:= FIconBackColor; ACanvas.FillRect(IConRect); if FMenuItem.Enabled then
ACanvas.Font.Color := FFontColor else ACanvas.Font.Color := FDisabledFontColor;
if Selected then begin ACanvas.brush.Style := bsSolid; ACanvas.brush.color :=
FSelectedBkColor; ACanvas.FillRect(TextRect); ACanvas.Pen.color :=
FSelectedFontColor; ACanvas.Brush.Style := bsClear;
ACanvas.RoundRect(TextRect.Left, TextRect.Top, TextRect.Right,
TextRect.Bottom, 6, 6); if FMenuItem.Enabled then ACanvas.Font.Color :=
FSelectedFontColor; end; X1 := IConRect.Left + 2; if B nil then
ACanvas.Draw(X1, IConRect.Top + 1, B); if FMenuItem.Checked then begin
ACanvas.Pen.color := FCheckedColor; ACanvas.Brush.Style := bsClear;
ACanvas.RoundRect(IConRect.Left, IConRect.Top, IConRect.Right,
IConRect.Bottom, 3, 3); end; if not FMenuItem.IsLine then begin
SetBkMode(ACanvas.Handle, TRANSPARENT); ACanvas.Font.Name := 'Tahoma';
if FMenu.IsRightToLeft then ACanvas.Font.Charset := ARABIC_CHARSET; if
FMenu.IsRightToLeft then TextFormat := DT_RIGHT + DT_RTLREADING else
TextFormat := 0; if FMenuItem.Default then begin Inc(TextRect.Left, 1);
Inc(TextRect.Right, 1); Inc(TextRect.Top, 1); ACanvas.Font.Color := clGray;
DrawTextEx(ACanvas.Handle, PChar(txt), Length(txt),
TextRect, TextFormat, nil); Dec(TextRect.Left, 1); Dec(TextRect.Right,
1); Dec(TextRect.Top, 1); ACanvas.Font.Color := FFontColor; end;
DrawTextEx(ACanvas.Handle, PChar(txt), Length(txt),
TextRect, TextFormat, nil); txt := ShortCutToText(FMenuItem.ShortCut) + ''; if
FMenu.IsRightToLeft then TextFormat := DT_LEFT else TextFormat :=
DT_RIGHT; DrawTextEx(ACanvas.Handle, PChar(txt),
Length(txt), TextRect, TextFormat, nil); end else begin

```

```
ACanvas.Pen.Color := FSeparatorColor;    ACanvas.MoveTo(ARect.Left + 10,  
TextRect.Top + Round((TextRect.Bottom - TextRect.Top) / 2));  
ACanvas.LineTo(ARect.Right - 2, TextRect.Top +  
Round((TextRect.Bottom - TextRect.Top) / 2)) end;    B.free;    end;    end.
```

По материалам <http://delphi.3000.com>