

How to create a TGroupBox with a checkbox that disables / enables all contained controls when checked

QUESTION

Has anyone seen a with a checkbox attached to the caption? The controls in the groupbox would be disabled if the checkbox is not checked.

```
unit MyGrpBox;

interface

uses
  Windows, Messages, SysUtils, Classes, Controls, Graphics, Dialogs,
  StdCtrls, Math;

type
  TMyGroupBox = class(TGroupBox)
  protected
    FChecked: boolean;
    FCheckBoxRect: TRect;
    procedure WMSetFocus(var Msg: TWMSetFocus); message WM_SETFOCUS;
    procedure WMKillFocus(var Msg: TWMKillFocus); message WM_KILLFOCUS;
    procedure Paint; override;
    procedure Loaded; override;
    procedure MouseDown(Button: TMouseButton; Shift: TShiftState;
      X: Integer; Y: Integer); override;
    procedure KeyDown(var Key: Word; Shift: TShiftState); override;
    procedure PaintCheckBox; virtual;
    procedure DoOnOffControls(AOn: boolean); virtual;
    procedure SetChecked(AValue: boolean);
  public
    constructor Create(AOwner: TComponent); override;
  published
    property Checked: boolean read FChecked write SetChecked;
  end;

procedure Register;

implementation

procedure Register;
begin
  RegisterComponents('My Components', [TMyGroupBox]);
end;

constructor TMyGroupBox.Create(AOwner: TComponent);
begin
  inherited Create(AOwner);
  FChecked := true;
  FCheckBoxRect := Rect(0, 0, 0, 0);
end;

procedure TMyGroupBox.WMSetFocus(var Msg: TWMSetFocus);
begin
  inherited;
  Invalidate;
end;

procedure TMyGroupBox.WMKillFocus(var Msg: TWMKillFocus);
begin
  inherited;
  Invalidate;
end;
```

```

const
  xx_h = 13;
  xx_w = 13;

procedure TMyGroupBox.Paint;
var
  XRect, XCaptionRect: TRect;
  XH, XW: integer;
  XFlags: Longint;
begin
  inherited Paint;
  XH := max(xx_h + 2, Canvas.TextHeight('0'));
  XW := xx_w + Canvas.TextWidth(Text) + 5;
  XRect := RECT(8, 0, 8 + XW, XH);
  FCheckBoxRect := XRect;
  FCheckBoxRect.Right := XRect.Left + 17;
  XCaptionRect := XRect;
  XCaptionRect.Left := FCheckBoxRect.Right;
  Canvas.Brush.Color := Color;
  Canvas.FillRect(XRect);
  PaintCheckBox;
  if Enabled then
    Canvas.Font.Color := Font.Color
  else
    Canvas.Font.Color := clInactiveCaption;
  XFlags := DrawTextBiDiModeFlags(DT_SINGLELINE);
  XFlags := XFlags + DT_VCENTER + DT_SINGLELINE + DT_LEFT;
  DrawText(
    Canvas.Handle, PChar(Text), Length(Text), XCaptionRect, XFlags
  );
  if Focused then
    DrawFocusRect(Canvas.Handle, XCaptionRect);
end;

procedure TMyGroupBox.Loaded;
begin
  inherited Loaded;
  DoOnOffControls(Checked);
end;

procedure TMyGroupBox.MouseDown(Button: TMouseButton;
  Shift: TShiftState; X: Integer; Y: Integer);
begin
  inherited MouseDown(Button, Shift, X, Y);
  if (Button = mbLeft)
    and PtInRect(FCheckBoxRect, Point(X, Y)) then
    Checked := not Checked;
end;

procedure TMyGroupBox.KeyDown(var Key: Word; Shift: TShiftState);
begin
  inherited KeyDown(Key, Shift);
  if Key = VK_SPACE then
    Checked := not Checked;
end;

procedure TMyGroupBox.PaintCheckBox;
var
  XGlyph: TBitmap;
  xX, xY, xStepY, xStepX: integer;
begin
  XGlyph := TBitmap.Create;
  try
    XGlyph.Handle := LoadBitmap(0, PChar(OBM_CHECKBOXES));
    xY := FCheckBoxRect.Top +
      (FCheckBoxRect.Bottom - FCheckBoxRect.Top - xx_h) div 2;
    xX := FCheckBoxRect.Left;
    xStepX := 0;
    xStepY := 0;
    if Enabled then
    begin
      if Checked then
        xStepX := xStepX + xx_w
      end
    end
  end
end;

```

```

else
begin
  if Checked then
    xStepX := xStepX + xx_w * 3
  else
begin
  xStepX := xStepX + xx_w;
  xStepY := xStepY + xx_h*2;
end;
end;
Canvas.CopyRect (
  Rect(xX, xY, xX + xx_w, xY + xx_h),
  XGlyph.Canvas,
  Rect(xStepX, xStepY, xx_w + xStepX, xx_h + xStepY)
);
finally
  XGlyph.Free;
end;
end;

procedure TMyGroupBox.DoOnOffControls(AOn: boolean);
var
  i: integer;
begin
  for i := 0 to ControlCount - 1 do
    Controls[i].Enabled := AOn;
  end;

procedure TMyGroupBox.SetChecked(AValue: boolean);
begin
  if FChecked <> AValue then
begin
  FChecked := AValue;
  Invalidate;
  if not (csReading in ComponentState)
    and not (csLoading in ComponentState) then
begin
  PaintCheckBox;
  DoOnOffControls(AValue);
end;
end;
end;

end.

```

Original resource:	The Delphi Pool
Author:	Serge Gubenko
Added:	2009-10-26
Last updated:	2009-10-26