Drawing rotated text

To create rotated (angled) text you don't need any fancy extra components. Delphi and Windows API provide you with all the functions you need.

This function will use the currently assigned *Canvas.Font* font. If you need to change it do so before calling this function.

```
procedure AngleTextOut(ACanvas: TCanvas; Angle, X, Y: Integer; Text: string);
var
  NewFontHandle,
  OldFontHandle: hFont;
  LogRec       : TLogFont;
begin
  GetObject(ACanvas.Font.Handle, SizeOf(LogRec), Addr(LogRec));
  LogRec.lfEscapement := Angle * 10;
  LogRec.lfOrientation := LogRec.lfEscapement;
  NewFontHandle := CreateFontIndirect(LogRec);
  OldFontHandle := SelectObject(ACanvas.Handle, NewFontHandle);
  ACanvas.TextOut(X, Y, Text);
  NewFontHandle := SelectObject(ACanvas.Handle, OldFontHandle);
  DeleteObject(NewFontHandle);
end;
```

Note: This routine will only work with fonts that support rotation, such as TrueType fonts. If you use the routine with raster fonts, like MS Sans Serif, the text will not be rotated.

There is a similar routine, *DrawAngledText*, in the *Code Snippets Database*.

Example

In this example we will display some angled text in bold 14pt Comic Sans MS as spokes in a wheel. We will make the wheel appear to turn by using a timer to adjust the angle of the text.

Start a new VCL application. Add a field named *fAngle* to the private section of the form's class declaration as follows:

```
  private
    { Private declarations }
    fAngle: Integer;
```

Now create an *OnPaint* event handler for the form. Complete the event handler as follows:

```
procedure TForm1.FormPaint(Sender: TObject);
const
  cText = 'Hello World!'; // text to display
  cPadding = 8;           // padding between text and edge of "wheel"
  cOffset = 20;           // offset of "wheel" from top left of client area
var
  Radius: Integer;  // radius of "wheel"
  I: Integer;       // loop control
begin
  // set up font
  Canvas.Font.Name := 'Comic Sans MS';
  Canvas.Font.Size := 14;
  Canvas.Font.Style := [fsBold];
  // calculate radius of wheel
  Radius := Canvas.TextWidth(cText) + cPadding;
  // draw text "spokes"
  for I := 0 to 3 do
    AngleTextOut(
      Canvas, fAngle + I * 90, Radius + cOffset, Radius + cOffset, cText
    );
  // draw "wheel rim"
  Canvas.Brush.Style := bsClear;
  Canvas.Pen.Width := 2;
  Canvas.Ellipse(cOffset, cOffset, 2 * Radius + cOffset, 2 * Radius + cOffset);
end;
```

Each time the form repaints we draw the text four times, each piece of text is at 90 degrees to its predecessor. We finally draw a circle to represent the wheel rim.

All that remains to do now is to update the angle at which the spokes are drawn by updating *fAngle*. Drop a *TTimer* on the form, set its *Interval* property to 250 and and add the following *OnTimer* event handler.

```
procedure TForm1.Timer1Timer(Sender: TObject);
begin
  Dec(fAngle, 2);
  if fAngle = -90 then
    fAngle := 0;
  Invalidate;
end;
```

This event handler simply changes *fAngle* by 2 degrees and then invalidates the form to make it redraw. We use a negative increment to make the wheel spin clockwise.

Code updated and example created by Peter Johnson

Author:	Unknown
Added:	2007-06-02
Last updated:	2013-10-12