Drawing disabled text

You can draw disabled text, like that you see when you set a *TLabel*'s *Enabled* property to false, by using the *DrawState* Windows API function.

The following routine draws disabled text on a canvas using the canvas' current font:

```
function DrawDisabledText(const Canvas: TCanvas; const X, Y: Integer;
  const Text: string): Boolean;
begin
  Result := DrawState(
    Canvas.Handle,            // device context to draw on
    0,                        // brush (not required)
    nil,                      // output function (not required)
    Integer(PChar(Text)),     // text to be rendered
    Length(Text),             // length of text to be rendered
    X, Y,                     // co-ords of top left of text
    0, 0,                     // size of text (not required)
    DST_TEXT or DSS_DISABLED  // text type and state
  );
end;
```

The *DST_TEXT* flag tells the API routine to display text and *DSS_DISABLED* tells it to draw the text disabled.

If your text contains accelerator characters (i.e. uses the "&" sign to cause the following character to display underlined), then use the *DST_PREFIXTEXT* flag in place of *DST_TEXT*.

> There is a version of *DrawDisabledText*, that can optionally handle accelerator characters, in the *Code Snippets Database*.

Example

In this example we will display the text "Hello World!" in the default font both normally (using *Canvas.TextOut*) and disabled (using *DrawDisabledText*).

Start a new VCL application and create an OnPaint event handler for the form as follows:

```
procedure TForm1.FormPaint(Sender: TObject);
const
  cText = 'Hello World!';
begin
  Canvas.TextOut(8, 8, cText);
  DrawDisabledText(Canvas, 8, 32, cText);
end;
```

If you want to use a different font, set the canvas' font property as required before calling *DrawDisabledText*. For example to change the above code to use 14pt Arial Bold, revise the *FormPaint* method as follows:

```
procedure TForm1.FormPaint(Sender: TObject);
const
  cText = 'Hello World!';
begin
  Canvas.Font.Name := 'Arial';
  Canvas.Font.Size := 14;
  Canvas.Font.Style := [fsBold];
  Canvas.TextOut(8, 8, cText);
  DrawDisabledText(Canvas, 8, 32, cText);
end;
```

Author:	Peter Johnson
Contributor:	Peter Johnson
Added:	2007-10-15
Last updated:	2013-10-12