

```
program txt2pdf;

{$mode objfpc}{$H+}

uses
  Classes, SysUtils, CustApp, fpreport, fpreportpdfexport, fpTTF,
  fpreportstreamer, fpjson, jsonparser;

type

  { TPrintApplication }

  TPrintApplication = class(TCustomApplication)
  private
    FReport : TFPReport;
    FLines : TStringList;
    FData : TFPReportUserData;
    FLineIndex : Integer;
    procedure DoFirst(Sender: TObject);
    procedure DoGetEOF(Sender: TObject; var IsEOF: boolean);
    procedure DoGetNames(Sender: TObject; List: TStringList);
    procedure DoGetNext(Sender: TObject);
    procedure DoGetValue(Sender: TObject; const AValueName: string; var
      AValue: variant);
  protected
    procedure DoRun; override;
  public
    Constructor Create(AOwner : TComponent); override;
    Destructor destroy; override;
  end;

  { TPrintApplication }

  procedure TPrintApplication.DoGetNames(Sender: TObject; List: TStringList);
  begin
    List.Add('Line');
  end;

  procedure TPrintApplication.DoGetEOF(Sender: TObject; var IsEOF: boolean);
  begin
    IsEOF:=FLineIndex>=FLines.Count;
  end;

  procedure TPrintApplication.DoFirst(Sender: TObject);
  begin
```

```
  FLineIndex:=0;  
end;
```

```
procedure TPrintApplication.DoGetNext(Sender: TObject);  
begin  
  Inc(FLineIndex);  
end;
```

```
procedure TPrintApplication.DoGetValue(Sender: TObject; const  
  AValueName: string; var AValue: variant);  
begin  
  AValue:=FLines[FLineIndex];  
end;
```

```
procedure TPrintApplication.DoRun;
```

```
Var
```

```
  PG : TFPReportPage;  
  PH : TFPReportPageHeaderBand;  
  PF : TFPReportPageFooterBand;  
  DB : TFPReportDataBand;  
  M : TFPReportMemo;  
  PDF : TFPReportExportPDF;  
  Fnt : String;
```

```
begin  
  Fnt:='DejaVuSans';  
  Terminate;  
  FLines.LoadFromFile(ParamStr(1));  
  gTTFontCache.ReadStandardFonts;  
  gTTFontCache.BuildFontCache;  
  PaperManager.RegisterStandardSizes;  
  // Page  
  PG:=TFPReportPage.Create(FReport);  
  PG.Data:=FData;  
  PG.Orientation := poPortrait;  
  PG.PageSize.PaperName := 'A4';  
  PG.Margins.Left := 15;  
  PG.Margins.Top := 15;  
  PG.Margins.Right := 15;  
  PG.Margins.Bottom := 15;  
  // Page header  
  PH:=TFPReportPageHeaderBand.Create(PG);  
  PH.Layout.Height:=10; // 1 cm.  
  M:=TFPReportMemo.Create(PH);
```

```
M.Layout.Top:=1;
M.Layout.Left:=1;
M.Layout.Width:=120;
M.Layout.Height:=7;
M.Text:=ParamStr(1);
M.Font.Name:=Fnt;
M.Font.Size:=10;
M:=TFPReportMemo.Create(PH);
M.Layout.Top:=1;
M.Layout.Left:=PG.Layout.Width-41;
M.Layout.Width:=40;
M.Layout.Height:=7;
M.Text:='[Date]';
M.Font.Name:=Fnt;
M.Font.Size:=10;
// Page footer
PF:=TFPReportPageFooterBand.Create(PG);
PF.Layout.Height:=10; // 1 cm.
M:=TFPReportMemo.Create(PF);
M.Layout.Top:=1;
M.Layout.Left:=1;
M.Layout.Width:=40;
M.Layout.Height:=7;
M.Text:='Page [PageNo]';
M.Font.Name:=Fnt;
M.Font.Size:=10;
// Actual line
DB:=TFPReportDataBand.Create(PG);
DB.Data:=FData;
DB.Layout.Height:=5; // 0.5 cm.
DB.StretchMode:=smActualHeight;
M:=TFPReportMemo.Create(DB);
M.Layout.Top:=1;
M.Layout.Left:=1;
M.Layout.Width:=PG.Layout.Width-41;
M.Layout.Height:=4;
M.Text:='[Line]';
M.StretchMode:=smActualHeight;
M.Font.Name:=Fnt;
M.Font.Size:=10;
// Set up data
FData.OnGetNames:=@DoGetNames;
FData.OnNext:=@DoGetNext;
FData.OnGetValue:=@DoGetValue;
```

```
FData.OnGetEOF:=@DoGetEOF;
FData.OnFirst:=@DoFirst;
// Go !
FReport.RunReport;
PDF:=TFPReportExportPDF.Create(Self);
try
  PDF.FileName:=ChangeFileExt(Paramstr(1),'.pdf');
  FReport.RenderReport(PDF);
finally
  PDF.Free;
end;
end;

constructor TPrintApplication.Create(AOwner: TComponent);

begin
  Inherited;
  FReport:=TFPReport.Create(Self);
  FLines:=TStringList.Create;
  FData:=TFPReportUserData.Create(Self);
end;

destructor TPrintApplication.destroy;
begin
  FreeAndNil(FData);
  FreeAndNil(FLines);
  FreeAndNil(FReport);
  inherited destroy;
end;

var
  Application: TPrintApplication;
begin
  Application:=TPrintApplication.Create(nil);
  Application.Title:='Print File Application';
  Application.Run;
  Application.Free;
end.
```