ITF Ingenieurbüro, Dipl. Ing. Thomas Friedmann
info@itf-it.de - www.itf-it.de

Version 1.5 – 17.12.2012

Manual
TPDFImage for Delphi

This Component allows displaying and printing PDF files in Delphi applications without the need for Acrobat Reader to be installed. This is done by using ghostscript as renderer for the PDF files. You only have to include gsdll32.dll and the folders /lib and /fonts (included in this ZIP) with your application.

Restrictions:

a.) This component can‘t write PDF files.

> **New in Version 1.3:** By using SynPDF (Synopse, http://synopse.info/forum/), a Freeware PDF creation library (native Delphi) you can also write PDF files with TPDFImage. But the resulting PDF contains only embedded graphics, TOC and searching is gone then. I personally use this to extract pages from scanned PDF documents. Can be activated with compiler definition USESYNPDF if SynPDF is in the searchpath of Delphi. (see TestPDF project)

b.) Since a PDF is RENDERED to a graphic there is no search possible inside the document. Therefore the primary use of this component is to display or print smaller PDF files. When these files are created by scanning a document this component is perfect in use.

> **New in Version 1.2:** By changing the method to count available pages you can now also open very large PDF files (> 200 pages) in an acceptable time (3-5s). This was tested with the manual of VirtualTreeView from Mike Lische (810 pages).
>
> **New in Version 1.4:** Tested and optimized for use in multithreaded environments. Several smaller enhancements.
>
> **New in Version 1.5:** Smaller corrections for Delphi 7 and up. Without the use of SynPDF TPDFImage saves the current page as Bitmap when using SaveToStream.

License

Installation

It is enough to copy the files itfGSApiDynamic.pas and itfPDFImage.pas to a folder and to add this folder to your search path in Delphi. You only need to add itfPDFImage to your uses clause to make the unit registering TPDFImage as class for opening PDF files (and .ps and .eps) at TPicture in graphics.

For runtime you need the ghostscript API DLL (gsdll32.dll) which you can find at any ghostscript Installation at the folder *bin*. Additional you need the folder /fonts und /lib with their content in your application folder (included in ZIP file).

The class TPDFImage inherits from TBitmap and inherits all properties and methods from TBitmap.

Additional Properties

Name	Datatype	Function
Resolution	Integer	Sets the resolution in DPI for rendering the PDF. Default is the DPI value TScreen delivers. For printing you should set a resolution that fits your needs on quality.
Zoom	Integer	Sets a zoom factor for displaying in a TImage. It's independent from resolution but these two values are responsible for what resolution ghostscript is told to render at.
PageCount	Integer	After loading a PDF-file with LoadFromFile or LoadFromStream this property (read only) shows the number of pages for this file.
CurrentPage	Integer	Contains the actual rendered page. By modifying this value in the range of 1 .. PageCount you can show another page of the file.

Additional methods

Name	Parameter	Function
FirstPage	-	Shows first page of the file
NextPage	-	Shows next page of the file
PreviousPage	-	Shows previous page of the file
LastPage	-	Shows last page of the file
ExtractPagesToFile	AFileName:String APageFrom, APageTo:Integer AAppend:Boolean	**New in Version 1.3** Extracts a page range to a new PDF file. ATTENTION: the created PDF only contains embedded graphics so TOC and searching is gone. With AAppend = TRUE and AFileName is an existing PDF the extracted Pages are append to the existing PDF. ATTENTION: the PDF is also converted to a Graphical PDF. To save all pages use SaveToFile or SaveToStream.

ExtractPagesToStream	AStream:TStream APageFrom, APageTo:Integer	**New in Version 1.3** Extracts a page range to a Stream. (look at ExtractPagesToFile)
SaveToStream	AStream:TStream	**New in Version 1.5** When using SynPDF the PDF is rendered and saved as PDF. If not using SynPDF the current page is saved as Bitmap.

Global Variables

New to Version 1.2

Name	Default	Function
ProgressivePageCount	TRUE	Defines if a jumpsearch is used for scanning the pagecount. It's more effective for large PDF (> 200 pages).
PathToGSDLL	<Programpath>	Defines path to gsdll32.dll
PathToGSLib	<Programpath>\lib	Defines Path to \lib
PathToGSFonts	<Programpath>\fonts	Defines Path to \fonts
MultiThreaded	FALSE	Changes behavior for freeing internal used bitmaps when running multithreaded so there is no memory lost.

Example of usage

```
unit fMainSimple;
```

Neu in Version 1.5
Wenn SynPDF verwendet wird, wird das PDF als PDF gespeichert (gerendert). Wenn SynPDF nicht verwendet wird dann wird die aktuelle Seite als Bitmap gespeichert.

```
interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, ExtCtrls, itfPDFImage;

type
  TForm1 = class(TForm)
    Image1: TImage;
    btnLoad: TButton;
    btnPrev: TButton;
    btnNext: TButton;
    Label1: TLabel;
    btnZoomIn: TButton;
    btnZoomOut: TButton;
    procedure btnLoadClick(Sender: TObject);
    procedure btnPrevClick(Sender: TObject);
    procedure btnNextClick(Sender: TObject);
    procedure btnZoomInClick(Sender: TObject);
    procedure btnZoomOutClick(Sender: TObject);
  private
    { Private-Deklarationen }
  public
    { Public-Deklarationen }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.btnLoadClick(Sender: TObject);
begin
  Image1.Picture.loadFromFile('anleitung.pdf');
  Label1.Caption:='Pagecount = ' + IntToStr(TPDFImage(Image1.Picture.Graphic).Pagecount);
end;

procedure TForm1.btnPrevClick(Sender: TObject);
begin
  TPDFImage(Image1.Picture.Graphic).PreviousPage;
```

```
end;

procedure TForm1.btnNextClick(Sender: TObject);
begin
  TPDFImage(Image1.Picture.Graphic).NextPage;
end;

procedure TForm1.btnZoomInClick(Sender: TObject);
begin
  TPDFImage(Image1.Picture.Graphic).Zoom:=TPDFImage(Image1.Picture.Graphic).Zoom + 25;
end;

procedure TForm1.btnZoomOutClick(Sender: TObject);
begin
  TPDFImage(Image1.Picture.Graphic).Zoom:=TPDFImage(Image1.Picture.Graphic).Zoom - 25;
end;

end.
```

Hints

It is important to add itfIPDFImage to your uses clause of the form using TImage / TPicture to show PDF files because only then TPDFImage is registered at TPicture (graphics) as default class for PDF Files to open.

To access the additional properties of TPDFImage for multi page PDF files you need to cast the graphic of TPicture or TImage to TPDFImage like shown:

TPDFImage(Image1.Picture.Graphic).PreviousPage.

Additionally you can also create an instance of TPDFImage manually:

```
clsPDF:=TPDFImage.Create
clsPDF.CurrentPage:=3;
clsPDF.Resolution:=300;
clsPDF.LoadFromFile(‘annots.PDF’);
```

In this example CurrentPage and Resolution are set before loading the PDF so initially page 3 is rendered with 300 DPI directly.

The resulting Bitmap can like show below directly send to the printer:

Printer.Canvas.Draw(0,0,clsPDF)

or shown on screen with Image1.Picture.Bitmap.Assign(clsPDF)

New to Version 1.2:
The Ghostscript API is completely rewritten to use dynamic loading.

If you
- like or
- hate this component or
- you just find it usefull or
- you want to ask something or
- you have a suggestion or
- you see some copyrights violations or
- you want to complain about my english

feel free to mail me at info@itf-it.de

Thomas Friedmann

ITF Ingenieurbüro