



How to use TrueType fonts in PL/PDF (Standard and Unicode)

V2.4.0

The PDF specification enables the use of TrueType fonts in PDF documents as embedded fonts. PL/PDF supports the use of embedded TTF fonts.

PLPDF v2.2.0 -

PLPDF v2.2.0 contains the new version of TTF handling, but useable TTFLoader or Manually version for creating TTF description records.

This version has new TTF packages:

- plpdf_gl: glyph definition
- plpdf_ttf_parser: TTF file parser

Usage:

1. Obtain the desired TTF file.
2. Load the PLPDF_TTF_FILE table:
 - a. ID (NUMBER): unique identifier
 - a. FONTFILE_NAME (VARCHAR2(255)): TTF file name with extension (.ttf)
 - b. FONTFILE_DATA (BLOB): the binary TTF font
2. Generate the data necessary for the embedding: run plpdf_ttf_parser.StoreTTF procedure:
 - a. P_FONT_FILE_ID: the value of the ID in PLPDF_TTF_FILE
 - b. P_ENC: encoding value, use one of the supported encoding values specified in the PL/PDF documentation (see SetEncoding command). The default value is cp1252. For Unicode use utf16. The encoding is dependent on the TTF file. Please check what the necessary encoding for your font file is.
 - c. P_COMMIT: execute COMMIT statement after inserting records
The generated data is stored in the PLPDF_ADD and PLPDF_ADD_CW tables. The ID is taken from the PLPDF_TTF_S sequence. The ID can be used to reference the font during the generation of the PDF document.

PLPDF v1.*.* - v2.1.0

with TTFLoader

PLPDF TTFLoader is a simple Java GUI utility for generating and uploading TTF files into PLPDF tables. TTFLoader is the user friendly version of manually steps.

Manually (original steps)

To use TTF fonts:

3. Obtain the desired TTF file: to embed the TTF font, the file with the TTF extension is needed. Please take into account that the usage of certain fonts are under copyright. Important: check that the TTF file contains all the characters needed.
4. **For non-Unicode font usage:** Generate the Adobe Font Metrics (AFM) file: the AFM file may be generated by the TTF2PT1 program (<http://ttf2pt1.sourceforge.net/>). The binary version can be downloaded from www.plpdf.com. From the command line run the following command: `ttf2pt1 -a <fontfile>.ttf <fontfile>`. A specific font filename must be given to generate the AFM file. The AFM file is created as `<fontfile>.afm`.



For Unicode font usage: Generate the Unicode Font Metrics (UFM) file: the UFM file may be generated by the TTF2UFM program. The binary version can be downloaded from www.plpdf.com. From the command line run the following command: `ttf2ufm -a <fontfile>.ttf <fontfile>`. A specific font filename must be given to generate the UFM file. The UFM file is created as `<fontfile>.ufm`.

5. Load the PLPDF_TTF_FILE table:
 - a. ID (NUMBER): unique identifier
 - b. FONTFILE_NAME (VARCHAR2(255)): TTF file name with extension (.ttf)
 - c. FONTFILE_DATA (BLOB): the binary TTF font
 - d. AFMFILE_DATA (BLOB): the uploaded AFM file
 - e. UFMFILE_DATA (BLOB): the uploaded UFM file (Unicode support)

To load BLOB type data use SQL*Loader or TOAD.

An example for the SQL*Loader control file is:

```
LOAD DATA
INFILE *
INTO TABLE plpdf_ttf_file
APPEND
FIELDS TERMINATED BY ','
(ID,
FONTFILE_NAME,
AFMFILE_DATA1 filler,
FONTFILE_DATA LOBFILE (FONTFILE_NAME) TERMINATED BY EOF, AFMFILE_DATA
LOBFILE (AFMFILE_DATA1) TERMINATED BY EOF
)
BEGINDATA
10,times.ttf,times.afm
```

6. Load the PLPDF_TTF_AFM table: The content of the AFM file is stored in the table line by line. Use `afm.ctl` or `ufm.ctl` to load the table, and then overwrite the value of the FILE_ID field to the unique identifier specified in the PLPDF_TTF_FILE table.

For non-Unicode font usage: use afm.ctl

For Unicode font usage: use ufm.ctl

7. Generate the data necessary for the embedding: run `plpdf_ttf.StoreTTF` procedure (set numeric character before running with "alter session set NLS_NUMERIC_CHARACTERS='.,';") where the parameters are:
 - a. P_FONT_FILE_ID: the value of the ID in PLPDF_TTF_FILE
 - b. P_ENC: encoding value, use one of the supported encoding values specified in the PL/PDF documentation (see SetEncoding command). The default value is cp1252. For Unicode use utf16. The encoding is dependent on the TTF file. Please check what the necessary encoding for your font file is.The generated data is stored in the PLPDF_ADD and PLPDF_ADD_CW tables. The ID is taken from the PLPDF_TTF_S sequence. The ID can be used to reference the font during the generation of the PDF document.

Use TTF while generating a PDF document

The create data can now be used while generating a PDF document. Here is how to use the fonts in the PDF generator:

- Declare a `Plpdf_Type.t_addfont` type variable, e.g. `l_ttf`
`Plpdf_Type.t_addfont;`



- Retrieve the font using the `Plpdf_TTF.GetTTF` procedure. The parameter is the ID field of the `PLPDF_ADD` table, e.g. `l_ttf := Plpdf_Ttf.GetTTF(21);`
- Add the TTF font to the PDF document: run the `Plpdf.addTTF` procedure, where the parameters are:
 - o `P_FAMILY`: the name that can be used to reference the font in the `SetPrintFont` procedure.
 - o `P_STYLE`: the font style (some TTF fonts only contain the regular style):
 - `NULL`: regular
 - `B`: Bold
 - `I`: Italic
 - `BI`: Bold, Italic
 - o `P_DATA`: data needed to embed the font. Use the result of the `Plpdf_TTF.GetTTF` function.Example: `Plpdf.addTTF('ArialMT', NULL, l_ttf);`
- After this the font can be referenced by the `SetPrintFont` procedure using the `P_FAMILY`.

Warning: using TrueType fonts in conjunction with PDF file Protection may significantly decrease the speed of PDF generation, because the embedded file is also encrypted using the RC4 encryption.