Long-term archiving with PL/PDF

v2.3.1

PDF archiving with Oracle database is the best solution for implementing secure and standard based document management system. PL/PDF is a solution for creating archive PDF (PDF/A) and store it in database.

What is PDF/A?

In September 2005 the International Organization for Standardization (ISO) approved the new PDF/A standard for archiving electronic documents. According to the standard ISO 19005-1, PDF/A is a derivative of PDF that "provides a mechanism for representing electronic documents in a manner that preserves their visual appearance over time, independent of the tools and systems used for creating, storing or rendering the files."

The PDF/A classification is divided into two parts, PDF/A-1 and PDF/A-2. The first classification, PDF/A-1, is further subdivided into two more categories, PDF/A-1a, and PDF/A-1b. The primary difference between PDF/A-1a and PDF/A-1b is the way in which each handles the extraction of text:

- PDF/A-1a: This level, also referred to as Level A Conformance, is fully compliant with the ISO 19005-1 Standard. This version includes tagging, so that text can be extracted and viewed by multiple devices including hand-helds.
- PDF/A-1b: Referred to as Level B Conformance, this category is considered to be the minimal compliance level for PDF/A. This level guarantees that the document can be displayed and read on a computer monitor, but the legibility of the text is not guaranteed.
- PDF/A-2: This is a newer addition to the PDF/A standard and is still being formulated by the Technical Committee. Essentially, PDF/A-2 will deal with some of the newer features being added to the PDF Reference like digital signatures.

PDF/A Rules (from Wikipedia)

A key element to this reproducibility is the requirement for PDF/A documents to be 100% self-contained. All of the information necessary for displaying the document in the same manner every time is embedded in the file. This includes, but is not limited to, all content (text, raster images and vector graphics), fonts, and color information. A PDF/A document is not permitted to be reliant on information from external sources (e.g. font programs and hyperlinks).

Other key elements to PDF/A compatibility include:

- Audio and video content are forbidden.
- JavaScript and executable file launches are forbidden.

- All fonts must be embedded and also must be legally embeddable for unlimited, universal rendering. This also applies to the so-called PostScript standard fonts such as Times or Helvetica.
- Colorspaces specified in a device-independent manner.
- Encryption is disallowed.
- Use of standards-based metadata is mandated.

Advantages To Managing PDF (Unstructured Data) in Oracle (from Oracle)

Databases are often used to catalog and reference documents, images and media content stored in files through “pointer-based” implementations. To store unstructured data inside database tables, Binary Large Objects, or BLOBs have been available as containers for decades.

Oracle Text is the leading text searching, retrieval and management system to be integrated into a database environment, it useful tool for PDFs.

There are many reasons organizations store unstructured data inside Oracle database management systems.

- Robust Administration, Tuning and Management: Content stored in the database can be directly linked with associated data. Metadata and content are maintained in sync; they are managed under transactional control. The database also offers robust services for backup, recovery, physical and logical tuning.
- Simplicity of Application Development: Oracle’s support for a specific type of content includes SQL language extensions, PL/SQL and JAVA APIs, Xpath and Xquery (in the case of XML) and, in many cases, JSP Tag Libraries, as well as algorithms that perform common or valuable operations through built in operators.
- High Availability: Oracle’s Maximum Availability Architecture makes “Zero data-loss” configurations possible for all data. Unlike common configurations where attribute information is stored in the database with pointers to unstructured data in files, only a single recovery procedure is required in the event of failure.
- Scalable Architecture: In many cases, the ability to index, partition, and perform operations through triggers, view processing, or table and database level parameters allows for dramatically larger datasets to be supported by applications that are built on the database rather than on file systems.
- Security: Oracle Database allows for fine-grained (row level and column level) security. The same security mechanisms are used for both structured and unstructured data. When using many file systems, directory services do not allow fine-grained levels of access control. It may

not be possible to restrict access to individual users; in many systems enabling a user to access to any content in the directory gives access to all content in the directory.

PL/PDF as PDF/A creator

PL/PDF version v2.3.1 supports PDF/A-1b standard with simple procedures and restrictions:

- Use Embeded Unicode TTF fonts for all text
- Use „plpdf.SetPDFA1B;" setting PDF/A conformance
- Do not use encryption
- Do not use file or URL media annotation
- Do not use JavaScript

Implementation example

We have created a simple example for demonstrating PDF/A creation with PL/PDF. We used PL/PDF v2.3.1, it's first version wich supports PDF/A.

We have to use TTF font for all text printing, we have choosed DejaVu Serif TTF from http://dejavu-fonts.org. It's a free and popular font package, it supports Latin, Greek, Cyrillic, Georgian script.

Our steps are:

- Upload TTF file into PLDF_TTF_FILE table: we set ID=20 and FONTFILE_NAME= DejaVuSerif.ttf and FONTFILE_DATA containts TTF file as BLOB value. We used PL/SQL Developer for upload file as BLOB.
- Generate TTF descriptive data:

```
begin
  plpdf_ttf_parser.storettf(
    p_font_file_id => 20,
    p_enc => 'utf16',
    p_commit => true
    );
end;
/
```

- Check result in PLPDF_TTF_AD table:

```
select *
from plpdf_ttf_add t
where t.file_id = 20;
```

ID=521, it's the parameter of plpdf_ttf.GetTTF procedure

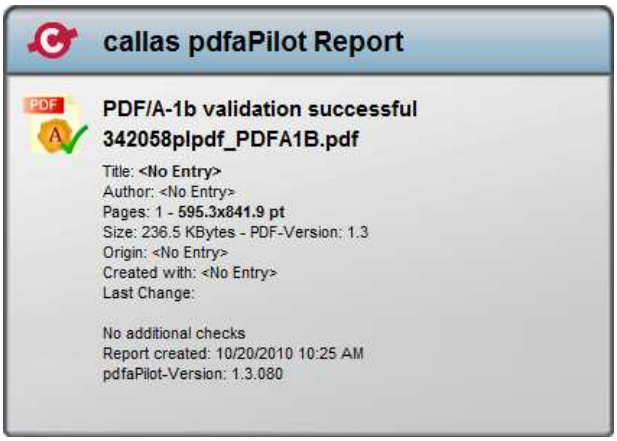
- Create test procedure

```
create or replace procedure test1_pdfa is
   -- PDF/A example
   ---------------
   l_ttf Plpdf_Type.t_addfont;
   l_blob blob;
begin
   l_ttf := Plpdf_Ttf.GetTTF(521);
   Plpdf.init;
   plpdf.SetPDFA1B;
   plpdf.addTTF(p_family => 'DejaVuSerif',p_data => l_ttf);
   Plpdf.NewPage;
   plpdf.SetPrintFont('DejaVuSerif',null,12);
   Plpdf.PrintCell(100,10,unistr('Hello world! űáéúőóüöí'));
   Plpdf.SendDoc(l_blob);

   -- or store
  delete from store_blob;
  insert into store_blob (blob_file, created_date) values (l_blob,
sysdate);
  commit;

end;
/
```

Procedure containts:

  - TTF font embedding and usage in text printing
  - setting PDF/A compilance

- we download result as plpdf_PDFA1B.pdf
- we validated PDF with callas pdfaPilot (it's a free trial service), see http://www.datalogics.com/products/callas/callaspdfA-onlinedemo.asp result is:



good result! thanks for your attention…