



PL/PDF Toolkit
Installation Guide
v2.0.8



Contents

1.	System requirements	3
2.	Get the PL/PDF Toolkit program	3
3.	Create the PLPDF_TK user (if separated install from PLPDF)	3
4.	Prepare Database for AES128 Encryption	3
5.	Java grants	3
6.	Load plpdf_toolkit.jar file	4
7.	Install MergeX storage table	4
8.	Install PLPDF_TOOLKIT packages	4
9.	Install PLPDF_XFDF packages (optional)	4
10.	Test the toolkit (optional)	6
11.	Good to know	6



Reporting in your Oracle database

1. System requirements

Oracle 11g RDBMS Release 1 or higher with **Oracle Jserver (JVM)**
Oracle 11g Express Edition is **not** supported.

2. Get the PL/PDF Toolkit program

Please download the program from www.plpdf.com/downloads. Unzip the plpdf-toolkit.zip into a directory (<unzip_directory>), example c:\downloads.

3. Create the PLPDF_TK user (if separated install from PLPDF)

Start SQL*Plus and connect with an administrator user (example system) to the database. Create the PLPDF_TK user. This example only shows how to create a minimal user (script: 1_create_user.sql). Check out Oracle 11g SQL Reference on how to set up a user:

http://docs.oracle.com/cd/B28359_01/server.111/b28286/statements_8003.htm

```
CREATE USER plpdf_tk IDENTIFIED BY plpdf_tk;  
GRANT CONNECT TO plpdf_tk;  
GRANT RESOURCE TO plpdf_tk;
```

Another way you can execute 1_create_user.sql file as administrator user.

4. Prepare Database for AES128 Encryption

Use LoadJava utility to load "Bouncy Castle" Crypto API libraries.

```
loadjava -user <username>/<password>@<database> -force -verbose  
-resolve -jarsasdbobjects bcprov-<jdk-version>-<num>.jar  
loadjava -user <username>/<password>@<database> -force -verbose  
-resolve bcmath-<jdk-version>-<num>.jar
```

We used bcprov-jdk14-132.jar and bcmath-jdk14-132.jar (Installkit contains these files).

You can download jars from

http://www.bouncycastle.org/latest_releases.html.

You have an easier way. Simply use the loadjar_bcprov.cmd and loadjar_bcmath.cmd files.

5. Java grants

Start SQL*Plus and connect with an administrator user (example system) to the database.

```
dbms_java.grant_permission( '<USERNAME>',  
'SYS:java.security.SecurityPermission', 'putProviderProperty.BC', ''  
);
```

```
dbms_java.grant_permission( '<USERNAME>',  
'SYS:java.security.SecurityPermission', 'insertProvider.BC', '' );
```



Reporting in your Oracle database

```
dbms_java.grant_permission( '<USERNAME>',  
'SYS:java.security.SecurityPermission', 'insertProviderProperty.BC',  
'');
```

```
dbms_java.grant_permission( '<USERNAME>',  
'SYS:java.lang.RuntimePermission', 'getClassLoader', '' );
```

Granting file permission:

- In **Windows**:

```
dbms_java.grant_permission( '<USERNAME>',  
'SYS:java.io.FilePermission', 'com\plpdf\res\-', 'read' );  
end;
```

- In **Unix**-like environments:

```
dbms_java.grant_permission( '<USERNAME>',  
'SYS:java.io.FilePermission', 'com/plpdf/res/-', 'read' );  
end;
```

Another way you can execute `java_grant.sql` file as administrator user.

6. Load plpdf_toolkit.jar file

Use LoadJava utility to load `plpdf_toolkit_v208.jar` file.

```
loadjava -user <username>/<password>@<database> -force -verbose  
-resolve plpdf_toolkit_v208.jar
```

You have an easier way. Simply use the `loadjar.cmd` file.

7. Install MergeX storage table

CONNECT `plpdfTk/plpdfTk@<database>` and execute this command (command from SQL*Plus).

```
@<unzip directory>\plpdfTk_merge_inputs.sql;
```

8. Install PLPDF_TOOLKIT packages

CONNECT `plpdfTk/plpdfTk@<database>` and execute this command: **set scan off**; ("Set scan off" turns off substitution variables.)

Load the parameter package (command from SQL*Plus):

```
@<unzip directory>\plpdf_toolkit_par.sql;
```

Load the toolkit package (command from SQL*Plus):

```
@<unzip directory>\plpdf_toolkit.sql;
```

9. Install PLPDF_XFDF packages (optional)

CONNECT `plpdfTk/plpdfTk@<database>` and execute this command: **set scan off**; ("Set scan off" turns off substitution variables.)

Load the XML package (command from SQL*Plus):

```
@<unzip directory>\plx_sc.sql;
```



Reporting in your Oracle database

Load the XFDF package (command from SQL*Plus):

```
@<unzip directory>\plpdf_xfdf.sql;
```



Reporting in your Oracle database

10. Test the toolkit (optional)

CONNECT plpdf_tk/plpdf_tk@<database> and create the test table.

```
create table store_blob_tk
(
  id number primary key,
  blob_file blob,
  filename varchar2(255),
  description varchar2(2000),
  crd date default sysdate
);
```

Another way you can execute <unzip_directory>\test\store_blob_tk.sql file.

Start SQL*Plus and connect with an administrator user (example system) to the database. Create the test directory.

```
create or replace directory TOOLKIT_TESTFILES as
'<unzip_directory>\test\test_files';
```

Start SQL*Plus and connect with an administrator user (example system) to the database. Directory grants.

```
begin
  dbms_java.grant_permission( 'PLPDF_TK', 'SYS:java.io.FilePermission',
    'TOOLKIT_TESTFILES', 'read' );
end;
```

```
grant read on directory TOOLKIT_TESTFILES to public;
```

Another way you can execute
<unzip_directory>\test\directory_grant.sql file.

CONNECT plpdf_tk/plpdf_tk@<database> and execute this command: **set scan off;** ("Set scan off" turns off substitution variables.)

Load the test files into the test table (command from SQL*Plus):

```
@<unzip_directory>\test\load_test_files.sql;
```

Load the test package (command from SQL*Plus):

```
@<unzip_directory>\test\plpdf_toolkit_test.sql;
```

11. Good to know

The bcprov-jdk14-132.jar is a digitally signed JAR file. The signature information becomes part of the embedded manifest file. The JAR itself is not signed, but instead every file inside the archive is listed along with its checksum; it is these checksums that are signed. When the JVM loads signed JAR files, it can validate the signatures and refuse to load classes that do not match the signature.

Starting with Oracle 11g RDBMS release 1, when you load the contents of a JAR into the database, you have the option of creating a database object (Database Resident JAR) representing the JAR itself. In this



Reporting in your Oracle database

way, you can retain an association between this JAR object and the class, resource, and source objects loaded from the JAR. This enables you to use signed JARs, for example the `bcprov-jdk14-132.jar` file. To create Database Resident JAR object you must to use the following option of the `loadjava` tool:

```
-jarsasdbobjects
```

This option allows to the database to manage the JAR as a single unit. The class, resource, and source objects loaded from the JAR cannot be created or dropped directly. So when you load the `bcmail-jdk14-132.jar` and the `plpdf_toolkit_v204.jar` file, then the `loadjava` tool will not load the manifest files and prompt `ORA-29537` error.

In the following cases it's not an error.

Loading the `bcmail-jdk14-132.jar`

```
creating : resource META-INF/MANIFEST.MF
loading  : resource META-INF/MANIFEST.MF
Error while creating resource META-INF/MANIFEST.MF
ORA-29537: class or resource cannot be created or dropped directly
ORA-06512: at line 1
```

```
creating : resource META-INF/BCKEY.SF
loading  : resource META-INF/BCKEY.SF
Error while creating resource META-INF/BCKEY.SF
ORA-29537: class or resource cannot be created or dropped directly
ORA-06512: at line 1
```

```
creating : resource META-INF/BCKEY.DSA
loading  : resource META-INF/BCKEY.DSA
Error while creating resource META-INF/BCKEY.DSA
ORA-29537: class or resource cannot be created or dropped directly
ORA-06512: at line 1
```

Loading the `plpdf_toolkit_v204.jar`

```
creating : resource META-INF/MANIFEST.MF
loading  : resource META-INF/MANIFEST.MF
Error while creating resource META-INF/MANIFEST.MF
ORA-29537: class or resource cannot be created or dropped directly
ORA-06512: at line 1
```