DidiSoft OpenPGP Library for Java version 3.0

Table of contents

Introduction 3
Setup 4
- JAR files 4
- Switching from Trial/Evaluation/ version to Production 4
- Javadoc in Eclipse 4
- OracleDB 4

Migration guide 6
- From version 2.7 to 3.0 6
- From version 2.6.5 to 2.6.6 6
- From version 2.6.2 to 2.6.3 6
- From version 2.6.1 to 2.6.2 6
- From version 2.5.x to 2.6.0 7

Examples 8
- Example Files 8
- Examples Online 8

Appendix 10
- Common Exceptions 10
- Exporting keys from a GnuPG keystore 10

Support 11

Introduction

Introduction

This documentations refers to DidiSoft OpenPGP Library for Java.
Intended audience: software engineers, software architects, system administrators.

About the Library

DidiSoft OpenPGP Library for Java is a 100% Java library with no external dependencies.

The library provides functions for OpenPGP encryption, decryption, signing, verification of signed data, clear text signing, one pass signing and encryption, key pair generation, key signing, key revocation, etc.

The library uses internally the open source **BouncyCastle library** in a **no-conflict mode**. This means that you can still use other versions of the BouncyCastle library in your project without worry for class collision conflicts with our library JAR files.

Setup

JAR files

The library consists of three JAR files located in the **Bin** folder of the library distribution ZIP file:

1) bcpg-jdk15on-15-lw.jar
2) bcprov-jdk15on-15-lw.jar
3) pgplib-3.0.x.jar

They must be copied, referenced and distributes with your software in order the library to work.

Switching from Trial/Evaluation/ version to Production

After a purchase you will receive **download** instructions for the **production copy** of the library.

Please **download** the **production copy** ZIP file and **replace** in your project the **evaluation** version JAR files with the once from the **/Library** folder of the **production copy** ZIP archive.

The same process should be applied also for **upgrade to a newer version** of the library.

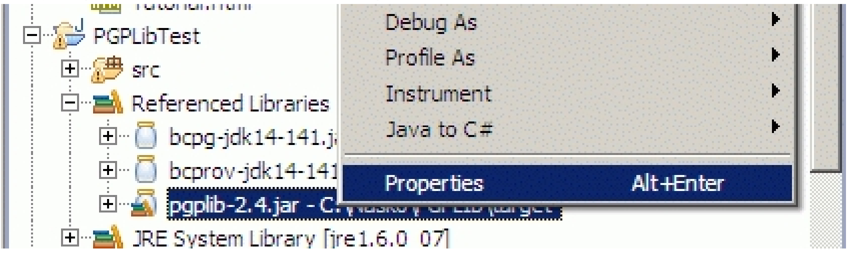
Javadoc in Eclipse

This article is a short list of steps to perform in order to see more meaningful **tooltips** when programming with DidiSoft OpenPGP Library for Java. It assumes that you use Eclipse as your Java IDE.

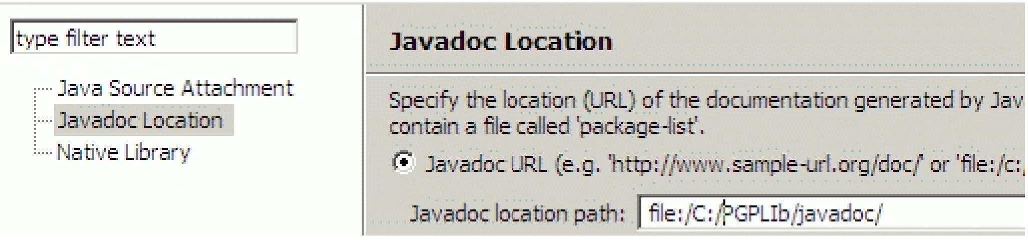
1. Download and unpack library ZIP.

2. Start a new **Eclipse project** and reference the three JARS located in the **Bin** folder in the location from step 1.

3. In your project **Referenced Libraries** section in the Eclipse **Package Explorer** tab right click pgplib-x.x.jar (x.x is your version in use), select **Properties.**



4. In the Javadoc Location dialog enter the location of the **JavaDoc** folder where the library was extracted in step 1.



Specify the location (URL) of the documentation generated by Jav
contain a file called 'package-list'.

● Javadoc URL (e.g. 'http://www.sample-url.org/doc/' or 'file:/c:

Javadoc location path: file:/C:/PGPLIb/javadoc/

5. Now the JavaDoc should appear when you type methods or properties of the objects from the library, or simply press **F2** when you are over an already typed method.

OracleDB

The library can be used inside Oracle Database version 11 and 12 for usage in Java Stored Procedures.

In order to load the library we have to use the **loadjava** tool tha ships with Oracle:

Assuming that we have an Oracle Database user with name '*user*' and password '*pass*'
Loading the library:

```
loadjava.bat -v -r -u user/pass \didisoft\jce-jdk13-151.jar
loadjava.bat -v -r -u user/pass \didisoft\bcpg-jdk13-151.jar
loadjava.bat -v -r -u user/pass \didisoft\pgplib-2.7.0.jar
```

These **security permissions** must also be granted to the database user '*user*':

```
SQL>call dbms_java.grant_permission( user', 'SYS:java.security.SecurityPermission',
'putProviderProperty.BC', " );
SQL>call dbms_java.grant_permission( 'user', 'SYS:java.security.SecurityPermission', 'insertProvider.BC', " )
;
SQL>commit;
```

Migration guide

If not specified explicitly the upgrade to a newer version of the library should be no different from switching from trial version to production.

Below you will find specific guidelines how to migrate from a particular version, where extra steps are needed.

From version 2.7 to 3.0

Replace the JAR files:

1) bcpg-jdk15on-151.jar -> bcpg-jdk15on-15-lw.jar
2) bcprov-ext-jdk15on-151.jar -> bcprov-jdk15on-15-lw.jar
3) pgplib-2.7.x.jar -> pgplib-3.0.0.jar

From version 2.6.5 to 2.6.6

In version 2.6.6 all the encryption and signing methods that work with streams leave the output streams open.
In previous versions the output streams were closed by the methods.

List of affected methods:

- PGPLib.encryptStream
- PGPLib.encryptStreamPBE
- PGPLib.signStream
- PGPLib.signStreamVersion3
- PGPLib.signAndEncryptStream
- PGPLib.signAndEncryptStreamVersion3

In order to migrate properly to version 2.6.6 we have to take care to explicitly close the output streams, like this:

```
PGPLib pgp = new PGPLib();

OutputStream outputStream = ...
try {
 pgp.encryptStream( dataStream, keyStream, outputStream, true, false);
} finally {
 outputStream.close();
}
```

From version 2.6.2 to 2.6.3

In version 2.6.3 by default the library is compiled for JDK 1.5 and BouncyCastle provider 1.49

Migration guide:

1) Replace pgplib-2.6.jar with pgplib-2.6.3.jar
2) Replace bcpg-jdk14-145-ecc.jar with bcpg-jdk15on-149-ecc.jar
3) bcprov-ext-jdk14-145.jar with bcprov-ext-jdk15on-149.jar

Backward (JDK 1.4) compatibility

For JDK 1.4 compatibility you can continue to use the jar files located under the /Library/jdk14 folder.
They will continue to be maintained with the same set of new features and mehods as the core JAR files.

From version 2.6.1 to 2.6.2

In version 2.6.2 additional code for Elliptic Curve cryptography was added to the bcpg-jdk14-145.jar file. In order to avoid using the old file the file has been renamed to:

bcpg-jdk14-145-ecc.jar

Migration guide:

1) Replace **bcpg-jdk14-145.jar** with **bcpg-jdk14-145-ecc.jar**

From version 2.5.x to 2.6.0

As of version 2.6.0 the plain encrypt methods (**PGPLib.encrypt**...) throw **java.io.IOException** in addition to com.didisoft.pgp.PGPException.

In order to migrate from version 2.5 you will also have to catch **java.io.IOException** into an additional catch clause.

Methods affected:
PGPLib.encryptStream
PGPLib.encryptFile
PGPLib.encryptStreamPBE
PGPLib.encryptFilePBE

Examples

Example Files

In the library distribution ZIP package you will find a folder named **Examples.**

Create a new Java project with your favorite IDE of choice and add the files from the **Examples\src** folder.

The other files in the Examples folder are used as data for some of the examples.

Examples Online

All the examples below are available online at our web site:
http://www.didisoft.com/java-openpgp/examples/

Quick introduction to OpenPGP
Getting Started with the library
Setup instructions
Exception handling guidelines

Most common functions

Encrypt
Decrypt
Decrypting with a password
Sign
Verify
Sign and Encrypt
Decrypt and Verify
Clear text sign

KeyStore and key generation.

Properties of a Key
KeyStore introduction
Generate RSA keys
Generate DH/DSS keys
Import keys
Export keys
Deleting keys
Changing private key password

Key revocation

Introduction to OpenPGP key revocation
Revoke key directly
Revocation certificate
Designated revoker

Advanced Topics

Set preferred cypher (symmetric key algorithm)
Set preferred compression
Set preferred hashing

Miscellaneous

Inspecting OpenPGP archives
Logging

Appendix

Common Exceptions

Some common exceptions that may occur when working with the library are:

org.bouncycastle.openpgp.PGPException: Exception creating cipher

org.bouncycastle.openpgp.PGPException: exception constructing public key

org.bouncycastle.openpgp.PGPException: exception encrypting session key

java.lang.SecurityException: Unsupported keysize or algorithm parameters

Resolution:

Try to download the files listed in JCE Unlimited Strength Policy files and try again.

If the problem appears after that, please contact us.

Exporting keys from a GnuPG keystore

List keys contained in the GnuPG keystore:

```
gpg --list-keys
```

Export Public key

```
gpg --export my_key -o my_public_key.gpg
```

Export Private key

```
gpg --export-secret-key my_key -o my_secret_key.gpg
```

Support

Technical support

To receive general information or **technical support**, please contact us at support@didisoft.com.

Sales

For questions related to sales, volume licensing, or OEM licensing, please contact us at sales@didisoft.com.

Product Updates

If you have purchased the library you can access our Customers' Portal where you can download new versions.

Newsletter

To receive product update news, you can subscribe to our Newsletter

For further information, visit us at www.didisoft.com
If you have any ideas, wishes, questions or criticism, don't hesitate to contact us. We will be glad to hear from you.