BeeSoft Abeona 2012.1

User Manual

www.beesoft.eu

Content

1 Introduction to Abeona

Abeona is the tool and library for creating and validating license files for Java applications. It is based on a set of properties they are characteristic for the customer and purchased application's configuration, and also on some cryptography.

The basic idea is very simple:

- you protect your application with license file, that is standard Java properties file
- properties can contain whatever you want - customer name, user MAC address, application serial number, enabled (purchased) modules, and so on...
- from those properties is calculated standard MD5 hash code, encoded with private RSA key and stored as a digital signature to the properties
- you have license file built
- when is your application running, license is loaded from file, signature is decoded with RSA public key and decoded hash code is compared to hash code calculated from properties now
- if there is some difference, license is invalid

If you are new to the RSA cryptography, you should know that public / private keys pair are very large numbers bound together via mathematical theory about asymmetric cryptography. And in practice, all you need to know is that what you encode using private key, can be decoded with public key only. You can publish public key only, private key must stand secret.

Abeona uses 2048 bits keys. There is assumption computers will break this cipher in the year 2030. So you can protect your applications for next 20 years.

Note: This software is named *Abeona* after ancient Roman Goddess who protects children when they leave the parents' home.

2 Installation

Simply unzip downloaded file to appropriate directory (you must create such directory). You will get this structure:

```
ABEONA                 - this is the root directory when you unzip
    abeona_2012_1.jar  - runnable JAR file
    abeona_2012_1.sha  - SHA-256 hash code of the JAR file
    license.txt        - license agreement
    release.txt        - list of the release changes
    DOC                - documentation directory
         manual.pdf    - Abeona manual, this is what you are reading
         API           - Javadoc directory
    EXAMPLES           - Java examples for Abeona usage
```

2.1 Abeona directory

Abeona creates its own working directory when first time running . This directory is located in <user-home> directory (on Windows systems *C:\Documents and Settings\<user-name>*) and is called *abeona*.

In this directory is stored Abeona product database and (if you purchased it) Abeona license file.

2.2 Abeona license

Abeona is protected by itself. The same system we offer to you is used to protect Abeona. So if you want use Abeona commercially, you must purchase valid license.

After you have payed Abeona, you will get Abeona license file to your email. Copy this file to Abeona working directory (see previous chapter) and start Abeona.

You should see you license number and name in the bottom left corner of application window.

3 Running Abeona GUI Tool

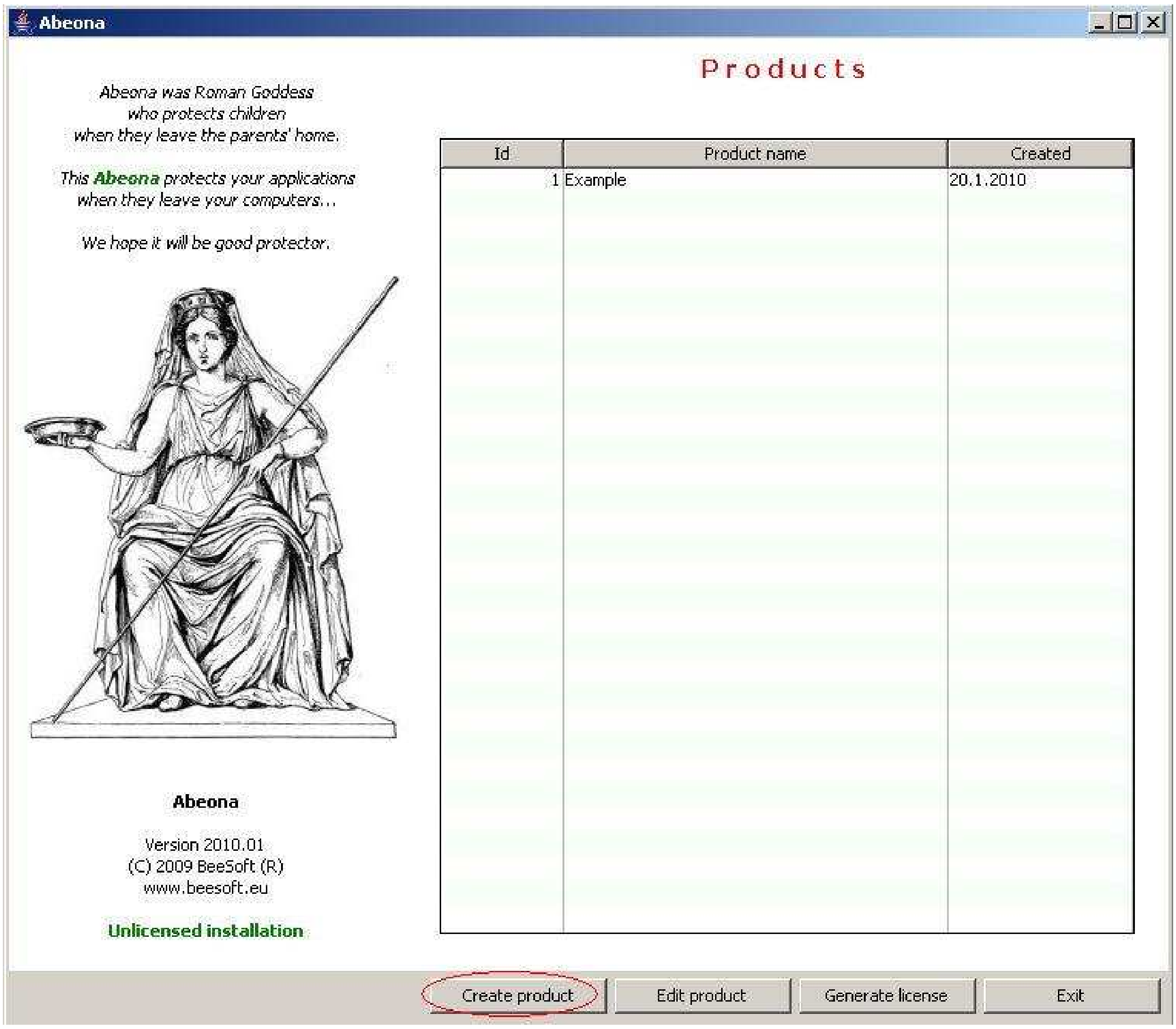
Abeona GUI tool can be runned from command line with this command:

```
java -jar abeona_2012_1.jar
```

You must complete path to Java, if you have not set environment property (PATH=). You must complete path to JAR file if you are not in directory with abeona.jar.

3.1 List of Products

This is the first screen you will see when running Abeona:



Abeona was Roman Goddess
who protects children
when they leave the parents' home.

This **Abeona** protects your applications
when they leave your computers...

We hope it will be good protector.

Abeona

Version 2010.01
(C) 2009 BeeSoft (R)
www.beesoft.eu

Unlicensed installation

Id	Product name	Created
1	Example	20.1.2010

Note green text in left bottom corner. This is an information about your Abeona license. You can see two possible values here:

1. **Unlicensed installation** if you have not payed Abeona license
2. **License number** and **license owner** name loaded from license file

Table lists all products you have created.

There are buttons at bottom:

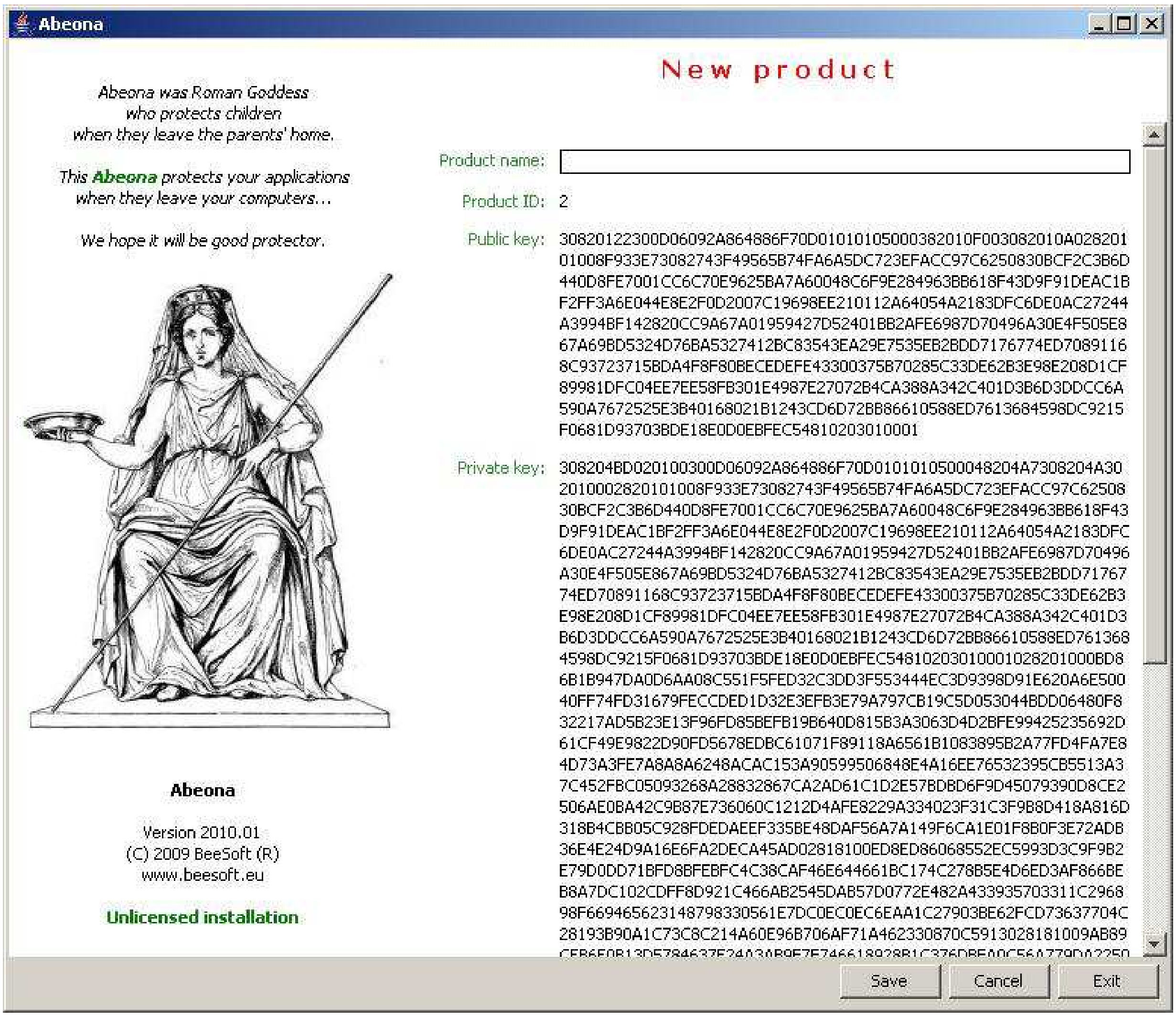
- *Create product* - create new product (record in table)
- *Edit product* - when you select product (click on record) and press this button, form to edit selected product appears.
- *Generate license* - generates new license file. If you have already selected product, such product will be offered in appeared form.

- *Exit* - closes Abeona tool

3.2 Creating Product

When you finishes with new application you will need create a new Product in Abeona system. Product is a simple record with a name (usually the name of your application) and with a template of your future license file.

Press *Create product* button to get this form:



Abeona was Roman Goddess
who protects children
when they leave the parents' home.

This **Abeona** protects your applications
when they leave your computers...

We hope it will be good protector.

Abeona

Version 2010.01
(C) 2009 BeeSoft (R)
www.beesoft.eu

Unlicensed installation

Product name:

Product ID: 2

Public key:

```
30820122300D06092A864886F70D01010105000382010F003082010A028201
01008F933E73082743F49565B74FA6A5DC723EFACC97C6250830BCF2C3B6D
440D8FE7001CC6C70E9625BA7A60048C6F9E284963BB618F43D9F91DEAC1B
F2FF3A6E044E8E2F0D2007C19698EE210112A64054A2183DFC6DE0AC27244
A3994BF142820CC9A67A01959427D52401BB2AFE6987D70496A30E4F505E8
67A69BD5324D76BA5327412BC83543EA29E7535EB2BDD7176774ED7089116
8C93723715BDA4F8F80BECEDEFE43300375B70285C33DE62B3E98E208D1CF
89981DFC04EE7EE58FB301E4987E27072B4CA388A342C401D3B6D3DDCC6A
590A7672525E3B40168021B1243CD6D72BB86610588ED7613684598DC9215
F0681D93703BDE18E0D0EBFEC54810203010001
```

Private key:

```
308204BD020100300D06092A864886F70D0101010500048204A7308204A30
2010002820101008F933E73082743F49565B74FA6A5DC723EFACC97C62508
30BCF2C3B6D440D8FE7001CC6C70E9625BA7A60048C6F9E284963BB618F43
D9F91DEAC1BF2FF3A6E044E8E2F0D2007C19698EE210112A64054A2183DFC
6DE0AC27244A3994BF142820CC9A67A01959427D52401BB2AFE6987D70496
A30E4F505E867A69BD5324D76BA5327412BC83543EA29E7535EB2BDD71767
74ED70891168C93723715BDA4F8F80BECEDEFE43300375B70285C33DE62B3
E98E208D1CF89981DFC04EE7EE58FB301E4987E27072B4CA388A342C401D3
B6D3DDCC6A590A7672525E3B40168021B1243CD6D72BB86610588ED761368
4598DC9215F0681D93703BDE18E0D0EBFEC5481020301000102820100BD8
6B1B947DA0D6AA08C551F5FED32C3DD3F553444EC3D9398D91E620A6E500
40FF74FD31679FECCDED1D32E3EFB3E79A797CB19C5D053044BDD06480F8
32217AD5B23E13F96FD85BEFB19B640D815B3A3063D4D2BFE99425235692D
61CF49E9822D90FD5678EDBC61071F89118A6561B1083895B2A77FD4FA7E8
4D73A3FE7A8A8A6248ACAC153A90599506848E4A16EE76532395CB5513A3
7C452FBC05093268A28832867CA2AD61C1D2E57BDBD6F9D45079390D8CE2
506AE0BA42C9B87E736060C1212D4AFE8229A334023F31C3F9B8D418A816D
318B4CBB05C928FDEDAEEF335BE48DAF56A7A149F6CA1E01F8B0F3E72ADB
36E4E24D9A16E6FA2DECA45AD02818100ED8ED86068552EC5993D3C9F9B2
E79D0DD71BFD8BFEBFC4C38CAF46E644661BC174C278B5E4D6ED3AF866BE
B8A7DC102CDFF8D921C466AB2545DAB57D0772E482A433935703311C2968
98F669465623148798330561E7DC0EC0EC6EAA1C27903BE62FCD73637704C
28193B90A1C73C8C214A60E96B706AF71A462330870C5913028181009AB89
```

Abeona generates new values of private and public key for created product when opening this form. It can take 3 - 15 seconds, so be patient, please.

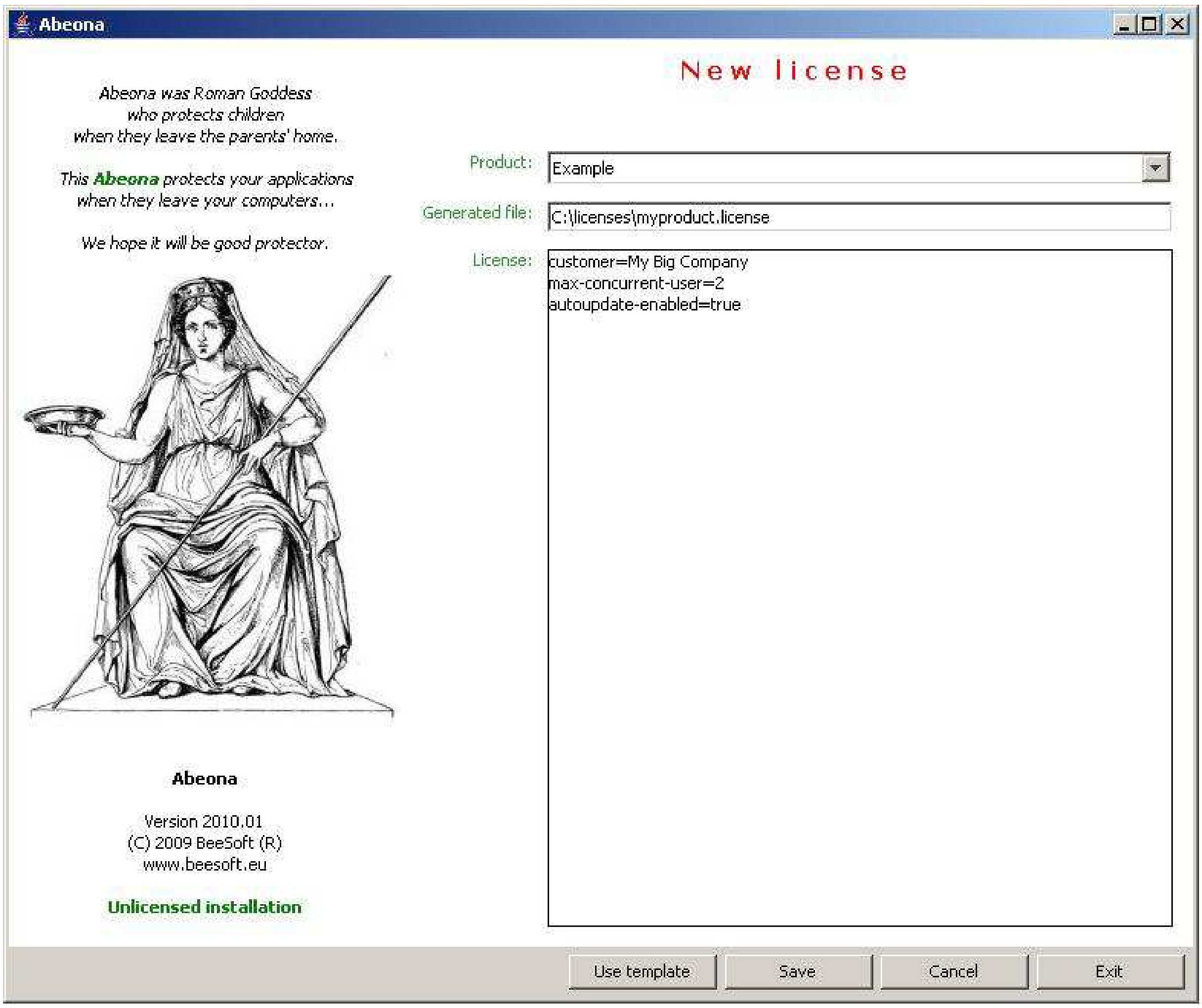
In this form you can fill two fields:

- **Product Name** - name of your application
- **License Template** - list of properties you want generate to license files. There is not necessary to fill this field (you can create / modify license properties when you generate license file, of course), but if you fill it, Abeona will offer it for you when generating license. That means less work and mistakes.

Press *Save* button to save new product to database.

3.3 Generate License

A new license you can generate from product list form.



Abeona was Roman Goddess
who protects children
when they leave the parents' home.

This ***Abeona*** *protects your applications*
when they leave your computers...

We hope it will be good protector.

Abeona

Version 2010.01
(C) 2009 BeeSoft (R)
www.beesoft.eu

Unlicensed installation

Product: Example

Generated file: C:\licenses\myproduct.license

License:

```
customer=My Big Company
max-concurrent-user=2
autoupdate-enabled=true
```

There are three fields you can / must fill:

- From **Product** combobox select product you want generate license for.
- **Generated file** contains path and name of generated file
- **License** is content of the license. Here you fill properties you want sign.

Press *Use template* button to fill **License** field from selected product template.

Press *Save* button to save new license to the file.

4 Generate License From Code

License file can be generated not only from GUI tool, but also from code. This capability is advantageous when you want generate license file on-line (e.g. via servlet).

All you need to generate license file from code is:

- properties (names and values) you want sign in generated license file
- ID of product you generate license for. This ID you can see in GUI tool in *product list form* or in *edit product form*

Your code is very simple, as you can see in next example. Note, properties and product ID in this example are fixed, you must use them as arguments in real life.

```
import java.io.IOException;
import java.util.Properties;
import eu.beesoft.abeona.Abeona;
import eu.beesoft.abeona.LicenseException;
import eu.beesoft.abeona.LicenseGenerator;
import eu.beesoft.abeona.Product;
import eu.beesoft.abeona.Database;

/**
 * This example demonstrates how to create a new license file out of GUI tool
* just from code.
 *
 * @author Eugen Kocak
 */
public class BuildLicenseFromCode {

        public BuildLicenseFromCode (Abeona abeona) {

                // prepare your properties for generated license
                Properties properties = new Properties ();
                properties.setProperty ("customer", "Very Big Company");
                properties.setProperty ("dbAccess", "enabled");
                properties.setProperty ("netUpdate", "disabled");
                properties.setProperty ("user.count", "25");

                // get product database
                // it contains all your product (application) records
                Database db = abeona.getDatabase ();

                // get product by ID
                // here we obtain product #10
                // you can see right product ID via GUI tool
                Product product = db.getProduct (10);

                // get license generator
                LicenseGenerator generator = abeona.getLicenseGenerator ();

                // sign and store license
                try {
                        Properties signedProperties = generator.generate
(properties, product);
                        generator.storeLicense (signedProperties, product,
                                "c:/myNew.license");
                }
                catch (LicenseException le) {
                        // TODO
                        // thrown when cannot sign properties
                }
                catch (IOException ioe) {
                        // TODO
                        // thrown when cannot write properties to file
                }
        }



        public static void main (String[] arguments) {
                // you must always call main() method in Abeona
                // it requires as argument name of class to run
                // if it is null, GUI tool is invoked
                Abeona.main (new String[] { BuildLicenseFromCode.class.getName
() });

// or you can run Abeona from command line this way:
// java  eu.beesoft.abeona.Abeona
// eu.beesoft.abeona.example.BuildLicenseFromCode
        }
}
```

There is also a new way of generating a license file from code on the server in release 2012.1. It uses Abeona as a singleton instance and generates license in a few steps. Here is an example:

```
        Properties properties = new Properties ();
        properties.setProperty ("customer", "Very Big Company");
        properties.setProperty ("dbAccess", "enabled");
        properties.setProperty ("netUpdate", "disabled");
        properties.setProperty ("user.count", "25");
        Abeona instance = Abeona.getInstance ();
        instance.generateLicense (selectedProductId, properties,
"d:/licenses/file001.license");
```

5 Validating License

To validate license in your application you must:

1. copy **public** key from suitable product in GUI tool into your application code (store it as final static variable or use as argument in next example)
2. obtain InputStream from license file
3. load properties to instance of License class

If somebody changed license file, properties loading throws exception. And if license properties are not valid, each call to method *getProperty (String)* throws LicenseException too.

Here is an example:

```
import java.io.InputStream;
import eu.beesoft.abeona.License;
import eu.beesoft.abeona.LicenseException;

/**
 * This example demonstrates how to use Abeona license system in real
 * application.
 *
 * @author Eugen Kocak
 */
public class StandaloneApplication {

        // public key generated by Abeona GUI tool
        // you must copy it to your application manually
        // in this example the default key (unlicensed Abeona) is used
        private final static String publicKey =
"30820122300D06092A864886F70D01010105000382010F003082010A02820101008F933E7308
2743F49565B74FA6A5DC723EFACC97C6250830BCF2C3B6D440D8FE7001CC6C70E9625BA7A6004
8C6F9E284963BB618F43D9F91DEAC1BF2FF3A6E044E8E2F0D2007C19698EE210112A64054A218
3DFC6DE0AC27244A3994BF142820CC9A67A01959427D52401BB2AFE6987D70496A30E4F505E86
7A69BD5324D76BA5327412BC83543EA29E7535EB2BDD7176774ED70891168C93723715BDA4F8F
80BECEDEFE43300375B70285C33DE62B3E98E208D1CF89981DFC04EE7EE58FB301E4987E27072
B4CA388A342C401D3B6D3DDCC6A590A7672525E3B40168021B1243CD6D72BB86610588ED76136
84598DC9215F0681D93703BDE18E0D0EBFEC54810203010001";



        /**
```

```
         * This piece of code shows how to load license file and how check /
use its
         * properties.
         *
         * @throws LicenseException
         */
        public static void main (String[] arguments) throws LicenseException {
                License license = new License ();

                // obtain input stream from license file
                // here you must use your own generated license file and path
      // to it
                InputStream propertiesStream = StandaloneApplication.class
                        .getResourceAsStream
("/eu/beesoft/abeona/example/example.license");

                // load properties from file to instance of License class
                // LicenseException is thrown here if something goes wrong
                license.load (propertiesStream, publicKey);

                // get properties from license
                // each call to getProperty() throws LicenseException if
//license is invalid
                String customerName = license.getProperty ("customer");
                System.out.println ("Customer: " + customerName);

                String serialNumber = license.getProperty ("serial-number");
                System.out.println ("Serial #: " + serialNumber);

                String isUpdateEnabled = license.getProperty ("update");
                System.out.println ("Update: " + isUpdateEnabled);

                // here you can see the check for user system name
                String licenseUser = license.getProperty ("user");
                System.out.println ("License user: " + licenseUser);
                String currentUser = License.getUserName ();
                System.out.println ("Current user: " + currentUser);
                if (!currentUser.equals (licenseUser)) {
                        // do something
                }

                // here is similar check for MAC address
                String licenseMacAddress = license.getProperty ("macAddress");
                System.out.println ("License MAC address: " +
licenseMacAddress);
                String currentMacAddress = License.getMacAddress ();
                System.out.println ("Current MAC address: " +
currentMacAddress);
                if (!currentMacAddress.equals (licenseMacAddress)) {
                        // do something
                }
        }

}
```

Of course, you must ship *abeona_2012.1.jar* file with your application.

6 Abeona own protection

One possible way to break Abeona license protection is replace original Abeona JAR file by fake JAR with the same name, same classes and methods, but with the different behaviour.

In our examples you can find class **ProtectAbeonaJar**. This class contains a few methods to protect Abeona JAR file against such attack. All you need to do is to copy these methods into your application code (there are no dependencies except declared imports of standard Java classes). Then you can invoke them when your application starts. Of course, you can rename them to increase the protection.

There are two methods to validate Abeona JAR file:

1. *validateAbeonaJarLocally()* - you can use this method to validate Abeona JAR file if your application has no Internet access. Method has built-in hash codes for all known Abeona releases, so it compares hash code of Abeona JAR file computed in runtime to those patterns. Disadvantage of this method is need to update this method whenever you update Abeona version.
2. *validateAbeonaJarRemote()* - this method is usable when your application has Internet access. It validates Abeona JAR you currently use against hash codes published on www.beesoft.eu. This method will work without any edit with future releases of Abeona.

Both methods use SHA-256 algorithm to compute hash codes and throw *java.lang.Exception* if there is not a compliance. It is your choice which method you use. Or you can use both depending on the situation.

7 License Server

Abeona classes have capability to work as license server.

Class **eu.beesoft.abeona.License** has implemented method

```
void load (String host, String port, String applicationName, String
publicKey)
```

to load properties from license server. Arguments *host* and *port* are connection strings to server (HTTP is used), *applicationName* is your own identifier for running application (the same name must be used on client and server side). Argument *publicKey* is public key of suitable product as in previous example. But this argument can also be *null* and in such case are properties just loaded from server but not validated.

Class **eu.beesoft.abeona.LicenseServer** is abstract class you can use to build your own license server. There is just one method you need serve:

```
        Properties getProperties (String application, String user);
```

Argument *application* is application name as received from client. Argument *user* is system name of current client user and is produced by License class.

To build license server you need create class derived from LicenseServer and write method getProperties(). What you will send to client or how you will check user request is in your hands only.