

---

# HSM Proxy Installation Manual

Version 0.2.0

Frank Cornelis

Copyright © 2013 FedICT

## Abstract

This manual describes the installation procedure for the HSM Proxy product.

1. Introduction .....	1
2. Application Server .....	1
2.1. Logging .....	2
2.2. Security Domains .....	3
2.3. Database .....	3
2.4. AJP Connector .....	6
2.5. Starting .....	6
2.6. Deploying .....	6
2.7. Undeploying .....	7
3. Apache .....	7
4. Compiling JBoss Application Server .....	7
A. HSM Proxy Installation Manual License .....	8
B. HSM Proxy Project License .....	8

## 1. Introduction

This installation manual assumes installation of the HSM Proxy product on Linux, preferably RHEL 6.4+ or CentOS 6.4+.

A Java runtime needs to be available. This can be either Java 6 or Java 7.

## 2. Application Server

The HSP Proxy product has been targeted towards JBoss AS 7.1.3.Final, and 7.2.0.Final. Instructions on how to compile the JBoss Application Server are given in [Section 4, “Compiling JBoss Application Server”](#). The application also works on JBoss EAP 6.1.0.Alpha1, and 6.1.0.Beta1.

In the following sections we assume that `$JBOSS_HOME` denotes the directory where the JBoss Application Server lives.

You need to configure the Application Server for HSM Proxy to be able to run properly. For the JBoss Application server, the configuration file is `$JBOSS_HOME/standalone/configuration/standalone.xml`

### 2.1. Logging

Add under `<subsystem xmlns="urn:jboss:domain:logging:1.2">`

```
<file-handler name="FEDICT" autoflush="true">
  <level name="DEBUG"/>
  <file relative-to="jboss.server.log.dir" path="fedict.log"/>
  <append value="true"/>
</file-handler>
<logger category="be.fedict" use-parent-handlers="false">
  <level name="DEBUG"/>
  <handlers>
    <handler name="FEDICT"/>
  </handlers>
</logger>
```



#### Logging level

Depending on the production load and/or security requirements, you can change the logging level to `INFO` or even `WARN`.

View the HSM Proxy specific logging via:

```
cd $JBASS_AS/standalone/log/
tail -F fedict.log
```

If you need to check the JBoss security layer for correct operations, add:

```
<logger category="org.jboss.as.security">
  <level name="TRACE"/>
  <handlers>
    <handler name="FEDICT"/>
  </handlers>
</logger>
<logger category="org.jboss.security">
  <level name="TRACE"/>
  <handlers>
    <handler name="FEDICT"/>
  </handlers>
</logger>
```

## 2.2. Security Domains

Add within `<subsystem xmlns="urn:jboss:domain:security:1.2">` under `<security-domains>`

```
<security-domain name="hsm-proxy-client" cache-type="default">
  <authentication>
    <login-module code="org.jboss.security.ClientLoginModule"
      flag="required">
      <module-option name="multi-threaded" value="true"/>
      <module-option name="restore-login-identity" value="true"/>
    </login-module>
  </authentication>
</security-domain>
<security-domain name="hsm-proxy-application" cache-type="default">
  <authentication>
    <login-module code="be.fedict.hsm.model.security.ApplicationLoginModule"
      flag="required"/>
  </authentication>
</security-domain>
<security-domain name="hsm-proxy-administrator" cache-type="default">
  <authentication>
    <login-module
      code="be.fedict.hsm.model.security.AdministratorLoginModule"
      flag="required"/>
  </authentication>
</security-domain>
```

## 2.3. Database

The HSM Proxy requires a database to operate. The HSM Proxy product itself is database agnostic and should run on any database which provides proper JDBC drivers.

### 2.3.1. MySQL

This section described the setup required to use MySQL as underlying database.

First we need to create a dedicated database user.

```
mysql -u root -p
CREATE USER 'hsmproxy'@'localhost' IDENTIFIED BY 'hsmproxy';
GRANT ALL PRIVILEGES ON hsmproxy.* TO 'hsmproxy'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

Now we create the database via:

```
mysql -u hsmproxy -p
CREATE DATABASE hsmproxy;
EXIT;
```

Next we need to configure the JDBC module within the application server.

First download the latest MySQL JDBC driver and unzip under your home directory:

```
unzip mysql-connector-java-5.1.25.zip
```

Create under `$JBASS_HOME/modules/` (JBoss AS 7.1.3.Final) or `$JBASS_HOME/modules/system/layers/base/` (JBoss AS 7.2.0.Final, or JBoss EAP 6.x):

```
mkdir -p com/mysql/main
cp ~/mysql-connector-java-5.1.25/mysql-connector-java-5.1.25-bin.jar com/mysql/main
vim com/mysql/main/module.xml
```

And put in `module.xml` the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.1" name="com.mysql">
  <resources>
    <resource-root path="mysql-connector-java-5.1.25-bin.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
  </dependencies>
</module>
```

Finally we configure `standalone.xml` as follows.

Under `<subsystem xmlns="urn:jboss:domain:datasources:1.1">` we add a new `<driver>` under `<drivers>`.

```
<driver name="com.mysql" module="com.mysql">
  <xa-datasource-class>
    com.mysql.jdbc.jdbc2.optional.MysqlXADataSource
  </xa-datasource-class>
</driver>
```

Under `<datasources>` we add a new `<datasource>` .

```
<datasource jndi-name="java:jboss/datasources/HSMProxyDS"
  pool-name="HSMProxyDS" enabled="true">
  <connection-url>jdbc:mysql://localhost:3306/hsmproxy</connection-url>
  <driver>com.mysql</driver>
  <transaction-isolation>TRANSACTION_READ_COMMITTED</transaction-isolation>
  <pool>
    <min-pool-size>10</min-pool-size>
    <max-pool-size>100</max-pool-size>
    <prefill>true</prefill>
  </pool>
  <security>
    <user-name>hsmproxy</user-name>
    <password>hsmproxy</password>
  </security>
  <statement>
    <prepared-statement-cache-size>32</prepared-statement-cache-size>
    <share-prepared-statements>true</share-prepared-statements>
  </statement>
</datasource>
```

Per default the MySQL database operates using the non-transactional MyISAM database engine. On the JBoss Application Server we use XA transactions to be able to cover transactions involving both the database and JMS. So we need to tune the MySQL database to use the InnoDB database engine. Check whether the MySQL server supports the InnoDB database engine via:

```
mysql -p
SHOW ENGINES;
QUIT;
```

In `/etc/my.cnf` add the following line:

```
[mysqld]
default-storage-engine=innodb
```

Restart the MySQL server via:

```
service mysqld restart
```

Now you can restart the application server. After deploying the HSM Proxy application, you can check the SQL database schema via tools like Squirrel SQL.



### Database password

In the above configuration we fix the database password to `hsmproxy`. Of course, you should change the database password for production deployments.

Check whether existing database tables use the InnoDB database engine via:

```
mysql -u hsmproxy -p hsmproxy
SELECT table_name,Engine FROM information_schema.tables WHERE table_schema =
'hsmproxy';
EXIT;
```

If required, you can convert existing database tables to the InnoDB database engine via:

```
ALTER TABLE table_name ENGINE=InnoDB;
```

## 2.4. AJP Connector

Enable the AJP connector by adding the following under `<subsystem xmlns="urn:jboss:domain:web:1.4">`:

```
<connector name="ajp" protocol="AJP/1.3" scheme="https" socket-binding="ajp"/>
```

## 2.5. Starting

Start the Application Server via:

```
./standalone.sh --server-config=standalone.xml
```

## 2.6. Deploying

Finally you can deploy the HSM Proxy application via:

```
cd $JBOSS_HOME/bin
./jboss-cli.sh
connect
deploy /path/to/your/hsm-proxy-deploy-0.2.0-production.ear
exit
```

## 2.7. Undeploying

Before upgrading the HSM Proxy application you have to undeploy the previous version. Undeploying the application can be done via:

```
cd $JBOSS_HOME/bin
./jboss-cli.sh
connect
cd deployment
ls
undeploy hsm-proxy-deploy-0.2.0-production.ear
ls
exit
```

## 3. Apache

The front-end Apache web server should be configured via `/etc/httpd/conf.d/hsm-proxy.conf`:

```
<Location "/hsm-proxy-admin">
  ProxyPass ajp://localhost:8009/hsm-proxy-admin
</Location>
<Location "/hsm-proxy-ws">
  ProxyPass ajp://localhost:8009/hsm-proxy-ws
</Location>
```

Gracefully restart the Apache web server via:

```
service httpd graceful
```



### Web Service Endpoint

Although the web service endpoint has been secured using WS-Security, please keep in mind that by making the web service endpoint available via Apache you expose yourself to possible attacks. It is advised to limit the exposure of the web service endpoint to the public internet if possible.

## 4. Compiling JBoss Application Server

First we need to checkout the source code from GIT.

```
git clone https://github.com/wildfly/wildfly.git
```

Next we build version 7.2.0.Final via:

```
cd wildfly/  
git checkout 7.2.0.Final  
mvn clean install -Drelease=true -Dmaven.test.skip=true
```

In case you work behind an HTTP proxy you might want to compile via:

```
mvn      clean      install      -Drelease=true      -Dmaven.test.skip=true      -  
Dhttp.proxyHost=proxy.yourict.net -Dhttp.proxyPort=8080
```

The ZIP distribution is located under `dist/target`.

## A. HSM Proxy Installation Manual License



This document has been released under the [Creative Commons 3.0](http://creativecommons.org/licenses/by-nc-nd/3.0/) [http://creativecommons.org/licenses/by-nc-nd/3.0/] license.

## B. HSM Proxy Project License

The HSM Proxy Project source code has been released under the GNU LGPL version 3.0.

This is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License version 3.0 as published by the Free Software Foundation.

This software is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this software; if not, see <http://www.gnu.org/licenses/>.