

This is a data repository that could subscribe to a node and archive all event messages.

The data repository is made of the mysql database hosted at a different machine than the Openfire XMPP server. The repository saves any message that is sent to this node, along with the time stamp.

## **Setting up and testing the prototype implementation of Repository service**

### **Installation**

These components are necessary to run:

- Openfire XMPP server (<http://www.igniterealtime.org/projects/openfire/>). Download and install the latest version.
- Repository software

### **Preliminary**

```
$ export IMF_HOME=$HOME/IMF
$ mkdir $IMF_HOME
```

Install java 1.6 sdk, recent versions of Maven and Ant. Be sure java, mvn and ant executables are on your path.

On geni-imf-dev host append the following to your .bashrc file:

```
export JAVA_HOME=/opt/java/jdk1.6.0_18
export ANT_HOME=/opt/java/apache-ant-1.8.0
export MAVEN_HOME=/opt/java/apache-maven-2.2.1
export PATH=$JAVA_HOME/bin:$ANT_HOME/bin:$MAVEN_HOME/bin:$PATH
```

### **Setting up Openfire**

Download and setup Openfire according to instructions. Notice, it is already running on [geni-imf-dev.renci.org](http://geni-imf-dev.renci.org), so this step can be skipped. It is setup to use GENI certificates and credentials.

## **Setting up the database**

To give remote access to [geni-imf-dev.renci.org](http://geni-imf-dev.renci.org) to the machine that is hosting the repository's mysql database

Login as root

install mysql client and server.

```
$ yum install mysql mysql-server
```

Start the MySQL server:

```
$ /sbin/service mysqld start
$ mysql -uroot -ppassword
```

Create a database on the sql server.

```
mysql> create database Repository;
```

Create a table which has an automatic timestamp on each record

```
mysql> CREATE TABLE general_repo (id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
data VARCHAR(200),cur_timestamp TIMESTAMP(8) );
```

Grant privileges.

```
grant all on Repository.* to 'root'@'%.%.%.%';
```

```
grant all on *.* to 'root'@'152.14.3.37';
```

(152.14.3.37 was the IP address of geni-imf-dev.renci.org at the time of preparing this document)

You should add port 3306 to the “Other Ports” section of your Firewall Configuration, to make it accessible for all hosts or networks using either tcp/udp protocol for mysql service.

### **Setting up the repository service**

Check out the code out of the repository :

```
$ cd $IMF_HOME  
$ svn co https://geni-imf.renci.org/svn/Repository/ imf-repository-service  
$ cd $IMF_HOME/imf-repository-service/src/psm  
$ mvn install assembly:assembly
```

Update the IP address of machine, that is hosting the repository's mysql database, in the main method of /home/akgrewal/IMF/imf-repository-service/src/psm/src/main/java/org/renci/geni\_imf/psm/SimpleSubscriber.java file.

```
String url = "jdbc:mysql://152.14.217.62/Repository";
```

(152.14.217.62 is the IP address of machine that is hosting the repository's mysql database)

Repeat the build attempt:

```
$ mvn install assembly:assembly
```

Download Connector/J from [here](#)

```
$ tar -zxvf mysql-connector-java-5.1.18.tar.gz
```

Specify class path entries to help Java interpreter find MySQL Connector/J (downloaded above) and PSM in the directories. (Please note that the location of your MySQL Connector's jar may be different.)

```
$ export set  
CLASSPATH=/home/akgrewal/IMF/jdbc-connection/mysql-connector-java-5.1.18-bin.jar:$CLASSPATH
```

```
$ export set CLASSPATH=$IMF_HOME:target/psm-0.1-jar-with-dependencies.jar:$CLASSPATH
```

### **Authorization**

The repository will require “can\_sub” credentials to the node namespace Repository.

Instructions on how to generate the .jks file are as follows:

1. Obtain cert-keypairs by emailing to Anirban Mandal ( anirban AT renci.org ) what is the node namespace for which you need pub/sub credentials.

(For steps, 2 & 3 you may refer to [Creating Flukes keystore.mov.](#))

2. Creating a pkcs12 file with your cert and key:

Eg:

```
openssl pkcs12 -export -in ashu-cert.pem -inkey ashu-key.pem -out ashu-gcf-encrypted.p12
```

Remember the password you enter in this step. You will need that later in step 3.d Some clients just need this .p12 file and this password to log in.

3. Importing cert-keypair to java keystore.

Use Portecle: <http://portecle.sourceforge.net/>

Click Launch.

- Create a new keystore of type JKS
- Click on "Import Key Pair"
- Locate the .p12 file (ashu-gcf-encrypted.p12 in the example)
- Enter the password protecting the .p12 file
- Import the key and certificate, but change the key alias to something simpler than a GUID
- Enter and confirm a new key password (I usually give the same password as the one protecting .p12 file)
- Click on "Save Keystore"
- Enter and confirm the new keystore password (use same as the key password)

This will result in the java keystore. Our repository service will need path to this keystore and the password above to log in.

create a subscriber.properties file in \$IMF\_HOME:

```
IMF.pubsub.server=geni-imf-dev.renci.org:5222
```

```
IMF.pubsub.usecertificate=true
```

```
MF.pubsub.login=<JID> --- the CN in the client certificate -- for eg. 74cf1400-a764-4ba1-af54-f0c52e9425f6
```

```
IMF.pubsub.password=<password protecting the jks file below>
```

```
IMF.pubsub.keystorepath=<path to java keystore with your cert-keypair> -- for eg. /home/akgrewal/Cert/ashutoshgrewal(s)-ben.jks
```

```
IMF.pubsub.keystoretype=jks
```

```
IMF.pubsub.truststorepath=<path to java keystore with your cert-keypair> -- for eg. /home/akgrewal/Cert/ashutoshgrewal(s)-ben.jks
```

```
IMF.pubsub.root=<node namespace> -- for eg. Repository
```

substituting all <parameters> with actual values. Save the file in \$IMF\_HOME - it must be on Java CLASSPATH to be available to PSM.

start the repository

```
$ java org.renci.geni_imf.psm.SimpleSubscriber
```

Click subscribe.

Note - for our implementation the modules need VPN access to each other's running machines