

Trust- \mathcal{X}

Installation and execution instructions

This document contains the instructions to install and run the new version of the Trust- \mathcal{X} framework.

1 System prerequisites

The prerequisites are:

JavaSDK version 1.6 or greater;

Apache ant version 1.7 or later;

MySQL version 5.0 or later or

SQLite version 3.5 or later.

2 Database installation

The new version of the Trust- \mathcal{X} framework both support MySQL and SQLite database.

2.1 SQLite database

Test SQLite databases are ready to be used in the `Database/` directory. They are the files with the `.db` suffix.

2.2 MySQL database

To use the MySQL database:

- Install the database server;
- Create a database using the MySQLAdministrator GUI or the command line;
- Load in the just created database one of the dump file located in the `Database/` directory such as:
 - `damp.sql`

- DumpTrustx2.nuovo
 - DumpTrustx2_group.nuovo
 - DumpTrustx2
 - DumpTrustx2_group
- Create a user with at least the SELECT privilege on the database created at step 2;

3 Software execution

The decompression of the archive `TrustX-AEOLUS.tar.bz2` will create a directory with name `TrustX-AEOLUS`. Enter such directory using a terminal (on a *nix system) or the command line (on Windows).

Type in the command `ant -p` to list the available execution target.

```

stefano.braghin@heth:~/public_html -- bash -- 93x39
sleipnir:TrustX-AEOLUS stefano$ ant -p
Buildfile: build.xml

Main targets:

build          Compile the classes composing the project
clean         Delete all classes and jar(s)
doc           Generate the JavaDoc for the current project
groupClient   Run the group client, MySQL and IP
groupClient2  Run the group client, MySQL and IP
groupClientJ  Run the group client, MySQL and JXTA
groupHandler  Starts the GroupHandler
groupServer   Run the group server, MySQL and IP
groupServer2  Run the group server, MySQL and IP
groupServerJ  Run the group server, MySQL and JXTA
jar           JARs the task
launchPeer   Launchs a set K peers
shell        Run the AEOLUS Jxta Shell
testClient   Run the client with MySQL and IPv4
testClient2  Run the client with SQLite and IP
testClientL  Run the client with SQLite and IP
testGUI      Run the test for the GUI
testGroupClient Run the group agent
testGroupServer Test the group manager
testServer   Run the server with MySQL and IP
testServer2  Run the server with SQLite and IP
testServerL  Run the server with SQLite and IP
Default target: build
sleipnir:TrustX-AEOLUS stefano$

```

Figure 1: The output of the execution of `ant -p`

As an example, run in a terminal **ant testServer2** to execute a server instance of TrustX and run in another terminal **ant testServer2** to execute a client instance.

The server will open a IPv4 connection listening on port 4444, the client will connect to localhost:4444. Both will use Database/db1.db as database.

The targets **testServer** and **testClient** act as the “2” counterpart but use a MySQL database.

To modify such behavior open the file build.xml with a text editor such as notepad or emacs.

The arguments of a server target are the following:

- Listening protocol and port, specified with an URI of the form:

```
<protocol>://:<port>
```

The available protocols are ip and ssl and <port> may be any legal port number;

- Database, specified with an URI of one of the following form:

```
mysql://<username>:<password>@<database host>/<database name>
```

```
sqlite:<path to the database>
```

<database host> can both be an hostname and an IP address. it does not allow the specification of the connection port, hence the MySQL server must be listening on the default port (3306). <path to the database> can both be absolute or relative.

```
<target name="testServer" description="Run the server" depends="jar" >
  <java fork="true" classname="trustx.TrustX">
    <arg value="ip://:4444" />
    <arg value="mysql://TrustX2:TrustX2@localhost/TrustX2" />
    <classpath>
      <path refid="classpath" />
      <path location="{jar.name}" />
    </classpath>
  </java>
</target>
```

Figure 2: The **testServer** target

The arguments of a client target are:

- Connection protocol and port, specified with an URI of the form:

`<protocol>://<hostname>:<port>`

Protocol and port numbers follow the same restriction of a server target. `<hostname>` may both be an IP or a hostname;

- Database, specified as in the server target;
- Resource to negotiate. The third argument of a client target is the initially requested resource. It has to be available in the server or the negotiation will end immediately without success. The resource is specified using an expression of the following form:

`ResourceName(Attr1 pred1 Val1, Attr2 pred2 Val2, ..., AttrN predN ValN)`

Where each `predi` is one of {`<`,`<=`,`=`,`=>`,`>`}. If no attribute is specified the parenthesis should be omitted, as in the example of Figure 3.

```
<target name="testClient" description="Run the client" depends="jar" >
  <java fork="true" classname="trustx.TrustX">
    <arg value="ip://localhost:4444" />
    <arg value="mysql://TrustX2:TrustX2@localhost/TrustX2" />
    <arg value="A" />
    <classpath>
      <path refid="classpath" />
      <path location="{jar.name}" />
    </classpath>
  </java>
</target>
```

Figure 3: The `testClient` target

Finally, the Trust- \mathcal{X} framework can be executed within the JXTA shell typing the command `ant shell`. To create a group, a credential with name “TXM” and with attribute “name” set to the name of the group has to be inserted in the table `peer_resources` of the database used by the authentication service.