What is TAP

The aim of the Thredds Admin Portal (TAP) project lies in the idea of solving the huge problem existing in the Unidata Thredds application with the management of users, roles and dataset access. The Thredds user authentication is delegated to Tomcat Basic which gets users and roles from a given Realm. In order to authorize users, Thredds checks whether the dataset is restricted and if so, gets the user roles to make a decision. TAP manages the datasource mentioned to control user access to protected datasets.

Derby datasource setup

TAP and Thredds get users and roles from a database created specifically to model all the entities and processes involved. The main idea is to include easily a database instance to work with. Derby embedded was discarded because it is not possible to access it in a production server from another JVM simultaneously. Derby network allow users to access db instances even from outside when they are running. To accomplish this step you need two components:

- Derby 10.10.1.1 library
- Derby database

First, place the db-derby-10.10.1.1-bin library provided in a reachable folder. Second, place the preconfigured database. We suggest to include the derbydb folder provided in the Tomcat?s content folder. For example, CATALINA_HOME/content/tap/derbydb. The Derby database must be initialized in the Tomcat startup. Execute the following command to initialize it:

```
$JRE_HOME/bin/java -jar $PATH_TO_DERBY_LIB /db-derby-10.10.1.1-bin/lib/derbyrun.jar server start -p YOUR_PORT -h 0.0.0.0 &
```

To start derby successfully add a socket permission in JAVA 7 by including in $JRE_HOME/lib/security/java.policy the following line:

```
permission java.net.SocketPermission "HOST:PORT", "listen,resolve";
```

Expose our datasource in Tomcat

Tomcat gets users and roles from conf/tomcat-users.xml by default. We are going to change this in order to get users and roles from a given database. First of all, include both derbyclient.jar and derbynet.jar in $CATALINA_HOME/lib. After doing that, we need to add a new resource called ?jdbc/admin? in GlobalNamingResurces?:

```
<Resource name="jdbc/adminDB" auth="Container" type="javax.sql.DataSource" factory="org.apache.tomcat.dbcp.dbcp.BasicDataS
   validationQuery="SELECT count(*) FROM users" maxActive="20" maxIdle="10" username="password" password="secret"
   driverClassName="org.apache.derby.jdbc.ClientDriver" url="jdbc:derby://host:port/derbypath/derbydb" readOnly="false"/>
```

Set the realm

A Realm is a ?database? of usernames and passwords that identify valid users of a web application (or set of web applications), plus an enumeration of the list of roles associated with each valid user. The servlet container will be connected to the database and it also be aware of the username and the corresponding roles. Define this realm inside <Engine> in your server.xml:

```
<Realm className="org.apache.catalina.realm.DataSourceRealm" digest="MD5" debug="0" dataSourceName="jdbc/adminDB"
   userTable="USERS" userNameCol="USERNAME" userCredCol="PASSWORD" userRoleTable="V_USERS_ROLES" roleNameCol="ROLENAME"/>
```

Thredds and TAP deployment

The last part of this tutorial is manage to start both applications successfully. Move both thredds.war and tap.war to $CATALINA_HOME/webapps. Start the Tomcat instance and the Derby network instance. If Thredds is not able to start due to a directory error, please create a folder called ?thredds? in $CATALINA_HOME/content to solve that issue.

Thredds restricted dataset

Thredds allows users to restrict dataset access in two different ways:

- URL restriction using Tomcat: difficult to maintain. You need to restrict every dataset by URL and set the role in the web.xml
- Dataset restriction using TDS Catalog: most commonly used by adding an attribute on a dataset or datasetScan element in the TDS catalog. Eg, restrictAccess=?roleName?

If you set the mentioned attribute in a Dataset, users need two roles to access it: restrictedDatasetUser and roleName. That means, every user who wants to access to a dataset needs the restrictedDatasetUser role by default and also the role of the dataset. Here is where TAP does the job for you.

Initial TAP setup