



IA2TAP Table Access Protocol service at IA2

Document No.	IA2TAP
Issue/Rev. No.	1.0
Date	Mar 25, 2017
Page	1 of 6

Document No.: IA2TAP
Issue/Rev. No.: 1.0
Date: March 25, 2017
Authors: Robert Butora, Marco Molinaro
Approved by: Marco Molinaro, Riccardo Smareglia

Contents

1 Acknowledgements	1
2 System overview	1
2.1 Prerequisites	1
3 Configure and install with scripts	2
4 Configure and install manually	2
4.1 Set up access rights to databases	3
4.2 TAP Schema Manager (TSM)	3
4.2.1 TSM install	3
4.2.2 TSM initial access	4
4.3 Connection to databases	4
4.4 IA2TAP	4
4.4.1 Configure	5
4.4.2 Deploy and run	6

1 Acknowledgements

This work has received funding from the European Community’s Seventh Framework Programme (FP7–SPACE–2013–1) under grant agreement number 606740.

2 System overview

We assume a user has a set of catalogues which he wants to make remotely available from his computer. The catalogues are in form of tables in a relational database. For instance, let "CatDB1", "CatDB2", etc... be the names of the databases where the catalogues reside. We will expose those tables with the help of the "IA2TAP" service. We assume only one computer is used and so all needed software (MySQL, Java, Glassfish) is running on the same machine. We will add "IA2TAP", the service itself and the "TAP Schema Manager" which allows to configure external visibility of the CatDB’s. One can simple expose all databases, or parts of them by "hiding" some tables and/or columns.

IA2TAP keeps its own configuration data in a database. It uses "Connection Pool" technology to reach this configuration database. The Connection Pools are offered (and set up) in Glassfish. To reach its configuration data, IA2TAP expects a Connection Pool named **jdbc/ia2tapConfig**, so it must exist.

IA2TAP uses a second Connection Pool to reach the CatDB’s themselves and the TAP-schema database which describes them. This TAP-schema database must reside on the same SQL server as the target database(s). Its name is defined in **ia2tap.config** table **tap_schema** field, so it can be freely chosen (see Section 4.2.2). And the connection pool to reach it, is given in in the table **ia2tap.config**, field **jdbc_resource**.

2.1 Prerequisites

It is assumed that the computer has already installed and is running:

- **MySQL:** server version 5.5.x with CatDB’s created in it.



IA2TAP Table Access Protocol service at IA2

Document No.	IA2TAP
Issue/Rev. No.	1.0
Date	Mar 25, 2017
Page	2 of 6

- **Java:** JRE version 1.8
- **Glassfish:** version 4.x (make sure your network's firewall allows external access to 8080 port)
- **MySQL connector:** Glassfish supports connections to various databases. We need driver for the MySQL database¹.

And you have on hand the following packages:

- **TSM:** Table-Schema Manager install-package, version 1.0.3: `tsm-1.0.3-glassfish.zip`
- **IA2TAP:** deployment package, version 1.2.0: `IA2TAP-1.2.0.war`
- **SQL scripts:** two scripts needed to configure the IA2TAP service: `ia2tap*.sql`

Also make sure that the Glassfish administration command `asadmin` and the SQL-server client `mysql`, as well as `jar` command from JDK can be executed from the shell, and so path to their locations are listed in your `PATH` variable of the shell environment. Additionally `zip` command is also used, so `zip` package should be installed.

3 Configure and install with scripts

There are three scripts supplied, which perform all commands in the section describing manual installation. The scripts are

- **tsm-setup:** configures and deploys TSM
- **connpool-setup:** creates proper connection pools in Glassfish
- **ia2tap-setup:** configures and deploys IA2TAP service itself.

The TSM setup script sets `ia2tap` as password. This password is asked as first, whenever accessing TSM. You can change it later as described in Section 4.2.1. When running TSM the first time, you have to choose the name of the TSM database. These setup scripts assume `catdb_tap_schema` as `TAP_SCHEMA` name, so that name must be entered.

Additionally, the scripts need MySQL user with sufficient rights (see Section 4.1) to create database and insert data, and a Glassfish administrator account to be able to deploy applications.

If all prerequisites on place, it should be enough to run

```
connpool-setup.sh <dbUserName> <dbUserPwd> <glassfishAdminName> <glassfishAdminPwd>
tsm-setup.sh      <dbUserName> <dbUserPwd> <glassfishAdminName> <glassfishAdminPwd>
ia2tap-setup-configDB.sh <dbUserName> <dbUserPwd>
ia2tap-setup.sh   <glassfishAdminName> <glassfishAdminPwd> <remoteUserName> <remoteUserPwd>
```

where `<remoteUserName>` `<remoteUserPwd>` are user name and password of your choice. It will be used by remote client when accessing the exposed CatDB's via IA2TAP.

4 Configure and install manually

We need to install TAP Schema Manager (TSM) and set up CatDB visibility. Then we need to install and configure the IA2TAP service itself.

¹Can be downloaded from MySQL web-site. At the time of writing this document it was: `mysql-connector-java-5.1.40-bin.jar` It needs to be copied into a folder where Glassfish can find it. For instance `<glassfish root>/glassfish/domains/domain1/lib/ext`.



IA2TAP Table Access Protocol service at IA2

Document No.	IA2TAP
Issue/Rev. No.	1.0
Date	Mar 25, 2017
Page	3 of 6

4.1 Set up access rights to databases

The database server where CatDB's reside must allow access from TSM and IA2TAP. Access is associated with a database user, which must have sufficient grants to be able to create and modify tables.

Set correct GRANTS for TSM to access (CREATE,WRITE) the database server. The user and the grants can be added with following commands. For a TSM running on a computer with IP number AAA.BBB.CCC.DDD create user called "uia2tap" with password "pia2tap" and set privileges:

```
mysql> CREATE USER 'uia2tap'@'AAA.BBB.CCC.DDD' IDENTIFIED BY 'pia2tap';
mysql> GRANT ALL PRIVILEGES ON *.* TO 'uia2tap'@'AAA.BBB.CCC.DDD';
```

Note: make sure that SQL GRANT-precedence rules for that user do not hide the rule above. If user with password already exists skip the first command.

4.2 TAP Schema Manager (TSM)

The TSM allows to create and edit TAP schema database with a graphical interface. The TAP schema database will define what portion of CatDB's will be visible to external users. You can expose all the CatDB database or only portions of it, filtering out whole database(s), or only some tables from a database, or only some columns of a table. The TSM does not support filtering out rows.

The TSM is build as a web application and so runs under Glassfish.

4.2.1 TSM install

To install the TSM, take the distribution package `tsm-1.0.3-glassfish.zip` and unzip to some directory. It contains: `tsm-1.0.3.war` the applications itself, `webapp.properties` the configuration file and a script `update-war.sh` which updates the configuration file into the war file.

Two credentials need to be configured inside the `webapp.properties`: `password=<tsmpassword>` is the password to login to a deployed TSM. The other, are the credentials to access the CatDB databases and these are defined in a file: `credentials_config_path=/home/<yourusername>/.tsm/config.xml`. This configuration file can be created and edited, it is read each time TSM is started:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<credentials-config>
  <credentials database-type="MYSQL" hostname="localhost" password="pia2tap"
  port="3306" username="uia2tap"/>
</credentials-config>
```

After the `config.xml` and `webapp.properties` were created and edited, run the update script and deploy TSM to Glassfish:

```
./update-war.sh
asadmin --user <adminuser> --passwordfile <filename> deploy --contextroot tsm tsm-1.0.3.war
```

where `<filename>` contains one line `AS_ADMIN_PASSWORD=<adminpassword>`.

The `update-war.sh` essentially performs:

```
zip -u tsm-1.0.3.war WEB-INF/classes/webapp.properties
```

Note that the sql-user `uia2tap` needs to have sufficient GRANTS to CREATE, INSERT and read database tables.



IA2TAP Table Access Protocol service at IA2

Document No.	IA2TAP
Issue/Rev. No.	1.0
Date	Mar 25, 2017
Page	4 of 6

4.2.2 TSM initial access

After this the TSM should be accessible from a web browser as:

`http://localhost:8080/tsm`

When prompted for password, supply the password `tsmpassword` specified in the previous section. Then TSM asks you to choose a database name where it stores its own data, for instance `catdb_tap_schema`. After that, the database will be created. Then the TSM would present a list of databases on the server specified in the `~/.tsm/config.xml` file, among them also the CatDB's. Choose the CatDB's and the `catdb_tap_schema` will be initialized. Now, one can select visibility of the database tables and columns.

Usage of the TSM is described in detail in its user manual². These settings can be changed anytime later, also while the system is running.

4.3 Connection to databases

The service uses Connection Pools to access the databases. There are two connection pools needed: one to reach the CatDB's together with the TSM's database to control visibility, and an other database, which stores the configuration of the running IA2TAP service itself.

So two Connection Pools are needed:

- `jdbc/ia2tapConfig`: a pool to connect to IA2TAP configuration data
- `jdbc/catalogues`: a pool to reach CatDB's and the related TSM database

Note that `jdbc/ia2tapConfig` is hardcoded so must exist. The `jdbc/catalogues` is set in IA2TAP configuration, so can be chosen.

First create the connection pool for configuration data and add its resource name:

```
asadmin create-jdbc-connection-pool
--datasourceclassname com.mysql.jdbc.jdbc2.optional.MysqlDataSource
--restype javax.sql.DataSource
--property user=uia2tap:password=pia2vlkb:url=
"jdbc:mysql://palantir19.oats.inaf.it:3306/ia2tapConfig" ia2tapConfig
asadmin create-jdbc-resource --connectionpoolid ia2tapConfig jdbc/ia2tapConfig
```

The resource name is needed for Glassfish to locate the configuration pool. Next, create the database access pool and add its resource name:

```
asadmin create-jdbc-connection-pool
--datasourceclassname com.mysql.jdbc.jdbc2.optional.MysqlDataSource
--restype javax.sql.DataSource --property user=uia2tap:password=pia2vlkb:url=
"jdbc:mysql://palantir19.oats.inaf.it:3306/catdb_tap_schema" catalogues
asadmin create-jdbc-resource --connectionpoolid catalogues jdbc/catalogues
```

To list existing Connection Pools in Glassfish run: `asadmin list-jdbc-connection-pools`. It should list the two pools created above.

4.4 IA2TAP

IA2TAP is a web-service and so it is distributed as a standard war-file: `IA2TAP-1.2.0.war` and can be directly deployed to Glassfish.

²TSM user manual is provided alongside this package or can be retrived at IA2: <https://owncloud.ia2.inaf.it/index.php/KT7onaTbs0vRyD1>. This package refers to version 1.0.1 of the TSM manual



IA2TAP Table Access Protocol service at IA2

Document No.	IA2TAP
Issue/Rev. No.	1.0
Date	Mar 25, 2017
Page	5 of 6

4.4.1 Configure

The service configuration consists of specifying the following content and parameters:

- **context root:** URL-path how to reach the catalogues from remote client
- **access rights:** user name and password to use from the remote client
- **connection to database:** a pool to reach CatDB's and related TSM database (jdbc/catalogues in previous section 4.3)
- **TSM database name:** database name created in Section 4.2.2 above, which sets visibility of CatDB's
- **paths:** decide where TAP will store temporary data and results
- **logging:** give path where IA2TAP puts logs

As mentioned in Section 4.3, the configuration values of IA2TAP are stored in a database reachable by jdbc/ia2tapConfig connection. We will create and fill in that database.

Let's fill in the configuration tables inside the ia2tap-config.sql script.

First added the SQL command which inserts data into Config table in ia2tap-config.sql file:

```
INSERT INTO config (result_path, result_url, jdbc_resource,  
                   tap_id, tap_schema, context_root,  
                   data_model, data_model_ivo_id, public_database,  
                   log4j_properties_path)  
VALUES ('/home/<yourusername>/TAP', 'http://localhost:8080',  
        'jdbc/catalogues', 'catdb_tap_id', 'catdb_tap_schema', '/IA2TAP',  
        NULL, NULL,  
        1, 'not-used');
```

Next decide IA2TAP user and password, which will be used when accessing this service from the remote client. This is done into the ia2tap-shiro.ini configuration file available inside the package.

The ia2tap-shiro.ini content can also be used to connect to different authentication solutions, like an LDAP database of users.

Next, login to SQL-server and create a database for IA2TAP configurations.

```
CREATE DATABASE ia2tapConfig;
```

Setup and configurations are done through the ia2tap-setup.sh and ia2tap-setup-configDB.sh scripts that include also the configuration of the directories needed as specified in paths and logging. The directories must be writable by the service (namely the user used to run glassfish).

It is useful to place them under the home directory where Glassfish is running. Change to glassfish' home directory and run commands:

```
mkdir TAP  
mkdir TAP/log4j-properties  
mkdir TAP/catdb_tap_id  
mkdir TAP/catdb_tap_id/results
```

So the full tree looks like:

```
TAP  
|---log4j-properties  
|---catdb_tap_id  
    |--- results
```

where catdb_tap_id is the tap_id given in the configuration file above.
Configuration of the service's logging capability happens in file :



IA2TAP Table Access Protocol service at IA2

Document No.	IA2TAP
Issue/Rev. No.	1.0
Date	Mar 25, 2017
Page	6 of 6

TAP/log4j-properties/log4j.properties

Here we can specify the file where IA2TAP will write its log with the following keyword:

```
log4j.appender.FILE.File=/home/glassfish/TAP/ia2tap.log
```

4.4.2 Deploy and run

Then IA2TAP can be deployed with:

```
asadmin --contextroot IA2TAP IA2TAP-1.2.0.war
```

where we explicitly specify the context root so that it matches the one in the `ia2tap-config.sql` script in Section 4.4.1.

Then the running IA2TAP can be accessed by TOPCAT pointing to the URL:

```
http://localhost:8080/IA2TAP
```

When asked for user and password, provide those specified in Section 4.4.1.

Check that the access is possible from external network, to verify that no firewall is blocking the port 8080.