**TAP Server deployment and userguide**

**This document covers the details of installation of TAP web service for the**

**System Requirements:**  (Platform independant)

* Java 1.6 /+

http://www.oracle.com/technetwork/java/javase/downloads/index.html

* RabbitMQ

Install it according to the steps given on the website and start the broker service once installed.

http://www.rabbitmq.com/

* Web Server (Apache, Tomcat, glassfish)

**Project Specifications**:

1. tapwebservice.properties file (edit this file according to the requirements before deploying the service)
2. Depending on the requirement of SQL server, add the jdbc driver for the same in lib (in WEB-INF/lib).
3. Use ant command in project root folder to generate deployable war
4. Deploy it on Apache web server (or any suitable)
5. Before deploying, make sure the databases are available according to specifications in the properties file. And rabbitmq server instance is running.

**Databases and tables**:

System uses dedicated tables as listed below , a separate tapdatabase can be created which will host all tables below.

tapjobstable : It will hold all asynchronous jobs and related info TAP\_SCHEMA.schemas: schemas supported by this tap service TAP\_SCHEMA.tables: tables under the schema mentioned above TAP\_SCHEMA.columns: columns metadata available in above columns TAP\_SCHEMA.keys: tables which join by keys tapuserdata: this table holds all the information related to upload tables tapuserauth: this tables has all information about user authentication (SSO NVO)

***TAP\_data.sql***, is the schema for all above tables and it is available in Installation directory.

\*\*\*All the above tables should be in one database, not necessarily in the database for which TAP service will be used (for example: not in sdssdr8).

Once above tables are created , make sure to update TAP\_Schema tables with proper metadata information about the database (e.g metadata of all tables available in SDSS is inserted in TAP\_Schema).

**For uploadedtables**:

Create a new database (ex. tapuploads) which will hold the uploaded tables. This database should be either on the same server where main data(ex. SDSS DR8) is available or atleast a linked server. This is because the select query for the joins should run properly.

Access information can be given in properties file.

e.g In properties file following information is about the databases. Edit it according to your database information.

These examples use jtds driver for MSSQL.

*# Database related info, data accessed by tapservice*

*# eg: this is sdss data*

*database.UserName = skyuser*

*database.UserPassword = user123*

*database.Driver = net.sourceforge.jtds.jdbcx.JtdsDataSource*

*database.URL = jdbc:jtds:sqlserver://<database-server>:1433/dr8data*

*# storing job and related details*

*# This database has all tables related to async job and also TAP\_Schema*

*jobs.databaseuser = skytap*

*jobs.databasepassword = web123*

*jobs.databaseUrl = jdbc:jtds:sqlserver://<database-server>:1433/tapdata*

*# Store uploaded tables*

*upload.databaseuser = skytest*

*upload.databasepassword = skytest123*

*upload.databaseUrl = jdbc:jtds:sqlserver://<database-server>:1433/uploadTables*

Notice these examples above, there are 3 users

skyuser, skytap and skyupload

1. For users like ‘skyuser ‘ (which is user of main database for which tap service is written) permissions should be READ only.
2. For users like skyweb and skyupload ,read write permissions should be given
3. User for main db ‘skyuser’ should also be user (READ Only) of uploadtables database(e.g. uploadTables)

**Usage:**

Service supports all the following queries given with examples

There is simple UI developed in jsp, which can be accessible from index page of the service.

**Sample queries:**

1. Simple Query :: Select Top 10 p.objid, p.ra, p.dec, p.u, p.g, p.r, p.i, p.z From PhotoObj as p Where p.u Between 0 And 19.6
2. Spatial query :: SELECT o.ra, o.dec FROM photoobjAll o WHERE CONTAINS( POINT('J2000', o.ra, o.dec), CIRCLE('J2000', 180, 0, 0.3)) = 1

**Sample Commands:**

All the TAP commands or queries return results in the standard form specified in TAP V1.0 , IVOA recommendations.

**Using web interface**

Once service deployed use following link in browser

http://<webserver>/<webservice context >/

It will take to the index page of service where all the information to run the service

Using command line program to access the tap service

1. To list job details. curl http:// <webserver>/<webservice context>/ tap/async/<jobid>
2. To change the phase of the query curl -d PHASE=ABORTED http://<webservice context>/tapwebservice/tap/async/<jobid>/phase
3. To get the destruction time of job . curl http:// <webserver>/<webservice context>/tap/async/<jobid>/destruction
4. To change the destruction of the job curl -d DESTRUCTION="2011-07-06 16:19:19.283" http:// <webserver>/<webservice context>/tap/async/<jobid>/destruction
5. To Submit sync query curl -d REQUEST=doQuery -d LANG=ADQL -d QUERY="SELECT TOP 10 p.objid,p.ra,p.dec,p.u,p.g,p.r,p.i,p.z FROM PhotoObj as p WHERE p.u BETWEEN 0 AND 19.6" http:// <webserver>/<webservice context>/tap/sync
6. To terminate the query curl -d TERMINATION=2000 http:// <webserver>/<webservice context>/tap/async/<jobid>/executionDuration
7. To get the query execution time curl  [http:// <webserver>/<webservice context>/tap/async/<jobid>/executionDuration](http://localhost:8080/tapwebservice/tap/async/%3cjobid%3e/executionDuration)
8. To submit async query, it returns jobid created with link to job details. curl -d REQUEST=doQuery -d LANG=ADQL -d QUERY="SELECT TOP 1000 p.objid,p.ra,p.dec,p.u,p.g,p.r,p.i,p.z FROM PhotoObj as p WHERE p.u BETWEEN 0 AND 19.6"  [http:// <webserver>/<webservice context>/tap/async](http://localhost:8080/tapwebservice/tap/async)
9. To get the results of async job curl  [http:// <webserver>/<webservice context>/tap/async/<jobid>/results/result](http://localhost:8080/tapwebservice/tap/async/%3cjobid%3e/results/result)

**Using web service client java object:**

Here is an example using java-ws client.

public void testSync() {

WebResource webResource = com.sun.jersey.api.client.Client.create().resource("http://localhost:8080/tapwebservice/tap/sync");

MultivaluedMap formData = new MultivaluedMapImpl();

formData.add("REQUEST", "doQuery");

formData.add("LANG", "ADQL");

formData.add("QUERY", "SELECT TOP 10 ra FROM PhotoObjAll");

ClientResponse response = webResource.type("application/x-www-form-urlencoded").post(ClientResponse.class, formData);

//use response according to requirement

}

public void testAsync() {

WebResource webResource = com.sun.jersey.api.client.Client.create().resource("http://localhost:8080/tapwebservice/tap/async");

MultivaluedMap formData = new MultivaluedMapImpl();

formData.add("REQUEST", "doQuery");

formData.add("LANG", "ADQL");

formData.add("QUERY", "SELECT TOP 10 ra FROM PhotoObjAll");

ClientResponse response = webResource.type("application/x-www-form-urlencoded").post(ClientResponse.class, formData);

}

**VOSpace Integration and usage**:

There are two types of communication between VOSpace from TAP .

1. Pull Data From VOSpace into TAP
2. Pull Data into VOSpace from TAP

Here are details for usage of these functionalities

**VOSpace authentication:**

VAO vospace works with Single Sign On authentication. To use tap ‘UPLOAD’ or ‘STORERESULT’ parameters, service checks this authentication by checking the certificate file obtained by user while doing VAO SSO sign on.(for more details go to: ) Following are sample commands for these functionalities in TAP.

**1.Pull Data From VOSpace into TAP:**

According to TAP Specifications , VOSpace data can be uploaded in TAP . Following is the specification for the sample given below

It tells tap to upload the data from given VOSpace endpoint. Tap service resolves the “vos:” pattern for given vospace URL. eg**:** vos://edu.jhu!vospace

Sample command to upload data to TAP service on some pending job :

If you have not uploaded data while creating the job and if job is still in the pending phase UPLOAD parameter can be specified.

curl -d UPLOAD=table1,http://localhost:8080/WebSource/Sample1.xml; table2,vos://edu.jhu!vospace/data1/datanode1; https://localhost:8443/tapwebservice/tap/async/22e31a57-5edc-48b2-b586-4c49e09bfebd

Sample command to upload data while creating job:

curl -d REQUEST=doQuery -d LANG=ADQL -d QUERY="SELECT TOP 10 p.objid,p.ra,p.dec,p.u,p.g,p.r,p.i,p.z FROM PhotoObj as p WHERE p.u BETWEEN 0 AND 19.6" -d UPLOAD= table1,http://localhost:8080/WebSource/Sample1.xml; table2,vos://edu.jhu!vospace/data1/datanode1; <https://localhost:8443/sdss/tap/async> -E C:\geteec:deotappassphrase -k

**2.Pull Data into VOSpace from TAP**

Sample command to store TAP Query results in user’s vospace:

curl -d RESULTSTORE=VOSPACE <https://localhost:8443/sdss/tap/async/a057587f-92a6-4e94-82a2-72b91f88b150> -E C:\geteec:deotappassphrase -k

Here by default results are stored in the ‘tapresults’ directory in the user’s VOSpace. Otherwise there are other options which could be given along with the RESULTSTORE parameter in above example.

Parameters explanation in the above command:

*RESULTSTORE=VOSPACE* : to store results in vospace

[*https://localhost:8443/sdss/tap/async/a057587f-92a6-4e94-82a2-72b91f88b150*](https://localhost:8443/sdss/tap/async/a057587f-92a6-4e94-82a2-72b91f88b150) : this is job end point

*-E C:\geteec:deotappassphrase* : specify certificate file name and passphrase

*-k : curl specific command for key*

**More about vospace authentication and authorization:**

There are two ways to use signle sign on

1. Using OpenId

2. Using OAuth library

Currently TAP works with NVO SSO using certification option. This certificate , user has to provide once and then it will be used by service. TAP uses Single Sign On certificates generated by the NVO SSO portal, where is user can register once and get the certificates. These certificates are associated with the passphrase given by user. TAP service currently uses OAuth library.

First time user:

1. Register on NVO portal site. <https://sso.us-vo.org/>
2. Download certificate
3. Remember pass key associated with certificate

First time accessing secured TAP resources:

To upload the data in TAP and to store the TAP results in VOSpace, it goes through authentication. If one of these resources accessed by user for the first time, user will be asked to do following steps.

1. Get request token by providing certificate and passphrase to TAPservice

Curl https://<tapservice>/authentication -E <cetificatefile>:<passkey>

1. Get request token authorized from the service

Curl https://<vospace>/authorize\_token <cetificatefile>:<passkey> with token parameter

1. Once token is authorized send the request (upload or storeresult) to tap async job

Curl –d upload:tablename,tableurl https://<tapservice>/tap/async/<jobid> -E <cetificatefile>:<passkey>

1. For the first time Tap service will use authorized token to get a access token and store it in database.
2. For any further secure request to access VOSpace through TAP the access token stored in database is used.
3. For all the secure requests , certificate and passkey are mandatory.