New release of the CDS/ARI libraries

ADQL, UWS and TAP Libraries

Grégory Mantelet

Astronomisches Rechen Institut (ARI)
Heidelberg, Germany

17th June 2015

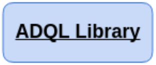
Table of contents

A. General reminder

3 *Java generic* libraries:

A. General reminder

3 *Java generic* libraries:



- ADQLLib:
  - Parse a query,
  - Build an AST,
  - Browse & Manipulate AST,
  - Translate into SQL *(for instance)*.

A. General reminder

3 *Java generic* libraries:

- ADQLLib:
  - Parse a query,
  - Build an AST,
  - Browse & Manipulate AST,
  - Translate into SQL *(for instance)*.
- UWSLib

A. General reminder

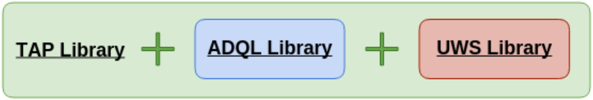
3 *Java generic* libraries:

TAP Library
+
ADQL Library
+
UWS Library

- ADQLLib:
  - Parse a query,
  - Build an AST,
  - Browse & Manipulate AST,
  - Translate into SQL *(for instance)*.
- UWSLib & TAPLib: Java frameworks to build resp. a customizable UWS and TAP service using the Servlet API.

A. General reminder

3 *Java generic* libraries:



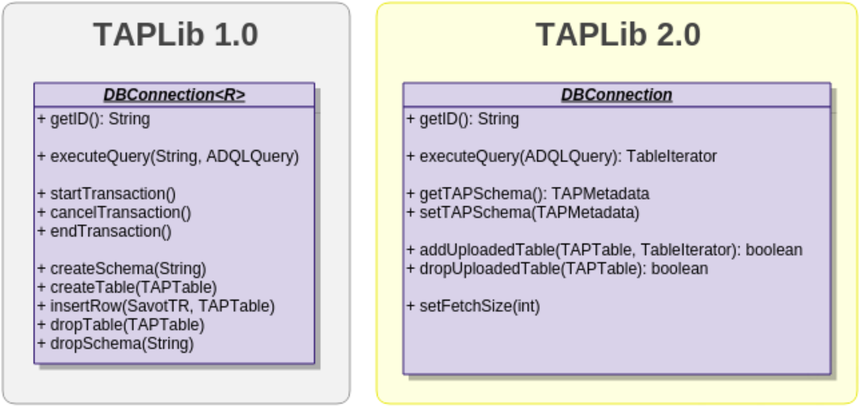
- ADQLLib:
  - Parse a query,
  - Build an AST,
  - Browse & Manipulate AST,
  - Translate into SQL *(for instance)*.
- UWSLib & TAPLib: Java frameworks to build resp. a customizable UWS and TAP service using the Servlet API.

http://cdsportal.u-strasbg.fr/adqltuto, /uwstuto, /taptuto

B. Libraries updates

1. Interaction with database

- Interface DBConnection simplified and more generic


TAPLib 1.0

DBConnection<R>

+ getID(): String

+ executeQuery(String, ADQLQuery)

+ startTransaction()
+ cancelTransaction()
+ endTransaction()

+ createSchema(String)
+ createTable(TAPTable)
+ insertRow(SavotTR, TAPTable)
+ dropTable(TAPTable)
+ dropSchema(String)

TAPLib 2.0

DBConnection

+ getID(): String

+ executeQuery(ADQLQuery): TableIterator

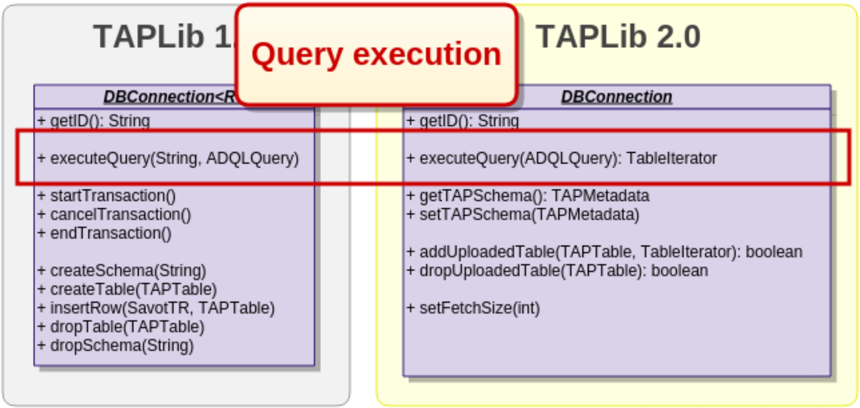
+ getTAPSchema(): TAPMetadata
+ setTAPSchema(TAPMetadata)

+ addUploadedTable(TAPTable, TableIterator): boolean
+ dropUploadedTable(TAPTable): boolean

+ setFetchSize(int)

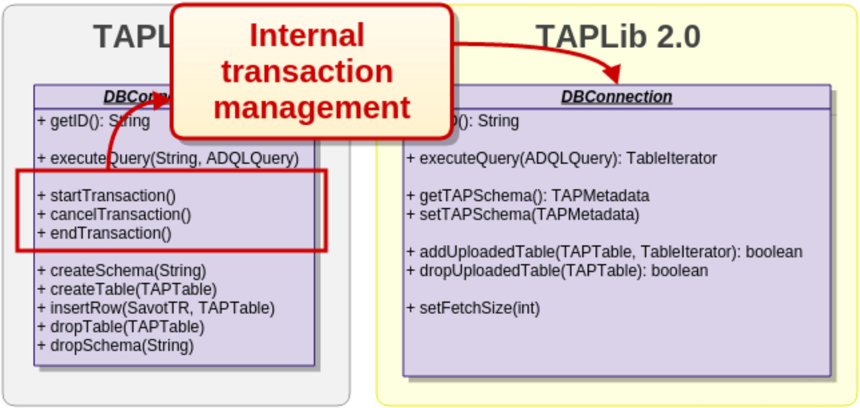
1. Interaction with database

- Interface DBConnection simplified and more generic



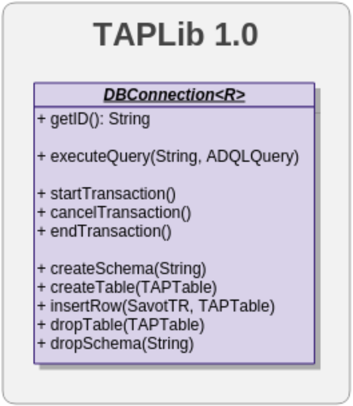
1. Interaction with database

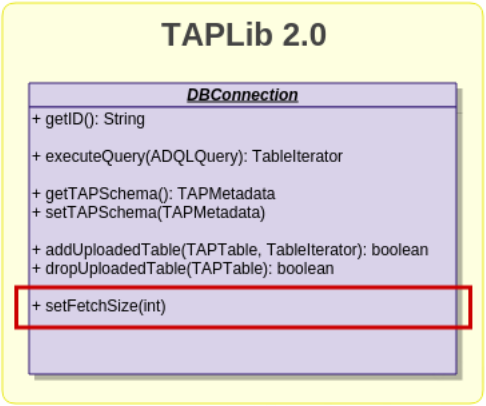
- Interface DBConnection simplified and more generic


TAPL
DBConn
+ getID(): String
+ executeQuery(String, ADQLQuery)
+ startTransaction()
+ cancelTransaction()
+ endTransaction()
+ createSchema(String)
+ createTable(TAPTable)
+ insertRow(SavotTR, TAPTable)
+ dropTable(TAPTable)
+ dropSchema(String)
Internal transaction management
TAPLib 2.0
DBConnection
D(): String
+ executeQuery(ADQLQuery): TableIterator
+ getTAPSchema(): TAPMetadata
+ setTAPSchema(TAPMetadata)
+ addUploadedTable(TAPTable, TableIterator): boolean
+ dropUploadedTable(TAPTable): boolean
+ setFetchSize(int)

1. Interaction with database

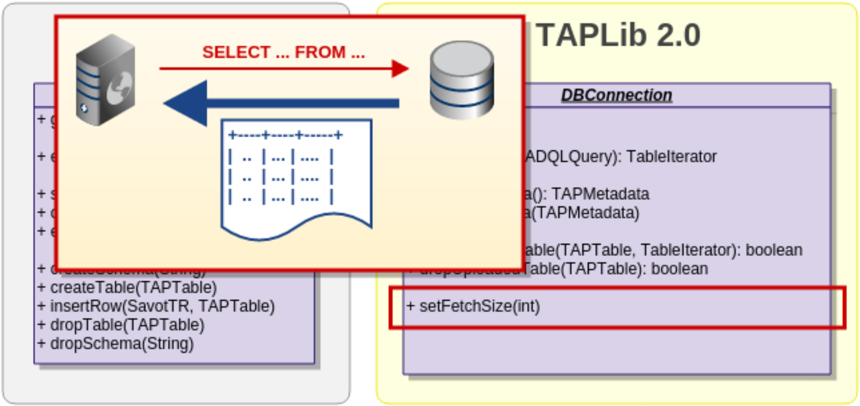
- Interface DBConnection simplified and more generic
- Fetch size customizable




TAPLib 2.0
DBConnection
+ getID(): String
+ executeQuery(ADQLQuery): TableIterator
+ getTAPSchema(): TAPMetadata
+ setTAPSchema(TAPMetadata)
+ addUploadedTable(TAPTable, TableIterator): boolean
+ dropUploadedTable(TAPTable): boolean
+ setFetchSize(int)

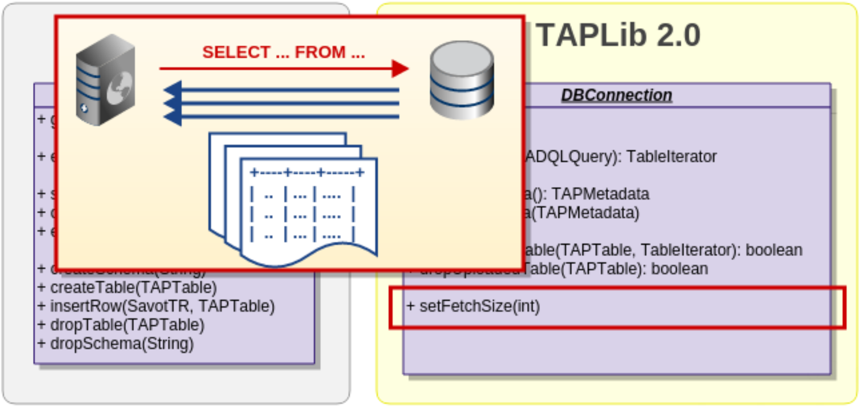
1. Interaction with database

- Interface DBConnection simplified and more generic
- Fetch size customizable



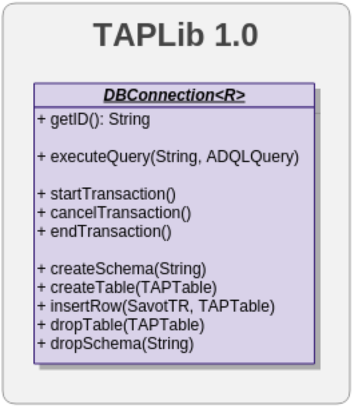
1. Interaction with database

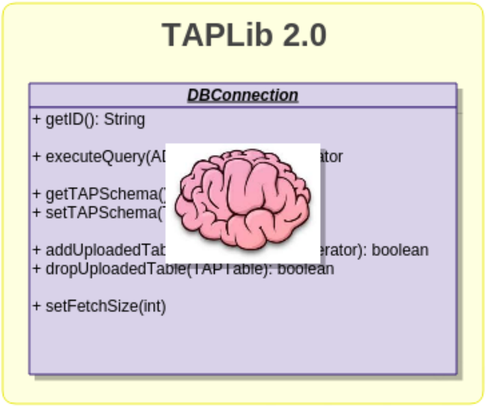
- Interface DBConnection simplified and more generic
- Fetch size customizable



1. Interaction with database

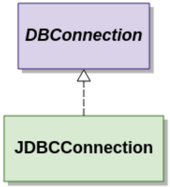
- Interface DBConnection simplified and more generic
- Fetch size customizable





1. Interaction with database

- Interface DBConnection simplified and more generic
- Fetch size customizable
- Generic JDBC concrete implementation



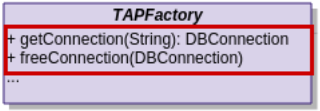
Should work with at least:
Postgres, Oracle, MySQL, SQLite, JavaDB/Derby

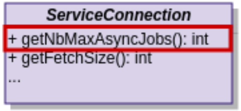
Able to inspect DB metadata in order to detect supported features:

- SELECT → set fetch size
- UPDATE/DELETE/... → Transaction + batch queries
- If schema not supported → tableName = {schemaName}_{tableName}
- default case sensitivity of DB
- etc...

1. Interaction with database

- Interface DBConnection simplified and more generic
- Fetch size customizable
- Generic JDBC concrete implementation
- Connection pool plug


TAPFactory
+ getConnection(String): DBConnection
+ freeConnection(DBConnection)
...



PostgreSQL

DBPool, BoneCP, ...

2. ADQL

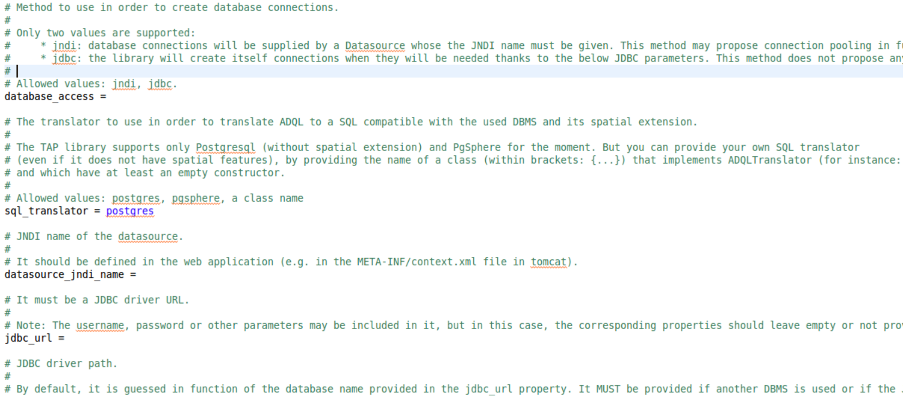
- Types (roughly) checked for columns and functions
- Better definition and management of geometries and coordinate systems
- UDF declaration simplification

3. Output formats

- Migration from SAVOT to STIL
- VOTable 1.3
- New VOTable serializations:
  - BINARY2
  - FITS
- FITS
- HTML

4. TAP configuration file

New TAP service building: **Configuration file**



```
# Method to use in order to create database connections.
#
# Only two values are supported:
#     * jndi: database connections will be supplied by a Datasource whose the JNDI name must be given. This method may propose connection pooling in fu
#     * jdbc: the library will create itself connections when they will be needed thanks to the below JDBC parameters. This method does not propose an
# 
# Allowed values: jndi, jdbc.
database_access =

# The translator to use in order to translate ADQL to a SQL compatible with the used DBMS and its spatial extension.
#
# The TAP library supports only Postgresql (without spatial extension) and PgSphere for the moment. But you can provide your own SQL translator
# (even if it does not have spatial features), by providing the name of a class (within brackets: {...}) that implements ADQLTranslator (for instance:
# and which have at least an empty constructor.
#
# Allowed values: postgres, pgsphere, a class name
sql_translator = postgres

# JNDI name of the datasource.
#
# It should be defined in the web application (e.g. in the META-INF/context.xml file in tomcat).
datasource_jndi_name =

# It must be a JDBC driver URL.
#
# Note: The username, password or other parameters may be included in it, but in this case, the corresponding properties should leave empty or not pro
jdbc_url =

# JDBC driver path.
#
# By default, it is guessed in function of the database name provided in the jdbc_url property. It MUST be provided if another DBMS is used or if the
```

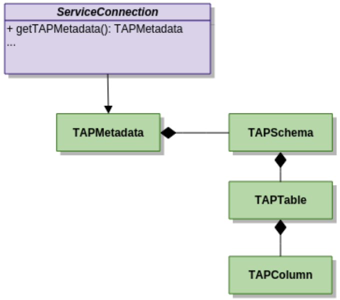
Much more details at [*http://cdsportal.u-strasbg.fr/adqltuto*](http://cdsportal.u-strasbg.fr/adqltuto)

5. Metadata declaration

- TAP metadata from...

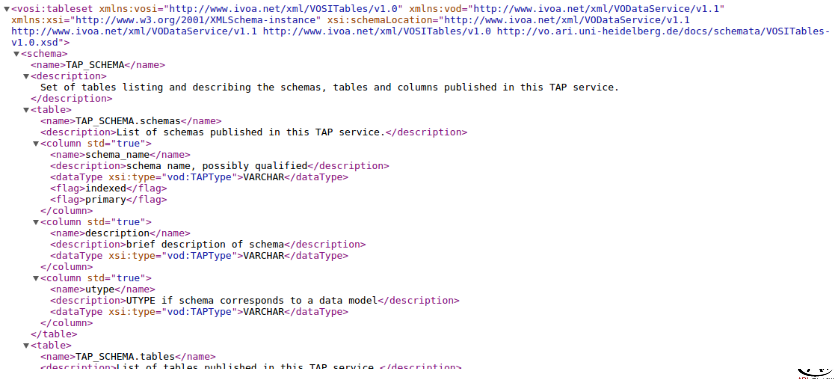
5. Metadata declaration

- TAP metadata from...
  - ...manually (i.e. programmatically)



5. Metadata declaration

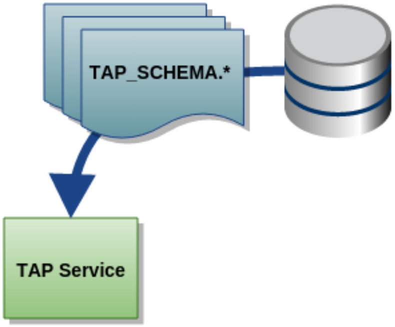
- TAP metadata from...
  - ...manually (i.e. programmatically)
  - ...a VOSI XML file



```
▼<vosi:tableset xmlns:vosi="http://www.ivoa.net/xml/VOSITables/v1.0" xmlns:vod="http://www.ivoa.net/xml/VODataService/v1.1"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.ivoa.net/xml/VODataService/v1.1
 http://www.ivoa.net/xml/VODataService/v1.1 http://www.ivoa.net/xml/VOSITables/v1.0 http://vo.ari.uni-heidelberg.de/docs/schemata/VOSITables-
 v1.0.xsd">
  ▼<schema>
     <name>TAP_SCHEMA</name>
    ▼<description>
       Set of tables listing and describing the schemas, tables and columns published in this TAP service.
     </description>
    ▼<table>
       <name>TAP_SCHEMA.schemas</name>
       <description>List of schemas published in this TAP service.</description>
      ▼<column std="true">
         <name>schema_name</name>
         <description>schema name, possibly qualified</description>
         <dataType xsi:type="vod:TAPType">VARCHAR</dataType>
         <flag>indexed</flag>
         <flag>primary</flag>
       </column>
      ▼<column std="true">
         <name>description</name>
         <description>brief description of schema</description>
         <dataType xsi:type="vod:TAPType">VARCHAR</dataType>
       </column>
      ▼<column std="true">
         <name>utype</name>
         <description>UTYPE if schema corresponds to a data model</description>
         <dataType xsi:type="vod:TAPType">VARCHAR</dataType>
       </column>
     </table>
    ▼<table>
       <name>TAP_SCHEMA.tables</name>
       <description>List of tables published in this TAP service.</description>
```

5. Metadata declaration

- TAP metadata from...
  - ...manually (i.e. programmatically)
  - ...a VOSI XML file
  - ...database



C. Evolution of the libraries

1. TAP

- *TAP 1.1*
- Resource `/examples`
- About TAP metadata:
  - Double quotes
  - Qualified names
- *Storage in VOSpace*

2. UWS

- Implementation of UWS 1.1 new features:
  - Job filtering on PHASE
  - Blocking behavior `/jobs/job-id?WAIT`
  - ARCHIVED phase
  - *You want to test it?* *`http://wiki.ivoa.net/internal/IVOA/InterOpJune2015Apps/uws1.1.war`*
  - *See* branch uws1.1 *on GitHub*

3. TAP & UWS

- User identification
  - Currently: generic and let free
  - A VO solution would be desirable, particularly if using an OpenID-like solution: *logging on one VO service lets access another VO service with the same account even if none has been created on the 2nd service.*

4. ADQL

- *ADQL 2.1 (in a parallel GitHub branch)*

Conclusion I

TAP - `http://cdsportal.u-strasbg.fr/taptuto`

- **Not backward compatible!**
- Documentation incomplete! Coming little by little.
- Javadoc

UWS - `http://cdsportal.u-strasbg.fr/uwstuto`

- Missing documentation! Coming little by little.
- Javadoc

Conclusion I

TAP - `http://cdsportal.u-strasbg.fr/taptuto`

- **Not backward compatible!**
  - Migration instructions (from 1.0 to 2.0) on the website
- Documentation incomplete! Coming little by little.
- Javadoc

UWS - `http://cdsportal.u-strasbg.fr/uwstuto`

- Missing documentation! Coming little by little.
- Javadoc

Conclusion I

TAP - http://cdsportal.u-strasbg.fr/taptuto

- **Not backward compatible!**
    - Migration instructions (from 1.0 to 2.0) on the website
- Documentation incomplete! Coming little by little.
    - 2 Getting Started sections
    - complete documentation about the TAP configuration file
- Javadoc

UWS - http://cdsportal.u-strasbg.fr/uwstuto

- Missing documentation! Coming little by little.
- Javadoc

Conclusion I

TAP - `http://cdsportal.u-strasbg.fr/taptuto`

- **Not backward compatible!**
  - Migration instructions (from 1.0 to 2.0) on the website
- Documentation incomplete! Coming little by little.
  - 2 Getting Started sections
  - complete documentation about the TAP configuration file
- Javadoc

UWS - `http://cdsportal.u-strasbg.fr/uwstuto`

- Missing documentation! Coming little by little.
  - **BUT** 2 examples (UWSService and UWSServlet) which will be a base for Getting Started
- Javadoc

Conclusion II

ADQL - `http://cdsportal.u-strasbg.fr/adqltuto`

- Documentation *complete* and *up-to-date*
- Online ADQL validator: `http://cdsportal.u-strasbg.fr/adqltuto/validator.html`
- Javadoc

All last developments on GitHub:
`https://github.com/gmantele/taplib`