TAP and ADQL

Florian Rothmaier, Janine Fohlmeister

Astronomisches Rechen-Institut Heidelberg

November 7, 2012
Table Access Protocol (TAP) ...

- is a **web service protocol** providing access to collections of tabular data ("tablesets")

- accepts queries against a tableset and responds to a query by generating an output table

- supports various query languages
  - support for **ADQL** (Astronomical Data Query Language) is mandatory

- is available via **TOPCAT** (besides various other services like VizieR or cone search)
  - TOPCAT’s TAP client enables you to access astronomical tables at remote data centres all over the world
Astronomical Data Query Language (ADQL)

- based on **SQL** (Structured Query Language)
- consists of a subset of the SQL grammar extended for astronomical needs
  - e.g. support for geometrical functions

**basic query syntax:**

```sql
SELECT [ TOP <#rows> ] <column name #1> <column name #2> ...
FROM <table name>
[ WHERE <conditions> ]
[ GROUP BY <columns> ]
[ ORDER BY <columns> ]
```

- squared brackets above mean: **optional**
- to select all columns of a table, the * symbol can be used, e.g.
  ```sql
  SELECT * FROM mytable
  ```
an application like **TOPCAT** which “speaks” TAP

- a **data centre** providing one or several tablesets of interest

- a **query** formulated in ADQL
access TAP via TOPCAT

- click on "VO" → "Table Access Protocol (TAP) Query"
search the registry for TAP services

- click on “Submit Query” (you can leave the “Keywords” line blank)

- click on “Enter Query”
select a table, e.g. “rave.main” (spectroscopic radial velocities of 50000 stars in the Milky-Way Southern hemisphere)
- column information is displayed

enter your ADQL query in the “ADQL Text” box, e.g.

```
SELECT [ TOP 10 ] raj2000, dej2000, rv FROM rave.main
```
- hint: clicking button “Examples” provides several example queries
query output: table with three columns (right ascension, declination, radial velocity) and ten rows (the first ten rows of our input table)

- use TOPCAT to display your data, cross-match them with other tabular data, ...
the mixer allows you to have columns from different input tables in your output table

technically, it’s a table join
  ▶ ADQL knows a so-called JOIN ... USING and a JOIN ... ON

SELECT [ TOP <setLimit> ] <columnList>
  FROM <table1>
  JOIN <table2>
  USING <columnName>

  ▶ columnName is a column which is present in table1 and table2

SELECT [ TOP <setLimit> ] <columnList>
  FROM <table1>
  JOIN <table2>
  ON <searchCondition>

  ▶ searchCondition defines the conditions which must be satisfied to join rows from table1 and table2, e.g. matching coordinates
you will get and learn more recipes in the following by our short exercise course

don’t hesitate to ask if you get stuck or feel unsure

We hope that you’ll enjoy the course...
and get inspired to satisfy your (g)astronomical delights...
References


M. Demleitner: GAVO web page offering a short course on ADQL and TAP, http://docs.g-vo.org/adql/html/.