TOPCAT/STILTS TAP Client Enhancements

Mark Taylor (Bristol)

with much help from

Grégory Mantelet

Markus Demleitner

IVOA Interop

Trieste

22 October 2016

`$Id: toptap.tex,v 1.6 2016/10/21 09:57:29 mbt Exp $`

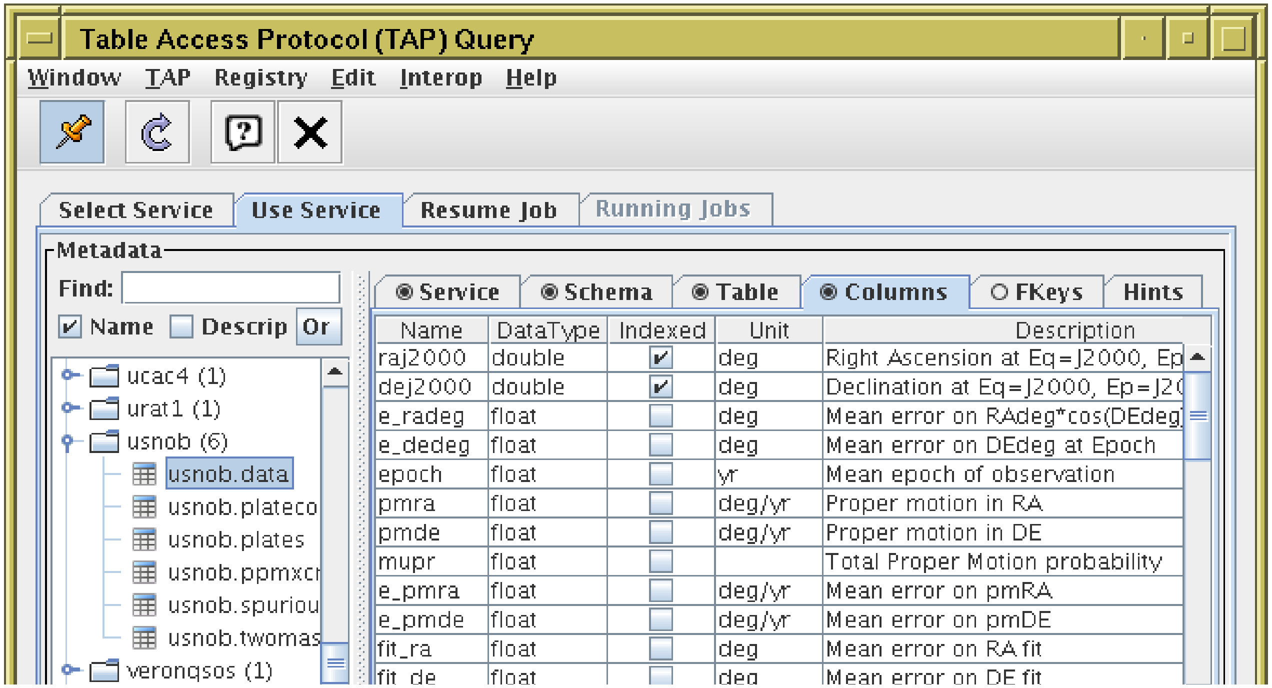
Outline

- TOPCAT TAP GUI
  - Background: TOPCAT TAP GUI metadata acquisition
  - Use of `column_index`
  - Non-standard table metadata display
  - VOSI 1.1 `detail` negotiation
- TAP library
  - UWS 1.1 blocking
- STILTS commands
  - tapquery `executionduration`/`destruction` params
- Validation (`taplint`)
  - New TAP-examples stage
  - ObsCore 1.1 validation
  - Per-service URL specification

Background: Metadata Acquisition

TOPCAT's TAP GUI acquires metadata from TAP service

- Service metadata is required for presentation to the user
- Acquire it in two main ways (with variations):
  - ▷ `/tables` endpoint (VODataService `vs:TableSet`)
  - ▷ TAP_SCHEMA table queries
- Default strategy depends on size of TAP service (total column count)
- **TAP|Metadata Acquisition** menu lets you configure it by hand



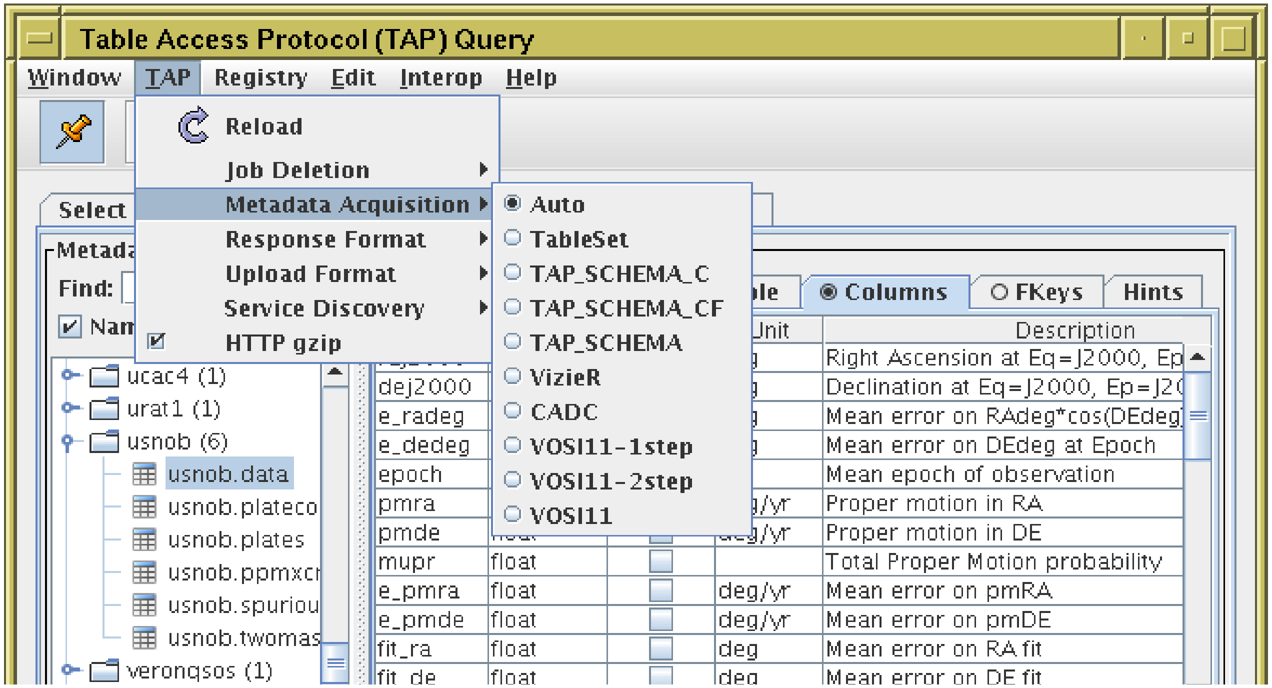
ucac4 (1)
urat1 (1)
usnob (6)
usnob.data
usnob.plateco
usnob.plates
usnob.ppmxcr
usnob.spuriou
usnob.twomas
veronqsos (1)

Name	DataType	Indexed	Unit	Description
raj2000	double	☑	deg	Right Ascension at Eq=J2000, Ep
dej2000	double	☑	deg	Declination at Eq=J2000, Ep=J2(
e_radeg	float	□	deg	Mean error on RAdeg*cos(DEdeg)
e_dedeg	float	□	deg	Mean error on DEdeg at Epoch
epoch	float	□	yr	Mean epoch of observation
pmra	float	□	deg/yr	Proper motion in RA
pmde	float	□	deg/yr	Proper motion in DE
mupr	float	□		Total Proper Motion probability
e_pmra	float	□	deg/yr	Mean error on pmRA
e_pmde	float	□	deg/yr	Mean error on pmDE
fit_ra	float	□	deg	Mean error on RA fit
fit_de	float	□	deg	Mean error on DE fit

Background: Metadata Acquisition

TOPCAT's TAP GUI acquires metadata from TAP service

- Service metadata is required for presentation to the user
- Acquire it in two main ways (with variations):
  - ▷ `/tables` endpoint (VODataService `vs:TableSet`)
  - ▷ TAP_SCHEMA table queries
- Default strategy depends on size of TAP service (total column count)
- **TAP|Metadata Acquisition** menu lets you configure it by hand



			Unit	Description
			g	Right Ascension at Eq=J2000, Ep
dej2000			g	Declination at Eq=J2000, Ep=J2(
e_radeg			g	Mean error on RAdeg*cos(DEdeg)
e_dedeg			g	Mean error on DEdeg at Epoch
epoch				Mean epoch of observation
pmra			g/yr	Proper motion in RA
pmde			deg/yr	Proper motion in DE
mupr	float	□		Total Proper Motion probability
e_pmra	float	□	deg/yr	Mean error on pmRA
e_pmde	float	□	deg/yr	Mean error on pmDE
fit_ra	float	□	deg	Mean error on RA fit
fit_de	float	□	deg	Mean error on DE fit

Use of column_index

Standards:

- TAP-1.1 (WD) defines new columns in TAP_SCHEMA tables:
  - ▷ table_index in table TAP_SCHEMA.tables *(sec 4.2)*
  - ▷ column_index in table TAP_SCHEMA.columns *(sec 4.3)*

⇒ Services can define preferred display order for columns and tables

- In TAP 1.0 these are optional, in TAP 1.1 they are mandatory, but may be NULL

TOPCAT GUI change:

- Column metadata listing now orders rows following column_index if present
  - ▷ Service does not need to be declared as TAP 1.1-compliant
  - ▷ Only effective when TOPCAT is doing TAP_SCHEMA-mode metadata acquisition (default mode for “large” TAP services)
- Ignores table_index (alphabetic table listing less confusing)
- Would ignore schema_index if it existed too

Notes for TAP service providers:

- Consider providing (non-NULL) column_index columns in TAP_SCHEMA.columns
  - ▷ Otherwise column metadata may be listed in random order in GUI
  - ▷ Order should preferably be consistent with tableset

Display of non-standard table/column metadata

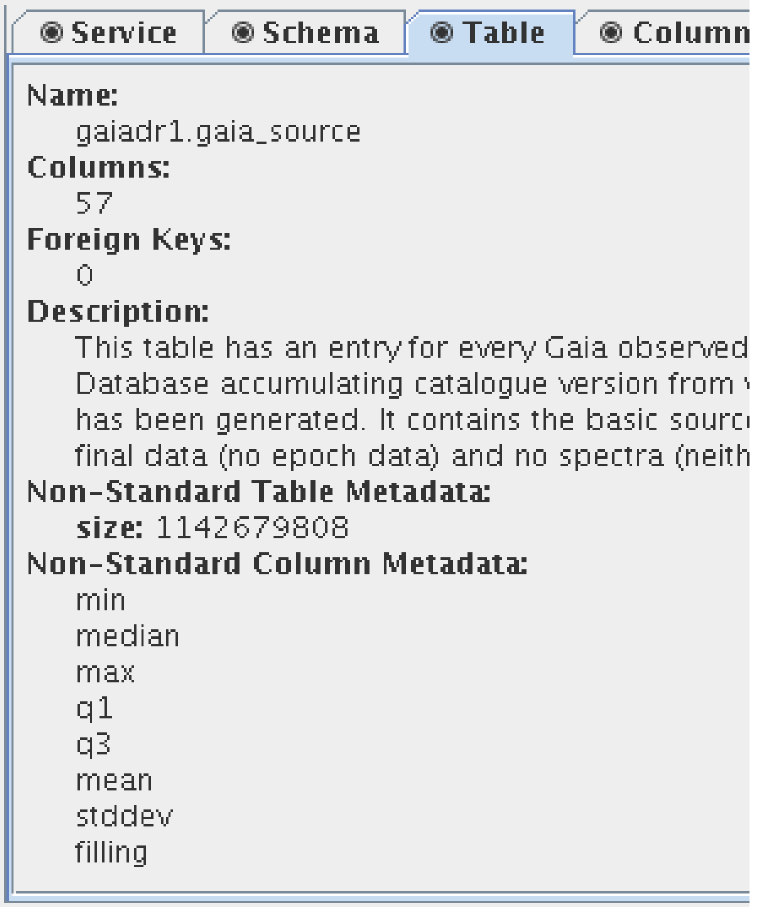
Standards:

- TAP permits additional (non-standard) columns in TAP_SCHEMA tables
- VODataService permits attributes from foreign namespaces

⇒ Services can provide non-standard schema/table/column metadata items

TOPCAT GUI change:

- TOPCAT now gathers and displays non-standard as well as standard per-column and per-table metadata
  - ▷ Tables: extra heading in **Table** tab
  - ▷ Columns: extra columns in **Columns** tab
- No attempt to understand semantics



Name:
gaiadr1.gaia_source
Columns:
57
Foreign Keys:
0
Description:
This table has an entry for every Gaia observed
Database accumulating catalogue version from
has been generated. It contains the basic sourc
final data (no epoch data) and no spectra (neith
Non-Standard Table Metadata:
size: 1142679808
Non-Standard Column Metadata:
min
median
max
q1
q3
mean
stddev
filling

Notes for TAP service providers:

- Consider providing useful non-standard metadata (row counts, column statistics)
  - ▷ extra columns in TAP_SCHEMA tables *and* foreign attributes in vs:TableSet
- "Implementation detail" TAP_SCHEMA cols now visible
- Standardise **row_count**? (but allow NULL values)

VOSI `detail` Negotiation

Standards:

- VOSI 1.1 (PR) defines non-binding `detail` parameter for `/tables` endpoint *(sec 3.4)*

  `detail=min`: client prefers table metadata *without* columns (compact)
  `detail=max`: client prefers table metadata *with* columns (verbose)

  - ▷ With-column metadata always available from table-specific endpoint
  - ▷ Services MAY (e.g. VizieR) always omit columns
  - ▷ VOSI 1.0 services should (MUST?) ignore any `?detail=...` parameters

⇒ Client/server preferences can be indicated retrieving scalable VODataService metadata

TOPCAT TAP client change:

- New **VOSI11** metadata acquisition mode:
  - ▷ No `detail` value specified initially (service decides)
  - ▷ If metadata has no detail, per-table detail requested later as required
  - ▷ Doesn't check if service is declared as VOSI 1.1 (since behaviour is compatible)
- Currently only available from Metadata Acquisition menu
- May be default mode in future (non-huge services? all services?)

Notes to TAP service providers:

- Consider implementing VOSI 1.1 detail negotiation

UWS Blocking

Standards:

- UWS 1.1 *(REC!)* defines `WAIT` parameter for job status requests *(sec 2.2.1.2)*

⇒ Clients can use blocking requests instead of repeated polls: more responsive, more efficient

TOPCAT/STILTS client change:

- If `uws:Job` document has `version="1.1"`, uses blocking not repeated polls

Notes to service providers:

▷ Consider implementing UWS 1.1 (both WAIT behaviour and `version="1.1"` jobs)

STILTS `tapquery`

STILTS `tapquery` command-line TAP client

- Introduce new parameters:
  - ▷ `executionduration=<seconds>`: sets UWS timeout limit
  - ▷ `destruction=<iso8601>`: sets UWS job destruction time
- This is just previously-missing functionality
- TO-DO: add the same option in TOPCAT TAP client

TAP Examples Validation

Standards:

- TAP 1.1 (WD) *(sec 2.6)* and DALI 1.1 (WD) *(sec 2.3)* define TAP Examples document
  - ▷ XHTML doc at `/examples` endpoint containing service-specific ADQL examples
  - ▷ Very helpful for users!

`taplint` validation change:

- New validation stage `EXA`
  - ▷ Checks document is conformant RDFa Lite (though doesn't validate XHTML)
  - ▷ For each example instance in document:
    - ○ Checks ADQL syntax
    - ○ Checks referenced table/column names against service metadata
    - ○ Attempts to execute query
  - ▷ Failures report Warnings (not Errors); ADQL examples may e.g. contain placeholders
- TOPCAT already makes use of TAP examples

Notes to TAP service providers:

- Provide service-specific TAP examples!
- And validate them!

ObsCore Validation

Standards:

- ObsCore 1.1 (PR) has some new/changed features from 1.0:
    - ▷ Changed DM declaration, new columns, corrected Utypes, extended enums, ...

`taplint` validation change:

- Detects from TAPRegExt `dataModel` declaration which ObsCore version is in use
- Performs version-specific validation tests
- Very latest changes (PR-ObsCore-v1.1-20160923) only in STILTS pre-release (v3.0-9+)

taplint Per-Service URL Specification

taplint validator invocation enhancements

- taplint hits many different endpoints (relative URLs /sync, /async, /tables, /capabilities, /examples, ...)
- By default these are all relative to value of the tapurl parameter
- New feature: you can now specify them individually if you want (parameters syncurl, asyncurl, tablesurl, capabilitiesurl, examplesurl, ...)
- Gives you more flexibility about invoking the validator, e.g.:
  - ▷ Play around with authentication:

    ```
    stilts taplint tapurl=http://example.com/tap
                   syncurl=http://example.com/tap/auth-sync
                   asyncurl=http://example.com/tap/auth-async
    ```

  - ▷ Try out not-yet-deployed service documents:

    ```
    stilts taplint tapurl=http://example.com/tap
                   examplesurl=file://home/mbt/experiments/examples.xml
    ```

  - ▷ ... etc

Summary

Various client updates matching 1.1-series standards

- Client behaviour becomes a bit nicer to use
- Takes advantage of service 1.1-series updates where present
- Behaviour stays the same where service behaviour not updated
- Where possible, client doesn't query services for standard versions,
  it just works better when advanced capabilities are present
  - ▷ in many cases, backward compatibility makes that easy ...
  - ▷ ... but dealing with optionally present TAP_SCHEMA columns is hard work