Taplint, the TAP Service Validator

Mark Taylor

H. H. Wills Physics Laboratory, Tyndall Avenue, University of Bristol, UK;
m.b.taylor@bristol.ac.uk

Abstract. TAP, the Table Access Protocol, is a widely used Virtual Observatory specification allowing client software to interact with remote database services in a standardised way. This paper presents `taplint`, a tool for assessing the compliance of deployed TAP services with the the dozen or so formal specifications that form the TAP protocol stack. We provide an overview of its capabilities and operation, and the context within which it is used to improve robustness of data services.

1. Introduction

TAP, the Table Access Protocol (Dowler et al. 2019) is a Virtual Observatory protocol suite allowing client software to interact with remote database services in a standardised way, including acquiring rich metadata and submitting simple or complex queries in an SQL-like language. It is a VO success story, underlying much current access to astronomy data archives, for instance providing the primary access to ESA's Gaia Archive (Gaia Collaboration 2016). The protocol stack involved however is quite complex, involving a dozen or so separate IVOA specifications, and implementors face many opportunities to make mistakes.

The International Virtual Observatory Alliance (IVOA) is the standards body within which TAP has been developed. The IVOA *Document Standards* specification (Genova et al. 2017), which defines how protocol standards are developed and adopted, requires that a validator tool, as well as interoperable implementations, must be available before a standard can be endorsed. This requirement is in place partly as a check of the written specification, since implementing a validator is a good way to pick up ambiguities and inconsistencies in the text, and partly as an aid to both implementors and users of the protocols. Running validation tests on a service, ideally as part of the development and deployment process, can identify errors and issues that could otherwise cause trouble for service users.

Within this context, the `taplint` tool has been developed as a validator for deployed TAP services. Since TAP is closely related to a number of other IVOA and some non-IVOA specifications, `taplint` serves as a validator for a number of standards beyond the TAP document itself.

2. Usage and Availability

`taplint` is provided as part of STILTS (Taylor 2006), which is a command-line package for manipulating catalogues and other tables, written in Java. To run it, the only

requirement is a Java Runtime Environment (Java 8 or later), and the `stilts.jar` file that can be downloaded from the package web page[1]. The validator can then be run by supplying on the command line the base URL of the TAP service to test:

```
java -jar stilts.jar taplint tapurl=http://example.com/tap
```

Additional command-line parameters may optionally be supplied to control which testing stages are run, the form of the output, and some details of service interaction.

When run, the tool writes a series of reports to standard output. By default the output is line-oriented, `grep`-friendly, and intended for human consumption (though JSON output is also available). Each report line is of the form:

```
T-SSS-MMMMxN aaaaa...
```

where the parts have the following meanings:

`T:` report type, one of `I`(nfo), `W`(arning), `E`(rror), `F`(ailure) and `S`(ummary). Info gives information about the test being performed, Warning reports behaviour that is questionable or not recommended by standards, and Error reports behaviour contrary to standards

`SSS:` 3-character stage identifier, indicating which part of the test sequence is in progress

`MMMM:` 4-character unique code identifying the test being run; there are currently 200+ of these

`xN:` repeat count for multiple occurrences of reports with the same `SSS-MMMM`

`aaaa...:` informative message

An example report would be (presented here on two lines for readability):

```
E-OBS-CUTP-7 Wrong Utype in ObsCore column access_format:
             Access.format != obscore:Access.Format
```

To limit the quantity of output no more than a fixed maximum, by default 10, reports with the same `SSS-MMMM` will be written. This prevents the output file becoming unwieldy when the same or a similar problem is encountered in multiple tests.

Considerable efforts are made to explain the nature of the issues reported in the message text. However, failures can be subtle, and moreover the validator, like other software, may contain bugs, so users are encouraged to contact the author with queries about `taplint` output or behaviour.

A typical run might take three minutes and generate 200 lines of output, but these values can vary widely depending on the service. Full details of the tool's operation are given in the STILTS documentation[2].

---

[1] `http://www.starlink.ac.uk/stilts/stilts.jar`

[2] `http://www.starlink.ac.uk/stilts/sun256/taplint.html`

3. Standards Covered

The TAP standard itself defines how clients can interrogate metadata, pose queries, and retrieve results from remote database services. However it does this with reference to many other IVOA (and some non-IVOA) standards concerning serialization, service interaction, metadata encoding, capability declaration and more; there are also further standards defining domain-specific data models that sit on top of TAP. `taplint` understands most of these, and runs specific tests that a target service is behaving as prescribed.

Behaviour defined by the following IVOA specifications is tested to a greater or lesser extent: TAP, VOTable, UWS, VODataService, ADQL, VOResource, VOSI, TAPRegExt, DALI, ObsCore, ObsLocTAP, EPN-TAP, UCD, VOUnits, SoftID and SSO. These documents are available from the IVOA *Documents and Standards* web page[3], and are the result of much collaborative work over many years between IVOA members.

4. Tests Performed

`taplint` runs a battery of test queries against the TAP service, posing as a client and checking that the response in each case is as mandated by the relevant specifications. It tries to test as many aspects of the required behaviour as it can, especially in the areas of metadata provision, output formats, job submission and compliance with data models.

The tests are grouped in a sequence of *stages*; by default all those applicable to the target service are performed, but the list of stages can be restricted if only certain aspects are of interest for a given run. Tests include validation of output documents against relevant XML schemas, careful checking of VOTable output against all the requirements of the VOTable specification, submitting ADQL queries in various modes, checking that table metadata is standards-compliant and consistent between various ways of obtaining it, testing asynchronous job submission behaviour, and testing that declared metadata is in line with data models applying to served data.

`taplint` tries its best to test that all the requirements and recommendations of the standards it knows about are implemented as specified in those standards. It should be noted however that no validator can supply all possible inputs to a complex service like TAP, so just because `taplint` does not identify errors in a service does not guarantee that none are present.

5. Discussion

`taplint` provides a way to identify flaws in running TAP services during development and operations, and can thus contribute to the efforts of service providers in improving the experience for their users, since broken or partially working services can be a major cause of user frustration. It has been used at many data centers for this purpose, including by ASDC, CADC, CDS, CfA, ESA, ESO, GAVO, IPAC, MAST and NASA, as well as providing the TAP-specific component of the bulk validation services run by ESA, PADC and NASA to assess overall VO operational status (see e.g. Savalle

[3] `https://www.ivoa.net/documents/`

et al. 2020). Data providers developing or operating a TAP service are encouraged to consider running `taplint` to check its compliance with the standards.

The VO standards landscape is in constant flux, and `taplint` requires frequent updates in response to new and updated specifications, changing or contested interpretations of existing documents and user feedback. Recent improvements (STILTS v3.4-2) include a new UCD and Unit validation stage, validation of the data and metadata specified by the new EPN-TAP and ObsLocTAP data models, better stage selection control, checking service VO component identification, improved reporting of VOTable issues, and validation of content declared by xtype.

Writing validation tools like `taplint` is (like much of the work in the VO) time-consuming and not very glamorous, but they can provide an important contribution to the robustness of the service ecosystem, as well as feeding back to improve the standards definition process.

Acknowledgments. This work has been supported by STFC and GAVO. Development of the software has benefitted from much input from its users and collaborative discussions with VO experts.

References

Dowler, P., Rixon, G., Tody, D., & Demleitner, M. 2019, Table Access Protocol Version 1.1, IVOA Recommendation 27 September 2019

Gaia Collaboration 2016, A&A, 595, A2. 1609.04172

Genova, F., Arviset, C., Demleitner, M., Glendenning, B., Molinaro, M., Hanisch, R. J., & Rino, B. 2017, IVOA Document Standards Version 2.0, IVOA Recommendation 17 May 2017

Savalle, R., Le Sidaner, P., & Normand, J. 2020, in Astronomical Data Analysis Software and Systems XXVII, edited by P. Ballester, J. Ibsen, M. Solar, & K. Shortridge, vol. 522 of Astronomical Society of the Pacific Conference Series, 339

Taylor, M. B. 2006, in ADASS XV, edited by C. Gabriel, C. Arviset, D. Ponz, & S. Enrique, vol. 351 of ASP Conf. Ser., 666