



*International
Virtual
Observatory
Alliance*

TAP-1.0 Errata Version 1.0

IVOA Note 2014-12-22

Working group

Data Access Layer

This version

<http://www.ivoa.net/documents/TAP1Err1/20141222>

Latest version

<http://www.ivoa.net/documents/TAP1Err1>

Previous versions

This is the first public release

Author(s)

Marco Molinaro, the IVOA DAL WG members

Editor(s)

Marco Molinaro

Abstract

The following set of Errata is intended to amend the TAP-1.0 (Dowler et al. 2010) recommendation. Reported Errata are mainly clarifications about the content of the current TAP-1.0 document with respect to the actual usage and implementations. The contents are: the names of the tables for the TAP upload mechanism; multiple UPLOAD posts after job creation; column types for VOTable (Ochsenbein et al. 2013) objects serialised from database records; the *size* reserved ADQL keyword used as column name in *TAP_SCHEMA.columns* table; the usage of VOTable in error responses for TAP.

Status of This Document

This is an IVOA Note expressing suggestions from and opinions of the authors. It is intended to share best practices, possible approaches, or other perspectives on interoperability with the Virtual Observatory. It should not be referenced or otherwise interpreted as a standard specification.

A list of current IVOA Recommendations and other technical documents can be found at <http://www.ivoa.net/Documents/>.

Contents

1	Introduction	3
2	UPLOAD Table Names	3
2.1	Erratum Content	4
2.2	Impact Assessment	4
3	Multiple UPLOAD Posts	4
3.1	Erratum Content	5
3.2	Impact Assessment	5
4	Database Column Type in VOTable	5
4.1	Erratum Content	5
4.2	Impact Assessment	6
5	The <i>size</i> Column in <i>TAP_SCHEMA.columns</i>	6
5.1	Erratum Content	6
5.2	Impact Assessment	7
6	VOTable usage in TAP	7
6.1	Erratum Content	7
6.2	Impact Assessment	8
A	Changes from Previous Versions	8

Acknowledgments

These Errata takes direct input from the TAP Implementation Notes (Demleitner et al. 2013) and following discussions held at the Spring (ESAC, Spain) and Fall (Banff, Canada) 2014 IVOA Interoperability Meetings.

Conformance-related definitions

The words “MUST”, “SHALL”, “SHOULD”, “MAY”, “RECOMMENDED”, and “OPTIONAL” (in upper or lower case) used in this document are to be interpreted as described in IETF standard, Bradner (1997).

The *Virtual Observatory (VO)* is general term for a collection of federated resources that can be used to conduct astronomical research, education, and outreach. The *International Virtual Observatory Alliance (IVOA)* is a global collaboration of separately funded projects to develop standards and infrastructure that enable VO applications.

1 Introduction

This Note contains erratum content for the following aspects of the TAP-1.0 (Dowler et al. 2010) recommendation:

- names of uploaded tables (Sec. 2.5 of TAP-1.0) in Sec. 2;
- multiple UPLOAD posts (Sec. 2.5.1 of TAP-1.0) in Sec. 3;
- database column types (Sec. 2.5 of TAP-1.0) in Se. 4;
- *size* field in *TAP_SCHEMA.columns* (Sec. 2.6.3 of TAP-1.0) in Sec. 5;
- VOTable usage (Sec. 2.9 of TAP-1.0) in Sec. 6.

2 UPLOAD Table Names

Section 2.5 of TAP-1.0 requires the name of the uploaded tables to be a

...legal ADQL table name with no catalog or schema (e.g. an unqualified table name)

This may allow ADQL-2.0 (Ortiz et al. 2008) *delimited_identifiers* to be used for uploaded tables as the ADQL *table_name* expands to either *regular_identifiers* or *delimited_identifiers*. This, however, was clearly not the intention of the text, as the use of *delimited_identifiers* is not (fully) supported by the syntax of the UPLOAD parameter (Dowler et al. 2010, , Sec. 2.5.1), i.e.

```
UPLOAD=table_a,http://host_a/path;table_b,http://host_b/path
```

To clarify this issue, the proposal has been made to replace the text within parenthesis in the above quoted excerpt of TAP-1.0 to allow in the UPLOAD parameter only table names as strings following the *regular_identifier* ADQL syntax.

2.1 Erratum Content

The TAP-1.0 recommendation states, for the table upload mechanism (Sec. 2.5 of [Dowler et al. 2010](#)), that

The client specifies the name of the uploaded table; this name must be a legal ADQL table name with no catalog or schema (e.g. an unqualified table name).

The parenthetical exemplification may allow any ADQL-2.0 *identifier* to be used as a string in the UPLOAD parameter posted to the TAP service accepting uploads, including *delimited_identifier* ones that the UPLOAD parameter syntax doesn't support.

This Erratum updates the TAP-1.0 recommendation amending the third sentence in the first paragraph of Sec. 2.5 of that document by substituting it with the following:

The client specifies the name of the uploaded table; this name must be a legal ADQL table name with no catalog or schema (**i.e. a string following the *regular_identifier* production of the ADQL-2.0 standard**).

2.2 Impact Assessment

This Erratum emendation could, in theory, invalidate existing clients that might want to use *delimited_identifiers* in uploads. Due to the difficulties with the UPLOAD parameter syntax, however, that would not really be supported in TAP-1.0, either. This solution, thus, clarifies the TAP-1.0 document and will be maintained in the next minor version of the protocol itself.

3 Multiple UPLOAD Posts

Another topic related to the upload mechanism in TAP refers to TAP-1.0 specification at Sec. 2.5.1. Table upload can happen at job creation time, but the UWS-1.0 ([Harrison & Rixon 2010](#)) recommendation allows parameter posting also after the job creation phase. This means that it should be clarified what should happen when an UPLOAD parameter is posted to a job that has already one or more uploads in place. The proposal is to consider UPLOAD posts as *accumulating*, i.e. each UPLOAD parameter will trigger the creation of one or more tables. What will be left unspecified is the behaviour of uploading multiple tables having the same name.

3.1 Erratum Content

The behaviour of a TAP service when an UPLOAD parameter is posted to it after job creation and when uploaded tables are already present is unspecified. Since UWS allows posting parameters after job creation (Sec. 2.1.11, [Harrison & Rixon 2010](#)), Section 2.5.1 of TAP-1.0 needs to specify what happens when the UPLOAD parameter is posted into a job that already has one or more uploads.

This Erratum fixes this unspecified behaviour by adding at the end of the TAP-1.0 Section 2.5.1 the following paragraph:

UPLOAD parameters are accumulating, i.e., each UPLOAD parameter posted to a job will create one or more tables in the TAP_UPLOAD schema. When the table names from two or more upload items agree after case folding, the service behaviour is unspecified. Clients thus cannot reliably overwrite uploaded tables; to correct errors, they have to tear down the existing job and create a new one.

3.2 Impact Assessment

Considering that the multiple upload behaviour here described and set as normative was not specified in the recommendation, the change may only affect server side implementations of the TAP specification whether they had chosen a different behaviour before this Erratum was issued.

4 Database Column Type in VOTable

The ADQL-2.0 specification does not explicitly talk about types. The TAP specification, using the ADQL-2.0 as the mandatory query language, needed to cover this normative lack to map VOTable datatypes with server side database datatypes. This is the meaning of the table in Sec. 2.5 (page 19) of TAP-1.0. However, this table, especially the last column (*database column type*), due to the missing ADQL normative on types, cannot be taken as conformance criteria and needs a better description of its meaning.

4.1 Erratum Content

The datatype mapping between VOTable serialization and database objects is delegated to the table at page 19 (Sec. 2.5) of the current TAP specification ([Dowler et al. 2010](#)). Due to the normative lack of ADQL-2.0 with respect to datatypes this mapping cannot be considered a normative part of the TAP recommendation.

This Erratum sanitizes this situation adding the following text before Sec. 2.5.1 of TAP-1.0 document:

Note that the last column of datatype mapping table in this section is not normative. Implementations SHOULD try to make sure that the actual types chosen are at least signature-compatible with the recommended types (i.e., integers should remain integers, floating-point values floating-point values, etc.), such that clients can reliably write queries against uploaded tables.

For columns with xtype adql:REGION, this is particularly critical, since databases typically use different types to represent various STC-S objects. Clients are advised to assume that such columns will be approximated with polygons in the actual database table.

4.2 Impact Assessment

Erratum changes to TAP-1.0 shouldn't impact server side or client side TAP implementations given that actual mapping usage already should have taken into account the missing normative about types.

Moreover, considering the fact that it is clear that for a predictable behaviour across individual ADQL implementations, ADQL should talk about types, a Type System will be provided within the next revision of the current ADQL-2.0 specification.

5 The *size* Column in *TAP_SCHEMA.columns*

The table *TAP_SCHEMA.columns* as specified in Sec. 2.6.3 of the current TAP standard has a column named *size*. This is unfortunate since SIZE is an ADQL-2.0 reserved word (see *SQL reserved keywords* on Sec. 2.1.2 of ADQL-2.0), and thus must be quoted in queries. This is not stated in the TAP specification.

5.1 Erratum Content

SIZE is a SQL-92 reserved keyword adopted as a reserved keyword also in ADQL-2.0. This means that, to be used in an ADQL statement as an identifier (e.g. a column name), it must be put in double quotes to avoid collision with the reserved word meaning.

TAP-1.0 (see Sec. 2.6.3 of [Dowler et al. 2010](#)) specifies that the *TAP_SCHEMA.columns* table must have a column named *size* to specify the length of a variable length datatype. This means that that column must be double quoted when used in an ADQL query. However the standard doesn't report this need.

This Erratum fills this omission appending the following paragraph to the end of TAP-1.0 Sec. 2.6.3:

To use *size* in a query, it must be put in double quotes since it collides with an ADQL-2.0 reserved word. Since delimited identifiers are case-sensitive, for the *size* column both clients and servers MUST always (in particular, in the DDL for *TAP_SCHEMA*) use lower case exclusively.

5.2 Impact Assessment

The normative part of this Erratum does not impact any server or client side implementation of current TAP protocol because it simply states what should in any case happen in real TAP/ADQL usage. If the *size* column of *TAP_SCHEMA.columns* were not quoted, a SQL server would raise an error in response (probably a malformed statement exception).

To avoid the reserved keyword collision, in the next major version of TAP, this column will be called *arraysize*, in next minor revision(s) the name of the column will be kept for back-compatibility reasons.

6 VOTable usage in TAP

The wording in TAP in Sec. 2.9 and Subsec. 2.9.1 of the current specification about the use of VOTable (Ochsenbein et al. 2013) is somewhat inconsistent about the format of VOTable error documents. It is clear that Sec. 2.9 is discussing both successful and error outputs, and in the case of an error no *TABLE* element will normally be present, only one or more *INFO* elements. However, while the intention is clear from the fourth example in Subsec. 2.9.1, it is not clear from the wording.

6.1 Erratum Content

The wording of Sec. 2.9 and Subsec. 2.9.1 of current TAP specification is inconsistent with the rules for VOTable error documents. Section 2.9 says

The VOTable must contain a *RESOURCE* element identified with the attribute *type='results'*, containing a single *TABLE* element with the results of the query.

and Section 2.9.1 says

The *RESOURCE* element must contain, before the *TABLE* element,
...

The inconsistency comes from the fact that in case of a response error no *TABLE* element will normally be present, only one or more *INFO* elements.

This Erratum removes the inconsistency by changing:

- first sentence in the third paragraph of Sec. 2.9 from

The VOTable must contain a *RESOURCE* element identified with the attribute *type='results'*, containing a single *TABLE* element with the results of the query.

to

The VOTable must contain a *RESOURCE* element identified with the attribute *type='results'*, containing exactly one *TABLE* element with the results of the query if the job execution was successful or no *TABLE* element if the job execution failed to produce a result.

- first paragraph of Subsec. 2.9.1 from

The *RESOURCE* element must contain, before the *TABLE* element, an *INFO* element with attribute *name="QUERY_STATUS"*. The value attribute must contain one of the following values:

to

The *RESOURCE* element must contain an *INFO* element with attribute *name="QUERY_STATUS"* indicating the success of the operation. For *RESOURCE* elements that contain a *TABLE* element, this *INFO* element must appear lexically before the *TABLE*. The following values are defined for this *INFO* element's *value* attribute:

6.2 Impact Assessment

Since the wording changes introduced by this Erratum are only meant to better describe the usage of an existing recommendation (the VOTable one) inside the current TAP protocol, no effects on current TAP services and consuming application is foreseen.

A Changes from Previous Versions

No previous versions yet.

References

- Bradner, S. (1997), 'Key words for use in RFCs to indicate requirement levels', RFC 2119.
URL: <http://www.ietf.org/rfc/rfc2119.txt>
- Demleitner, M., Harrison, P. & Taylor, M. (2013), 'TAP Implementation Notes, Version 1.0', IVOA Note.

Dowler, P., Rixon, G. & Tody, D. (2010), 'Table access protocol version 1.0', IVOA Recommendation.

URL: <http://www.ivoa.net/Documents/TAP>

Harrison, P. & Rixon, G. (2010), 'Universal worker service pattern, version 1.0', IVOA Recommendation.

URL: <http://www.ivoa.net/Documents/UWS>

Ochsenbein, F., Williams, R., Davenhall, C., Demleitner, M., Durand, D., Fernique, P., Giaretta, D., Hanisch, R., McGlynn, T., Szalay, A., Taylor, M. & Wicenec, A. (2013), 'Votable format definition, version 1.3', IVOA Recommendation.

URL: <http://www.ivoa.net/documents/VOTable/>

Ortiz, I., Lusted, J., Dowler, P., Szalay, A., Shirasaki, Y., Nieto-Santisteban, M. A., Ohishi, M., O'Mullane, W., Osuna, P., the VOQL-TEG & the VOQL Working Group (2008), 'IVOA astronomical data query language', IVOA Recommendation.

URL: <http://www.ivoa.net/Documents/latest/ADQL.html>