International
Virtual
Observatory
Alliance

TableReg: Registering TAP-Queriable Tables Conforming to Standard Schemas

Version 1.0

IVOA Note 2024-03-15

Working Group
Registry

This version
https://www.ivoa.net/documents/TableReg/20240315

Latest version
https://www.ivoa.net/documents/TableReg

Previous versions
This is the first public release

Author(s)
Markus Demleitner

Editor(s)
Markus Demleitner

Abstract

An increasingly popular pattern in the Virtual Observatory, pioneered by Obscore, is to define a schema for one or more tables in a database and then publish data or metadata by putting conforming tables into TAP services. This document discusses how such resources should be represented in the VO Registry to facilitate data discovery, in particular global, all-VO dataset discovery.

It turns out that the existing registration patterns for Obscore, RegTAP, and EPN-TAP require some adjustments. The document therefore also proposes transition strategies for these.

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 https://www.ivoa.net/documents/.

Contents

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 RFC2119 (Bradner, 1997).

The *Virtual Observatory (VO)* is a 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

Beginning with Obscore 1.0 (Tody and Micol et al., 2011), an increasing number of Virtual Observatory standards at their core just define a table schema – understood here as a well-defined set of columns within one or more relations – and rely on the IVOA's Table Access Protocol TAP (Dowler and Rixon et al., 2019) to let clients actually run queries. Standards of this type include:

- Obscore (Louys and Tody et al., 2017) – a table for metadata of observational data products
- RegTAP (Demleitner and Harrison et al., 2019) – a 13-table schema with metadata of VO resources
- ObsLocTAP (Salgado and Ibarra et al., 2021) – a table schema to communicate plans for observations and metadata of completed observations
- EPN-TAP (Erard and Cecconi et al., 2022) – a table schema for solar system data

More such standards are currently being developed. Examples include LineTAP (Castro Neves and Moreau et al., 2023) and the Obscore extension for radio data.

Of course, resources complying to these standards must be made discoverable to be useful. Both Obscore and RegTAP have employed the `dataModel` element specifically introduced into TAPRegExt (Demleitner and Dowler et al., 2012) to declare the presence of tables adhering to a standard schema in a TAP service.

In practice, however, the `dataModel` scheme has some severe shortcomings:

1. Lack of resource metadata: In resource records located during discovery, the global VOResource metadata (title, authors, and perhaps most importantly coverage in space, time and spectrum) is that of the TAP service, which the standard tables may share with any number of other tables. Hence, it will at best be overly general. More often than not, it will be severely misleading.

2. Unclear relationships: In particular for Obscore – rather typically serving several data collections at once –, a serious problem is that data collection records can only generically say that they are served by the TAP service (cf. Demleitner and Taylor (2019) for the general scheme of relating data collections and services). From that, clients cannot deduce whether the data is available through, say, obscore, or only in some custom table.

3. Suitability: Adherence to a data model simply is not a property of a TAP service. It is a property of a specific table or schema.

In addition, when standards do not define "singletons" (i.e., a TAP service can only contain one instance of each complying table, and the tables' names are fixed) but instead table schemas applicable to arbitrarily-named tables, the discovery process also needs to yield table names. EPN-TAP was the first standard with such a requirement. As a solution, it switched to using the table's *utype* element in the VODataService (Demleitner and Plante et al., 2021) *tableset* for discovery.

This mode of discovery was subsequently also employed in ObsLocTAP and LineTAP. It still has a drawback, though: Clients discovering a service with a *table* (or *schema* in the case of RegTAP) with the bespoke *utype* have a hard time determining whether what they have found is the data collection's record (and hence the global metadata pertains to the data collection itself) or the record of the TAP service serving the data collection (in which case the global metadata pertains to the TAP service and will be essentially unrelated to the data collection in question).

The registration schema this note proposes remedies that by prescribing that for resource records for standard-compliant tables, the resource type *vs:CatalogResource* (against *vs:CatalogService* of the actual TAP service) must be used.

In Sect. 2, we discuss the proposed scheme and give XML snippets illustrating the various components necessary for efficient global discovery. One reason to introduce the scheme is to enable expressing inclusion relationships between resources for the benefit of clients doing global discovery. Sect. 3 discusses this mechanism in more detail. Sect. 4 addresses the question of how to transition from what the standards currently say (and the services and clients implement) to a VO adopting the scheme proposed here. The particular example of Obscore is treated in Sect. 5 and can largely serve as boilerplate text for future standards of its kind.

2 Registering and Discovering Standard Tables

In the scheme proposed here, TAP-queriable resources conforming to standard table schemas will be registered as *vs:CatalogResource*-typed Resources. If they consist of only a single table or of multiple, independently publishable tables (e.g., Obscore, EPN-TAP), an IVOA identifier (Demleitner and Plante et al., 2016) is defined for this particular table type. This will refer to a StandardsRegExt (Harrison and Burke et al., 2012) *StandardKey*, generally in the defining standard's resource record. This standard key SHOULD contain a version tag precise to the minor version. Looking back, the convention of long version tags may seem questionable, but it is at least

conceivable that clients may want to constrain their discovery to tables containing additions made during a major version cycle.

For robustness and flexibility, clients should not use the information on the minor version to infer presence or absence of certain features (e.g., a column added in a minor update), however, but rather directly check for that feature's presence, and standards should provide means for doing so independently of the full version tag.

Hence, the VODataService *table* element for a table conforming to Obscore 1.2 would begin like this:

```
<table>
  <name>ivoa.obscore</name>
  <title>Obscore Table in the Fictional Data Centre</title>
  <description>
    This is example metadata for use in the
    TableReg specification.
  </description>
  <utype>ivo://ivoa.net/std/obscore#table-1.2</utype>
  ...
```

where Obscore's StandardsRegExt record contains a fragment:

```
<key>
  <name>table-1.2</name>
  <description>The data model for a table conforming to version 1.2
  of this specification. This implies a set of nn mandatory columns,
  as well as the table's name, ivoa.obscore.
  </description>
</key>
```

Standards defining full schemas, i.e., sets of interconnected tables that only make sense together – at the moment, RegTAP is our only example – will similarly define a StandardKey to use with a schema element. The RegTAP registry record currently has:

```
<key>
  <name>1.1</name>
  <description>The data model for the tables making up the relational
   registriy in version 1.1. This key is used to locate TAP services
  implementing RR in TAPRegExt dataModel or VODataService schema/@utype
  elements.
  </description>
</key>
```

Hence, the tableset of the record for the Heidelberg RegTAP service ivo://org.gavo.dc/rr/q/create contains:

```
<schema>
   <name>rr</name>
   <title>GAVO RegTAP Service</title>
   <description>
     Tables containing the information in the IVOA Registry[...]
   <utype>ivo://ivoa.net/std/RegTAP#1.1</utype>
   <table>...
```

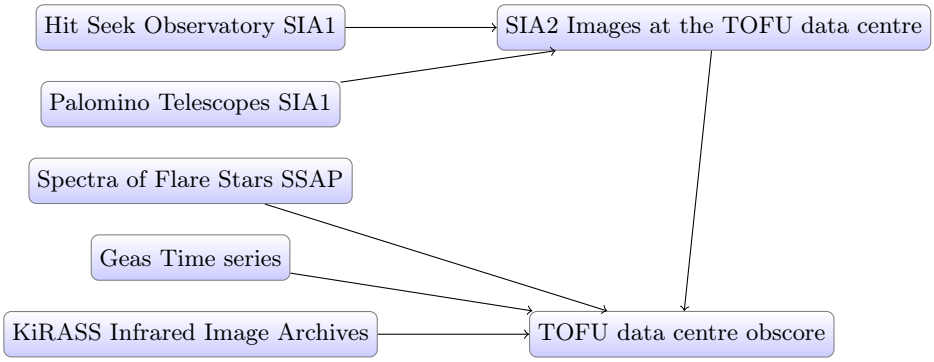

Figure 1: Benefits of making relationships between resources explicit. The nodes in this graph correspond to services, the arrows point from a a service to a collective service also serving the data served by the first service. By evaluating the relationships, a client can deduce that the data published through the seven services can be queried by running one Obscore query on the TOFU data centre.

It is not recommended to follow the RegTAP's example of omitting an indicator of what is referred to in the standard key's name. Hence, prefer, for instance, `schema-1.1` to `1.1`; it is always conceivable that additional versioned entities may require standard keys.

Let us stress that for standard discovery, the minor version must be ignored. Clients should be written such that if they work for any 1.x version, they work for all 1.x versions, except of course where compelling reasons exist to require features not present in earlier minor versions.

3 Expressing Relationships Between Tables and Other Resources

An important reason to enumerate resources conforming to a schema is global discovery of whatever is described in the table; in the case of Obscore, this would be observational datasets.

Furthermore, an important reason to define registry records for such resources is that data collections published using multiple standards or through multiple services can machine-readably declare that querying one such resource is enough to cover the entire data available. In that way, clients doing global discovery can skip services publishing data they already queried using other services. Consider Fig. 1 for an illustration of the dramatic savings enabled by making such relationships explicit.

In VOResource, relationships are declared between registry resources using the *`relationship`* element, containing a *`relationshipType`* and one or more *`relatedResource`*-s. For relationships between data collections and

services making them queriable, the Endorsed Note on Discovering Data Collections (Demleitner and Taylor, 2019) prescribes *IsServedBy* from the IVOA relationship type vocabulary[1].

We argue that this term can be re-used here, even though the relationship's label might appear somewhat misplaced when a CatalogService-typed record declares that it is served by a CatalogResource-typed record; that is what happens if a SIA-published data collection says it is also present in an obscore table.

More specifically, where data contained in or published through resource A is also contained in or published through a more general (in the sense of: making other data collections queriable, too) resource B, the resource record of record A should declare a relationship to B with a `relationshipType` of *IsServedBy*.

Resources must not declare circular *IsServedBy* relationships.

4 Transitioning to a TableReg World

There are already several standards registering TAP-published tables in one way or another. For standards currently in Working Draft or later, we give transition plans, both in terms of the standards process and the actual records in the VO Registry, here. Numbers in the following sections are for March 2024.

4.1 Obscore

To enable viable global dataset discovery, reparing the registration pattern is most urgent for Obscore. Against that, the current, `dataModel`-based registration pattern is furthest away from what is proposed here. Hence, it is probably unavoidable to issue a new minor version of Obscore.

In order to provide a smooth transition, the current pattern of using an ivoid in the embedding TAP service's `dataModel` field in the TAP capability must be retained. In addition, a new standard for Obscore 1.2 will contain an adapted version of the material in sections 5.1 and 5.2.

To enable viable global datatset discovery as soon as possible, once sufficient consensus within the Registry and Data Model WGs has been found, the Registry WG should contact the operators of the obscore services currently active in order to advise them on creating an additional Obscore registry record with the required declarations.

[1] http://www.ivoa.net/rdf/voresoure/relationship_type

4.2 EPN-TAP

Discovery of EPN-TAP 2.0 (Erard and Cecconi et al., 2022) resources is already based on giving a table utype in a tableset. It says:

> Normally, however, the tableset will be contained in a VODataService *CatalogService* record with a TAP capability, and this capability will be an auxiliary capability as per DDC (Demleitner and Taylor, 2019). For one-table services, a full TAPRegExt (Demleitner and Dowler et al., 2012) capability is also allowed.

Against the present proposal, this admits pure-TAP services and does not require the use of *vs:CatalogResource*-typed resource records (although in practice the wide majority of EPN-TAP publishers chose that resource type despite the recommendation in the standard, which predates the availability of *vs:CatalogResource*).

Leaving open the resource type complicates the discovery pattern significantly, since clients have to filter out duplicates when a table is present in both the TAP service's and the resource's tableset. We hence propose to require the use of *vs:CatalogResource* in the future, where we consider an adivsory erratum sufficient; in the end, updates to the existing resource records need to be done in a cooperation between the Registry WG and the data providers anyway.

In February 2024, there were 488 resources with tables with the EPN-TAP table utype in their tableset. Of these,

- 475 have the epncore utype, 13 (from four different authorities) the legacy vopdc utype (which is probably a good measure for how many resource records may be hard to update)
- 245 have a TAP auxiliary standard id, 243 a full TAP id.
- 249 are of the type *vs:CatalogService*, 239 of type *vs:CatalogResource* (in a perfect system, there are zero or one *vs:CatalogService* records per *vs:CatalogResource* record; zero would be when the TAP services comes without a tableset).

Investigating more closely, it turns out that only four servers (counted by access URL) chose the *vs:CatalogService* resource type for their table resource record, and only one TAP server is missing from the registry. It seems entriely possible to rectify these problems on short notice.

4.3 ObsLocTAP

ObsLocTAP (Salgado and Ibarra et al., 2021) already prescribes the pattern proposed here. The discovery query given is missing a constraint on the resource type, though. We believe this is easily repaired through an erratum.

4.4 RegTAP

In its section 7, RegTAP 1.1 (Demleitner and Harrison et al., 2019) already offers two registration patterns. One is based on *dataType* in the TAP capabilities. In the next minor version, this mechanism will be deprecated. Instead, the alternative scheme, based on *vg:Registry*-typed records, needs to be more fully specified and advertised as the primary means of locating RegTAP searchable registries. Also, since RegTAP is a complete schema, the utype to search will sit on the *schema* element.

Against the pattern given in sect. 5.2, we would hence give

```
SELECT DISTINCT table_name, access_url
FROM rr.res_table
  NATURAL JOIN rr.capability
  NATURAL JOIN rr.interface
  NATURAL JOIN rr.resource
WHERE
  table_utype LIKE 'ivo://ivoa.net/std/regtap#table-1.%'
  AND standard_id LIKE 'ivo://ivoa.net/std/tap%'
  AND intf_role='std'
  AND res_type='vg:registry'
```

as the canonical discovery pattern.

Keeping *vg:Registry* as resource type rather than going to *vs:CatalogResource* as proposed here is convenient, as it allows the declaration of parameters of any OAI-PMH service that may accompany the RegTAP endpoint. Accepting this slight inconsistency also seems justified since registry discovery plays a minor role in user code.

vg:Registry also does not admit the declaration of *coverage* as per VODataService 1.2. Given that space-time coverage is not a useful concept at least for full registries, and the discovery of partial searchable registries (which conceivably could be profit from coverage declarations) has not been a relevant use case so far, this seems an acceptable deficit.

At this point, there is no particular urgency for this change; given the (theoretical) equivalence of the RegTAP services, it is unlikely richer metadata on the registry records will be required any time soon.

5 Obscore Tables in the Registry

This section is intended both as the blueprint for what Obscore 1.2 should say (in addition to the transitional *dataModel* declaration in the capabilities) and as a template on which to base the Registry sections of similar standards.

5.1 Registering Obscore Tables

Obscore tables are registered using VODataService (Plante and Stébé et al., 2010) tablesets, where the table utype is set to

`ivo://ivoa.net/std/obscore#table-1.2`.

The tableset is contained in a resource record of the VODataService type *vs:CatalogResource* with a TAP capability, where this capability is an auxiliary capability as per DDC (Demleitner and Taylor, 2019). The TAP service serving the table must also be registered, and an *IsServedBy* relationship must be declared from the Obscore record to the TAP record.

When registry records for data collections published through the Obscore table are also published – and publishers are strongly urged to do that –, an *IsServedBy* relationship must also be declared from the individual collections' records to the Obscore record.

An example for a registry record in VOResource comes with this document[2].

The noteworthy points in the record are:

- A *relationship* element referencing the main TAP service through which the service is queriable as per DDC:

```
<relationship>
  <relationshipType>IsServedBy</relationshipType>
  <relatedResource ivo-id="ivo://org.gavo.dc/tap"
    >GAVO Data Center TAP service</relatedResource>
</relationship>
```

- The declaration for the auxiliary capability, including the access URL so clients do not need to follow the relationship just declared if all they need is the access URL:

```
<capability standardID="ivo://ivoa.net/std/TAP#aux">
   <interface role="std" version="1.1" xsi:type="vs:ParamHTTP">
     <accessURL use="base">http://dc.zah.uni-heidelberg.de/tap</accessURL>
   </interface>
</capability>
```

- Most importantly, the declaration of the table utype that lets clients discover that this particular table contains Obscore data:

```
<table>
  <name>ivoa.obscore</name>
  <title>GAVO Data Center Obscore Table</title>
  <description>The IVOA-defined obscore table, containing generic
  metadata for datasets within this datacenter.</description>
```

[2] https://www.ivoa.net/documents/TableReg/20240315/example-record.xml

```
    <utype>ivo://ivoa.net/std/obscore#table-1.1</utype>
    ...
  </table>
```

5.2 Discovering Obscore Tables

Obscore clients in general are interested in TAP endpoints serving Obscore tables as well as the table's metadata, such as its coverage in space, time, and spectrum. By the registration pattern given in 5.1, this translates into resources with TAP (auxiliary) capabilities that have a standard key for version 1 Obscore in a table utype; this will normally also match the TAP service record itself (as it generally also gives the tableset). Therefore, an additional constraint on the record type is introduced.

Translated into RegTAP (Demleitner and Harrison et al., 2019), the following query would return TAP access URLs and the table names:

```
SELECT DISTINCT table_name, access_url
FROM rr.res_table
  NATURAL JOIN rr.capability
  NATURAL JOIN rr.interface
  NATURAL JOIN rr.resource
WHERE
  table_utype LIKE 'ivo://ivoa.net/std/obscore#table-1.%'
  AND standard_id LIKE 'ivo://ivoa.net/std/tap%'
  AND intf_role='std'
  AND res_type='vs:catalogresource'
```

The regular expression in the utype match makes sure minor version increments do not prevent resource discovery; by IVOA versioning rules, all Obscore tables of minor version 1 can be operated by all Obscore clients of version 1. We do not constrain the version of the TAP service. Clients may want to adapt the TAP discovery pattern to match their specific needs.

Clients not prepared to negotiate authentication should also add a constraint `AND authenticated_only=0` to avoid unnecessary attempts to access protected resources.

Clients can add additional constraints (e.g., on publishers, coverage, or, in VODataService 1.3 or later, data product types) to this basic query as usual in VOResource, i.e., by `NATURAL JOIN`-ing the tables containing the columns and adding additional `WHERE` clauses. Constraints against the embedding TAP service, however, require a more complex join through the `rr.relationship` table; this is not expected to be a common use case.

Incidentally, for Obscore 1.2, `table_name` in this query will always be `ivoa.obscore`. This item hence is only relevant for standards that allow for flexible table names.

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. http://www.ietf.org/rfc/rfc2119.txt.

Castro Neves, M., Moreau, N. and Demleitner, M. (2023), 'LineTAP: IVOA relational model for spectral lines', IVOA Working Draft 23 March 2023. https://ivoa.net/documents/LineTAP/20230323/.

Demleitner, M., Dowler, P., Plante, R., Rixon, G. and Taylor, M. (2012), 'TAPRegExt: a VOResource Schema Extension for Describing TAP Services Version 1.0', IVOA Recommendation 27 August 2012, arXiv:1402.4742. doi:10.5479/ADS/bib/2012ivoa.spec.0827D, https://ui.adsabs.harvard.edu/abs/2012ivoa.spec.0827D.

Demleitner, M., Harrison, P., Molinaro, M., Greene, G., Dower, T. and Perdikeas, M. (2019), 'IVOA Registry Relational Schema Version 1.1', IVOA Recommendation 11 October 2019. doi:10.5479/ADS/bib/2019ivoa.spec.1011D, https://ui.adsabs.harvard.edu/abs/2019ivoa.spec.1011D.

Demleitner, M., Plante, R., Linde, T., Williams, R. and Noddle, K. (2016), 'IVOA Identifiers Version 2.0', IVOA Recommendation 23 May 2016, arXiv:1605.07501. doi:10.5479/ADS/bib/2016ivoa.spec.0523D, https://ui.adsabs.harvard.edu/abs/2016ivoa.spec.0523D.

Demleitner, M., Plante, R., Stébé, A., Benson, K., Dowler, P., Graham, M., Greene, G., Harrison, P., Lemson, G., Linde, T. and Rixon, G. (2021), 'VODataService: A VOResource Schema Extension for Describing Collections, Services Version 1.2', IVOA Recommendation 02 November 2021. doi:10.5479/ADS/bib/2021ivoa.spec.1102D, https://ui.adsabs.harvard.edu/abs/2021ivoa.spec.1102D.

Demleitner, M. and Taylor, M. (2019), 'Discovering Data Collections Within Services Version 1.1', IVOA Endorsed Note 20 May 2019. doi:10.5479/ADS/bib/2019ivoa.rept.0520D, https://ui.adsabs.harvard.edu/abs/2019ivoa.spec.0520D.

Dowler, P., Rixon, G., Tody, D. and Demleitner, M. (2019), 'Table Access Protocol Version 1.1', IVOA Recommendation 27 September 2019. doi:10.5479/ADS/bib/2019ivoa.spec.0927D, https://ui.adsabs.harvard.edu/abs/2019ivoa.spec.0927D.

Erard, S., Cecconi, B., Le Sidaner, P., Demleitner, M. and Taylor, M. (2022), 'EPN-TAP: Publishing Solar System Data to the Virtual Observatory Version 2.0', IVOA Recommendation 22 August 2022. https://ui.adsabs.harvard.edu/abs/2022ivoa.spec.0822E.

Harrison, P., Burke, D., Plante, R., Rixon, G., Morris, D. and IVOA Registry Working Group (2012), 'StandardsRegExt: a VOResource Schema Extension for Describing IVOA Standards Version 1.0', IVOA Recommendation 08 May 2012, arXiv:1402.4745. doi:10.5479/ADS/bib/2012ivoa.spec.0508H, https://ui.adsabs.harvard.edu/abs/2012ivoa.spec.0508H.

Louys, M., Tody, D., Dowler, P., Durand, D., Michel, L., Bonnarel, F., Micol, A. and IVOA DataModel Working Group (2017), 'Observation Data Model Core Components, its Implementation in the Table Access Protocol Version 1.1', IVOA Recommendation 09 May 2017. doi:10.5479/ADS/bib/2017ivoa.spec.0509L, https://ui.adsabs.harvard.edu/abs/2017ivoa.spec.0509L.

Plante, R., Stébé, A., Benson, K., Dowler, P., Graham, M., Greene, G., Harrison, P., Lemson, G., Linde, T. and Rixon, G. (2010), 'VODataService: a VOResource Schema Extension for Describing Collections, Services Version 1.1', IVOA Recommendation 02 December 2010, arXiv:1110.0516. doi:10.5479/ADS/bib/2010ivoa.spec.1202P, https://ui.adsabs.harvard.edu/abs/2010ivoa.spec.1202P.

Salgado, J., Ibarra, A., Ehle, M., Gabriel, C., Kretschmar, P., Kuulkers, E., Merín, B., Ness, J.-U., Salazar, E., Saxton, R., Cecconi, B., Foster, K., Demleitner, M., Dempsey, J., Molinaro, M., Sánchez, C., Taylor, M., Díaz Trigo, M., Fernández, M., Kennea, J., Kettenis, M., Matt, G., Osborne, J., de Oña Wilhelmi, E., Salbol, E. J., Sivakoff, G., Tao, L., Tohuvavohu, A., Tibbetts, M. and Workman, B. (2021), 'Observation Locator Table Access Protocol Version 1.0', IVOA Recommendation 24 July 2021. doi:10.5479/ADS/bib/2021ivoa.spec.0724S, https://ui.adsabs.harvard.edu/abs/2021ivoa.spec.0724S.

Tody, D., Micol, A., Durand, D., Louys, M., Bonnarel, F., Schade, D., Dowler, P., Michel, L., Salgado, J., Chilingarian, I., Rino, B., de Dios Santander, J. and Skoda, P. (2011), 'Observation Data Model Core Components, its Implementation in the Table Access Protocol Version 1.0', IVOA Recommendation 28 October 2011, arXiv:1111.1758. doi:10.5479/ADS/bib/2011ivoa.spec.1028T, https://ui.adsabs.harvard.edu/abs/2011ivoa.spec.1028T.