



VAMDC-TAP Data Access Protocol Specification

Document Information

Editors:	L. Nenadovic
Authors:	G. Rixon
Contributors:	VAMDC WP6 working group
Type of document:	standards documentation
Status:	draft
Distribution:	public
Work package:	WP6
Version:	11.05
Date:	08/06/2011
Document code:	
Document URL:	http://www.vamdc.org/documents/vamdc-tap_v11.05.pdf

Abstract: VAMDC-TAP is a protocol for a class of web services giving access to databases in VAMDC. The protocol is a specialization of IVOA's TAP protocol. The required features of TAP and the specializations are defined here.

Version History

Version	Date	Modified By	Description of Change
V0.1	30/09/2010	G. Rixon	first draft
V11.05	27/05/2011	L. Nenadovic	editing for first release

Disclaimer

The information in this document is subject to change without notice. Company or product names mentioned in this document may be trademarks or registered trademarks of their respective companies.

All rights reserved

The document is proprietary of the VAMDC consortium members. No copying or distributing, in any form or by any means, is allowed without the prior written agreement of the owner of the property rights.

This document reflects only the authors' view. The European Community is not liable for any use that may be made of the information contained herein.

Acknowledgements

VAMDC is funded under the "Combination of Collaborative Projects and Coordination and Support Actions" Funding Scheme of The Seventh Framework Program. Call topic: INFRA-2008-1.2.2 Scientific Data Infrastructure. Grant Agreement number: 239108.

CONTENTS

1	The VAMDC variant of the Table Access Protocol (TAP)	1
1.1	Required features	1
1.2	Query language	2
1.3	Format of results	2
1.4	Standard view of database	2
1.5	Registration	3
1.6	Making a synchronous query	3
1.7	HTTP Header Information	3
1.8	HTTP result codes	5
2	References	6

THE VAMDC VARIANT OF THE TABLE ACCESS PROTOCOL (TAP)

Many data-sets in VAMDC include information that can be rendered in the VAMDC-XSAMS data model. Data in that common model could be transformed to and from a table model which uses the same columns for all data-sets. If all the data-sets had this table model as part of the schemata of their databases, then a SQL query to that model would work on all data-sets, and the results could be written in a common format.

VAMDC-TAP is a protocol for data-access services that provide the common table model matching VAMDC-XSAMS and which can return the results of queries in VAMDC-XSAMS. VAMDC-TAP services accept queries in a restricted form of SQL (VSS1: VAMDC SQL Sub-set #1) and return results in VAMDC-XSAMS or in certain tabular formats. Implementations of VAMDC-TAP map queries from the common table-model to the actual schemata of their databases.

VAMDC-TAP provides “virtual data”. I.e., it associates data selection criteria, defined by a query text, with an archived data-set, defined by the address to which the query is sent, the two combined in one URL. Each such URL represents the results of the query as if they had been pre-computed and stored on a web server. The data URLs are semi-permanent; they can be copied between application, bookmarked, emailed to colleagues, etc.

VAMDC-TAP is based on IVOA’s [Table Access Protocol \(TAP\)](#). TAP already provides virtual data and allows us to plug in our query language VSS1 and our data model VAMDC-XSAMS.

VAMDC-TAP is defined as a web-service protocol. That means that VAMDC-TAP services are driven by GET and POST requests to HTTP (or HTTPS) URIs. Low-level details of the protocol are defined by the HTTP RFCs. Further, the service can be implemented in any language and on any database engine without breaking interoperability.

Section 2 lists the features required in a conforming VAMDC-TAP service. Sections 3 to 6 inclusive define details of these features. Section 7 is not part of the specification but instead explains how to form and execute a VAMDC-TAP query.

1.1 Required features

A VAMDC-TAP implementation must be organized as a TAP service. This means that the implementation must be a RESTful web-service and must provide web resources in the pattern mandated by TAP.

In the TAP standard, some features are mandatory and some are optional. VAMDC-TAP uses a sub-set of the mandatory features and some of the optional ones.

A VAMDC-TAP service must support at least the following features:

- synchronous data-query, as specified in TAP.
- XML description of service capabilities as specified in TAP by reference to the [Virtual Observatory Support Interfaces \(VOSI\)](#) standard
- VSS1 query language

- XSAMS as an output format
- a standard view of the relational data (see below)

If the service provides all of these features then it may be registered as VAMDC-TAP.

A VAMDC-TAP service should support all the mandatory features of TAP. If it does so, it may be registered as both VAMDC-TAP and as TAP. (See section below for an explanation of the distinction.)

If a VAMDC-TAP service is to be registered as both TAP and VAMDC-TAP, then it should provide the VOSI output that specifies the details of tables and columns for use in queries. Without these metadata TAP is very hard to use.

A VAMDC-TAP service may support output formats other than XSAMS. The most-useful formats are tabular: VOTable for virtual-observatory integration and CSV for use in spreadsheets. If the service is to be registered as a TAP service, the TAP standard says that the service must support VOTable.

A VAMDC-TAP service may support query languages other than VSS1. If the service is to be registered as a TAP service, the TAP standard says that the service must support Astronomical Data Query Language (ADQL).

1.2 Query language

A VAMDC-TAP service must support VAMDC SQL sub-set #1 or VSS1 as its primary query-language. This language is specified in a query by setting the HTTP parameter LANG to VSS1. The service must not be sensitive to the case of the parameter value, but clients should use upper case for this value.

1.3 Format of results

A VAMDC-TAP service must support XSAMS as its primary output-format. This format is specified in a query by setting the HTTP parameter FORMAT to XSAMS. The service must not be sensitive to the case of the parameter value, but clients should use upper case for this value.

When the format is XSAMS, the service must return the results with the MIME type application/xml. The service must not use text/xml for these results.

TAP allows the value of the FORMAT parameter to be either the MIME type of the results or a keyword denoting the format. The MIME type for XSAMS is ambiguous and can easily be confused with VOTable or other XML formats. Clients must not use the MIME type when specifying XSAMS output.

1.4 Standard view of database

A VAMDC-TAP service must provide a relational view of its database that can be queried using the SQL-sub-set language VSS1. VSS1 has no FROMs clause, so VSS1 queries implicitly address the database as a single table. A VAMDC-TAP service must be arranged to support this.

When implementing a VAMDC-TAP service for a particular database, the implementor must define the *restrictable* quantities: these are the columns of the standard view on which constraints may be placed in the WHERE clause of a query. The restrictables must be taken from the set defined in VAMDC's standard dictionary, in which the names, data types, units and semantics are specified. A given implementation need not support every item in the dictionary. The restrictables for a service must be noted in the service's registration.

The implementor of a service must also define the *returnables*: these are the terms from the VAMDC dictionary which will appear in the results of a query. The returnables must also be noted in the registration.

If a service supports both TAP and VAMDC-TAP, the tables available for a TAP query need not be related to the standard view. One of the main reasons for supporting TAP is to give access to a wider range of tables.

1.5 Registration

A VAMDC-TAP service must be registered. The registration document must be of type `CatalogService (v1.0)` or `CatalogService (v1.1)` as defined by IVOA (i.e. it must use the `VODataService` standard in either of two versions).

The VAMDC-TAP capability must be denoted in the registration document by a capability block with the attribute `standardID=ivo://vamdc/std/VAMDC-TAP` and the structural type `VAMDC-TAP`. In this schema, the returnables are denoted by `returnable` elements the values of which are the names of the returnables. Similarly, the restrictables are denoted by `restrictable` elements. There may be zero or more of each type (but a service with zero of either type is not very useful). All the `returnable` elements must come before the first `restrictable` element.

If the service is to be registered as both TAP and VAMDC-TAP, then the TAP capability must be denoted by a capability block with the attribute `standardID=ivo://ivoa.net/std/TAP`. This capability must have structural type `Capability`. This type has no TAP-specific metadata.

The two capability blocks are siblings in the registration document. A service that fully supports TAP will have both a VAMDC-TAP and a TAP capability.

A service registered with a TAP capability should have, in its registration document, details of the queryable tables and columns, denoted according to the [VODataService specification](#).

The capabilities of a service must be made available to the registry in its VOSI-capabilities output; this avoids the need to enter them manually in the registry. If the service has a TAP capability, the details of tables and columns should be presented in its VOSI-tables output.

1.6 Making a synchronous query

The base URL for a TAP service can be discovered from the registry. From this, the access URL for the query can be derived: add `/sync`¹ to the base URL and then add parameters to define the specific query.

Parameter name	Meaning	Supported values in VAMDC-TAP
REQUEST	Requested operation	doQuery
LANG	Name of query language	VSS1, ADQL
FORMAT	Format for results of query	XSAMS, VOTABLE, application/xml ²
QUERY	Text of query	As per query language

Parameter names are insensitive to case: service implementations must accept any mix of case.

The parameter values are URL-escaped to replace illegal characters with hexadecimal codes (e.g. each space is replaced by `%20`). In practice, only the `QUERY` parameter needs to be escaped. Clients of the service must escape the parameters before sending the request.

This is a plausible example of a query URL, fully decorated with parameters:

```
http://some.server/some/path/sync?REQUEST=doQuery&LANG=VSS1&FORMAT=XSAMS&QUERY=select%20*
```

Here, the base URL, found in the registry, is `http://some.server/some/path`. The query is `SELECT *`.

The query is initiated by an HTTP-GET request to the access URL. The HTTP response carries the results of the query in the specified format.

1.7 HTTP Header Information

1.7.1 Statistics

A VAMDC-TAP service should provide information/statistics about the amount of data that will be returned for a specific query in the HTTP headers of the reply to the query. This allows a user (e.g. the portal) to use the `HEAD`

¹ This access-URL identifies the web-resource for synchronous queries. Asynchronous queries are sent to a separate web-resource.

² Implies `VOTable`.

method (instead of GET) on the same query-URL to gather information before the query is actually executed and the data transferred.

The names of the headers to be used are

- VAMDC-COUNT-ATOMS
- VAMDC-COUNT-MOLECULES
- VAMDC-COUNT-SPECIES
- VAMDC-COUNT-SOURCES
- VAMDC-COUNT-STATES
- VAMDC-COUNT-COLLISIONS
- VAMDC-COUNT-RADIATIVE
- VAMDC-COUNT-NONRADIATIVE

Their values should be the count of the corresponding blocks in the XSAMS schema, e.g. the number of radiative transitions that will be returned for this query. With a reasonable database layout the nodes should easily be able to gather these numbers by running COUNT queries on their corresponding tables.

1.7.2 Volume limitation

A VAMDC-TAP service can limit the amount of data it returns via the synchronous interface, for example to prevent the fetching of the whole database or for performance reasons. The service must then fill the HTTP-header of the response with the field VAMDC-TRUNCATED that contains the percentage that the returned data represent with respect to the total amount available for that query. It is up to each service to decide both where to put the limit and how to implement it, for example the number of states or transitions.

1.7.3 Document size estimate

VAMDC-APPROX-SIZE HTTP header is intended to provide the estimation of the size of the response document. It should return an integer value, representing estimate uncompressed document size in megabytes.

Volume limitation example

Volume example is implemented for the Django-based prototypes and activated for the VALD node which now returns max 1000 transitions (plus corresponding states and sources, of course). Similar limits are easily done for the other nodes in a few lines of code. In addition to the HTTP-header, the VAMDC-XSAMS generator also puts a comment into the beginning of the XML-document which also notifies of the truncation.

For example, a query like this:

```
wget -S -O bla.xml "http://vamdc.fysast.uu.se:8888/node/vald//tap/sync/?REQUEST=doQuery&LANG=VSS1"
```

will show the HTTP-header as:

```
VAMDC-TRUNCATED: 2.9 %
```

In Django node software implementation you will also find the following section at the top of the returned XML:

```
<!--
  ATTENTION: The amount of data returned has been truncated by the node.
  The data below represent 2.9 percent of all available data at this node that
  matched the query.
-->
```

1.8 HTTP result codes

Following HTTP result codes should be implemented by the node software for the SYNC TAP endpoint:

HTTP Code	meaning	Content type	Response body
200	Request processed normally, data is present.	application/x-xsams+xml	XSAMS instance document
204	Request processed, but no matching data found	none	none
400	bad request with malformed query string or missing restrictable	application/x-votable+xml	votable with error message
404	not used, will be encountered if the endpoint is wrong	unspecified, may be application/x-votable+xml	unspecified, may be the votable with error message
500	internal crash	unspecified, may be application/x-votable+xml	unspecified, may be the votable with error message

REFERENCES

- International Virtual Observatory Alliance (IVOA) : <http://www.ivoa.net>
- Table Access Protocol (TAP) : <http://www.ivoa.net/Documents/TAPRegExt/20110127/index.html>
- Virtual Observatory Support Interfaces (VOSI) : <http://www.ivoa.net/Documents/VOSI/20100311>