



*International
Virtual
Observatory
Alliance*

IVOA Provenance Table Access Protocol (ProvTAP)

Version 1.0

IVOA Working Draft 2019-10-07

Working group

DM

This version

<http://www.ivoa.net/documents/ProvTAP/20191007>

Latest version

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

Previous versions

Author(s)

François Bonnarel, Mireille Louys, Markus Nullmeier, Michèle Sanguillon, Mathieu Servillat, IVOA Data Model Working Group

Editor(s)

François Bonnarel

Abstract

This document describes the ProvTAP protocol for accessing provenance information according to the IVOA ProvenanceDM standard. It defines how the elements of ProvDM are described in the TAP schema tables and provides guidelines for implementing with TAP 1.1.

Status of this document

This is an IVOA Working Draft for review by IVOA members and other interested parties. It is a draft document and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use IVOA Working Drafts as reference materials or to cite them as other than “work in progress”.

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

Contents

1	Introduction	2
1.1	context	2
1.2	Role within the VO architecture	3
2	ProvTAP in action	3
3	VOSI availability and capabilities	5
4	VOTable serialization for provenance metadata and ProvTAP TAP schema	5
4.1	Tables in the VOTable serialisation or in the ProvTAP TAP schema	5
4.2	Detailed table description	6
	Appendices	13
	Appendix A TAP schema in xml	13
	Appendix B TAP service content exemple. Database dump in VOTable.	13
	Appendix C Changes from Previous Versions	13
	Bibliography	13

1 Introduction

1.1 context

ProvTAP is a TAP service specialization for delivering IVOA Provenance metadata and build up serialisation documents consistently with the IVOA Provenance data model (Servillat and Riebe et al., 2019). As for any TAP service (Dowler and Rixon et al., 2010), ProvTAP is providing ADQL query responses consisting of a single table each, gathering columns selected from various tables defined in the service's TAP schema. The default format of the Query Response is VOTable. Other formats (JSON, TSV, etc.) may also be optionally supported.

The definition of a ProvTAP service is similar to the approach used in the ObsCore specification (Louys and Bonnarel et al., 2011) where ObsCore data model plays the role of underlying model for ObsTAP services. It is

also inspired by the multiple table design proposed in the the RegTAP specification (Demleitner and Harrison et al., 2013) with respect to VOResource data model.

Basically, data-model consistency for a TAP service is asserted by the datamodel element in the capability of the service. The relational view of the model is expressed via the table structure in the TAP schema and the definition of each column datatype, ucd, utype attributes in the same schema.

ProvTAP is not the only DAL service able to provide provenance metadata. The ProvSAP (Bonnarel and Riebe et al., 2018) specification offers a complementary method based on a search-by-parameters strategy and does not require the server to adopt the TAP architecture. ProvSAP services allow to retrieve sequences or subgraphs of Provenance metadata associated to a starting entity or activity allowing different combinations.

1.2 Role within the VO architecture

IVOA ProvTAP specification is used to define TAP services with a specific TAP schema expressing view of the IVOA provenance Data Model. It relies on DALI (Dowler and Demleitner et al., 2013) for general DAL definitions. It benefits from existing IVOA notations and standards like UCD, VOUnits, VOTable, VOResource and extensions of the latter such as VODataservice and TAPREgExt.

Fig. 1 shows the dependencies of this document with respect to other existing standards.

2 ProvTAP in action

Here are a few examples of ADQL queries for ProvTAP (values are taken from the test metadata presented in appendix) :

- To retrieve all activity metadata for activities sharing the same activity-Description, by looking for the activities with the same a_description value:

```
SELECT * FROM Activity WHERE
Activity.a_description = 'hipsgen15'
```
- To retrieve all activities associated with a specific agent:

```
SELECT WasAssociatedWith.waw_activity, Activity.a_name,
      Activity.a_comment FROM WasAssociatedWith
INNER JOIN Activity ON
WasAssociatedWith.waw_activity_id = Activity.id
WHERE WasAssociatedWith.waw_agent = 'agent_1_1'
```

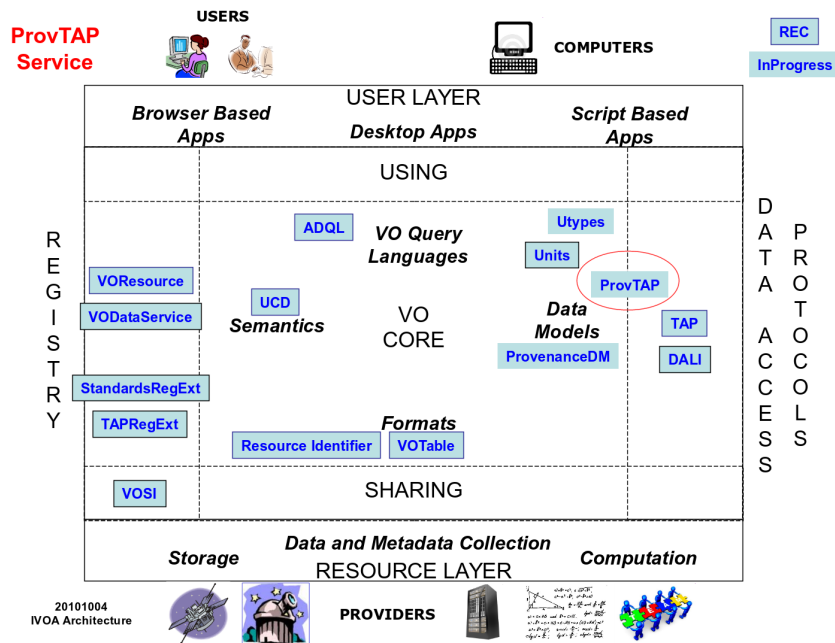


Figure 1: Architecture diagram for the ProvTAP specification. It is based on a view of IVOA Provenance data model and relies on TAP access technology. As such it is fully integrated in the IVOA framework

- To retrieve all entities identifiers attributed to curator agents:
 SELECT WasAttributedTo.wat_entity FROM WasAttributedTo
 WHERE WasAttributedTo.wat_role = 'curator'

The result of the first ADQL query is the following.

```
<?xml version="1.0" encoding="UTF-8"?>
<VOTABLE version="1.2" xmlns="http://www.ivoa.net/xml/VOTable/v1.2"
  xmlns:ex="http://www.example.com/provenance"
  xmlns:ivo="http://www.ivoa.net/documents/rer/ivo/"
  xmlns:voprov="http://www.ivoa.net/documents/dm/provdm/voprov/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.ivoa.net/xml/VOTable/v1.2 http://www.ivoa.net/xml/VOTable/VOTable-1.2.xsd">
  <RESOURCE type="provenance">
    <DESCRIPTION>Provenance VOTable</DESCRIPTION>
    <TABLE >
      <FIELD id="a_ID" name="a_id" ucd="meta.id" utype="voprov:Activity.id" arraysize="*" datatype="char"/>
      <FIELD id="a_nameID" name="a_name" ucd="meta.title" utype="voprov:Activity.name" arraysize="*" datatype="ch
      <FIELD id="a_startID" name="a_startTime" ucd="time.start" utype="voprov:Activity.startTime" arraysize="*" d
      <FIELD id="a_endID" name="a_endTime" ucd="time.start" utype="voprov:Activity.endTime" arraysize="*" datatyp
      <FIELD id="a_commentID" name="a_comment" ucd="meta.description" utype="voprov:Activity.comment" arraysize="
      <FIELD id="a_descID" name="a_description" ucd="meta.id" utype="voprov:Activity.description_id" datatype="
    <DATA>
      <TABLEDATA>
        <TR>
          <TD>act:CDS/P/HI4PI/NHI</TD>
          <TD>Generation of HI4PI NHI HiPS</TD>
          <TD>2011-02-14T12:00</TD>
        </TR>
      </TABLEDATA>
    </TABLE>
  </RESOURCE>
</VOTABLE>
```

```

        <TD>2011-02-14T12:00</TD>
        <TD>Generation of HI4PI NHI survey (full-sky HI column density distribution) HiPS</TD>
        <TD>hips-gen15</TD>
    </TR>
</TABLEDATA>
</DATA>
</TABLE>
<INFO name="QUERY_STATUS" value="OK"/>
</RESOURCE>
</VOTABLE>

```

3 VOSI availability and capabilities

According to the DALI specification for VO services , a provenance service implementing ProvTAP must provide a VOSI availability interface as well as a capabilities interface with entries for ProvTAP. The `standardIds` for these provenance interfaces are:

```
ivo://ivoa.net/std/ProvenanceDM#ProvTAP-1.0
```

The capability for a TAP service to support the Provenance DM is expressed by the `dataModel` element as:

```

<dataModel ivoId="ivo://ivoa.net/std/ProvenanceDM-1.0">
  ProvenanceDM-1.0
</dataModel>

```

For ProvTAP, the VOSI tables interface must also be provided.

4 VOTable serialization for provenance metadata and ProvTAP TAP schema

The table and column definitions available in the TAP schema, including their metadata (`unit`, `ucd`, `utype`, `datatype`, `description`) exactly define the VOTable mapping of the IVOA Provenance data model (see table and column definitions and metadata).

Below is a serialisation format mapping the Provenance data model into VOTable. This format is useful to provide full serialisation of Provenance metadata for an archive or data reduction workflow in VOTable. It can also be used to build the TAP schema of a ProvTAP service.

4.1 Tables in the VOTable serialisation or in the ProvTAP TAP schema

The following table gives the full list of tables necessary to map the IVOA Provenance data model. The contents of the ‘name’ and ‘utype’ columns of this table are normative.

Name	utype	Description	status
Entity	voprov:Entity	instances of <i>Entity</i> class	Mandatory
ValueDescription	voprov:ValueDescription	instances of <i>ValueDescription</i> class	Mandatory
DatasetDescription	voprov:DatasetDescription	instances of <i>DatasetDescription</i> class	Mandatory
Activity	voprov:Activity	instances of <i>Activity</i> class	Mandatory
ActivityDescription	voprov:ActivityDescription	instances of <i>ActivityDescription</i> class	Mandatory
Agent	voprov:Agent	instances of <i>Agent</i> class	Mandatory
Parameter	voprov:Parameter	instances of <i>Parameter</i> class	Optional
ParameterDescription	voprov:ParameterDescription	instances of <i>ParameterDescription</i>	Optional
ConfigFile	voprov:ConfigFile	instances of <i>ConfigFile</i> class	Optional
ConfigFileDescription	voprov:ConfigFileDescription	instances of <i>ConfigFileDescription</i>	Optional
Used	voprov:Used	instances of <i>Used</i> relationships	Mandatory
UsageDescription	voprov:UsageDescription	instances of <i>UsageDescription</i> relationships	Optional
GenerationDescription	voprov:GenerationDescription	instances of <i>GenerationDescription</i> relationships	Optional
WasGeneratedBy	voprov:WasGeneratedBy	instances of <i>WasGeneratedBy</i> relationships	Mandatory
WasAssociatedWith	voprov:WasAssociatedWith	instances of <i>WasAssociatedWith</i> relationships	Mandatory
WasAttributedTo	voprov:WasAttributedTo	instances of <i>WasAttributedTo</i> relationships	Mandatory
WasConfiguredBy	voprov:WasConfiguredBy	instances of <i>WasConfiguredBy</i> relationships	Optional
WasDerivedFrom	voprov:WasDerivedFrom	instances of <i>WasDerivedFrom</i> relationships	Optional
WasInformedBy	voprov:WasInformedBy	instances of <i>WasInformedBy</i> relationships	Optional
HadMember	voprov:HadMember	instances of <i>HadMember</i> relationships	Optional

Table 1: List of tables in the TAP schema

When the ProvTAP service is requested by generic ADQL queries involving optional tables which it doesn't include, it should not fail and the response MUST be a table headed by the queried columns names and containing no data. An example of this is a query constraining the unit or ucd of a parameter sent to a service not implementing the configuration classes.

4.2 Detailed table description

Tables 2 to 21 provide full descriptions of the FIELDS of each of the mapping tables. The contents of all these tables' columns are normative. For convenience, in all these tables an arrow (\rightarrow) indicates columns that contain references (foreign keys) to other tables. The column names are prefixed by a short-cut for the table name in order to ensure unique column names in queries with table joins for ProvTAP.

Name	ucd	utype	datatype	status
e_id	meta.id	voprov:Entity.id	char	M
e_name	meta.title	voprov:Entity.name	char	O
e_location	meta.ref.url	voprov:Entity.location	char	O
e_generated	time.start	voprov:Entity.generatedAtTime	char	O
e_invalidated	time.stop	voprov:Entity.invalidatedAtTime	char	O
e_comment	meta.description	voprov:Entity.comment	char	O
e_classtype	meta.code.class	voprov:Entity.classtype	char	M
e_value	stat.value	voprov:Entity.value	char	O
→ e_description	meta.id	voprov:Entity.description_id	reference	O

Table 2: Column description for Entity table. The e_classtype column may have the following two values : "dataset" and "value" given for instance using the VOTable OPTION element.

ProvTAP is actually dealing with two different kind of entities: "datasets" and "values". According to the value of the classtype attribute the description is given either by an instance of the "dataset description" class, or by an instance of the "value description" class.

Name	ucd	utype	datatype	status
dd_id	meta.id	voprov:DatasetDescription.id	char	M
dd_name	meta.title	voprov:DatasetDescription.name	char	O
dd_description	meta.description	voprov:DatasetDescription.description	char	M
dd_doculink	meta.ref.url	voprov:DatasetDescription.doculink	char	O
dd_type	meta.code.class	voprov:DatasetDescription.type	char	M
dd_subtype	meta.code.class	voprov:DatasetDescription.subtype	char	M
dd_content	meta.description	voprov:DatasetDescription.contentType	char	M

Table 3: Column description for DatasetDescription table. The value of type string belongs to a set of possible literal values given for instance using the VOTable OPTION element. This list may include the following words "data", "document", "vizationalization", "device", etc...

Name	ucd	utype	datatype	status
vd_id	meta.id	voprov:VaueDescription.id	char	M
vd_name	meta.title	voprov:ValueDescription.name	char	O
vd_description	meta.description	voprov:ValueDescription.description	char	M
vd_doculink	meta.ref.url	voprov:ValueDescription.doculink	char	O
vd_type	meta.code.class	voprov:ValueDescription.type	char	M
vd_subtype	meta.code.class	voprov:ValueDescription.subtype	char	O
vd_valueType	meta	voprov:ValueDescription.valueType	char	M
vd_unit	meta.unit	voprov:ValueDescription.unit	char	O
vd_ucd	meta.ucd	voprov:ValueDescription.ucd	char	O
vd_utype	meta	voprov:ValueDescription.utype	char	O
vd_min	stat.min	voprov:ValueDescription.min	char	O
vd_max	stat.max	voprov:ValueDescription.max	char	O
vd_options	meta	voprov:ValueDescription.doculink	char	O
vd_default	meta	voprov:ValueDescription.doculink	char	O

Table 4: Column description for ValueDescription table. The value of type, subtype and valueType strings belong to a set of possible literal values given for instance using the VOTable OPTION element. For type this list may include the following words "data", "document", "visualization", "device", etc... ValueType list should include all valid VOTable datatypes.

Name	ucd	utype	datatype	status
a_id	meta.id	voprov:Activity.id	char	M
a_name	meta.title	voprov:Activity.name	char	M
a_startTime	time.start	voprov:Activity.startTime	char	M
a_endTime	time.stop	voprov:Activity.endTime	char	M
a_comment	meta.description	voprov:Activity.comment	char	O
→ a_description	meta.id	voprov:Activity.description_id	char	O

Table 5: Column description for Activity table.

Name	ucd	utype	datatype	status
ad_id	meta.id	voprov:ActivityDescription.id	char	M
ad_name	meta.title	voprov:ActivityDescription.name	char	O
ad_version	meta	voprov:ActivityDescription.version	char	O
ad_description	meta.description	voprov:ActivityDescription.description	char	M
ad_doculink	meta.ref.url	voprov:ActivityDescription.doculink	char	O
ad_type	meta.code.class	voprov:ActivityDescription.type	char	O
ad_subtype	meta.code.class	voprov:ActivityDescription.subtype	char	O

Table 6: Column description for ActivityDescription table. The value of type and subtype strings belong to a set of possible literal values given for instance using the VOTable OPTION element. When applicable, one of the following type values should be used : "Observation", "Simulation", "Reduction", "Calibration", "Reconstruction", "Selection", "Analysis".

Name	ucd	utype	datatype	status
ag_id	meta.id	voprov:Agent.id	char	M
ag_name	meta.title	voprov:Agent.name	char	M
ag_type	meta.code.class	voprov:Agent.type	char	M
ag_comment	meta.description	voprov:Agent.comment	char	O
ag_email	meta.email	voprov:Agent.email	char	O
ag_affiliation	meta.	voprov:Agent.affiliation	char	O
ag_phone	meta.	voprov:Agent.phone	char	O
ag_address	meta.address	voprov:Agent.address	char	O
ag_url	meta.ref.url	voprov:Agent.url	char	O

Table 7: Column description for agent table. The value of type string belongs to a set of possible literal values ("Person", "Organization" and "SoftwareAgent") given for instance using the VOTable OPTION element.

Name	ucd	utype	datatype	status
p_id	meta.id	voprov:Parameter.id	char	M
p_name	meta.title	voprov:Parameter.name	char	M
p_value	stat.value	voprov:Parameter.value	param dependent	M
→ p_description	meta.id	voprov:Parameter.parameterDescription_id	reference	M

Table 8: Column description for Parameter table.

Name	ucd	utype	datatype	status
cf_id	meta.id	voprov:ConfigFile.id	char	O
cf_name	meta.title	voprov:ConfigFile.name	char	O
cf_location	meta.ref.url	voprov:ConfigFile.location	char	O
cf_comment	meta.description	voprov:ConfigFile.comment	char	O
→ cf_description	meta.id	voprov:ConfigFile.ConfigFileDescription_id	reference	O

Table 9: Column description for Configuration File.

Name	ucd	utype	datatype	status
wcb_artefact	meta.code	voprov:WasConfiguredBy.artefactType	char	M
→ wcb_configfile	meta.id	voprov:WasConfiguredBy.ConfigFile_id	reference	O
→ wcb_parameter	meta.id	voprov:WasConfiguredBy.parameter_id	reference	O
→ wcb_activity	meta.id	voprov:WasConfiguredBy.activity_id	reference	M

Table 10: Column description for WasConfiguredBy relationship table.

Name	ucd	utype	datatype
→ pd_activitydescription	meta.id	voprov:ParameterDescription.activityDescription_id	char
pd_id	meta.id	voprov:ParameterDescription.id	char
pd_name	meta.title	voprov:ParameterDescription.name	char
pd_description	meta.description	voprov:ParameterDescription.description	char
pd_doculink	meta.ref.url	voprov:ParameterDescription.doculink	char
pd_valueType	meta	voprov:ParameterDescription.valueType	char
pd_unit	meta.unit	voprov:ParameterDescription.unit	char
pd_ucd	meta.ucd	voprov:ParameterDescription.ucd	char
pd_utype	meta	voprov:ParameterDescription.utype	char
pd_min	stat.min	voprov:ParameterDescription.min	-
pd_max	stat.max	voprov:ParameterDescription.max	-
pd_options	meta	voprov:ParameterDescription.options	-
pd_default	meta	voprov:ParameterDescription.default	-

Table 11: Column description for ParameterDescription table. The datatype for min, max and options depends from the datatype attribute value !

Name	ucd	utype	datatype	status
cfid_id	meta.id	voprov:ConfigFileDescription.id	char	M
cfid_name	meta.title	voprov:ConfigFileDescription.name	char	M
cfid_doculink	meta.ref.url	voprov:ConfigFileDescription.doculink	char	M
cfid_content	meta.code.mime	voprov:ConfigFileDescription.contentType	char	M
cfid_description	meta.description	voprov:ConfigFileDescription.description	char	O
cfid_type	meta.code.class	voprov:ConfigFileDescription.type	char	O
cfid_subtype	meta.code.class	voprov:ConfigFileDescription.subtype	char	O

Table 12: Column description for ConfigFileDescription table.

Name	ucd	utype	datatype	status
→ u_entity	meta.id	voprov:Used.entity_id	reference	M
→ u_activity	meta.id	voprov:Used.activity_id	reference	M
→ u_usedDescription_id	meta.id	voprov:Used.usedDescription_id	reference	O
u_role	meta.code.class	voprov:Used.role	char	O
u_time	time.start	voprov:Used.time	char	M

Table 13: Column description for Used relationship table.

Name	ucd	utype	datatype	status
→ ud_id	meta.id	voprov:UsageDescription. id	char	M
→ ud_entityDescription	meta.id	voprov:UsageDescription. entityDescription_id	char	M
→ ud_activityDescription	meta.id	voprov:UsageDescription. activityDescription_id	char	M
ud_role	meta.code.class	voprov:UsageDescription.role	char	M
ud_type	meta.code.class	voprov:UsageDescription.type	char	M

Table 14: Column description for UsedDescription relationship table. The value of the role string is a free literal value.

Name	ucd	utype	datatype	status
→ wgb_entity	meta.id	voprov:WasGeneratedBy.entity_id	char	M
→ wgb_activity	meta.id	voprov:WasGeneratedBy.activity_id	char	M
→ wgb_generationDescription	meta.id	voprov:WasGeneratedBy.GenerationDescription_id	char	O
wgb_role	meta.code.class	voprov:WasGeneratedBy.role	char	O

Table 15: Column description for WasGeneratedBy relationship table. The value of a role string belongs to a set of possible free literal values given for instance using the VOTable OPTION element.

Name	ucd	utype	datatype	status
→ gd_id	meta.id	voprov:GenerationDescription.id	char	M
→ gd_entityDescription	meta.id	voprov:GenerationDescription.entityDescription_id	char	M
→ gd_activityDescription	meta.id	voprov:GenerationDescription.activityDescription_id	char	M
gd_role	meta.code.class	voprov:GenerationDescription.role	char	M
gd_type	meta.code.class	voprov:GenerationDescription.type	char	M

Table 16: Column description for GenerationDescription relationship table. The value of a role string is a free literal value. The values of the type attribute is taken from the mandatory list "Main", "Calibration", "Preview", "Setup", "Quality", "Log", "Context" and can be given using the VOTable OPTION element.

Name	ucd	utype	datatype	status
→ waw_agent	meta.id	voprov:WasAssociatedWith.agent_id	char	M
→ waw_activity	meta.id	voprov:WasAssoatciatedWith.activity_id	char	M
waw_role	meta.code.class	voprov:WasAssociatedWith.role	char	O

Table 17: Column description for WasAssociatedWith relationship table. The value of role string belongs to a set of possible literal values given for instance using the VOTable OPTION element. This should be taken among : "Author", "Contributor", "Coordinator", "Creator", "Curator", "Editor", "Funder", "Investigator", "Observer", "Operator", "Provider", "Publisher".

Name	ucd	utype	datatype	status
→ wat_entity	meta.id	voprov:WasAttributedTo.entity_id	char	M
→ wat_agent	meta.id	voprov:WasAttributedTo.agen_id	char	M
wat_role	meta.code.class	voprov:WasAttributedTo.role	char	M

Table 18: Column description for WasAttributedTo relationship table. The value of role string belongs to a set of possible literal values given for instance using the VOTable OPTION element. This should be taken among : "Author", "Contributor", "Coordinator", "Creator", "Curator", "Editor", "Funder", "Investigator", "Observer", "Operator", "Provider", "Publisher".

Name	ucd	utype	datatype	status
→ wib_informant	meta.id	voprov:WasInformedBy.informant_id	char	M
→ wib_informed	meta.id	voprov:WasInformedBy.informed_id	char	M

Table 19: Column description for WasInformedBy relationships table.

Name	ucd	utype	datatype	status
→ wdf_usedEntity	meta.id	voprov:WasDerivedFrom.usedEntity_id	char	M
→ wdf_generatedEntity	meta.id	voprov:WasDerivedFrom.generatedEntity_id	char	M

Table 20: Column description for WasDerivedFrom relationship table.

Name	ucd	utype	datatype	status
→ hm_collection	meta.id	voprov:HadMember.collection_id	char	M
→ hm_member	meta.id	voprov:HadMember.member_id	char	M

Table 21: Column description for HadMember relationships table.

Appendix A TAP schema in xml

The ProvTAP TAP schema will be found on the IVOA xml document repository (currently available at:

<https://wiki.ivoa.net/internal/IVOA/ObservationProvenanceDataModel/ProvTAP-schema-new.xml>)

Appendix B TAP service content exemple. Database dump in VOTable.

The exemple available at URL :

<https://wiki.ivoa.net/internal/IVOA/ObservationProvenanceDataModel/ProvHiPS.xml> provides information on data collections used or gathered by CDS in the context of Aladin. It starts from Schmidt plate collections and goes up to HiPS datasets and collections. These metadata are organized below into a set of related VOTables consistent with the ProvTAP TAP schema.

Appendix C Changes from Previous Versions

mireille: changed stopTime everywhere in endTime and ucd "time.stop" in "time.end" to ensure compatibility with provDM tables.

Bibliography

Bonnarel, F., Riebe, K., Servillat, M., Louys, M., Nullmeier, M., Rothmaier, F., Sanguillon, M. and Group", I. D. M. W. (2018), 'IVOA provenance simple access protocol (provsap)', IVOA DM WG internal draft.
<http://volute.g-vo.org/svn/trunk/projects/dm/provenance/ProvSAP/>

Demleitner, M., Harrison, P., Molinaro, M., Greene, G., Dower, T. and Perdikeas, M. (2013), 'IVOA registry relational schema', IVOA Working Draft.
<http://www.ivoa.net/documents/RegTAP/>

Dowler, P., Demleitner, M., Taylor, M. and Tody, D. (2013), 'Data access layer interface, version 1.0', IVOA Recommendation.
<http://www.ivoa.net/documents/DALI/20131129/>

Dowler, P., Rixon, G. and Tody, D. (2010), 'Table access protocol version 1.0', IVOA Recommendation.
<http://www.ivoa.net/documents/TAP>

Louys, M., Bonnarel, F., Schade, D., Dowler, P., Micol, A., Durand, D., Tody, D., Michel, L., Salgado, J., Chilingarian, I., Rino, B., de Dios Santander, J. and Skoda, P. (2011), 'Observation data model core components and its implementation in the Table Access Protocol, version 1.0', IVOA Recommendation.
<http://www.ivoa.net/documents/ObsCore/20111028/REC-ObsCore-v1.0-20111028.pdf>

Servillat, M., Riebe, K., Boisson, C., Bonnarel, F., Galkin, A., Louys, M., Nullmeier, M., Renaud-Tinacci, N., Sanguillon, M. and Streicher, O. (2019), 'IVOA provenance data model', IVOA Proposed Recommendation.
<http://www.ivoa.net/documents/ProvenanceDM/>