



Publication Year	2021
Acceptance in OA	2022-05-31T12:09:14Z
Title	Observation Locator Table Access Protocol Version 1.0
Authors	Salgado, Jesús, Ibarra, Aitor, Ehle, Matthias, Gabriel, Carlos, Kretschmar, Peter, Kuulkers, Erik, Merín, Bruno, Ness, Jan-Uwe, Salazar, Emilio, Saxton, Richard, Cecconi, Baptiste, Foster, Karl, DEMLEITNER, MARKUS, Dempsey, James, MOLINARO, Marco, Sánchez, Celia, Taylor, Mark, Díaz Trigo, María, Fernández, Mónica, Kennea, Jaime, Kettenis, Mark, Matt, Giorgio, Osborne, Julian, de Oña Wilhelmi, Emma, Salbol, Edward J., Sivakoff, Gregory, Tao, Lian, Tohuvavohu, Aaron, Tibbetts, Michael, Workman, Bill
Handle	http://hdl.handle.net/20.500.12386/32132



*International
Virtual
Observatory
Alliance*

Observation Locator Table Access Protocol Version 1.0

IVOA Recommendation 2021-07-24

Working group

Data Model Working Group

This version

<http://www.ivoa.net/documents/ObsLocTAP/20210724>

Latest version

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

Previous versions

Author(s)

Aitor Ibarra, Jesús Salgado, Matthias Ehle, Carlos Gabriel, Peter Kretschmar, Erik Kuulkers, Bruno Merín, Jan-Uwe Ness, Emilio Salazar, Richard Saxton, Baptiste Ceconi, Karl Foster, Markus Demleitner, James Dempsey, Marco Molinaro, Celia Sánchez, Mark Taylor, María Díaz Trigo, Jaime Kennea, Mark Kettenis, Giorgio Matt, Julian Osborne, Emma de Oña Wilhelmi, Edward J. Salbol, Gregory Sivakoff, Lian Tao, Aaron Tohuvavohu, Bill Workman

Editor(s)

Jesús Salgado, Aitor Ibarra

Abstract

The Observation Locator Table Access Protocol (ObsLocTAP) defines a data model for scheduled observations and a method to run queries over compliant data, using several Virtual Observatory technologies.

The data model builds on the ObsCore data model, removing elements associated with dataset access that are not available during the planning phase. In this way, this standard is focused on access to metadata related to the planning of a certain observatory, more than on access to the scientific data products. Also, the data model will be focused on discovery of planned observations, which is very useful information for multi-wavelength coordination observations, re-planning information propagation, follow-up of Targets of Opportunity alerts, preparation of proposals, etc.

As with ObsCore, a serialisation into a relational table is defined, which allows users to run complex queries using the IVOA Table Access Protocol. The document also prescribes how to register and discover ObsLocTAP services.

Status of this document

This document has been reviewed by IVOA Members and other interested parties, and has been endorsed by the IVOA Executive Committee as an IVOA Recommendation. It is a stable document and may be used as reference material or cited as a normative reference from another document. IVOA's role in making the Recommendation is to draw attention to the specification and to promote its widespread deployment. This enhances the functionality and interoperability inside the Astronomical Community.

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

Contents

1	Overview	5
2	Use cases	6
2.1	Common tooling for observers across observatories	7
2.2	Avoiding unnecessary proposals for observation time	7
2.3	Support for weighting proposals	7
2.4	Identification of planned observations in a certain spectral range for a certain astronomical event	7
2.5	Follow-up of Targets of Opportunity	8
2.6	Modification of the plans due to external conditions or anomalies	8

3	Observation Locator Elements	8
3.1	Observation Locator data model	9
3.2	Compulsory metadata	12
3.3	Implementation of ObsLocTAP as a TAP service	13
3.4	TAP Schema	13
4	Use Cases as TAP/ADQL queries	15
4.1	Discovery of observations scheduled or planned for a certain observatory	15
4.2	Identification of planned observations in a certain spectral range for a certain astronomical object	15
4.3	Modification of the plans due to external conditions or anomalies	16
4.4	Follow-up of Target of Opportunities	16
5	Registering ObsLocTAP services	17
A	Changes from Previous Versions	18
B	Registration Resource Example	19

Acknowledgments

The authors acknowledge the comments from Observatories members and from the IVOA members in general.

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.

Role within the VO Architecture

Fig. 1 shows the role ObsLocTAP plays within the IVOA architecture (Arviset and Gaudet et al., 2010). ObsLocTAP reuses various model elements (including geometrical entities) of ObsCore and, as ObsCore itself, relies on the TAP specification to deploy its relational content. Also, queries

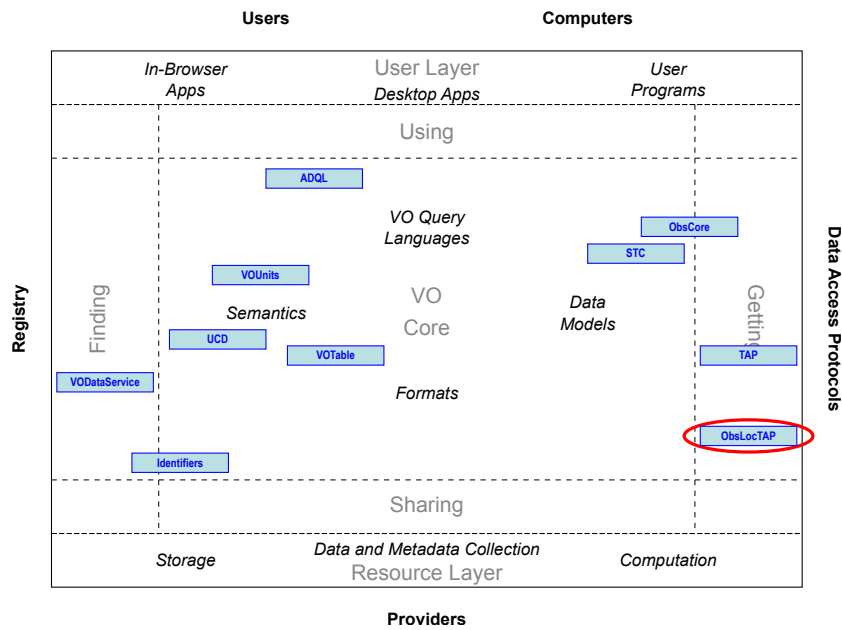


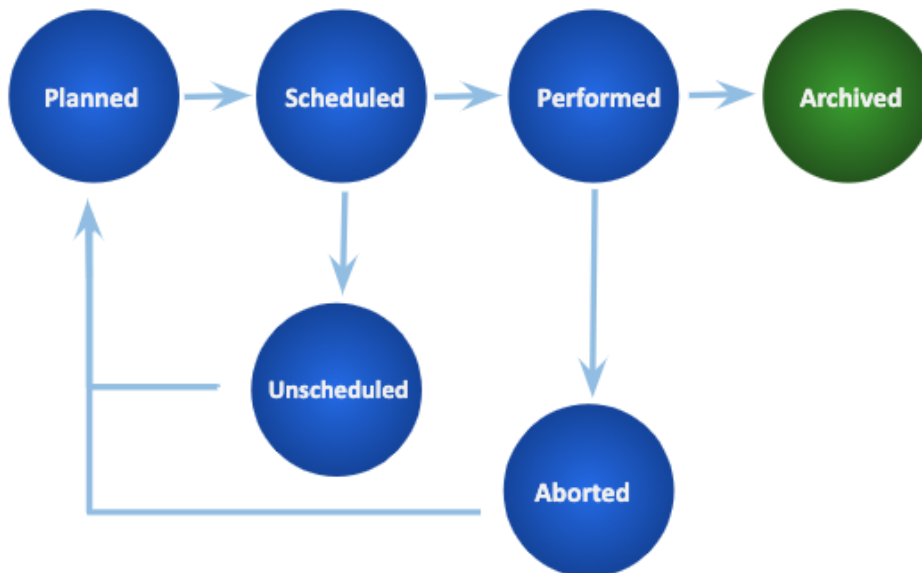
Figure 1: Architecture diagram for this document

are written following ADQL as the query language and the default result is defined in VOTable format, including units in a VOUnits compatible format, UCDs to describe columns and Identifiers to point to the data model. Finally, ObsLocTAP services can be registered by making use of the VODataService standard format.

1 Overview

The **Observation Locator Table Access Protocol** (ObsLocTAP henceforth) specifies in a standard format, services to retrieve information about planned, scheduled and performed observations of a given target (or coordinates) for a given astronomical observatory based on the existing ObsCore data model. This standard does not describe the access to data obtained after the processing of the observational activity, as that is the goal of ObsCore (archived observations), although the discovery could be done in a similar way. Therefore, although there is some overlap on the data model described in both standards, entities, use cases and communities are different.

In order to standardize the workflow of one observation, status is defined as follows:



- **Planned:** a possible observation, in most of the cases coming from a certain proposal, has been identified. There is not yet an association to a certain time period when the observation can be executed.
- **Scheduled:** mission planners allocate a certain period when the observation can be executed. Also, some changes could be expected on coordinates (due to some refinement) and other fields.
- **Unscheduled:** the observation has been removed from the schedule. The observation could be recovered as planned into the planning log.

- **Performed:** the observation has been performed successfully. This is only at the operational level as there is not guarantee of scientific results.
- **Aborted:** the observation has not been correctly performed. The observation could be recovered as planned into the planning log.
- **Archived:** the observation has data available and can be found in a science archive. This archived phase is the main target of the ObsCore specification, so it can be considered out of the scope of the present specification.

This new standard would allow client applications to use a standardised VO service to aggregate and display the information of former and future observations. This new standard will help in the preparation of multi-observatory multi-instrument observation campaigns, whose coordination needs certain standards for efficient communication.

The ObsLocTAP interface makes use of the IVOA TAP (Dowler and Rixon et al. 2019) to access observational metadata. The IVOA TAP protocol is a powerful and quite flexible system, easily adapted to a wide variety of use cases simply by defining agreed-upon table structures. Users can write complex queries against these using the common Astronomical Data Query Language (ADQL; Osuna and Ortiz et al. 2008)

There are several open source TAP implementations that could be used by the community (minimizing the implementation effort for the observatories) and it has been efficiently used in operational archives like, e.g., the Gaia Archive.

In the current document, we will focus on the description of the Data Model, tables and use cases. No deviation from standard TAP or ADQL is foreseen, where TAP defines the mechanisms to exchange queries and data as well as error messages and ADQL the query language syntax.

2 Use cases

The planning process can be usually described as follows (although observatories can have different procedures to fulfil this process):

- The observatory receives observatory proposals from scientists that are reviewed by an expert panel (e.g., OTAC = Observing Time Allocation Committee).
- Proposals are ranked by their scientific relevance.
- Proposals are related to one or more observations that are inserted into the observation planning system of the observatory.

- Observation planners decide which observations can be scheduled in the short-medium plan trying to maximize the relevance of a certain observation period (e.g., per night or orbit revolution) and taking into account the constraints of the observatory (e.g., visibility of the object, geometrical constraints like the Sun or the Earth for space-based observatories, etc).
- In case of unexpected events like, e.g., targets of opportunity, the scheduled plan could be replaced by another one modifying the short or medium-term plans.

Standardizing the way to publish this short/medium plan through ObsLocTAP services provides higher scientific impact of the observing time, follow-up of astronomical events and better cooperation between observatories. We present here a selected list of use cases. Later in the document, we will show how these use cases can be fulfilled using the current specification:

2.1 Common tooling for observers across observatories

Standardization of the publication of observing plans will facilitate the implementation of a tool that lets observers monitor the progress of their proposed observations that works for all observatories that adopt ObsLocTAP. This kind of application has been requested by the community for a long time. Ideally, that would obviate custom, per-observatory web forms, saving work for observatories and observers alike.

2.2 Avoiding unnecessary proposals for observation time

An astronomer wants to propose an observation and can, before doing so, globally determine if a possibly suitable observation is already scheduled. This might prevent costly duplicate observations.

2.3 Support for weighting proposals

The bodies (e.g., OTAC) that decide on the observation schedule can use the observation schedules of their own and other observatories to raise or lower priorities on a given proposal. This might even take into account observation histories, giving demerits to possibly unnecessary repeated observations.

2.4 Identification of planned observations in a certain spectral range for a certain astronomical event

A scientist could be interested in the planned observations for a certain astronomical object at different spectral ranges. For example, this is commonly done in the GRB (Gamma Ray Burst) events follow-up, where the event is

decreasing in energy and multiple observations must be done in decreasing energy order by different observatories to fully characterize the event.

By using ObsLocTAP services, users could discover and propose sequential observations that will allow the future SED characterization of this astronomical event.

2.5 Follow-up of Targets of Opportunity

Observing plans can be changed very fast due to targets of opportunity (ToOs).

There are two different types of ToOs in astronomy:

- Unpredictable ToOs: Astronomical events that require immediate or almost immediate observations and that may even require coordination between different observatories.
- Predictable ToOs: These astronomical events are related (not always) to known transient phenomena or due to coordinated observations of targets special interest.

For the first type, the short-term plan can be affected in a very short time scale, as when they trigger follow-up observations of a certain astronomical event.

VOEvent initiatives are focused on the announcement of the astronomical event and in the propagation of the information obtained due to follow-up observations, but only when the observations are already executed. By the implementation of ObsLocTAP services, the decision how short-term observing plans are modified can be published live to the scientific community.

2.6 Modification of the plans due to external conditions or anomalies

Usually, observatory plans can be modified due to various reasons. They could be changed by anomalies in the instruments, astronomical events, external conditions, etc. For ground-based observations, atmospheric conditions could imply the modification of the planning, whereas for space-based observatories, radiation due to solar flares could force observations to be stopped.

It is difficult for the scientist to follow-up the status of the plan. This could be fulfilled by the implementation of ObsLocTAP services.

3 Observation Locator Elements

3.1 Observation Locator data model

In the IVOA ObsCore specification, two main data models are connected to describe the data model behind most of the use cases needed to discover observed datasets:

- **Characterization:** Provides a description of metadata associated with a certain observation. In particular, the different axes (e.g., the spatial axis, the time axis, spectral axis...) are used to describe the coverage of the observation on different observables.
- **ObsDataSet:** Provides support to discover data associated with a certain performed observation.

In the case of planned (not executed) observations, only the Characterization DM could be used, as the observations covered are not yet performed and hence most of the metadata associated with the ObsDataSet will be empty. Also, the data produced after the processing of the observational metadata could be not present yet for recently executed observations and, in other cases, data could be present only within the mission archive but not in the planning system.

Although it is correctly set in the ObsCore DM that the association multiplicity between the Characterization DM and ObsDataSet is $0..*$, the ObsCore specification clearly considers that metadata related to the data access is set for all the records. In the present document the data access metadata will not be included.

Also, although some of the elements could be known a priori, it is difficult to estimate the number of elements in the different axes (space, time, spectral, ...). This implies that this information will not be present for most of the planning lists. This is why this information has been removed as well from the present data model.

That implies the data model for the use cases described in the present document is reduced to:

Column Name	Description	Constraint
t_planning	Time in MJD when this observation has been added or modified into the planning log	not null
target_name	Astronomical object observed, if any	
obs_id	Observation ID	not null
obs_collection	Name of the data collection	
s_ra	Central right ascension, ICRS	
s_dec	Central declination, ICRS	
s_fov	Diameter (bounds) of the covered region	
s_region	Sky region covered by the data product (expressed in ICRS frame)	
s_resolution	Spatial resolution of data as FWHM	
t_min	Start time in MJD	not null for execution_status Scheduled or Performed
t_max	Stop time in MJD	not null for execution_status Scheduled or Performed
t_exptime	Total exposure time	not null for execution_status Scheduled or Performed
t_resolution	Temporal resolution FWHM	
em_min	Start in spectral coordinates	
em_max	Stop in spectral coordinates	
em_res_power	Spectral resolving power	
o_ucd	UCD of observable (e.g., phot.flux.density, phot.count, etc.)	
pol_states	List of polarization states or NULL if not applicable	
pol_xel	Number of polarization samples	
facility_name	Name of the facility used for this observation	not null
instrument_name	Name of the instrument used for this observation	
t_plan_exptime	Planned or scheduled exposure time	
category	Observation category. One of the following values: Fixed, Coordinated, Window, Other	not null
priority	Priority level { 0, 1, 2}	not null
execution_status	One of the following values: Planned, Scheduled, Unscheduled, Performed, Aborted	not null
tracking_type	One of the following values: Sidereal, Solar-system-object-tracking, Fixed-az-el-transit	not null

Table 1: ObsPlan columns description

ObsLocTAP services are TAP services that expose a table called obsplan. This is a mandatory feature. The obsplan table exposed by an ObsLocTAP service MUST include all the columns listed, even though null values may be permitted in some of the columns. Additional non-standard columns are permitted in a particular obsplan table implementation.

The purpose of the observation locator query is to allow users/clients to retrieve the information of planned observations of a particular target or sky coordinates. The most basic query parameters will be the sky coordinates (Right Ascension and Declination). Additional parameters may be used to customize the visibility checks.

As can be seen, most of the data model elements come from the IVOA Characterization data model and the exact description of these fields is present in the IVOA ObsCore standard (Louys and Tody et al. 2017).

s_region, as described on IVOA ObsCore standard, can be used to precisely specify the covered spatial region of a data product. The s_region geometrical field is allowed to be serialised according to DALI (Dowler and Demleitner et al. 2017) specification, as referenced e.g. on TAP (current combination is DALI 1.1 and TAP 1.1 but future evolution would be valid), or using a STC-S format described on the section 6 of TAP 1.0 (Tody and Micol et al. 2011). In the WHERE clause, s_region constraints will generally make use of ADQL geometries (CIRCLE, POLYGON, POINT) and the geometry predicate functions (CONTAINS, INTERSECTS).

In case the exact final region cannot be specified (e.g., because an undefined position angle of the planned observation), s_region can be approximated by implementors as a CIRCLE with the center position as **s_ra** and **s_dec** and a radius of **s_fov**/2.

The planning time (**t_planning**) has been added in order to allow the discovery of the time when this observation (scientific entity) has been added (or modified) to the plan, in order to let clients ask for changes in the plan incrementally (i.e., without having to retrieve and compare locally the full set of future observations with a previously download one).

The planning exposure time (**t_plan_exptime**) has been added in order to show any possible inconsistency between the scheduled exposure time and the real performed exposure time (**t_exptime**). If an observation was performed without problems, both quantities usually have the same values. Any discrepancy between these two values could reflect problems or deviations between scheduled observations and performed observations or a more accurate value.

The **category** element has been added in order to help users to identify those observations that were scheduled in coordination with other facilities or those observations that have to be executed at a fixed time.

The **priority** element has been added in order to help users to identify those observations that were scheduled with a high priority (value=2) and

therefore cannot be removed from the planning or are hard to re-schedule. If the observation can be easily shifted to a different time (value=0), the user could query the service to identify this type of observations.

Execution status (**execution_status**) provides a description of the current status of the planned observation. Initially, observations are inserted as "Planned" into the system. Once the time when the observation is going to be executed is defined, the status changes to "Scheduled". A possible "Unscheduled" status is possible if the observation is removed from the planning due to any possible operations decision. The final state will be one of "Performed" or "Aborted". "Aborted" and "Unscheduled" observations could be removed from the planning table or they could return to the "Planned" status for a possible future schedule.

Tracking type (**tracking_type**) is a field of the obsplan table to characterize the tracking foreseen during the observation execution. In most cases, the value would be **Sidereal**. For these observations the pointing is maintained on the same position on the sky during all the observation period.

Other possible value is **Solar-system-object-tracking**, to be used when the pointing of the observatory varies to maintain the selected solar system target or any other moving object within the observatory field of view on the same position of the observatory point of view. In this case, the values for **s_ra** and **s_dec** could be defined as null or, if possible, a representative position within the expected sidereal movement. In this case, **target_name** would be an identifier of the solar system or moving object.

The third possible value is **Fixed-az-el-transit**, to be used when the observatory points to a fixed position in local azimuth-elevation coordinates, allowing for a target to cross the field of view. **s_ra** and **s_dec** would contain the value of the sidereal pointing at the time of the transit of the named target in **target_name**, which would be an identifier of the object to be observed.

3.2 Compulsory metadata

In contrast to normal ObsTAP services where most of the metadata is compulsory, in the case of ObsLocTAP, this cannot always be maintained. In some cases, the target associated with the observations could still be confidential information which needs to be hidden due to, e.g., observatory confidentiality constraints.

The same approach of hidden information status could also be applied to the instruments or configurations used so the metadata should be present as follows:

- For observations that requires some metadata to be hidden from the planning point of view, (e.g. the policy of a certain observatory is to hide the astronomical target of certain private proposals) the minimal

metadata to be shown for "Scheduled" and "Performed" observations is the time coverage so it is shown that this period has been blocked for a certain observation. That means that an entry should appear with a defined start time, end time and duration as relevant. The facility name should also be provided.

- If the astronomical target is still hidden, coordinates and target should be set to *null* in the obsplan table.
- Observational metadata about instrument, spectral coverage and polarization states should appear with a general default value, describing a non-detailed mode, but with values that are more general than the exact ones to be used during the real observation. For example, a general X-ray spectral coverage could appear in *em_min* and *em_max* for an X-ray observatory but not the exact details of the instrument mode that will be used for the confidential observation.
- Exact details of the observation could be updated whenever the observation passes from the hidden state to the public state from the metadata point of view.

3.3 Implementation of ObsLocTAP as a TAP service

The primary access path to ObsLocTAP information is through TAP (Dowler and Rixon et al. 2010) database queries. In order to fulfil some advanced use cases, ObsLocTAP services MUST provide, at least, the following ADQL (Osuna and Ortiz et al. 2008) geometrical features: CIRCLE, POLYGON, POINT, INTERSECTS and CONTAINS.

3.4 TAP Schema

The following table gives the entries for the `ivoa.obsplan` table in `TAP_SCHEMA.columns`

column_name	data type	units	utype	UCD
t_planning	(real)	d		
target_name	(string)		Target.name	meta.id;src
obs_id	(string)		DataID.observationID	meta.id
obs_collection	(string)		DataID.collection	meta.id
s_ra	(real)	deg	Char.SpatialAxis.Coverage.Location.Coord.Position2D.Value2.C1	pos.eq.ra
s_dec	(real)	deg	Char.SpatialAxis.Coverage.Location.Coord.Position2D.Value2.C2	pos.eq.dec
s_fov	(real)	deg	Char.SpatialAxis.Coverage.Bounds.Extent.diameter	phys.angSize;instr.fov
s_region	(region)		Char.SpatialAxis.Coverage.Support.Area	pos.outline;obs.field
s_resolution	(real)	arcsec	Char.SpatialAxis.Resolution.Refval.value	pos.angResolution
t_min	(real)	d	Char.TimeAxis.Coverage.Bounds.Limits.StartTime	time.start;obs.exposure
t_max	(real)	d	Char.TimeAxis.Coverage.Bounds.Limits.StopTime	time.end;obs.exposure
t_exptime	(real)	s	Char.TimeAxis.Coverage.Support.Extent	time.duration;obs.exposure
t_resolution	(real)	s	Char.TimeAxis.Resolution.Refval.value	time.resolution
em_min	(real)	m	Char.SpectralAxis.Coverage.Bounds.Limits.LoLimit	em.wl;stat.min
em_max	(real)	m	Char.SpectralAxis.Coverage.Bounds.Limits.HiLimit	em.wl;stat.max
em_res_power	(real)		Char.SpectralAxis.Resolution.ResolPower.refVal	spect.resolution
o_ucd	(string)		Char.ObservableAxis.ucd	meta.ucd
pol_states	(string)		Char.PolarizationAxis.stateList	meta.code;phys.polarization
pol_xel	(integer)		Char.PolarizationAxis.numBins	meta.number
facility_name	(string)		Provenance.ObsConfig.Facility.name	meta.id;instr.tel
instrument_name	(string)		Provenance.ObsConfig.Instrument.name	meta.id;instr
t_plan_exptime	(real)	s	Char.TimeAxis.Coverage.Support.Extent	time.duration;obs.exposure
category	(string)			
priority	(integer)			
execution_status	(string)			
tracking_type	(string)			

Table 2: TAP Schema description

The concrete types used in the database and in table output are up to the implementation, but the data type annotations above must result in VOTable column output with `datatype`, `xtype` and `arraysize` attributes as follows: (*real*) means floating point values (numeric datatype); (*integer*) means integer values (integer datatype); (*string*) means string values (character datatype and a suitable arraysize); and (*region*) means *either* an STC-S-format string as described in section 6 of TAP 1.0, *or* one of the region encodings such as Circle or Polygon described in DALI.

4 Use Cases as TAP/ADQL queries

In this section, we present ways to implement the use cases mentioned in the first part of this note as ADQL queries to an ObsLocTAP service.

4.1 Discovery of observations scheduled or planned for a certain observatory

Planned observations on a certain time period can be discovered using the following ADQL query:

```
SELECT *
FROM ivoa.obsplan
WHERE
    t_min < <end_time>
    AND t_max > <start_time>
```

Where `<start_time>` and `<end_time>` are the minimum and maximum times of the period to be requested in MJD. This query provides observations that overlap with a certain time range, e.g.,

```
SELECT *
FROM ivoa.obsplan
WHERE
    t_min < 58700
    AND t_max > 58500
```

To find observations planned between 2019-01-17 and 2019-08-05.

4.2 Identification of planned observations in a certain spectral range for a certain astronomical object

Sending the same ADQL query to all the available ObsLocTAP servers can discover planned observations that cover a certain spectral range:

```

SELECT *
FROM ivoa.obsplan
WHERE
    t_min < <end_time>
    AND t_max > <start_time>
    AND em_min < <spectral_coordinate_max>
    AND em_max > <spectral_coordinate_min>

```

where the overlap pattern has been used for the spectral coordinate. Where ObsLocTAP services give coverage information in their Registry records, services outside of the spectral area of interest can already be skipped based on resource metadata. More info on registration of ObsLocTAP services is provided in section 5.

The values of the spectral coordinate should be defined, as per the ObsCore data model defines, in meters. While this may be inconvenient for certain spectral ranges, this homogeneity allows the use of the same ADQL query to all the servers.

4.3 Modification of the plans due to external conditions or anomalies

Follow-up of changing observing plans can be tracked with a simple ADQL query sent to the ObsLocTAP server:

```

SELECT *
FROM ivoa.obsplan
WHERE
    t_planning > <saved_copy_time>
    AND t_max < <maximum_time_requested>

```

In this case, users can send a first query to the ObsLocTAP server and save the copy of the response. Then, a next query can be sent later to the service using the `t_planning` metadata to discover observations planned after the previous query, so changes in the planning can easily be identified. In order not to overload the server, another time constraint can be used on the, e.g., `t_max` metadata.

4.4 Follow-up of Target of Opportunities

Observing plans can be tracked with a simple ADQL query sent to the ObsLocTAP servers described in Section 4.3.

In order to exactly track changes in the plan due to a known target of opportunity, a geometrical constraint can be added to the ADQL query as follows:

```

SELECT *
FROM ivoa.obsplan
WHERE
  t_planning > <saved_copy_time>
  AND t_max < <maximum_time_requested>
  AND 1=INTERSECTS(s_region,
  CIRCLE('', <TOO_ra>, <TOO_dec>, <RADIUS>))

```

by adding an intersects operation of the observation field of view and a circle centered on the coordinates of a certain target of opportunity, $\langle TOO_ra \rangle$ and $\langle TOO_dec \rangle$, and with an uncertainty radius defined as $\langle radius \rangle$, e.g.,

```

SELECT *
FROM ivoa.obsplan
WHERE
  t_planning > 58500
  AND t_max < 58502
  AND 1=INTERSECTS(s_region,
  CIRCLE('', 114.8251, 1.6179, 0.016666))

```

where we are checking whether there are any newly planned observations from 58500 (2019-01-17) on the next two days (58502 or 2019-01-19) around the target PKS 0736+017, with a radius of 1 arcmin.

5 Registering ObsLocTAP services

ObsLocTAP services are registered as VODataService CatalogResources. They MUST have one or more auxiliary TAP capabilities pointing to the TAP service(s) at which the ivoa.obsplan table can be queried. They furthermore MUST have a tableset, with the ivoa.obsplan table's utype set to:

```
ivo://ivoa.net/std/obsloctap#table-1.0
```

A RegTAP 1.1 query discovering the access URLs of all ObsLocTAP services served through TAP 1.x services would then be:

```

SELECT ivo_id, res_title, access_url
FROM rr.resource
  NATURAL JOIN rr.capability
  NATURAL JOIN rr.interface
  NATURAL JOIN rr.res_table
WHERE
  table_utype='ivo://ivoa.net/std/obsloctap#table-1.0'
  AND standard_id LIKE 'ivoa.net/std/tap%'
  AND intf_role='std'

```

AND authenticated_only=0

Clients probably want to make sure they only query one access url per ivo-id
(this is in case multiple TAP capabilities are given for one resource).

A Changes from Previous Versions

Change	Section	Version
First Version	All	WD-20180713
Service renamed to ObsLocTAP	All	WD-20180713
Typos	All	WD-20180904
Data model re-definition (introduction of ivoa:obsplan)	All	WD-20190215
Better definition of “category” and “Priority”		
Adding better distinction between planned, scheduled, performed and archived observations	All	WD-20190215
Correction of s_fov as a circle radius. Possible use of more complex footprints as objects to be defined	All	WD-20190215
Introduction of s_region to cover more complex footprints	All	WD-20190909
Many typos correction	All	WD-20190909
Use cases upgrade	4	WD-20190909
Add registering description	5	WD-20190909
Migration to LaTeX/github	All	PR-20200514
Formating changes and typos	All	PR-20201007
Add comments from community workshops	All	PR-20201007
Add registration example	All	PR-20201007
Add semantics group TCG review suggestions	All	PR-20210118
Add operations group TCG review suggestions	All	PR-20210329
Add comments from TCG review suggestions, including Solar System IG, GWS WG, Registry WG and TCG chair	All	PR-20210528
tracking_type as compulsory as per request	All	PR-20210609
Recommendation 1.0	All	PR-20210724


```

<language>
  <name>ADQL</name>
  <version ivo-id="ivo://ivoa.net/std/ADQL#v2.0">2.0</version>
  <description>ADQL 2.0</description>
  <languageFeatures type="ivo://ivoa.net/std/TAPRegExt#features-adqlgeo">
    <feature>
      <form>BOX</form>
    </feature>
    <feature>
      <form>POINT</form>
    </feature>
    <feature>
      <form>CIRCLE</form>
    </feature>
    <feature>
      <form>POLYGON</form>
    </feature>
    <feature>
      <form>REGION</form>
    </feature>
    <feature>
      <form>CENTROID</form>
    </feature>
    <feature>
      <form>COORD1</form>
    </feature>
    <feature>
      <form>COORD2</form>
    </feature>
    <feature>
      <form>DISTANCE</form>
    </feature>
    <feature>
      <form>CONTAINS</form>
    </feature>
    <feature>
      <form>INTERSECTS</form>
    </feature>
    <feature>
      <form>AREA</form>
    </feature>
  </languageFeatures>
</language>
<outputFormat ivo-id="ivo://ivoa.net/std/TAPRegExt#output-votable-binary2">

```

```

        <mime>application/x-votable+xml;serialization=binary2</mime>
        <alias>votable</alias>
    </outputFormat>
    <outputFormat ivo-id="ivo://ivoa.net/std/TAPRegEXT#output-votable-td">
        <mime>application/x-votable+xml;serialization=tabledata</mime>
        <alias>votable_plain</alias>
    </outputFormat>
    <outputFormat>
        <mime>text/csv</mime>
        <alias>csv</alias>
    </outputFormat>
    <outputFormat>
        <mime>application/json</mime>
        <alias>json</alias>
    </outputFormat>
    <outputFormat>
        <mime>application/fits</mime>
        <alias>fits</alias>
    </outputFormat>
</capability>
<tableset>
<schema>
<name>ivoa</name>
<table>
    <name>ivoa.obsplan</name>
    <description>The Integral observations in the
        ObsLocTAP schema.</description>
    <utype>ivo://ivoa.net/std/obsloctap#table-1.0</utype>
    <column>
        <name>t_planning</name>
        <description>Time in MJD when this observation has been added
            or modified into the planning log</description>
        <unit>d</unit>
        <dataType xsi:type="vs:VOTableType">double</dataType>
        <flag>indexed</flag>
    </column>
    <column>
        <name>target_name</name>
        <description>Astronomical object observed, if any</description>
        <ucd>meta.id;src</ucd>
        <dataType arraysize="*" xsi:type="vs:VOTableType">char</dataType>
        <flag>indexed</flag>
    </column>
</table>
<!--

```

```

... more columns ...
-->
  <column>
    <name>execution_status</name>
    <description>Execution status provides a description of the current status
    of the planned observation. One of the following values:
Planned, Scheduled,
    Unscheduled, Performed, Aborted</description>
    <ucd/>
    <dataType xsi:type="vs:VOTableType">int</dataType>
    <flag>indexed</flag>
  </column>
</table>
</schema>
</tableset>
</ri:Resource>

```

References

- Arviset, C., Gaudet, S. and the IVOA Technical Coordination Group (2010), 'IVOA architecture', IVOA Note.
<http://www.ivoa.net/documents/Notes/IVOAArchitecture>
- Bradner, S. (1997), 'Key words for use in RFCs to indicate requirement levels', RFC 2119.
<http://www.ietf.org/rfc/rfc2119.txt>
- Dowler, P., Demleitner, M., Taylor, M. and Tody, D. (2017), 'Data access layer interface, version 1.1', IVOA Recommendation.
<http://www.ivoa.net/documents/DALI>
- Dowler, P., Rixon, G. and Tody, D. (2010), 'Table Access Protocol Version 1.0', IVOA Recommendation 27 March 2010, arXiv:1110.0497.
<http://doi.org/10.5479/ADS/bib/2010ivoa.spec.0327D>
- Dowler, P., Rixon, G., Tody, D. and Demleitner, M. (2019), 'Table Access Protocol Version 1.1', IVOA Recommendation 27 September 2019.
<https://ui.adsabs.harvard.edu/abs/2019ivoa.spec.0927D>
- 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.
<http://doi.org/10.5479/ADS/bib/2017ivoa.spec.0509L>
- Osuna, P., Ortiz, I., Lusted, J., Dowler, P., Szalay, A., Shirasaki, Y., Nieto-Santisteban, M. A., Ohishi, M., O'Mullane, W., VOQL-TEG

Group and VOQL Working Group. (2008), ‘IVOA Astronomical Data Query Language Version 2.00’, IVOA Recommendation 30 October 2008, arXiv:1110.0503.

<http://doi.org/10.5479/ADS/bib/2008ivoa.spec.10300>

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.

<http://doi.org/10.5479/ADS/bib/2011ivoa.spec.1028T>