



EPN2020-RI

EUROPLANET2020 Research Infrastructure

H2020-INFRAIA-2014-2015

Grant agreement no: 654208

Deliverable D6.10

VESPA 2nd set of standards documentation

Due date of deliverable: 30/6/2019

Actual submission date: 30/6/2019

Start date of project: 01 September 2015

Duration: 48 months

Responsible WP Leader: Stéphane Erard, LESIA / Observatoire de Paris

Project funded by the European Union's Horizon 2020 research and innovation programme		
Dissemination level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Service)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (excluding the Commission Services)	

Project Number	654208
Project Title	EPN2020 - RI
Project Duration	48 months: 01 September 2015 – 30 August 2019

Deliverable Number	D6.10
Contractual Delivery date	30.6.2019
Actual delivery date	30.6.2019
Title of Deliverable	VESPA 2 nd set of standards documentation
Contributing Work package (s)	WP6
Dissemination level	PU
Author (s)	S. Erard, B. Cecconi, P. Le Sidaner (and many contributors)

Abstract: This documentation set focuses on the current EPN-TAP protocol documentation. EPN-TAP is the main protocol to describe, search, and access Solar System data in the Virtual Observatory. The version provided here is a mature draft which will be proposed for validation at the IVOA steering committee in Oct. 2019.

Document history (to be deleted before submission to Commission)				
Date	Version	Editor	Change	Status
30/6/2019	1.0	S. Erard		

This documentation set focuses on the EPN-TAP protocol documentation.

EPN-TAP is the main protocol to describe, search, and access Solar System data in the Virtual Observatory (VO), and encompasses the whole range of Solar System studies (including heliophysics). As such, it is proposed for validation as a standard at IVOA (International Virtual Observatory Alliance).

The document version provided here is a mature Working Draft supported by the Solar System Interest Group of IVOA, which will be proposed for validation to the IVOA steering committee in Oct. 2019. After validation, IVOA Working Drafts become Proposed Recommendations (a period during which they are commented by the whole community, and checked for consistency with other standards), then Recommendations.

EPN-TAP is the first standard to include Solar System studies in the Virtual Observatory. It is also the foundation of the VESPA data distribution system, underlying the VESPA portal (<http://vespa.obspm.fr>) and providing interoperability to data services installed in Europlanet 2020 (currently 50 of them) and beyond.

It is completed with many actions led with VO developers to adapt existing visualization and processing tools to this type of data, and with similar collaborations in other fields: space agencies in the frame of IPDA (International Planetary Data Alliance), mapping tools with plugins and additions to GIS applications. These actions were presented in previous VESPA deliverable documents and activity reports.

Additional information and related discussions and documents are available on the VESPA public wiki site: <https://voparis-wiki.obspm.fr/>

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), 'Keywords for use in RFCs to indicate requirement levels', RFC 2119.

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

Dowler P., Rixon G., Tody D., Demleitner M. (2019) 'Table Access Protocol Version 1.1', IVOA Proposed recommendation

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



*International
Virtual
Observatory
Alliance*

EPN-TAP: Publishing Solar System Data to the Virtual Observatory

Version 2.0

IVOA Working Draft 2019-04-02

Working group

SSIG

This version

<http://www.ivoa.net/documents/epntap/20190402>

Latest version

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

Previous versions

This is the first public release

Author(s)

Stéphane Erard, Baptiste Cecconi, Pierre Le Sidaner

Editor(s)

Baptiste Cecconi, Markus Demleitner

Abstract

This document defines the EPN-TAP framework, which is using TAP with the EPNcore metadata dictionary. The EPNcore metadata dictionary defines the core components that are necessary to perform data discovery in the Solar System related science fields. It includes keywords to describe data products coverage (temporal, spectral, spatial, photometric), origin (instrument), content (target, physical parameters), access, reference, etc. Its implementation with TAP (Table Access Protocol) is presented, including service registration guidelines. Topical extension metadata dictionaries are also presented.

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	4
1.1	The EPNcore metadata dictionary	4
1.2	EPN-TAP rules applying to table structure and parameter usage	5
1.3	Architecture of EPN-TAP services	6
2	The EPNcore metadata keyword dictionary	8
3	EPNcore Table	13
3.1	Mandatory parameters	13
3.1.1	Granule references	13
3.1.2	Data Description	13
3.1.3	Target description	17
3.1.4	Axes	20
3.1.5	Data origin	25
3.1.6	Granule call-back info	26
3.2	Optional parameters	27
3.2.1	Data Access Reference	27
3.2.2	Supplementary descriptions	28
3.2.3	Description of coordinate frame	30
3.2.4	Target configuration & observing geometry	31
3.2.5	Vertical scales on planets	32
3.3	Extensions	33
3.3.1	Particle spectroscopy extension	33
3.3.2	Solar System Objects extension	33
3.3.3	Maps	34
3.3.4	Contributive works	34

3.3.5	Experimental spectroscopy	35
3.3.6	APIS extension	37
3.3.7	Events	37
3.3.8	Other extensions	38
A Changes from Previous Versions		38

Acknowledgments

EPNcore has been developed by the VESPA (Virtual European Solar and Planetary Access) team. VESPA has been designed in the frame of Europlanet 2020 Research Infrastructure (EPN2020RI) project, based on an assessment phase in the Europlanet Research Infrastructure (EPN-RI) programme, JRA4 work package (IDIS activity).

The EuroPlaNet-RI project was funded by the European Commission under the 7th Framework Program, grant 228319 “Capacities Specific Programme”. The Europlanet 2020 Research Infrastructure (EPN2020RI) project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 654208. Additional funding was provided in France by the CNRS / Action Spécifique Observatoire Virtuel and Programme National de Planétologie / INSU, as well as CNES, through the participation to IPDA activities.

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.

List of Acronyms

- ADQL Astronomical Data Query Language
- EPN Europlanet
- TAP Table Access Protocol
- UCD Unified Content Descriptors

- VESPA Virtual European Solar and Planetary Access

1 Introduction

This document defines the EPN-TAP framework. EPN-TAP is a protocol used to describe and access data related to the study of the Solar System in general, including observational, modeled, and experimental data. It consists in 1) the usual TAP mechanism; 2) the EPNcore metadata dictionary; 3) a set of rules defining table structure and parameter usage. EPN-TAP relies on the TAP standard with no modification (<http://www.ivoa.net/documents/TAP/20181024/>), so that only items 2 and 3 are described here. All EPN-TAP services are accessible through standard TAP clients.

Its implementation with TAP (Table Access Protocol) is presented, including service registration guidelines (TBC).

The first version of EPN-TAP was developed as an assessment study during the Europlanet-RI programme, and was limited to test purposes. It was redesigned to EPN-TAP 2.0 before the start of Europlanet 2020 and completed during the programme based on a large set of data services and use cases. This later version is described here.

1.1 The EPNcore metadata dictionary

The EPNcore metadata dictionary defines the core components that are necessary to perform data discovery in science fields related to the Solar System. The EPNCore metadata dictionary includes keywords to describe data products coverage (temporal, spectral, spatial, photometric), origin (instrument), content (target, physical parameters), access, references, etc. These keywords are intended either as search parameters or descriptive ones, although in TAP they can all be searched by value. In this sense, EPNcore bears similarities with the ObsCore protocol, from which it borrows several concepts.

An EPN-TAP service consists in a single table describing a list of "granules" or data products, usually a data file, a call to a web service, or a set of scalar values. Several such tables can reside on a single server, and grouping of data is at the discretion of the data provider. Only one data product can be described and linked per line of the table. Observations, simulations, or experimental data are supported.

The global philosophy of EPN-TAP is to describe all tables with a common set of mandatory parameters which can be used to query all EPN-TAP services together, which is the scope of the VESPA portal. A unique query can then be sent to / answered by all data services without generating errors. This also requires that all parameter values are provided in a standard form, in particular with standard units. In other words, mandatory parameters have to be present in an EPN-TAP service and must provide values in standard form. Although most of them can be left empty when non-relevant or unknown, care should be taken to fill as many parameters as possible to provide a better service to the user.

In addition to these mandatory parameters, the EPN-TAP dictionary includes optional parameters which are grouped in two levels: common, general purpose parameters provide complementary information; parameters from thematic extensions have been defined from a set of

related data services and must be used consistently in data services in these fields. The first group includes file access information and additional description of data and target. The second group defines additional axes and provides extra parameters to describe similar observations consistently. Any optional parameter can be used whenever required.

Finally, EPN-TAP allows data providers to use entirely new parameters in a data service to provide specific information, when no existing parameter applies. This may occur in particular when the data consist in a few scalar values included in the table itself as individual parameters. Such parameters are best defined in consistency with existing ones.

Section 2 provides a description of parameters and their usage. All the currently defined parameters are listed in Annex with their type, unit, and UCD.

EPNcore usage is not limited to data distribution and access through TAP. It also provides a standard way to describe Planetary Science data, e.g. to handle local databases and private projects while benefiting from interfaces with VO tools, in particular with TOPCAT, Aladin, and CASSIS which were closely associated to this development.

1.2 EPN-TAP rules applying to table structure and parameter usage

- **Rules related to data services and table structure include:**

- A service consists in one database schema containing only one table. The table must be called `<schema>.epn_core`.
- Only one data product is linked per table line, and line parameters describe that product. The same data provided under two different file formats must be described on two table lines.
- Related files providing documentation of the data product can however be associated through `thumbnail_url` (quick-look image) and `datalink_url` (various documents and links).
- Data services are declared in the IVOA registry of services according to TAP guidelines.
- Data service output must pass validation; an EPN-TAP service validator is available here: <http://voparis-validator.obspm.fr>.

- **Rules related to parameters include:**

- Parameters defining a range along an axis most often appear as a pair with `*_min` and `*_max` variations. Both values must be provided to support ranges correctly. If only one value is available, it must appear in both parameters.
- Parameters accept values in lower case, except when a global standard applies. This is the case in particular with Solar System object names, which follow the IAU naming convention.
- Empty parameters can be void or can contain the NULL value.
- All floating-point parameters are in double precision.

- Some parameters can be multivalued. Lists of values must be provided as hash-lists (values separated by internal # character).
- Although each line contains at most one linked data product, special conventions exist: Thumbnails may be provided on the same granule line, as far as they are intended for quick-look in the VESPA portal or similar user interfaces; inclusion of thumbnails is recommended. In contrast, large previews must be handled as separated data products. Associated documents, or in some cases alternative formats, can be attached via the `datalink_url` parameter. Web services or SODA data services are also linked via this parameter.
- Some parameters only accept values from a predefined list. Such lists are provided or discussed in section 2.
- String parameters support internal spaces (not leading or trailing spaces).
- Special characters and quotes are not allowed in string parameters, and may be changed to `_` (which is the single-character wildcard in TAP queries).
- Although additional, service-specific parameters can be used, care must be taken to have no duplication. In general, parameters must appear only once in a service table.
- Such free parameters, when very specific to a service, may use a prefix related to this service to prevent conflicts (e.g., `myservice_myparameter`). Associated UCDs must be extracted from the current version of the standard (UCD1+). Associated errors should use the syntax: `myparameter_error`, or `parameter_error_min` and `parameter_error_max`. The associated UCDs must start with `"stat.error;"` or `"stat.error;stat.min;"` & `"stat.error;stat.max;"`

1.3 Architecture of EPN-TAP services

Figure 1 shows the VO standards used in VESPA (Arviset and Gaudet et al., 2010). In addition to the standards listed here, EPN-TAP also potentially makes use of: DALI, VOresource, VOSI, STC, UWS, VOtable.

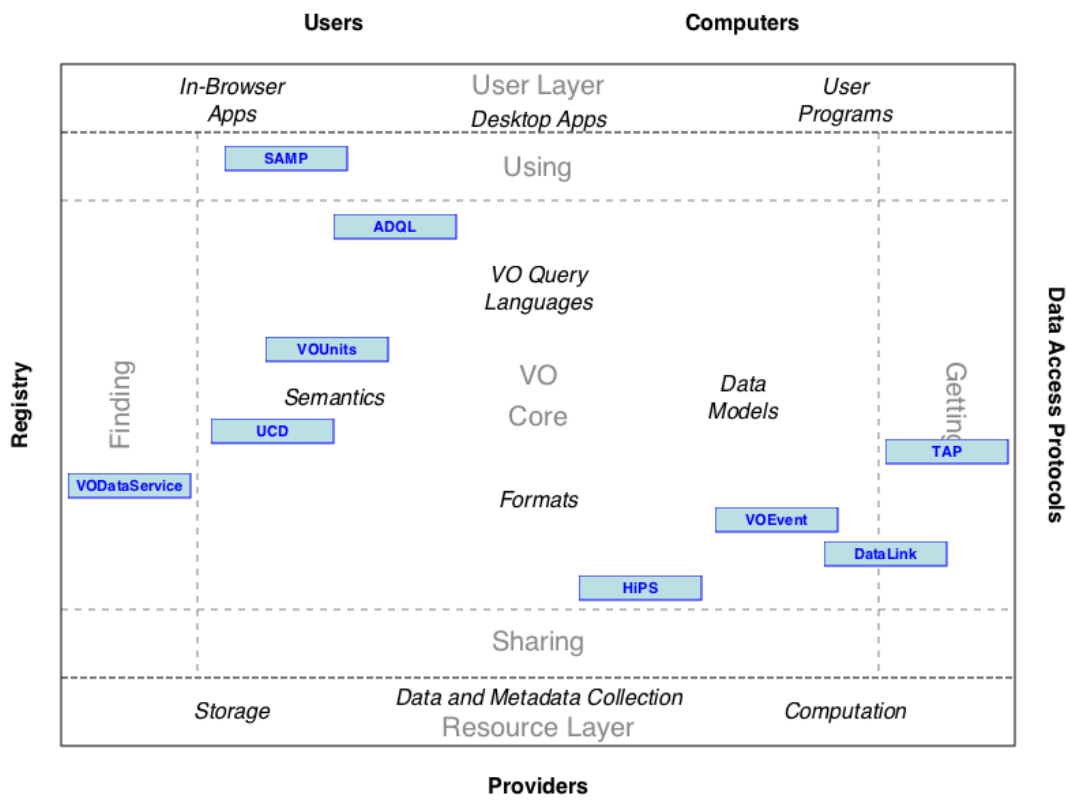


Figure 1: Architecture diagram for this document

2 EPN-TAP v2 parameter description

This section provides a complete description of EPN-TAP parameters, with meaning and usage.

Definition, types, UCDS, and a short definition are provided in Annex

Mandatory parameters

1- Granule references

These 3 keywords must always be informed. If granule grouping is not relevant, use a single value (and obs_id can be identical to granule_uid).

granule_uid

Unique granule ID in the data service.

There can be only one file associated to a granule (plus possibly a thumbnail for quick-look purpose in a search interface).

obs_id

Associates granules derived from the same data (e.g. various representations / processing levels). May be the ID of original observation.

granule_gid

Associates granules of same type (e.g. same map projection, or geometry data products) - think of it as a simple and convenient way to group or differentiate types of data.

When several files relate to the same data, this parameter helps distinguishing them: this will allow the user to select the type of data of interest. E. g., a service may provide links to calibrated images, plus raw data and ancillary information for every granule; these will share the same obs_id, but will have different granule_gid.

2- Data Description

dataprodect_type

The dataprodect_type parameter describes the high level scientific organization of the data product linked by the access_url parameter (see below), or directly included in the table (in

which case the value is 'ci' for catalogue_item). EPNCORE currently defines several types listed below. The data provider must select the type most adapted to his data. In complex situations (e.g., when a file contains several data products), several types can be used to describe the same granule by using a hash-list — although using several granules to describe the file content may be a better solution.

In EPN-TAP these types are identified by a 2-characters ID, so that multivalued queries are unambiguous. Possible IDs are listed below with their meaning:

- **im** (= image): scalar field with two spatial axes, or association of several such fields, e.g., images with multiple color planes, from multichannel or filter cameras. Preview images (e.g. map with axis and caption) also belong here. Conversely, vectorial 2D fields are described as catalogue (see below).
- **ma** (= map): scalar field / rasters with two spatial axes covering a large area and projected either on the sky or on a planetary body, associated to a map_projection parameter (with a short enumerated list of possible values); each pixel is associated to 2D coordinates. This is mostly intended to identify complete coverages that can be used as reference basemaps.
- **sp** (= spectrum): measurements organized primarily along a spectral axis, e.g., radiance spectra. This includes spectral aggregates (series of related spectral segments with non-connected spectral ranges, e.g., from several channels of the same instrument, various orders from an échelle spectrometer, composite spectra, etc).
- **ds** (= dynamic_spectrum): consecutive spectral measurements through time, organized primarily as a time series. This typically implies successive spectra of the same target / field of view.
- **sc** (= spectral_cube): sets of spectral measurements with 1 or 2D spatial coverage, e.g., imaging spectroscopy. The choice between image and spectral_cube is dictated by the characteristics of the instrument (which dimension is most resolved & which dimensions are acquired simultaneously). The choice between dynamic_spectrum and spectral_cube is related to the uniformity of the field of view and by practices in the science field.
- **pr** (= profile): scalar or vectorial measurements along 1 spatial dimension, e.g., atmospheric profiles, atmospheric paths, sub-surface profiles, traverses...
- **vo** (= volume): measurements with 3 spatial dimensions, e.g., internal or atmospheric structures, including shells/shape models (3D surfaces).
- **mo** (= movie): sets of chronological 2D spatial measurements (consecutive images)
- **cu** (= cube): multidimensional data with 3 or more axes, e.g., all that is not described by other 3D data types such as spectral cube or volume. This is intended to accommodate unusual data with multiple dimensions.
- **ts** (= time_series): measurements organized primarily as a function of time (with exception of dynamical spectra and movies, i.e. usually a scalar quantity). Typical examples of time series include space-borne dust detector measurements, daily or seasonal curves measured at a given location (e.g. a lander), and light curves.
- **ca** (= catalogue): applies to a single granule providing a catalog of object parameters, a list of features, a table in another TAP service, a list of events... "Spatial vectors" (e.g., vector information from a GIS, spatial footprints...) belong here. This is relevant, e.g., for collections of vectorial elements (e.g. crater contours or ROI definitions) which can

be handled directly in a specialized environment such as a GIS. This includes maps of vectors, e.g., wind maps.

- **ci** (= catalogue_item): applies when the service itself provides a catalogue, with entries described as individual granules. The service can be, e. g., a list of asteroid properties or spectral lines. Catalogue_item can be limited to scalar quantities (including strings), and possibly to a single element. This organization allows the user to search inside the catalogue from the TAP query interface.
- **ev** (= event): introduces individual VOevents formatted according to IVOA standard (or possibly events with other formatting, TBC)

Example TAP queries:

```
select * from epn_core
where dataproduct_type like '%im%'
```

or

```
select * from epn_core
where ivo_hashlist_has(lower(dataproduct_type), 'im') =1
```

will return only image data (the second syntax handles lists of values in a robust way).

measurement_type

The measurement_type parameter defines the physical quantities contained in the data, using UCDs. It relates to the reported quantity, not to the type of experiment. Therefore only UCD related to physical quantities can be used; e.g., phys.absorption;em.opt.I is eligible, while stellar_occultation is not.

The "UCD1+" list from IVOA must be used as a reference. New UCDs relevant for Solar System studies are regularly discussed, therefore recent extensions of this list must be also considered (i.e., at time of writing: <http://www.ivoa.net/documents/UCD1+/20180527/index.html>). Quantities derived from modeling/simulation are described by the regular UCD with ";meta.modelled" appended.

Whenever several quantities are comprised in the data, the measurement_type parameter must describe all these quantities, using multiple UCDs in a hash-list.

Examples:

- For images in general (i.e., actual measurements with a camera), the relevant UCD is obs.image (or obs.image;stat.uncalib if not calibrated).
- For spectra: phot.flux.density describes a *flux* vector (irradiance), while radiance and reflectance are currently being defined. The associated spectral vector is described by

UCDs em.wl, em.freq or em.energy, and the related error is described by stat.error;phot.flux.density (for flux).

processing_level

The processing_level parameter is intended to provide the user with a quick evaluation of data "usability". When the original dataset uses a specific encoding of processing levels, this one should be used to meet the expectations of historical users; e. g., a service deriving from a telescopic archive will preferably use the processing levels from this archive. When no practice exists for a dataset, EPN-TAP uses a simplified version of CODMAC / PDS3 levels as described below.

Several classifications are in use in different contexts, as summarized in the table below. EPNCore uses the CODMAC / PDS3 levels but removes the intermediate calibration levels; this is equivalent to using the simplified PDS4 levels and maintaining a separated level for ancillary data. "Partially calibrated" datasets are in general considered as not calibrated, but this evaluation is up to the data provider, depending on context. "Ancillary" data include all extra information documenting the measurements, in particular coordinates or geometry files. Several processing levels can be included in the same service (in particular calibrated and ancillary data, but also raw data). When mixed in the same file, several values may be provided as a hash-list.

Most EPN_TAP data services are expected to include Calibrated or Derived data.

EPN-TAP	CODMAC	PSA	NASA	PDS3	PDS4	ObsTAP	Description
1	1 (raw)	(0?)		UDR	Telemetry	Level 0	Unprocessed Data Record (low-level encoding, e.g. telemetry from a spacecraft instrument. Normally available only to the original team)
2	2 (edited)	1	0	EDR	Raw	Level 1 (std data format)	Experiment Data Record (often referred to as "raw data": decommutated, but still affected by instrumental effects)
3	3 (calibrated)	2	1A	RDR	Calibrated	Level 2 (calibrated)	Reduced Data Record ("calibrated" in physical units)

5	4 (resampled)		1B	REFDR	Derived	~ Level 3 (enhanced)	Reformatted Data Record (mosaics or composite of several observing sessions, involving some level of data fusion)
5	5 (derived)	3	2-5	DDR	Derived	~ Level 4 (analysis results)	Derived Data Record (result of data analysis, directly usable by other communities with no further processing)
6	6 (ancillary)			ANCDR	Derived		Ancillary Data Record (extra data specifically supporting a data set, such as coordinates, geometry...)

Notes:

- This table is a compilation of information from PSA, PDS4, & ObsCore documents
- The PDS3 column corresponds to the PDS3/PSA PRODUCT_TYPE keyword
- Descriptions are extracted from PSA documentation, with comments.

3- Target description

target_name

The target_name parameter identifies a target by name or ID. The target may be any Solar System body, exoplanet, planetary sample, or meteorite, plus in some cases astronomical objects or spacecraft. Any other feature (craters, regions, atmospheric layers...) must be named using the optional feature_name parameter (see 4.3.3). This parameter can be multivalued only to describe several targets related to a granule (e.g. with events). Alternative names of the same target must not be listed here, but may be provided through the optional alt_target_name parameter.

The best practice is to use the official designation of the target as defined by IAU. This parameter is case sensitive (mixing lower/upper cases) and all values must use the standard spelling and case; unusual characters (such as intermediate spaces) are allowed, except quotes (preferably changed to _ , which is the single-character wildcard in TAP queries). Data providers must be aware that services which do not use the IAU designations might not be accessible by the clients. Conversely, users must be aware that some services containing data of interest might not be visible, if they do not use the recommended IAU nomenclature for planetary bodies. The SSODnet name resolver from IMCCE may help data providers (as well as users) to handle multiple denominations; it is available from the VESPA portal to support queries.

Concerning celestial objects (at fixed position, i. e., stars, galaxies...) the name should be identifiable through Simbad (<http://simbad.u-strasbg.fr/simbad/>).

Other best practices are listed below:

- The Exoplanet Encyclopedia provides a nearly complete list of currently known extrasolar planets: <http://exoplanet.eu/>
- Meteorite catalogs can be found here: <http://www.nhm.ac.uk/research-curation/research/projects/metcat/search/indexsing.dsml> and <http://www.lpi.usra.edu/meteor/index.php>
- The catalog of lunar samples is available here: <http://www.lpi.usra.edu/lunar/samples/>
- Other planetary samples are listed in topical web sites, e.g. samples from the Stardust mission are described here: <http://curator.jsc.nasa.gov/stardust/catalog/>
- Asteroids: Usage is to use preferably name (if it exists) or principal designation (number is not used here, can be included in alt_target_name)
- Calibration targets: Values can relate to existing names in a given archive (e.g., the PSA contains values such as bias, checkout, dark, flatfield, internal source...)

Example TAP query:

```
select * from epn_core
where target_name like 'Ceres' or target_name like 'Vesta'
and target_type like 'dwarf_planet' or target_class like 'asteroid'
```

Will return data only from 1 Ceres or 4 Vesta. Complex queries may also include parentheses.

Example:

1P is the official IAU designation for comet Halley

target_class

The target_class parameter identifies the type of the target. Solar System bodies are defined without ambiguity by the couple target_class and target_name; in other cases, targets may have no proper name (i.e., samples). The possible values for target_class are:

- from IAU list [RD22]:
 - asteroid
 - dwarf_planet
 - planet
 - satellite
- extra values defined in EPNCore:
 - comet
 - exoplanet
 - interplanetary_medium
 - sample
 - sky

- spacecraft
- spacejunk
- star
- calibration

Usage:

- Any target has a unique target class.
- "interplanetary_medium" refers in particular to interplanetary dust.
- "sample" refers to lunar or planetary samples, to meteorites, but also to terrestrial samples, e.g., in laboratory studies.
- "satellite" stands for natural satellites only - artificial satellites are handled though spacecraft or spacejunk.
- "star" is used typically for calibration targets, and for the Sun.
- "sky" should be used for all other celestial bodies, usually referred to by their sky coordinates. It also includes the Interstellar Medium.
- "calibration" is used for observations only related to instrument or signal calibration, including dark current, flat field, reference sample (in lab), etc (use of "calibration" with planetary bodies is left to data providers).
- planetary rings are not considered as a target class, but appear in target_region, with their planet indicated in target_name.

Comment:

The SsODNet resolver uses "Types" of objects which are all included in EPN-TAP list: "Asteroid", "Spacecraft", "Comet", "Exoplanet", "Spacejunk", "Satellite", "Planet", "Dwarf Planet", "Star" Additional EPN-TAP classes are: interplanetary_medium, sample, sky, calibration

4- Axes

EPN-TAP describes the data location along 8 main axes: time, spectral, spatial (3 coordinates), and viewing geometry (3 angles).

Locations along the time/space/spectral axes are described with a range and a resolution or sampling step.

Each quantity is a set of 2 parameters providing min and max values (except spatial_frame_type and s_region). They must both be present with the same value when min = max.

time (min/max)

The time_min and time_max parameters provide the date and time of acquisition in the observer frame.

The time parameters are always provided in UTC and formatted in Julian Days (expressed in double precision). EPNCORE uses standard JD to avoid ambiguity with time origin (as opposed to ObsCore which uses Modified JD). The use of double precision floats insures an accuracy on the order of 1 ms, which is considered sufficient for search purposes (the initial accuracy is preserved in the data itself).

Examples TAP queries:

- Search data described by a time range:

```
select * from epn_core where time_min > '2455197.5' and time_max < '2455927.5'
```

- Search data described by a start time parameter

```
select * from epn_core where time_min between '2455197.5' and '2455927.5'
```

Time range is by default provided in the observer frame (i.e., at the observer location), which is almost always the native time in the data. For instance, space-borne observations are usually documented with spacecraft on-board time, which is expected here (provided as JD, not as on-board clock timing). For other cases, the location where time is measured must be provided though the `time_origin` parameter.

To support space-borne vs ground-based campaigns, multiple spacecraft observations, or survey of periodic events, measurement times need to be corrected for light path. Whenever comparisons are potentially involved, the use of the `target_time_min` and `target_time_max` parameters is recommended in addition to `time_min` and `time_max` to simplify this kind of comparison.

Non-compulsory parameters may be used to accommodate additive, specially formatted time scales such as native on-board time. The information of the `time_min` and `time_max` parameters is however greatly recommended for observations, as it is used by default to search datasets.

time_sampling_step (min/max)

These parameters provide the sampling step for measurements of dynamical phenomena, and for computations. This is the time between 2 successive measurements or data, which is mostly relevant when the measurements are regularly spaced. This may also be used as an input parameter, e.g., for ephemeris computations. This parameter is intended to allow the user to search for time-resolved observations of dynamic phenomena.

time_exp (min/max)

These parameters correspond to the integration time (or exposure time) of measurements. They provide an estimate of the time resolution for dynamical phenomena, as well as an indication of relative S/N ratio inside a given dataset. This time is usually shorter than the `time_sampling_step`

if both are present. It provides the overall integration time, i.e. individual exposure time x number of summed frames when relevant.

spectral_range (min/max)

The spectral_range parameters define the upper and lower bounds of the spectral domain of the data. This quantity is conventionally expressed on a frequency scale in Hertz. The spectral range and associated parameters only apply to electromagnetic waves. See the optional parameters particle_spectral_* for particle energy or mass detection.

Since this is the standard parameter to identify a spectral range, it is recommended to fill it even for images obtained through a filter ([central wvl ± FWHM/2, or even central wvl in both parameters, is enough for search purposes](#)).

Conversions

```
with c = 2.99792458E8 (m/s)
spectral_range_min = c*E6 /spectral_range_max_micron
spectral_range_max = c*E6 /spectral_range_min_micron
(noticethe inversion)
```

spectral_sampling_step (min/max)

The spectral_sampling_step parameters provide the spectral separation between the centers of two adjacent filters or channels. Like the spectral_range quantities, they are expressed on a frequency scale in Hz.

These parameters are most relevant for radio observations (long wavelengths / low frequency). They can also help distinguish between Nyquist and sub-Nyquist sampling rates (together with resolution).

Conversions

```
with c = 2.99792458E8 (m/s)
df = c*1E6 dlam / lam^2 (from wavelength, with lam and dlam in micron)
df = c*1E3 dlam / lam^2 (same if dlam provided in nm)
df = c/1E-2 du (from wavenumber u in cm-1)
```

spectral_resolution (min/max)

The spectral_resolution parameters actually provide the (dimensionless) resolving power: $|\lambda / \Delta(\lambda)| = |f_q / \Delta(f_q)|$

These parameters are mostly intended to provide an order of magnitude, e.g., to distinguish between Fourier spectrometers, grating spectrometers or filter cameras, or between observations related to surfaces or atmospheres. This is often most relevant for optical/infrared observations (short wavelengths / high frequency).

Conversion

```
spectral_resolution_min = abs( spectral_range_min_micron / max(DeltaLambda)
)
spectral_resolution_max = abs( spectral_range_max_micron / min(DeltaLambda)
)
```

c1 (min/max)

c2 (min/max)

c3 (min/max)

These parameters provide up to three spatial coordinates of the measured target. The coordinates depend on the spatial frame type defined below. All services must handle three spatial coordinates, even if the third one is always set to NULL. In body-fixed coordinates c1 is longitude, always counted eastward; c1min is then the westernmost longitude and c1max the easternmost one (so as accommodate 0 meridian crossing). Note that the c3 parameter is related to the observed area; the target distance (e. g., geocentric distance for ground based observations, or spacecraft distance) is introduced by optional parameters "earth_distance" or "target_distance".

In order to make uniform requests possible, spatial coordinates provided in the epn_core table must be standardized. However, they can be provided in several systems types, as defined by the spatial_frame_type parameter. [The native coordinate system used with the dataset can be described by parameters spatial_coordinate_description and spatial_origin.](#) This is intended to provide this information prior to loading the data, especially when several coordinate systems are available in the same service. Descriptions for EPN-TAP are provided in [RD17]. Secondary coordinates can be introduced using additional parameters, e.g., c1 and c2 ranges identify the visible region of a planetary disk, while extra RA / Dec parameters provide location on the sky at this moment, and subobserver_longitude_min/max and subobserver_latitude_min/max provide the coordinates of the disk center.

c1_resol (min/max)

c2_resol (min/max)

c3_resol (min/max)

These parameters provide a simple estimate of resolution, e.g. the FWHM of the PFS on the sky (in degrees), depending on spatial_frame_type. The client front-end may propose more appropriate units to the user, depending on context (e.g., angular resolution in mas, distance in m...).

spatial_frame_type

Provides the "flavor" of the coordinate system, which defines the nature of the spatial coordinates (c1,c2,c3) in the epn_core table and queries, and the way they are defined. This may

be different from the coordinate system associated to / included in the data themselves. A value is required (use "none" if not applicable, "body" may also be OK in most cases). The possible types are described below:

- **celestial:** 2D angles on the sky, i. e. , right ascension c1 and declination c2 + possibly heliocentric distance in c3 (in au) – although this is a special case of spherical frame, the order and conventions are different. Assumes ICRS coordinates; right ascension is provided in degrees. For ground-based observations, Earth distance may be provided in the earth_distance parameter (rather than c3).
- **body:** 2D angles in body-fixed frame: longitude c1 and latitude c2 + possibly altitude as c3.

A planetocentric system with eastward longitudes in the range $(0,360)^\circ$ is required for service interoperability. Current IAU frame is assumed, in particular for definition of prime meridian. IAU 2009 planetocentric convention applies, in particular eastward longitudes and a north pole located on the north side of the invariant plane of the Solar System for planets and satellites.

Parameter c3 is measured above the reference ellipsoid, can be <0 for interiors. Two other parameters are available to provide other vertical scales (radial_distance and altitude_fromshape).

Planetocentric rotating frames are defined as spherical (not body).

- **cartesian:** (x,y,z) as (c1,c2,c3). This includes spatial coordinates given in pixels.
- **spherical:** (r, theta, phi) as (c1,c2,c3), as defined in [ISO standard 80000-2:2009](#); r = radius; theta = zenith angle/colatitude/inclination; phi = azimuth (E longitude). Angles are provided in degrees. When related to the sky or tied to a solid body, "celestial" (with RA/Dec) or "body" (with longitude/latitude) frames must be used instead.
- **cylindrical:** (r, theta, z) as (c1,c2,c3); r = radius; theta = azimuth; z = height. The angle is provided in degrees.
- **none:** to be used when no spatial frame is defined for the dataset. This is intended to prevent useless searches when space coordinates are not defined (a non-zero value is required by the mixin in DaCHS).
- **healpix:** TBC (Nside and pixel#? Should also specify or assume ordering (nested/ring)) - [TBC, wait for IVOA decision.](#)

This parameter, although related to the specific coordinate system in use, is mostly intended to identify the nature of the coordinates handled by the service (e. g., angles versus distances). This parameter is provided as a column of the epn_core table, to ensure it can be queried through the basic TAP mechanism. Although it will in general remain constant along the table for simple services, this parameter can vary from granule to granule and a value must therefore be provided in any query that includes spatial coordinates. Whenever additional coordinates are provided, they must be stored in extra columns of the table. If several different frames are mixed to provide the main coordinates, the use of different granule_gid may help clarify the situation. At any rate, easy access to the data must be considered during the design of the service.

incidence_angle (min/max)

The incidence angle parameters define the upper and lower bounds of the incidence angle range in the data (also known as Solar Zenith Angle). This is always indicated in decimal degrees, and may range from 0 to 90° (with 0° indicating the normal to the surface). Incidence and emergence angles may be counted relative to the normal of the ellipsoid model, or to the local normal (e. g., using a 3D shape model). In case the two systems are included in the data, these keywords introduce the values relative to the ellipsoid (local values may be available through non-compulsory parameters).

emergence_angle (min/max)

The emergence angle parameters define the upper and lower bounds of the emergence angle range in the data (a.k.a. viewing angle). This is always indicated in decimal degrees, and may range from 0 to 90° (with 0° indicating the normal to the surface). Incidence and emergence angles may be counted relative to the normal of the ellipsoid model, or to the local normal (e. g., using a 3D shape model). In case the two systems are included in the data, these keywords introduce the values relative to the ellipsoid (local values may be available through non-compulsory parameters).

phase_angle (min/max)

The phase angle parameters define the upper and lower bounds of the phase angle range in the data (i.e., scattering angle - 180°, or light source-target-observer angle). It is always indicated in decimal degrees, and may range from -180 to 180° (with 0° corresponding to opposition, i. e., light source in the back of the observer). Negative values may refer, e. g., to geometry before opposition, depending on context. Phase (*phi*), incidence and emergence are partly related:

$$\text{abs}(i - e) < \text{phi} < i + e$$

If the azimuth angle *a* is available instead of the phase angle, the latter can be derived from knowledge of the three angles:

$$\cos \text{phi} = \cos i \cos e + \cos a \sin i \sin e$$

s_region

This parameter introduces a footprint for spatially extended data products in 2D, most notably on the sky (using RA, Dec) or on planetary surfaces (using E longitude, latitude). This is a single parameter with no min/max declinations. It must contain a PgSphere spoly variable with syntax: '{(lon1,lat1), (lon2,lat2), ... }' (with no quotes) where (lon1,lat1) = (10.d, 5d), character d included (it stands for degrees). Pairs (lon1,lat1) must sample the dataproduct contour in sequence, provided in direct (counter-clockwise) sense. The result of the query is an STC-S string (as in ObsCore; currently limited to polygons).

In addition, the resource descriptor (q.rd for DaCHS) must contain the line (for body coordinates):

```
<stc> Polygon UNKNOWNFrame [s_region] </stc>
```

Open issue: this parameter is responsive to the STC-S string standard of IVOA, which is at draft level and has changed recently at the time of writing - the indication of a coordinate system is no longer included in the string, and the exact support for E longitude (body) is uncertain.

5- Data origin

instrument_host_name

This parameter provides the name of the observatory or spacecraft that performed the measurements. A hash-list of host names must be provided for integrated data sets. In the epn_core table, the acronym is preferred to the full name to avoid long strings and related errors (both values can be provided in the list). An observatory list is being developed in Europlanet2020 (VESPA / NA1 activity) and will be implemented in a resolver by merging existing lists: [Observatory Facility Database](#)

Reference lists include:

- for ground-based observations, the list of IAU observatory codes: <http://www.minorplanetcenter.net/iau/lists/ObsCodesF.html> However, this list is not intended to include all ground-based observatories, and a complement still needs to be identified (including e. g. radio-telescopes).
- a reasonably complete list of radio-telescopes is available here: http://en.wikipedia.org/wiki/List_of_radio_telescopes.
- the WiseRep list also provides a database of telescopes and instruments: <https://wiserep.weizmann.ac.il/aux/telescopes>, <https://wiserep.weizmann.ac.il/aux/instruments>.
- Concerning space-borne data, the most complete list of international planetary missions and orbital observatories is found here (included in a complete list of space missions with ID): <http://nssdc.gsfc.nasa.gov/nmc/>. Planetary missions are also listed here: <http://nssdc.gsfc.nasa.gov/planetary/chronology.html>.
- Alternatively, the PDS dictionary defines values for many mission names: <http://pds.nasa.gov/tools/dictionary.shtml>. Other mission names are supported by the SPICE system, but only as ID codes: http://www-int.stsci.edu/~sontag/spicedocs/req/naif_ids.html

instrument_name

Identifies the instrument(s) that acquired the data. A hash-list of instruments must be provided for integrated datasets. Service providers are invited to include multiple values for instrument name, e.g., complete name & usual acronym. This will allow queries on either "VISIBLE AND INFRARED THERMAL IMAGING SPECTROMETER" or "VIRTIS" to produce the same reply.

Example:

VISIBLE AND INFRARED THERMAL IMAGING SPECTROMETER#VIRTIS

Concerning space-borne data, a list of recommended values (to be implemented in a resolver) is discussed here: [Observatory Facility Database](#)

The most complete ready-to-use list of international planetary missions and orbital observatories is found here: <http://nssdc.gsfc.nasa.gov/nmc/>

Instruments on board planetary missions in particular are listed here: <http://nssdc.gsfc.nasa.gov/nmc/experimentSearch.do>

6- Granule call-back info

These parameters provide ancillary information for routine processing after a query. These 4 keywords must always be informed. Dates are provided in ISO 8601 format reusing this pattern: “YYYY-MM-DDThh:mm:ss”.

service_title

Provides an acronym for the service/table title. This is used to refer to the service providing the data, therefore this should be identical to the name of the schema and must be constant in the view (constraint TBC).

When designing the service, care must be taken to use a schema name not already ascribed to another EPN-TAP service - it *must* be unique on the provider side in any case.

creation_date

Provides the date when the granule was introduced

modification_date

Provides the date when the granule was last updated (intended to speed up mirroring between sites)

release_date

Provides the date when the granule becomes public (intended to support a proprietary period, usage TBD)

Optional parameters

EPN-TAP can query any parameters included in the metadata table. Some of these parameters are defined precisely but are relevant only to very specific data services. Those are not mandatory, but they must be implemented as defined in this section when present. Besides, the *names of optional parameters are reserved for this particular usage and must not be used to introduce other quantities.*

7- Data Access Reference

If the data are not included in the table (i.e., provided in separated files), these 3 parameters must be present and informed. If alternative formats are also provided, they must be described in other granules with the same obs_id and possibly different group ID. Whenever the data consists in a few scalar fields, these parameters can be replaced by parameters providing the data itself in the table, which makes them searchable (e.g. mass, in a table providing the masses of Solar System bodies).

access_url

The data of interest are often stored in a file, not in the table itself. In this very usual case, the access_url parameter provides a complete path to the data products on the network, so that they are accessible for download by plotting or processing tools. All URLs in the epn_core table are case sensitive and must provide an actual and unique link. The link may be a call to a web service (e.g., CRISM service) or the output of a script (e. g., Titan atmospheric profiles service); in both cases the link must include the adequate arguments. In any case, this parameter must link to the actual data, not to a file of metadata nor a document. Datalinks (which typically open a page with either a list of links or an input interface for parameters) can be used for this later purpose, and must be provided via the datalink_url parameter.

access_format

Access_format provides the format of the data file linked through the access_url parameter. The data may be stored under native format, and no format conversion is required to set up an EPN-TAP service; this field can therefore include reference to unusual formats. However, VO ready formats are required to take advantage of visualization and processing in VO tools. Consistently with ObsCore, possible values are MIME-types written in lower cases; the most common ones are listed on this page: [Data Formats and MIME Types](#).

access_estsize

The access_estsize field provides an order of magnitude (in kilobytes) of the file available via the corresponding URL. It is intended to provide an indication that can help to tune download functionalities in an application, depending on data volume and transfer bit rate.

Other parameters may be used to describe the data files:

thumbnail_url

The thumbnail_url parameter contains the URL of a reduced version of the data product used for quick-look purpose (e.g., a small jpeg image, typically 200x200 pixels). This may be handy in the case of big data files or unusual data formats, to facilitate data selection by the user. The

EPN-TAP client uses this thumbnail for on-line quick-look, which therefore provides important added value to a service. Preferred formats include jpeg and png, which are handled easily by a basic viewer. Larger or more elaborate previews must be provided as independent granules and identified via a `granule_gid` different from that of the data. All URLs in the `epn_core` table are case sensitive and must provide an actual link.

file_name

The `file_name` parameter introduces the name of the *data* file, with extension but no path information. In many data services, the file name encodes the most relevant metadata and may be a very handy access mechanism for specialists. In services providing sets of files in a complicated directory tree (e.g., related to a spacecraft operation plan), the `file_name` parameter is a handy key to perform automatic operations such as mirroring, pipeline processing, etc - a web service is available to retrieve a file from the `file_name` parameter in any EPN-TAP service ([File grabbing interface](#)). Its use is therefore always recommended when data are provided in separated files, i. e., not included in the table.

All `file_name` values in the `epn_core` table are case sensitive and must reflect the filename of the product provided through `access_url`; for instance, if `access_url` is a link to a script converting an ascii file to VOTable, `file_name` must refer to the outcome of this script, in this case the VOTable.

access_md5

This parameter introduces a MD5 Hash for the file when available (link to a real file), to be used as a checksum.

accref

Although not an EPN-TAP parameter, this name is reserved for internal use. It is apparently used to introduce a URL with `datalink` in TAP / DaCHS (present in `epncore2` mixin), and is also generated automatically in the `q.rd` when using the `localfile` mixin (when accessing data files located on the DaCHS server itself). It may also be required to handle proprietary periods in DaCHS, TBC.

datalink_url

This parameter is used to provide extra accesses through a `datalink/SODA` interface. It can either provide a table of hard links (`dlmeta`) or a dialogue to setup input values on the fly and call a web service on the data server (`dlget`). Links can be parameterized with input values retrieved from the current granule at the time of ingestion; if several EPN-TAP parameters need to be passed, they must be first concatenated in an extra parameter in the table (often called `ds_id`).

8- Supplementary descriptions

bib_reference

The `bib_reference` parameter introduces an individual bibliographic reference at granule level. This provides the origin of the data, e. g., if the resource is a compilation of data from various origins. References are best provided as bibcode (as used e. g. by ADS), as DOI, or as arXiv reference, although other formats are acceptable. A generic regular expression for bibcode (from IPDA discussions) is: `[0-9]{4}[A-Za-z0-9&\.]{5}[A-Za-z0-9]{9}[A-Z\.]`

See <http://adsabs.github.io/help/actions/bibcode>

publisher

Specifies the publisher of the *data service* (not necessarily at the origin of the data). Currently a free format string.

internal_reference

The `internal_reference` parameter can be used to identify granules (or sets of granules) intimately related to the current one. E.g., in a service containing both observations and results of analysis of observation sets, `internal_reference` can be used to provide the set of observations used to compute a result. This contains a list of `granule_uid` in the same service.

This is specifically intended to provide internal references in services which would otherwise need to be split in several tables, and must it be used only as a last resort (clever use of `obs_id` and `granule_gid` is usually more satisfying).

external_link

The `external_link` parameter can be used to provide extra information that does not fit easily in the table, and is intended for human reading only. This parameter must contain a single URL. This is typically an extended discussion of a granule on a web site, which may include images, tables, or other documents. For instance, the individual planet pages of the Encyclopedia of exoplanets are linked with this parameter.

species

The `species` parameter introduces the chemical species of interest in simple data services. The formatting is very basic and simply uses the standard formula in ascii, e. g., H₂O for water, CO₂ for carbon dioxide or Fe for iron. This is one of the few query parameter provided in case sensitive form, using the standard chemical notation. This format can only accommodate atoms and simple molecular species, and does not support isotopic variations.

An example application is related to atmospheric composition: a table providing vertical profiles of gaseous species with altitude. All columns are abundances and are described by the same `measurement_type` parameter. The use of the "species" parameter allows identifying the various species and accessing the requested information. This of course is restrained to simple cases.

If more elaborated compositional information must be included, the use of another parameter providing InChiKeys is recommended - TBC (inChiKey only related to molecule, including

isotopes, but not physical state / phase; does not include #, therefore consistent with the hash-list syntax).

waveband

provides a rough indication of the spectral domain, e.g. when variable in a large archive. Values are from VODataService / ConeSearch (beware of slight differences in SSA: ultraviolet instead of UV, and no EUV):

- Radio
- Millimeter
- Infrared
- Optical
- UV
- EUV
- X-ray
- Gamma-ray

filter

This parameter introduces the standard name of a filter used during measurements. This is reserved for filters in imaging mode (no grating/grism #, etc). There is no predefined list, because of the large variety of possible denominations, but the best practice is to use a short and accurate ID. This VO service provides an extended list of references: <http://svo2.cab.inta-csic.es/svo/theory/fps3/>

This is intended to document the results of a search, rather than a search parameter. Therefore, it is recommended to also fill the spectral_range_min/max parameters to describe filter imaging - this is the only way to make filter imaging easily searchable.

alt_target_name

This parameter introduces alternative names of the target, especially when they are more usual than the official IAU one (e.g., Halley vs 1P).

A frequent usage is to store a hash-list of all alternative names for small bodies to avoid ambiguities (e.g., for asteroids: name, number, principal and provisional designations).

feature_name

introduces a supplementary name to provide more details about the observed target. It is intended in particular to accommodate a local name (crater, surface feature, region...) whereas target_name is reserved to identify the whole body (Mars, Moon, Ceres...). Use of official features names defined by IAU (<http://planetarynames.wr.usgs.gov/>) is preferred when relevant.

target_region

specifies a type of region on the target. This parameter only introduces generic regions (e.g., atmospheric layer, internal structure...), not specific local names which must be handled using the `feature_name` parameter.

Values are taken from standard sources:

- IVOA thesaurus: <http://www.ivoa.net/rdf/Vocabularies/vocabularies-20091007/IVOAT/>
- IAU thesaurus <http://www.mso.anu.edu.au/library/thesaurus/>
+ another version: <http://www.vocabularyserver.com/trex/en/>
The latter seems more recent and more complete (although the interface is not practical)
- Spase dictionary <http://www.spase-group.org/>

Example:

"atmosphere", "surface", "ionosphere"

The same sources are used for the declaration file in the registry.

Description of coordinate frame

Spatial_coordinate_description

Spatial_origin

These two parameters provide a description of the spatial frame in use, depending on the `spatial_frame_type` parameter.

`Spatial_coordinate_description` provides an acronym of the Coordinate Reference System as discussed in [Planetary Coordinate Systems](#). For body-fixed frames, IAU/SPICE/WMS strings such as IAU20xx:49900 are eligible - in this case, the final 00 which stands for planetocentric E-handed is the only valid option (refers to EPN-TAP standard on C1/C2 coordinates) - **TBC**. If absent the current IAU coordinate system is assumed, in particular for the definition of the prime meridian.

`Spatial_origin` may be used to identify frame center in specific situations, **TBC**

Examples (TBC)

```
ICRS, BODY?, Mars_IAU2000? IAU2000:49900  
Geocenter
```

time_origin

indicate where the time is measured. This knowledge is required to cross-correlate event-based observations, in particular to indicate light-path differences. It applies to the `time_min` and `time_max` parameters (while `target_time` always refers to the target in the FoV). If this parameter is not informed, time is expected to be provided in the observer frame. Example values for `time_origin` are:

Earth, (solar system bodies), (spacecraft)

time_scale

Always UTC in *data* services. This may be relaxed in computational services such as ephemeris, using standard acronyms for time scales.

Target configuration & observing geometry

The next pairs of parameters provide additional location in time and space, and require min/max values. They are independent and can appear separately:

solar_longitude (min/max)

Solar longitude (a.k.a. heliocentric longitude, or ecliptic longitude of the Sun, traditionally noted Ls) is the Sun-Planet vector angle counted from the planet position at N hemisphere spring equinox. It provides a measurement of season.

Ls = 90° corresponds to the northern summer solstice, Ls = 180° to the northern autumn equinox, and Ls = 90° to the northern winter solstice. Although it is most usually applied to Mars and Titan (using Saturn's Ls), this notion can be enlarged to any planetary body without ambiguity.

This should not be confused with the true anomaly of the body (which is the same angle counted from the perihelion position), nor with the longitude of the subsolar point (see below).

local_time (min/max)

provide the local time at the observed area. These parameters are provided in unit of target rotation divided by 24 and are measured from local midnight (ranges from 0 to 24, must increase with time at a given location). They are provided in decimal hours.

target_distance (min/max)

The target_distance parameters introduce the distance of the observer to the observed area (in km) along the line of sight. This is mostly intended for space borne data, where it provides the spacecraft-target distance in km. For ground-based observations the earth_distance_min/max parameters should be used instead (in au).

target_time (min/max)

The target_time parameters introduce the time measured in UTC scale at the target. This is intended to directly correlate simultaneous observations such as ground-based support and space-borne observations, or multi-spacecraft campaigns. Values in ISO 8601 format reusing this pattern: “YYYY-MM-DDThh:mm:ss”.

earth_distance (min/max)

sun_distance (min/max)

these two parameters provide the corresponding distance of Earth or Sun to the target at the time of observation (in au). When the target is the Sun or Earth, use the target_distance_min/max parameters to provide the distance to the observer.

subobserver_longitude (min/max)

subobserver_latitude (min/max)

Provide coordinates of sub-observer point, in particular sub-Earth point (disk center) for ground-based observations (this is different from the FoV location which is provided in C1/C2 parameters). The min/max pair is required to support long exposures related to time series, especially on giant planets to test target attitude.

subsolar_longitude (min/max)

subsolar_latitude (min/max)

Provide coordinates of sub-solar point, e.g., especially for disk images. The min/max pair is required to support long exposures related to time series.

ra & dec

sky coordinates of the target can be provided in addition to standard coordinates, in which case they must be stored in these parameters (no min/max variations to maintain compatibility with existing software). This may for instance document the location of a planet in a celestial image, while the main coordinates c1/c2 are used to describe the observed area as longitude and latitude. ICRS coordinate system is assumed; right ascension is provided in degrees (similar to ObsCore). RA, DEC parameters are interpreted correctly by most VO tools.

Vertical scales on planets

In body-fixed frame, parameters C3min/max, if provided, introduce a vertical range counted from a reference surface, typically the reference ellipsoid for the current target.

Two other parameters can be used provide different vertical scales, and must be considered for atmosphere-related services when available/relevant:

radial_distance (min/max)

Distance of observed area (at C1/C2) from body center, measured in km. Not to be confused with target_distance (which provides the distance from observer to observed area).

altitude_fromshape (min/max)

Altitude of observed area (at C1/C2) above local surface, measured in km. The local surface is provided by a DTM or a shape model. This parameter typically provides the height in the atmosphere.

C3 can be used to select services/data in a given altitude range above the ellipsoid, while radial_distance and altitude_fromshape provide other convenient vertical scales to compare observations from various origins. This use of C3min/max also applies to planetary interiors (C3 is then <0) and measurements at high altitude/distance.

Extensions

EPN-TAP extensions are subsets of optional parameters related to a specific field of data, based on several implemented services.

When the corresponding quantities are present, they must be introduced by these parameters to favor interoperability of services.

All parameters of an extension may be present when the extension is in use. All parameters are available in other situations, provided that the meaning and usage are preserved.

9- Particle spectroscopy extension

These parameters are related to the spectral distribution of particles only - for electro-magnetic waves, the spectral_* parameters apply.

When used, these parameters define an extra axis and must all be present. This set defines an additional axis for particles energy, all with min/max values.

particle_spectral_type

The particle_spectral_type parameter introduces the type of axis in use: either energy (provided in eV), mass (in amu), or mass/charge ratio (in amu/qe).

particle_spectral_range (min/max)

The particle_spectral_range parameters define the upper and lower bounds of the spectral domain for particles. Depending on the particle_spectral_type parameter, this quantity is expressed on an energy, mass, or mass/charge scale, with respective units eV, amu, or amu/qe.

particle_spectral_sampling_step (min/max)

The particle_spectral_sampling_step parameters provide the spectral separation between measurements, in the same scale and unit as particle_spectral_range. This parameter is mostly intended to provide an order of magnitude.

particle_spectral_resolution (min/max)

The particle_spectral_resolution parameters correspond to the actual resolution of the measurements, and are provided in the same scale and unit as particle_spectral_range. This parameter is mostly intended to provide an order of magnitude.

10- Solar System Objects extension

Service providing descriptions of solar system objects contain no observations (only inferred properties).

In this case, the target_distance_min/max parameters can provide distances from the frame origin (typically heliocentric), in km for consistency - [TBC, not favorite](#)

mean_radius, equatorial_radius, polar_radius

These parameters are used to provide solar system object sizes (in km)

mass

Solar system object mass (in kg)

sidereal_rotation_period

Solar system sidereal rotation period (in hour)

semi_major_axis

in au,

inclination, etc

See discussion page: [orbital parameters](#)

When target_class = **asteroid or dwarf_planet**, the following parameters can be used:

dynamical_class

introduces the class of small body, from an enumerated list. The draft list includes: TNO, MBA, NEO - add OCC (Oort Cloud comet), JFC (Jupiter family comet) here, and Centaur, Trojan (but which planet?)

dynamical_type

introduces a subdivision of the above, from an enumerated list. It currently includes:

- NEO (complete): Atira, Aten, Apollo, Amor
- TNO (complete, but check values): res 2:5 (develop?), res 1:2 (develop?), Plutino, Hot classical, Cold classical, Scattered disk object, Detached object, Inner Oort object (not strictly KBO?)

MPC service also contains the following (unclear) values as orbit_type: 4 NEO types + Object with perihelion distance < 1.665 AU, Hungaria, MBA, Phocaea, Hilda, Jupiter Trojan, Distant Object, Unclassified

11- Maps

map_projection

Provides map projection description and parameters as a free string. For instance, several services use this to store proj4 parameters.

map_height & map_width

Provide dimensions of raster maps / WCS.

pixelscale (min/max)

Pixel size at a surface, in km/pixel.

12- Contributive works

Services including data provided multiple sources can identify them with parameters of type **observer_***, e.g., observer_id, observer_location, etc

This is used in particular for services compiling amateur data, of facility-related data services. [List TBD](#), [see also PVOL](#).

original_publisher

Services compiling data from many sources can use this parameter to refer to the source of the data

data_calibration_desc

The **data_calibration_desc** parameter from the spectroscopy extension can be used to provide information on post-processing (this is preferred over a "comment" parameter)

producer_name & producer_institute

provide reference to who produced the data, especially for experimental data services

13- Experimental spectroscopy

Specific *_desc parameters describe the sample and the experimental setup. Although optional, they are important to provide the context. They introduce free descriptive strings which are not intended as search parameters.

Note: VOTable are often better supported than fits files by VO spectral tools, and are therefore the preferred option to distribute spectral files if format conversion has to be considered. For dimensionless data (e.g., reflectance), units="" (VOunit standard)

producer_name & producer_institute

provide reference to who measured the sample (from Contributive Works extension).

sample_id

Additional ID of the sample, e. g., a specific fraction of a meteorite (in addition to target_name). Intended to refer to a pre-existing catalogue of a collection, will therefore contain a name/ID mainly for local use.

sample_classification

provides composition as group, class, sub-class, etc... of sample concatenated in a hash-list (this allows for flexible searches with LIKE and %; this is in practice the only way to ensure that we get all results). It should include the specification "meteorite" plus the meteorite type when applicable, as well as description of (main) mixtures ingredients. Meteorite types as in Krot et al 2005. Dana or Strunz classification tags can be used for minerals. Minor/trace components are not welcome here (would multiply false alarms).

grain_size (min/max)

provide the particle size range in μm . A very large value (eg, $>1000 \mu\text{m}$) can be used locally in a service to identify bulk material - [we need to define a code for this \(-1 could do, but we also have to reserve a code for N/A\)](#). This is really *grain_size*, since *particle* is reserved for particle spectroscopy.

azimuth (min/max)

azimuth angle in degrees - [check if negative values of angles can have a special meaning \(also for phase angle\)](#)

pressure & temperature

Provide experimental conditions, **in bar** and K - although not recommended by IAU, unit bar is accepted for pressure in this context, as it refers to terrestrial/lab conditions.

sample_desc

free string or hash-list describing the sample, its origin, and possible preparation

setup_desc

free string or hash-list describing the experimental setup if needed - may include aperture (size of sample measured), etc

data_calibration_desc

free string or hash-list describing data post-processing / calibration

geometry_type

Describes the geometry for spectral measurements, [from an enumerated list](#). Proposed values are:

- direct
- directional
- bidirectional
- specular
- biconical
- hemispherical
- hemispherical-directional
- directional-hemispherical
- hemispheric-hemispherical
- other geometry
- unknown

spectrum_type

Explicitly provides the type of spectral measurement, from an enumerated list - this is where 'radiance factor', 'reflectance', 'reflectance factor', etc... are defined. The purpose is to give a precise description of complex measurements independently from UCIDs, which are not expected to reach this level of description.

See draft list on this page: [Lab spectroscopy extension](#)

measurement_atmosphere

Provides description of experimental conditions as a free string. Measurements under vacuum are indicated here with the word 'vacuum'.

Use of general parameters

More general parameters have restricted usage in this case:

thumbnail_url Provides a link to a small spectral plot - caution should be taken to maintained minimal readability in small format in the VESPA portal. Larger plots can be included as separated granules (previews).

datalink_url External documents are often available to provide extra information in this context, e.g. chemical analysis or descriptive image of the sample. Such documents can be considered an extension of the table for the current granule, as opposed to other data products which must be defined as different granules/lines. The best solution identified consists in providing such links under the `datalink_url` parameter (as `dlmeta`) - the alternative idea to use specific parameters such as `document_url`, `image_url`... is deprecated.

target_class = 'sample', constant

target_name provides the name or ID of the sample — a value is mandatory in this case. It introduces the name of a meteorite or lunar sample when applicable. Various parts of the same sample can be indicated and described in `sample_desc` (such as "Location A", etc) or in `sample_id`.

measurement_type provides the type of measurement/scale as a UCD (REFF, I_over_F, etc...), and is completed by **spectrum_type** which is expected to be more accurate in general.

species is also available, see if usage can be enlarged (e.g., to InChiKeys), and desirable (absence of a species here may suggest that no measurement is available in a services, while usage will be limited to some species in practice).

original_publisher may be used to identify a source database in a collective service such as SSHADE

Special cases to handle:

- Photometric measurement sets: some wlv and many angles, distributed together
- Band lists: tables with characteristics and attributions - EPNCORE may not be the best possible solution, TBC
- Optical constants: use two associated spectra? Or only one if complex data are available.

14- APIS extension

A specific extension has been designed for compatibility with the APIS service, so that all services in the field of planetary aurorae can be queried together and consistently.

Definition here: [EPNcore extension for APIS](#)

This page contains a table of Extra parameters defined for APIS, which can be used for other services distributing observational data from a large facility.

15- Events

The following parameters are proposed to handle VOevents (**Work in progress**):

Use of general parameters

access_url is expected to link to a VOevent file (**TBC**) with **dataprodu****ct_type** = ev

access_format = text/xml (for broad identification) or = application/x-voevent+xml

instrument_host_name & **instrument_name** are expected to be "simulation" + reference of code used, including version number if the event is a prediction (standard use applies for observation)

target_name groups all targets in a hash list (e.g., Sun#Earth) - TBC

obs_id is identical for related events

event_type

Provides a type of event from a pre-defined list. Possible values include:

- meteor_shower
- fireball
- lunar_flash
- comet_tail_crossing

event_status

Following the VOevent standard, this can be:

- prediction
- observation
- utility (e. g., local event related to an instrument, like a change of detector)

event_cite

Following the VOevent standard, this can be:

- followup

- supersedes
- retraction (an alternative is to update the event lists)

16- Other extensions

Other extensions may be defined by topical working groups, based on data services and use cases. They will complete the current list.

Annex: EPN-TAP V2.0 parameters properties

Name	Value required (not NULL)?	SQL type	Unit / Format / Range	Description	UCD
granule_uid	Y	Text		Unique ID in data service.	meta.id
granule_gid	Y	Text		Common to granules of same type	meta.id
obs_id	Y	Text		Associates granules derived from the same data	meta.id;obs
dataprodukt_type	Y	Text		Organization of the data product, from enumerated list	meta.code.class
target_name		Text		Standard IAU name of target (must match target_class), case sensitive.	meta.id;src
target_class	Y	Text		Type of target, from enumerated list	src.class
time_min		Double	d (date as JD)	Start time (in JD). UTC measured at time_origin location (default is observer's frame)	time.start;obs
time_max		Double	d (date as JD)	Stop time (in JD). UTC measured at time_origin location (default is observer's frame)	time.end;obs
time_sampling_step_min		Double	s	Min time sampling step	time.resolution;stat.min
time_sampling_step_max		Double	s	Max time sampling step	time.resolution;stat.max
time_exp_min		Double	s	Min integration time	time.duration;obs.exposure;stat.min
time_exp_max		Double	s	Max integration time	time.duration;obs.exposure;stat.max

spectral_range_min		Double	Hz	Min spectral range (as frequency)	em.freq;stat.min
spectral_range_max		Double	Hz	Max spectral range (as frequency)	em.freq;stat.max
spectral_sampling_step_min		Double	Hz	Min spectral sampling step	<i>em.freq.step;stat.min</i>
spectral_sampling_step_max		Double	Hz	Max spectral sampling step	<i>em.freq.step;stat.max</i>
spectral_resolution_min		Double		Min spectral resolution (resolving power)	spect.resolution;stat.min
spectral_resolution_max		Double		Max spectral resolution (resolving power)	spect.resolution;stat.max
c1min		Double	(1) Longitude from 0. to 360. RA from 0. to 24.	Min of first coordinate	<i>pos;stat.min</i> pos.distance;stat.min or pos.radius;stat.min (does not exist) for spherical & cylindrical pos.eq.ra;stat.min for celestial pos.bodyrc.lon;stat.min for body pos.cartesian.x;stat.min for Cartesian <i>pos.healpix for healpix - TBC</i> empty ("") for none (and no unit)
c1max		Double	(1)	Max of first coordinate	<i>pos;stat.max, etc</i>
c2min		Double	(1) Latitude and Dec from -90. to +90.	Min of second coordinate	<i>pos;stat.min</i> pos.angDistance;stat.min or pos.az.zd;stat.min (for zenithal distance) for spherical or pos.az.azi;stat.min (for azimuth) for cylindrical pos.eq.dec;stat.min for celestial pos.bodyrc.lat;stat.min for body pos.cartesian.y;stat.min for Cartesian empty ("") for none (and no unit)

c2max		Double	(1)	Max of second coordinate	pos;stat.max, etc
c3min		Double	(1)	Min of third coordinate	pos;stat.min pos.AngDistance;stat.min or pos.az.azi;stat.min (for azimuth) for spherical pos.distance;stat.min for cylindrical pos.distance;stat.min for celestial pos.bodyrc.alt;stat.min for body pos.cartesian.z;stat.min for Cartesian empty ("") for none (and no unit)
c3max		Double	(1)	Max of third coordinate	pos;stat.max, etc
s_region		spoly	PgSphere spoly	ObsCore-like footprint for celestial or body-fixed frames.	phys.outline;obs.field
c1_resol_min		Double	(1)	Min resolution in first coordinate	pos.resolution;stat.min <i>if linear</i> pos.angResolution;stat.min <i>if angular</i>
c1_resol_max		Double	(1)	Max resolution in first coordinate	pos.resolution;stat.max <i>if linear</i> pos.angResolution;stat.max <i>if angular</i>
c2_resol_min		Double	(1)	Min resolution in second coordinate	pos.resolution;stat.min <i>if linear</i> pos.angResolution;stat.min <i>if angular</i>
c2_resol_max		Double	(1)	Max resolution in second coordinate	pos.resolution;stat.max <i>if linear</i> pos.angResolution;stat.max <i>if angular</i>
c3_resol_min		Double	(1)	Min resolution in third coordinate	pos.resolution;stat.min <i>if linear</i> pos.angResolution;stat.min <i>if angular (spherical only)</i>

c3_resol_max		Double	(1)	Max resolution in third coordinate	pos.resolution;stat.max <i>if linear</i> pos.angResolution;stat.min <i>if angular (spherical only)</i>
spatial_frame_type	Y	Text	(1)	Flavour of coordinate system, defines the nature of coordinates. From enumerated list. Use "none" if undefined.	meta.code.class;pos.frame
incidence_min		Double	deg	Min incidence angle (solar zenithal angle)	pos.incidenceAng;stat.min
incidence_max		Double	deg	Max incidence angle (solar zenithal angle)	pos.incidenceAng;stat.max
emergence_min		Double	deg	Min emergence angle	pos.emergenceAng;stat.min
emergence_max		Double	deg	Max emergence angle	pos.emergenceAng;stat.max
phase_min		Double	deg	Min phase angle	pos.phaseAng;stat.min
phase_max		Double	deg	Max phase angle	pos.phaseAng;stat.max
instrument_host_name		Text		Standard name of the observatory or spacecraft	meta.id;instr.obsty
instrument_name		Text		Standard name of instrument	meta.id;instr
measurement_type		Text		UCD(s) defining the data	meta.ucd
processing_level		Integer		Dataset-related encoding, or simplified CODMAC calibration level	meta.calibLevel
creation_date	Y	Timestamp	(ISO-8601 String)	Date of first entry of this granule	time.creation
modification_date	Y	Timestamp	(ISO-8601 String)	Date of last modification	time.processing
release_date	Y	Timestamp	(ISO-8601 String)	Start of public access period (set to creation_date if no proprietary period)	time.release
service_title	Y	Text		Title of resource = schema name	meta.title
Common optional parameters					

access_url		Text		URL of the data file, case sensitive (additional files may be linked through datalink_url). Can point to a script. If present, next 2 parameters must also be present.	meta.ref.url;meta.file
access_format		Text	(mime type in lowercase)	File format type	meta.code.mime
access_estsize		Integer	kbyte	Estimate file size in kbyte (with this spelling)	phys.size;meta.file
access_md5		Text		MD5 Hash for the file when available (real file, not script)	<i>meta.checksum;meta.file</i>
thumbnail_url		Text		URL of a thumbnail image with predefined size (png ~200 pix, for use in a client only)	meta.ref.url;meta.preview
file_name		Text		Name of the data file only, case sensitive	meta.id;meta.file
datalink_url		Text	(url)	Provides links to files or services on the server	meta.ref.url;meta.datalink
species		Text		Identifies a chemical species, <i>case sensitive</i>	meta.id;phys.atmol
waveband		Text		Electro-magnetic band, from enumerated list	instr.bandpass
alt_target_name		Text	can be a hash list	Provides alternative target name(s)	meta.id;src
target_region		Text		Type of region or feature of interest	obs.field
feature_name		Text		Secondary name (e.g. standard name of a region of interest)	obs.field
publisher		Text		Resource publisher	meta.curation

bib_reference		Text		Bibcode or doi preferred; can be a URL or anything else	meta.bib
internal_reference		Text		Related granule_uid(s) in the current service	meta.id.cross
external_link		Text	(url)	Web page providing more details on the <i>granule</i> .	meta.ref.url
spatial_coordinate_description		Text		ID of specific coordinate system and version / properties	meta.code.class;pos.frame
spatial_origin		Text		Defines the frame origin	meta.ref;pos.frame
time_origin		Text		Defines <i>where</i> the time is measured (e. g., ground vs spacecraft). Default is observer's frame.	meta.ref;time.scale
time_scale		Text		Always UTC in <i>data</i> services - from enumerated list	time.scale
subsolar_longitude_min		Double	deg	Minimum sub-solar point longitude	pos.bodyrc.lon;stat.min
subsolar_longitude_max		Double	deg	Maximum sub-solar point longitude	pos.bodyrc.lon;stat.max
subsolar_latitude_min		Double	deg	Minimum sub-solar point latitude	pos.bodyrc.lat;stat.min
subsolar_latitude_max		Double	deg	Maximum sub-solar point latitude	pos.bodyrc.lat;stat.max
subobserver_longitude_min		Double	deg	Minimum sub-observer point longitude (sub-Earth for ground based observations)	pos.bodyrc.lon;stat.min
subobserver_longitude_max		Double	deg	Maximum sub-observer point longitude (sub-Earth for ground based observations)	pos.bodyrc.lon;stat.max

subobserver_latitude_min		Double	deg	Minimum sub-observer point latitude (sub-Earth for ground based observations)	pos.bodyrc.lat;stat.min
subobserver_latitude_max		Double	deg	Maximum sub-observer point latitude (sub-Earth for ground based observations)	pos.bodyrc.lat;stat.max
ra		Double	deg	Right ascension	pos.eq.ra;meta.main
dec		Double	deg	Declination	pos.eq.dec;meta.main
radial_distance_min		Double	km	Min distance from center (in body-fixed frame)	pos.distance;pos.bodyrc;stat.min
radial_distance_max		Double	km	Max distance from center (in body-fixed frame)	pos.distance;pos.bodyrc;stat.max
altitude_fromshape_min		Double	km	Min altitude above shape model / DTM (in body-fixed frame)	pos.bodyrc.alt;stat.min
altitude_fromshape_max		Double	km	Max altitude above shape model / DTM (in body-fixed frame)	pos.bodyrc.alt;stat.max
solar_longitude_min		Double	deg	Min Solar longitude Ls (location on orbit / season)	pos.ecliptic.lon;pos.heliocentric;stat.min
solar_longitude_max		Double	deg	Max Solar longitude Ls (location on orbit / season)	pos.ecliptic.lon;pos.heliocentric;stat.max
local_time_min		Double	h	Min local time at observed region	time.period.rotation;time.phase;stat.min
local_time_max		Double	h	Max local time at observed region	time.period.rotation;time.phase;stat.max
target_distance_min		Double	km	Min observer-target distance	pos.distance;stat.min
target_distance_max		Double	km	Max observer-target distance	pos.distance;stat.max
target_time_min		Timestamp	(ISO-8601 String)	Min observing time in target frame	time.start;src
target_time_max		Timestamp	(ISO-8601 String)	Max observing time in target frame	time.end;src

earth_distance_min		Double	au	Min Earth-target distance	pos.distance;stat.min
earth_distance_max		Double	au	Max Earth-target distance	pos.distance;stat.max
sun_distance_min		Double	au	Min Sun-target distance	pos.distance;stat.min
sun_distance_max		Double	au	Max Sun-target distance	pos.distance;stat.max
Parameters from extensions					
obs_mode		Text		Observing mode	meta.code;instr.setup
detector_name		Text		Detector name	meta.id;instr.det
opt_elem		Text		Optical element name	meta.id;instr.param
filter		Text		Identifies filter in use, typically for images	meta.id;instr.filter
instrument_type		Text		type of instrument	meta.id;instr
acquisition_id		Text		ID of the data file/acquisition in the original archive	meta.id
proposal_id		Integer		Proposal identifier	meta.id;obs.proposal
proposal_pi		text		Proposal principal investigator	meta.id.PI;obs.proposal
proposal_title		Text		Proposal title	meta.title;obs.proposal
campaign		Text		Name of the observational campaign	meta.id;meta.code
target_description		Text		Original target keywords	meta.note;src
proposal_target_name		Text		target name as in proposal title	meta.note;obs.proposal
target_apparent_radius		Double		Apparent radius of the target	phys.angSize;src
north_pole_position		Double	deg	North pole position angle with respect to celestial north pole	pos.posAng
target_primary_hemisphere		Text		Primary observed hemisphere	meta.id;obs.field
target_secondary_hemisphere		Text		Secondary observed hemisphere	meta.id;obs.field

platesc		Double	arcsec/pixel	spatial resolution per pixel or platescale (on sky only)	instr.scale
orientation		Double		Position angle of image y axis (on sky only)	pos.posAng
observer_name		Text		Observer name	obs.observer;meta.main
observer_institute		Text		Observer institute	meta.note;meta.main
observer_id		Integer		Image observer's PVOL numeric identifier	meta.id.PI
observer_code		Text		Image observer's PVOL username	meta.id.PI
observer_country		Text		Image observer's country of residence	meta.note;obs.observer
observer_lon		Double		Observer's approximate longitude	obs.observer;pos.earth.lon
observer_lat		Double		Observer's approximate latitude	obs.observer;pos.earth.lat
mass		Double	kg	Mass of object	phys.mass
sidereal_rotation_period		Double	h	Object rotation rate	time.period.rotation
mean_radius		Double	km		phys.size.radius
equatorial_radius		Double	km		phys.size.radius
polar_radius		Double	km		phys.size.radius
diameter		double	km	Target diameter, or equivalent diameter for binary objects	phys.size.diameter
semi_major_axis		Double	au		phys.size.smajAxis
inclination		Double		Orbit inclination	src.orbital.inclination
eccentricity		Double		Orbit eccentricity	src.orbital.eccentricity
long_asc		Double	deg	Longitude of ascending node, J2000.0	src.orbital.node
arg_perihel		Double	deg	Argument of Perihelion, J2000.0	src.orbital.periastron
mean_anomaly		Double	deg	Mean anomaly at the epoch	src.orbital.meanAnomaly

dynamical_class		Text		Class of small body, from enumerated list	meta.code.class;src
dynamical_type		Text		Subdivision of the class, from enumerated list	meta.code.class;src
taxonomy_code		Text		Code for target taxonomy, from enumerated list	src.class.color
magnitude		Double	mag	Absolute magnitude. For small bodies, from HG magnitude system	phys.magAbs
flux		Double	mJy	Target flux	phot.flux.density
albedo		Double		Target albedo	phys.albedo
map_projection		Text		ID from enumerated list, or parameters (refers to a standard)	<i>pos.projection</i>
map_height		Double	pixel	Map size in px	phys.size
map_width		Double	pixel	Map size in px	phys.size
map_scale		Text		Format TBD	pos.wcs.scale
pixelscale_min		Double	km/pixel	Min pixel size on a surface	instr.scale;stat.min
pixelscale_max		Double	km/pixel	Max pixel size on a surface	instr.scale;stat.max
particle_spectral_type		Text		From enumerated list	meta.id ; phys.particle
particle_spectral_range_min		Double			phys.energy ; phys.particle ;stat.min phys.mass ; phys.particle ;stat.min
particle_spectral_range_max		Double			phys.energy ; phys.particle ;stat.max phys.mass ; phys.particle ;stat.max
particle_spectral_sampling_step_min		Double			spect.resolution ; phys.particle ;stat.min
particle_spectral_sampling_step_max		Double			spect.resolution ; phys.particle ;stat.max
particle_spectral_resolution_min		Double			spect.resolution ; phys.particle ;stat.min
particle_spectral_resolution_max		Double			spect.resolution ; phys.particle ;stat.max

original_publisher		Text		Refers to the source of the data, e. g., in compilations of experimental data	meta.note;meta.main
producer_name		Text		Data producer name, especially in compilations of experimental data	meta.note;meta.main
producer_institute		Text		Data producer institute, e. g., in compilations of experimental data	meta.note;meta.main
sample_id		Text		Provides a local ID in an existing catalogue	meta.id;src
sample_classification		Text	hash list	Information related to class, sub-class, species...	meta.note;phys.composition
sample_desc		Text	can be a hash list	Describes the sample, its origin, and possible preparation	meta.note
data_calibration_desc		Text	can be a hash list	Provides information on post-processing	meta.note
setup_desc		Text	can be a hash list	Describes the experimental setup	meta.note
geometry_type		Text	can be a hash list	Type of observation, from enumerated list	meta.note;instr.setup
spectrum_type		Text	can be a hash list	Type of spectral observation, from enumerated list TBD	meta.note;instr.setup
grain_size_min		Double	μm	Min particle size in μm	phys.size;stat.min
grain_size_max		Double	μm	Max particle size in μm	phys.size;stat.max
azimuth_min		Double	deg	Min azimuth angle for illumination	pos.azimuth;stat.min
azimuth_max		Double	deg	Max azimuth angle for illumination	pos.azimuth;stat.max
pressure		Double	bar	Ambient pressure	phys.pressure

measurement_atmosphere		text		Describes experimental conditions. "vacuum" for measurements under vacuum.	meta.note;phys.pressure
temperature		Double	K	Ambient temperature	phys.temperature
event_type		Text		Type of event from enumerated list	TBD
event_status		Text		From enumerated list	TBD
event_cite		Text		From enumerated list	TBD

(1): depending on context (as given by spatial_frame_type).