EPN-TAP V2.0 parameters

Note: from 5 July 2021, these pages are no longer the primary source of the EPN-TAP doc - see instead the IVOA latex source: https://github.com/ivoa-std/EPNTAP

List of EPN-TAP parameters
(refreshed/completed April 2019, Oct 2020, June 2021, Oct 2021) (SE)

You can use this file to keep track of your service parameters: EPN-TAP_parameters_List_template.xlsx

EPN-TAP

- EPN-TAP is a VO access protocol dedicated to Planetary Science data. It is based on the TAP mechanism from IVOA, completed with sets of parameters and associated lists of values. In this regard, it is similar to ObsTAP but with a different scope.
- EPN-TAP version 2 is a major update of the protocol to accommodate larger services and simplify setup and use of data services. All parameters are described here.

Parameters which must be provided

are now clearly identified - those are not only mandatory, they also must provide a value.
They are mostly related to service description and granule identification.

+ See notes below the table.

Thematic extensions

Some science fields will require optional parameters, which need to be used consistently between services addressing the same field. Such extensions have to be designed by sub-groups involved in the corresponding data services, either as providers or users. This includes:

- Lab spectroscopy: parameters to describe mineralogical samples (and possibly other samples)
- Solar System objects: covers orbital/rotational parameters, physical properties, and taxonomy
- APIS: for consistency with APIS service. Contains parameters for observing programs (most parameters are actually included in other extensions)
- Contributive works / observing programs: enlargement of APIS extension to other data
- Exoplanets / planetary systems properties
- Map extension (to be enlarged)
- Events: covers the VOevent standard and other types of events
- Particle spectroscopy (to be finalized)
- Results of planetary 3D modelling run (in progress)
- Bibliographic entries? May be manageable otherwise, through bibcode / doi interpretation

Support file

You can use this file to keep track of your service parameters: EPN-TAP_parameters_List_template.xlsx

Name	SQL type	Unit	Description	UCD	UCD in Obscore 1.1 (9/5/2017 REC version)	Utype (tentative)	Comments
EPNCore mandatory parameters (Must be present, possibly empty) **(bold face: a value is required)**				Current value current but dubious or *undefined*	— ? : closest sense _ : N/A in ObsCore	from epntap v2 mixin (aug 2017) equivalent /close in ObsCore doc 1.1	
granule_uid	Text		Unique ID in data service	meta.id	meta.id		Can be alphanum.
granule_gid	Text		Common to granules of same type	meta.id	meta.id		E.g. same map projection, or geometry data products. Can be alphanum.
obs_id	Text		Associates granules derived from the same data	meta.id; obs	meta.id	obscore: DataID. observationID	E.g. various representations / processing levels. Can be alphanum., may be the ID of original observation. Keep it simple in intricate situations.

dataproduct_type	Text		Organization of the data product, from enumerated list	meta.code.class	meta.id	Epn.dataProductType obscore: ObsDataset.dataProductType	
measurement_type	Text		UCD(s) defining the data	meta.ucd	meta.ucd	Epn.Measurement_type	Add ;meta.modelled if simulation or model Add ;stat.uncalib if uncalibrated data - in which case processing_level must be 0 or 1
processing_level	Integer		Dataset-related encoding, or simplified CODMAC calibration level	meta.calibLevel	meta.code;obs.calib	~ obscore: ObsDataset.calibLevel	To be replaced by PDS4 values in v2.1?
target_name	Text		Standard IAU name of target (must match target_class), case sensitive	meta.id;src	meta.id;src	Epn.TargetName	Case sensitive Services with no target_name do exist
target_class	Text		Type of target, from enumerated list	src.class	src.class	Epn.TargetClass	
time_min	Double	d	Start time (in JD). UTC measured at time_origin location (default is observer's frame)	time.start;obs	time.start;obs.exposure	Char.TimeAxis.Coverage.Bounds.Limits.Interval.StartTime	
time_max	Double	d	Stop time (in JD). UTC measured at time_origin location (default is observer's frame)	time.end;obs	time.end;obs.exposure	Char.TimeAxis.Coverage.Bounds.Limits.Interval.StopTime	
time_sampling_step_min	Float	s	Min time sampling step	time.resolution;stat.min	time.resolution	Epn.Time.Time_sampling_step_min	
time_sampling_step_max	Float	s	Max time sampling step	time.resolution;stat.max		Epn.Time.Time_sampling_step_max	
time_exp_min	Float	s	Min integration time	time.duration;obs.exposure;stat.min	time.duration;obs.exposure	Epn.Time.Time_exp_min	
time_exp_max	Float	s	Max integration time	time.duration;obs.exposure;stat.max		Epn.Time.Time_exp_max	
spectral_range_min	Float	Hz	Min spectral range (as frequency)	em.freq;stat.min	em.wl;stat.min (always as wl)	Epn.Spectral.Spectral_range_min	Always as frequency
spectral_range_max	Float	Hz	Max spectral range (as frequency)	em.freq;stat.max	em.wl;stat.max	Epn.Spectral.Spectral_range_max	
spectral_sampling_step_min	Float	Hz	Min spectral sampling step	em.freq;spect.binSize;stat.min	meta.number	Epn.Spectral.Spectral_sampling_step_min	
spectral_sampling_step_max	Float	Hz	Max spectral sampling step	em.freq;spect.binSize;stat.max	meta.number	Epn.Spectral.Spectral_sampling_step_max	
spectral_resolution_min	Float		Min spectral resolution (resolving power)	spect.resolution;stat.min	spect.resolution (relates to resolving power)	Epn.Spectral.Spectral_resolution_min	Now (2019) provides resolving power |(lambda / delta(lambda)| = |f /Df| How do we accommodate FWHM for filters?
spectral_resolution_max	Float		Max spectral resolution (resolving power)	spect.resolution;stat.max		Epn.Spectral.Spectral_resolution_max	Now (2019) provides resolving power |(lambda / delta(lambda)| = |f /Df| How do we accommodate FWHM for filters?
c1min	Float	(1)	Min of first coordinate, depends on the frame	see table below	pos.eq.ra	Epn.Spatial.Spatial_range.c1min	Typo in current mixin (.lonG => .lon UCDs for cyl and sph coord are from PEN-UCDlist-20210430
c1max	Float	(1)	Max of first coordinate, depends on the frame			Epn.Spatial.Spatial_range.c1max	
c2min	Float	(1)	Min of second coordinate, depends on the frame.		pos.eq.dec	Epn.Spatial.Spatial_range.c2min	
c2max	Float	(1)	Max of second coordinate, depends on the frame			Epn.Spatial.Spatial_range.c2max	

c3min	Float	(1)	Min of third coordinate			Epn.Spatial.Spatial_range.c3min	
c3max	Float	(1)	Max of third coordinate			Epn.Spatial.Spatial_range.c3max	
s_region	Text	(3)	ObsCore-like footprint in 2D (if spatial_frame_type = celestial or body)	pos.outline;obs.field	pos.outline;obs.field	obscore:Char.SpatialAxis.Coverage.Support.Area	(was initially instr.fov, to be corrected) ObsCore value updated (was phys.angArea;obs) to phys.outline, then corrected to pos.outline Must have xtype= adql:REGION to work with TAP Frame may be identified in q.rd (UNKNOWNFrame). Use value given in spatial_frame_type - very unclear… Do we need another param for GIS interface?
c1_resol_min	Float	(2)	Min resolution in first coordinate	(2)	pos.angResolution;stat.min	Epn.Spatial.Spatial_resolution.c1_resol_min	pos.resolution restored in 2018 In body fixed frame, use pixelscale_min/max for resolution at the surface
c1_resol_max	Float	(2)	Max resolution in first coordinate	(2)	pos.angResolution;stat.max	Epn.Spatial.Spatial_resolution.c1_resol_max	_
c2_resol_min	Float	(2)	Min resolution in second coordinate	(2)		Epn.Spatial.Spatial_resolution.c2_resol_min	_
c2_resol_max	Float	(2)	Max resolution in second coordinate	(2)		Epn.Spatial.Spatial_resolution.c2_resol_max	_
c3_resol_min	Float	(2)	Min resolution in third coordinate	(2)		Epn.Spatial.Spatial_resolution.c3_resol_min	pos.resolution restored in 2018
c3_resol_max	Float	(2)	Max resolution in third coordinate	(2)		Epn.Spatial.Spatial_resolution.c3_resol_max	pos.resolution restored in 2018
spatial_frame_type	Text	(1)	Flavor of coordinate system, defines the nature of coordinates. From enumerated list. Use "none" if undefined	meta.code.class;pos.frame	_		A value is required by DaCHS (query will return errors if empty) Default value = none
incidence_min	Float	deg	Min incidence angle (solar zenithal angle)	pos.incidenceAng;stat.min	_	Epn.View_angle.Incidence_angle_min	UCD for angles included in 2018
incidence_max	Float	deg	Max incidence angle (solar zenithal angle)	pos.incidenceAng;stat.max	_	Epn.View_angle.Incidence_angle_max	UCD for angles included in 2018
emergence_min	Float	deg	Min emergence angle	pos.emergenceAng;stat.min	_	Epn.View_angle.Emergence_angle_min	UCD for angles included in 2018
emergence_max	Float	deg	Max emergence angle	pos.emergenceAng;stat.max	_	Epn.View_angle.Emergence_angle_max	UCD for angles included in 2018
phase_min	Float	deg	Min phase angle	pos.phaseAng;stat.min		Epn.View_angle.Phase_angle_min	
phase_max	Float	deg	Max phase angle	pos.phaseAng;stat.max		Epn.View_angle.Phase_angle_max	
instrument_host_name	Text		Standard name of the observatory or spacecraft	meta.id;instr.obsty	meta.id;instr.tel	Provenance.ObsConfig.Facility.name	
instrument_name	Text		Standard name of instrument	meta.id;instr	meta.id;instr	Provenance.ObsConfig.Instrument.name	
service_title	Text		Title of resource = schema name	meta.title			May be used to handle multiservice results
creation_date	Timestamp	(4)	Date of first entry of this granule	time.creation	time;meta.dataset		
modification_date	Timestamp	(4)	Date of last modification	time.processing			Used to handle mirroring UCD value being discussed in 2018

release_date	Timestamp	(4)	Start of public access period (set to creation_date if no proprietary period)	time.release	time.release	obscore:Curation.releaseDate	The value is in ISO 8601 format reusing this pattern: ("YYYY-MM-DDThh:mm:ss") If release_date is in the future, the data is proprietary.
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	meta.ref.url	Obs.Access.Reference	Use this to link data! Could accommodate a datalink with access_format = 'application/x-votable+xml;content=datalink' (from ObsCore) - but this is a funny idea…
access_format	Text		RFC 2045 media type (mime), required to be all-lower case	meta.code.mime	meta.code.mime	Obs.Access.Format	
access_estsize	Integer	kbyte	Estimate file size in kbyte (with this spelling)	phys.size;meta.file	phys.size;meta.file	Obs.Access.Size	
access_md5	Text		MD5 Hash for the file when available (real file, not script)	meta.checksum;meta.file			
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	meta.title;obs — ?		ObsCore obs_title is for a short free text string describing the granule. Do we want this?
datalink_url	Text		Provides links to files or services on the server	meta.ref.url			Associated mime-type is 'application/x-votable+xml;content=datalink' (from ObsCore)
bib_reference	Text		Bibcode or doi preferred; can be a URL or anything else. Refers to the *granule*	meta.bib	meta.bib	obscore:Curation.reference	Bibcode & doi can be completed in TOPCAT
publisher	Text		Resource publisher	meta.curation	meta.ref.uri;meta.curation	~ obscore:Curation.publisherID	
processing_level_desc	Text		Describes specificities of the processing level	meta.note			
internal_reference	Text		Related granule_uid(s) in the current service	meta.id.cross			Use to link one granule to a set of other granules. To be used only if required - e.g. to solve situations that would otherwise require several tables
external_link	Text		Web page providing more details on the *granule*	meta.ref.url			Link to an individual page in a web site associated to the database, e.g., a planet page in Exoplanets service. This is a way to provide extra granule information which cannot be accommodated in the table.
species	Text		Identifies a chemical species, *case sensitive*	meta.id;phys.atmol			This is the only case sensitive parameter (with target_name)
messenger	Text		Vector of measured signal, including electromagnetic band, from enumerated list	instr.bandpass			
filter	Text		Identifies filter in use, typically for images	meta.id;instr.filter			Informative only, free format (no list, but see http://svo2.cab.inta-csic.es/svo/theory/fps3/). Search can only rely on spectral range, as ObsCore does.
alt_target_name	Text		Provides alternative target name(s). Can be a hash list	meta.id;src			
feature_name	Text		Secondary name (e.g. standard name of a region of interest)	meta.id;src;obs.field			
target_region	Text		Type of region or feature of interest	meta.id;src;obs.field			
shape	Text		introduces an ascii (ST)MOC, v2 (2D footprint on celestial, spherical, or body-related frames, possibly including time)	pos.outline;obs.field			Must have xtype="MOC" (follow DALI recommendation) outline doesn't fit definition (refers to a contour)
spatial_coordinate_description	Text		ID of specific coordinate system and version / properties	meta.code.class;pos.frame			~COOSYS, but includes planetary ones Still TBD, needs to be OGC compliant. Discussion in progress here: EPN-TAP v2: Current discussion topic
spatial_origin	Text		Defines the frame origin	meta.ref;pos.frame			
time_refposition	Text		Defines *where* the time is measured (e.g., ground vs spacecraft). Default is observer's frame	meta.ref;time.scale			target_time is of course always on target.
time_scale	Text		Defaults to UTC in *data* services - from enumerated list	time.scale			

solar_longitude_min	Float	deg	Min Solar longitude Ls (location on orbit / season)	pos.ecliptic.lon;pos.heliocentric;stat.min			
solar_longitude_max	Float	deg	Max Solar longitude Ls (location on orbit / season)	pos.ecliptic.lon;pos.heliocentric;stat.max			
local_time_min	Float	h	Min local time at observed region	time.phase; time.period.rotation; stat.min			
local_time_max	Float	h	Max local time at observed region	time.phase; time.period.rotation; stat.max			
target_distance_min	Float	km	Min observer-target distance	pos.distance; stat.min			
target_distance_max	Float	km	Max observer-target distance	pos.distance; stat.max			
target_time_min	Timestamp	(4)	Min observing time in target frame	time.start;src			(simplest way to look for coordinated observations)
target_time_max	Timestamp	(4)	Max observing time in target frame	time.end;src			
earth_distance_min	Float	AU	Min Earth-target distance	pos.distance; stat.min			
earth_distance_max	Float	AU	Max Earth-target distance	pos.distance; stat.max			
sun_distance_min	Float	AU	Min Sun-target distance	pos.distance; stat.min			
sun_distance_max	Float	AU	Max Sun-target distance	pos.distance; stat.max			
subobserver_longitude_min	Float	deg	Minimum sub-observer point longitude (sub-Earth for ground based observations)	pos.bodyrc.lon;stat.min			_
subobserver_longitude_max	Float	deg	Maximum sub-observer point longitude (sub-Earth for ground based observations)	pos.bodyrc.lon;stat.max			_
subobserver_latitude_min	Float	deg	Minimum sub-observer point latitude (sub-Earth for ground based observations)	pos.bodyrc.lat;stat.min			_
subobserver_latitude_max	Float	deg	Maximum sub-observer point latitude (sub-Earth for ground based observations)	pos.bodyrc.lat;stat.max			_
subsolar_longitude_min	Float	deg	Minimum sub-solar point longitude	pos.bodyrc.lon;stat.min			Provided in the most natural body-related coordinate frame, E-handed - seems to require 'body'
subsolar_longitude_max	Float	deg	Maximum sub-solar point longitude	pos.bodyrc.lon;stat.max			Provided in the most natural body-related coordinate frame, E-handed - seems to require 'body'
subsolar_latitude_min	Float	deg	Minimum sub-solar point latitude	pos.bodyrc.lat;stat.min			_
subsolar_latitude_max	Float	deg	Maximum sub-solar point latitude	pos.bodyrc.lat;stat.max			_
ra	Float	deg	Right ascension	pos.eq.ra;meta.main			deg only (like ObsCore)
dec	Float	deg	Declination	pos.eq.dec;meta.main			

radial_distance_min	Float	km	Min distance from observed area to body center	pos.distance;pos.bodyrc;stat.min			
radial_distance_max	Float	km	Max distance from observed area to body center	pos.distance;pos.bodyrc;stat.max			
altitude_fromshape_min	Float	km	Min altitude of observed area above shape model / DTM	pos.bodyrc.alt;stat.min			
altitude_fromshape_max	Float	km	Max altitude of observed area above shape model / DTM	pos.bodyrc.alt;stat.max			
Parameters from extensions							
APIS extension							
obs_mode	Text		Observing mode	meta.code;instr.setup			From APIS + observation extensions (with adapted UCDs)
detector_name	Text		Detector name	meta.id;instr.det			
opt_elem	Text		Optical element name	meta.id;instr.param			
instrument_type	Text		Type of instrument	meta.id;instr			Informative only (not a reliable search parameter): free format, no reference list intended.
acquisition_id	Text		ID of the data file/acquisition in the original archive	meta.id			
proposal_id	Text		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;obs.proposal			
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	Float	arcsec	Apparent radius of the target	phys.angSize;src			
north_pole_position	Float	deg	North pole (of target) position angle with respect to celestial north pole	pos.posAng			Group of 5 parameters very specific to APIS. Name is ~ OK, but actually provides the position angle of the planet axis. Use "orientation" for the image.
target_primary_hemisphere	Text		Primary observed hemisphere	meta.id;obs.field			
target_secondary_hemisphere	Text		Secondary observed hemisphere	meta.id;obs.field			
platesc	Float	arcsec/pix	Pixel angular size or platescale (on sky only)	instr.scale			
orientation	Float	deg	Position angle of image y axis (on sky only), direct sense from north direction	pos.posAng			Provides the direction of the polar axis in the image, counted clockwise from north.
measurement_unit	Text		Physical unit, same as Bunit in fits	meta.unit			
Contributive work extension							
observer_name	Text		Observer name	meta.id.PI;obs.observer			
observer_id	Integer		Observer's numeric identifier	meta.id.PI			Group of 5 from PVOL, OK for general use but UCDs have to be changed in PVOL. meta.pubid in PVOL

observer_code	Text		Observer's service username	meta.id.PI			meta.pubcode in PVOL
observer_institute	Text		Observer institute	meta.note			
observer_country	Text		Observer's country of residence	meta.note;obs.observer			meta.pubcountry in PVOL
observer_location	Text		Broad location of the observer or telescope. Can be used when the exact location cannot be released	pos;obs.observer - not in mixin!			meta.pubcountry in PVOL
observer_lon	Float	deg	Observer's approximate longitude	obs.observer;pos.earth.lon			meta.publon in PVOL
observer_lat	Float	deg	Observer's approximate latitude	obs.observer;pos.earth.lat			meta.publat in PVOL
original_publisher	Text		Refers to the source of the data, e.g., in compilations of observations or experimental data	meta.note			Experimental spectroscopy + contributive work extensions
Solar System objects extension							
mean_radius	Float	km		phys.size.radius			
equatorial_radius	Float	km		phys.size.radius			
polar_radius	Float	km		phys.size.radius			
diameter	Float	km	Target diameter, or equivalent diameter for binary objects	phys.size.diameter			Used in tnosarecool, not very consistent (use radius?)
mass	Float	kg	Mass of object	phys.mass			Solar System Objects extension (generic values in catalogues, not observations)
sidereal_rotation_period	Float	h	Object rotation rate	time.period.rotation			
semi_major_axis	Float	AU		phys.size.smajAxis			
inclination	Float	deg	Orbit inclination	src.orbital.inclination			
eccentricity	Float		Orbit eccentricity	src.orbital.eccentricity			
long_asc	Float	deg	Longitude of ascending node, J2000.0	src.orbital.node			
arg_perihel	Float	deg	Argument of perihelion, J2000.0	src.orbital.periastron			
mean_anomaly	Float	deg	Mean anomaly at the epoch	src.orbital.meanAnomaly			
epoch	Double	d	Epoch of interest in JD	time.epoch			
magnitude	Float	mag	Absolute magnitude. For small bodies, from HG magnitude system	phys.magAbs			Actually depends on service (eg, spectro_planets vs DynAstVO vs tnosarecool). UCD may include mention of the photometric band.
flux	Float	mJy	Target flux	phot.flux.density			
albedo	Float		Target albedo	phys.albedo			
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	src.class.color			Possible values depend on target type and possibly on service

Map extension							
map_projection	Text		ID from enumerated list, or string with parameters (referring to a standard)	*pos. projection*			Map extension
map_height	Float	pix	Map size in px	phys. size			
map_width	Float	pix	Map size in px	phys. size			
map_scale	Text		Preferably a ratio (e. g., "1:50000")	pos.wcs. scale			
pixelscale_min	Float	km/pix	Min pixel size on a surface	instr. scale; stat.min			
pixelscale_max	Float	km/pix	Max pixel size on a surface	instr. scale; stat.max			
Particle spectroscopy extension							
particle_spectral_type	Text		From enumerated list	meta.id; phys. particle			
particle_spectral_range_min	Float			phys. energy; phys. particle; stat.min phys. mass; phys. particle; stat.min			
particle_spectral_range_max	Float			phys. energy; phys. particle; stat.max phys. mass; phys. particle; stat.max			
particle_spectral_sampling_step_min	Float			spect. resolution;phys. particle; stat.min			
particle_spectral_sampling_step_max	Float			spect. resolution;phys. particle; stat.max			
particle_spectral_resolution_min	Float			spect. resolution;phys. particle; stat.min			
particle_spectral_resolution_max	Float			spect. resolution;phys. particle; stat.max			
Experimental spectroscopy extension							
producer_name	Text		Data producer name, especially in compilations of experimental data	meta. note			
producer_institute	Text		Data producer institute, e. g., in compilations of experimental data	meta. note			
sample_id	Text		Provides a local ID in an existing catalogue	meta.id; src			In addition to target_name
sample_classification	Text		Information related to class, sub-class, species… as hash list	meta. note; phys. composition			This uses standard names for classes…
sample_desc	Text		Describes the sample, its origin, and possible preparation. Can be a hash list	meta. note			
species_inchikey	Text		Fixed length string identifying the species. Can be a hash list	meta.id; phys. atmol			Follows IUPAC standard (Heller et al 2015)

grain_size_min	Float	um	Min sample particle size	phys.size;stat.min			
grain_size_max	Float	um	Max sample particle size	phys.size;stat.max			
azimuth_min	Float	deg	Min azimuth angle for illumination	pos.azimuth;stat.min			*Check meaning/requirements for <0 values?* UCD added in 2018 (instead of pos.azimuthAng requested - OK)
azimuth_max	Float	deg	Max azimuth angle for illumination	pos.azimuth;stat.max			UCD added in 2018
pressure	Float	bar	Ambient pressure	phys.pressure			VOunits says: Pascal.
measurement_atmosphere	Text		Describes experimental conditions. "vacuum" for measurements under vacuum	meta.note;phys.pressure			
temperature	Float	K	Ambient temperature	phys.temperature			
setup_desc	Text		Describes the experimental setup. Can be a hash list	meta.note			May include Aperture (size of sample measured), etc
data_calibration_desc	Text		Provides information on post-processing. Can be a hash list	meta.note			*(preferably to a "comment" parameter)*
geometry_type	Text		Type of observation, from enumerated list. Can be a hash list	meta.note;instr.setup			
spectrum_type	Text		Type of spectral observation, from enumerated list TBD. Can be a hash list	meta.note;instr.setup			Alternative to UCD, very detailed
Event extension							
event_type	Text		Type of event from enumerated list	meta.code.class			Events extension *If dataproduct_type = ev* *UCDs should be provided with the standard*
event_status	Text		From enumerated list	meta.code.status			
event_cite	Text		From enumerated list	meta.code.status			

(1): depending on context (as given by spatial_frame_type), see table below

Longitude and RA range from 0. to 360; Latitude and Dec range from -90. to +90.

For spatial_frame_type = "none": no value is provided, UCD are empty strings (""), and no unit is provided

(2) Spatial resolution parameters have the same unit as spatial coordinate parameters. The associated UCD combine either pos.resolution (if linear) or pos.angResolution (if angular) with secondary stat.min or stat.max

c1: only body and celestial are angular; c2: only cartesian is linear; c3: only spherical is angular

(3): Any contour type that works with ADQL's geometry operators (CONTAINS, INTERSECTS...) is legal here

(4): Timestamps are provided as ISO-8601 String as specified by DALI. On VOtable output, xtype="timestamp" attribute is required.

Frame coordinates UCD /units	**celestial**	**body**	**cartesian**	**spherical**	**cylindrical**
c1min	pos.eq.ra;stat.min	pos.bodyrc.lon;stat.min	pos.cartesian.x;stat.min (in km)	pos.spherical.r;stat.min (in m)	pos.cylindrical.r;stat.min (in km)
c1max	pos.eq.ra;stat.max	pos.bodyrc.lon;stat.max	pos.cartesian.x;stat.max (in km)	pos.spherical.r;stat.max (in m)	pos.cylindrical.r;stat.max (in km)
c2min	pos.eq.dec;stat.min	pos.bodyrc.lat;stat.min	pos.cartesian.y;stat.min (in km)	pos.spherical.colat;stat.min	pos.cylindrical.azi;stat.min
c2max	pos.eq.dec;stat.max	pos.bodyrc.lat;stat.max	pos.cartesian.y;stat.max (in km)	pos.spherical.colat;stat.max	pos.cylindrical.azi;stat.max

c3min	pos.distance; stat.min (in AU)	pos.bodyrc.alt;stat.min (from surface only, implicitly from reference level) ~~or~~ ~~pos.distance;pos.bodyrc;stat.min (from center)?~~ (in km)	pos.cartesian.z; stat.min (in km)	pos.spherical.azi; stat.min	pos.cylindrical.z; stat.min (height, in km)
c3max	pos.distance; stat.max (in AU)	pos.bodyrc.alt;stat.max (from surface only, implicitly from reference level) ~~or~~ ~~pos.distance;pos.bodyrc;stat.max (from center)?~~ (in km)	pos.cartesian.z; stat.max (in km)	pos.spherical.azi; stat.max	pos.cylindrical.z; stat.max (height, in km)

Example table for IDs:

File name-type	granule_uid	granule_gid	obs_id
A-Raw	1	native	A
A-Calib	2	calibrated	A
A-geom	3	geometry	A
A-proj	4	projected	A
B-Raw	5	native	B
B-Calib	6	calibrated	B
B-geom	7	geometry	B
B-proj	8	projected	B

Syntax

• **multivalued lists** = first entry#second entry#…#last entry, or scalar (with no #)
Values separator = #
No quotes around the list

This can be parsed by ADQL/RegTAP function ivo_hashlist_has like this:
*select * from vvex.epn_core where 1 = ivo_hashlist_has(lower(target_name),'Venus')*

Where the *lower* function is mandatory to handle values possibly containing upper cases (this is implicit on the 2nd argument)

Beware that only complete elements between separators will be found. The provider has to split the string according to expected searches, e.g.:
Composite Infrared Spectrometer#CIRS
not ~~*Composite Infrared Spectrometer (CIRS)*~~

Parameters supporting multivalued lists include:
dataproduct_type *(only when present in the same file; best avoided when possible)*
target_name *(for different targets only, but only one target can be described in the granule; use alt_target_name for other names of the same target)*
alt_target_name
target_class (*in association with target_name*)
instrument_host_name *(e.g. acronym and full name)*
instrument_name *(e.g. acronym and full name)*
measurement_type *(when present in the same file)*
processing_level *(when present in the same file)*
bib_reference

• **NULL and special values**:
A standard query on a parameter will not return granules with NULL/void value. E.g. target_name LIKE '%toto%' will only select granules with this value (standard ADQL behavior).
NULL/void has to be tested specifically (e.g., when it means "I don't know") using the IS operator (IS is used only to test the NULL value in ADQL):

> target_name LIKE '%toto%' OR target_name IS NULL
>
> Syntax IS NULL stands for both strings and numerical parameters (the = operator is accepted in this context only by latest DaCHS servers)

No inf, inf, or NaN value in ADQL? At least Inf/-Inf should be there, as per DALI.

• **UCDs**: the above table has been reviewed against the UCD documents, including latest discussions (4/2019). Review against PDS4 and IPDA to be performed.

> 2018 discussions / conclusions have been included here: https://wiki.ivoa.net/twiki/bin/view/IVOA/UCDList1dot42017June2018FebRFM

• ***_min vs *_max parameters**:

If only one value is available, it must appear in both fields

• **Optional parameters**: some of these come in sets that are logically related; if one is present, the related ones must be present also (e. g., 3 access_* parameters)

• **Granule_gid**: any general indication to providers? I.e.: preview, native, calibrated, geometry…
A client should be able to display the values present in a service (feasible in TOPCAT)

• Reshuffle previous "service parameters":

- Mandatory :
  publisher - make it mandatory???
  add publisher_did as in Obscore? (for DaCHS/registry; provides unique ID of service for this publisher/server), with UCD = meta.ref.uri;meta.curation
- Optional
  spatial_coordinate_description (default = none)
  spatial_origin (default = body center or SS barycenter? Or observer location)
  time_origin (default = observer)
  time_scale (default = UTC – no other values allowed in *data* services? [only in computational services, e.g. ephemeris])
  Same values to be used in registry declaration

• **Call-back parameters / reference**

Currently using service_title (= schema name) + granule_uid.

May use ivoID in the future.

• **Other parameters**

The most recent extra parameters often have names starting in prefix_*, where prefix identify the scope or context (e.g., spase_, vims_, image_, etc). Seems to be a good practice.

Accref is introduced by the EPN-TAP localfile mixin, but not used - in principle not included in TAP response, be may be present anyway. It may be better to hide it also in the portal.

• **Parameters introducing error bars/uncertainties**

Some parameters providing a scalar value X in the EPN-TAP table may be associated with an error bar in a related parameter. This is currently (4/2019) entered as:

In Basecom: Xerr (to be changed when upgrading to mixin version)
In DynAstVO: X_error
In Exoplanets: X_error_min; X_error_max
In planets: X_uncertainty
TNOsarecool: X_sigma_plus, X_sigma_minus (to be changed?)

The associated UCDs ***start*** with stat.error; or stat.error;stat.min; & stat.error;stat.max;

• **Support for PDS3 detached labels** (proposal)

Solution with datalink seems OK: data files under access_url and detached labels provided under datalink_url in a link table - although no attempt made to read them from the portal yet (use VIR unpublished service to test this).

• **Utypes**

Need to clean up current doc (2.0). Utypes are = DM fields. They are supposedly used to identify the meaning of parameters and help e. g. tools to grab required quantities - This will not work in some areas though, e. g. with spectral tools as they currently use UCD instead of Utype for this purpose (not many tools appear to actually rely on Utype in fact). See discussion here for usage (a bit old?): http://www.ivoa.net/documents/Notes/UTypesUsage/20130213/NOTE-utypes-usage-1.0-20130213.html

To handle this in practice:

- Associate each parameter to a specific Utype in EPNCore - all names need to start with the epncore: prefix/namespace.
- Then map epncore Utype to other DM (find equivalent parameter, or trace back the original templates of EPNCore parameters - often from ObsCore)
- Reuse Utype from other DM each times it makes sense - TBC: the epncore: namespace is still required (even when using Utypes from other DM)
  This allows tools to handle EPNCore parameters like existing parameters from other DM, i.e., with no specific implementation
  Pb is that small differences in the use of parameters may preclude reusing the same Utype (TBC: does that applies to units also?)
- Cross our fingers: known Utype (from other DMs) may be usable in existing tools (e.g. a tool supporting Provenance would grab equivalent info in EPN-TAP services transparently)
  Unclear if the use of the namespace makes it more complicated in tools.

Example of V2 with APIS database:

EPNcore Table v2

granule_uid	granule_gid	obs_id	access_url	access_format	thumbnail_url
o5g202x4q_x2d	original_data	o5g202x4q	o5g202x4q_x2d.fits	image/fits	o5g202x4q_x2d_small.jpg
o5g202x4q_x2d_prev	original_data_preview	o5g202x4q	o5g202x4q_x2d.jpg	image/jpg	o5g202x4q_x2d_small.jpg
o5g202x4q_proc	processed_data	o5g202x4q	o5g202x4q_proc.fits	image/fits	o5g202x4q_proc_small.jpg
o5g202x4q_proc_prev	processed_data_preview	o5g202x4q	o5g202x4q_proc.jpg	image/jpg	o5g202x4q_proc_small.jpg
o5g202x4q_cyl	cylindrical_projection	o5g202x4q	o5g202x4q_cyl.jpg	image/jpg	o5g202x4q_cyl_small.jpg
o5g202x4q_pol_n	polar_projection_north	o5g202x4q	o5g202x4q_pol_n.jpg	image/jpg	o5g202x4q_pol_n_small.jpg
o5g202x4q_pol_s	polar_projection_south	o5g202x4q	o5g202x4q_pol_s.jpg	image/jpg	o5g202x4q_pol_s_small.jpg