Europlanet-RI /
Europlanet Table Access Protocol

Draft 0.26 (in progress)
11 / 06 / 2012

EC Grant agreement n°228319

AUTHORS

Pierre Le Sidaner
DIO / OV, Observatoire de Paris
Stéphane Erard
LESIA, Observatoire de Paris / CNRS
Baptiste Cecconi
CDPP / LESIA, Observatoire de Paris / CNRS
Michel Gangloff
CDPP / IRAP, CNRS, Université Paul Sabatier
Jérôme Berthier
IMCCE, Observatoire de Paris / CNRS
Nicolas André
CDPP / IRAP, CNRS, Université Paul Sabatier
Natacha Bourrel
CDPP / IRAP, CNRS, Université Paul Sabatier

Document change record

Issue	Date	Description of the modification	contributions
Draft 0.1	15/07/2010	First Draft	P. Le Sidaner
0.11	11/01/10	Evolution	P. Le Sidaner
0.12	28/11/2011	Description of parameters	B. Cecconi, S. Erard, P. Le Sidaner
0.13	12/12/2011	Modification on many parameters	All VOPDC planetology
0.14	4/01/2012	Correction from Baptiste	B. Cecconi
0.15	7/1/2012	Corrections from Stéphane, integration of comments from recent meetings	S. Erard
0.16	20/01/2012	Merging all comments (SE, BC, MG)	P. Le Sidaner
0.17	9/02/2012	General update	S. Erard, P. Le Sidaner
0.18	13/02/2012	Reworked Section 2 + Added Appendices + reformatting + complete check & new comments throughout	S. Erard
0.19	13/02/2012	Comparison with ObsCore doc, extra comments	S. Erard
0.20	17/02/2012	Integrated comments	SE, JB, JCM, BC
0.21	25/02/2012	Further comments included Previous Appendix B => separate doc Completed service output, added UCDs Developed section 2	SE, PLS, BC
0.22	27/02	Evolution on time, spectral and spatial axis change idis (tap & core) to epn for Europlanet	BC, N. Andre, N. Bourel, C. Jacquey, M. Gangloff, PLS
0.23	22-27/03	Improvements, nasty details + consistency check	PLS, SE
0.24	6/06	Add 3 new parameters (angles)	SE, PLS
0.25	9/06	Better definition of “best practice”, introduction of Species	SE, PLS
0.26	11/06	Clarified calibration levels, answered some questions. Added element_name, Time_origin (tentative)	SE, PLS

Reference documents

[RD1] Planetary data access protocol (PDAP). IPDA draft 1.0 (latest to date, Nov. 2011)
http://planetarydata.org/projects/inactive-projects/data-access/documents/pdap-versions/pdap-v1.0-09-11-2011/view

[RD2] Keywords
RFC 2119: Key words for use in RFCs to Indicate Requirement Levels, S. Bradner, ed. IETF (Internet Engineering Task Force), March 1997. Available at http://www.rfc-editor.org/rfc/rfc2119.txt

[RD3] Virtual Observatory Support Interface (VOSI)
http://ivoa.net/Documents/VOSI/20101206/index.html

[RD4] The Committee on Small Body Nomenclature handles Minor Planet Names and Designations, Comet Names and Designations, Cross Listed Objects:
http://www.ss.astro.umd.edu/IAU/csbn/
In addition, the IAU Working Group for Planetary System Nomenclature (WGPSN) defines feature names on planetary surfaces (which are not necessarily relevant for EPN-TAP):
http://planetarynames.wr.usgs.gov/

[RD5] EPN data model version 1.18a (last version to date) can be found here:
http://www.europlanet-idis.fi/documents/public_documents/Data_Model_v1.18a.pdf

[RD6] TAP protocol
http://ivoa.net/Documents/TAP/

[RD7] ObsTAP and ObsCore
http://ivoa.net/Documents/ObsCore/

[RD8] UCD UType concept
http://ivoa.net/Documents/cover/UCDlist-20070402.html

[RD9] IAU Working Group on Cartographic Coordinates and Rotation Elements of the Planets and Satellites
http://astrogeology.usgs.gov/Page/groups/name/IAU-WGCCRE

[RD10] IVOA Astronomical Data Query Language Version 2.00 http://ivoa.net/Documents/latest/ADQL.html

[RD11] Name resolver returning body official names and astronomical coordinates at a specific time:
http://vodev.imcce.fr/webservices/ssodnet/?resolver

[RD12] IAU nomenclature for object types:
http://planetarynames.wr.usgs.gov/Page/Planets

[RD13] Space time and coordinate in IVOA
http://ivoa.net/Documents/latest/STC.html

[RD14] IVOA Registry interface
http://ivoa.net/Documents/RegistryInterface/

[RD15] Unit in the IVOA (current draft)
http://ivoa.net/Documents/VOUnits/

[RD16] EPN-RI Interoperable Data Access / Data Organization conventions, V0.0a1

[RD17] EPN-RI Coordinate systems, V0.1

[RD18] EPN-TAP documentation:
http://voparis-europlanet.obspm.fr/xml/TAPCore/

Acronym list

EPNCore	Set of core parameters from EPN-DM, mandatory for EPN-TAP compatibility
EPN-TAP	Specific protocol to access Planetary Science data in Europlanet-VO
EPN-DM	Specific Data Model to describe Planetary Science data in Europlanet-VO
epn_core	Name of table / view of a database which contains the EPN-TAP parameters. Required for EPN-TAP compatibility.
IVOA	International Virtual Observatory Alliance
IPDA	International Planetary Data Alliance
PDAP	(Planetary Data Access Protocol) Protocol to access planetary data space archives, developed and maintained by IPDA.
TAP	(Table Access Protocol) One of the protocols developed by the IVOA to access astronomical data.
ObsTAP	TAP protocol applied to the Observation Data Model of IVOA
ObsCore	set of core parameters from the Observation Data Model of IVOA
ADQL	(Astronomical Data Query Language)
UCD	(Unified Content Descriptor) Define measured physical quantities in the IVOA.
Utype	Description of data properties, in relation with a Data Model.

Abstract

The goal of this document is to describe a protocol to access and retrieve planetary data.
This protocol will allow selecting a subset of data from an archive in a standard way. This document describes the requirements to allow data providers and users to implement an EPN-TAP compliant service, based on the IVOA (International Virtual Observatory Alliance) TAP (Table Access Protocol) specifications [RD6] and the EuroPlaNet Data Model [RD5].

Status of this document

This document is a draft, and must be approved by IDIS partners.

Acknowledgements

This work has been conducted in the frame of Europlanet-RI JRA4 work package.
The EuroPlaNet-RI project is funded by the European Commission under the 7th Framework Program, grant #228319 "Capacities Specific Programme".

Document change record 2
Reference documents 3
Acronym list 5
Abstract 6
Status of this document 6
Acknowledgements 6
1 - Introduction *9*
2 - Main concepts of EPN-TAP *9*
3 - Requirements for compliance *11*
4 - EPN-TAP queries *12*
4.1 Service Behavior 13
4.2 Compulsory parameters 14
4.2.1. Resource Type 14
4.2.2 Data Product Type 15
4.2.3 Target Name 16
4.2.4 Target Class 16
4.2.5 Time min/max 17
4.2.6 Time scale 18
4.2.7 Time sampling step min/max 18
4.2.8 Exposure time min/max 19
4.2.9 Spectral range min/max 19
4.2.10 Spectral sampling step 20
4.2.11 Spectral resolution 20
4.2.12 Spatial range (c1,c2,c3) 20
4.2.13 Spatial resolution 21
4.2.14 Spatial frame type 21
4.2.15 Incidence angle min/max 22
4.2.16 Emergence angle min/max 22
4.2.17 Phase angle min/max 23
4.2.18 Instrument host name 23
4.2.19 Instrument name 24
4.2.20 Measurement type 24
4.3 Optional parameters 24
4.3.1 Species 25
4.3.2 Element_name 25
4.4 Application to special datasets 25
5 - Service response *26*
5.1 - Query response Metadata 27
5.1.1 Data information 27
5.1.1.a publisher 27
5.1.1.b reference 28
5.1.1.c title 28
5.1.1.d collection_id 28
5.1.1e processing level 28
5.1.2 Access Reference (access_url) 29
5.1.3 Access Format (access_format) 29
5.1.4 Estimated Size (access_estsize) 29

5.1.5 Description of data unit and dimension 29
5.1.5a Units.expression 30
5.1.5b Units.scaleSI 30
5.1.5c Units.dimEquation 30
5.1.6 Region of interest 30
5.1.7 Description of coordinate frame 31
5.1.8 Description of time frame 31
5.2 - Data response 31

AppendixA: Unit conversions 32
A.1 Spectral axes 32
A.2 Spatial axes 33
A.3 Time axes 34

AppendixB: Use cases 35
B.1 Tabular 35
B.2 Several related tables 35
B.3 Files 36
B.4 Computational 37

1 - Introduction

The EPN-TAP protocol is directly derived from IVOA's TAP [RD6], a simple protocol to access data organized in tables, here adapted for Planetary Science data.
EPN-TAP is an extension of IVOA TAP (called ObsTAP) on the same level as ObsCore (IVOA's Observation Data Model).
In this document, the EPN Data Model is used to describe many types of Planetary Science data using a standard terminology. EPN-TAP uses a subset of this terminology to define standard query parameters.
EPN-TAP uses IVOA TAP associated with a subset of the EPN Data Model [RD5] called EPNCore. Some parameters are directly derived from the PDAP protocol from IPDA [RD1].

The description of EPN-TAP compliant services must follow the rules defined by IVOA VOSI [RD3], which defines the capabilities of the services, as well as service availability.
Because EPN-TAP is IVOA-TAP compliant, the discovery of all EPN-TAP services can be performed using an IVOA registry. A specific extension of IVOA registries to describe EPN-TAP services in a more accurate way still has to be defined.

Practical use cases are listed in Appendix B to support reflection about EPN-TAP.

2 - Main concepts of EPN-TAP

TAP is a protocol dedicated to access relational database tables. It uses ADQL (the Astronomical Data Query Language, [RD10]) to query the databases.
To allow similar queries on all EPN-TAP services, we will consider that the EPNCore data model is implemented in the database as a view. In order to be accessed through EPN-TAP, all databases must therefore include a view called epn_core, containing all the parameters described in section 4.2. This view/table is used as a catalog (or index) of the accessible data in the database.

In this system, the user writes his query on the client interface. The client pre-processes this query and sends it to the server. The server in turn looks for matches in the database catalogue and sends back an answer. This process is illustrated in Fig. 1.

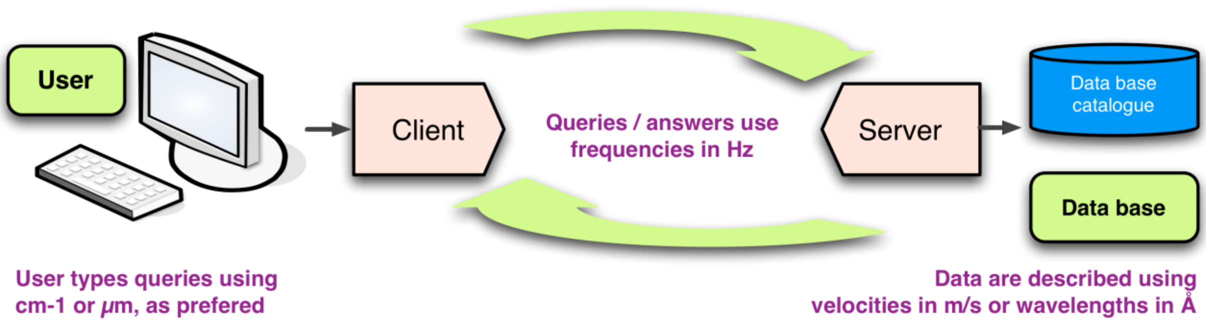

Fig. 1: Client/ server general scheme for EPN-TAP

In order to handle the multiplicity of situations, several parameters are normalized in the protocol regardless of the content of the databases. For instance, a spectroscopy database may provide measurements on a wavelength scale in microns, while the user wants to query the data in nm. In EPN-TAP queries, spectral axes are always described on a frequency scale in Hz. However, the client interface may propose a variety of units to the user, and translate them in Hz to write the query; similarly, the server will perform a transform to access the data in native form. It is therefore essential that such transforms are exactly reciprocal on both sides of the query system (see Appendix A). A similar situation occurs e.g. for time scales, which the protocol addresses in Julian Days.

The EPN-TAP protocol is closely related to the TAP protocol, and mainly differs by the definition of its core parameters. For efficiency reasons, the server side is expected to only implement TAP, while the client performs most EPN-specific operations and turn them into fully TAP-compliant queries, which can be handled directly by the database through ADQL. In addition, the epn_core view of the database must be formatted in order to accept queries on EPN core parameters (Fig. 2). In the previous example, if the spectral axis is provided in wavelengths in the database, the epn_core view must include a transformed version in frequency. Such transforms must be symmetric from the ones performed by the client, and must follow the rules provided in Appendix A.

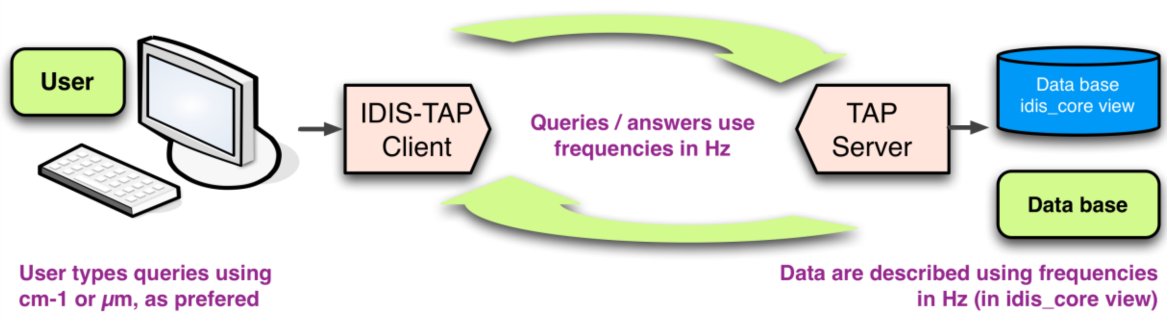

Fig. 2: Practical implementation for EPN-TAP services

Similarly, the parameter names are mostly used as tags to pass the values between the client and the server. Since they are used to handle a variety of situations, science fields... they may not reflect the exact meaning of the parameters in the frame of a specific database. This again is not an issue, since these parameter names are not normally seen by the user.

The server looks for lines of the epn_core table matching the query (Fig. 3). The answer is therefore an excerpt of the table containing the corresponding EPNCore parameters and the data, embedded in a VOTable. If the data area is simple, it can be included directly in the returned VOTable. Whenever its structure is complex, it must be described for further processing, e.g., by plotting tools. In particular the main product must be clearly identified.

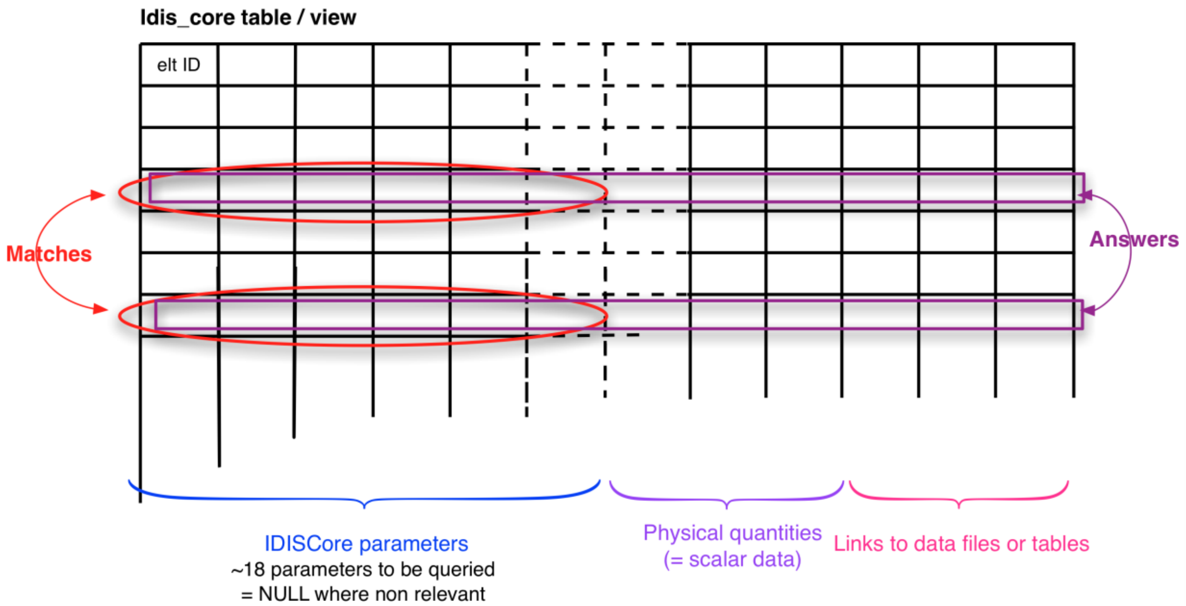

Fig. 3: Query of the epn_core table and returned values

A particular situation occurs with the spatial coordinates, because of the extreme diversity encountered in Planetary Science. In order to be able to simply formulate a query, the general type of coordinate system (e.g. celestial coordinates, geographical coordinates, Cartesian coordinates in a volume…) must be known in advance. For this reason the description must be included in the column description of the TAP response [RD6] and in the metadata returned by the service.

"Capabilities" of TAP services are accessible as described in VOSI. The capability query may include basic references to the service, including credits.

Non-mandatory columns of the epn_core table can also be queried through EPN-TAP. This is done along the lines defined in the VOSI protocol. This mechanism provides a complete access to the data service (in contrast for example to the PDAP protocol).

Although the epn_core table contains a list of the "granules" (typically the data files) available in the service, datasets can also be defined inside the epn_core table. Such datasets may consist in subsets of granules or physical quantities, selected according to various criteria by the data provider. Therefore, a complex PDS data set for example can be sliced into several pieces accessed independently though EPN-TAP. This will allow data providers to make their data available in EPN-TAP without going through the burden of generating alternative versions of their databases.

3 - Requirements for compliance

The compliance of the service will address alls steps. We will use the standard keywords of the W3C "MUST", "REQUIRED", "SHOULD", and "MAY" to describe the requirements [RD2].

A compliant TAP service has to verify all the MUST and REQUIRED statements. As far as possible SHOULD statements will be reached and MAY statements can be considered as best practice.
"Best practice" is also mentioned on several occasions as an option, e.g. concerning target names. This implies that the data service provider can use another naming scheme if he wishes to do so. It is stressed however that the capacity to identify relevant data and to make the services interoperable relies on the use of standard conventions by the data providers. In this example, a data service that does not use the standard IAU target names will not respond correctly to general EPN queries and may therefore not be visible to the user. Although not mandatory, the "best practice" is therefore strongly recommended in any case.

4 - EPN-TAP queries

EPN-TAP is an extension of IVOA TAP and is compliant with the TAP standard. It typically contains:

- TAP using ADQL [RD10].
- VOSI as capability and metadata [RD3]
- TAP queries can be synchronous or asynchronous. For specification of asynchronous mode one can refer to the TAP document [RD6].

In addition, it also contains some extensions (like ObsCore does):

- A table called epn_core must be present and must contain all the EPNCore parameters
- There is one reserved value, which must be accepted for all parameters: NULL
- If a parameter is set to NULL, left empty, or omitted in the query, the server will ignore it (i.e., data are not filtered using this parameter). For example, a query using a single parameter resource_type = granule will reply with the complete list of granules / data files in the service. However, the EPN-TAP client may set a default value for some parameters, in particular for resource_type and time_scale (see 4.2).

The TAP query is very generic and there is no mandatory parameter associated with it. EPNCore defines a subset of parameters that must be handled by a compliant service even if they are not relevant for this service. In this case, the corresponding fields in the epn_core table should be set to "empty" or NULL. If the client sends a query using such a non-relevant parameter, the service must answer with no data (i.e., elements containing NULL for the query parameter will not be included in the answer).

The client must use the HTTP GET or POST protocols to send queries to services. The query is composed of the URL of the service, and ADQL language [RD10] is used to express the request.

Example query:
*http://<server address>/tap/sync/request=doquery & lang=adql & query=select * from epn_core where t_min between '2455197.5' and '2455927.5' and target_class = 'comet' and target_name = 'halley'*

The service will return all data of any type for comet Halley from 2455197.5 (010/01/2010) to 2455927.5 (01/01/2012) in Julian days

All parameters are case insensitive. In practice, applications and clients are encouraged to present parameters using only lower-case characters. We will follow this rule in the current document.

EPN-TAP services may also contain parameters not included in epn_core. All such parameters can be accessed providing the corresponding table name.

All information of the service table (as well as availability) is accessible using VOSI [RD3]. All the metadata tables related to a data service can be obtained via:
HTTP GET http://<server address>/tap/tables

VOSI also provides information about general service capabilities (e.g., IVOA protocols supported by the service).
The capabilities can be obtained via:
HTTP GET http://<server address>/tap/capabilities

Finally, "Availability" gives information on the current status (up/down...).
HTTP GET http://<server address>/tap/availability
See VOSI document for details on the availability resource.

4.1 Service Behavior

Parameters are "multivalued" in the sense that a single query can introduce multiple values for a given parameter.
ADQL provides standard operations on parameters to combine possible conditions (and, or, like...), which are all implemented as well as parentheses. Standard ADQL wildcards are also implemented [RD10].

Example:
target_name = 'mars' or target_name = 'venus '

As explained above, if a parameter is set to NULL or omitted in the query, it will not be used to filter the data. In some cases, the EPN-TAP may set a default value to some parameters, as indicated below.

Example
*http://<server address>/tap/sync/request=doquery & lang=adql & query=select * from epn_core where t_min > '2455197.5' and t_max < '2455927.5'*

Will return all kind of data from 2455197.5 (010/01/2010) to 2455927.5 (01/01/2012) in Julian days (target is not specified).

4.2 Compulsory parameters

Similarly to IVOA's ObsCore [RD7], EPNCore defines a set of parameters describing the EPN-TAP services.
These parameters must all be present in the "epn_core" table of the database, and must be understood as they are defined below. All EPNCore parameters are listed and described in this section.
Parameters are characterized using their UCDs and Utypes [RD8].
Other parameters present in the data services may be queried by EPN-TAP through similar mechanisms, but are not systematically supported.

4.2.1. Resource Type

Name in epn-core table : resource_type
Type : string
unit : unitless
Utype : Epn.ResourceType
UCD : meta.id;class

Defines the scope of the query. There are two possible values:
dataset and **granule**

The epn_core table lines mostly correspond to individual data objects, and point to a single file (or group of files); this corresponds to the "granule" level of information, typically the smallest level described in the catalogue and distributed by the data service.
Subsets of granules may also be grouped together and summarized on a single line of the epn_core table, which defines a "dataset". Therefore the same EPN service may contain several datasets.
Responses to queries related to granules or datasets are formally similar.

A granule is the smallest element reachable in a data service: either a file, a group of associated files, a table entry, or some kind of data computed on the fly. This typically corresponds to a line in the epn_core table.

The dataset field definitions are derived from the component granules in the most straightforward way (e.g., dataset time_min is the minimum time_min of included granules...).
The answer to a query using resource_type=dataset may include a URL access to a full dataset archive in one file. Providers could omit this link, for instance in case of very large datasets they don't intend to distribute globally.
Providers may include extra fields in the epn_core table containing the ID of the datasets comprising the granule.

In many cases, the EPN-TAP client will set a default to resource_type = dataset
This will allow test queries to return information on datasets only, therefore limiting the number of answers.

4.2.2 Data Product Type

Name in idis-core table: dataproduct_type
Type : string
unit : unitless
Utype : Epn.dataProductType
UCD : meta.id;class

The data product type describes the high level scientific organization of the data product being considered. The EPN DataModel currently defines the following types:

* **image**: associated scalar fields with two spatial axes, e.g., images with multiple color planes, from multichannel cameras for example. Maps of planetary surfaces are considered as images.

* **spectrum**: data product which spectral coverage is the primary attribute, e.g., a set of spectra.

* **dynamic_spectrum**: consecutive spectral measurements through time, organized as a time series.

* **spectral_cube**: sets of spectral measurements with 1 or 2D spatial coverage, e.g., imaging spectroscopy. The choice between Image and spectral_cube is related to the characteristics of the instrument.

* **profile**: scalar or vectorial measurements along 1 spatial dimension, e.g., atmospheric profiles, atmospheric paths, sub-surface profiles…

* **volume**: other measurements with 3 spatial dimensions, e.g., internal or atmospheric structures.

* **movie**: sets of chronological 2D spatial measurements

* **cube**: multidimensional data with 3 or more axes, e.g., all that is not described by other 3D data types such as spectral cubes or volume.

* **time_series**: measurements organized primarily as a function of time (with exception of dynamical spectra). A Spacecraft dust detector measurement is a typical example of a time series.

* **catalog**: can be a list of events, a catalog of object parameters, a list of features.... It can be limited to scalar quantities, and possibly limited to a single element. E.g., a list of asteroid properties. Time_series, Profile, and Catalog are essentially tables of scalar values. In Time_series the primary key is time; in Profile it is altitude or distance; in Catalog, it may be a qualitative parameter (name, ID…).

* **spatial_vector**: list of summit coordinates defining a vector, e.g., vector information from a GIS, spatial footprints…

Usage:
*select * from epn_core where resource_type = 'granule' and dataproduct_type='image'*

will return only image data.

4.2.3 Target Name

Name in epn-core table: target_name
Type : string
unit : unitless
Utype : Epn.TargetName
UCD : meta.id;src

The target_name element 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.
The best practice is to use the official name of the target as defined by IAU [RD4].
It is stressed however that the IAU nomenclature for planetary bodies is only recommended, but not necessarily used by all services. Users should be aware that some services containing data of interest may not be easily visible; data providers should realize that services that do not use the IAU names may not be interoperable. To help data providers in this task, a name resolver is provided by VOParis-IMCCE to handle multiple denominations [RD11].

The Exoplanet Encyclopedia provides a complete list of currently known extrasolar planets:
http://exoplanet.eu/index.php

Meteorite catalogs can be found here:
http://www.nhm.ac.uk/research-curation/research/projects/metcat/search/indexsing.dsml
http://www.lpi.usra.edu/meteor/index.php

The catalog of Lunar sample is available here:
http://www.lpi.usra.edu/lunar/samples/

Other planetary samples are listed in topical web sites, e.g. Stardust samples are described here:
http://curator.jsc.nasa.gov/stardust/catalog/

Usage:
*select * from epn_core where target_name='ceres' or target_name='vesta' and target_type='Dwarf_Planet' or target_class ='Asteroid'*

returns only data from 1 Ceres or 4 Vesta (see ADQL syntax). Complex queries may also include parentheses.

4.2.4 Target Class

Name in epn-core table: target_class

Type : enum string
unit : unitless
Utype : Epn.TargetClass
UCD : src.class

The target_class element identifies the type of a named target. A target is defined without ambiguity by a couple of parameters: target_class and target_name (although some targets may have no proper name).

The EPN DataModel defines the possible values for target_class:
asteroid, dwarf_planet, planet, satellite
(types from IAU list [RD12])
comet, exoplanet, interplanetary_medium, ring, sample, sky, spacecraft, spacejunk, star.
(extra types defined for EPN)

Usage:
Any target has a unique target type.
"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 - other cases are handled though spacecraft or spacejunk.
"sky" may be used for other celestial bodies, usually referred to by their sky coordinates. Also includes Interstellar Medium.
"star" is used typically for calibration targets, and for the Sun.

4.2.5 Time min/ max

Name in epn-core table: t_min, t_max
Type : double
unit : 'd' — Julian day
t_min
Utype : Char.TimeAxis.Coverage.Bounds.Limits.Interval.StartTime
UCD : time.start;obs
t_max
Utype : Char.TimeAxis.Coverage.Bounds.Limits.Interval.StopTime
UCD : time.end;obs

The time parameter(s) provides the date and time of acquisition. It may be a couple of parameters to handle long periods in some datasets.
This parameter is provided in Julian days, expressed as a double precision float. Although ObsCore uses Modified JD, EPNCore uses JD to avoid ambiguity with time origin. With double precision floats, the accuracy is on the order of 1 ms.
Whenever time is defined by a single parameter, both t_min and t_max contain the same value in the database.

Examples:
*http://<server address>/tap/sync/request=doquery & lang=adql & query=select * from epn_core where t_min > '2455197.5' and t_max < '2455927.5'*

will search data described by a time range.

*http://<server address>/tap/sync/request=doquery & lang=adql & query=select * from epn_core where t_min between '2455197.5' and '2455927.5'*

will search data described by a start time parameter.

4.2.6 Time scale

Name in epn-core table : time_scale
Type : string
unit : 'unitless'
Utype : stc.timeScaleType
ucd : time.scale

This parameter defines which time scale is used to express the time range in the query (TBC), and which time scale is to be used in the answer (for instance when querying an ephemeris server). If not provided, default should be UTC – this may be set by the client.

Other time scales can be used, as defined in STC [RD13]:
- TT : Terrestrial Time: the basis for ephemeris
- TDB : Barycentric Dynamic Time: the independent variable in planetary ephemeris; time at the Solar System barycenter, synchronous with TT on an annual basis; sometimes called TEB
- TCG : Terrestrial Coordinate Time
- TCB : Barycentric Coordinate Time; runs slower than TDB but is consistent with physical constants
- TAI : International Atomic Time; runs 32.184 s behind TT
- UTC : Coordinated Universal Time; currently (2006) runs 33 leap seconds behind TAI
- GPS : Global Positioning System's time scale; runs 19 s behind TAI, 51.184 s behind TT

This parameter is used in queries to define the time scale used by computational services such as ephemeris tools.
Service providers may want to use non-compulsory parameters to accommodate specially formatted time scales (such as native format from on-board clocks), see section 5.
The time provided is not necessarily measured on ground, and therefore may need to be corrected for light path. The location where time is measured is provided though the Time_Origin output parameter (see section 5).

*SE : removed Local and related comments – now uses Time_Origin output parameter, TBC.

4.2.7 Time sampling step min/max

Name in epn-core table : t_sampling_step_min, t_sampling_step_max

Type : double
unit : 's'
t_sampling_step_min
Utype :
UCD : time.interval;stat.min
t_sampling_step_max
Utype : ??
UCD : time.interval;stat.max

This parameter provides the sampling time for measurements of dynamical phenomena, and for computations. This is the time between 2 successive measurements or data. This may be a query parameter e.g. for ephemeris computations.
Time “resolution” may be derived from this quantity, if measurements are regularly spaced — however, this depends on the data type.

4.2.8 Exposure time min/ max

Name in epn-core table: t_exp_min, t_exp_max
Type : double
unit : 's'
t_exp_min
Utype : ?
UCD : time.duration;obs.exposure;stat.min
t_exp_max
Utype : ?
UCD : time.duration;obs.exposure;stat.max

This parameter corresponds to the integration time of measurements. This time is usually shorter than the time_sampling_step if both are present.

4.2.9 Spectral range min/ max

Name in epn-core table: spectral_range_min, spectral_range_max
Type : double
unit : 'Hz'
spectral_range_min
Utype : ?
UCD in case of electromagnetic radiation em.freq;stat.min
UCD in case of spectral energy from particle phys.energy;phys.part;stat.min
spectral_range_max
Utype : ?
UCD in case of electromagnetic radiation em.freq;stat.max
UCD in case of spectral energy from particle phys.energy;phys.part;stat.max
en y réfléchisant, je vois pas comment on peux mettre un Utype à ce paramètre qui peux représenter deux grandeurs physiques différentes suivant le contexte (une onde EM ou une particule pour les plasma)
demander à ce que soit crée phys.part au niv des UCD

The spectral_range defines the upper and lower bounds of the spectral domain of the data. As mentioned previously, this quantity is is expressed on a frequency scale in Hertz.

Conversions to the native unit are provided in Appendix A.

The spectral scale applies to both electromagnetic waves and particle detections. For this reason, two UCDs may be associated to this parameter.

4.2.10 Spectral sampling step

Name in epn-core table: spectral_sampling_step_min, spectral_sampling_step_max
Type : double
unit : 'Hz'
spectral_sampling_step_min
Utype : : ?
UCD : spect;stat.min
spectral_sampling_step_max
Utype : : ?
UCD : spect;stat.max

The spectral_sampling_step is the spectral separation between the centers of two adjacent filters or channels. Like all spectral quantities, it is expressed on a frequency scale in Hz. Conversions to the native unit are provided in Appendix A.
This parameter is mostly intended to provide an order of magnitude, e.g., to distinguish between grating spectrometers and Fourier spectrometers, or between observations related to surfaces or atmospheres. In can also help distinguishing between Nyqvist and sub-Nyqvist sampling rate.

4.2.11 Spectral resolution

Name in epn-core table : spectral_resolution_min, spectral_resolution_max
Type : double
unit : 'Hz'
spectral_resolution_min
Utype :
UCD : spec.resolution;stat.min
spectral_resolution_max
Utype :
UCD : spec.resolution;stat.max

The spectral_resolution corresponds to the spectral bandwidth used for the measurement (Full Width at Half Maximum). In case of a filter camera this is the filter bandwidth; in case of a spectrometer this is identical to the spectral resolution.
This parameter is mostly intended to provide an order of magnitude, e.g. to distinguish between grating spectrometers and filter cameras.

4.2.12 Spatial range (c1,c2,c3)

Name in epn-core table : c1min, c2min, c3min, c1max, c2max, c3max
Type : vector of double
unit : depending on context (deg, m)

Utype : char.SpatialAxis.Coverage
c1min, c2min, c3min
UCD pos;stat.min ou obs.field;stat.min
c1max, c2max, c3max
UCD pos;stat.max ou obs.field;stat.max

This parameter provides up to three spatial coordinates, in a form depending on the spatial frame type. All services should handle three dimensions, even if the third one is set to NULL.

The query will be made on the coordinate system proposed by the provider. More precise description of the coordinate system is given in the response metadata. Descriptions for Europlanet are provided in [RD17].

4.2.13 Spatial resolution

Name in epn-core table: spatial_resolution
c1_resol_min, c2_resol_min, c3_resol_min, c1_resol_max, c2_resol_max, c3_resol_max
Type : double
unit : depending on context (deg, m, ...) — same as spatial_range
c1_resol_min, c2_resol_min, c3_resol_min
Utype :
UCD : depending on context phys.size;stat.min ou pos.angResolution;stat.min
c1_resol_max, c2_resol_max, c3_resol_max
Utype :
UCD : depending on context phys.size;stat.min ou pos.angResolution;stat.max

This parameter provides a simple estimate of resolution, either the FWHM of the PFS on the sky (in degrees), or the pixel size on a surface (in m), depending on spatial_frame_type.

The client frontend may propose more appropriate units to the user, depending on context (e.g., angular resolution in mas, distance in m…).

4.2.14 Spatial frame type

Name in epn-core table: spatial_frame_type
Type : string
unit : unitless
UCD : meta.id;class

Provides the "flavor" of the coordinate system, which defines the nature of the coordinates. This must be known to write a query on spatial coordinates. The possible types are described below:

celestial (2D angles on the sky, e.g. right ascension c1 and declination c2 + possibly distance from origin c3 – this is a special case of spherical frame),
body (2D angles on a rotating body, e.g. longitude c1 (and latitude c2 + possibly a z c3 coordinate)
longitudes always increase eastward,
the Z coordinate can be the distance counted either from the reference surface or from

the body center, TBC (frames are related to body centers, but a surface reference is required for atmospheric structure)

Cartesian (x,y,z) as (c1,c2,c3)

cylindrical (r, theta, z) as (c1,c2,c3) angles are defined in degrees

spherical (r, theta, phi) as (c1,c2,c3) angles are defined in degrees as in usual spherical systems (E longitude, zenithal angle/colatitude). If related to the sky, "celestial" coordinates with RA/Dec must be used.

This parameter, although related to the specific coordinate system in use, is only intended to identify the nature of the coordinates handled by the server (e.g., angles versus distances). Services must declare the spatial_frame_type they handle in answer to an initial, general query. This will then allow the client to provide it with relevant queries on the data. The actual coordinate system is also indicated in the return parameters.

4.2.15 Incidence angle min/max

Name in epn-core table: incidence_min, incidence_max
Type : double
unit : 'degree'
incidence_min
Utype : ?
UCD pos.incidenceAng.min
incidence_max
Utype : ?
UCD pos.incidenceAng.max

The incidence parameters define the upper and lower bounds of the incidence angle variation in the data (also known as Solar Zenithal Angle in some fields). This is always indicated in decimal degrees.
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 keywords).

4.2.16 Emergence angle min/max

Name in epn-core table: emergence_min, emergence_max
Type : double
unit : 'degree'
emergence_min
Utype : ?
UCD pos.emergenceAng.min
emergence_max
Utype : ?
UCD pos.emergenceAng.max

The emergence parameters define the upper and lower bounds of the emergence angle variation in the data. This is always indicated in decimal degrees.
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 keywords).

4.2.17 Phase angle min/max

Name in epn-core table: phase_min, phase_max
Type : double
unit : 'degree'
phase_min
Utype : ?
UCD pos.phaseAng.min
phase_max
Utype : ?
UCD pos.phaseAng.max

The phase parameters define the upper and lower bounds of the phase angle variation in the data. This is always indicated in decimal degrees (+ define convention for negative values?).
Phase, incidence and emergence are partly related:

$$abs(i - e) < \varphi < i + e$$

If the azimuth angle α is provided instead of the phase angle, the latter can be derived from knowledge of the three angles:

$$\cos \varphi = \cos i \cos e + \cos \alpha \sin i \sin e$$

4.2.18 Instrument host name

Name in epn-core table: instrument_host_name
Type : string
unit : unitless
Utype : Provenance.ObsConfig.Facility.name
UCD : meta.class

This parameter provides the name of the observatory or spacecraft that performed the measurements. The best practice is to use names from the lists indicated below. A list of host names should be provided for integrated data sets.

For ground-based observations, the reference is the list of IAU observatory codes:
http://www.minorplanetcenter.net/iau/lists/ObsCodesF.html

Concerning space-borne data, the most complete list of international planetary missions and orbital observatories is found here:
http://nssdc.gsfc.nasa.gov/nmc/
Planetary missions in particular are 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

(TBC – Spice is unambiguous but only uses IDs, PDS values are explicit but somewhat arbitrary)

4.2.19 Instrument name

Name in epn-core table: instrument_name
Type : string
unit : unitless
UTYPE : Provenance.ObsConfig.Instrument.name
UCD meta.id;instr

Identifies the instrument(s) that acquired the data. A list of instruments should be provided for integrated data sets.

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.

Concerning space-borne data, the most complete 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

4.2.20 Measurement type

Name in epn-core table: measurement_type
Type : string
unit : unitless
Utype : Char.ObservableAxis.ucd
UCD : meta.ucd

The measurement_type parameter defines the physical quantities contained in the data, using a UCD.
The list of UCD is related to all quantities included in the service, not only those used for the data listed in the epn_core table (e.g., this also includes secondary data present in data files).
The provider should use the UCD1+ list from IVOA and should extend it only if necessary [RD8]:

http://www.ivoa.net/Documents/REC/UCD/UCDlist-20070402.pdf

Extra UCDs will be proposed/requested to IVOA. In the meantime, an extended list of UCDs will be made available in the EPN-DM document (in progress).

4.3 Optional parameters

EPN-TAP can query parameters not included in the EPN-Core. Some of these parameters are defined precisely but are relevant only to rather specific data services. Those are not considered mandatory and are not part of the EPN-core, but they should be implemented as defined here when present.

4.3.1 Species

Name in epn-core table: species
Type : string – standard formulas only
unit : unitless
Utype : ?
UCD : phys.composition.species (TBC)

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 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 the vertical abundances of many gaseous species with altitude. All columns are abundances and are described by the same measurement_type parameter. Only the use of the "species" parameter allows identifying the various species and accessing the requested information.

*SE: this one is used to introduce basic compositional information in the search parameters. But the perimeter is very restrained.

4.3.2 Element_name

Name in epn-core table: element_name (?)
Type : string – free format
unit : unitless
Utype : ?
UCD : (TBC)

The element_name parameter introduces a secondary target name to provide more details about the observation. It can for instance accommodate a crater or surface feature name, while target_name is always related to the whole body.

*SE: this one seems needed very often, name TBC.

4.4 Application to special datasets

EPN-TAP, because it is directly derived from TAP, may not be optimal to query computational services.
Simulated data are accessible the same way as observational data, but the simulation parameters may be difficult to access through this mechanism.
Similarly, experimental data are accessible but experimental setup and sample description may be hard to reach. VAMDC-TAP may be a better choice to handle this kind of data.

Access to various data structures can be assessed through the Use Cases listed in

Appendix B.

Tabular data are by construction easier to handle with EPN-TAP.
If the dataset mostly consists in files, EPN-TAP tells nothing about the file structure. As a consequence, the user cannot usually plot the files.

Answers to queries should also tell
- File format and how to read it
  => format and readerURL in the DM would do
- where to find physical quantities in the file (data of interest + axes) – this is not handled in VO plotting tools => the Virtis/Aladin demonstrator is a use case for this.
- What is the main data product if several are available? (e.g., data file in Virtis db, rather than geometry; column 3 instead of 5 in a table...)
  => some parameter requied to indicate the file (if several) + file area to be read (e.g. which data product in a PDS file, which extension/column in a FITS...)
- What is/are the main axis (generally time or coordinates I suppose)?
- Exact values of the query parameters from the epn_core table (since the query may include ranges)
- Any other matter?

The issue of format conversion is not addressed here. Concerning PDS data (which may be difficult to deal with), it may be convenient to read the data on the server and transfer them either in a VOTable or in a temporary FITS or ascii file. This kind of functionalities are server properties.

5 - Service response

Response of the service should comply with VOTable 1.2 or higher.
The VOTable must contain RESOURCE elements identified with the attribute type="results", containing a single TABLE element with the results of the query.
Additional RESOURCE elements may be present, but the usage of any such elements is not defined here and the TAP client will not use them.

The RESOURCE element must contain, before the TABLE element, an INFO element with attribute name = "QUERY_STATUS". The value attribute must contain one of the following values:
- "OK": the query executed successfully and a result table is included in the resource. This only means that the query worked successfully on the service, not that data are retrieved (i.e., nothing fulfills the query).
- "ERROR": an error was detected at the level of the TAP protocol, or the query failed to execute.

Another INFO element contains a Service Version attribute identifying the protocol version supported by the service.

The content of the INFO elements should be a message suitable for display to the user describing the status. See TAP document [RD6] for more details.

Example:

```
<?xml version="1.0"?>
<VOTABLE version="1.2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.ivoa.net/xml/VOTable/v1.2"
xsi:schemaLocation="http://www.ivoa.net/xml/VOTable/v1.2
http://www.ivoa.net/xml/VOTable/VOTable-1.2.xsd">
  <RESOURCE type="results">
   <INFO name="QUERY_STATUS" value="OK"> </INFO>
   <INFO name="SERVICE_PROTOCOL" value="1.0">EPN-TAP</INFO>
   <TABLE>
...
   </TABLE>
 </RESOURCE>
</VOTABLE>
```

If no result fulfills the query, the TABLE element must be present and empty (i.e., the table has no data tag).

5.1 - Query response Metadata

Some metadata are returned together with the data for user's information.

5.1.1 Data information

Information about the origin of data, credits, property… should come with the data themselves to trace and notify their origin.
To minimize the length of the response, only the publisher, the reference of publication (bibcode), the title, and the IVO dataset identifier will be returned. No parameter is compulsory in the query response.

The IVOA unique identifier will allow accessing all information from the registry (See IVOA registry interface [RD14]): publisher, owner, origin, credit….
Service providers should declare their services in the registry system used by EPN, similar to the IVOA one — details will be provided in a future document.

5.1.1.a publisher

Type : string
UCD : meta.ref

A short string identifying the publishing institute of the data (e.g., a data archive or data center) or an indexing service such as the ADS.

5.1.1.b reference

Type : string
UCD meta.bib

A bibliographic reference, which can be a URL or a Bibcode.

5.1.1.c title

Type : string
UCD : meta.ref

This parameter provides the title of the data service.

5.1.1.d collection_id

Type : string
UCD : meta.id

This identifier corresponds to the collection of data in the service.

5.1.1e processing level

EPN expression: processing_level
Type : integer
Utype PSR:processingLevel
UCD : meta.class.qual

In the framework of EPN-TAP, this parameter is intended to provide the user with a quick evaluation of data "usability". Several classifications are in use in different contexts, as summarized in the table below. EPN-TAP uses the CODMAC levels (IDs coded as integers). "Partially calibrated" data sets are in general considered as not calibrated, but this evaluation is up to the data provider, depending on context. Although the best practice is to separate calibration levels in different data services, several levels can be grouped and therefore indicated in the same answer.

(information from PSA + ObsCore doc)

CODMAC level	PSA level	NASA level	PRODUCT_TYPE (PDS/ PSA)	ObsTAP	Description (from PSA, with comments)
1 (raw)	1a		UDR	Level 0	Unprocessed Data Record (low level encoding, e.g. telemetry from a spacecraft instrument. Normally available only to the original team)
2 (edited)	1b	0	EDR	Level 1 (std data format)	Experiment Data Record (often referred to as "raw data": decommutated, but still affected by instrumental effects)
3 (calibrated)	2	1A	RDR	Level2	Reduced Data Record ("calibrated" in physical units)
4 (resampled)		1B	REFDR		Reformatted Data Record (mosaics or composite of several observing sessions, involving some level of data fusion)

5 (derived)	3	2-5	DDR	Level3	Derived Data Record (results of data analysis, directly usable by other communities with no further processing)
6 (ancillary)			ANCDR		Ancillary Data Record (extra data specifically supporting a data set, such as coordinates, geometry...)

5.1.2 Access Reference (access_url)

Utype : Obs.Access.Reference
UCD : meta.ref.url

This item contains a URL that can be used to download the data product (as a file of some sort).

5.1.3 Access Format (access_format)

Utype : Obs.Access.Format
UCD : meta.id;class

A non-exhaustive list of formats is proposed: VOTable, Fits, CSV, ASCII, PDS + standard image formats.
*** SE: the DM document include a “format” element to accommodate this information
It also includes a readerURL element to provide reference to adequate software.

5.1.4 Estimated Size (access_estsize)

Utype : Obs.Access.Size
UCD : phys.size;meta.file

The Access.Size field contains the approximate size (in kilobytes) of the file available via the corresponding URL. This provides an order of magnitude of the size of a data product before download, hence only an approximate value is required. It is only a useful indication that can help to tune download functionalities in an application according to high volumes of data and transfer bit rate.

5.1.5 Description of data unit and dimension

Service output values will be associated with physical units. The expression should be taken from IVOA recommendations (see IVOA document in progress [RD15]).
In addition, scaling factors and a dimensional equation could be necessary for direct representation.

We use here a description scheme used in the "Single Spectral Lines Data Model" of IVOA (section 2.1.2 of the v1.0 document). See also the VOUnit proposal (http://www.ivoa.net/internal/IVOA/UnitsDesc/WD-VOUnits-1.0-20111216.pdf).
It is described with 3 sub-entities defined as follows:

5.1.5a Units.expression

to be taken from IVOA document
UCD meta.unit

Example:
"Jansky" or "W.m^-2.Hz^-1"

5.1.5b Units.scaleSI

Scaling factor to convert the unit to its International System (SI) of Units equivalent.
Utype : ?

Examples:
- 1 cm has a scale factor of 1E-2, since 1cm = 0.01 m (its SI equivalent)
- 1 Jansky has a scale factor of 1E-26, since 1Jy = 1E-26 W.m^-2.Hz^-1

5.1.5c Units.dimEquation

Utype : ?
Dimensional equation representation of the unit. The format is a string with the dimensional equation, where M is mass, L is length, T is time, K is temperature and Q is electric charge.
(*SE: this should be a unit of electric current!!! Check if this is a bug in the IVOA!)
For ease of notation, the caret "^" indicating powers of ten can be removed - as is customary in Dimensional Analysis practices.

Example:
"ML-1T-3" and "ML^-1T^-3" are equivalent

(SE, to be studied latter) :
- how do we handle several parameters in a service? Or tables of parameters? Can this be only located in the service response?
- how do we document accuracy/noise level? Only in the data?

So I guess (BC OK)
1) units.exp, unit.scale, unit.dimEq must be provided for each data axis (in a table) / every quantity present (in file db)
2) in tabular data, noise/accuracy is another column – it needs to be documented.
How do we proceed with data in files?

5.1.6 Region of interest

Target_region
Type : string
unit : unitless
ucd : meta.main

This parameter briefly identifies the region of interest for the data set, in complement to target name – it is not mandatory. This parameter only introduces generic regions, not specific local names (see examples below).

The best practice is to take the values from standard sources:
- IAU thesaurus http://www.mso.anu.edu.au/library/thesaurus/
  + another version: http://www.vocabularyserver.com/trex/en/

The second one seems more recent and more complete (although the interface is not practical)
- Spase dictionary "http://www.spase-group.org/"

Example:
"Atmosphere", "Surface", "Ionosphere"

5.1.7 Description of coordinate frame

Spatial_coordinate_description
Spatial_origin

Provide description of the spatial frame(s) in use.
Possible values are detailed in [RD17], which is partly adapted from STC [RD13]

Examples (TBC)
BODY, Mars_IAU2000
ICRS, Geocenter

5.1.8 Description of time frame

Time_coordinate_description ???
Time_origin

Tells where the time is measured. This knowledge is required to cross-correlate event-based observations.
Possible values are:
Earth, (body names), (spacecraft names) — TBC

**SE : The STC apparently handles this using
- ReferencePosition = TOPOCENTER in TimeFrame
- a coordinate file in AstroCoords (FITS with ascii table in extension).

This may be OK, but we also need to provide the spacecraft name explicitly– hence the Time_origin parameter.

5.2 - Data response

Data response can be in an access URL (see associated parameter) or directly in the VOTable response. In this case, the fields corresponding to those parameters may have UCD, unit and a description associated with it.

(SE, to be studied latter) :
- Should include an indication of data structure (i.e. where to find values in a file…)

AppendixA: Unit conversions

Since these transforms are only used to convert database units to EPN-TAP conventional units, they don't need to be extremely accurate, but this accuracy would limit access to data natively provided in other units.
It is important that the conversions performed in the client are reciprocal to the ones used to build the epn_core view.

A.1 Spectral axes

The epn_core table must provide the spectral axes according to EPN-TAP standard (currently frequency in Hz).
A client should typically provide the possibility to enter either unit, and convert them to EPN-TAP standard.

Units are transformed assuming propagation in vacuum.

In addition to physical units, EPN-TAP should address instrumental units somehow (channel number, dimensionless)
+ Need to include separately Radio velocity and Optical velocity?

Symbol (units)	**Wavelength**	**Wavenumber***	**Frequency**	**Photon Energy****
	λ (nm)	**υ (cm-1)**	**ν (Hz)**	**Ep (eV)**
Conversion Factors	λ	$10^7/\lambda$	$3 \times 10^{17}/\lambda$	$1240/\lambda$
	$10^7/\upsilon$	υ	$3 \times 10^{10}\upsilon$	$1.24 \times 10^{-4}\upsilon$
	$3 \times 10^{17}/\nu$	$3.33 \times 10^{-11}\nu$	ν	$4.1 \times 10^{-15}\nu$
	1240/Ep	8056 x Ep	2.42×10^{14}Ep	Ep
Conversion Examples	200	5×10^4	1.5×10^{15}	6.20
	500	2×10^4	6×10^{14}	2.48
	1000	10^4	3×10^{14}	1.24

BC: as we propose to include particle spectra in that item, could we replace "Photon Energy" by "Energy" ?
SE : I guess for particles we consider kinetic energy, so the conversion won't stand (?)

Uses approximate conversion factors (except wavelength <=> wavenumber)
(as taken from here: http://www.newport.com/Optical-Radiation-Terminology-and-Units/381842/1033/content.aspx)

- Alternative:

taken from here:
http://www.photonics.byu.edu/fwnomograph.phtml

Equation: $f * \lambda = c$
Equation: $E = {}^{hc}/_{\lambda}$

where:
f = frequency in Hertz (Hz = ${}^{1}/_{sec}$)
λ = wavelength in meters (m)
E = energy in electron Volts (eV)
c = the speed of light (299792458 ${}^{m}/_{s}$)
h = Plank's constant (6.626068 × 10^{-34} ${}^{m2kg}/_{s}$)

- Many on-line converters are available to check the conversions, i.e:
http://heasarc.nasa.gov/cgi-bin/Tools/energyconv/energyConv.pl

Conversion of velocity and frequency

The exact relativistic equation for the conversion of the observed frequency, ν, into radial velocity, v, reads
$v = c\ (\nu_0^2 - \nu^2)\ /\ (\nu_0^2 + \nu^2)$
where c denotes the speed of light, and ν_0 is the rest frequency of the observed spectral line.
$v_{rad} = c\ [1 - (\nu\ /\ \nu_0)]$
The advantage of the »radio definition« is that equal increments in frequency correspond to equal increments in radial velocity. However, the »radio definition« has been deprecated by the International Astronomical Union (IAU) and should not be used anymore.

(hum... anyway, this requires additional mention of the reference frequency)

** com SE : beware that translating resolution and sampling step is not easy...we should also provide conversion procedures

A.2 Spatial axes

Native axes
Data are projected in a frame related to the instrument or acquisition process.
Typically: X/Y for a camera, X/time for an imaging spectrometer
Should correspond to (TBC):
Spatial_frame_type = Cartesian
Spatial_coordinate_description = native

Spatial frame type = celestial
In the epn_core table:
TBD

(JB) Coefficient for au to km conversion?

Spatial frame type = body
In the epn_core table:
- Longitudes are provided in East-handed convention (longitudes ranging from 0 to 360

degrees eastward).
- Latitudes follow IAU conventions:
North latitudes are positive.
North pole is located in the northern celestial hemisphere for planets and big satellites; all small bodies are defined to have direct rotation (i.e., North pole is in the southern celestial hemisphere if rotation is retrograde).

The transform from W to E convention reads like this (IDL syntax):

```
longE= -longW
ind=where(longE lt 0.)
if (ind(0) ge 0) then longE(ind)=longE(ind)+360
```

A client should typically provide the possibility to enter either E or W longitudes, and convert them to the East-handed EPN convention.

TBC for other systems

A.3 Time axes

In epn_core table:
Times are provided in seconds.
Dates are provided as Julian days — many conversion tools are available (beware that these must be double precision floats).

There must be some way to tell where dates are given (Earth, target, spacecraft...) i.e., if they have to be corrected for lighttime. For PDS data this is by construction s/c time.
Use a time_origin parameter ?

A client should typically accept either Julian days or calendar dates and convert them to Julian days.

If the time is not provided in a standard scale and cannot be converted easily into JD, the data service cannot use the main time axis. This concerns in particular non-calibrated data sets where time is provided in an arbitrary unit, e.g. as a spacecraft SCET. In such cases, the best solution is to use a secondary time axis stored in a service-specific parameter. This axis will not be available for cross searches anyway.

Secondary time scales can also appear together with the main JD time scale, e.g. to introduce seasonal information related to a planet, but not easily convertible (e.g. areocentric longitude of the Sun Ls, true anomaly...).

AppendixB: Use cases

This appendix contains a list of actual use cases to support the development of EPN-TAP.

B.1 Tabular

• A list of asteroid properties
Example: size and mass of large asteroids
This is a table with one line / entry. Each line contains all the metadata and data relevant to an object. Each line is for a different target.
Parameters include target name/class, no localization or time.
Data consist in several scalar quantities included in the table (several columns)
The granule is one line, the dataset is the table itself (+ references).

• A list of Martian craters
Example: List of lobate ejecta craters by F. Costard.
A table providing few data for each object, one line / entry but there is only one target as defined above (Mars).
Parameters are: longitude/latitude, no time.
Data are: ejecta extension, type, ID or crater name (several columns)
The granule is one line, the dataset is the table itself (+ references).

• Ulysses/URAP Thermal Noise Time Series (as proposed by CDPP)
(Example 9.3 of EPN-DM doc [RD5]).
Scalar measurements through time, averaged.
Data are 6 scalar quantities + three “support parameters” (ie reconstructed data): Sun distance + heliographic coordinates. Those are actually EPNcore query parameters.
Hence, the data set is a simple table providing 6 scalar quantities for each time step.

B.2 Several related tables

• Vertical atmospheric profiles
Example: Titan db at VOParis
A list of vertical profiles of P / T / mixing ratio of 10 main species as a function of altitude.
Metadata are location of profiles (longitude/latitude), time…
Data are contained in the individual profiles + perhaps a season parameter stored as an extra column in the list.
The granules are separated tables (or files) containing individual profiles.

• STEREO/waves Level 2 data (Example 9.1 of EPN-DM doc [RD5]) seems to fall here.
It would do if all daily files are indexed in a general catalogue. If they are concatenated, it would belong to B.1.

• Cassini/RPWS/HFR/SKR Dynamic Spectra
(Example 9.2 of EPN-DM doc [RD5]) also seem to fall in this category
The list is an ordered time table
Parameter is time only (?)
Data include 8 scalar values and 4 spectra for each time step
Dataset metadata include frequency table, etc.
Granules are described as time slots extracted from the index table... It seems that a more consistent approach would be to define the granule as a set of data for a given time step.

Such cases do not appear very different from a series of files (section B.3 below).

B.3 Files

• Simple imaging database
Example: BDIP at LESIA
A list of images defined by target, observation time... + specific parameters (phase angle, target size, orientation...)
Parameters are target, observation time, location on the sky, observatory...
Data are the images + the specific scalar parameters included in the list
The granules are the images, which may be available in several formats (ie: several versions of the images may be available).
Several coordinate systems are used: RA-DEC for planet position in the sky, planetary coordinates for sub-terrestrial and sub-solar points. The main coordinates are clearly those of the sub-terrestrial point, on which the requests will normally be performed. RA-DEC coordinates are only provided as supplementary parameters.

• Lab spectroscopy database
Example: MROCRISM spectral library (PDS)
A list of spectra of mineral defined by sample / mineralogical class / spectral range / origin
Parameters include spectral range / resolution (almost constant)... and do not provide adequate description of the data.
Data include the spectral files + detailed description of measurement and samples.
The granules are formatted files + possibly separated labels containing complete information of data.
If we can't use target / target class for a minimal description of the samples, it is impossible to query the data on relevant parameters! Even so, most parameters of interest are not accessible through EPN-TAP (i.e., particle size).
This is therefore a good example of why we need to be able to query non-EPN_core parameters!

• Observational db including support data

Example: Virtis/Venus-Express archive (PDS)
A list of observing sessions defined by time / location (longitude/latitude) / instrumental setup and observing conditions. Several instrumental channels are included. Several calibration levels are included.
Parameters are time / location / target / spectral range / integration time…
Data are raw + calibrated + geometry files + preview images + possibly derived products. All files have complex content (measurements + wavelength table, uncertainty…) and may vary in structure (internally described in PDS or FITS header).
Data also include "support parameters" which do not belong to the EPNCore: instrument channel, functioning mode…
The granules are calibrated data file + geometry file + preview (+ possibly raw/derived data file)
The EPNcore parameters may be sufficient to query the database efficiently, but the type of information retrieved must be described.

An alternative is to allow the provider to propose several data services from his dataset, therefore using several epn_core tables (and different service declarations): one addressing the M-channel, the other one the H-channel; one for raw data, the other one for calibrated data… At any rate, calibrated data and geometry files must be handled together.

B.4 Computational

• Ephemeris server
Example: Miriade service
A list of parameters computed on the fly as time series.
Virtual data set (only metadata are defined).
Parameters are time / time scale / target / time step…
The granules are tables of computed values