



International Planetary Data Alliance (IPDA)

Planetary Data Access Protocol (PDAP)

Version: V1.1 09.Sep.2011
Latest version: <http://www.planetarydata.org/>
Editor(s): Pedro Osuna, Jesús Salgado

Author(s): Pedro Osuna (ESA/PSA)
Jesús Salgado (ESA/PSA)
Steve Hughes (NASA/PDS)
Sean Kelly (NASA/PDS)
Yukio Yamamoto (JAXA)
[New authors as per NASA/Europlanet contributions to be agreed in next IPDA meeting]
and the IPDA technical expert group members

Abstract

This specification defines a protocol for the retrieval of distributed planetary data.

The interface is meant to be reasonably simple to implement by service providers. The basic summarized functionality of the protocol is as follows: a query defining a certain restriction on the available data is used as a first step to query for candidate data. The service returns a list of candidate hits formatted as a VOTable. For each candidate hit an access reference URL may be used to retrieve the real data. Data may be returned in a variety of formats.

This specification describes the “simple” way to access the data through an HTTP protocol. Other types of more complex access, eventually using SQL-like language and more complex transports as SOAP, will be worked on within the IPDA technical experts group.

Status of This Document

This is the first release of this document.

This is an IPDA Working Draft for review by IPDA members and other interested parties. It is a draft document and may be updated, replaced, or obsoleted by other documents at any time.

A list of [current IPDA Recommendations and other technical documents](http://planetarydata.org/technical-reports/) can be found at <http://planetarydata.org/technical-reports/>.

Contents

1	Overview	4
2	Requirements for Compliance	4
3	Planetary Data Service Types	5
4	General Service Metadata Query	6
4.1	Query format	6
4.2	PDAP classes description	7
4.3	Compulsory Query Parameters	9
4.3.1	INSTRUMENT_TYPE	9
4.3.2	INSTRUMENT_NAME	9
4.3.3	START_TIME	10
4.3.4	STOP_TIME	10
4.3.5	TARGET_TYPE	10
4.3.6	TARGET NAME	11
4.3.7	RETURN_TYPE	11
4.3.8	RESOURCE_CLASS	11
4.4	Query Parameters per Resource Class	12
4.4.1	Resource Class DATA_SET	12
4.4.2	Resource Class PRODUCT	12
4.4.3	Resource Class MAP_PROJECTED	13
4.5	Pagination Query Parameters	14
4.6	Successful Response	15
4.6.1	PDAP Version Information	15
4.6.2	Pagination Metadata Response	16
4.6.3	Generic Metadata Response per Resource Class	17
4.6.4	General Reference Attributes	23
5	Service Metadata	24
6	PDAP extensions for specific products	25
	Appendix A: Range definition for input parameters	26
	Appendix B: Valid Map Projected Data Service response in VOTable	26
	Appendix C: Registering a PDAP compliant service	28
	References	29

1 Overview

The Planetary Data Access Protocol (PDAP) is an International Planetary Data Alliance (IPDA) [1][2] protocol that defines how to interchange planetary data. This protocol has been developed as a combination of OGC [3] protocols (mainly for map-projected data) and IVOA protocols (for astronomical data but a science level data) like the Simple Image Access Protocol (SIAP) [4].

This close to other protocols definition allows easy integration with other data sources from other Virtual Observatories, but using particular language and characterization of planetary data.

The PDAP is a two steps protocol. That means that two steps need should be performed in order to access the data, in the following sequence:

- Step 1: Metadata Access
- Step 2: Data Access

In the first step, the archive is asked for the metadata pertaining to the data that obey a certain restriction in the input parameters. Among the compulsory output parameters of this metadata is a reference to the real file holding data. This allows for the second step of the protocol to be executed, as the data are readily made available through the declared link.

2 Requirements for Compliance

The keywords "MUST", "SHOULD", and "MAY" as used in this document are to be interpreted as described in RFC 2119 [7].

An implementation is compliant if it satisfies all the MUST level requirements for the protocols it implements. An implementation that satisfies all the MUST level and all the SHOULD level requirements for its protocols is said to be "unconditionally compliant"; one that satisfies all the MUST level requirements but not all the SHOULD level requirements for its protocols is said to be "conditionally compliant".

Compliance with this specification requires that an access web service be maintained with the following characteristics:

1. The **Metadata Query** web method MUST be supported as defined in section 4 below.

Through this web method, clients search for available data that match certain client-specified criteria. The response is a table that describes the available data, including metadata and access references (implemented here as URLs) for retrieving them.

2. The **Data Retrieval** web method **MUST** be supported as defined within the different use cases in section 4.6 below.

This method allows clients to retrieve data through a simple synchronous HTTP GET/POST request using the access reference (URL) returned earlier by the metadata query. The response to the GET/POST request is the requested data, returned with a MIME type such as "image/fits", "image/jpeg", "image/IMG" and so forth.

3. Finally the service **MUST** be **registered** by providing the information defined in Appendix C below.

Registration allows clients to use a central registry service to locate compliant PDAP services and select an optimal subset of services to query, based on the characteristics of each service and the data collections it serves.

3 Planetary Data Service Types

The PDAP protocol is very basic in nature: few input/output parameters are required, making the protocol very flexible and open. This flexibility allows the handling of as different data types as images, spectra, plasma information, atmospheric information, etc.

The general PDAP protocol defines "compulsory" input and output parameters within its working scheme for the general case. However, specific services might be specialised in specific subsystems or instruments, for instance only serving images. In this case, specific parameters are needed to ensure interoperability at the granularity level of the specific service.

Current general IPDA PDAP standard will only describe two types of services:

- General Data service
- Map projected data service

The General Data service abides by the input/output parameters described for the general PDAP protocol, whereas the Map projected data Service contains an extended set of input/output parameters with its own set of "compulsory" items.

The Map Projected Data Service will be described below as a specific case of the General Data service

Other services, like Spectral services, Atmospheric services, Plasma services, etc., will certainly need to be specified in a near future. Those services shall be described in PDAP extensions specifications. The way to generate a PDAP extension specification is described in section 6.

4 General Service Metadata Query

The purpose of the metadata query web method is to allow users to search for data for a given set of input parameters. The service will return references to data which most closely match what is requested. How closely the service can match the specifications of the data requested by the client will depend upon the type of data service. The service returns a table listing all available data satisfying the query. Descriptive metadata is returned for each hit to allow the client to decide which, if any, data it wishes to retrieve. An access reference is provided for each hit to permit subsequent staging and retrieval.

4.1 Query format

The metadata query input is transmitted as an HTTP GET/POST request represented by a URL with two parts:

1. A base URL of the form:
[http://<server-address>/<path>? \[<extra-arg>& \[...\]\]](http://<server-address>/<path>? [<extra-arg>& [...]])

Examples:

```
http://psa.esac.esa.int/aio/jsp/metadata.jsp?  
http://starbrite.jpl.nasa.gov/q?
```

Note that when it contains extra GET/POST arguments, the base URL ends in an ampersand, **&**; if there are no extra arguments, then it ends in a question mark, **?**. Every query to a given metadata query service uses the same base URL.

2. Constraints expressed as a list of ampersand-delimited GET/POST arguments, each of the form: *<name>=<value>*

Example:

```
TARGET_NAME=MARS&MISSION_NAME=MARS_EXPRESS&DATA_SET_ID=TSK-  
C-RCC-CAL-RSAWIRTANENV1.0
```

The constraints represent the query parameters which can vary for each successive query.

The baseURL and constraint list are concatenated to form the query.

Example:

```
http://psa.esac.esa.int:8000/aio/jsp/metadata.jsp?TARGET_NAME=MARS&
MISSION_NAME=MARS_EXPRESS&RESOURCE_CLASS=DATA_SET
```

4.2 PDAP classes description

The RESOURCE_CLASS input parameter defines the granularity of the PDAP response. As commented before, a valid PDAP response would be, in a general sense, a result table in a certain format (by default VOTable). Every row in the result table contains metadata information of the same kind of abstract object, e.g., the table response could contain a set of datasets (if RESOURCE_CLASS=DATA_SET is selected) one row per dataset, a set of products if RESOURCE_CLASS=PRODUCT is selected) one row per product, etc.

This schema is extensible to other kind of objects, typically extensions of the basic object PRODUCT. In this general PDAP specification document, a product extension object called MAP_PROJECTED is also allowed and described.

This is a summary of the classes supported by present specification and described in the UML diagram below:

- A data set is a self-consistent data package, as per PDS documentation description, that is compliant with an agreed directory structure. See [5] for reference. A data set contains a set of products.
- A product is the atomic self-consistent entity from the scientific point of view or the lowest data entity that has a PDS label for PDS compliant products. It can be composed by one or more physical files.
- Both data sets and products can be downloaded using a data access reference.
- As one example of product extension, a map projected product is also included in present specification. This map projected product class has associated a footprint object indirectly described in present specification.
- As an example, although not included in current general PDAP specification, UML contains possible extensions of Map Projected product. Any new extension could require a separate PDAP specification extension document.

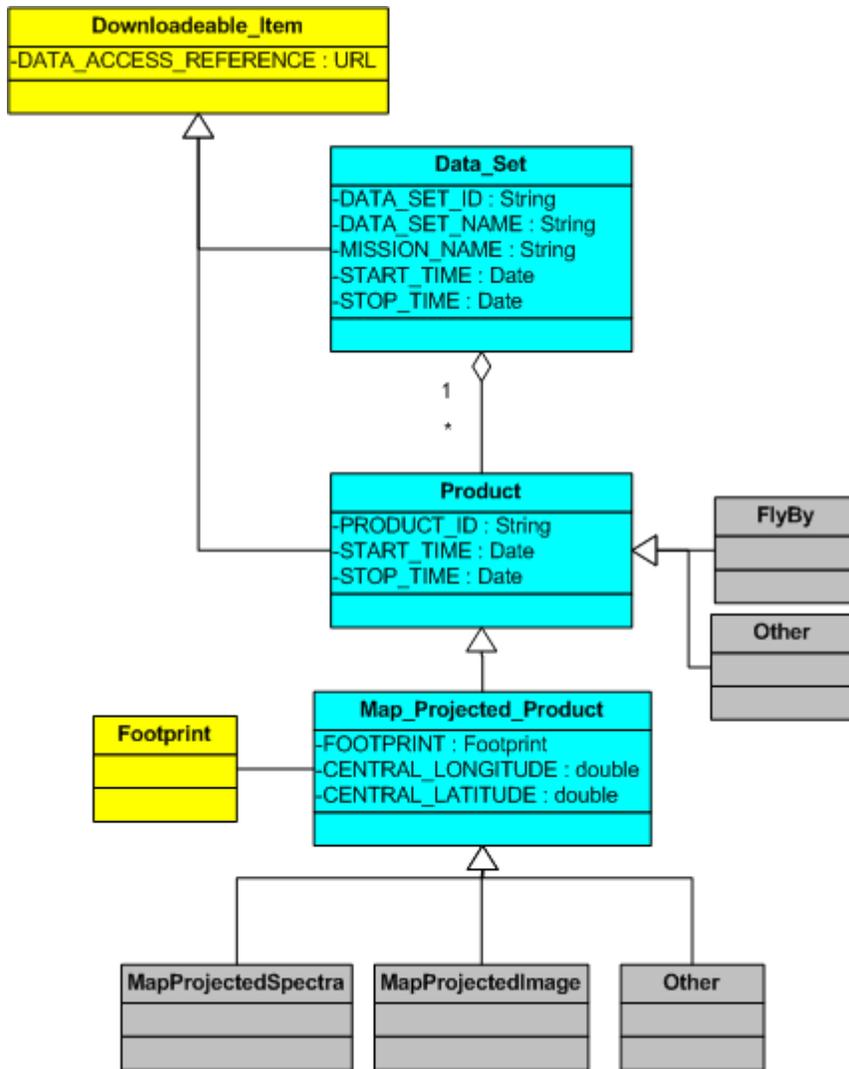


Figure 1: Simplified UML diagram of the resource classes described in the present PDAP specification. In light blue, classes directly described in the specification; in yellow, classes indirectly described in the spec; in gray, examples of possible extensions of map_projected product or product that could be designed in separate PDAP extensions specifications.

Other possible extensions like spectra, plasma data, atmospheric data, etc could be created and easily plugged into the system by the definition of the object data model extension and the direct identification of the new query parameters and output response that this new object could require. Extensions could be done from map projected product if a footprint could be associated to this product or directly from product (e.g. for fly-by products).

4.3 Compulsory Query Parameters

The following is a list of the compulsory query parameters that any service implementing the PDAP protocol **MUST** be able to understand. In this context, the “compulsory” attribute means that the service must be able to respond whenever any of these parameters is input in the “key=<value>” pair of the aforementioned base URL, but it is not needed that the service give compulsorily any result.

At the same time, not all the compulsory input parameters should be present in a valid PDAP URL query.

Values should be URL encoded (e.g. spaces replaced by + or %20)

4.3.1 INSTRUMENT_TYPE

The INSTRUMENT_TYPE element identifies the type of an instrument.

Examples:

POLARIMETER
RADIOMETER
SPECTROREFLECTOMETER
VIDICON CAMERA

See [5] for a full list of accepted values.

4.3.2 INSTRUMENT_NAME

The INSTRUMENT_NAME element provides the full name of an instrument.

Examples:

FLUXGATE
MAGNETOMETER
NEAR_INFRARED MAPPING
SPECTROMETER
HIGH RESOLUTION STEREO CAMERA

See [5] for a full list of accepted values.

4.3.3 START_TIME

The START_TIME element provides the date and time of the beginning of an event or observation (whether it be a spacecraft, ground-based, or system event) in UTC system

Formation rule:

YYYY-MM-DDThh:mm:ss[.fff]

Example:

1997-03-03T16:18:39.000

4.3.4 STOP_TIME

The STOP_TIME element provides the date and time of the end of an event or observation (whether it be a spacecraft, ground-based, or system event) in UTC system

Formation rule:

YYYY-MM-DDThh:mm:ss[.fff]

Example:

1997-03-11T19:29:36.000

4.3.5 TARGET_TYPE

The TARGET_TYPE element identifies the type of a named target.

Examples:

PLANET
SATELLITE
RING
REGION
FEATURE
ASTEROID
COMET
PLASMA ENVIRONMENT

See [5] for a full list of accepted values.

4.3.6 TARGET_NAME

The TARGET_NAME element identifies a target. The target may be a planet, satellite, ring, region, feature, asteroid or comet.

Examples:

GANYMEDE
MARS
5797 BIVAJ

4.3.7 RETURN_TYPE

The return type describes the type of output that should be received after the metadata query. The default response format will be VOTable [6]. Other possible formats could be HTML, CSV, KML and ASCII although these are not formally defined in present specification.

Examples:

VOTABLE
HTML
CSV

4.3.8 RESOURCE_CLASS

The RESOURCE_CLASS input parameter defines the granularity of the PDAP response. See section 4.2 for a detailed description.

Examples:

DATA_SET
PRODUCT
MAP_PROJECTED

4.4 Query Parameters per Resource Class

Compliant services for specific resource classes could require the implementation of extra compulsory query parameters. We describe in the present specification the ones related with the resource classes covered in this general PDAP specification (DATA_SET, PRODUCT and MAP_PROJECTED).

Query input parameters for other resource classes will be described in the relevant PDAP core specification extensions documents (see section 6)

4.4.1 Resource Class DATA_SET

PDAP services implementing RESOURCE_CLASS=D granularity response **MUST** implement the general PDAP compulsory query input parameters plus the following query input parameter

4.4.1.1 DATA_SET_ID

The DATA_SET_ID element is a unique alphanumeric identifier for a data set or a data PRODUCT. The DATA_SET_ID value for a given data set or product is constructed according to flight project naming conventions. In most cases the DATA_SET_ID is an abbreviation of the DATA_SET_NAME.

Example:

MR9/VO1/VO2-M-ISS/VIS-5-CLOUD-V1.0

4.4.2 Resource Class PRODUCT

PDAP services implementing RESOURCE_CLASS=PRODUCT granularity response **MUST** implement the general PDAP compulsory query input parameters plus the following query input parameter

4.4.2.1 DATA_SET_ID

The DATA_SET_ID element is a unique alphanumeric identifier for a data set or a data PRODUCT. The DATA_SET_ID value for a given data set or product is constructed according to flight project naming conventions. In most cases the DATA_SET_ID is an abbreviation of the DATA_SET_NAME. For not PDS

complaints resources, DATA_SET_ID value should be a unique identifier that allows the grouping of similar products, although it could not have real entity.

Example:

MR9/VO1/VO2-M-ISS/VIS-5-CLOUD-V1.0

4.4.2.2 PRODUCT_ID

The PRODUCT_ID data element represents a permanent, unique identifier assigned to a data product by its producer within a dataset. That means that the combination DATA_SET_ID.PRODUCT_ID should be unique. The PRODUCT_ID can describe the lowest-level data object that has a PDS [5] label.

Example:

TSK_C_RCC_3_19970303T165718

4.4.3 Resource Class MAP_PROJECTED

As already mentioned in point 3, the case of the map projected data access is a very special one in the Inter-planetary science world, as a lot of the information from Planetary science has historically come from e.g. images from the planetary bodies with associated geometrical information.

Rather than defining a new protocol for map projected data access, we have decided to include it in the more general PDAP protocol as a specific case. However, specific input/output parameters are required for this type of data. In particular, the Region Of Interest (ROI) of the observation must be defined for the map projected data access in order to characterize the geometrical information attached to the data.

Although we agree that the definition of these parameters might be difficult due to the very nature of the different planetary bodies, etc., we define a four-corners ROI search as compulsory input parameters, and a so called FOOTPRINT for the output parameters. This definition is quite close to OpenGIS Map Services but using specific planetary data characterization through PDS keywords and describing a standard way to access scientific data.

A *Map Projected Data Service* **MUST** implement the general PDAP input/output parameters plus the following query input parameters to define a Region Of Interest (ROI) for the product search.

4.4.3.1 LONGITUDE

A Map Projected Data Service **MUST** implement a parameter to specify a LONGITUDE range (For range definition, please refer to appendix A)

For a Planetocentric, body-fixed, rotating coordinate system, longitude is defined as: The angle increasing eastward between the prime meridian and the vector from the coordinate system origin to the point of interest, projected into the equatorial plane. This is a right-handed coordinate system. For a Planetographic, body-fixed, rotating coordinate system, longitude is defined as: The angle between the prime meridian and the vector from the coordinate system origin to the point of interest, projected into the equatorial plane. Planetographic longitudes are defined to increase with time for a distant observer. Thus, they increase to the west for prograde rotators, and to the east for retrograde rotators. For the Earth, Moon and Sun, PDS also supports the traditional use of the range (-180,180)

4.4.3.2 LATITUDE

A Map Projected Data Service **MUST** implement a parameter to specify a LATITUDE range (For range definition, please refer to appendix A). Combined with the LONGITUDE range, both parameters define a region of interest.

For a Planetocentric, body-fixed, rotating coordinate system, latitude is defined as: The angle between the equatorial plane and a vector connecting the point of interest and the origin of the planetocentric coordinate system. Positive in the hemisphere north of the equator (i.e., hemisphere to the north of the solar system invariant plane) and negative in the southern hemisphere. For a Planetographic, body-fixed, rotating coordinate system, latitude is defined as: The angle between the equatorial plane and a vector through the point of interest that is normal to a biaxial ellipsoid reference surface. Positive in the hemisphere north of the equator (i.e., hemisphere to the north of the solar system invariant plane) and negative in the southern hemisphere.

4.5 *Pagination Query Parameters*

A PDAP service may implement pagination. This capability should be discovered by a service metadata query as per section 5. In order to implement pagination, the following query parameters should be implemented at server level.

4.5.1.1 PAGE_SIZE

The query parameter PAGE_SIZE will be used by the client to select the page size of the expected result. This should be considered a maximum size. In case the result of the query contains less than the number requested, the full response will be offered in the result.

4.5.1.2 PAGE_NUMBER

The query parameter PAGE_NUMBER will be used by the client to select the page number of the expected result. It should be used in combination with PAGE_SIZE

4.6 Successful Response

The default output returned by a **Metadata Query** is a VOTable [6] version 1.1, an XML table format, returned with a MIME-type of application/x-votable+xml. The table lists all the data available to the client that match the query constraints. The following requirements are placed on the contents of the table when the query successfully returns a list of available data:

1. The VOTable **MUST** contain a RESOURCE element, identified with the tag type="results", containing a single TABLE element which contains the results of the query. The VOTable is permitted to contain additional RESOURCE elements, but the usage of any such elements is not defined here. If multiple resources are present it is recommended that the query results be returned in the first resource element.
2. The RESOURCE element SHOULD contain an INFO with name="QUERY_STATUS". Its value attribute should set to "OK" if the query executed successfully, regardless of whether any matching products were found. All other possible values for the value attribute are described in [section 4.3](#) below.

Examples:

```
<INFO name="QUERY_STATUS" value="OK"/>
<INFO name="QUERY_STATUS" value="OK">
Successful Search</INFO>
```

3. Each table row represents a different data available to the client.
4. Each row of the output VOTable MUST contain FIELDS where a certain *utype* or ID have been set.

4.6.1 PDAP Version Information

A PDAP compliant output **SHOULD** contain an INFO tag with name PDAP_VERSION and value the PDAP specification version implemented by the service. This INFO tag must be inside the RESOURCE tag due to VOTable compatibility. (see RESOURCE_CLASS=METADATA output as an example). This information should appear in a RESOURCE_CLASS=METADATA discovery query.

This tag has two different uses:

- Provide information to client applications about the version of the PDAP specification implemented by the PDAP server so clients could prevent parsing problems.
- Allow automatically discover version information by registry servers.

Example:

```
<INFO name="PDAP_VERSION" value="0.5"/>
```

See section Appendix B for a more detailed example.

4.6.2 Pagination Metadata Response

A VOTable response of a service implementing pagination may contain the following PARAMs at RESOURCE level:

- It **MAY** contain one PARAM with the name **TOTAL_RECORDS** and the total number of records that fulfill the query as value
- It **MAY** contain one PARAM with the name **PAGE_SIZE** and number of records present in the current VOTable response
- It **MAY** contain one PARAM with the name **PAGE_NUMBER** page number that correspond with the current VOTable response as value

Example of VOTable header including pagination:

```
<?xml version="1.0" ?>
<!DOCTYPE VOTABLE (View Source for full doctype...)>
<VOTABLE version="1.1">
  <RESOURCE type="results">
    <DESCRIPTION>PSA Metadata Query Service</DESCRIPTION>
    <INFO name="PDAP VERSION" value="0.3"/>
    <INFO name="QUERY STATUS" value="OK" />
    <PARAM name="TOTAL RECORDS" value="10000"/>
    <PARAM name="PAGE NUMBER" value="3"/>
    <PARAM name="PAGE SIZE" value="1000"/>
  <TABLE>
```

4.6.3 Generic Metadata Response per Resource Class

We describe in this section the metadata response fields to be present in the response of services implementing the resource classes described in the present core PDAP specification (DATA_SET, PRODUCT and MAP_PROJECTED).

Metadata response fields for other resource classes will be described in the relevant PAP core specification extensions documents (see section 8).

4.6.3.1 Resource Class DATA_SET

These are the resource class DATA_SET specific metadata response fields:

- Exactly one field **MUST** have one *utype*="pdap:DATA_SET.DATA_SET_ID" with datatype="char", and arraysize="*", with obvious contents (see previous definitions).
- Exactly one field **MUST** have one *utype*="pdap:DATA_SET.MISSION_NAME " with datatype="char", and arraysize="*", with obvious contents (see previous definitions).
- Exactly one field **MUST** have one *utype*="pdap:DATA_SET.START_TIME" and one *utype*="pdap:DATA_SET.STOP_TIME" with datatype="char" and arraysize="*" with obvious contents (see previous definitions)
- Exactly one field **MUST** have one *ID*="RESOURCE_CLASS" with datatype="char", and arraysize="*". For this protocol version, the only one value accepted is DATA_SET. See same field description for PRODUCT RESOURCE_CLASS.
- Exactly one field **MUST** have one *utype*="pdap:DATA_SET.DATA_SET_NAME", with datatype="char", and arraysize="*", containing a short (usually one line) description of the data set. This should concisely describe the data set to a user, typically identifying the data source (e.g., survey name), object name or field coordinates, bandpass/filter, and so forth. Note that a client application may present the user with a combined list of data from many different queries and sources: the title string is important to identify each data to the user.
- Exactly one field **MUST** have *ID*="DATA_ACCESS_REFERENCE", with datatype="char" and arraysize="*", specifying the URL to be used to access or retrieve the data. Since the URL will often contain metacharacters the URL is normally enclosed in an XML CDATA section (<![CDATA[...]]>) or otherwise encoded to escape any embedded metacharacters.

Example:

```
<![CDATA[  
http://psa.esac.esa.int:8000/aio/jsp/product.jsp?dataSetID=MEX-M-  
ASPERA3-2-EDR-ELS-V1.0&PROTOCOL=HTTP]]>
```

4.6.3.2 Resource Class PRODUCT

These are the resource class PRODUCT specific metadata response fields:

- Exactly one field **MUST** have one *utype*="pdap:PRODUCT.PRODUCT_ID" (the combination of DATA_SET_ID plus PRODUCT_ID is required to identify the product) with datatype="char", and arraysize="*", with obvious contents (see previous definitions).
- Exactly one field **MUST** have one *utype*="pdap:DATA_SET.MISSION_NAME " with datatype="char", and arraysize="*", with same contents than previous definition.
- Exactly one field **MUST** have one *utype*="pdap:PRODUCT.START_TIME" and one field **MUST** have *utype*="pdap:PRODUCT.STOP_TIME", with obvious contents (see previous definitions)
- Exactly one field **MUST** have one *ID*="RESOURCE_CLASS" with datatype="char", and arraysize="*". Even when there is a query parameter **RESOURCE_CLASS** to filter the query output, this field could contain different heterogeneous values, e.g. IMAGE and SPECTRUM (if these resource classes are defined in an external document) for a PRODUCT query. The use of this parameter would be to help clients to handle products in a better way, using e.g. specific helper applications.

Examples of valid values:

PRODUCT, MAP_PROJECTED

- Exactly one field **MUST** have one *utype*="pdap:PRODUCT.PRODUCT_ID", with datatype="char", and

arraysize="*", containing a short (usually one line) description of the product. Note that a client application may present the user with a combined list of data from many different queries and sources: the title string is important to identify each data to the user.

- Exactly one field **MUST** have *ID*="DATA_ACCESS_REFERENCE", with datatype="char" and arraysize="*", specifying the URL to be used to access or retrieve the data. Since the URL will often contain metacharacters the URL is normally enclosed in an XML CDATA section (<![CDATA[...]]>) or otherwise encoded to escape any embedded metacharacters.

Example:

```
<![CDATA[http://psa.esac.esa.int:8000/aio/jsp/product.jsp?dataSetID=MEX-M-HRSC-3-RDRV2.0&productID=H1483\_000\_P22.IMG&PROTOCOL=HTTP] ]>
```

- Exactly one field **COULD** have *ID*="ICON_ACCESS_REFERENCE", with datatype="char" and arraysize="*", specifying the URL to be used to access to an ICON representation of the product. Since the URL will often contain metacharacters the URL is normally enclosed in an XML CDATA section (<![CDATA[...]]>) or otherwise encoded to escape any embedded metacharacters.

Example:

```
<![CDATA[http://psa.esac.esa.int:8000/aio/jsp/product.jsp?dataSetID=MEX-M-HRSC-3-RDRV2.0&productID=H1483\_000\_P22.IMG&rule=retrieveBrowseImage&compression=lose&PROTOCOL=HTTP] ]>
```

4.6.3.3 Resource Class MAP_PROJECTED

As MAP_PROJECTED is a PRODUCT extension, output response **MUST** contain all the compulsory fields for the PRODUCT class, except the ones overwritten in present resource class description.

These are the resource class MAP_PROJECTED specific metadata response fields:

- Exactly one field **MUST** have one utype="MAP_PROJECTED.FOOTPRINT" with datatype="char", and arraysize="*". The **FOOTPRINT** will be specified as follows:

- **Polygonal** regions

POLYGON(LON_1:LAT_1,LON_2:LAT_2,....,LON_n:LAT_n)

- **Circular** regions:

CIRCLE(CENTER_LON:CENTER_LAT,RADIUS)

- **Landmarks:**

POINT(LON:LAT)

- Exactly one field **MUST** have one `utype="pdap:MAP_PROJECTED.CENTER_LONGITUDE"` with `datatype="char"`, and `arraysize=""`.
The *CENTER_LONGITUDE* element provides a reference longitude for certain map projections. For example, in an Orthographic projection, the center_longitude along with the center_latitude defines the point or tangency between the sphere of the planet and the plane of the projection. The map_scale (or map_resolution) is typically defined at the center_latitude and center_longitude.
- Exactly one field **MUST** have one `utype="pdap:MAP_PROJECTED.CENTER_LATITUDE"` with `datatype="char"`, and `arraysize=""`.
The *CENTER_LATITUDE* element provides a reference latitude for certain map projections. For example, in an Orthographic projection, the center_latitude along with the center_longitude defines the point or tangency between the sphere of the planet and the plane of the projection. The map_scale (or map_resolution) is typically defined at the center_latitude and center_longitude.
- At a maximum of one field **MAY** have one `utype="pdap:MAP_PROJECTED.COORDINATE_SYSTEM_TYPE"` with `datatype="char"`, and `arraysize=""` describing the type of coordinate system used for the footprint and the center longitude/latitude fields.
There are three basic types of coordinate systems: body-fixed rotating, body-fixed non-rotating and inertial. A body-fixed coordinate system is one associated with a body (e.g., planetary body or satellite). In contrast to inertial coordinate systems, a body-fixed coordinate system is centered on the body and rotates with the body (unless it is a non-rotating type). For the inertial coordinate system type, the coordinate system is fixed at some point in space. If this field is not present the default value is assumed to be body-fixed rotating Planetographic.

List of possible values:

BODY-FIXED NON-ROTATING
BODY-FIXED ROTATING
INERTIAL
SCALAR

- At a maximum of one field **MAY** have one *utype*="pdap:MAP_PROJECTED.COORDINATE_SYSTEM_NAME" with *datatype*="char", and *arraysize*="*" providing the full name of the coordinate system to which the state vectors are referenced and used for the footprint and the center longitude/latitude fields.
There are three basic types of coordinate systems: body-fixed rotating, body-fixed non-rotating and inertial. A body-fixed coordinate system is one associated with a body (e.g., planetary body or satellite). In contrast to inertial coordinate systems, a body-fixed coordinate system is centered on the body and rotates with the body (unless it is a non-rotating type). For the inertial coordinate system type, the coordinate system is fixed at some point in space. If this field is not present the default value is assumed to be body-fixed rotating Planetographic.

List of possible values:

BODY FIXED SPHERICAL COORDS
EARTH-SUN LINE CARTES COORDS
ECLIPTIC INERTIAL CART COORDS
ECLIPTIC INERTL SPHERCL COORDS
EQUATORIAL INERT SPHRCL COORDS
EQUATORIAL INERTIAL CART COORD
JUPITER MINUS SYSTEM III
MAST_FRAME
MB FRAME
MEAN INERTIAL HG 1950
MI_FRAME
NEPTUNE WEST LONGITUDE SYSTEM
NON-ROTATING SPIN COORDINATES
PLANET CENTERED CYLINDRICAL
PLANETOCENTRIC
PLANETOGRAPHIC
PVO INERTIAL SPACECRAFT COORDS
PVO SPINNING SPACECRAFT COORDS
ROVER_FRAME
SATURN MINUS LONGITUDE SYSTEM
SC CENTERED ECLIPTIC COORDS
URANUS MINUS LONGITUDE SYSTEM
URANUS WEST LONGITUDE SYSTEM
VENUS SOLAR ORBITAL COORDS
S/C-COORDS
INSTRUMENTCOORDS
ECLIPJ2000
EARTH MEAN EQUATOR J2000
NAC_CAMERA_FRAME
WAC_CAMERA_FRAME

According to the above set of parameters, a possible PDAP map projected data query could be as follows:

Example:

```
http://psa.esac.esa.int:8000/aio/jsp/metadata.jsp?TARGET_NAME=MARS
&LATITUDE=25/30&LONGITUDE=35/40&
RESOURCE_CLASS=MAP_PROJECTED&
RETURN_TYPE=HTML
```

- Exactly one field **MUST** have one *ID*="RESOURCE_CLASS" with datatype="char", and arraysize="*". Even when there is a query parameter **RESOURCE_CLASS** to filter the query output, this field could contain different heterogeneous values, e.g. IMAGE and SPECTRUM (if these resource classes are defined in an external document). The use of this parameter would be to help clients to handle products in a better way, using e.g. specific helper applications.

See Appendix B for an example valid Map Projected Service response in VOTable (with positional parameters).

4.6.4 General Reference Attributes

The following metadata fields should be present inside any PDAP response for all the resource classes implementations.

- Exactly one field **MUST** have one *utype*="pdap:<resource_class>.REFERENCE_FORMAT" with datatype="char", and arraysize="*", where <resource_class> could be DATA_SET or PRODUCT. If the field content is the same for all the records within the response, this information could be added through a PARAM in the VOTable response.
- Exactly one field **MUST** have one *utype*="pdap:<resource_class>.PUBLISHER" with datatype="char", and arraysize="*", where <resource_class> could be DATA_SET or PRODUCT. A *PUBLISHER* is an entity responsible for making the resource available. Examples of a Publisher include a person, an organisation, or a service. Typically, the name of a Publisher should be used to indicate the entity. If the field content is the same for all the records within the response, this information could be added through a PARAM in the VOTable response.
- Exactly one field **MUST** have one *utype*="pdap:<resource_class>.CONTRIBUTOR" with datatype="char", and arraysize="*". A *CONTRIBUTOR* is an entity responsible for making contributions to the content of the resource. Examples of a Contributor include a person, an organisation, or a service. Typically, the name of a Contributor should be used to indicate the entity. If the field content is the same for all the records within the response, this information could be added through a PARAM in the VOTable response.
- Exactly one field **MUST** have one *utype*="pdap:<resource_class>.PUBLISHING_DATE" with datatype="char", and arraysize="*". If the field content is the same for all

the records within the response, this information could be added through a PARAM in the VOTable response.

- Exactly one field **MUST** have one *utype*="pdap:<resource_class>.RIGHTS" with datatype="char", and arraysize="*". The *RIGHTS* entry represent a person or organization owning or managing rights over the resource. Recommended best practice is to use the URI or name of the Rights Holder to indicate the entity. If the field content is the same for all the records within the response, this information could be added through a PARAM in the VOTable response.

5 Service Metadata

This type of query gives as a result the list of parameters that the server accepts as inputs, which would be an expanded set composed of the compulsory parameters plus others that the server would understand, which are considered "OPTIONAL".

Due to the very nature of the data, different Data Sets might have different metadata optional values associated. This would mean that a "RESOURCE_CLASS=METADATA" query might need a further specification in the input parameters to do a finer grain filtering of the sought metadata parameters.

For the time being we will only specify a general metadata query, and will leave for further discussions which parameters might be used to filter the different cases depending on the Data Sets.

According to the aforementioned points, a prototypical query of this type would be like:

http://psa.sciops.esa.int/PAIO/piap.jsp?RESOURCE_CLASS=METADATA

and a prototypical output could be something like:

```
<?xml version="1.0" ?>
<!DOCTYPE VOTABLE (View Source for full doctype...)>
-<VOTABLE version="1.1">
  -<RESOURCE type="results">
    <INFO name="PDAP_VERSION" value="0.3" />
    <DESCRIPTION>ESA Planetary Science Archive PDAP Query
      Service</DESCRIPTION>
    -<PARAM name=" INPUT:CENTER_LONGITUDE">
      <DESCRIPTION>Planetocentric Center longitude center of Region of
        Interest</DESCRIPTION>
    </PARAM>
```

```

-<PARAM name=" INPUT:CENTER_LATITUDE">
  <DESCRIPTION>Planetocentric Center latitude center of Region of
  Interest</DESCRIPTION>
</PARAM>
[...]etc.... all compulsory input parameters should appear here

-<PARAM name="INPUT:SUB_SOLAR_LONGITUDE">
  <DESCRIPTION>" The sub_solar_longitude element provides the
  longitude of the subsolar point. The subsolar point is that point on
  a body's reference surface where a line from the body center to the
  sun center intersects that surface. Note: The
  coordinate_system_type data element should be used in
  conjunction with this data element." </DESCRIPTION>
</PARAM>

-<PARAM name="INPUT:SUB_SOLAR_LATITUDE">
  <DESCRIPTION>"The sub_solar_latitude element provides the latitude
  of the subsolar point. The subsolar point is that point on a body's
  reference surface where a line from the body center to the sun
  center intersects that surface" </DESCRIPTION>
</PARAM>

[...] etc., any optional input search parameters accepted by the server.

</RESOURCE>
</VOTABLE>

```

This type of query serves the purpose of unveiling the type of information that can be queried to the server. Smart clients can use this feature to build on-the-fly User Interfaces to fill in required parameters to perform complex queries to the server.

6 PDAP extensions for specific products

PDAP is based in the resource class concept, i.e. a specific class (data set, product, map_projected, etc) requires specific metadata to be handled properly at client side. However, some other extensions are possible and to include all of them in current specification would require a permanent living document.

In order to allow future extensions, the following rules should be followed to create a new resource class that better describe your data:

- Select a new resource class identifier. Current reserved values are DATA_SET, PRODUCT and MAP_PROJECTED
- Identify PDS keywords that could be used to characterize the data
- A standard resource class document extension should be created and raised to IPDA level for review. If considered of general use, this new resource class definition would be incorporated as a standard IPDA specification.

Appendix A: Range definition for input parameters

For a simple HTTP GET/POST service, the constraints in the queries for the input parameters need to be done in a valid URL.

However, the URL standard is not very specific defining special constraints as, e.g., ranges. We describe here some examples of accepted range syntax for URL PDAP services:

param = x	Equality	Equivalent to param=x
param = x/y	Closed range	Equivalent to param>=X AND param<=y
param = x/	Open range	Equivalent to param>=x
param = x,y	List	Equivalent to param=x OR param=y
param = x/a,b/y	Range list	Equivalent to [param<=x AND param<=a] OR [param>=b and param<=y]

Appendix B: Valid Map Projected Data Service response in VOTable

```
<?xml version="1.0" ?>
<!DOCTYPE VOTABLE (View Source for full doctype...)>
<VOTABLE version="1.1">
  <RESOURCE type="results">

    <INFO name="PDAP_VERSION" value="0.3" />

    <DESCRIPTION>PSA Metadata Query Service</DESCRIPTION>
    <INFO name="QUERY_STATUS" value="OK" />
    <TABLE>

<FIELD ID="PRODUCT.PRODUCT_ID" ucd="PRODUCT_ID" utype="pdap:PRODUCT.PRODUCT_ID"
datatype="char" arraysize="*" />

<FIELD ID="DATA_ACCESS_REFERENCE" ucd="DATA_ACCESS_REFERENCE"
datatype="char" arraysize="*" />

<FIELD ID="PRODUCT.FOOTPRINT" ucd="FOOTPRINT"
utype="pdap:MAP_PROJECTED.FOOTPRINT" datatype="char" arraysize="*" />

<FIELD ID="CENTRAL_LONGITUDE" ucd="LONGITUDE"
utype="pdap:MAP_PROJECTED.CENTRAL_LONGITUDE" datatype="double" />

<FIELD ID="CENTRAL_LATITUDE" ucd="LATITUDE"
utype="pdap:MAP_PROJECTED.CENTRAL_LATITUDE" datatype="double" />

<FIELD ID="DATA_SET.DATA_SET_ID" ucd="DATA_SET_ID" utype="pdap:DATA_SET.DATA_SET_ID"
datatype="char" arraysize="*" />

<FIELD ID="DATA_SET.DATA_SET_NAME" ucd="DATA_SET_NAME"
utype="pdap:DATA_SET.DATA_SET_NAME" datatype="char" arraysize="*" />

<FIELD ID="DATA_SET.MISSION_NAME" ucd="MISSION_NAME" utype="pdap:DATA_SET.MISSION_NAME"
datatype="char" arraysize="*" />
```

```

<FIELD ID="DATA_SET.PRODUCER.FULL_NAME" ucd="FULL_NAME
utype="pdap:DATA_SET.PRODUCER.FULL_NAME" datatype="char" arraysize="*" />

<FIELD ID="DATA_SET.PRODUCER.INSTITUTION_NAME" ucd="INSTITUTION_NAME"
utype="pdap:DATA_SET.PRODUCER.INSTITUTION_NAME" datatype="char" arraysize="*" />

<FIELD ID="DATA_SET.PRODUCER.NODE_NAME" ucd="NODE_NAME"
utype="pdap:DATA_SET.PRODUCER.NODE_NAME" datatype="char" arraysize="*" />

<FIELD ID="PRODUCT.TARGET_NAME" ucd="TARGET_NAME" utype="pdap:PRODUCT.TARGET_NAME"
datatype="char" arraysize="*" />

<FIELD ID="PRODUCT.TARGET_TYPE" ucd="TARGET_TYPE" utype="pdap:PRODUCT.TARGET_TYPE"
datatype="char" arraysize="*" />

<FIELD ID="PRODUCT.INSTRUMENT_ID" ucd="INSTRUMENT_ID" utype="pdap:PRODUCT.INSTRUMENT_ID"
datatype="char" arraysize="*" />

<FIELD ID="PRODUCT.INSTRUMENT_NAME" ucd="INSTRUMENT_NAME" utype="
pdap:PRODUCT.INSTRUMENT_NAME" datatype="char" arraysize="*" />

<FIELD ID="PRODUCT.START_TIME" ucd="START_TIME" utype="pdap:PRODUCT.START_TIME"
datatype="char" arraysize="*" />

<FIELD ID="PRODUCT.STOP_TIME" ucd="STOP_TIME" utype=" pdap:PRODUCT.STOP_TIME"
datatype="char" arraysize="*" />

<FIELD ID="ICON_ACCESS_REFERENCE" ucd="ICON_ACCESS_REFERENCE" datatype="char"
arraysize="*" />

<FIELD ID="RESOURCE_CLASS" ucd="RESOURCE_CLASS" datatype="char" arraysize="*" />

<DATA>
<TABLEDATA>
<TR>
<TD>H1428_0000_BL2.IMG</TD>
</TD>
<![CDATA[
http://psa.esac.esa.int:8000/aio/jsp/product.jsp?dataSetID=MEX-M-HRSC-3-RDR-
V2.0&productID=H1428_0000_BL2.IMG&PROTOCOL=HTTP
]]>
</TD>

<TD>POLYGON(30.941999435424805:50.5369987487793,30.961999893188477:50.29100036621094,30.9
81000900268555:50.04399871826172,31.000999450683594:49.797000885009766,31.020000457763672
: [more polygon data]
84000015258789:49.2239990234375,28.820999145507812:49.47600173950195,28.802000045776367:4
9.724998474121094,28.783000946044922:49.97100067138672,28.76300048828125:50.2169990539550
8,28.742000579833984:50.4630012512207)</TD>
<TD>29.7412395123</TD>
<TD>50.3191296217</TD>
<TD>MEX-M-HRSC-3-RDR-V2.0</TD>
<TD>HRSC REDRs</TD>
<TD>MARS EXPRESS</TD>
<TD>Deutsches Zentrum fuer Luft und Raumfahrt (DLR), Institut fu</TD>
<TD>DLR</TD>
<TD />
<TD>MARS</TD>
<TD>SATELLITE, SATELLITE, PLANET</TD>
<TD>HRSC</TD>
<TD>HIGH RESOLUTION STEREO CAMERA</TD>
<TD>2005-02-26 14:36:19.316666666</TD>
<TD>2005-02-26 14:46:09.236666666</TD>
= <TD>
- <![CDATA[
http://psa.esac.esa.int:8000/aio/jsp/product.jsp?dataSetID=MEX-M-HRSC-3-RDR-
V2.0&productID=H1428_0000_BL2.IMG&rule=retrieveBrowseImage&compression=loose&protocol=HTT
P
]]>
</TD>
<TD>MAP_PROJECTED</TD>
</TR>
</TABLEDATA>
</DATA>
</TABLE>
</RESOURCE>
</VOTABLE>

```

Appendix C: Registering a PDAP compliant service

Registration of PDAP services will be done through a standard IPDA registration service or a external registry associated to other VO initiatives. Although the technical details of the registration of PDAP services is out of the scope of the present document, the following metadata would be required for registration:

- **Title:** The name of the PDAP service. There are no restrictions on the format of the value; however, it is recommended that if the service is associated with a particular data collection, the service Title contain a recognizable form of the collection's name.
- **Publisher:** The entity responsible for making the service available. This is usually the organization or data center name (e.g. "Space Telescope Science Institute/MAST"); however, it can be a person's name.
- **Contact.Name:** The name of a party (i.e. person or group) that can be contacted regarding the content, implementation, or maintenance of the service.
- **Contact.Email:** The email address of a party (i.e. person or group) that can be contacted regarding the content, implementation, or maintenance of the service.
- **ReferenceURL:** A URL pointing to additional information about the service or the PDAP data provided by the service. This can be a human-readable document specifically describing this service, or it can just be the home page for the data provider.

References

- [1] IPDA International Planetary Data Alliance, <http://planetarydata.org/>
- [2] Osuna P. et al, NASA-PDS/ESA-PSA Planetary Data Interoperability White Paper, <http://planetarydata.org/documents/white-paper-wp/>
- [3] Jeff de la Beaujardiere et al, OGC Web Map Service Interface. OGC standard, <http://www.opengeospatial.org/standards/wms>
- [4] Tody D, Plante R., Simple Image Access Protocol, IVOA Recommendation, <http://www.ivoa.net/Documents/SIA/>
- [5] NASA PDS, Planetary Data System Standards Reference, <http://pds.nasa.gov/tools/standards-reference.shtml>
- [6] Ochsenbein F, Williams R., VOTable Format Definition, IVOA Recommendation, <http://www.ivoa.net/Documents/VOTable/>
- [7] Bradner S, Keywords for use in RFCs to Indicate Requirement Levels, <http://www.ietf.org/rfc/rfc2119.txt>