esac

European Space Astronomy Centre
Camino Bajo del Castillo s/n
Urb. Villafranca del Castillo
28692 Villanueva de la Cañada
Madrid
Spain
Tel. (34) 91 813 1100
Fax (34) 91 813 1139
www.esa.int

# PSA PDS4 ARCHIVING GUIDE

|  |  |
|---|---|
| **Prepared by** | **S. Martinez, T. Lim, M. Costa, M.S. Bentley** |
| **Reference** | **ESDC-PSA-TN-0002** |
| **Issue / Revision** | **2 / 3** |
| **Date of Issue** | **26/07/2018** |
| **Status** | **Issued** |
| **Document Type** | **TN** |
| **Distribution** | **-** |

Original - https://issues.cosmos.esa.int/socciwiki/display/PSA/ESDC-PSA-TN-0002+PSA+PDS4+Archiving+Guide

# APPROVAL

| **PSA PDS4 Archiving Guide** | |
|---|---|
| **Issue** 2 | **Revision** 3 |
| **Author** S. Martinez, T. Lim, M. Costa, M.S. Bentley | **Date** 02-08-2017 |
| **Approved by** | **Date** |
| B. Merin | 23-11-18 |
| S. Besse | 23-11-18 |
| I. Barbarisi | 23-11-18 |

# CHANGE LOG

| **Reason for change** | **Issue** | **Revision** | **Date** |
|---|---|---|---|
| First draft for iteration with missions using PDS4 | D | 1 | 13/11/2015 |
| Overview of the archiving process added | D | 2 | 22/12/2015 |
| Name of partially processed collection updated to “data_processed” | D | 2 | 03/02/2016 |
| “miscellaneous_ancillary” collection added to mission and instrument bundles<br>Extension for collection inventory files changed to cvs | D | 2 | 04/02/2016 |
| Naming convention for context products updated, following PDS4 DPH 1.4.0 | 1 | 0 | 11/02/2016 |
| Minor updates following comments from ALTEC for ExoMars RSP | 1 | 1 | 18/10/2016 |
| Removed SPICE Collection, updated context lids | 1 | 2 | 26/04/2017 |
| Added SPICE Bundle Sections. Updated SPICE references | 2 | 0 | 18/06/2017 |
| Added policies for data products, more details on labels and specified namespaces.<br>Added requirements and recommendations on Geometry metadata, FITS and CDF. | 2 | 1 | 26/07/2018 |
| Minor updates following internal review | 2 | 2 | 18/09/2018 |
| Corrected SPICE Bundle Section | 2 | 3 | 03/10/2018 |

# CHANGE RECORD

| **Issue** 2 | **Revision** 3 | | |
|---|---|---|---|
| **Reason for change** | **Date** | **Pages** | **Paragraph(s)** |
| Naming convention for context products updated, following PDS4 DPH 1.4.0. | 18/06/2017 | All | All |
| Minor updates following comments from ALTEC for ExoMars RSP | 18/10/2016 | 6, 16, 17, 18, 28 | 2, 4.1, 4.3, 4.4, 6.2.2.1 |
| Minor updates in the introduction to justify the use of PDS4 | 02/08/2017 | 5 | 1.1 |
| Added target context rules and host bundle to table 5 | 30/01/2018 | sect 3.2.1, 3.3.1 | All |

| | | | |
|---|---|---|---|
| Added policies for data products, more details on labels and specified namespaces | 26/06/2018 | All | All |
| Author list and approval list modified | 29/06/2018 | metadata | metadata |
| Added guidelines for use of FITS and CDF in PDS4 | 25/07/2018 | sect. 11 | All |
| Added guidelines for Geometry metadata | 26/07/2018 | sect. 8 | All |
| Minor updates following internal review | 18/09/2018 | All | All |
| Corrected SPICE Bundle Section | 03/10/2018 | sect. 10 | All |

**Table of contents:**

# 1. INTRODUCTION

## 1.1 Purpose and Scope

This document compiles a set of guidelines, rules, common terminologies and conventions to be followed by all data providers when generating and submitting their science data and associated products to the Mission Science Ground Segment (SGS) for archiving within the Planetary Science Archive (PSA). This guide is also intended as a quick introduction to the PDS4 standards (fully described in [AD.01], [AD.02] and [AD.03]).

The rules and conventions defined in this document are intended to cover all needs at the level of the PSA overall. Specific rules and conventions defined at the level of each mission are provided as appendices to this document. As such, the conventions here-in are expected to remain stable and static for the most part. The dynamic aspects defined for each mission within the appendices will be edited and updated as needed to cover arising needs and requirements.

The use of PDS4 standard should be perceived as a tremendous benefit for the data producer in providing scientific products in a framework that will enhance its usability by the science community (see also RD.04). The benefits are at first for the Planetary Science Archive (PSA) as a multi-mission and multi-instrument archive, and also towards the international community since the PDS4 standard ensure:

- long term preservation of the data products
- easy ingestion of these data products into the PSA
- discovery of these data products through the PSA search interface
- cross mission search of these data products with other planetary missions both through the PSA and other data access protocols

## 1.2 Applicable Documents

The following documents, of the issue given hereunder, are pertinent to the extent specified herein and impose requirements to the SGS or the SGS schedule. They are referenced in the form [AD XX]:

[AD.01] PDS4 Standards Reference (SR)

[AD.02] PDS4 Data Dictionary (DDDB)

[AD.03] PDS4 Information Model Specification (IM)

## 1.3 Reference Documents

The following documents, of the issue given hereunder, although not part of this document, amplify or clarify its contents. If no issue given, the most recent issue should be used. They are referenced in the form [RD.XX]:

[RD.01] PDS4 Data Providers Handbook (DPH)

[RD.02] PDS4 Concepts

[RD.03] PSA Acronyms and Abbreviations

[RD.04] PSA Concept Document (not yet issued)

## 1.4 Abbreviations and Acronyms

See PSA online Acronyms and Abbreviations [RD.03].

# 2. OVERVIEW OF THE ARCHIVING PROCESS

This document is both aimed at data providers to the PSA and PSA users who wish to gain an insight into PDS4 data in the archive. The PDS4 data standard is defined at (http://pds.nasa.gov/pds4/) and while this defines file formats and data dictionaries in some detail, the organisation of the data can vary e.g. between NASA and ESA archives. The PSA has defined a common organisation for ESA missions to enable navigation and transparency.

The top level structure for all PDS4 data entities is a bundle. For archiving in the PSA, each mission is required to have a single bundle for each instrument, plus two separate “mission” bundles, one containing mission wide, non-instrument specific data such as spacecraft housekeeping data, and the other mission bundle contains SPICE files. These "mission" bundles plus the instrument bundles are typically sufficient to store all mission data, however in some cases, where the mission is split into separate vehicles, it may be more appropriate to put such data into separate "host" bundles. An example of this is the ExoMars 2020 RSP mission where the Rover and Surface Platform become, to a large extent, separate missions on the surface and in this case information will be generally stored at host level.

Each bundle is then subdivided into a set of collections. The available collection types are detailed in Chapter 3. Generally the PSA implementation aims to separate data types into an easy to navigate manner so an instrument bundle will contain separate collections for documentation, instrument science data, calibration products etc. For the instrument science data there are actually separate collections for each processing level and this is standardised by PDS4 to four levels: raw, partially processed, calibrated and derived.

The structure of some collections such as documentation are instrument specific so will vary between instruments and missions. However instrument science data does have a cross-mission substructure detailed in Chapter 3 and divisions into mission phase, sub-instrument etc. are defined as being cross-mission for the PSA. This bundle/collection structure can be considered as a set of folders which organise the actual data products of the instrument. When an archive user queries the archive and the results are displayed, this structure will not be immediately apparent. Instead, depending on the query, a set of products matching the query will be displayed. However once a set of products is selected for download, the downloaded entity will be organised as a set of folders matching the bundle structure i.e. as a bundle created on the fly.

The definition of products (both the file formats and the metadata) is tightly constrained by the PDS4 standards and data dictionary (see links in chapter 2). Each product contains at least two files, one for the metadata (always in an XML file) and the other(s) for data with different file types allowed. Rules regarding the science products are detailed in chapter 5 for product structures and chapter 6 for the metadata. Geometry is dealt with separately in chapter 7.

Products are allowed different versions and these are stored internally in the "version ID" commonly referred to as the "VID". Each product also contains a unique logical identifier "LID" which is attached to the version to create a "LIDVID". This identifier makes the product unique not just in the PSA, but in the entire set of federated PDS-compliant archives. The implementation of LIDVIDs used in the PSA is described in chapter 4.

## 2.1 PDS Standards

All science data delivered to the PSA must be compliant with the NASA’s Planetary Data System (PDS) standards. The applicable version of the standards is PDS4, with the precise version determined by each mission at the time of archive definition, and documented accordingly in the appendices. Changes to the applicable version of the PDS standards for any mission will be documented in the relevant appendix.

Further information about PDS4 can be found here: http://pds.nasa.gov/pds4/

## 2.2 PDS4 Terminology

The terminology described in this section is used in the PDS4 documentation to refer to the elements and structure of a PDS4 label. Note that, although PDS4 labels are XML-formatted files, the terminology used in PDS4 differs from the XML terminology.

Attribute:

```
<attribute>VALUE</attribute>
```

Class: Grouping of attributes and classes.

```
<Class>
 <attribute>VALUE</attribute>
 <attribute>VALUE</attribute>
 </Class>
```

## 2.3 PDS4 Requirements for PSA

The combination of the documents listed below give the set of common requirements that apply across PDS.

- The PDS4 Data Dictionary (DDDB) is the fundamental reference for definitions of classes and attributes (see PDS4 terminology in section 2.1.1).
- The PDS4 Information Model (IM) is the fundamental reference for PDS4 structure; its requirements can be validated automatically using eXtensible Markup Language (XML) schemas.
- The PDS4 Standards Reference (SR) is a compilation of policies, rules, and other PDS4 constraints that are not given explicitly in the previous references.

PSA has defined ***archive-specific requirements*** (highlighted in red in parts the present document including the next chapter) that further constrain — but do not conflict with — the common PDS4 requirements. These specific rules apply to all science data to be ingested and preserved in the Planetary Science Archive. Further constraints to the standards can be applied at mission level, and these are described in the relevant appendix for each mission.

### 2.3.1 PDS4 schemas

Schema (or Data Dictionary in the PDS context) and Schematron (rules) files are used to set rules on the layout and context of XML files plus the values this can take. Generally the schema files create a list of (metadata) "attributes" and define how these are organised into "classes". The core set of definitions for PDS4 is available on the PDS website and comprises the PDS4 dictionary and the rules associated with these attributes in the PDS4 Schematron. Together these are known as the PDS Information Model.

PDS is organised into a set of discipline nodes and most nodes also maintain a Discipline Dictionary which extends the core dictionary and provides classes and attributes for specialised areas such as atmospheric studies, imaging etc. When a PDS4 product is created the use of the core is mandatory and one or more discipline dictionaries are optional.

In addition, the PSA maintains a similar dictionary covering extensions for all missions providing PDS4 data to the PSA; rules and policies relating to the use of this dictionary are described below. The PSA dictionary can be found on the PSA public FTP site at ftp://psa.esac.esa.int/pub/mirror/PSA/psa_bundle/xml_schema.

Finally, a mission team is likely to define mission or instrument metadata which is specific to that mission or instrument. Usually as a minimum a mission level dictionary is required where all these data are stored. For missions with complex instruments the mission team may also elect to create separate instrument dictionaries. Both dictionary types are considered by PDS to be "mission level".

It is understood at PSA level that the creation and maintenance of dictionaries and Schematron is not a straightforward process and therefore, in general, the mission archive scientist will take responsibility of creating these for each instrument team. These dictionaries will then be used by the archive scientist to create template XML label files for the instrument teams to use for the production of their data products.

**Note for Data Providers Wishing to Create their own Dictionaries:** Please note that PDS are in the process of making the use of LDD Tool to create mission level dictionaries mandatory. As of PDS version 1.10, this tool lacks much of the required functionality and does not even recognise the registered PSA namespace, however any data provider planning to create their own dictionaries should use this tool and aid the PSA in providing feedback to PDS.

### 2.3.2 Namespaces

Namespaces are the mechanism by which dictionaries and rules files are located within PDS and by which naming conflicts are avoided. Each namespace must be registered within PDS and each URL must resolve. The registration and URL will be handled by the PSA team at ESAC in all cases where the PSA is the archiving authority.

The PSA namespace is:

| Namespace | URLs |
|---|---|
| http://psa.esa.int/psa/v1 | • http://psa.esa.int/psa/v1/PDS4_PSA_1000.xml<br>• http://psa.esa.int/psa/v1/PDS4_PSA_1000.xsd<br>• http://psa.esa.int/psa/v1/PDS4_PSA_1000.sch |

Mission namespaces must follow the pattern from this example of the ADRON instrument on the ExoMars 2020 Rover:

### Mission:

| | Namespace | URLs |
|---|---|---|
| Mission (emrsp) | http://psa.esa.int/psa/emrsp/v1 | • http://psa.esa.int/psa/emrsp/v1/PDS4_PSA_EMRSP_1000.xml<br>• http://psa.esa.int/psa/emrsp/v1/PDS4_PSA_EMRSP_1000.xsd<br>• http://psa.esa.int/psa/emrsp/v1/PDS4_PSA_EMRSP_1000.sch |
| Host (rm) | http://psa.esa.int/psa/emrsp/rm/v1/ | • http://psa.esa.int/psa/emrsp/rm/v1/PDS4_PSA_EMRSP_RM_1000.xml<br>• http://psa.esa.int/psa/emrsp/rm/v1/PDS4_PSA_EMRSP_RM_1000.xsd<br>• http://psa.esa.int/psa/emrsp/rm/v1/PDS4_PSA_EMRSP_RM_10 |

| | | |
|---|---|---|
| | | 00.sch |
| Instrument (ADRON) | http://psa.esa.int/psa/emrsp/rm/v1/ | • http://psa.esa.int/psa/emrsp/rm/v1/PDS4_PSA_EMRSP_RM_ADRON_1000.xml<br>• http://psa.esa.int/psa/emrsp/rm/v1/PDS4_PSA_EMRSP_RM_ADRON_1000.xsd<br>• http://psa.esa.int/psa/emrsp/rm/v1/PDS4_PSA_EMRSP_RM_ADRON_1000.sch |

Namespaces used are declared within each product and further details are given below. Note is it permissible, and indeed common at mission level, to have a dictionary (schema) but not a set of rules (Schematron). It is equally possible for a mission to use only Schematron rules, which can be applied to provide additional mission-specific rules on PDS and PSA dictionaries. For example, a mission Schematron could make the inclusion of an optional attribute in a discipline dictionary mandatory for a specific mission.

# 3. DATA ORGANISATION

This section describes the hierarchical structure (generic structure) that must be used to organise the science products resulting from any investigation/mission/instrument. For a detailed description of the data organisation, see section 2 of the PDS4 Std. Reference [AD.01]. The PSA recommended organisation, based on this generic structure, is described in Section 3.2, and should be follow where possible.

The highest level of organisation defined in PDS4 is called "bundle" (top-level directory). Each bundle is organised into a number of "collections" (sub-directories), each collection containing a set of closely related products based on their type and purpose. The types of collections allowed in PDS4 are listed in Table 1.

Products assigned to a collection must be organised into lower level sub-directories, following the most convenient criteria for each type of data.

| ***Collection Type*** | ***Description*** | ***Typical PDS4 Product Classes in this Collection Type*** |
|---|---|---|
| data | Contains primary products i.e. scientific data resulting from instrument observations. | Product_Observational |
| calibration | Contains calibration products used at any stage of the calibration process. | Product_Observational<br>Product_Document<br>Product_File_Text |
| document | Contains documentation and supporting information to assist in understanding and using the primary products. | Product_Document<br>Product_File_Text |
| geometry | Contains geometry products including pointing, orientation and positioning information typically provided in tables of calculated values. GIS shape files can also be considered for this collection, but discussions on an appropriate format are ongoing.<br>See further information in section 8.2. | Product_Observational<br>Product_Document |
| browse | Contains browse products including overview representations or quick-look plots/reports of the primary products.<br>See further information in section 8.2. | Product_Browse<br>Product_Thumbnail<br>Product_Document |
| miscellaneous | Contains miscellaneous products including any additional information and documentation products not easily classified as one of the other collections. | Product_Observational<br>Product_Document<br>Product_File_Text |
| context | Contains products that provide descriptions of the mission, spacecraft, instrumentation and targets. | Product_Context |
| xml_schema | Contains XML Schemas and related products used for generating and validating the products. | Product_XML_Schema |
| spice_kernels | Contains SPICE ancillary data files "kernels" products used for geometry computations. | Product_SPICE_Kernel |

Table 1: PDS4 Collection Types

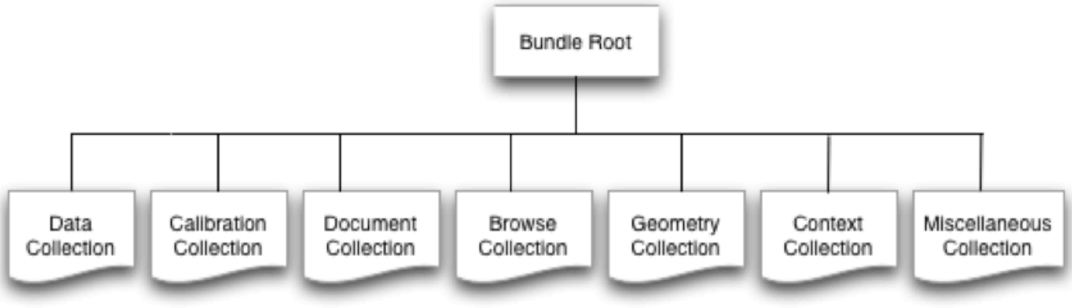


Figure 1 PDS4 bundle structure

## 3.1 Bundle Contents and Organisation Rules

The physical (on-disk) structure of a typical bundle (top-level directory) is described in this section and summarised in Table 2. Please note that this "Contents and Rules" discussion does not apply to the SPICE Bundle. For more information on the SPICE bundle Contents and Organisation see Section 9. Multiple collections of the same type are allowed in a bundle (unless specifically stated in Table 2 , see column “No. Of Occurrences”). If a bundle contains multiple collections of the same type, the name of the sub-directories must be formed by adding a suffix with a brief descriptive title after the collection type (i.e. <collection_type>_<title>, see collection types in Table 1 ). For example, if two calibration collections are included in a bundle, the sub-directories could be named: *calibration_ground* and *calibration_flight*. The corresponding mission annex typically provides additional guidelines for each mission.

- **bundle.xml** file (**required**): This is an inventory file that describes the bundle contents listing references to all the collections included in the bundle.
- **readme.txt** file (***required***): This file provides a general overview of the bundle contents and organisation in human readable format. This file is described in the bundle.xml label.
- **data** sub-directory (***required***): Contains primary products (main science data results). Products in a data collection are often organised into lower level sub-directories according to processing level, target, instrument mode, etc. and further split into ranges of orbits/dates, etc.
    - **collection_data.xml/.csv** files (required): Each data collection must contain an inventory file (table format) listing all the products inside the collection, with the corresponding XML label.
- **calibration** sub-directory (***required***): Contains calibration products used at any stage of the calibration process.
    - **collection_calibration.xml/csv** files (required): Calibration collection inventory file (table format) and corresponding XML label.
- **browse** sub-directory (optional): Contains browse representations (quick-look, thumbnails) of the primary products. Browse products are often organised into lower level sub-directories following the same criteria as for the primary products.
    - **collection_browse.xml/csv** files (required; if browse collection present): Browse collection inventory file (table format) and corresponding XML label.

- **document** sub-directory (***required***): Contains all documents (and related information e.g. figures, tables, etc.) deemed useful for understanding, interpreting and using other collections in the bundle.
  - **collection_document.xml/csv** files (required): Document collection inventory file (table format) and corresponding XML label.
- **geometry** sub-directory (optional): Contains geometry products describing the observing geometry of the primary products, including illumination, pointing, orientation and positioning information typically provided in tables of calculated values.
  - **collection_geometry.xml/csv** files (required; if geometry collection present): Geometry collection inventory file (table format) and corresponding XML label.
- **miscellaneous** sub-directory (optional): Contains any supplementary products that does not fit within the scope of the other collections.
  - **collection_miscellaneous.xml/csv** files (required; if miscellaneous collection present): Miscellaneous collection inventory file (table format) and corresponding XML label.
- **context** sub-directory (***required***): This collection contains a set of products describing the mission, the spacecraft, the instruments and any relevant target. All products in this collection must be included by reference to the mission's master context collection generated and managed by the relevant SGS. Therefore, no physical copies of the products need to be included in the bundle.
  - **collection_context.xml/csv** files (required): Context collection inventory file (table format) and corresponding XML label.
- **xml_schema** sub-directory (***required***): Contains all XML Schema and Schematron files referenced by the labels in the bundle used for generating and validating the products. All products in this collection must be included by reference to the mission's master context collection generated and managed by the relevant SGS. Therefore, no physical copies of the products need to be included in the bundle.
  - **collection_xml_schema.xml/csv** files (required): XML Schema collection inventory file (table format) and corresponding XML label.

**IMPORTANT:** Delivery of data may be in the form of either products or bundles. If the data provider is delivering bundles, then complete bundles are expected, including all of the bundle and collection inventory files (i.e. bundle.xml and collection_*.xml/csv). In the case of the delivery of products such as the ExoMars 2016 and BepiColombo missions, then only products are delivered to the PSA, and the bundle and collection inventory filed are created within the PSA.

**IMPORTANT:** Each mission archive will contain a unique "context" collection and "xml_schema" collection that will contain all context products and XML schemas used by the mission. These products must be included in each instrument bundle by reference (i.e. Instrument Teams do not need to provide these products when delivering the data).

Products inside the collections described above must reside in lower level sub-directories. Exceptions may be made for collections containing only a few products, in which case the products can be stored directly in the collection sub-directory.

### 3.1.1 Mandatory and Optional Collections

| ***Data Organisation*** | ***Number of Occurrences*** | ***Naming Convention*** |
|---|---|---|
| **<bundle_identifier>/** | | |

| | | |
|---|---|---|
| bundle.xml | 1 | fixed |
| readme.txt | 1 | fixed |
| **data**[_title] | 1..n | fixed prefix |
| collection_data[_title].xml/.csv | 1 | fixed prefix |
| **calibration**[_title] | 1..n | fixed prefix |
| collection_calibration[_title].xml/.csv | 1 | fixed prefix |
| **browse**[_title] | 0..n | fixed prefix |
| collection_browse[_title].xml/.csv | 1 | fixed prefix |
| **document**[_title] | 1..n | fixed prefix |
| collection_document[_title].xml/.csv | 1 | fixed prefix |
| **geometry**[_title] | 0..n | fixed prefix |
| collection_geometry[_title].xml/.csv | 1 | fixed prefix |
| **miscellaneous**[_title] | 0..n | fixed prefix |
| collection_miscellaneous[_title].xml/.csv | 1 | fixed prefix |
| **context (*)** | 1 | fixed |
| collection_context.xml/.csv | 1 | fixed |
| **xml_schema (*)** | 1 | fixed |
| collection_xml_schema.xml/.csv | 1 | fixed |

Table 2: Data organisation and content requirements of a PSA bundle

*(*) All products are secondary members i.e. products are added by reference, no physical copies of the products are required to be present in the bundle.*

## 3.2 Bundle Organisation Criteria

The following rules define the number of bundles and collections to be used by a mission to organise the data. Other organisations are possible and may be used in exceptional cases (e.g. more convenient structure for a specific type of data or instrument). Data organisation and naming conventions must be simple, intuitive and self-explanatory, and must facilitate manual browsing of the contents to the end-users.

**IMPORTANT:** This is a recommended standard that should be followed where possible. Exceptions should be discussed and agreed within a mission's DHAWG in coordination with PSA personnel.

### 3.2.1 Types of Bundles and associated Collections

| ***LID Formation Rule***<br>***(see section 4 )*** | ***File Name*** | ***Directory Name*** | **Bundle Entries (Collections)** |
|---|---|---|---|
| ***PSA Bundle*** | | | |
| urn:esa:psa | bundle_psa.xml | psa | • context |

| | | | |
|---|---|---|---|
| | | | • xml_schema |
| ***Mission Bundle*** | | | |
| urn:esa:psa:<mission_name> | bundle_<mission_acronym>.xml | <mission_name> | • context<br>• document<br>• miscellaneous_ancillary<br>• xml_schema |
| ***Host Bundle*** | | | |
| urn:esa:psa:<mission_name>_<host_name> | bundle_<mission_acronym>_<host_acronym>.xml | <host_name> | • context<br>• document<br>• miscellaneous_ancillary<br>• xml_schema |
| ***Instrument Bundle*** | | | |
| urn:esa:psa:<bundle_identifier> | <bundle_identifier>.xml | <bundle_identifier> | • data_telemetry<br>• data_raw<br>• data_partially_processed<br>• data_calibrated<br>• data_derived<br>• browse<br>• calibration<br>• geometry<br>• miscellaneous_ancillary<br>• document<br>• context (S)<br>• xml_schema (S) |
| ***SPICE Bundle*** | | | |
| urn:esa:psa:<mission_acronym>_spice | bundle_<mission_acronym>_spice.xml | <mission_acronym>_spice | • document<br>• miscellaneous<br>• document<br>• spice_kernels |

Table 3: Data organisation and content according to the type of bundle

*(S) This is a secondary member collection i.e. all products are added by reference, no physical copies of the products are required to be present in the bundle.*

Where:

<bundle_identifier> = <mission_acronym>[_<host_acronym>]_<instrument_id>

(see section 4.1 for details on the formation rule for bundle identifier)

### 3.2.2 Collections

| *LID Formation Rule* | *File Name* | *Directory Name* |
|---|---|---|
| urn:esa:psa:<bundle_identifier>:<collection_identifier> | collection_<collection_identifier>.<ext><br>where: <ext> = xml, csv | <collection_identifier> |

Table 4: Naming conventions for collections

Where:

<collection_identifier> = <collection_type>_[<collection_descriptor>]

(see section 4.2 for details on the formation rule for bundle identifier)

### 3.2.3 PSA Bundle Organisation Criteria

- One mission bundle
- One bundle per host, if required
- One SPICE bundle per mission
- One bundle per instrument (instrument bundle)
    - Data collections (primary data) within the instrument bundle are organised as follows:
        - One or more data collections for each processing level (named e.g. data_raw, data_calibrated)
        - Products inside each "data" collection must be organised in sub-directories according to the following criteria:
            - mission phase
                - sub-unit or sensor (when applicable)
                - other criteria e.g. experiment type, observation type, instrument mode or purpose of the data

Example:

data_<processing_level>/<mission_phase>/

- [<sensor>]
    - [ experiment type or observation type ]
        - Range of orbits (e.g. ORBIT_00100_00199)
        - Range of days (e.g. YYYYMMDD_YYYYMMDD)

- Supplementary collections (i.e. browse, calibration, document, geometry, miscellaneous, etc.) are organised in sub-directories according to the purpose of their products. Browse, calibration and geometry collections where products are one-to-one related to the primary products (in a data collection) must be organised following the same directory structure followed by the data collection.

- There is no restriction on the number of bundles and collections that can be delivered, with the exception of the rules defined in section 3.1.
- Data organisation and naming conventions must be simple, intuitive and self-explanatory, and must facilitate manual browsing of the contents to the end-users. Directory and file naming conventions applicable to each mission are as defined in the appendices to this document.

## 3.3 Context Products

Context products describe the mission, the spacecraft, the instruments and any relevant target. Context products are placed in the mission's bundle context collection, generated and managed by the relevant SGS. These products will be included by reference in the instrument bundles i.e. no physical copies of the products are required to be included in the bundle. The different types of context products are described in the table below, with naming conventions for the logical identifiers and filenames.

| *Type* | *LID / Filename* | *Context Area* | *Reference Type* |
|---|---|---|---|
| **Agency** | urn:nasa:pds:context:agency:agency.esa<br>PDS4_agency_ESA.xml | | is_agency |
| **Investigation** | urn:nasa:pds:context:investigation:mission.<mission_id><br>PDS4_mission_<mission_name>.xml | Investigation_Area | is_investigation |
| **Instrument Host** | urn:nasa:pds:context:instrument_host:spacecraft.<instrument_host_id><br>PDS4_host_<instrument_host_id>.xml | Observing_System | is_instrument_host |
| **Instrument** | urn:esa:psa:context:instrument:<instrument_id>.<instrument_host_id><br>PDS4_inst_<instrument_id>__<instrument_host_id>.xml | Observing_System | is_instrument |
| **Target** | urn:nasa:pds:context:target:<target_type>.<target_name> (LID, if under PDS governance)<br>urn:esa:psa:context:target.<target_name> (LID, if under PSA governance)<br>PDS4_target_<target_name>.xml | Target_Identification | data_to_target |
| **Personnel** | urn:esa:psa:context:personnel:personnel.<person_id><br>PDS4_personnel_<person_id>.xml | | |
| **Resource** | urn:esa:psa:context:resource:resource.<resource_id><br>PDS4_resource_<resource_id>.xml | | |

Table 5: Context file naming conventions

### 3.3.1 Use of Target Context

Currently the use of Target Context files are optional within the PDS standard and their adoption by NASA missions is varied. For the PSA we aim, as a multi-mission archive, to have consistency in their use, hence the set of policies outlined below are adopted.

#### *3.3.1.1 General Rules*

- For missions studying one or more bodies in the Solar System that body should by default be the target for all products. For example JUICE will have observations at Jupiter and at Ganymede. When at Jupiter all observations of Jupiter should have target Jupiter and when at Ganymede, the target is Ganymede.
- If the body is not in the PDS target list, a request will be made to PDS to include this and a temporary measure will be made of adding the target to the PSA bundle.
- Any PSA targets adopted by PDS will be deprecated.
- The PSA will not allow duplicate target files. Duplicates will be resolved via use of an alias list within the context product.
- If an occultation observation is being made the target should be the body the spacecraft is pointing at, for example solar occultation measurements made to study the atmosphere of Mars, the target should be Sun.
- For landers and rovers, the PDS target should almost always be the body the rover or lander has landed on.
- For Rovers, where local targets are required e.g. named rocks or sample numbers, a local target context will be produced and stored in the appropriate mission or host bundle. This level must be used for all cross-instrument targets. The use of local target context files for local targets identified by these types of experiments is mandatory for missions in the PSA.
- A second level of local targets is permissible at instrument level and these will be stored in the instrument bundle. For example the ExoMars Rover close up imager instrument will image a small part of the Martian surface identified by a local mission target. However features within this image may be labelled as local targets at the instrument level.
- For new generic targets e.g. we may add Plasma, names will be agreed by the PDS4 Archive Scientist team before adoption.

For information:

- PDS Target Context Products are available at: https://starbase.jpl.nasa.gov/pds4/context-pds4/target
    - And mirrored at PSA under: ftp ://psa.esac.esa.int/pub/mirror/pds/context/target
- PSA Target Context Products (to be deprecated if adopted by PDS): ftp://psa.esac.esa.int/pub/mirror/PSA/psa_bundle/context/target

#### *3.3.1.2 Solar System Bodies Naming Conventions*

- The Sun exists as a target within PDS.
- PDS also contains context files for all Solar System planets, these just use their name.
- All solar system dwarf planet context files should use number_name.
- Pluto is the allowed exception where only the name is used.
- Currently only 136108_Haumea and 136472_Makemake are hosted by PDS.
- Currently PDS has duplication in satellite target files with both the satellite name and number in one file and just the name in the other e.g. PDS4_target_M1_PHOBOS and PDS4_target_PHOBOS.
    - If available the file with the name and number should be used by the PSA missions as only the number is unambiguous.

  - The exception to this rule is the Moon, where the file only contains the name.
  - Unassigned satellites may use their discovery designation until numbered, once numbered the context file should be deprecated and replaced by the number file.
  - The discovery designation should be retained as an alias.
- Jupiter Saturn Neptune and Uranus all have generic Rings target context files in PDS in form PDS_target_PlanetInitial_RINGS, these should be used for Rings observations in the PSA.
- Similarly all asteroid files should be named PDS_target_number_asteroid. Currently there are no asteroids in the PDS4 list, so for now all files will be local
- *No policy is yet in place for comets*

#### *3.3.1.3 Calibration Targets*

Observations of dark sky should use the PDS_target_DARK_SKY. Hence the following, similarly names files should not be used:

- Deep Space
- Black Sky
- Space
- Skyflat
- Sky
- Flatfield
- Bias

As much as possible, specific targets relating to internal calibration sources should be avoided, instead one of the following are encouraged:

- Cal_Target – to be used for internal calibration targets
- Internal_Source – to be used for all illumination sources
- Dark – internal darks
- Rover
- Lander
- Spacecraft_deck

The following PDS files should not be used:

- Test_image
- Plaque
- Responsivity
- CalImage
- Calibration
- Cal_Lamps
- FF_Lamp
- Stim_Lamp

- Cal

When an observation is made where no target is specified the Non_Science target file should be used. This includes checkout observations hence the "Checkout" target should never be used.

Unknown targets are not allowed in the PSA, we will form a temporary name if necessary.

#### *3.3.1.4 Celestial Targets*

- All Celestial targets should have a target context file. If the file is not in PDS a file will be generated in the PSA bundle and a request will be made to PDS to include this.
- By convention calibration stars with local target context files in the PSA will be named via their HD number. *PDS is currently investigating using the SIMBAD name, but this name is often not in common use.*
- All celestial target files produced for the PSA will include an alias list with all alternate identifiers from SIMBAD included.
- For objects without an HD number the PSA will use the SIMBAD name.
- No convention is established for the naming of star clusters or Galaxies, it is expected for most the Messier identification will be adopted.
- For nebulae and fainter objects the NGC catalogue is preferred.
- All Celestial Targets should be named hence generic PDS files are excluded. Currently within PDS two such generic names exist, and should never be used in the PSA:
    - Starfield
    - OB_Star

#### *3.3.1.5 Other Targets*

The PSA allows the following alternative targets:

- Dust
- Interstellar_Particles
- Solar_Wind
- Io_Plasma_Torus

Additional alternative targets are likely to be added to this list. These will be added by agreement with the PDS4 AS team.

# 4. LOGICAL IDENTIFIERS

Every product (including bundles and collections) in PDS is assigned an identifier that allows it to be uniquely identified across the system. This identifier is referred to as Logical Identifier (LID). Logical identifiers are constructed as four fields (for a bundle), five fields (for a collection) or six fields (for a basic product), as defined below, with colons delimiting the fields.

| ***Logical Identifiers*** | |
|---|---|
| bundle | urn:esa:psa: <***bundle_identifier***> |
| collection | urn:esa:psa:<bundle_identifier>: <***collection_identifier***> |
| product | urn:esa:psa:<bundle_identifier>:<collection_identifier>: <***product_identifier***> |

Table 6: Logical Identifiers Formation Rules

Bundle, collection and product identifiers are formed according to the conventions described in the following sections with the exception of the SPICE Bundle, collection and product identifiers which are described in Section 9

A version identifier (VID) may be appended to the logical identifier (separated by a double colon i.e. ::<version_id>), allowing different versions of the same product to be referenced uniquely.

urn:esa:psa:<bundle_identifier>***::<version_id>***

urn:esa:psa:<bundle_identifier>:<collection_identifier>***::<version_id>***

urn:esa:psa:<bundle_identifier>:<collection_identifier>:<product_identifier>***::<version_ id>***

The combination of a logical identifier and a version identifier is known as Versioned Logical Identifier (LIDVID).

## 4.1 Bundle Identifiers Formation Rule

A unique logical identifier shall be assigned to every bundle generated by a mission. Bundle logical identifiers shall be based on the following definition:

urn:esa:psa:<bundle_identifier>

Where <bundle_identifier> shall take the form of :

<mission_id>[_<instrument_host_id>]_<instrument_id>[_<descriptor>]

where:

***<mission_id>*** is a fixed part used for all bundles delivered by a mission. The value to be used for each mission is defined in the relevant mission appendix. Where a mission comprises more than one spacecraft, the value should be of the form ***<mission_id>_<instrument_host_id>*** , for example **bc_mpo** for BepiColombo Mercury Planetary Orbiter;

***<instrument_id>*** is the instrument identifier (when applicable). Instrument identifiers are defined by each mission's SGS and listed in the appendices.

***<descriptor>*** , details on bundle types/structure and naming conventions are provided in the corresponding mission appendix

## 4.2 Collection Identifiers Formation Rule

A unique logical identifier shall be assigned to every collection generated by a mission. Collection logical identifiers shall be based on the following definition:

***<collection_type>[_<descriptor>]***

Where:

***<collection_type>*** is the collection type (see Table 1).

***<descriptor>*** , details on collection types/structure and naming conventions are provided in the corresponding mission appendix.

## 4.3 Product Identifiers Formation Rule

A unique logical identifier (LID) shall be assigned to every product generated by a mission. Logical identifiers are based on the following formation rule:

urn:esa:psa:<bundle_identifier>:<collection_identifier>:**<product_identifier>**::*<version_id>*

The **<product identifier>** is based on the following convention:

***<instrument_acronym>_[<processing_level>]_[<type>]_[<subunit>]_<descriptor>***

[ ] = optional fields

Where:

***<instrument_acronym>*** , see table further below

***<processing_level>:*** Allowed values: tm | raw | par | cal | der (mostly only applicable to primary products: data)

***<type>***

- For primary products, possible split is: hk | sc
- For supplementary products, specify type: calib | thumb | qla | geo | doc | misc (list controlled by SGS)

***<subunit>*** is the sensor / detector / sub-unit identifier (when applicable), see table further below

***<descriptor>:*** additional information e.g.

- observation id
- observation type
- measurement type
- start / stop date (<YYYYMMDDThhmmss> or <YYYYMMDDThhmmss>_<YYYYMMDDThhmmss>; UTC time of the first (and last) measurement in the data product)
- orbit number (<NNNNN> or <NNNNN>_<MMMMM>; orbit number of the first (and last) measurement in the data product, padded with zeroes)

**IMPORTANT:**

- PDS product labels will have extension ".xml"
- PDS product file extension will depend on the format of the data they contain (".tab", ".fit", ".cdf", ".img")

- Orbit and date may be applicable to certain types of products
- Version is inside the label, follows PDS4 versioning scheme described in section 4.4
- Product identifiers will be used as product filenames (except for context products)
- Logical identifiers shall be lowercase. This restriction does not apply to filenames, PDS4 is case insensitive for filenames.

## 4.4 Version Numbers

The PDS4 versioning scheme is defined in detail in the PDS4 Standards Reference [AD.01]. Rules specific to the implementation of the versioning scheme in PSA are listed below.

- Bundle, collection, and product version identifiers will be sequential, and of the form M.n (with M initialised to “1”, and n initialised to “0”)
- 0.n will be used until the first official delivery to the PSA (e.g. for test/sample data, during data preparation or archive development)
- Version identifiers will be incremented every time there is a delivery to the PSA.
- Version identifiers are not included in the product filename (i.e. the product filename does not change)

For **observational products** (and supplementary products e.g. geometry products, browse products) typically generated with a pipeline:

- M will be incremented for major changes/updates, n will be incremented for minor updates (following a criteria similar to the versioning of the data processing software)
- Version numbers will increment independently of the software version e.g. when using different input parameters / files.
- In order to keep the traceability of a given version of a product to the software version and the input parameters / files, this information will be included in the label of the product.
    - *Details of the required metadata for this purpose will be included in a future version of the document.*
- All versions of a product will be available in the archive for retrieval (i.e. no limitation on the number of versions to keep available to the users by default). If this becomes unmanageable for a specific mission / case, a limit will be defined and applied to that particular case.

For **derived products**, typically produced as a result of combining multiple products (e.g. maps, profiles), the traceability to the input products will be kept using a separate index file. This index file will contain at least the LIDVID (versioned logical identifier) of all input products. *Details of the content of this index file need further discussion, to be included in a future version of the document.*

For **bundles and collections**:

- Each time a product is updated / added to a collection, the applicable collection and bundle versions will be incremented. The current policy at PSA is to increase major version once a month and minor version once a week.
- Bundle and collection versions will not be exposed to the users through the user interface, but will be used by the SGS/SOC to track which product versions have been delivered in a given delivery package

- When the delivery package to the PSA contains only individual products (typical case for systematic deliveries), the corresponding bundle and collection products are produced by the “bundle generation” software, that will increment the version numbers accordingly
- A criteria for incrementing major/minor numbers shall be defined to avoid large numbers (e.g. increment M monthly, following the public release schedule)

# 5. PRODUCTS

A PDS4 product is composed of an XML-formatted label file with metadata and one or more files containing the actual data (organised into “objects” a.k.a. data structure extensions). For example, a planetary image, the histogram of its pixel values, a table with housekeeping information, and the descriptions of their structure could be organised as a product.

The terminology used in the PDS4 documentation to classify the different types of products is outlined in Table 7.

| ***PDS4 Product Classification*** | | ***PDS4 Product Class*** |
|---|---|---|
| *Basic* | *Observational* | Product_Observational |
| | *Supplementary* | Product_Browse |
| | | Product_Ancillary |
| | | Product_Thumbnail |
| | | Product_Document |
| | | Product_File_Text |
| | | Product_Context |
| | | Product_SPICE_Kernel |
| | | Product_XML_Schema |
| *Aggregate* | | Product_Bundle |
| | | Product_Collection |

Table 7: PDS4 Product Classification

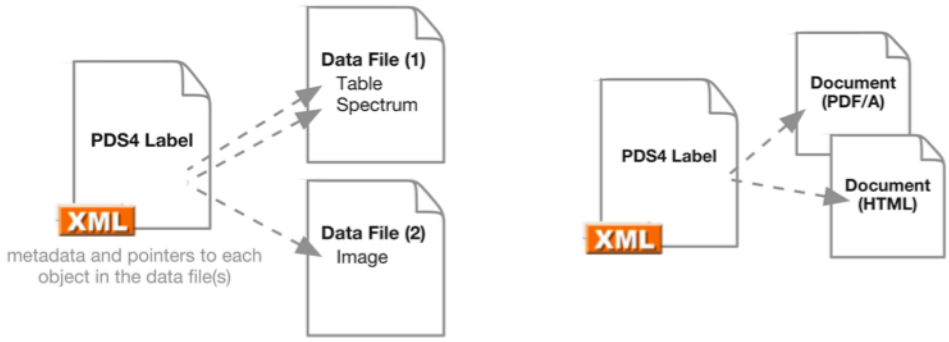


Figure 2: Illustration of a PDS4 Product_Observational and a PDS4 Product_Document

## 5.1 Fundamental Rules

This section provides a summary of the fundamental rules (see the PDS4 Data Provider's Handbook [RD.01]) that should be taken into account when creating PDS4 products. For a full description of the applicable rules, see the PDS4 Standards Reference [AD.01], section 4.

Objects must be formatted in one of the data structures listed in Table 7.

Each object must be contained in a single file (i.e. an object cannot be spread over multiple files).

A file may contain multiple objects if the objects belong to the same product (except for documents, in which case each object must be in a separate file).

Objects within a file are not required to use the same data structure (e.g. a product can be composed of a header, formatted as a parsable structure, and an image, formatted as an array structure, and both objects can be stored in the same data file).

When multiple objects are contained in a file, objects must be stored sequentially (i.e. one object shall be stored after the other, objects cannot be interleaved).

## 5.2 Data Structures

The following section (section 5.2.1) provides the list of the *data structures* (“*PDS4 extension classes”)* that are allowed in observational products and supplementary products.

These data structures are extensions of the four *basic data structures* (also known as “*PDS4 base classes*”, see Table 8). The basic data structures only define how to read the bytes from a file. They contain no interpretation beyond that. For the interpretation, PDS4 defines the extensions (see Table 9) to expand and restrict the associated metadata for more specific types.

| ***Basic Structure*** | ***Encoding Type*** | ***Description*** | ***Use*** |
|---|---|---|---|
| *Table Base* | Binary or Character (ASCII) | Fixed-length records (i.e. fixed width rows) of heterogeneous elements (different data type or size). Only 2-dimensional tables are permitted. Fields (i.e. columns) within each record are of fixed width and begin at fixed locations. Grouped fields are allowed (i.e. set of repeating fixed-width elements within one field). Note that ASCII character tables *must* use carriage-return-line-feed (<cr><lf>) line endings.<br>See section 4B in the PDS4 Std. Reference [AD.01].<br>Note that variable-width tables are described in PDS4 as a subclass of the “Parsable Byte Stream” (see Table_Delimited in Table 7). | Suitable for tabular data. |
| *Array Base* | Binary or Character (ASCII) | Fixed-width rows of homogeneous elements (same data type and size) in any number of dimensions.<br>See section 4A in the PDS4 Std. Reference [AD.01]. | Suitable for images, spectra, spectral cubes, maps. |
| *Parsable Byte Stream* | Binary or Character (ASCII) | Byte stream that can be parsed with standard rules (comma separated fields, standard punctuation). No decoding software is required.<br>See section 4C in the PDS4 Std. Reference [AD.01]. | Suitable for plain text, HTML, XML, tabular data with variable length fields and records (delimited text). |
| *Encoded Byte Stream* | Binary or Character (ASCII) | Byte stream that must be decoded by software before use. The use of this type of structure is restricted by PDS to a limited set of PDS approved external standards (e.g. PDF/A, JPEG, GIF).<br>See section 4D in the PDS4 Std. Reference [AD.01]. | Suitable for documents, thumbnails. ***This structure should not be used for observational products.*** |

Table 8: PDS4 Basic Data Structures

### 5.2.1 Data Structure Extensions

The PDS4 Information Model (IM) [AD.03] is the fundamental reference for the PDS4 data structures. Its requirements are specified in the form of PDS4 schemas.

| ***Product Type*** | ***PDS4 Product Class*** | ***Allowed PDS4 Extensions***<br>*(A.k.a. data structures or objects)* | ***Derived from***<br>*(See Table 8 )* |
|---|---|---|---|
| Observational | Product_Observational (1) | Array_1D (Deprecated) | Array Base |
| | | Array_2D_Image | |
| | | Array_2D_Image | |
| | | Array_2D_Map | |
| | | Array_2D_Spectrum | |
| | | Array_3D | |
| | | Array_3D_Image | |
| | | Array_3D_Movie | |
| | | Array_3D_Spectrum | |
| | | Table_Binary | Table Base |
| | | Table_Character | |
| | | Table_Delimited | Parsable Byte Stream |
| | | Stream_Text | |
| | | Header | |
| | | Encoded_Header | Encoded Byte Stream |
| Supplementary | Product_Browse | Any of the objects listed for Product_Observational, see above | |
| | | Encoded_Image | Encoded Byte Stream |
| | Product_Thumbnail | Encoded_Image | |
| | Product_Document | -- | Encoded Byte Stream |
| | Product_File_Text | Stream_Text | Parsable Byte Stream |
| | Product_Context | Agency, Facility, Instrument, Instrument_Host, Investigation, Resource, Target | |
| | Product_SPICE_Kernel | SPICE_Kernel | Parsable Byte Stream |
| | Product_XML_Schema | XML_Schema | Parsable Byte Stream |

(1) Observational products (Product_Observational) can include, in addition to the observational data, additional data files useful for the interpretation and analysis of the data. These additional files can use, in addition the objects listed under Product_Observational, the Encoded_Image data structure.

(2) Product_Bundle and Product_Collection (aggregate products) have not been included in this table. These products use the Stream_Text data structure.

Table 9: PDS4 Extensions allowed for each product type

**IMPORTANT:** Instrument Teams for each mission should consult with their SGS representatives to choose the most appropriate data structure(s) (see column “Allowed PDS4 Extensions” in Table 9) for their products. A combination of several data structures might be needed in some cases.

## 5.3 Data Types

Detailed information of the allowed data types can be found in the PDS4 Standards Reference [AD.01]. These data types are used to describe the format of the individual fields/elements of the data structures described in section 5.2.1.

Character data types (used in ASCII tables): see PDS4 Standards Reference [AD.01], section 5B.

Binary data types (used in binary tables and arrays): see PDS4 Standards Reference [AD.01], section 5C.

### 5.3.1 Date and Time Types

### 5.3.2 Date and Time Formats

For a complete list of PDS4 Date and Time Types and applicable rules, see PDS4 Standards Reference [AD.01]. The summary provided below includes additional requirements for PSA.

PDS date and time formats are based on the extended format of cardinal and ordinal date/time strings as defined in ISO 8601:2004. Specifically, all dates and/or times must be represented using one of the two formats below:

YYYY-MM-DDThh:mm:ss.ffffffZ (calendar format)

or

YYYY-DOYThh:mm:ss.ffffffZ (ordinal format)

Where:

**YYYY** is the 4-digit year

**MM** is the 2-digit month (possible values 01-12)

**DD** is the 2-digit day of month (possible values (01-31)

**DOY** is the 3-digit day of year (possible values 01-365, or 366 in a leap year)

**hh** is the 2-digit hour (possible values 00-23)

**mm** is the 2-digit minute (possible values (00-59)

**ss** is the 2-digit second (possible values 00-59, or 60 if needed for a leap second)

**ffffff** is decimal seconds (number of digits commensurate with precision)

**Z** denotes UTC (***required*** when the time is given in UTC)

### 5.3.3 Spacecraft Clock Formats

The Spacecraft Clock format is a native time format specific to each mission and is therefore defined in each mission-specific dictionary (see relevant mission annex). The Spacecraft Clock (SCLK) values represent the on-board time (OBT) counters of the spacecraft and instrument computers. The SCLK counter is given in the headers of the telemetry packets.

The SCLK typically contains the time as 32 bits of unit seconds followed by 16 bits of fractional seconds. And it is represented as a decimal real number in floating point notation with 5 digits after the decimal point. An integer number followed by a slash represents a reset of the spacecraft clock (e.g., “1/” or “2/”). The SCLK count values have the form:

n/dddddddddd:ddddd, where n is an integer number, up to 10 digits after the slash, and 5 digits after the decimal point.

### 5.3.4 OBT to UTC Time Conversion

UTC time is a function of the Time Correlation Packets and the on-board time (OBT). The Time Correlation Packets contains linear segments that map the on-board time to UTC time. The linear segment is represented by a time offset and a time gradient.

The conversion function is:

Time in UTC = offset + (OBT(seconds) + (OBT(fractional part) * 2-16)) * gradient

Time Correlation Packets are converted to a SPICE spacecraft clock kernel (SCLK) and distributed in the SPICE bundle. The SPICE spacecraft clock kernel (SCLK) can be used to convert the time stamps in the telemetry data (OBT in spacecraft clock format) to the corresponding UTC time.

The relevant mission-specific annex to this document will describe the specifics of OBT to UTC conversion as they apply to a given mission.

# 6. PSA POLICIES REGARDING DATA PROVISION

## 6.1 General Policies

Data Providers should always be aware that archival products have the purpose of serving the entire community therefore should be as user friendly as possible. The following statement is from the PDS Data Providers Handbook: "The general philosophy is that in order to preserve data for the long term, formats must be as simple as possible, well- described, and not reliant on specific software, because that software may be unavailable in the future."The following is the PDS Standard Policies:

*"All data must be provided in one or more of the following formats:*

1. *Fixed-width binary or ASCII tables that are composed of identically structured records*
2. *N-dimensional homogeneous arrays of binary elements (N<=16)*
3. *Variable-width ASCII tables composed of identically defined records, where the individual, variable-width fields in the record are delimited and the variable length records are also delimited*
4. *NAIF / SPICE kernel files."*

Furthermore the PSA requires that:

- Products are constructed as simply as possible, generally for each label there should only be a single product, any other implementation will require prior agreement.

- Where possible Table data products should use ASCII_Real or ASCII_Integer with fixed width and comma as delimiter, this ensures the data can be viewed easily.

Note for each processing stage more than one product may be supplied. Housekeeping data which have been converted into physical units can be considered either Partially_Processed or Calibrated data, according to the mission in question and the corresponding annex should be consulted to see which applies.

## 6.2 Raw Data

- The provision of Raw data to the PSA is mandatory.
- Raw data should respect the general policy, hence the TM needs to be parsed and re-formatted into readable tables, images, or other TBA formats as appropriate.
- In some cases a data provider may wish to supply a product which is essentially the TM still in the original Hex byte format but re-packaged into a PDS4 product. Such products will be allowed by the PSA but **only** if also accompanied by more easily parsable Raw products.

## 6.3 Partially Processed Data

- The provision of Partially_Processed science data to the PSA is optional.
- If instrument science data requires a conversion to physical units before further processing, the PSA strongly encourages archiving data at this level.
- Furthermore for instruments where part of the processing is strongly dependent on an algorithm selected by the operator, archival of data from this stage of processing is also

strongly encouraged. Examples of this would be the storage of interferograms from a Fourier Transform Spectrometer, and images from a camera before geometric calibration.

- Generally there is the expectation that calibration data will only be processed to this level.

## 6.4 Calibrated Data

- The provision of Calibrated data to the PSA is mandatory.
- As per the PDS standard Calibrated Products should have all instrument aspects removed, therefore typical examples of Calibrated Products would include spectra and non-mosaiced images.

## 6.5 Derived Data

- The provision of Derived data is optional, but strongly encouraged to enhance the scientific usability of the archive.
- Where clearly different derived products are supplied for an instrument, the possibility exists to create different derived collections. For example a spectrometer studying the atmosphere of a planet may supply both temperature profiles and maps of a particular species. Similarly a camera team may wish to supply stereo images, DTM and mosaics.

## 6.6 Browse Products

Browse Products should be provided according to the PDS policies:

- The recommended formats for the browse images in PSA are JPEG and PNG.
    - JPEG is recommended for pictures and PNG for plots.
- The browse image size is flexible and must ensure that enough details are visible.
    - anything with a size of less than 128 pixels should be considered as insufficient
    - no maximum size is given, but if the file size exceeds one or two megabytes the PSA should be consulted
- Note that browse images in JPEG format should use the file extension ".JPG".

Use (DPH 1.10.0, appendix B)

- Product_Browse
- Product_Thumbnail (image index)

Each Browse image must be accompanied by a label file. Browse products should link to science products using the LIDVID as an Internal_Reference. Science products must also link to browse products using Internal_References, the types are:

- data_to_browse
- browse_to_data

When more than one browse product applies, by PSA convention, the first reference to a browse product will become the primary browse product. This convention is necessary because no way of specifying a primary browse product is yet available in the PDS4 standard.

Browse Product PDS4 templates will be available in the PSA bundle.

**Table 1 Example of Browse Product label**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```
<?xml-model             href="http://pds.nasa.gov/pds4/pds/v1/PDS4_PDS_1800.sch"
schematypens="http://purl.oclc.org/dsdl/schematron"?>
<Product_Browse              xsi:schemaLocation="http://pds.nasa.gov/pds4/pds/v1
http://pds.nasa.gov/pds4/pds/v1/PDS4_PDS_1800.xsd      http://psa.esa.int/psa/v1
http://psa.esa.int/psa/v1/PDS4_PSA_1000.xsd"
xmlns="http://pds.nasa.gov/pds4/pds/v1"    xmlns:psa="http://psa.esa.int/psa/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

     <Identification_Area>

     <logical_identifier>urn:esa:psa:bc_mpo_mag:browse:mag_raw_sc_browse_201604
13t213630</logical_identifier>
     <version_id>1.0</version_id>
     <title>MPO-MAG    PDS    raw    science    data    browse    product</title>
     <information_model_version>1.8.0.0</information_model_version>
     <product_class>Product_Browse</product_class>
     <Modification_History>
            <Modification_Detail>
                <modification_date>2018-07-11</modification_date>
                <version_id>1.0</version_id>
                <description>First                         Version</description>
            </Modification_Detail>
     </Modification_History>
     </Identification_Area>

     <Reference_List>
        <Internal_Reference>

     <lid_reference>urn:esa:psa:bc_mpo_mag:data_raw:mag_raw_sc_20160413t213630<
/lid_reference>
            <reference_type>browse_to_data</reference_type>
        </Internal_Reference>
    </Reference_List>

     <File_Area_Browse>
        <File>
            <file_name>mag_raw_sc_browse_20160413T213630.png</file_name>
            <creation_date_time>2018-07-11T12:47:52.942Z</creation_date_time>
            <file_size                            unit="byte">334940</file_size>
            <md5_checksum>a0cdad54d3840fd9dd4e5a92fe2b4f9d</md5_checksum>
        </File>
        <Encoded_Image>
            <offset                                       unit="byte">0</offset>
            <encoding_standard_id>PNG</encoding_standard_id>
        </Encoded_Image>
    </File_Area_Browse>

</Product_Browse>
```

The following reference should be added to the applicable science product:

**Table 2 Example of Browse Product label**

```
<Reference_List>
    <Internal_Reference>

     <lid_reference>urn:esa:psa:bc_mpo_mag:browse:mpo_raw_sc_browse_20160413t21
3630</lid_reference>
        <reference_type>data_to_browse</reference_type>
    </Internal_Reference>
</Reference_List>
```

## 6.7 Calibration Data Products

Calibration data may be data files of derived parameters used as inputs to data processing, e.g. pixel number to wavelength, bias value vs gain etc. Calibration data may also be data taken periodically in-flight and used in a time based way for data processing. Examples of these data may be internal or external darks or flat fields, reference spectra of the Sun or other celestial objects etc.

For storage in the PSA three collections are allowed:

- calibration_files (for 'static' data processing files)
- calibration_raw
- calibration_partially_processed

Data taken via in-flight measurements must use the Product_Observational PDS class. Data files used for data processing, taken from ground-based measurements or derived from in-flight data should use the Product_Ancillary.
Any calibration observational product or file used as part of a data processing system for processing a particular calibrated data product must be referenced by that data product with the Processing_Inputs class (see below for more information on product labels). From PDS version 1.11, the Processing_Inputs class in the PDS mission dictionary should be used along with PSA schema version 1.1. For any PDS version prior to 1.11, the Processing_Inputs class from PSA schema 1.0 should be used.Note only observational data from the Raw and Partially_Processed collections will be visible in the PSA Filter Menu/Table View interface. The other data (coefficients, look up tables etc.) will be available through the FTP interface and via the download manager which will shortly be implemented, providing they are linked from appropriate products.

## 6.8 Geometry Products

Geometry information may be provided via products in the Geometry collection or as part of observational data products. A minimum set of geometry data to be included with all observational products is currently under consideration and specific policies regarding this will appear in future versions of this document.

# 7. PRODUCT LABELS

PDS labels are required for describing the contents and format of each individual product. In PDS4, all labels are detached (i.e. separate file) from the files containing the actual data. PDS4 labels are written in XML format. There is one PDS4 label (with file extension “xml”) for every product.

The following sections refer to the PDS4 standard from version 1.8 onward and are valid for all PDS4 data currently in the PSA. However as PDS4 is still a relatively new standard, some aspects are still undergoing change. At the time of this issue (2.1) the current version of PDS is 1.10. Where there have been changes since 1.8 or where changes are expected for the next version, the different cases will be treated separately in the appropriate section below. A full description of the changes between each version of the PDS4 standard can be found in the release notes of each Information Model release (see https://pds.jpl.nasa.gov/datastandards/documents/im/).

Figure 3 shows the general structure of a label, simplified to top-level classes that appear in a typical label of an observational product. Most of the top-level classes shown in this figure have nested classes. For detailed information in the structure and content of the PDS4 labels, see section 3 in the PDS4 Standards Reference [AD.01].

XML Declaration and Schema Reference Information

```
<Product Type>

<Identification Area>
<Alias_List>
<Citation_Information>
<Modification_History >

<Observation_Area>
<Time_Coordinates>
<Primary_Result_Summary>
<Investigation_Area>
<Observing_System>
<Target_Identification>

<Mission_Area>

<Discipline_Area>

<Reference_List>
<External_Reference>
<Internal_Reference>

<File_Area_Observational>
<File>
<Data Structure(s)>

<File_Area_Observational_Supplemental>
<File>
<Data Structure(s)>

</Product Type>
```

SourceURL:file:///E:/Downloads/ESDC-PSA-TN-0002%20PSA%20PDS4%20Archiving%20Guide-v184-20181004_1025.docx hereater

Figure 3: Simplified PDS4 label example

| ***Product Type*** | ***Top-level Classes*** | ***Reference*** |
|---|---|---|
| Product_Observational | Identification Area | See Product_Observational in the PDS4 |

| | | |
|---|---|---|
| | Observation_Area<br>Reference_List<br>File_Area_Observational<br>File_Area_Observational_Supplemental | Schema (XSD). |
| Product_Browse | Identification Area<br>Reference_List<br>File_Area_Browse | See Product_Browse in the PDS4 Schema (XSD). |
| Product_Thumbnail | Identification Area<br>Reference_List<br>File_Area_Encoded_Image | See Product_Thumbnail in the PDS4 Schema (XSD). |
| Product_Document | Identification Area<br>Context_Area<br>Reference_List<br>Document | See Product_Document in the PDS4 Schema (XSD). |
| Product_File_Text | Identification Area<br>Reference_List<br>File_Area_Text | See Product_File_Text in the PDS4 Schema (XSD). |
| Product_SPICE_Kernel | Identification Area<br>Context_Area<br>Reference_List<br>File_Area_SPICE_Kernel | See Product_SPICE_Kernel in the PDS4 Schema (XSD). |

Table 10: Top-level classes in PDS4 labels valid for PDS version 1.8 onwards

PDS4 labels must be compliant with:

- The class and attribute structures defined by the PDS, PSA and by the relevant mission in their respective XML Schemas (XSD files).
- The rules governing specific attributes and their values as set by the PDS, PSA and relevant mission in their respective XML Schematron files (SCH files).

## 7.1 Label Structure Requirements

Label structure requirements specific to the Planetary Science Archive and any mission archiving within the PSA (i.e. mandatory and optional attributes and classes, and rules restricting their values) are provided in the form of XML schemas (PSA PDS4 Schemas or <*Mission*> PDS4 Schemas).

*A summary of the requirements will be provided below classified by product type. This is under definition, to be completed in the next version of the document.*

### 7.1.1 XML Declaration

Each label starts with the XML Declaration and a declaration of the schematrons to be used.

The statement beginning with <?xml is an XML declaration. It means that the label is an XML versioned document. The statement beginning with <?xml-model is a processing instruction. It identifies the Schematron file against which the label is validated. A label may have more than one of these statements.

Note the xml-model statement does not allow multiple instances of the “href” statement. A separate statement should be used for each schematron specified e.g.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-model
href="http://pds.nasa.gov/pds4/pds/v1/PDS4_PDS_1A00.sch"
schematypens="http://purl.oclc.org/dsdl/schematron"?>
<?xml-model
href="http://pds.nasa.gov/pds4/geom/v1/PDS4_GEOM_1900_1510.sch"
schematypens="http://purl.oclc.org/dsdl/schematron"?>
```

Figure 4: Example XML declaration and declaration of schematrons

Each mission archive scientist will be able to supply a template file upon request.

### 7.1.2 Schema Reference Information

Next the Product class is opened and the schema locations and namespaces are declared.

```
<Product_Observational
xsi:schemaLocation ="http://pds.nasa.gov/pds4/pds/v1 http://pds.nasa.gov/pds4/pds/v1/PDS4_PDS_1800.xsd
http://psa.esa.int/psa/v1 http://psa.esa.int/psa/v1/PDS4_PSA_1000.xsd
http://psa.esa.int/psa/em16/cas/v1 http://psa.esa.int/psa/em16/cas/v1/PDS4_PSA_EM16_CAS_1000.xsd
http://pds.nasa.gov/pds4/geom/v1 http://pds.nasa.gov/pds4/geom/v1/PDS4_GEOM_1800_1401.xsd

" xmlns =" http://pds.nasa.gov/pds4/pds/v1 " xmlns:cas =" http://psa.esa.int/psa/em16/cas/v1 " xmlns:geom ="
http://pds.nasa.gov/pds4/geom/v1 " xmlns:psa =" http://psa.esa.int/psa/v1 " xmlns:xsi ="
http://www.w3.org/2001/XMLSchema-instance ">
```

Figure 5: Example of declaring schema locations and namespaces from an ExoMars 2016 CaSSIS Instrument data product.

All dictionaries used in the label must be declared, typically this will be the core PDS dictionary, any relevant discipline dictionaries such as the Geometry dictionary in the example above and any mission or instrument specific dictionaries.

### 7.1.3 Product Observational

The following table provides a list of the label requirements for an observational product.

| *Class* | *Parent Class* | *Required* | *Remarks* |
|---|---|---|---|
| ***Observation_Area*** | | | |
| Time_Coordinates | Observation_Area | Required | start_date_time and stop_date_time are mandatory, the class also allows some additional optional time coordinates. |
| Primary_Result_Summary | Observation_Area | Required | See next section |
| Investigation_Area | Observation_Area | Required | Mission information (investigation) shall follow the conventions in the corresponding mission appendix |
| Observing_System | Observation_Area | Required | Spacecraft and Instrument information (type and name) shall follow the conventions in the corresponding mission appendix |
| Mission_Area | Observation_Area | Required | See section 6.2.5 |
| Discipline_Area | Observation_Area | Required | See section 6.2.4 |

Table 11: Requirements for PDS4 labels of observational products

### 7.1.4 Primary_Result_Summary

This class is used to provide a high-level description of the type of product.

| *Element* | *Type* | *Required* | *No. Occurrences* | *Remarks* |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| purpose | Attribute | Required | 1 | See allowed values here. |
| processing_level | Attribute | Required | 1 | Indication of the level of processing that has been applied to the data. See allowed values here. |
| description | Attribute | Required | 1 | Brief description of the product. |
| Science_Facets | Class | Optional | 0..n | See definition and structure here. |

Table 12: Required attributes and classes in Primary Result Summary

## 7.2 Use of Classes and Attributes

### 7.2.1 The Mission Area

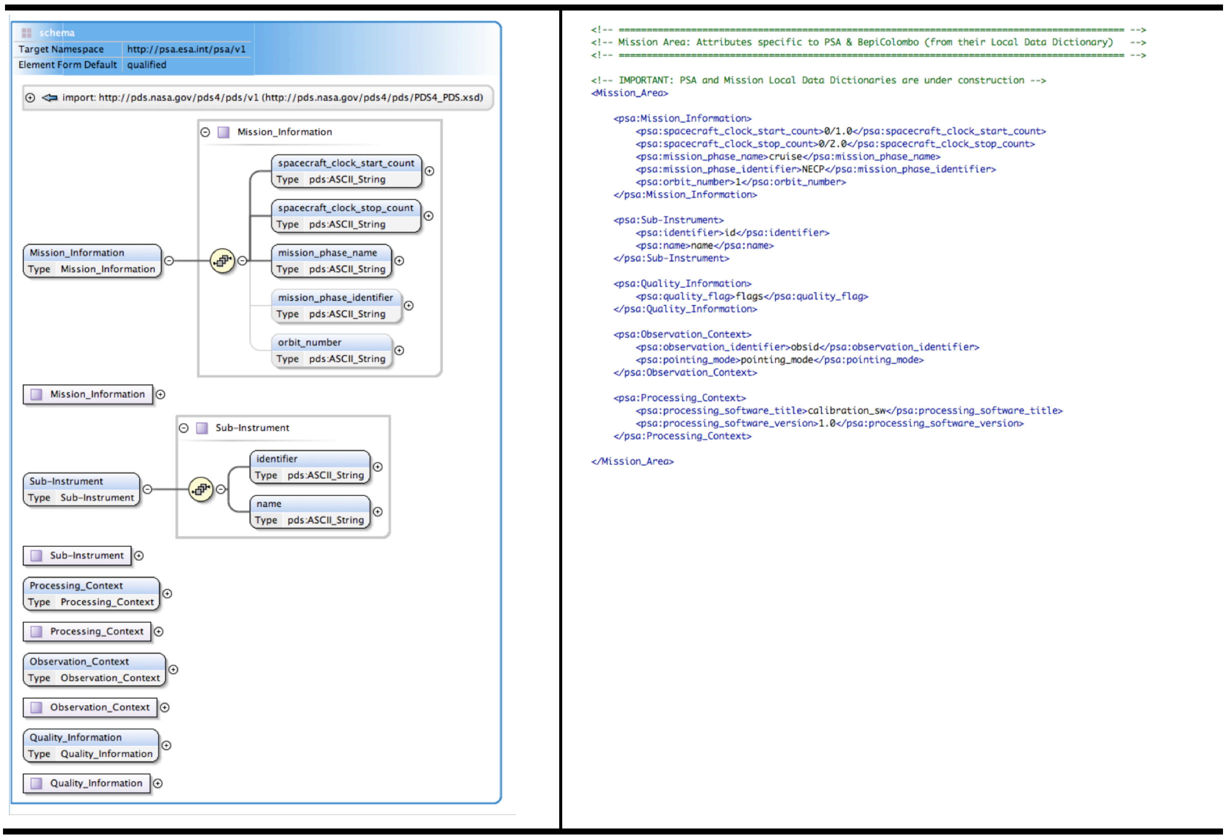
Mission specific classes and attributes must be included in the <Mission_Area> class of a PDS4 label, which is only available in the following product types:

- Product_Observational
- Product_Ancillary
- Product_Bundle
- Product_Collection
- Product_Document
- Product_Native
- Product_SPICE_Kernel
- Product_XML_Schema

It is generally excluded from other Product types such as Product_Browse as there is an expectation that internal references back to observational or other appropriate products will be used.

### 7.2.2 Use of the PSA Dictionary in the Mission Area

The PSA Dictionary hosts classes and attributes which are common to all missions. Therefore it is considered useful in a multi-mission archive to have these overriding set of attributes applied uniformly in the archive, simplifying the architecture.


schema
Target Namespace http://psa.esa.int/psa/v1
Element Form Default qualified
import: http://pds.nasa.gov/pds4/pds/v1 (http://pds.nasa.gov/pds4/pds/PDS4_PDS.xsd)
Mission_Information
spacecraft_clock_start_count Type pds:ASCII_String
spacecraft_clock_stop_count Type pds:ASCII_String
Mission_Information Type Mission_Information
mission_phase_name Type pds:ASCII_String
mission_phase_identifier Type pds:ASCII_String
orbit_number Type pds:ASCII_String
Mission_Information
Sub-Instrument
identifier Type pds:ASCII_String
Sub-Instrument Type Sub-Instrument
name Type pds:ASCII_String
Sub-Instrument
Processing_Context Type Processing_Context
Processing_Context
Observation_Context Type Observation_Context
Observation_Context
Quality_Information Type Quality_Information
Quality_Information

```
<!-- ===================================================================================== -->
<!-- Mission Area: Attributes specific to PSA & BepiColombo (from their Local Data Dictionary)   -->
<!-- ===================================================================================== -->

<!-- IMPORTANT: PSA and Mission Local Data Dictionaries are under construction -->
<Mission_Area>

    <psa:Mission_Information>
        <psa:spacecraft_clock_start_count>0/1.0</psa:spacecraft_clock_start_count>
        <psa:spacecraft_clock_stop_count>0/2.0</psa:spacecraft_clock_stop_count>
        <psa:mission_phase_name>cruise</psa:mission_phase_name>
        <psa:mission_phase_identifier>NECP</psa:mission_phase_identifier>
        <psa:orbit_number>1</psa:orbit_number>
    </psa:Mission_Information>

    <psa:Sub-Instrument>
        <psa:identifier>id</psa:identifier>
        <psa:name>name</psa:name>
    </psa:Sub-Instrument>

    <psa:Quality_Information>
        <psa:quality_flag>flags</psa:quality_flag>
    </psa:Quality_Information>

    <psa:Observation_Context>
        <psa:observation_identifier>obsid</psa:observation_identifier>
        <psa:pointing_mode>pointing_mode</psa:pointing_mode>
    </psa:Observation_Context>

    <psa:Processing_Context>
        <psa:processing_software_title>calibration_sw</psa:processing_software_title>
        <psa:processing_software_version>1.0</psa:processing_software_version>
    </psa:Processing_Context>

</Mission_Area>
```

Figure 6: PSA PDS4 dictionary (Mission Area) and an example of a <Mission_Area> section in a PDS4 label

The following policies apply to the use of the PSA dictionary:

- The Mission_Area is required for all for Product Observational products
- Mission_Information:
    - **mission_phase_name** is mandatory. "Not Applicable" may be used with agreement of the mission Archive Scientist e.g. ground based supporting data obtained in parallel with flight data. Validation rules are be added by each mission under the responsibility of the Archive Scientist.
    - **sub-instrument_type** The PSA has adopted the IPDA instrument type list and the value will be validated against the list below. Note the instrument_type attribute in the PDS instrument class must adhere to the PDS validation rule (see below). This attribute is optional but its use is strongly encouraged where applicable.
    - **spacecraft_clock_start/stop_count** Both the UTC time and Spacecraft Clock count should be included in the labels. The spacecraft clock count is from the PSA Mission Information class and the UTC is in the PDS Time_Coordinates class.
- Observation_Context (not mandatory for PSA but may be enforced at mission level)
    - **instrument_pointing_mode**
        - the following values are allowed: Surface, Atmosphere, Limb, No pointing, Space
            - if you feel these values do not adequately describe your data, contact the relevant mission archive scientist
    - **instrument_pointing_description**

        - mandatory if instrument_pointing_mode is included
    - **observation_identifier**
        - this field is used to record the identifier of the planned observation generating the labelled data
            - consult the mission archive scientist for the format and source of this ID.

### 7.2.3 Use of Mission Dictionaries

Mission specific classes and attributes are defined by the by the relevant Science Ground Segment, according to the mission needs, in XML Schema files that are referred to as Mission dictionaries. Individual Mission teams may elect to The Mission specific needs are defined in the appendices of this document.

Mission specific classes will include e.g. spacecraft clock start/stop time stamps, mission phase, orbit number, observation context information, data quality flags, processing context information (including processing SW version, input files/parameters, any other relevant processing information).

### 7.2.4 Use of Discipline Specific Classes and Attributes

The Discipline Area of a label is used to add attributes from the various PDS discipline dictionaries, which can be downloaded from here.

- Product_Observational
- Product_Context
- Product_Ancillary
- Product_Bundle
- Product_Collection
- Product_Document
- Product_Native
- Product_SPICE_Kernel
- Product_XML_Schema

### 7.2.5 Upcoming Changes to PDS and impact on the PSA

The success of the PSA dictionary had led to PDS creating the MSN (Mission) dictionary at discipline level and we aim to have data providers transitioning to using this dictionary from PDS version 1.11 onward. This will depend on whether the MSN LDD is stable by then and contains all PSA needed attributes (e.g. orbit currently not included).

PDS is currently undergoing a study on a change to instrument_type, currently this is validated against a core list but may move to implementing a new attribute instrument_type_list where a discipline level list may be used. If this goes ahead in version 1.11 the PSA will adopt the IPDA list and make it's use mandatory

Also currently under study is the LID formation rules for Targets, specifically within Target Context files. These changes mostly affect argets with primary/secondary bodies.

PDS Version 1.8, 1.9 and 1.10:

- Labels must use PSA schema version 1.0
- Data must reference the Mission_Information class in the PSA dictionary

PDS Version 1.11 onwards:

- Labels must reference PSA schema version 1.1

- Data must reference the Mission_Information class in the MSN dictionary
- Instrument_type may be replaced by instrument_type_list, if this is implemented the IPDA list must be used
- References to Target LIDs may need to be updated
- A PSA validation rule will be added to PSA schema version 1.1 which will check that the PDS attribute target_name matches the Target LID referenced, for example if the target_name attribute is "Mars" the referenced context file should not be the one for the Sun

- Add note to mention that as of PSA LDD 1.1 (compatible with PDS4 IM 1.10.1.0), Processing_Context:Processing_Inputs will be moved to the PDS4 IM (see applicable issue, CCB-177)
    - Reference_List was updated by adding two new associated classes Source_Product_Internal and Source_Product_External.

## 7.3 Attributes Used For the PSA Filter Menu

### 7.3.1 Instrument_Type

The instrument_type attribute does not appear in Product Observational as it can only be referenced from the Context_Area class. Therefore each Product Observational produced by an instrument must have an Observing _System_Component:Instrument with a LID reference back to the appropriate instrument context file. The PSA will then use this to reference link the instrument type to the observational data. For other products, which have a Context_Area, the instrument_type attribute can be used directly. Note as stated above the use of the PDS instrument_type attribute may change in version 1.11.

For Non-Instrument data such as spacecraft HK the observing system component = Instrument should be omitted and only the mission, and if appropriate, host should be used.

The IPDA instrument type list, currently only available to the PSA Sub-Instrument class, has been agreed by the IPDA members but has yet not been adopted by PDS. Therefore this list can only be used for a "mission level" attribute. The PSA uses this list to validate sub-instrument_type and this attribute must conform to one of these values:

| **In-Situ and Radio** | Remote sensing-like | Laboratory-like | Others |
|---|---|---|---|
| • Radio science | • Imager | • Microscope | • Seismometer |
| • Radar | • Imaging spectrometer | • Surface tool | • Interferometer |
| • Magnetometer | • Spectrometer | • Sub-surface tool | • Gravimeter |
| • Radiometer | • Mass spectrometer | • Biology experiment | • Spectrograph |
| • Particle analyser | • Neutron detector | • Weather station | • SPICE kernels |
| • Dust analyser | • Gamma ray detector | • Gas analyser | • Compilation |
| • Relaxation Sounder | • Altimeter | • Sensor suite | • Accelerometer |
| • Mutual Impedance Probe | • Polarimeter | • Chemistry laboratory | • Spacecraft sensors |
| • Radio Receiver | | • Temperature sensor | • Microphone |
| • Spectrum Analyser | | | • N/A |
| • Electric field instrument | | | |
| • Langmuir probe | | | |

### 7.3.2 Wavelength and the Science Facets Class

Although the use of wavelength_range in the Observation_Area:Primary_Result_Summary:Science_Facets is optional in PDS, it is mandatory for observational data in the PSA. Currently, as of PDS version 1.10 the following values are allowed:

- Far Infrared
- Gamma Ray
- Infrared
- Microwave
- Millimeter
- Near Infrared
- Radio
- Submillimeter
- Ultraviolet
- Visible
- X-ray

The PSA will also allow "Not_Applicable" and has raised an issue with PDS to get "Not_Applicable" added. The mandatory use of the wavelength attribute implies the mandatory use of the Science_Facets class in the Primary_Result_Summary. When this class is instantiated the discipline_name attribute must also appear. The allowed list for this attribute is currently:

- Atmospheres
- Fields
- Flux Measurements
- Geosciences
- Imaging
- Particles
- Radio Science
- Ring-Moon
- Systems
- Small Bodies
- Spectroscopy

Given that this list is a combination of names of actual disciplines with names of PDS nodes the PSA does not consider this a useful attribute and it is not used. Data Providers are free to put the disciple that best matches the data. Note the domain_name which is optional is Science_Facets is also not yet being used by PSA however data_providers are encouraged to populate this attribute as it is under consideration for future use. The current list of domain names are:

- Atmosphere
- Dynamics
- Heliosphere
- Interior

- Interstellar
- Ionosphere
- Magnetosphere
- Rings
- Surface

# 8. GEOMETRY

Geometry information may be provided via products in the Geometry collection or as part of observational data products. A minimum set of Geometry data to be included with all observational products is currently under consideration. Below a preliminary list of the requirements and recommendations, to be consolidated in future versions of this document.

***Required*** : The source data and methods used for generating the geometry data must be documented in the corresponding Experiment-to-Archive ICD (EAICD), along with documenting the coordinate/reference system used for the geometry data. The information should include a description of the quality (i.e., accuracy) of the geometry values.

## 8.1 Geometry Metadata expected in PDS labels

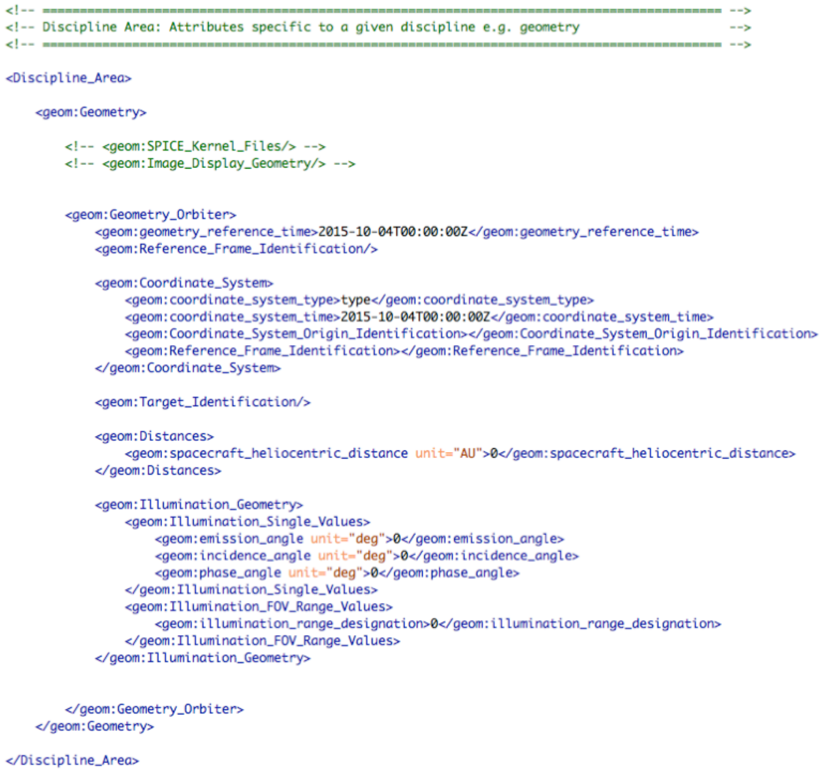
The rules on the layout and content of the geometry metadata in the PDS labels are defined in the PDS Geometry dictionary. Additional mission-specific guidelines can be found in the appropriate mission annex to this document.

The latest version of the PDS Geometry dictionary can be downloaded from the PDS4 website: https://pds.nasa.gov/pds4/schema/released/

As of PDS version 1.10, no official documentation (user guide) is available for the PDS Geometry dictionary. However, some guidelines can be found in the PDS Small Bodies Node (SBN) PDS4 Wiki, see relevant links below.

- Quick Introduction to the Geometry Dictionary
- Filling Out the Geometry Dictionary Classes

The table in section 8.1.3 includes a list of geometry attributes to be included in the label of the PDS products to be archived in the PSA, indicating whether the attribute is required, recommended or optional.



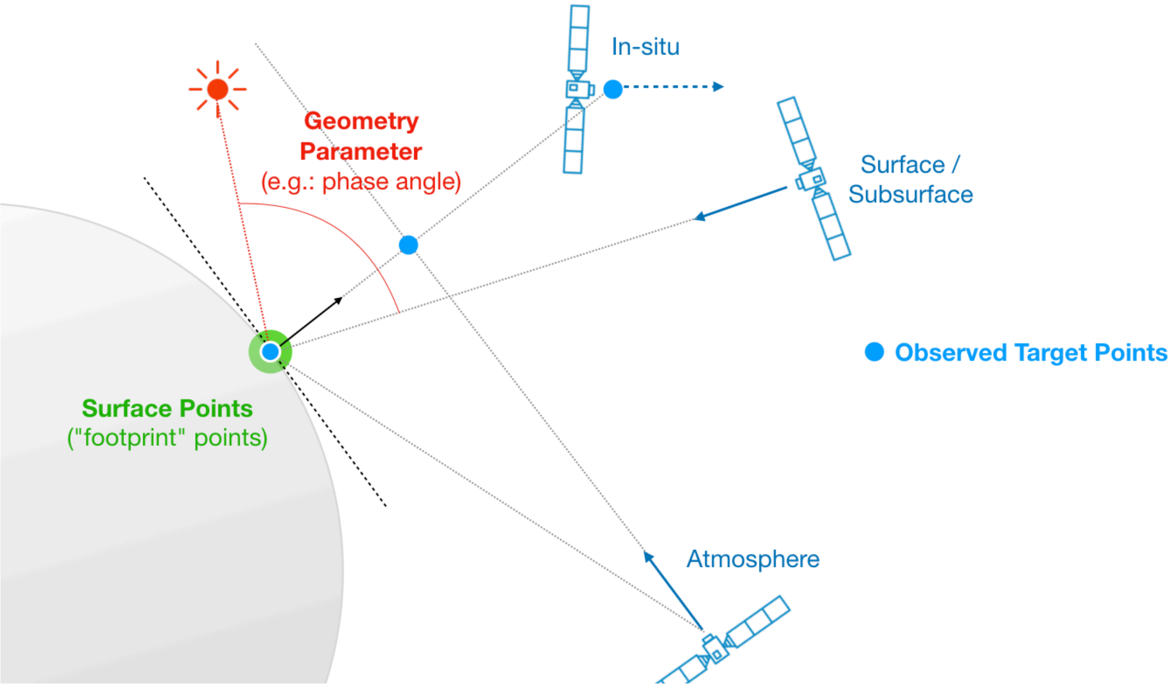
```
<!-- ========================================================================================= -->
<!-- Discipline Area: Attributes specific to a given discipline e.g. geometry                  -->
<!-- ========================================================================================= -->

<Discipline_Area>

    <geom:Geometry>

        <!-- <geom:SPICE_Kernel_Files/> -->
        <!-- <geom:Image_Display_Geometry/> -->


        <geom:Geometry_Orbiter>
            <geom:geometry_reference_time>2015-10-04T00:00:00Z</geom:geometry_reference_time>
            <geom:Reference_Frame_Identification/>

            <geom:Coordinate_System>
                <geom:coordinate_system_type>type</geom:coordinate_system_type>
                <geom:coordinate_system_time>2015-10-04T00:00:00Z</geom:coordinate_system_time>
                <geom:Coordinate_System_Origin_Identification></geom:Coordinate_System_Origin_Identification>
                <geom:Reference_Frame_Identification></geom:Reference_Frame_Identification>
            </geom:Coordinate_System>

            <geom:Target_Identification/>

            <geom:Distances>
                <geom:spacecraft_heliocentric_distance unit="AU">0</geom:spacecraft_heliocentric_distance>
            </geom:Distances>

            <geom:Illumination_Geometry>
                <geom:Illumination_Single_Values>
                    <geom:emission_angle unit="deg">0</geom:emission_angle>
                    <geom:incidence_angle unit="deg">0</geom:incidence_angle>
                    <geom:phase_angle unit="deg">0</geom:phase_angle>
                </geom:Illumination_Single_Values>
                <geom:Illumination_FOV_Range_Values>
                    <geom:illumination_range_designation>0</geom:illumination_range_designation>
                </geom:Illumination_FOV_Range_Values>
            </geom:Illumination_Geometry>


        </geom:Geometry_Orbiter>
    </geom:Geometry>

</Discipline_Area>
```

Figure 7: Example of geometry metadata in a PDS4 label

### 8.1.1 Surface point

As shown on the following figure, geometry parameters are computed at the *surface point* corresponding to the observed target point. Depending on the type of observation, the surface point would be:

- for surface observations, the surface point is the intersection between the line-of-sight and the target body surface;
- for atmosphere observations, the surface point is the nearest on the target body surface and the line-of-sight;
- for in-situ observations, the surface point is the sub-spacecraft point;



### 8.1.2 Coordinate system and units

- Longitudes and latitudes are expressed in planetocentric coordinate system.
- All angles are expressed in degrees.
- All distance are expressed in kilometres.

### 8.1.3 List of Geometry Attributes

The following table provides a set of geometry parameters currently being considered as guidelines for PDS4 labels. This list is provided as a guide only and the appropriate mission annex or the mission Archive Scientist should be consulted in case the use of a given parameter is unclear.

Names in bold are parameters used as inputs in the PSA user interface filter menu and entries in italics should be considered as "TBC".

| Name | PSA Requirement | Description | Comments | PDS4 attribute |
|---|---|---|---|---|
| SPICE Kernels | required | This class is used to cite the SPICE kernel files used in calculating the associated geometric values. | | Product_Observational: Observation_Area: Discipline_Area: Geometry: SPICE_Kernel_Files |
| Expanded Geometry | optional | Required when geometry is supplied as a separate | | Product_Observational: Observation_Area: Discipline_Area: Geometry: Expanded_Geometry |

| | | | | |
|---|---|---|---|---|
| | | geometry product. | | |
| *start_time* | | Earliest UTC time corresponding to the observation footprint | Different from product label START/STOP_TIME keywords values when there are several footprints for an observation (due to limb observation, not currently the case with geogen). | |
| *stop_time* | | Latest UTC time corresponding to the observation footprint | | |
| reference_time | required | Reference UTC time at which the geometry parameters are computed. | | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: geometry_reference_time |
| center_latitude | | Latitude of the observation footprint center point. | Planetocentric latitude of the pixel projected onto the target, in the range ±90° | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Surface_Geometry_Specific: Pixel_Intercept: pixel_latitude with reference_pixel_location = Center |
| center_longitude | | Longitude of the observation footprint center point. | Planetocentric longitude of the pixel projected onto the target, in the range 0-360° | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Surface_Geometry_Specific: Pixel_Intercept: pixel_longitude with reference_pixel_location = Center |
| **westernmost_longitude** | | Westernmost observation longitude of the footprint. | These give the extend / boundary box of a footprint and can be used to perform first-order geometry search, without using the actual footprint geometry. | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Surface_Geometry_Min_Max: minimum_longitude |
| **easternmost_longitude** | | Easternmost observation longitude of the footprint. | | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Surface_Geometry_Min_Max: maximum_longitude |
| **minimum_latitude** | | Minimum observation latitude of | | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: |

| | | | | |
|---|---|---|---|---|
| | | the footprint. | | Surface_Geometry_Min_Max: minimum_latitude |
| **maximum_latitude** | | Maximum observation latitude of the footprint. | | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Surface_Geometry_Min_Max: maximum_latitude |
| **local_true_solar_time** | | Local solar time for the surface point, evaluated at the reference time.<br>The local solar time is the angle between the planetocentric longitude of the Sun, as viewed from the center of the target body, and the planetocentric longitude of the surface point, expressed on a “24 hour” clock. | spice.et2lst() | Product_Observational: Observation_Area: Time_Coordinates: local_true_solar_time |
| solar_longitude | | Planetocentric longitude (Ls) of the sun for the target body at the reference time.<br>The planetocentric longitude is the angle between the body-sun vector at the time of interest and the body-sun vector at the vernal equinox. This provides a measure of season on a target body, | spice.lspcn(), corrected for light time and stellar aberration ('LT+S'). | Product_Observational: Observation_Area: Time_Coordinates: solar_longitude |

| | | | | |
|---|---|---|---|---|
| | | with values of 0 to 90 degrees representing northern spring, 90 to 180 degrees representing northern summer, 180 to 270 degrees representing northern autumn and 270 to 360 degrees representing northern winter. | | |
| sub_solar_latitude | | Latitude of the sub-solar point on the target body at the reference time.<br>The sub-solar point is the point on a body's reference surface where a line from the body center to the sun center intersects that surface. | spice.subslr() + spice.reclat(), corrected for light time and stellar aberration. | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Surface_Geometry_Specific: subsolar_latitude |
| sub_solar_longitude | | Longitude of the sub-solar point on the target body at the reference time .<br>The sub-solar point is the point on a body's reference surface where a line from the body center to the sun center intersects that surface. | spice.subslr() + spice.reclat() , corrected for light time and stellar aberration. | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Surface_Geometry_Specific: subsolar_longitude |
| **solar_distance** | | Distance | spice.spkpos() + | Product_Observational: |

| | | | | |
|---|---|---|---|---|
| | | from the center of the sun to the center of the target body at the reference time. | spice.vnorm(sunpos) | Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Distances_Specific: target_heliocentric_distance |
| spacecraft_solar_distance | | Distance from the spacecraft to the center of the sun at the reference time. | spice.spkpos() + spice.vnorm(sunpos-scpos) | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Distances_Specific: spacecraft_heliocentric_distance |
| spacecraft_altitude | | Distance from the spacecraft to the sub-spacecraft point on the target body at the reference time. | spice.subpnt() + spice.vnorm(srfvec=vector from observer to sub-observer point)<br>Candidate for min/max parameter | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Distances_Specific: spacecraft_target_subspacecraft_distance |
| **target_center_distance** | | Distance from the spacecraft to the center of the target body at the reference time. | spice.spkpos() + spice.vnorm(scpos)<br>Candidate for min/max parameter | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Distances_Specific: spacecraft_target_center_distance |
| sub_spacecraft_latitude | | Latitude of the sub-spacecraft point on the target body at the reference time . | spice.subpnt()+ spice.reclat()<br>When using a triaxial ellipsoid to model the surface of the target body, the sub-spacecraft point is defined as the nearest point on the target relative to the spacecraft.<br>The sub-spacecraft point is defined as the intercept, on the surface represented by the DSK data, of the line containing the spacecraft and the nearest point on the target's reference ellipsoid. If multiple such intercepts exist, the one closest to the observer is selected. | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Surface_Geometry_Specific: subspacecraft_latitude |
| sub_spacecraft_longitu | | Longitude of | | Product_Observational: |

| | | | | |
|---|---|---|---|---|
| de | | the sub-spacecraft point on the target body at the reference time. | | Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Surface_Geometry_Specific: subspacecraft_longitude |
| sub_spacecraft_solar_zenith_angle | | Solar zenith angle at the sub-spacecraft point on the target body surface at the reference time.<br>The solar zenith angle is the angle subtended between the direction towards the Sun and the local normal at the surface. | | |
| **target_right_ascension** | | Right ascension of the position vector of the target body center as seen from the spacecraft in the Earth mean equator and equinox frame (J2000). | spice.spkezr(target, et, 'J2000', 'LT+S', obsrvr)<br>spice.recrac() | Product_Observational: Observation_Area: Discipline_Area: Geometry: Image_Display_Geometry: Object_Orientation_RA_Dec: right_ascension_angle<br>If used, the Reference_Frame_Identification and the Geometry_Target_Identification classes should be included. |
| **target_declination** | | Declination of the position vector of the target body center as seen from the spacecraft in the Earth mean equator and equinox frame (J2000). | | Product_Observational: Observation_Area: Discipline_Area: Geometry: Image_Display_Geometry: Object_Orientation_RA_Dec: declination_angle<br>If used, the Reference_Frame_Identification and the Geometry_Target_Identification classes should be included. |
| sun_right_ascension | | Right ascension of the position | spice.spkezr('SUN', et, 'J2000', 'LT+S', | Product_Observational: Observation_Area: Discipline_Area: Geometry: Image_Display_Geometry: |

| | | | | |
|---|---|---|---|---|
| | | vector of the Sun as seen from the spacecraft in the Earth mean equator and equinox frame (J2000). | obsrvr)<br>spice.recrac() | Object_Orientation_RA_Dec: right_ascension_angle<br>If used, the Reference_Frame_Identification and the Geometry_Target_Identification classes should be included. |
| sun_declination | | Declination of the position vector of the Sun as seen from the spacecraft in the Earth mean equator and equinox frame (J2000). | | Product_Observational: Observation_Area: Discipline_Area: Geometry: Image_Display_Geometry: Object_Orientation_RA_Dec: declination_angle<br>If used, the Reference_Frame_Identification and the Geometry_Target_Identification classes should be included. |
| x_sc_sun_position | | X component of the position vector from spacecraft to Sun, expressed in J2000 coordinates, and corrected for light time and stellar aberration, evaluated at the reference time. | spice.spkezr('SUN', et, 'J2000', 'LT+S', obsrvr) | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Vectors: Vectors_Cartesian_Specific: Vector_Cartesian_Position_Sun_To_Spacecraft |
| y_sc_sun_position | | Y component of the position vector from spacecraft to Sun, expressed in J2000 coordinates, and corrected for light time and stellar aberration, evaluated at the reference time. | | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Vectors: Vectors_Cartesian_Specific: Vector_Cartesian_Position_Sun_To_Spacecraft |

| | | | | |
|---|---|---|---|---|
| z_sc_sun_position | | Z component of the position vector from spacecraft to Sun, expressed in J2000 coordinates, and corrected for light time and stellar aberration, evaluated at the reference time. | | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Vectors: Vectors_Cartesian_Specific: Vector_Cartesian_Position_Sun_To_Spacecraft |
| x_sc_sun_velocity | | X component of the velocity vector of Sun relative to the spacecraft, expressed in J2000 coordinates, and corrected for light time and stellar aberration, evaluated at the reference time. | | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Vectors: Vectors_Cartesian_Specific: Vector_Cartesian_Velocity_Spacecraft_Relative_To_Sun |
| y_sc_sun_velocity | | Y component of the velocity vector of Sun relative to the spacecraft, expressed in J2000 coordinates, and corrected for light time and stellar aberration , evaluated at the reference time. | | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Vectors: Vectors_Cartesian_Specific: Vector_Cartesian_Velocity_Spacecraft_Relative_To_Sun |
| z_sc_sun_velocity | | Z component of the | | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Vectors: |

| | | | | |
|---|---|---|---|---|
| | | velocity vector of Sun relative to the spacecraft, expressed in J2000 coordinates, and corrected for light time and stellar aberration , evaluated at the reference time. | | Vectors_Cartesian_Specific: Vector_Cartesian_Velocity_Spacecraft_Relative_To_Sun |
| x_sc_target_position | | X component of the position vector from the spacecraft to target body center, expressed in J2000 coordinates, and corrected for light time and stellar aberration, evaluated at the reference time. | spice.spkezr(target, et, 'J2000', 'LT+S', obsrvr) | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Vectors: Vectors_Cartesian_Specific: Vector_Cartesian_Position_Spacecraft_To_Target |
| y_sc_target_position | | Y component of the position vector from the spacecraft to target body center expressed in J2000 coordinates, and corrected for light time and stellar aberration, evaluated at the reference time. | | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Vectors: Vectors_Cartesian_Specific: Vector_Cartesian_Position_Spacecraft_To_Target |
| z_sc_target_position | | Z component of the | | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Vectors: |

| | | | | |
|---|---|---|---|---|
| | | position vector from the spacecraft to target body center expressed in J2000 coordinates, and corrected for light time and stellar aberration, evaluated at the reference time. | | Vectors_Cartesian_Specific: Vector_Cartesian_Position_Spacecraft_To_Target |
| x_sc_target_velocity | | X component of the velocity vector of the target body center relative to the spacecraft, expressed in J2000 coordinates, and corrected for light time and stellar aberration, evaluated at the reference time. | | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Vectors: Vectors_Cartesian_Specific: Vector_Cartesian_Velocity_Spacecraft_Relative_To_Target |
| y_sc_target_velocity | | Y component of the velocity vector of the target body center relative to the spacecraft, expressed in J2000 coordinates, and corrected for light time and stellar aberration , evaluated at the reference time . | | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Vectors: Vectors_Cartesian_Specific: Vector_Cartesian_Velocity_Spacecraft_Relative_To_Target |

| | | | | |
|---|---|---|---|---|
| z_sc_target_velocity | | Z component of the velocity vector of the target body center relative to the spacecraft, expressed in J2000 coordinates, and corrected for light time and stellar aberration , evaluated at the reference time . | | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Vectors: Vectors_Cartesian_Specific: Vector_Cartesian_Velocity_Spacecraft_Relative_To_Target |
| **min_incidence_angle** | | Minimum incidence angle.<br>The incidence angle is the angle between the local vertical at a given surface point and the vector from the surface point to the sun. | spice.ilumin()<br>When using a DSK as surface model for an atmosphere observation (line-of-sight not intersecting the surface), illumination angles are computed with respect to the DSK. Shouldn't it be computed with respect to the ellpsoid? | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Illumination_Geometry: minimum_incidence_angle |
| **max_incidence_angle** | | Maximum incidence angle.<br>The incidence angle is the angle between the local vertical at a given surface point and the vector from that the surface point to the sun. | | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Illumination_Geometry: Illumination_Min_Max: maximum_incidence_angle |
| **min_emission_angle** | | Minimum emission angle.<br>The emission | | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Illumination_Geometry: Illumination_Min_Max: |

| | | | | |
|---|---|---|---|---|
| | | angle is the angle between the surface normal at a given surface point and the vector from the surface point to the spacecraft. | | minimum_emission_angle |
| **max_emission_angle** | | Maximum emission angle.<br>The emission angle is the angle between the surface normal at a given surface point and the vector from the surface point to the spacecraft. | | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Illumination_Geometry: Illumination_Min_Max: maximum_emission_angle |
| **min_phase_angle** | | Minimum phase angle.<br>The phase angle is the angle between the vectors from the surface point to the spacecraft and from the surface point to the Sun. | | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Illumination_Geometry: Illumination_Min_Max: minimum_phase_angle |
| **max_phase_angle** | | Maximum phase angle.<br>The phase angle is the angle between the vectors from the surface point to the spacecraft and from the surface point to the Sun. | | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Illumination_Geometry: Illumination_Min_Max: maximum_phase_angle |

| | | | | |
|---|---|---|---|---|
| min_slant_distance | | Minimum slant distance.<br>The slant distance is the distance from the spacecraft to the nearest point on the detector line-of-sight to the target body surface. | *Surface:*<br>spoint, trgepc, srfvec, found = spice.sincpt(method, target, et, fixref, abcorr, obsrvr, dref, dvec)<br>spice.vnorm(srfvec)<br>*Atmosphere:*<br>trg_pnear, dist = spice.npedln(trg_radii[0], trg_radii[1], trg_radii[2], scpos, dvec)<br>los_pnear, dist = spice.nplnpt(scpos, dvec, trg_pnear)<br>slant = spice.vnorm(los_pnear-scpos)<br>*In-Situ:*<br>Should always be 0. | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Distances_Min_Max: minimum_spacecraft_target_boresight_intercept_distance |
| max_slant_distance | | Maximum slant distance.<br>The slant distance is the distance from the spacecraft to the nearest point on the detector line-of-sight to the target body surface. | | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Distances_Min_Max: maximum_spacecraft_target_boresight_intercept_distance |
| min_tangent_altitude | | Minimum tangent altitude.<br>The tangent altitude is the distance from the target body surface nearest point to the detector line-of-sight. | spice.npedln()<br>When using DSK as surface model, the ellipsoid is used anyway as surface model to derive the nearest point on the surface from the line-of-sight (no existing SPICE function to derive this, conversation with Nat Bachman). However, the tangent altitude and slant distance are computed based on the DSK. | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Distances_Min_Max: minimum_spacecraft_target_boresight_intercept_distance |
| max_tangent_altitude | | Maximum tangent | | Product_Observational: Observation_Area: Discipline_Area: |

| | | | | |
|---|---|---|---|---|
| | | altitude.<br>The tangent altitude is the distance from the target body surface nearest point to the detector line-of-sight. | | Geometry: Geometry_Orbiter: Distances_Min_Max: maximum_spacecraft_target_boresight_intercept_distance |
| horizontal_pixel_scale | | horizontal/vertical picture scale. Alternative: distance per horizontal/vertical resolution unit at the center of the observation (LoS). | | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Pixel_Dimensions: horizontal_pixel_field_of_view<br>with pixel_field_of_view_method = central pixel |
| vertical_pixel_scale | | | | Product_Observational: Observation_Area: Discipline_Area: Geometry: Geometry_Orbiter: Pixel_Dimensions: vertical_pixel_field_of_view<br>with pixel_field_of_view_method = central pixel |
| boresight_right_ascension | | Right ascension of the detector boresight vector, in the Earth mean equator and equinox frame (J2000), at the reference time. | mat_detec_j2000 = spice.pxform(self.detector.frame, 'J2000', et)<br>pvec = spice.mxv(mat_detec_j2000, dvec)<br>spice.recrad() | TBC |
| boresight_declination | | Declination of the detector boresight vector, in the Earth mean equator and equinox frame (J2000), at the reference | | TBC |

| | | | | |
|---|---|---|---|---|
| | | time. | | |
| boresight_target_angle | | The separation angle between the detector line-of-sight (boresight) and the target body center as seen from the spacecraft, at the reference time. | | TBC |
| boresight_solar_elongation | | Separation angle between the detector line-of-sight and the position vector of the Sun as seen from the spacecraft, at the reference time. | | TBC |

## 8.2 Geometry Products

Geometry products can be generated, and should be included in the Geometry collection.

Some observations may need to include separate geometry products including pointing, orientation and positioning information (e.g. pixel-by-pixel information for high-resolution cameras or hyper-spectral qubes), typically provided as tables or arrays of calculated values.

GIS shape files with the observation footprints (and some additional geometry information) can also be considered for this collection, but discussions on an appropriate format are ongoing.

*Specific guidelines will be added in a future version of this document.*

# 9. PERMISSIBLE VALUES

The permissible values described in this section are only those applicable to all data being delivered for archiving within the PSA. The more specific values that are applicable to each mission are included as appendices to this document. Each mission shall have their own appendix listing the specific values defined for attributes at the mission, instrument host, instrument and target level, plus any additional criteria applicable only to that mission. The PDS Small Bodies Node (SBN) provides a useful wiki page giving a quick reference for standard values in PDS labels - although this is not definitive, it is a useful reference. For a definitive list of allowed values, consult the Data Dictionary for the release of PDS4, or check the relevant Schema and Schematron files.

## 9.1 Data Processing Levels

Data processing levels to be used are defined in the table below, and are based on the PDS Policy on Data Processing Levels (see PDS4 Standards Reference [AD.01], section 10B).

| PDS4 Level Identifier | Definition |
|---|---|
| ***Telemetry*** | Original stream from an instrument encoded in telemetry packets. |
| ***Raw*** | Original data from an instrument. If compression, reformatting, packetization, or other translation has been applied to facilitate data transmission or storage, those processes will be reversed so that the archived data are in a PDS approved archive format. |
| ***Partially Processed*** | Data that have been processed beyond the raw stage but which have not yet reached calibrated status. |
| ***Calibrated*** | Data converted to physical units where all of the calibration based on the instrument has been done. |
| ***Derived*** | Results that have been distilled from one or more calibrated data products (for example, maps, gravity or magnetic fields, or ring particle size distributions).<br>Supplementary data, such as calibration tables or tables of viewing geometry, used to interpret observational data should also be classified as 'derived' data if not easily matched to one of the other three categories.<br>Geometry products and SPICE kernels will be classified as Derived. |

Table 13: Data processing levels

# 10. SPICE KERNELS

The SPICE (Spacecraft Planets Instrument C- Matrix Events) information system uses *ancillary data* to provide Solar System geometry information to scientists and engineers for planetary missions in order to plan and analyze scientific observations from space-born instruments . SPICE is used on all PSA supported missions to assist the science data processing, analysis and archiving activities. SPICE is developed by the NASA´s Navigation and Ancillary Information Facility (NAIF) and is implemented for the PSA supported missions by the ESA SPICE Service (ESS). More information is available at the ESS home page:

http://spice.esac.esa.int

The SPICE toolkit is available in FORTRAN, C at the NAIF web site (http://naif.jpl.nasa.gov). Interfaces to higher-level data analysis software in the Interactive Data Language (IDL), Matlab and JAVA are also provided. A Python wrapper approved by NAIF; SpiceyPy, is also available.

SPICE data is known as “kernels”. SPICE kernels are composed of navigation, orientation and other ancillary information structured and formatted for easy access. All SPICE kernels needed to compute the UTC time and geometric quantities for a mission must be archived along with the science data. Mission related SPICE kernels are generated mainly by the ESS (with the information provided by Flight Dynamics, the Science Ground Segment and by the Instrument Teams).

The archived SPICE kernels are peer-reviewed versions of the SPICE operational kernels datasets which are used in the study, development and operations of ESA Missions and which are available at the ESA SPICE FTP area:

ftp://spiftp.esac.esa.int/data/SPICE/ftp://spiftp.esac.esa.int/data/SPICE/<mission_name>/kernels/

In addition, both the SPICE operational kernels datasets and the archived SPICE bundles are mirrored in the NAIF server (https://naif.jpl.nasa.gov/pub/naif/).

A mission level bundle must be created containing all mission related SPICE kernels. The following SPICE kernel files might be included in each SPICE bundle:

- **SPK**: Spacecraft and natural bodies ephemeris files, also known as the planetary spacecraft ephemeris kernel file, including the spacecraft trajectory of reconstructed and predicted positions.
- **DSK:** Digital shape diles, include information of the digital shape of extended bodies (not always present)
- **CK**: Spacecraft orientation file, also known as the attitude c-kernel (CK) file.
- **SCLK**: Spacecraft clock file which contains the time correlation data to convert OBT to UTC.
- **FK**: Mission Reference frames file, also known as the frame kernel.
- **IK**: Instrument kernels.
- **PCK**: Planetary constants file (*.tpc), also known as the planetary constants kernel (PCK) file.
- **LSK**: NAIF leap seconds kernel.
- **MK**: Meta-kernel files, which include a list of the relevant kernels to setup a unique and unambiguous mission scenario.

Due to its unique nature the SPICE bundles have some particularities with respect to the rest of the bundles of a given mission. These particularities are described in the following sections.

## 10.1 Product types

SPICE Bundles will include products only of these types:

- Product_Bundle
- Product_Collection
- Product_SPICE_Kernel (under the "spice_kernelks" directory)
- Prouduct_Document (under the "document" directory)
- Product_Ancillary (under the "miscellaneous" directory)
- Product_Zipped (under the "miscellaneous directory, if needed")

## 10.2 Directory structure and File naming

SPICE_Kernels products follow a particular directory structure and file naming convention. The SPICE_kernel collection is composed by a series of directories that contain kernels of each kernel type. This replicates the directory structure of the ESA SPICE FTP and the canonical SPICE kernel dataset structure.

The products in the SPICE_kernels collection filename is –except for the meta-kernel files-composed by:

<ker_name>.<kernel_extension>

where:

- <ker_name>: is the kernel filename, in most cases very similar to the one which can be found in the ESA SPICE FTP area.
- <kernel_extension>: is the kernel extension. Typically, it consists of a first letter describing the kernel format: "b" for binary and "t" for text, followed by one or two letters that reference the kernel tye e.g.: "sp" for SPK, "c" for CK, "sc" for SCLK.

Note that the kernel filename <ker_name> always contains a version number which identifies the kernel as a unique file and that is considered part of the filename and not a version number per se. Hence in an archive it is common to find products with filename - LIDVID pair such as:

em16_tgo_v01.tf - urn:esa:psa:em16_spice:spice_kernels: em16_tgo_v0**1**.tf::**1**.0

em16_tgo_v02.tf - urn:esa:psa:em16_spice:spice_kernels: em16_tgo_v0**2**.tf::**1**.0

The directory structure and the products contained in each directory –including LIDVIDs – are described in the following table:

| **Data Organisation (bold) and**<br>**File name (italics)** | **N** [1] | **Product Type**<br>**LIDVID** |
|---|---|---|
| **<mission_acronym>_spice/** | 1 | |
| *bundle_<sc>_spice_v<ver>.xml* | 1..n | Product_Bundle<br>urn:esa:psa:<sc>_spice::<n>.0 |
| *readme.txt* | 1 | fixed |
| **document** | 1 | fixed |

| | | |
|---|---|---|
| *collection_document_v<ver>.xml*<br>*collection_document_inventory_v<ver>.csv* | 1..n | Product_Collection<br>urn:esa:psa:<sc>_spice:document::<n>.0 |
| *spiceds_v<ver>.xml/.html* | 1..n | Product_Document<br>urn:esa:psa:<sc>_spice:document:spiceds::<n>.0 |
| **miscellaneous** | 1 | fixed |
| *collection_miscellaneous_v<ver>.xml*<br>*collection_miscellaneous_inventory_v<ver>.csv* | 1..n | Product_Collection<br>urn:esa:psa:<sc>_spice:miscellaneous::<n>.0 |
| **spice_kernels** | 1 | fixed |
| *collection_spice_kernels_v<ver>.xml/.csv*<br>*collection_spice_kernels_inventory_v<ver>.csv* | 1..n | Product_Collection<br>urn:esa:psa:<sc>_spice:spice_kernels::<n>.0 |
| **ck** | 1 | |
| *<ker_name>.xml/.bc* | 1 | Product_SPICE_Kernel<br>urn:esa:psa:<sc>_spice:spice_kernels:ck_<filename.ext>::1.0 |
| **dsk** | 0,1 | |
| *<ker_name>.xml/.bds* | 1 | Product_SPICE_Kernel<br>urn:esa:psa:<sc>_spice:spice_kernels:dsk_<filename.ext>::1.0 |
| **fk** | 1 | |
| *<ker_name>.xml/.tf* | 1 | Product_SPICE_Kernel<br>urn:esa:psa:<sc>_spice:spice_kernels:fk_<filename.ext>::1.0 |
| **ik** | 1 | |
| <ker_name>.xml/.ti | 1 | Product_SPICE_Kernel<br>urn:esa:psa:<sc>_spice:spice_kernels:ik_<filename.ext>::1.0 |
| **lsk** | 1 | |
| <ker_name>.xml/.tls | 1 | Product_SPICE_Kernel<br>urn:esa:psa:<sc>_spice:spice_kernels:lsk_<filename.ext>::1.0 |
| **mk** | 1 | |
| <sc>_<des>_v<ver>.xml/.tm | 1..n | Product_SPICE_Kernel<br>urn:esa:psa:<sc>_spice:spice_kernels:mk_<sc>_<des>_<ver>::<n>.0 |
| **pck** | 1 | |
| <ker_name>.xml/.tpc/.bpc | 1 | Product_SPICE_Kernel<br>urn:esa:psa:<sc>_spice:spice_kernels:pck_<filename.ext>::1.0 |
| **sclk** | 1 | |
| <ker_name>.xml/.tsc | 1 | Product_SPICE_Kernel<br>urn:esa:psa:<sc>_spice:spice_kernels:sclk_<filename.ext>::1.0 |
| **spk** | 1 | |

| | | |
|---|---|---|
| <ker_name>.xml/.bsp | 1 | Product_SPICE_Kernel<br>urn:esa:psa:<sc>_spice:spice_kernels:spk_<filename.ext>::1.0 |

[1] Number of Occurrences

Table 14: Directory structure and products contained in each directory for SPICE archives

where:

- <sc> is the short mission name or acronym (e.g. em16, bc, etc.).
- <des> is a short description of the scope of a particular meta-kernel (scenario, mission phase, data type, instrument of applicability, etc.).
- <ver> is a three-digit incremental version number present in the file name.
- <n> is a one-digit incremental version number present in the VID, if the product filename includes <ver> then this is a one-digit version of <ver>.

Note that the miscellaneous collection can have other collections as needed containing Product_Ancillary described as Stream_Text, Table_Character of Encoded_Images. The document collection might contain other collections with Product_Document products. In addition a potential inclusion in the miscellaneous collection would be the checksum and orbnum data files for which the N and LIDVID would tentatively be as indicated in the table hereafter.

| **Data Organisation (bold) and File name (italics)** | **N [1]** | **Product Type<br>LIDVID** |
|---|---|---|
| **checksum** | **0,1** | |
| *checksum_v<ver>.xml/.tab* | 1..n | Product_Ancillary (described as Checksum_Manifest)<br>urn:esa:psa:<sc>_spice:miscellaneous:checksum_checksum::<n>.0 |
| **orbnum** | 0,1 | fixed |
| *<ker_name>.xml/.orb* | 1 | Product_Ancillary (described as Table_Character)<br>urn:esa:psa:<sc>_spice:miscellaneous: orbnum_<filename.ext>::1.0 |

## 10.3 LID/LIDVID Construction Rules

The LID/LIDVID construction rules, as it can be derived from the previous section are also particular for the SPICE bundles. The initial part of the LIDs for ESA missions will be “urn:esa:psa:<sc>_spice:" where <sc> is the mission acronym (e.g. em16, bc, etc.), e.g.:

urn:esa:psa:em16_spice:

There are four methods of constructing LIDVIDs depending on the product.

### 10.3.1 Path and full filename in LID

The first method ignores the version number referenced in the data filename (LID) for the VID construction and applies to following products:

- SPICE kernels under **spice_kernels** except data in **mk**;
- ancillary products under **miscellaneous**;
- documents under **document***;*

The LIDs will include the directory path and the full file name with extension and VIDs will always be set to 1, e.g.:

| | |
|---|---|
| **spice_kernels/fk/bc_mpo_v01.tf** | urn:esa:psa:bc_spice:spice_kernels:fk_bc_mpo_v01.tls::1.0 |
| **spice_kernels/fk/bc_mpo_v02.tf** | urn:esa:psa:bc_spice:spice_kernels:fk_bc_mpo_v02.tls::1.0 |
| **spice_kernels/spk/de430.bsp** | urn:esa:psa:bc_spice:spice_kernels:spk_de430.bsp::1.0 |
| **spice_kernels/spk/de431.bsp** | urn:esa:psa:bc_spice:spice_kernels:spk_de431.bsp::1.0 |

Table 15: Example of LIDVID as compared to the data filename for the "first method".

### 10.3.2 Path and filename without version in LID

The second method will include the directory path and the file name up to the version part and VIDs will always be set to the version part from the file name. This method applies to the following products:

- **mk** SPICE kernels under the spice_kernels;
- checksum tables;
- primary SPICE archive description documents (spiceds_v<ver>.html)

| | |
|---|---|
| **spice_kernels/mk/bc_v001.tm** | urn:esa:psa:bc_spice:spice_kernels:mk_bc::1.0 |
| **spice_kernels/mk/bc_v002.tm** | urn:esa:psa:bc_spice:spice_kernels:mk_bc::2.0 |
| **document/spiceds_v001.html** | urn:esa:psa:bc_spice:document:spiceds::1.0 |
| **document/spiceds_v002.html** | urn:esa:psa:bc_spice:document:spiceds::2.0 |

Table 16: Example of LIDVID as compared to the data filename for the "second method".

### 10.3.3 Subdirectory name only in LID

The third method includes only the subdirectory name and VIDs will always be set to the version part from the file name. This schema applies to:

- SPICE **document** collection products;
- SPICE **miscellaneous** collection products.
- **SPICE_kernels** collection top level directory products;

| | |
|---|---|
| **document/collection_document_v001.xml** | urn:esa:psa:bc_spice:document::1.0 |
| **miscellaneous/collection_miscellaneous_v002.xml** | urn:esa:psa:bc_spice:miscellaneous::2.0 |
| **spice_kernels/collection_spice_kernels_v010.xml** | urn:esa:psa:bc_spice:spice_kernels::10.0 |

Table 17: Example of LIDVID as compared to the data filename for the "third method".

### 10.3.4 No filename in LID

The last method includes only the initial part of the LID and VIDs will always be set to the version part from the file name. It applies to:

- all **SPICE bundle** products;

| | |
|---|---|
| **bundle_em16_spice_v001.xml** | urn:esa:psa:em16_spice::1.0 |
| **bundle_em16_spice_v002.xml** | urn:esa:psa:em16_spice::2.0 |

Table 16: Example of LIDVID as compared to the data filename for the "fourth method".

## 10.4 Product Reference and Collection Inventory Construction Rules

What follows is a set of rules provided by NAIF that the PSA implements to construct Product references and collection inventories:

- All products' Context_Area includes only Mission (*_to_investigation), Spacecraft (is_instrument_host), and one primary Target (*_to_target) LID references. These LIDs should be obtained from the PSA.
- All products' Reference_List includes the latest primary SPICE archive description document LID reference (*_to_document) (*except the primary SPICE archive description documents (spiceds_v<ver>.html) which can't reference themselves).
- Each MK's Reference_List also includes LIDVID references for all kernels (data_to_associate) listed in the MK.
- Each collection inventory lists LIDVIDs of *all* non-collection products provided under collection's directory at the time when collection product was created. In a particular collection inventory, ***P*** -for primary- is used only for newly added products (that don't appear in any of the collections with earlier versions) and ***S*** – for secondary - is used for products that have already been registered in a collection with an earlier version.
- Each Bundle label includes Bundle_Member_Entry'es only for the latest SPICE kernel collection LIDVID (bundle_has_spice_kernel_collection), the latest document collection LIDVID (bundle_has_document_collection) and the latest miscellaneous collection LIDVID (bundle_has_miscellaneous_collection). These collections have Primary statuses if they have not been registered in any earlier bundle versions. Otherwise they have Secondary statuses.

## 10.5 Start and Stop Date Time Assignment Rules

What follows is a set of rules provided by NAIF that the PSA implements to assign start and stop date time to products:

- *start_date_time* and *stop_date_time* appear in *Context_Area/Time_Coordinates* only in bundle, SPICE kernel collection, and SPICE kernel labels;
- for kernels for which time boundaries can determined from the data (SPK, CK, etc) *start_date_time* and *stop_date_time* set to those boundaries;
- for kernels for which time boundaries cannot be determined from the data (LSK, SCLK, PCK, etc) *start_date_time* and *stop_date_time* set to the default mission time range (from launch to an arbitrary date many decades into the future, e.g. 2050-01-01);
- for meta-kernels *start_date_time* and *stop_date_time* are set to the coverage provided by spacecraft SPK or CKs;
- for a SPICE collection the coverage is set to the boundaries of the combined coverage of the latest MKs that are part of this collection
- for a SPICE bundle the coverage is set to the boundaries of the coverage of the SPICE collection that is its member.

## 10.6 SPICE_kernel product label example

What follows is a preliminary example of a SPICE_kernel product label for:

spice_kernels/fk/em16_tgo_v16.tf

```
<?xml-model             href="http://pds.nasa.gov/pds4/pds/v1/PDS4_PDS_1B00.sch"
    schematypens="http://purl.oclc.org/dsdl/schematron"?>


<Product_SPICE_Kernel                   xmlns="http://pds.nasa.gov/pds4/pds/v1"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://pds.nasa.gov/pds4/pds/v1
http://pds.nasa.gov/pds4/pds/v1/PDS4_PDS_1B00.xsd">
```

```
    <Identification_Area>

<logical_identifier>urn:esa:psa:em16_spice:spice_kernels:fk_em16_tgo_v16.tf</lo
gical_identifier>
        <version_id>1.0</version_id>
        <title>em16_tgo_v16.tf</title>
        <information_model_version>1.11.0.0</information_model_version>
        <product_class>Product_SPICE_Kernel</product_class>
        <Citation_Information>
            <publication_year>2018</publication_year>
            <keyword>Observation Geometry</keyword>
            <description>SPICE FK file that provides TGO spacecraft,
structures, instruments and sensors frames. Created by the ESA SPICE
Service.</description>
        </Citation_Information>
    </Identification_Area>
    <Context_Area>
        <Time_Coordinates>
            <start_date_time>2016-03-14T08:13:00Z</start_date_time>
            <stop_date_time>2050-01-01T00:00:00Z</stop_date_time>
        </Time_Coordinates>
        <Primary_Result_Summary>
            <purpose>Observation Geometry</purpose>
            <processing_level>Derived</processing_level>
        </Primary_Result_Summary>
        <Investigation_Area>
            <name>ExoMars 2016</name>
            <type>Mission</type>
            <Internal_Reference>

<lid_reference>urn:esa:psa:context:investigation:mission.em16</lid_reference>
                <reference_type>data_to_investigation</reference_type>
            </Internal_Reference>
        </Investigation_Area>
        <Observing_System>
            <Observing_System_Component>
                <name>TGO</name>
                <type>Spacecraft</type>
                <Internal_Reference>

<lid_reference>urn:esa:psa:context:instrument_host:spacecraft.tgo</lid_referenc
e>
                    <reference_type>is_instrument_host</reference_type>
                </Internal_Reference>
            </Observing_System_Component>
        </Observing_System>
        <Target_Identification>
            <name>Mars</name>
            <type>Planet</type>
            <Internal_Reference>
                <lid_reference>urn:nasa:pds:context:target.mars</lid_reference>
                <reference_type>data_to_target</reference_type>
            </Internal_Reference>
        </Target_Identification>
    </Context_Area>
    <Reference_List>
        <Internal_Reference>

<lid_reference>urn:esa:psa:em16_spice:document:spiceds</lid_reference>
            <reference_type>data_to_document</reference_type>
        </Internal_Reference>
    </Reference_List>
    <File_Area_SPICE_Kernel>
```

```
        <File>
            <file_name>em16_tgo_v16.tf</file_name>
            <creation_date_time>2018-09-25T11:02:40</creation_date_time>
            <file_size                            unit="byte">159097</file_size>
            <md5_checksum>2bfd4c686e92c159ae50560dd0422d08</md5_checksum>
        </File>
        <SPICE_Kernel>
            <offset                                            unit="byte">0</offset>
            <object_length                    unit="byte">159097</object_length>
            <parsing_standard_id>SPICE</parsing_standard_id>
            <description>SPICE   FK   file   that   provides   TGO   spacecraft,
structures,  instruments  and  sensors  frames.  Created  by  the  ESA  SPICE
Service.</description>
            <kernel_type>FK</kernel_type>
            <encoding_type>Character</encoding_type>
        </SPICE_Kernel>
    </File_Area_SPICE_Kernel>
</Product_SPICE_Kernel>
```

# 11. EXTERNAL STANDARDS

This section defines specific requirements and recommendations that should be followed when using one of the following encoding standards to format the data (e.g. FITS, CDF).

The attribute <parsing_standard_id> is used in the PDS4 label to indicate the enconding standard in which the data are formatted. For a complete list of the allowed encoding standards in observational products, see allowed values of this attribute in the PDS4 Information Model (IM) [AD.03].

**IMPORTANT:** When using any of these external encoding data standards, it is strongly recommended to design data products that are PDS4-compliant and meet all PSA archiving requirements to avoid any reformatting or conversion of data only for archiving purposes. This will guarantee that the same data formats used and exchanged by the science team during the mission will be archived and preserved.

## 11.1 Specific Requirements and Recommendations for FITS

The values of the <parsing_standard_id> attribute to be used in PDS4 labels for FITS products is: "FITS 3.0".

FITS stands for **Flexible Image Transport System (FITS)**, and is the standard archival data format for astronomical data sets. Although FITS was originally designed for transporting image data on magnetic tape (which accounts for the 'I' and 'T' in the name), the capabilities of the FITS format have expanded to accommodate more complex data structures. This standard has been endorsed by the International Astronomical Union (IAU) for the interchange of astronomical data (IAU 1983).

The official reference document that defines the requirements for FITS format data files can be found at FITS Standard.

The FITS format is also commonly used by the planetary community to format and exchange their data, specially for tables and spectral data. However, the *de-facto* archiving standard for planetary data is PDS. Due to this, the recommendation is to design FITS data that are PDS4-compliant and meet all PSA archiving requirements to avoid any reformatting or conversion of data only for archiving purposes. This will guarantee that the same data formats used and exchanged by the science teams during the mission will be archived and preserved.

Some caveats on the PDS4 compatibility of FITS data formats and data types can be found in the PDS Small Bodies Node (SBN) PDS4 Wiki, as well as guidelines on how to generate PDS4 labels for FITS files.

The most significant compatibility issues are listed below:

- Must be compliant with FITS version 3.0. This is the PDS4 <parsing_standard_id> attribute in the Header object must be "FITS 3.0".
- FITS TABLE extension data is structurally compliant with PDS4 fixed-width ASCII table (Table_Character) requirements, but PDS requires that every line of the table data ends with a "carriage-return/linefeed" sequence.
- FITS BINTABLE extension data is structurally compliant with PDS4 binary table (Table_Binary) requirements but only if the format is not abused (e.g. attach header values to an observational data array is not allowed in PDS4).
- The only integer data types stored in a compliant FITS file are unsigned 8-bit bytes and signed 2-, 4-, and 8-byte integers, although PDS allows more integer data types.
- Array Axis Ordering. FITS array data are are stored such that the *NAXIS1* subscript varies fastest, *NAXIS2* next fastest, and so on up to *NAXISn*. So the storage order is first-index-fastest in FITS notation. This is the opposite in PDS.

- Padding in Tables. FITS standard requires that the BINTABLE data be padded with null ("\0") from the end of the last record to the end of the containing 2880-byte FITS block and TABLE data be padded with blanks to an even number of FITS 2880-byte blocks following the last record in the table. There is no requirement that these padding characters be included in the PDS4 label, and they are generally ignored.

FITS files comprise of one or more Header and Data Units (HDUs). The first (mandatory) HDU is called the primary HDU and *must* contain a primary header, and optional data. Subsequent HDUs are called FITS extensions and currently include image, ASCII table and binary table data formats, with accompanying headers. Each HDU in a FITS file must be described in the corresponding PDS4 label.

The PSA recommends the use of a **header only primary HDU** which should provide meta-data for the entire product (file), and one or more FITS extensions (additional HDUs) providing the actual data and corresponding meta-data. This is the structure given below.

### 11.1.1 Recommended FITS keywords for PDS4 archives

The table below includes a list of FITS keywords to be used to describe the data and metadata typically included in PDS4 products in the PSA.

*This is an initial proposal, to be consolidated in future versions of this document.*

In the Primary Header and Data Unit (HDU), the following structure is expected:

```
SIMPLE = T / conforms to FITS standard
BITPIX = 8 / array data type
NAXIS = 0 / number of array dimensions - 0 indicates there is no data array
for this HDU, only a header
EXTEND = T / FITS extensions are permitted here
< INSERT HERE THE METADATA LISTED BELOW AS APPLICABLE >
END
```

This indicates that the primary HDU has a header only, containing keywords in the following list. Any meta-data needed to analyse the data cannot only be in the FITS header (which is ignored by PDS4) and must also be included in the PDS4 label. Where corresponding PDS4 keywords exist they are included in the table.

| FITS Keyword | Data Type | Type of FITS Keyword | PSA Requirement | Definition | PDS4 Keyword |
|---|---|---|---|---|---|
| ***Identification Area*** | | | | | |
| LOGID | string | | | Logical identifier as per PSA conventions; unique at PDS level. | logical_identifier |
| VERSION | string | | | Version of the product, format: M.N (major.minor). | version_id |
| TITLE | string | | | Brief title for the product | title |
| HISTORY | string | | | | |
| ORIGIN | string | reserved | | Organization or institution responsible for creating the | not available in PDS4 (TBC) |

| | | | | | |
|---|---|---|---|---|---|
| | | | | FITS file. | |
| AUTHOR | string | reserved | | The value field shall contain a character string identifying who compiled the information in the data associated with the header. This keyword is appropriate when the data originate in a published paper or are compiled from many sources. | Citation_Information: author_list |
| REFERENC | string | reserved | | The value field shall contain a character string citing a reference where the data associated with the header are published. | Citation_Information: description |
| ***Observation Area*** | | | | | |
| DATE-OBS | string | reserved | | The date of the observation, in the format specified in the FITS Standard. The old date format was 'yy/mm/dd' and may be used only for dates from 1900 through 1999. The new Y2K compliant date format is 'yyyy-mm-dd' or 'yyyy-mm-ddTHH:MM:SS[.sss]'. | start_time |
| DATE-ST | string | | | Start date (first date of the measurements in the product) | start_time |
| DATE-END | string | | | Stop date (last date of the measurements in the product) | stop_time |
| PROLEVEL | string | | | Processing level. Allowed values are: Raw, Partially Processed, Calibrated, Derived | processing_level |
| MISSION | string | | | Mission Name | Observing_System_Component: name with type = Mission |
| INSTHOST | string | | | Instrument Host | Observing_System_Component: name with type = Spacecraft |
| INSTRU | string | reserved | | The value field shall contain a character string identifying the instrument used to acquire the data associated with the header. | Observing_System_Component: name with type = Instrument |
| INSTTYPE | string | | | Instrument Type; see allowed values in section | Observing_System_Component: type with type = Instrument |
| SUBINST | string | | | Sub-Instrument Identifier (typically an acronym) | Sub-Instrument: identifier |
| SUBINSTN | string | | | Sub-Instrument Name | Sub-Instrument: name |
| SUBINSTT | string | | | Sub-Instrument Type; see allowed values in section | Sub-Instrument: type |
| OBJECT | string | reserved | | The value field shall contain a character string giving a name for the object observed (see | Target_Identification: name |

| | | | | | |
|---|---|---|---|---|---|
| | | | | also the NAIF ID list). | |
| ***Mission Area: Mission Information*** | | | | | |
| CLK-ST | string | | | Spacecraft clock elapsed time for the first measurement in the data product | spacecraft_clock_start_count |
| CLK-END | string | | | Spacecraft clock elapsed time for the last measurement in the data product | spacecraft_clock_stop_count |
| PHASENA | string | | | Mission phase name; allowed values are defined in the corresponding mission dictionary. | mission_phase_name |
| PHASEID | string | | | Mission phase identifier (typically an acronym); allowed values are defined in the corresponding mission dictionary. | mission_phase_identifier |
| ORB-ST | Unsigned Integer | | | Orbit for the first measurement in the data product. | start_orbit_number |
| ORB-END | Unsigned Integer | | | Orbit for the last measurement in the data product. | stop_orbit_number |
| ***Mission Area: Observation Context*** | | | | | |
| OBSID | string | | | Observation Identifier; allowed values are defined by the mission in the corresponding Annex. | observation_identifier |
| POINTING (TBC) | string | | | Spacecraft pointing; allowed values are defined in the PSA dictionary. | spacecraft_pointing |
| ***Mission Area: Quality Information*** | | | | | |
| QUALFLAG | string | | | Quality flag; allowed values or guidelines are defined by the mission in the corresponding Annex. | quality_flag |
| ***Mission Area: Processing Context*** | | | | | |
| PROCSW | string | | | Processing Software Title | processing_software_title |
| PROCSWVE | string | | | Processing Software Version | processing_software_version |
| ***File Area Observational*** | | | | | |
| DATE | string | reserved | | The date on which the file was created, in the format specified in the FITS Standard. The old date format was 'yy/mm/dd' and may be used only for dates from 1900 through 1999. The new Y2K compliant date format is 'yyyy-mm-dd' or 'yyyy-mm-ddTHH:MM:SS[.sss]'. | File_Area_Observational: File: creation_date_time |

### 11.1.2 FITS Keywords for ASCII Table Extensions

See relevant information at the PDS SBN PDS4 Wiki > ASCII Tables.

```
XTENSION   =   'TABLE'                                    /  ASCII  table  extension
BITPIX      =   8                                                   /  8-bit  bytes
NAXIS      =  2                                       /  2-dimensional  ASCII  table
NAXIS1      =   <record_length>                          /  width  of  table  in  bytes
NAXIS2      =  <records>                                  /  number  of  rows  in  table
PCOUNT     =  0                                            /  size  of  special  data  area
GCOUNT     =  1                                   /  one  data  group  (required  keyword)
TFIELDS   = <fields>             / number of fields in each row
```

In PDS4, an ASCII Table is described with the class <*Table_Character*>, where:

- *NAXIS1* corresponds to the <*record_length*> attribute in the <*Record_Character*> class.
- *NAXIS2* corresponds to the <*records*> attribute in the <*Table_Character*> class.
- *TFIELDS* corresponds to the <*fields*> attribute in the <*Record_Character*> class. Unlike in *BINTAB* tables, ASCII tables may not have fields that are arrays, vectors, or complex numbers.

The required <*groups*> attribute will always have a value of zero (fields in FITS tables are not allowed to have repetition counters).

For each column (<*Field_Character*> in PDS4), the following keywords are needed for each column, where the 'n' is an integer counter of the field starting from one, and with no padding (e.g. TTYPE1):

| FITS | PDS4 Field_Character | Status |
|---|---|---|
| TTYPEn | <name> | mandatory |
| TBCOLn | <field_location> | mandatory |
| TFORMn | <data_type> (1) | mandatory |
| TUNITn | <unit> | mandatory (if applicable) |
| TSCALn | <scaling_factor> | optional |
| TZEROn | <value_offset> | optional |
| TNULLn | see below (2) | optional |
| TDISPn | see below (3) | optional |

(1) see table below.

(2) *TNULL* is a string used as a null data flag for the field and corresponds to the value found in the file - before scaling or offset are applied. It can be translated into one of the specific flags in the <*Special_Constants*> class, depending on the circumstances of the data.

(3) *TDISP* is an optional keyword that can be used to contain a more specific display format than that implied by the *TFORM* keyword.

| TFORM | PDS4 <data_type> |
|---|---|
| I*w* | ASCII_Integer (4) |
| F*w.d* | ASCII_Real |
| E*w.d* | ASCII_Real |

| | |
|---|---|
| Dw.d | ASCII_Real |

(4) Or more specific ASCII_* string type e.g. ASCII_Date_Time for date/time fields conforming to the standard ISO 8601 format.

### 11.1.3 FITS Keywords for Binary Table Extensions

See relevant information at the PDS SBN PDS4 Wiki > Binary Tables.

### 11.1.4 FITS Keywords for Array Extensions

See relevant information at the PDS SBN PDS4 Wiki > Arrays.

### 11.1.5 FITS Keywords for Planetary Surface Data (GeoFITS)

An extension of FITS metadata for planetary surface investigations in under definition by VESPA (Virtual European Solar and Planetary Access) in the frame of Europlanet H2020 , see GeoFITS.

### 11.1.6 Relevant FITS tools

- FITS data is already supported by GDAL via the external library CFITSIO (see the GDAL documentation).
- FV: Interactive FITS file editor. FV can be used with the DS9 image display.
- fitsverify - A FITS File Format-Verification Tool. A web version of this tool can be found at: https://fits.gsfc.nasa.gov/fits_verify.html

## 11.2 Specific Requirements and Recommendations for CDF

The Common Data Format (CDF) is widely used in the heliophysics community and by some instruments on planetary missions. CDF files can be archived in PDS4, providing they are formatted to meet the CDF-A standard (CDF-Archive) and include appropriate meta-data.

The values of the <parsing_standard_id> attribute to be used in PDS4 labels for CDF products is: "CDF 3.4 ISTP/IACG".

As per the guidelines defined by the PDS Plasma Planetary Interactions (PPI) Node, to ensure data in a CDF file will be archivable in PDS4 with no conversion, the following set of requirements should be met:

1. The CDF file must be CDF version 3.4 or later.
2. All data variables which are part of the CDF file are included in the same file.
3. No compression (file or variable).
4. No unused records (no superfluous, non-decodable records).
5. No fragmented variables (all data for a variable must be contiguous in the file).
6. No sparse variables. All data values are physical (data for all dimensions in a variable are written).
7. No virtual (calculated) variables.
8. Variables are stored in row majority order.
9. Use only "zVariables" for data.

In addition, the CDF file should include at least a minimum set of metadata as per the requirements listed in the following section: ISTP/IACG Metadata.

Reference documents are listed below, and can be found in the PDS Plasma Planetary Interactions (PPI) Node documentation page:

- CDF-A Specification: The CDF-A Specification describes the types of CDF files which meet basic archiving requirements. The specification includes HPDE and PDS extensions which define additional requirements for each domain.
- Guide to Archiving CDF Files in PDS4: Details on how to prepare CDF files for archiving in a PDS4 system with information about available tools.

### 11.2.1 ISTP/IACG Metadata

Detailed definition of the ISTP/IACG metadata attributes can be found in the following links:

- Global Attributes: https://spdf.gsfc.nasa.gov/istp_guide/gattributes.html
- Variable Attributes: https://spdf.gsfc.nasa.gov/istp_guide/vattributes.html

| CDF Attribute | PSA Requirement based on ISTP/IACG | ISTP/IACG Definition | Enumerated Values | PDS4 Metadata Use |
|---|---|---|---|---|
| ***Global Attributes*** | | | | |
| Acknowledgement | recommended | | n/a | n/a |
| *ADID_ref* | *no longer relevant* | | *n/a* | *n/a* |
| Data_type | **required** | ISTP defined exchangeable data product type. (enumeration) | | |
| Data_version | **required** | Project assigned version for the data. | n/a | Product_Observational: Identification_Area: version_id |
| Descriptor | **required** | The name of the instrument or sensor that collected the data. | *See applicable Mission annex.* | long name used in: Product_Observational: Observing_System_Component: name when type = Instrument |
| Discipline | **required** | The science discipline and subdiscipline. (enumeration) | • Planetary Physics>Atmospheres<br>• Planetary Physics>Fields<br>• Planetary Physics>Flux Measurements<br>• Planetary Physics>Particles<br>• Planetary Physics>Spectroscopy | Product_Observational: Observation_Area: Primary_Result_Summary: Science_Facets: discipline_name<br>*The value in PDS4 will include only the subdiscipline.* |

| | | | | |
|---|---|---|---|---|
| Generated_by | recommended | The generating data center/group. | n/a | n/a |
| Generated_with_software | recommended | | n/a | psa: Processing_Context: processing_software_version |
| Generation_date | recommended | Date stamp for the creation of the file. | n/a | Product_Observational: Identification_Area: Modification_History: Modification_Detail: modification_date |
| HTTP_LINK | recommended | The URL for the PI or Co-I web site holding on-line data. | n/a | n/a |
| Instrument_type | **required** | The ISTP defined instrument type. Multi-valued. (enumeration) | • Electric Fields (space)<br>• Engineering<br>• Ephemeris<br>• Gamma and X-Rays<br>• Magnetic Fields (space)<br>• Particles (space)<br>• Plasma and Solar Wind<br>• Radio and Plasma Waves (space) | Product_Observational: Observing_System_Component: type with type = Instrument<br>*See section 7.3.1; updates expected in PDS4 version 1.11.* |
| LINK_TEXT | recommended | Text describing on-line data available at PI or Co-I web sites. | n/a | n/a |
| LINK_TITLE | recommended | The title of the web site holding on-line data available at PI or Co-I web sites. | n/a | n/a |
| Logical_file_id | **required** | The name of the CDF file using the ISTP naming convention. | n/a | n/a |
| Logical_source | **required** | Source_name, data_type, and descriptor information. | n/a | n/a |
| Logical_source_description | **required** | Full words associated with the Logical_source. Value should be ~1 line working/useful title for the data file. | n/a | Product_Observational: Identification_Area: title |
| Mission_group | **required** | The assigned name of the mission or project. (enumeration) | *See applicable Mission annex.* | Product_Observational: Observation_Area: Investigation_Area: name |
| MODS | recommend | History of modifications | n/a | Product_Observational: |

| | | | | |
|---|---|---|---|---|
| | ed | made to the CDF data set. | | Identification_Area: Modification_History: Modification_Detail |
| Parents | optional | The parent CDF(S) for files of derived and merged data sets. | n/a | Product_Observational: Reference_List: Source_Product_Internal/External |
| *PDS_collection_id* | **required** | Collection identifier portion of the PDS4 logical_identifier of the primary collection with which the data product is a member. | n/a | Collection identifier portion in Product_Observational: Identification_Area: logical_identifier |
| *PDS_sclk_start_count* | recommended | Spacecraft clock (or Mission Elapsed Time) start count associated with the first start time of the file. | n/a | msn:spacecraft_clock_start_count |
| *PDS_sclk_stop_count* | recommended | Spacecraft clock (or Mission Elapsed Time) stop count associated with the last end time of the file. | n/a | msn:spacecraft_clock_stop_count |
| *PDS_start_time* | **required** | First start time (UTC) of the file in the format: YYYY-MM-DDThh:mm:ss.sssZ. Where YYYY = 4-digit year, MM = month of year, DD = day of month, hh = hour of day, mm= minute of hour, ss.sss = seconds of minute, and "T", and "Z" are literal. | n/a | Product_Observational: Observation_Area: Time_Coordinates: start_date_time |
| *PDS_stop_time* | **required** | Last stop time (UTC) of the file in the format: YYYY-MM-DDThh:mm:ss.sssZ. See PDS_start_time for details. | n/a | Product_Observational: Observation_Area: Time_Coordinates: stop_date_time |
| PI_affiliation | **required** | A recognizable abbreviation of the PI affiliation. | n/a | n/a |
| PI_name | **required** | First initial and last name of the PI. | n/a | n/a |
| Planet | recommended | | *See section 3.3.1 for allowed values.* | Target_Identification: name when type = Planet |
| Project | **required** | The name of the project i.e. mission full name. | *See applicable Mission annex.* | Product_Observational: Observing_System_Component: name when type = Mission |
| Rules_of_use | recommended | Citability and PI access restrictions. This may point to a World Wide | n/a | Some information can be provided in:<br>Product_Observational: |

| | | | | |
|---|---|---|---|---|
| | | Web page specifying the rules of use. | | Identification_Area: Citation_Information: description |
| Skeleton_version | optional | The skeleton file version number. | n/a | n/a |
| Software_version | optional | The version of the software that generated the CDF. | n/a | psa: Processing_Context: processing_software_version |
| Source_name | **required** | The mission or investigation that contains the sensors i.e. mission full name. | *See applicable Mission annex.* | Product_Observational: Observation_Area: Investigation_Area: name |
| TEXT | **required** | Description of the experiment. Value should be a full (long) description of the data file. | n/a | Product_Context: Instrument: description |
| Time_resolution | recommended | The time resolution of the file. | n/a | n/a |
| TITLE | optional | A title for the data set. | n/a | Product_Observational: Identification_Area: title |
| Validate | optional | Written by software for automatic validation of features. | n/a | n/a |
| spase_DatasetResourceID | recommended | Unique dataset identifier assigned by SPASE, of the form spase://VSPO/"UNIQUE_ID", where "UNIQUE_ID" is the ID assigned to the SPASE resource record for the dataset in the SPASE system. The SPASE resource record provides metadata about the dataset, including pointers to locations holding the data. | n/a | TBC |
| spase_DatasetResource | recommended | The SPASE XML description of the dataset that corresponds to the SPASE ResourceID. | n/a | TBC |
| ***Variable Attributes*** | | | | |
| AVG_TYPE | optional | | n/a | n/a |
| CATDESC | **required** | Approximately 80-character string which is a textual description of the variable. Brief (~1 line) description of the variable. | n/a | Array_*D/description |
| DEPEND_0 | **required** | | n/a | Axis_Array/axis_name |

| | | | | |
|---|---|---|---|---|
| DEPEND_i | **required** | Ties a dimensional data variable to a support_data variable. Contains the name of the variable. "i" is replaced with the index of the dimension. For ISTP the DEPEND_0 must be defined and have the value 'Epoch'. | n/a | Axis_Array/axis_name |
| DICT_KEY | optional | | | n/a |
| DISPLAY_TYPE | **required** | what type of plot to make (e.g. time_series, spectrogram, stack_plot, image) . Enumerated values list may be expanded as needed. | time_series<br>spectrogram<br>stack_plot<br>image | potentially used in Discipline_Area (giving hints for data visualization) |
| FIELDNAM | **required** | A description of the variable (up to 30 characters). | n/a | Array_*D/name |
| FILLVAL | **required** | The value used for missing or invalid variable values. Should be of same data type as the variable which they are describing. | n/a | potentially used in: Special_Constants/*_constant |
| FORMAT/ FORMAT_PTR | **required** | A Fortran format specification that is used when displaying a variable value. | n/a | mapped to: .../field_format |
| LABLAXIS/ LABL_PTR_i | **required** | The label for the y-axis of a plot or to provide a heading for a data listing. If labeling a variable with dimensions use the LABL_PTR_i form with "i" replaced with the index of the dimension. | n/a | n/a |
| MONOTON | optional | The monotonicity of a variable: INCREASE (strictly increasing values), DECREASE (strictly decreasing values), or FALSE (not monotonic). | INCREASE<br>DECREASE<br>FALSE | n/a |
| UNITS/UNIT_PTR | **required** | The units of the variable. In the UNIT_PTR form it contains the name of the variable which stores the UNITS string. To be used only if UNITS is not present. | n/a | Element_Array/unit |
| SCALEMIN | optional | The minimum value for scaling a variable when graphically displaying its values. Should be of same data type as the variable | n/a | potentially used in Discipline_Area (giving hints for data visualization) |

| | | | | |
|---|---|---|---|---|
| | | which they are describing. | | |
| SCALEMAX | optional | The maximum value for scaling a variable when graphically displaying its values. Should be of same data type as the variable which they are describing. | n/a | potentially used in Discipline_Area (giving hints for data visualization) |
| SCALETYP | recommended | | linear<br>log | potentially used in Discipline_Area (giving hints for data visualization) |
| VALIDMIN | **required** | The minimum valid value for a variable. Should be of same data type as the variable which they are describing. | n/a | use dependent upon variable |
| VALIDMAX | **required** | The maximum valid value for a variable. Should be of same data type as the variable which they are describing. | n/a | use dependent upon variable |
| VAR_NOTES | **required** | Ancillary information about the variable.<br>*Note: While ISTP/IACG classifies VAR_NOTES as optional, please include a complete description here as it will be used for the description in the PDS label file.* | n/a | Array_*D/description |
| VAR_TYPE | **required** | Identifies a variable as either data, support_data, metadata or ignore_data.<br>Value should be a full description of the variable. | data<br>support_data<br>metadata<br>ignore_data | n/a |

### 11.2.2 Use of PDS4 local data dictionaries (LDD) to describe CDF data products

Variables in CDF files are organised as arrays of various dimensions (1, 2, 3, … N).

- The core PDS4 DOES SUPPORT describing individual arrays (dimensions, data types, units, etc.).
- The Core PDS4 DOES NOT describe relationships between the arrays i.e.
  - Array 1 contains TIME(i) or ENERGY(i)
  - Arrays 2, 3, … N contain something B_VECTOR(x,y,z)(i) at TIME(i)

The "Particle" and "Alt" Local Data Dictionaries (LDDs) are designed to describe relationships between the arrays:

- Primary values (flux), axis values (time), and face values (bin variables).

In addition:

- The PSA and Mission local data dictionaries should be used to define mission specific keywords (e.g. mission_phase_name, …)

- Other useful LDDs (Geometry, Cartography, etc.)

All Local Data Dictionaries (LDDs) registered with PDS are available at: https://pds.nasa.gov/pds4/schema/released/

### 11.2.3 Relevant CDF tools

- Tools to verify ISTP/IACG compliance
    - SKTEditor/Other tool – verify that the CDFs comply with standards and tools
        - http://spdf.sci.gsfc.nasa.gov/skteditor /
    - CDFConvert – convert internal structure of existing CDF files
        - http://cdaweb.sci.gsfc.nasa.gov/pub/software/cdf/dist
            - this repository contains the complete CDF library, including various tools like cdfconvert
        - `> cdfconvert {src.cdf} {dest.cdf} -single -network -sparseness vars:srecords.no  -compression vars:none -zmode 2`
        - This will convert {src.cdf} (which can be a multiple file CDF) and write to {dest.cdf} a copy that is written as a single file with network encodings, no sparseness, no compression and that contains only zVariables.
- Tools to verify CDF-A compliance
    - http://release.igpp.ucla.edu/pds/cdf/
    - cdfcheck: Scan a CDF file and determine if the content is compliant with archive requirements. Checks include if the CDF files meets requirements specified for CDF tools, ISTP/CDF and PDS archiving.
    - Example: http://release.igpp.ucla.edu/pds/cdf/example/index.html
- Tools to create PDS4 labels for CDF files (Product_Observational)
    - Data providers can create tools to generate PDS4 labels from CDF files
    - PPI has created an application (igpp.docgen) which defines a environment that can process Apache Velocity templates into PDS4 labels.
    - http://release.igpp.ucla.edu/igpp/docgen /
    - Example: http://release.igpp.ucla.edu/igpp/docgen/example/cdf/generate-pds4.html
- Tools to read PDS4-compatible CDF data
    - PDS4-compatible CDF reader in Python: https://github.com/MAVENSDC/cdflib

### 11.2.4 How-to Steps

CDF is a complex data format but properly formatted files can be fully described in PDS4. Some simple rules to follow are:

- Make sure that the CDFs you build conform to the CDF ISTP/IACG spec: SKTEditor.
- Make sure that you create CDF-A compatible files: CDFConvert.
- Validate that your CDFs are truly CDF-A compliant: cdfcheck.
- Identify which LDDs are required to support your labels.
- Create mission specific LDDs if necessary.

- If using igpp.docgen to build your labels, create the Velocity template and the run the code.

# 12. INTEROPERABILITY

## 12.1 EPN-TAP Extension and Metadata Requirements

To be included in a future version of this document.