ESDC-PSA-TN-0002-AG

ISSUED 2.0 ESDC-PSA-TN-0002-signed

IN PROGRESS 2.1

Metadata

reference	ESDC-PSA-TN-0002
title	PSA PDS4 Archiving Guide
author	S. Martinez, T. Lim, D. Heather, M. Costa
modified	02/08/2017
issue	2
revision	0
status	Issued
doctype	TN
distribution	-
header	ESA UNCLASSIFIED - For Official Use
footer	ESA UNCLASSIFIED - For Official Use
organization	esac
address	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

Approval

PSA PDS4 Archiving Guide	
Issue 2	**Revision** 0
Author S. Martinez, T. Lim, D. Heather, M. Costa	**Date** 02-08-2017
Approved by	**Date**
C. Arviset	-
S. Besse	-
C. Vallat	-
I. Barbarisi	-

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 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

Change record

Issue 2	Revision 0		
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

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

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 single bundles 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. Each bundle is stored in the PSA as a single entity hence all the data from an instrument for the entire mission is stored in a single instrument bundle in the PSA.

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 actually 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” unique for each product. The implementation of LIDVIDs used in the PSA is described in chapter 4.

An overview of the data flow and associated processes will be included here in a future version of the document.

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 the present document) 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*

All requirements, data structures and permissible values mentioned above are available in the form of XML Schema and Schematron files, which can be used to automatically validate the PDS4 labels.

- Common PDS4 schemas (common requirements) are supplied by the PDS via the PDS data dictionary.
- The PSA has additional schemas and the PSA dictionary is publicly available in the PSA bundle (ftp://psa.esac.esa.int/pub/mirror/PSA).

Details of the PDS4 PSA schema location/management will be added in a future version of the document.

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 Product_Document Product_File_Text
document	Contains documentation and supporting information to assist in understanding and using the primary products.	Product_Document Product_File_Text
geometry	Contains geometry products including pointing, orientation and positioning information typically provided in tables of calculated values.	Product_Observational Product_Document
browse	Contains browse products including overview representations or quick-look plots/reports of the primary products.	Product_Browse Product_Thumbnail Product_Document
miscellaneous	Contains miscellaneous products including any additional information and documentation products not easily classified as one of the other collections.	Product_Observational Product_Document 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

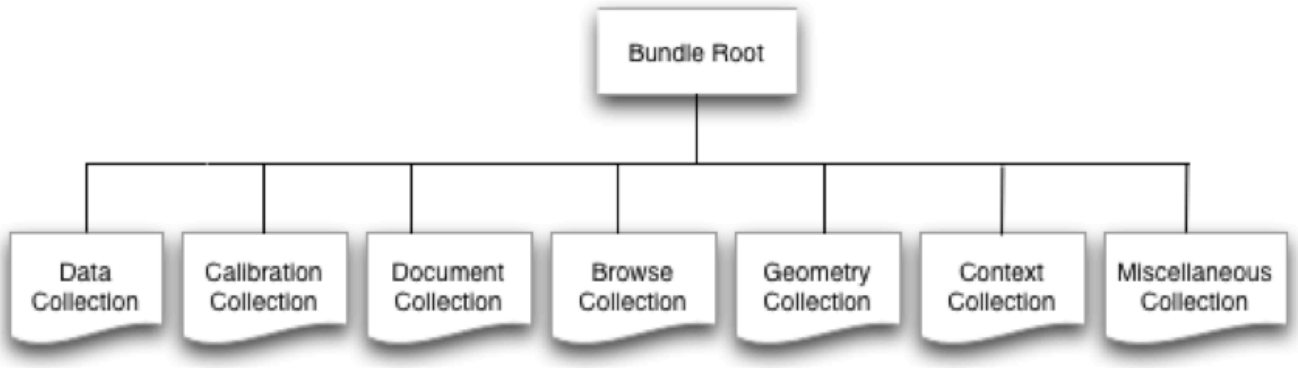

Figure 1 PDS4 bundle structure

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 do not apply to the SPICE Bundle. For more information on the SPICE Bundle Contents and Organisation see Section 9.

- **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 (e.g.).
    - **collection_geometry.xml/csv** files (required; if geometry collection present): Calibration 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.

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*.

3.1.1. Mandatory and Optional Collections

Data Organisation	*No. 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 ***(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 • document • miscellaneous_ancillary • xml_schema
Instrument Bundle			
urn:esa:psa:<bundle_identifier>	<bundle_identifier>.xml	<bundle_identifier>	• data_telemetry • data_raw • data_partially_processed • data_calibrated • data_derived • browse • calibration • geometry • miscellaneous_ancillary • document • context (S) • xml_schema (S)
SPICE Bundle			
urn:esa:psa:<mission_acronyme>_spice	bundle_<mission_acronym>_spice.xml	<mission_acronym>_spice	• document • miscellaneous • document • 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_<instrument_id>_<collection_identifier>.<ext> 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 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
            - subunit 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 PDS4_agency_ESA.xml		is_agency
Investigation	urn:nasa:pds:context:investigation:mission.<mission_id> PDS4_mission_<mission_name>.xml	Investigation_Area	is_investigation
Instrument Host	urn:nasa:pds:context:instrument_host:spacecraft. <instrument_host_id> PDS4_host_<instrument_host_id>.xml	Observing_System	is_instrument_host
Instrument	urn:esa:psa:context:instrument:<instrument_id>.<instrument_host_id> 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) urn:esa:psa:context:target.<target_name> (LID, if under PSA governance) PDS4_target_<target_name>.xml	Target_Identification	data_to_target
Personnel	urn:esa:psa:context:personnel:personnel.<person_id> PDS4_personnel_<person_id>.xml		
Resource	urn:esa:psa:context:resource:resource.<resource_id> PDS4_resource_<resource_id>.xml		

Table 5: Context file naming conventions

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 Std. Reference (section 6D.3.1, page 61) [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

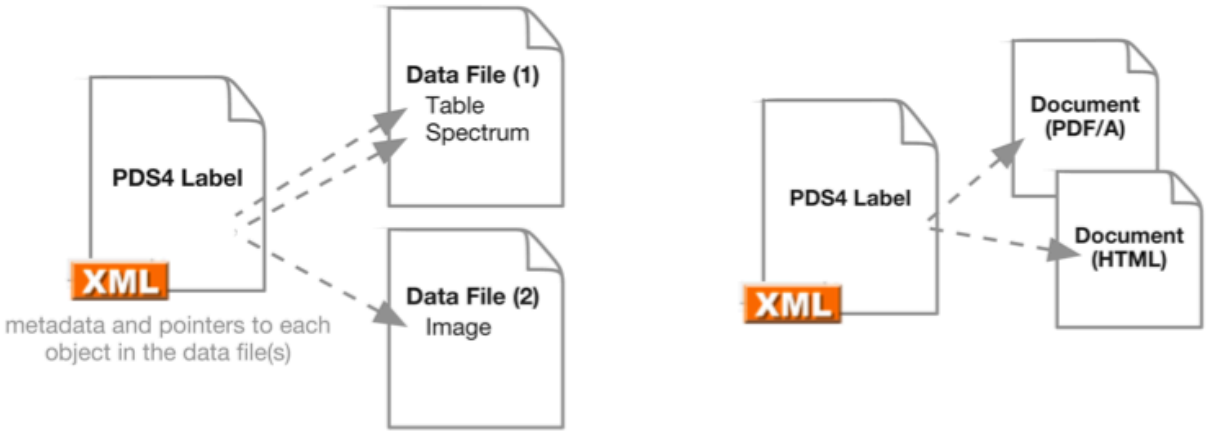

Figure 2: Illustration of a PDS4 Product_Observational and a PDS4 Product_Document

5.1. Fundamental Rules

This section provides a summary of the fundamental rules (see PDS4 DPH [RD.01], page 10) that should be taken into account when creating

PDS4 products. For a full description of the applicable rules, see 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). See section 4B in the PDS4 Std. Reference [AD.01]. 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. 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. 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). 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*** *(A.k.a. data structures or objects)*	***Derived from*** *(See* Table 8*)*
Observational	Product_Observational (1)	Array_1D	Array Base
		Array_2D	

		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.

5.4. Date and Time Types

5.4.1. 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.fffZ (calendar format)

Or

YYYY-DOYThh:mm:ss.fffZ (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)
- **fff** is decimal seconds (number of digits commensurate with precision)
- **Z** denotes UTC (***required*** when the time is given in UTC)

5.4.2. Spacecraft Clock Formats

This will be moved to the corresponding mission annex (mission specific).

The Spacecraft Clock format is a native time format specific to each mission and is therefore defined in each mission-specific dictionary (see mission PDS4 schemas in each appendix). 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. *It contains the time as 32 bits of unit seconds followed by 16 bits of fractional seconds.*

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

5.4.3. OBT to UTC Time Conversion

This will be moved to the corresponding mission annex (mission specific).

UTC time is a function of the Time Correlation Packets and the on-board time (OBT). The Time Correlation Packets contain 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.

6. 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.

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>
```

Figure 3: Simplified PDS4 label example

Product Type	*Top-level Classes*	*Reference*
Product_Observational	Identification Area Observation_Area Reference_List File_Area_Observational File_Area_Observational_Supplemental	See Product_Observational in the PDS4 Schema (XSD).
Product_Browse	Identification Area Reference_List File_Area_Browse	See Product_Browse in the PDS4 Schema (XSD).
Product_Thumbnail	Identification Area Reference_List File_Area_Encoded_Image	See Product_Thumbnail in the PDS4 Schema (XSD).
Product_Document	Identification Area Context_Area Reference_List Document Document_Format_Set	See Product_Document in the PDS4 Schema (XSD).
Product_File_Text	Identification Area Reference_List File_Area_Text	See Product_File_Text in the PDS4 Schema (XSD).

Product_SPICE_Kernel	Identification Area Context_Area Reference_List File_Area_SPICE_Kernel	See Product_SPICE_Kernel in the PDS4 Schema (XSD).

Table 10: Top-level classes in PDS4 labels

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).

6.1. Data Types

Attribute values (metadata) in the PDS4 labels must conform to the data types definitions and rules specified in the PDS4 Schemas. The definitions can be found in section 5A of the PDS4 Standards Reference [AD.01]. This includes boolean types, date and time types, numeric types and string types.

6.2. 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.

6.2.1. XML Declaration and Schema Reference Information

To be completed in the next version of the document.

6.2.2. 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	
Primary_Result_Summary	Observation_Area	Required	See section 0
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 6.2.3
Discipline_Area	Observation_Area	Required	

Table 11: Requirements for PDS4 labels of observational products

6.2.2.1. 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

6.2.3. Discipline Specific Classes and Attributes

PSA specific classes and attributes (if needed) will be included in the <Discipline_Area> class of a PDS4 label, which is only available in the following product types:

- Product_Observational
- Product_Context
- Product_SPICE_Kernel

PSA specific classes and attributes are defined in XML Schema files that are referred to as PSA dictionaries and are part of the PSA PDS4 Schemas.

6.2.4. Mission Specific Classes and Attributes

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

- Product_Observational
- Product_Context
- Product_SPICE_Kernel

Mission specific classes and attributes are defined by the PSA (if to be used consistently across missions) or by the relevant SGS (according to the mission needs) in XML Schema files that are referred to as PSA or <Mission> dictionaries accordingly. These dictionaries are part of the PSA and <Mission> PDS4 Schemas. 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).

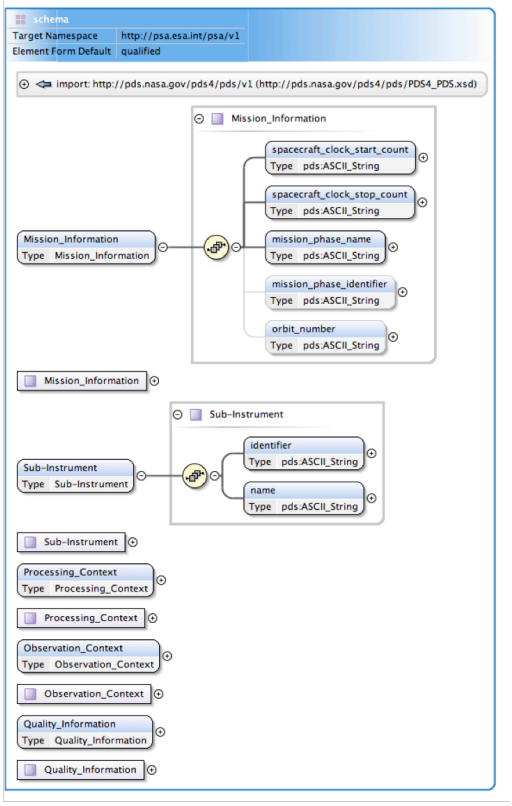

Figure 4: PSA PDS4 dictionary (Mission Area)

```
<!-- ============================================================================================ -->
<!-- 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 5: Example of a <Mission_Area> section in a PDS4 label

7. Geometry

This is work in progress (under definition within PDS).
Note: A PDS Geometry White Paper has been distributed by PDS (latest version dated 15-January-2014). This paper has been used to define PDS4 geometry requirements.

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

7.1. Requirements and Recommendations

This section will include a list of requirements and recommendations related to the geometry information (e.g. minimum geometry information in the label depending on the type of instrument, standard formats for capturing the observation footprints i.e. geographic coverage of the data).

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.

```
<!-- ============================================================================================ -->
<!-- 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 6: Example of geometry metadata in a PDS4 label

7.2. Coordinate Systems and Reference Frames

To be included in a future version of this document.

8. 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.

8.1. 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). 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. Geometry products and SPICE kernels will be classified as Derived.

Table 13: Data processing levels

9. 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/<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. At least, the following SPICE kernel files will 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.
- **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.

9.1. 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_v001.tf - urn:esa:psa:em16_spice:spice_kernels: em16_tgo_v00**1**.tf::**1**.0
- em16_tgo_v002.tf - urn:esa:psa:em16_spice:spice_kernels: em16_tgo_v00**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 File name (italics)	**N [1]**	**Product Type LIDVID**
<mission_acronym>_spice/	1	
bundle_<sc>_spice.xml	1	Product_Bundle urn:esa:psa:<sc>_spice::<n>.0
readme.txt	1	fixed
document	1	fixed
collection_document.xml/.csv	1	Product_Collection urn:esa:psa:<sc>_spice:document::<n>.0
spiceds.xml/.html	1	Product_Document urn:esa:psa:<sc>_spice:document:spiceds::<n>.0
miscellaneous	1	fixed
collection_miscellaneous.xml/.csv	1	Product_Collection urn:esa:psa:<sc>_spice:miscellaneous::<n>.0
checksum	1	
checksum_v<ver>.xml/.tab	1..n	Product_Ancillary (described as Checksum_Manifest) urn:esa:psa:<sc>_spice:miscellaneous:checksum_checksum::<n>.0
orbnum	0,1	fixed
<ker_name>.xml/.orb	0..n	Product_Ancillary (described as Table_Character) urn:esa:psa:<sc>_spice:miscellaneous: orbnum_<filename.ext>::1.0
spice_kernels	1	fixed
collection_spice_kernels.xml/.csv	1	Product_Collection urn:esa:psa:<sc>_spice:spice_kernels::<n>.0
ck	1	
<ker_name>.xml/.bc	1..n	Product_SPICE_Kernel urn:esa:psa:<sc>_spice:spice_kernels:ck_<filename.ext>::1.0
dsk	0,1	

<ker_name>.xml/.bds	0..n	Product_SPICE_Kernel urn:esa:psa:<sc>_spice:spice_kernels:dsk_<filename.ext>::1.0
fk	1	
<ker_name>.xml/.tf	1..n	Product_SPICE_Kernel urn:esa:psa:<sc>_spice:spice_kernels:fk_<filename.ext>::1.0
ik	1	
<ker_name>.xml/.ti	1..n	Product_SPICE_Kernel urn:esa:psa:<sc>_spice:spice_kernels:ik_<filename.ext>::1.0
lsk	1	
<ker_name>.xml/.tls	1..n	Product_SPICE_Kernel 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 urn:esa:psa:<sc>_spice:spice_kernels:mk_<sc>_<des>_<ver>::<n>.0
pck	1	
<ker_name>.xml/.tpc/.bpc	1..n	Product_SPICE_Kernel urn:esa:psa:<sc>_spice:spice_kernels:pck_<filename.ext>::1.0
sclk	1	
<ker_name>.xml/.tsc	1..n	Product_SPICE_Kernel urn:esa:psa:<sc>_spice:spice_kernels:sclk_<filename.ext>::1.0
spk	1	
<ker_name>.xml/.bsp	1..n	Product_SPICE_Kernel 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 collction might contain other collections with Product_Document products.

9.2. 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:

9.2.1. LIDVID construction

There are several methods of constructing LIDVIDs depending on the product. 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** except checksum tables *checksum_v<ver>*.tab;
- documents under **document** except *spiceds.html;*

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

miscellaneous/orbnum/bc_orb1.orb	urn:esa:psa:bc_spice:miscellaneous:orbnum_bc_orb1.orb::1.0
miscellaneous/orbnum/bc_orb2.orb	urn:esa:psa:bc_spice:miscellaneous:orbnum_bc_orb2.orb::1.0
spice_kernels/fk/bc_mpo_v001.tf	urn:esa:psa:bc_spice:spice_kernels:fk_bc_mpo_v01.tls::1.0
spice_kernels/fk/bc_mpo_v002.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".

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

- data in **mk** ;
- checksum tables *checksum_v<ver>*;

The LID will include the directory path and the file name up to the version number –with no reference to it- and VIDs will always be set to the version part from the file name, e.g.:

miscellaneous/checksum/checksum_v001.tab	urn:esa:psa:bc_spice:miscellaneous:checksum_checksum::1.0
miscellaneous/checksum/checksum_v002.tab	urn:esa:psa:bc_spice:miscellaneous:checksum_checksum::2.0
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

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

The third method, will include the version number embedded within its label and applies to the following products:

- SPICE **document** collection products;
- SPICE **miscellaneous** collection products, except for **orbnum** products;
- **SPICE_kernels** collection top level directory products;
- primary SPICE archive description document (spiceds.html);
- all **SPICE bundle** products;

The LID will include only the subdirectory name and VIDs will always be set to the version part indicated in the product label, for example:

document/collection_document.xml	urn:esa:psa:bc_spice:document::1.0
miscellaneous/collection_miscellaneous.xml	urn:esa:psa:bc_spice:miscellaneous::2.0
spice_kernels/collection_spice_kernels.xml	urn:esa:psa:bc_spice:spice_kernels::1.0
document/spiceds.html	urn:esa:psa:bc_spice:document:spiceds::1.0
bundle_bc_spice.xml	urn:esa:psa:bc_spice::4.0

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

Please note that this last method is the one that leads into divergences in between the NAIF SPICE Bundles and the PSA SPICE Bundles. For the NAIF SPICE Bundles, the products in the third method could be considered as part of the second method.

9.3. 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) LIDreferences. 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<n>.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.

9.4. 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* se t 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.

9.5. SPICE_kernel product label example

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

spice_kernels/fk/bc_mpo_v002.tf

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-model href="http://pds.nasa.gov/pds4/pds/v1/PDS4_PDS_1500.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_1500.xsd">

    <Identification_Area>
        <logical_identifier>
urn:esa:psa:bc_spice:spice_kernels:fk_bc_mpo_v02.tls</logical_identifier>
        <version_id>1.0</version_id>
        <title>bc_mcpo_v02.xml</title>
        <information_model_version>1.5.0.0</information_model_version>
        <product_class>Product_SPICE_Kernel</product_class>
        <Citation_Information>
            <publication_year>2017</publication_year>
            <keyword>Observation Geometry</keyword>
            <description> SPICE FK file that defines the reference frames for the
BEPICOLOMBO MPO spacecraft. Created by the ESA SPICE Service.</description>
        </Citation_Information>
```

```
    </Identification_Area>
    <Context_Area>
        <Time_Coordinates>
            <start_date_time>2018-10-19T00:00:00Z</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>BEPICOLOMBO</name>
            <type>Mission</type>
            <Internal_Reference>
                <lid_reference>urn:esa:psa:bc:context:investigation:mission.bc
</lid_reference>
                <reference_type>data_to_investigation</reference_type>
            </Internal_Reference>
        </Investigation_Area>
        <Observing_System>
            <Observing_System_Component>
                <name>MPO</name>
                <type>Spacecraft</type>
                <Internal_Reference>

<lid_reference>urn:esa:psa:bc:context:instrument_host:spacecraft.mpo</lid_reference>
                    <reference_type>is_instrument_host</reference_type>
                </Internal_Reference>
            </Observing_System_Component>
        </Observing_System>
        <Target_Identification>
            <name>Mercury</name>
            <type>Planet</type>
            <Internal_Reference>
                <lid_reference>urn:esa:pds:context:target.mercury
 </lid_reference>
                <reference_type>data_to_target</reference_type>
            </Internal_Reference>
        </Target_Identification>
    </Context_Area>
    <Reference_List>
        <Internal_Reference>
            <lid_reference>urn:esa:psa:bc_spice:document:spiceds</lid_reference>
            <reference_type>data_to_document</reference_type>
        </Internal_Reference>
    </Reference_List>
    <File_Area_SPICE_Kernel>
        <File>
            <file_name> spice_kernels/fk/bc_mpo_v002.tf
</file_name>
            <creation_date_time>2017-06-18T00:00:00Z</creation_date_time>
            <file_size unit="byte">5257</file_size>
            <md5_checksum> 25a2fff30b0dedb4d76c06727b1895b1</md5_checksum>
        </File>
        <SPICE_Kernel>
            <offset unit="byte">0</offset>
            <object_length unit="byte">5257</object_length>
            <parsing_standard_id>SPICE</parsing_standard_id>
            <description>SPICE FK file that defines the reference frames for the
```

```
BEPICOLOMBO MPO spacecraft. 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>
```

10. External Standards

This section defines specific requirements and recommendations that should be followed when using non-PDS standards to format the data (e.g. FITS, CDF, ISIS, VICAR).

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

10.1. Specific Requirements and Recommendations for FITS

Data in FITS format shall be compliant with FITS version 3.0.

Specific requirements and recommendations will be included in this section in a future version of this document.

10.2. Specific Requirements and Recommendations for CDF

Data in CDF format shall be compliant with CDF 3.4 ISTP/IACG and shall follow the requirements described in the PDS White Paper “Supporting CDF in PDS4” (*currently under discussion, latest version dated 30-January-2014*). A reduced version of the paper is available online here: PDS4 Archiving of CDF Files.

Specific requirements and recommendations for each mission are included in the appendices of this document.