JPL D-7669, Part 2

Planetary Data System Standards Reference

October 31, 2010
Version 4.0.1 (alpha)

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

Based on:

Information Model Specification
Version 0.1.1.1.c
generated Fri Oct 29 06:29:38 PDT 2010

schema files
built from the PDS4 ontology model V0.1.1.1.c
generated Fri Oct 29 06:29:36 PDT 2010

Protegé “.pont” file
version 3.3.1
build 430
generated Fri Oct 29 06:22:23 PDT 2010

Contents

PDS Standards Reference Change Log

Version	**Section**	**Change**
4.0.1	Appendix A	Added examples for Image_Grayscale

Chapter 1

Introduction

The PDS Standards Reference

1.1 PDS Data Policy

Only data that fully comply with PDS standards will be published as “Conforms to PDS Standards”. When the PDS assists in the preparation of data published in a non-compliant format, PDS participation shall be acknowledged with a statement such as “funded by PDS”. The PDS Management Council makes decisions on compliance waivers. Non-compliant data sets will only be incorporated into the PDS archives as ”safed” data. (Do we wish to drop the statements about waivers?)

1.2 Purpose

The purpose of this document is to serve as a reference document detailing PDS standards used in the preparation of PDS compliant data. This document is to be used in conjuction with other Planetary Data System publications:

- *Archive Preparation Guide* - a brief overview of the archiving process (Is this being replaced by the DPH?)

- *Data Provider’s Handbook* - an introduction and basic “how to” manual for archiving data with the Planetary Data System

- *PDS Policies and Processes* (Do we have a formal document name for this?)
- *Planetary Science Data Dictionary* - contains definitions of the standard classes and attributes used to describe PDS data
- *Proposer's Archiving Guide* - provides basic information on the archiving process (including PDS expectations) to scientists proposing for NASA planetary programs
- tutorials - contact the PDS Operator (pds_operator@jpl.nasa.gov) for more information on available tutorials

1.3 Scope

The information included here constitutes Version 4.0 of the Planetary Data System data preparation standards for producing archive quality data sets. This document covers the conceptual composition of an archive, its physical layout, and the current technology standards used for implementing the data and meta-data.

1.4 Audience

This document is intended primarily to serve the community of scientists and engineers responsible for preparing planetary science data sets for submission to the PDS. These include restored data from the era prior to PDS or from earlier versions of the PDS standards, mission data from active and future planetary missions, and data from earth-based sites. The audience includes personnel at PDS discipline and data nodes, mission principal investigators, and ground data system engineers. This document is intended for use by those people already somewhat familiar with the process of archiving data with the PDS. (Those new to the PDS should first read the *Data Provider's Handbook*.)

1.5 Document Organization

The first section of this document, "Introduction", provides introductory material and citations of other reference documents. The second section, "Conceptual Composition of an Archive" discusses the constituent components of an archive. The third section, "Physical Composition of an

Archive" discusses data format and organization. The fourth section, "Implementation Conventions", discusses the labeling standards employed by the PDS for representation of meta-data and file descriptions. The appendices provide a detailed description of each of the user classes used to describe PDS data.

1.6 Other Reference Documents

The following references are cited in this document:

- Batson, R. M., (1987) "Digital Cartography of the Planets: its Status and Future", *Photogrammetric Engineering & Remote Sensing 53*, 1211-1218.

- Davies, M.E., et al. (1991) "Report of the IAU/IAG/COSPAR Working Group on Cartographic Coordinates and Rotational Elements of the Planets and Satellites: 1991", *Celestial Mechanics, 53*, 377-397.

- Greeley, R. and Batson, R.M. (1990) *Planetary Mapping*, Cambridge University Press, Cambridge, 296p.

- *Guide on Data Entity Naming Conventions*, NBS Special Publication 500-149.

- *PDS 2010 Operations Concept*, JPL D-#### (Available from the PDS).

- Planetary Science Data Dictionary, JPL D-7116 Rev D, July 15, 1996, (Available from the PDS).

- *Planetary Data System Data Preparation Workbook Version 3.1*, JPL D-7669 Part 1, February 17, 1995, (Available from the PDS)

- *Issues and Recommendations Associated with Distributed Computation and Data Management Systems for the Space Sciences*, National Academy Press, Washington, DC, 111p.

1.7 External Standards

External standards which apply to the content of this document:

Consultative Committee for Space Data Systems (CCSDS)

- CCSDS 641.0-B-2 Parameter Value Language Specification (CCSD0006 and CCSD0008) (also available as ISO 14961:2002)

Institute of Electrical and Electronics Engineers (IEEE):

- IEEE 754-2008 *Standard for Binary Floating-Point Arithmetic*

International Standards Organization (ISO):

- ISO 646:1991 *ISO 7-bit coded character set for information interchange*
- ISO 8601:1988 *Data Element and Interchange Formats – Representations of Dates and Times* (Shouldn't this be 8601:2004? - EDR)
- ISO/IEC 11179-3:2003 *Metadata registries (MDR) – Part 3: Registry metamodel and basic attributes*
- ISO/IEC 11404:2007 *General-Purpose Datatypes (GPD)*
- ISO 14721:2003 *Open archival information system – Reference model*
- ISO/TS 15000-3:2004 *Electronic business eXtensible Markup Language (ebXML) – Part 3: Registry information model specification (ebRIM)*
- ISO/TS 15000-4:2004 *Electronic business eXtensible Markup Language (ebXML) – Part 4: Registry services specification (ebRS)*

World Wide Web Consortium (W3C):

- *Extensible Markup Language (XML) 1.1* 2nd ed., August 16, 2006
- *XML Schema Part 0: Primer* 2nd ed., October 28, 2004
- *XML Schema Part 1: Structures* 2nd ed., October 28, 2004
- *XML Schema Part 2: Datatypes* 2nd ed., October 28, 2004

1.8 Online Document Availability

All PDS documents pertaining to archive preparation are available online. Information on accessing these references may be found on the PDS website at the following URL:

http://pds.nasa.gov

To obtain a copy of these documents or for questions concerning these documents, contact the PDS Operator (at pds_operator@jpl.nasa.gov, 818-393-7165) or any PDS data engineer.

The examples provided throughout this document are based on both existing and planned PDS archive products, modified to reflect the current version of the PDS Standards. Discipline-specific extensions to the high-level classes defined in this document are created and augmented from time to time, as user community need arises. To check the current status of any discipline- or mission-specific class definition, consult a PDS data engineer.

Additional schemata and examples are available online at the following URL:

http://pds.nasa.gov/repository/pds4/SCHEMA/

Chapter 2

Overview

Some of this needs to be moved to the Product Structure Overview in Chapter 14, while this section expands more on the overall archive contents.

In order to successfully utilize scientific data, it is necessary to have access to both the primary (observational) data and ancillary data, software, and documentation needed to understand and use the observations. In order for a PDS archive to be complete, it must therefore contain all the necessary ancillary information.

The Planetary Data System is now, more than ever, a *system* not only for the long-term storage of planetary data but for its organization and retrieval, as well. In an environment in which increasingly sophisticated instruments are accumulating ever larger volumes of data, it has become critical to have a methodical system for locating and acquiring not only individual digital files, but also all of the ancillary data needed to correctly interpret and understand the information stored in those files.

In response to this need, the PDS4 standards are now built solidly on an underlying, rigorously defined data model. By adhering to this strict model, we are able to build a versatile tracking and search engine with the flexibility to handle many types of data and the resilience to be updated quickly in response to changes and updates.

Within the data model, all *things* are treated as one of three types of objects: physical, digital, or conceptual.

Physical objects are those which are tangible. Planets, spacecraft, and instruments are examples of physical objects which are tracked in the Planetary Data System.

Digital objects make up the bulk of the PDS holdings. These are the types of things one traditionally thinks of as planetary science data: digital files containing raster images, binary or character tables, documents, and software.

Conceptual objects are intangibles: missions, observing campaigns, and academic institutions are all examples of this type of object.

An object, accompanied by a description of that object, is called a *tagged digital object* (in the case of digital objects) or a *tagged nondigital object* (in the case of physical and conceptual objects).

One or more tagged objects, grouped together and described with a single label, constitute a *product*. This is true for all three types of objects mentioned above. Thus a planet and the description of that planet is considered to be a product no less than an image and the description of that image.

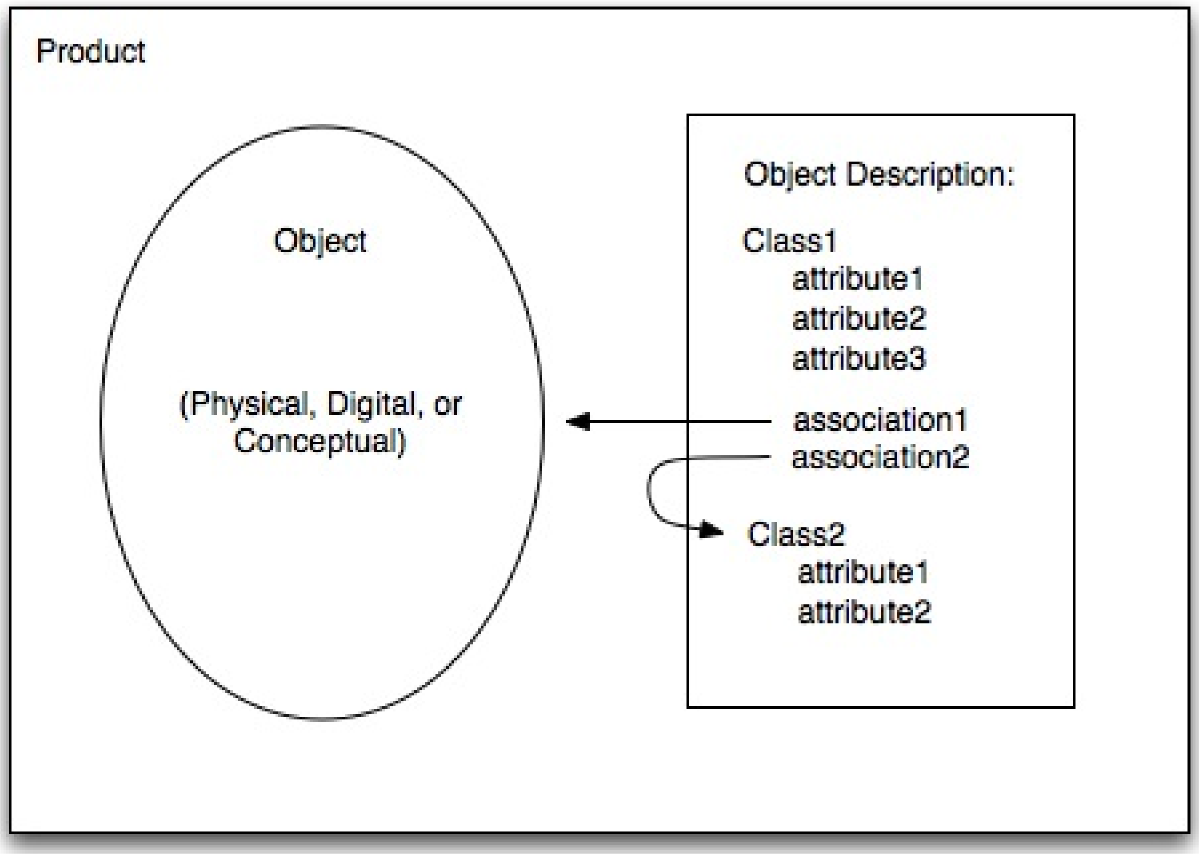
Furthermore, a *list* of products (accompanied by a description of the list) is also considered to be a product. Thus one may identify and track logical sets of products.

Within the PDS archive, three types of products are recognized: *standard* products, *collections*, and *bundles*. Each of these are describe in more detail in the following sections.

Each type of object recognized by the PDS is described in the data model with a *class*. Thus the Mission class is used to describe a mission (conceptual) object.

Classes contain *attributes* and *associations*. The attributes are used to provide information about the objects. For example, the Mission class contains start_date and stop_date attributes that provide information about a particular mission object.

Associations describe relationships between the class and other entities, be they objects or other classes.



Classes can be either *generic* or *specific*. The generic Image_Grayscale class is a high level description that can be applied to all grayscale images. Need an example and explanation of a specific class.

Needs something more before launching into next section.

2.1 Standard Products

The lowest level product in the Planetary Data System (PDS) is referred to as a standard product. A standard product comprises one or more objects constituting (typically) a single observation or a single document.

2.1.1 Browse Products

Browse products are an optional but often helpful part of an archive. They can encompass many different types of products. They might include low resolution versions of large image files, maps showing the locations of data acquisition, graphical representations of tabular data, or any other product that can facilitate in the visualization or quick browsing of data included in an archive.

Browse products may be either stand-alone products or may be included as a browse object within a primary product.

2.1.2 Calibration Products

All of the calibration files necessary to properly process and understand raw data must be included in an archive. This includes both raw calibration data as well as any derived calibration files. Flat field, dark current, and other similar on-board calibration images may optionally be considered to be calibration information rather than regular data products.

2.1.3 Cartographic Products

It is required that all PDS archives include a clear specification of any relevant coordinate systems and frames. Information specific to each data product should be included in the product labels or in an updatable tabular form within the archive, but information relevant to the entire archive should be included as documentation.

See the chapter on Cartographic Standards in this document for full details on the specification of coordinate systems in PDS products.

2.1.4 Catalogs and Database Products

Many data providers choose to provide supplementary information about the data in their archive that is not contained in each data product label. This may be information that is likely to be volatile and so is inappropriate for inclusion in data product labels, or information that is applicable to the entire archive. This information can be included in an archive in the form of a spreadsheet or fixed-length table, appropriate for loading into a spreadsheet application or relational database.

2.1.5 Context Products

Context products include information about the environment in which the data were acquired. They include detailed information about the instrument that acquired the data, the spacecraft, telescope, or other instrument host on which the instrument was mounted and the target (planet or other body) that was the subject of the data. It can also include information about the mission or campaign that operated the instrument and the institution or individuals who produced the archive products.

2.1.6 Data Dictionary Products

Under PDS4, each archive is required to include a dictionary containing a listing and definition for every attribute used in the meta-data describing each data product. This dictionary will ensure that the definitions for each attribute, as they were understood at the time of the product creation, are captured and retained for data users. It will also provide a mechanism for validation of the data, since limitations such as minimum and maximum values and valid data types will be included in the dictionary for each attribute, where appropriate.

2.1.7 Document Products

Documentation necessary for an understanding of the data should also be included in the archive. This can include project documentation such as data management plans, archive plans, and product software interface specifications. It can also include documentation introducing a user to an archive collection, explaining the process by which a particular set of data was produced, or any errors or problems encountered in the production of the data. Information about the calibration process should be included here.

2.1.8 Geometry Products

All PDS archives must contain SPICE or other geometry information sufficient to fully determine the precise timing and location of all data acquisition or other important events with respect to the target body, the instrument acquiring the data, and any illumination sources. Other geometry information that may be included in archives includes gazetteers of observed features.

2.1.9 Science Data Products

In this document, the term *data products* or *science data products* is used to refer to the primary output from instruments making scientific observations or measurements.

2.1.10 Software

TBD

2.2 Collections

The intermediate level product is referred to as a *collection*. A collection consists of a list of references to standard products, all of which are closely related and of a particular type. Collections are identified as either *primary* or *secondary*.

2.2.1 Primary Collections

A primary collection is essentially a list of closely related products of a single broad type, either observational or for support purposes. Products that will nearly always be accessed, processed, or otherwise used together should be identified with a single primary collection. The primary collection ID is used to generate standard product IDs, and will become a permanent entity in the PDS archive holdings.

For example, a primary collection might consist of all the raw (uncalibrated) images acquired by the Narrow Angle Camera mounted on the Voyager 1 spacecraft during its fly-by of Saturn. Another example might consist of all the calibrated meteorological data collected by the ASI/MET instrument package during the surface phase of the Mars Pathfinder mission. Primary collections can also consist of highly derived data, such as catalogs of objects provisionally identified with known asteroids or tables of elemental abundances for an in-situ geological sample.

Alternatively, a primary collection might be made up of all the SPICE data or a particular type related to a specific mission or all of the mission or target products ever submitted to the PDS.

standard product can only belong to one primary collection primary collections may only reference lidvids

2.2.2 Secondary Collections

Secondary collections are lists identifying products archived elsewhere in a primary collection. Secondary collection IDs are not usually used as a component in the formation of standard product IDs.

Examples of secondary collections might include all of the Voyager 1 NAC raw images that included Saturn's rings within their field of view, or all of the Voyager, Galileo, Hubble, and ground-based observations of the Shoemaker-Levy 9 impact into Jupiter.

Secondary collections can (and frequently will) be included alongside primary collections in archive bundles. For example, a secondary collection consisting of all the archive-related documents produced by the Stardust mission, or all of the SPICE files necessary to fully describe the geometry of a certain set of observations would likely be included in an archive bundle built around a primary collection of observational data.

secondary collections may either lids, lidvids, or both

(Anne: different view of the existing data)

2.3 Bundles

The highest level product is referred to as a *bundle*. Like collections, bundles consist of a list of references to products; however in this case, the referenced products may themselves be collections. A bundle identifies all of the collections and standard products necessary to perform useful science analysis on the data contained therein.

2.3.1 Archive Bundles

Fundamentally, a PDS archive is an aggregation of products with a common origin, history, or application. It includes primary (observational) data plus the ancillary data, software?, and documentation needed to understand and use the observations. The *archive bundle* is the entity used to associate all of these pieces. An archive bundle constitutes the list of all collections and standard products that will be delivered by a data provider to the PDS. (A large mission may deliver several archive bundles, distinguished by different instruments, mission phases, etc.)

In order for an archive bundle to be complete, it must contain references to all the necessary ancillary information. Not all forms of ancillary information are necessary for the interpretation of different types of data, so you will need to seek guidance from your discipline node in determining which types of ancillary files are required for your archive.

For large collections gathered over time, data may be broken up into deliveries. products and collections are physically organized in the archive bundle relationship. PDS node will merge the incoming files with the existing files following the instructions in the delivery package (see delivery packaging)

2.3.2 Special Bundles

Context Bundle

Certain primary collections containing context products are maintained as members of the context bundle. There is only one context bundle in the PDS. The member collections of the context bundle are the following:

INVESTIGATION (or MISSION)
NODE
OBSERVING_SYSTEM (or INSTRUMENT_HOST + INSTRUMENT)
PERSONNEL
PUBLICATIONS
SOFTWARE (TBD)
TARGET

Manifest Bundles

Does this exist? (It's currently in the model, but will we be keeping it?)

SPICE Bundles

A SPICE bundle is a bundle that references primarily SPICE collections. It will usually contain the complete set of SPICE data for a given mission.

System Bundles

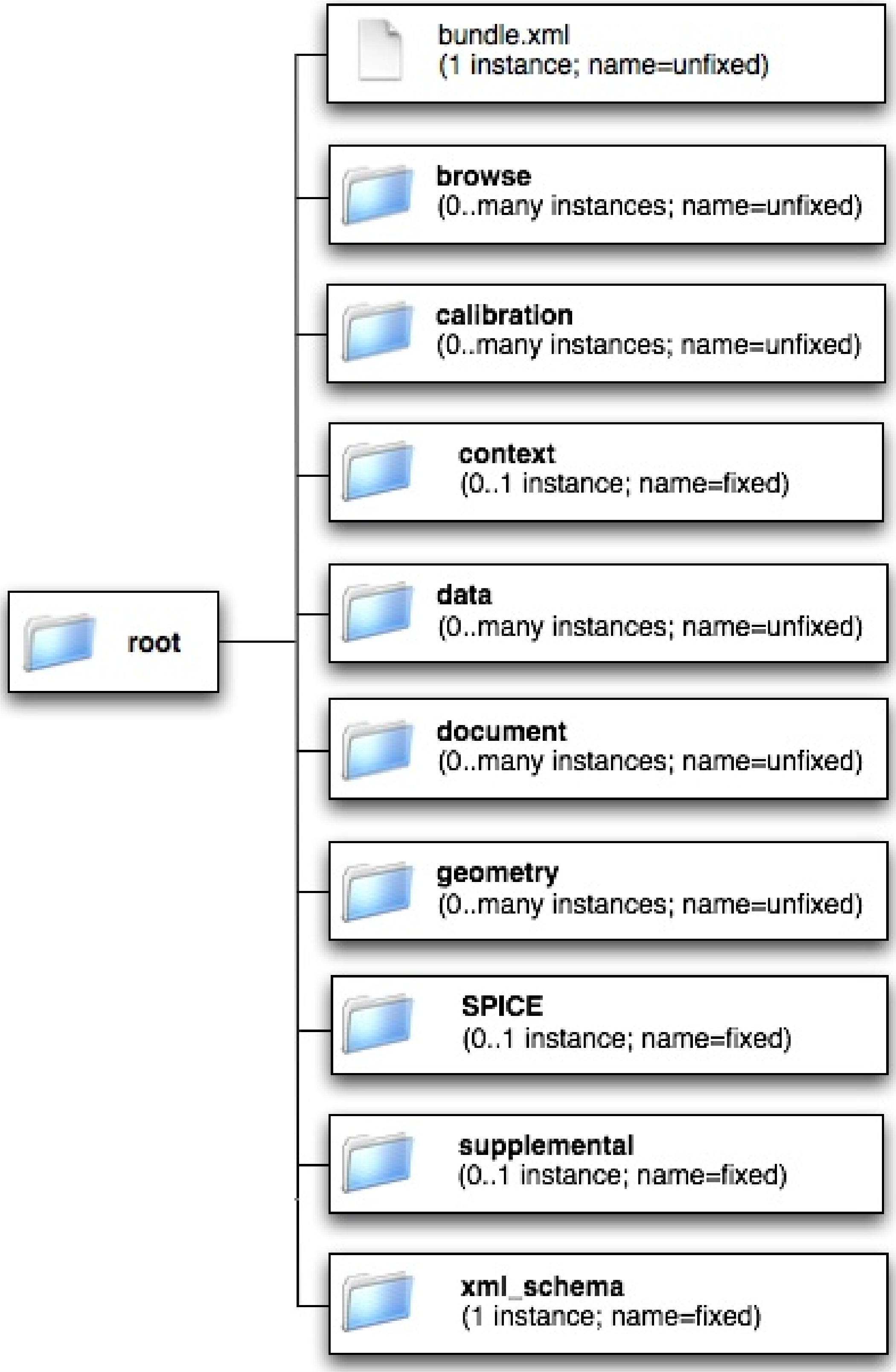
Don't include, since this is primarily for internal, housekeeping use?

Chapter 3

Physical Organization of an Archive

3.1 Directory Contents and Organization

Data delivered to at the PDS shall be assigned to directories based on product type. The following sections describe the contents of the root directory, followed by the contents of the subdirectories in alphabetical order. For descriptions of the types of products contained in each directory, see section ref. The root directory is the top-level directory of a volume.


bundle.xml
(1 instance; name=unfixed)
browse
(0..many instances; name=unfixed)
calibration
(0..many instances; name=unfixed)
context
(0..1 instance; name=fixed)
root
data
(0..many instances; name=unfixed)
document
(0..many instances; name=unfixed)
geometry
(0..many instances; name=unfixed)
SPICE
(0..1 instance; name=fixed)
supplemental
(0..1 instance; name=fixed)
xml_schema
(1 instance; name=fixed)

3.1.1 Root Directory Files

Data providers must deliver data organized in the following directory structure. PDS discipline nodes may or may not choose to retain that structure in their online repositories.

aareadme

This file contains a brief overview of the contents and organization of the associated archive, general instructions for its use, and contact information. The name has been chosen so that it will be listed first in an alphabetical directory listing.

archive manifest

This file contains a listing of every product that is a member of the archive, along with its directory path and file name. Checksums are optionally included for the sake of archive integrity checks.

errata

This file identifies and describes errors and/or anomalies found in the archive.

3.1.2 Browse Directory

This directory contains "quick-look" products designed to facilitate use of the archive. If appropriate, this directory may contain sub-directories paralleling the structure of the data subdirectories.

3.1.3 Calibration Directory

This directory contains calibration data and files necessary for the calibration of the science data products. TBD based on work with Lisa.

3.1.4 Context Directory

This directory contains all context products associated with the archive. These are the products identified in the Product_Cross_Reference_Area of the data products in the archive.

3.1.5 Data Directory

This directory contains the archive data products. This directory may be further sub-divided to prevent over-crowding of directories and facilitate archive organization.

3.1.6 Document Directory

3.1.7 Geometry Directory

Non-SPICE geometry data

(What is the relationship between cartographic information and the geometry directory?)

3.1.8 SPICE Kernel Directory

(all SPICE data inclu (same directory structure as for SPICE Bundles)

ck ek fk ik lsk pck sclk spk

3.1.9 Supplemental Directory

This directory contains any supplemental meta-data catalogs, database dumps, indices, or spreadsheets deemed by the data provider to be useful to the interpretation of the data in the archive.

ex. modification history errata update anaglyph pairs footprint files database dumps

3.1.10 XML Schema Directory

3.2 Data Transfers

Transfers from point to point within and external to the PDS can include packaging mechanisms like ZIP, gzip, tar, physical media such as thumb drives, external hard drives, etc.

Coordinate with the node. Be reasonable.

The transfer mechanism is worked out between the node and the data provider.

All transfers require the inclusion of a manifest. (MD5 manifest)

3.3 Delivery of Accumulating Archives

The following stuff should go in the APG or DPH?

two options

1. data provider delivers complete, updated collection product (inventory), along with new standard products

2. data provider delivers fragment inventory along with new products, which is integrated by node with previous version of collection product to create new version

Chapter 4

Product Structure Overview

Material here will be supplemented with text currently in Chapter 2; see note at the beginning of that chapter.

PDS products consist of one or more objects and a detached product label. Product labels are required to describe the contents and structure of the objects. PDS product labels are formatted in XML.

There is always a one-to-one correspondence between a PDS product and its label.[1] However, a PDS product may consist of multiple files.

As has been stated elsewhere, in PDS4 everything is considered to be a product. Thus, in addition to traditional data products such as images, tables, and histograms, missions and spacecraft are now also treated as products. There are three

4.1 Terminology: Products, Objects, Classes, and Files

[1]There is a possible exception to this rule in the case of *remote* labels; however, the acceptance of this particular kind of product in PDS4 archives has not yet been decided upon.

Chapter 5

Labels

PDS product labels are required for describing the contents and format of each individual product within an archive. Labels are populated using a standard set of classes, elements, and standard values for the elements. These classes, elements, and standard values are defined in the *Planetary Science Data Dictionary* (PSDD). Appendices A through C of this document provide general descriptions and examples of the the classes and their data element used in labeling PDS products.

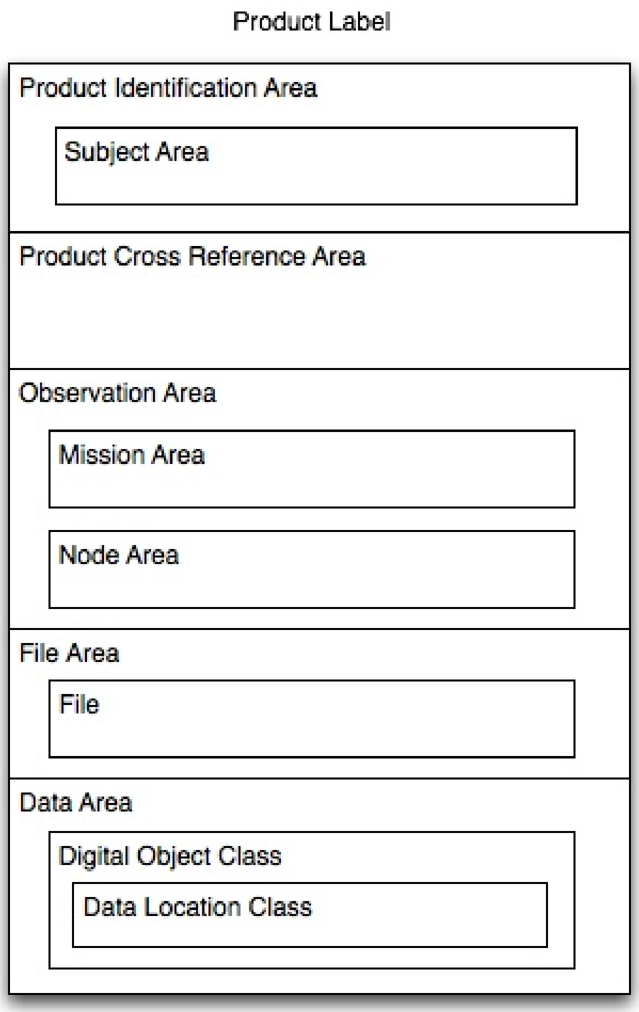
Under the PDS4 standard, all product labels are detached from the data files they describe. There is one label file for every product. Each label file may describe one or more data files.

For details on the specific grammar used to implement the labels, see section IV of this document, *Implementation Conventions*.

PDS product labels have a general structure that is used for the vast majority of products.

- Product Identification Area
- Product Cross Reference Area
- Observation Area
- File Area
- Data Area

Figure x.x shows how this general structure appears in a product label.


Product Label
Product Identification Area
Subject Area
Product Cross Reference Area
Observation Area
Mission Area
Node Area
File Area
File
Data Area
Digital Object Class
Data Location Class

5.1 The Product_Identification Area

5.1.1 The Subject Area

5.2 The Product_Cross_Reference Area

5.3 The Observation Area

5.4 The File Area

The "File_Area" of a label is used to describe the system files containing the digital objects described in the "Data_Area". It must contain a distinct File class for each file containing data for this product.

Each File class must be described by a local_identifier, unique within the product label. This identifier will be used in the Data_Location class (described below) to tie together a Digital Object class with the digital data it describes. Local_identifiers may be re-used from one label to another within a collection.

5.5 The Data Area

5.5.1 The Digital Object Classes

The Data_Location Class

Chapter 6

Fundamental Data Structures

There are four fundamental data structures that may be used for archiving data in the PDS. All data products delivered to the PDS must be constructed from one or more of these structures. These four fundamental structures are described using four base classes: Table_Base (used to describe heterogeneous repeating records of scalars), Array_Base (used for homogeneous n-dimensional arrays of scalars), Parsable_Byte_Stream, and Encoded_Byte_Stream. All other digital object classes in the PDS are derived from one of these four.

6.1 Table Base

In the PDS, column data may be stored in either of two different structures, either a fixed-width ”table”, or a variable-length ”spreadsheet” (delimited stream). The latter is described in section **??**, ”Unencoded Stream Base”.

Conceptually, tabular data files consist of a series of named columns containing both data locations and data values. The data may consist of both numbers and text strings.

The TABLE object is a uniform collection of rows containing ASCII or binary values stored in columns. The INTERCHANGE_FORMAT keyword is used to distinguish between TABLEs containing only ASCII columns and those containing binary data. The rows and columns of the TABLE object provide a natural correspondence to the records and fields often defined in interface specifications for existing data products. Each field is defined as a fixed-width COLUMN object; the value of the COLUMNS keyword is the total number of COLUMN objects defined in the label. All TABLE objects must have fixed-width records.

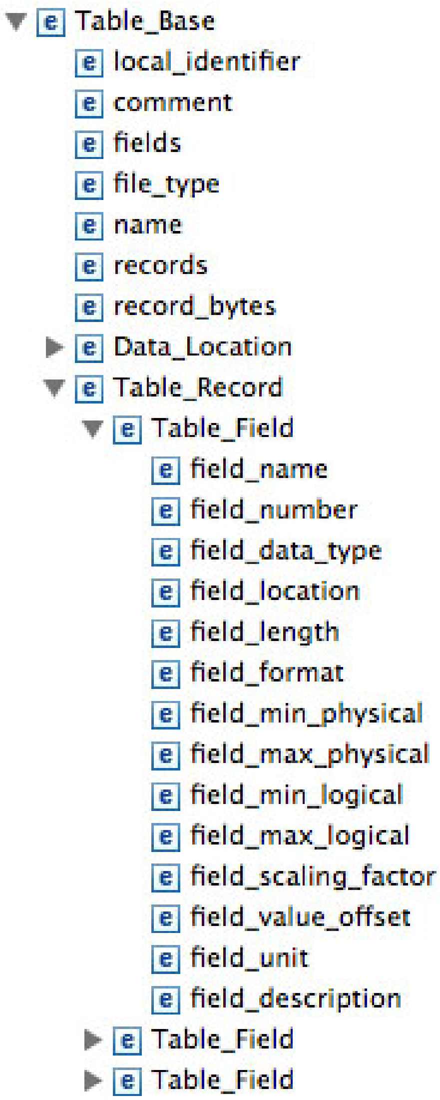


Figure 6.1: Table_Base

Physically, the data are stored as a sequence of repeating records where each record is terminated by both the CR and LF characters (ASCII characters xxx and yyy, respectively). Since both the column lengths and the record lengths are fixed, column values are identifiable by position alone. However, column delimiters may optionally be included.

The data may be represented in any of binary, ASCII, or UTF-8 values.

(include picture here)

6.2 Array Base

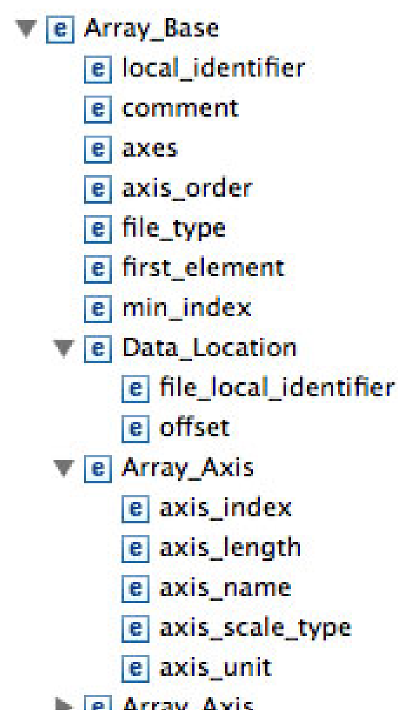
The second of the four basic structures consists solely of homogeneous pixels in fixed-length arrays in any number of dimensions, although two- and three-dimensional data are the most common. These arrays are described using a subclass of the Array_Base class, which has two associated classes (besides the Data_Location class), the Array_Axis and Array_Element classes.

The structure of the homogeneous pixels or ”elements” in an array is described using the Array_Element class. The amount of physical storage space required for each element is describe by the element_bytes attribute. The storage order and interpretation of the bytes comprising each element is indicated by the element_type attribute, which typically has values like ”SignedMSB2” or ”IEEE754Double”. (See the Planetary Science Data Dictionary for a complete listing of valid data types.)

The physical interpretation of the element values is described using the remaining three attributes. The element_unit attribute describes the ...??? The element_scaling_factor and element_offset_value attributes provide the numbers by which an element value must be transformed, using the following formula:

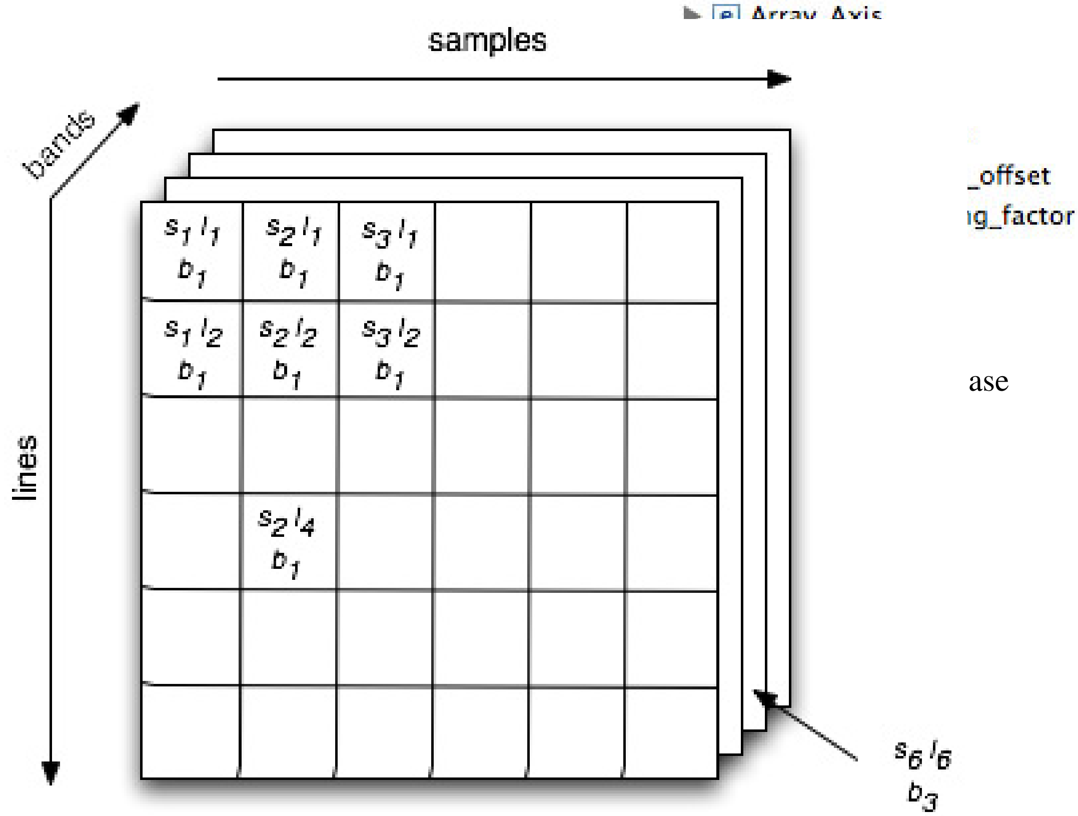
true value = (stored value x scaling factor) + offset value

The characteristics of each array axis are modeled using an Array_Axis class (there must be one Array_Axis class present in the label for each dimension of the array). The axis_length attribute provides the number of elements in the array along each particular axis. The axis_name attribute unambiguously labels each axis, and may have a value as simple as ”line”, ”sample”, or ”spectral band”. (Axis_unit and axis_scale_type still have significant problems.)



Finally, the Array_Base class, or one of its subclasses, wraps the whole thing together:

The axis_order attribute specifies the order in which the elements of the multi-dimensional array are stored. The PDS requires that the data be stored in a fastest-varying-pixel-first order. In other words, for the following conceptual representation of a 3-dimensional array:



the data stream must contain the pixels in the following order:

$s_1l_1b_1, s_2l_1b_1, s_3l_1b_1...s_1l_2b_1, s_2l_2b_1, s_3l_2b_1...s_1l_1b_2, s_2l_1b_2, s_3l_1b_2...$

This ordering of the elements is indicated by specifying an axes_order of "FIRST_INDEX_FASTEST". (I am not convinced that we have yet completely resolved the issue of tying together the axis order with the named Array_Axis classes. – EDR)

6.3 Parsable Byte Stream

XML / HTML / spreadsheet xhtml text files telemetry stream format files - conceivably

Alternatively, data may be stored in a variable-length structure. The PDS refers to these structures as "spreadsheets".

6.4 Encoded Byte Stream

The encoded stream base structure in the PDS is a byte stream that may only be interpreted after it has been "decoded", according to some well known standard. "Encoded" data may include data that has been compressed and needs to be decompressed before interpretation. Alternatively, it may include data encoded in some form of binary format, such as the Portable Document Format (PDF).

In order to interpret an encoded stream base, reading software would need to determine the value of the external_standard_id attribute and access the referenced standard for information on how to parse the byte stream. It is PDS policy that only publicly available, open source, widely accepted standards may be used for the encoding of data within the PDS.

Chapter 7

Data Storage Types

7.1 Character Data Types

Need to update

Value	Description
ANYURI	
ASCII_BOOLEAN_ONOFF	On / Off indicator; valid values ”ON” and ”OFF”
ASCII_BOOLEAN_TF	True / False indicator; valid values ”T” and ”F”
ASCII_BOOLEAN_YN	A boolean in ASCII format; valid values ”Y” and ”N”
ASCII_CHARACTER	ASCII character string
ASCII_CHARACTER_STRING	
ASCII_DATE	ASCII character string representing a date in the extended date format of ISO 8601: YYYY-MM-DD
ASCII_DATE_DOY	

ASCII_DATE_TIME	
ASCII_DATE_YMD	
ASCII_ENUMERATED	255 max characters
ASCII_IDENTIFIER	
ASCII_INTEGER	ASCII character string representing an integer.
ASCII_LOCAL_TIME	
ASCII_LOGICAL_IDENTIFIER	
ASCII_OCTET	
ASCII_REAL	ASCII character string representing a real number;
ASCII_TIME	
ASCII_VERSION_ID	
CHARACTER_LITERAL	
UTF8_CHARACTER	

7.2 Binary Data Types

7.2.1 Integers

Signed LSB Integers

This section describes signed integers stored in Least Significant Byte first (LSB) order. In this section the following definitions apply:

b0 – b7 Arrangement of bytes as they appear when reading a file (e.g., read byte b0 first, then b1, b2 and b3, up through b7)

i-sign Integer sign bit – bit 7 in the highest order byte

i0 – i7 Arrangement of bytes in the integer, from lowest order to highest order. The bits within each byte are interpreted from right to left (e.g., lowest value = bit 0, highest value = bit 7), in the following way:

8-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7
In i1, bits 0-7 represent 2^8 through 2^{15}
In i2, bits 0-7 represent 2^{16} through 2^{23}
In i3, bits 0-7 represent 2^{24} through 2^{31}
In i4, bits 0-7 represent 2^{32} through 2^{39}
In i5, bits 0-7 represent 2^{40} through 2^{47}
In i6, bits 0-7 represent 2^{48} through 2^{55}
In i7, bits 0-6 represent 2^{56} through 2^{62}

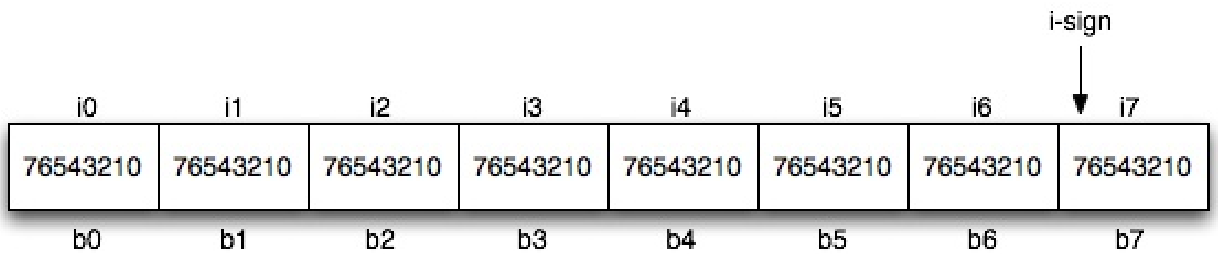
4-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7
In i1, bits 0-7 represent 2^8 through 2^{15}
In i2, bits 0-7 represent 2^{16} through 2^{23}
In i3, bits 0-6 represent 2^{24} through 2^{30}

2-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7
In i1, bits 0-6 represent 2^8 through 2^{14}

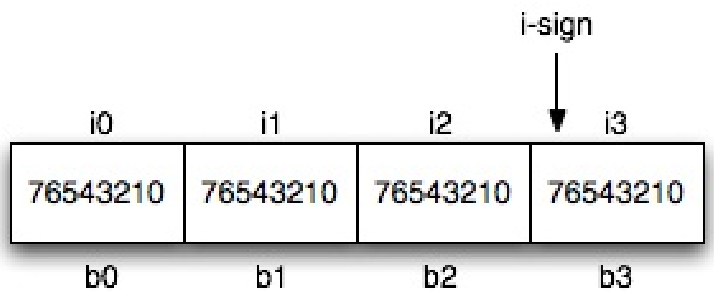
1-byte integers:
In i0, bits 0-6 represent 2^0 through 2^6

All negative values are represented in two's complement.

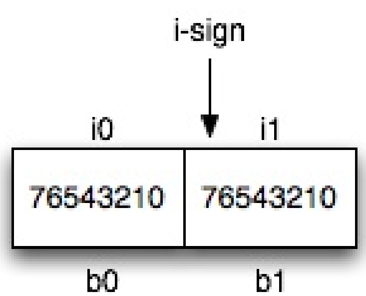
SignedLSB8



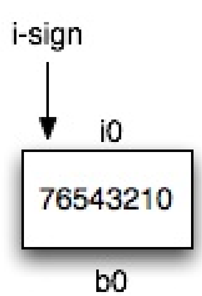
SignedLSB4



SignedLSB2



SignedByte



Unsigned LSB Integers

This section describes unsigned integers stored in Least Significant Byte first (LSB) format. In this section the following definitions apply:

b0 – b3 Arrangement of bytes as they appear when reading a file (e.g., read byte b0 first, then b1, b2 and b3)

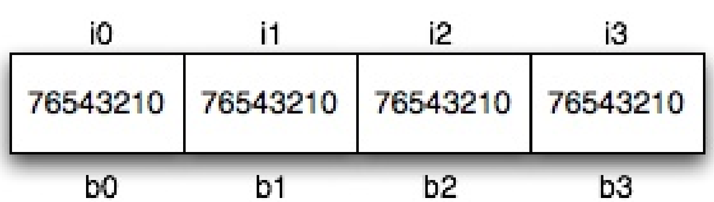
i0 – i3 Arrangement of bytes in the integer, from lowest order to highest order. The bits within each byte are interpreted from right to left (e.g., lowest value = bit 0, highest value = bit 7), in the following way:

4-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7
In i1, bits 0-7 represent 2^8 through 2^{15}
In i2, bits 0-7 represent 2^{16} through 2^{23}
In i3, bits 0-7 represent 2^{24} through 2^{31}

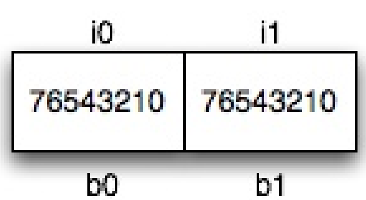
2-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7
In i1, bits 0-7 represent 2^8 through 2^{15}

1-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7

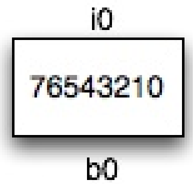
UnsignedLSB4



UnsignedLSB2



Unsigned Byte



Signed MSB Integers

This section describes the signed integers stored in Most Significant Byte first (MSB) order. In this section the following definitions apply:

b0 – b7 Arrangement of bytes as they appear when read from a file (e.g., read b0 first, then b1, b2, and b3 up through b7)

i-sign Integer sign bit – bit 7 in the highest order byte

i0 – i7 Arrangement of bytes in the integer, from lowest order to highest order. The bits within each byte are interpreted from right to left (e.g., lowest value = bit 0, highest value = bit 7) in the following way:

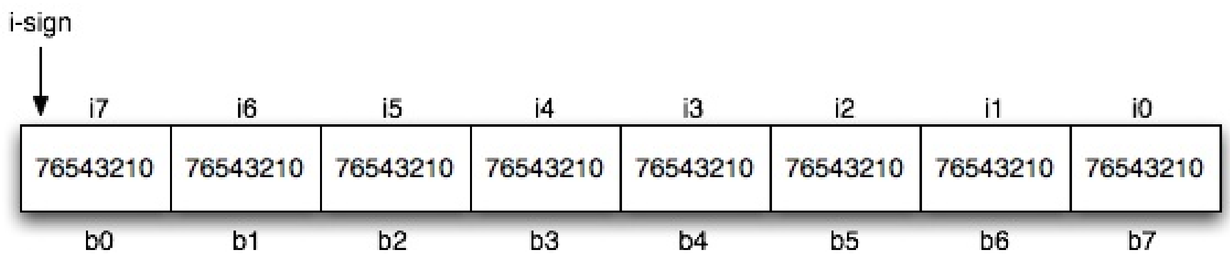
8-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7
In i1, bits 0-7 represent 2^8 through 2^{15}
In i2, bits 0-7 represent 2^{16} through 2^{23}
In i3, bits 0-7 represent 2^{24} through 2^{31}
In i4, bits 0-7 represent 2^{32} through 2^{39}
In i5, bits 0-7 represent 2^{40} through 2^{47}
In i6, bits 0-7 represent 2^{48} through 2^{55}
In i7, bits 0-6 represent 2^{56} through 2^{62}

4-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7
In i1, bits 0-7 represent 2^8 through 2^{15}
In i2, bits 0-7 represent 2^{16} through 2^{23}
In i3, bits 0-6 represent 2^{24} through 2^{30}

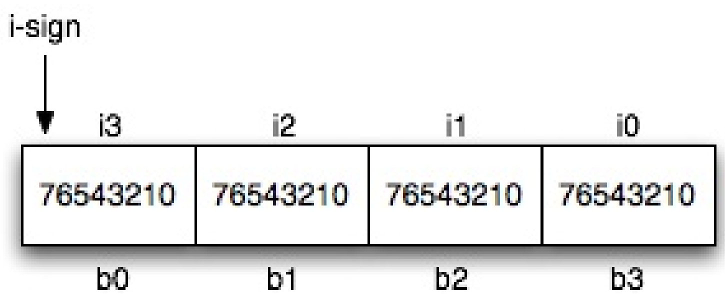
2-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7
In i1, bits 0-6 represent 2^8 through 2^{14}

1-byte integers:
In i0, bits 0-6 represent 2^0 through 2^6

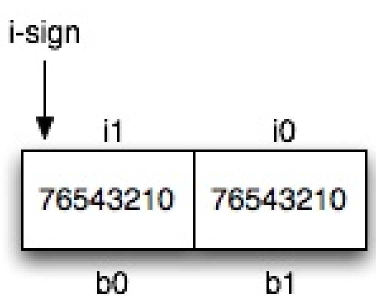
SignedMSB8



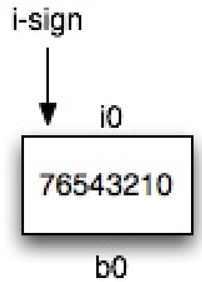
SignedMSB4



SignedMSB2



SignedByte



Unsigned MSB Integers

This section describes unsigned integers stored in Most Significant first (MSB) format. In this section the following definitions apply:

b0 – b3 Arrangement of bytes as they appear when read from a file (e.g., read b0 first, then b1, b2, and b3)

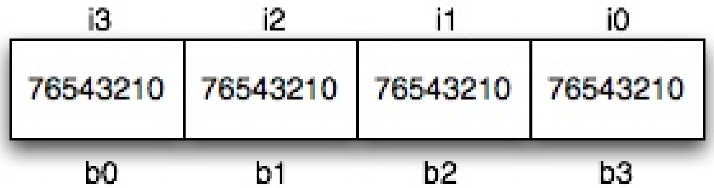
i0 – i3 Arrangement of bytes in the integer, from lowest order to highest order. The bits within each byte are interpreted from right to left (e.g., lowest value = bit 0, highest value = bit 7) in the following way:

4-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7
In i1, bits 0-7 represent 2^8 through 2^{15}
In i2, bits 0-7 represent 2^{16} through 2^{23}
In i3, bits 0-7 represent 2^{24} through 2^{31}

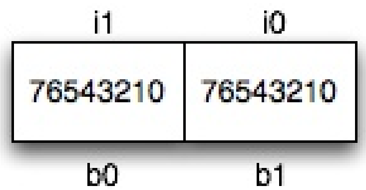
2-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7
In i1, bits 0-7 represent 2^8 through 2^{15}

1-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7

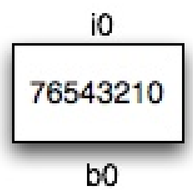
UnsignedMSB4



UnsignedMSB2



UnsignedByte



7.2.2 Reals

This section describes the internal format of IEEE-format floating-point numbers. In this section the following definitions apply:

b0 – b9 Arrangement of bytes as they appear when read from a file (e.g., read b0 first, then b1, b2, b3, etc.)

m-sign Mantissa sign bit

int-bit In 10-byte real format only, the integer part of the mantissa, assumed to be "1" in other formats, is explicitly indicated by this bit

e0 – e1 Arrangement of the portions of the bytes that make up the exponent, from lowest order to highest order. The bits within each byte are interpreted from right to left (e.g., lowest value = rightmost bit in the exponent part of the byte, highest value = leftmost bit in the exponent part of the byte) in the following way:

8-bytes (double precision):
In e0, bits 4-7 represent 2^0 through 2^3
In e1, bits 0-6 represent 2^4 through 2^{10}

Exponent bias = 1023

4-bytes (single precision):
In e0, bit 7 represents 2^0
In e1, bits 0-6 represent 2^1 through 2^7

Exponent bias = 127

m0 – m7 Arrangement of the portions of the bytes that make up the mantissa, from highest order fractions to the lowest order fraction. The order of the bits within each byte progresses from left to right, with each bit representing a fractional power of two, in the following way:

8-bytes (double precision):
In m0, bits 3-0 represent $1/2^1$ through $1/2^4$
In m1, bits 7-0 represent $1/2^5$ through $1/2^{12}$
In m2, bits 7-0 represent $1/2^{13}$ through $1/2^{20}$
In m3, bits 7-0 represent $1/2^{21}$ through $1/2^{28}$
In m4, bits 7-0 represent $1/2^{29}$ through $1/2^{36}$
In m5, bits 7-0 represent $1/2^{37}$ through $1/2^{44}$
In m6, bits 7-0 represent $1/2^{45}$ through $1/2^{52}$

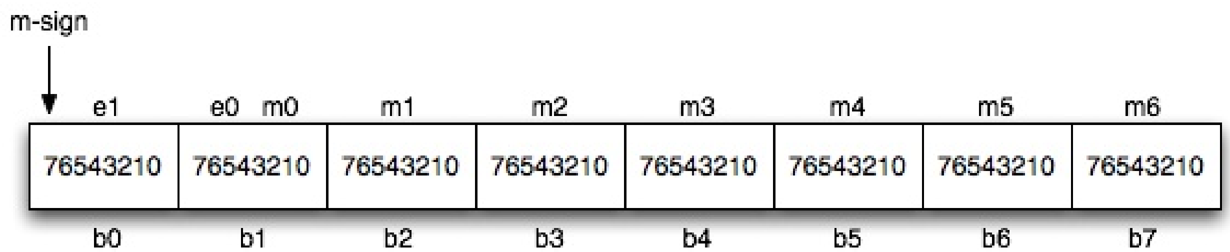
4-bytes (single precision):
In m0, bits 6-0 represent $1/2^1$ through $1/2^7$
In m1, bits 7-0 represent $1/2^8$ through $1/2^{15}$
In m2, bits 7-0 represent $1/2^{16}$ through $1/2^{23}$

The following representations all follow this format:

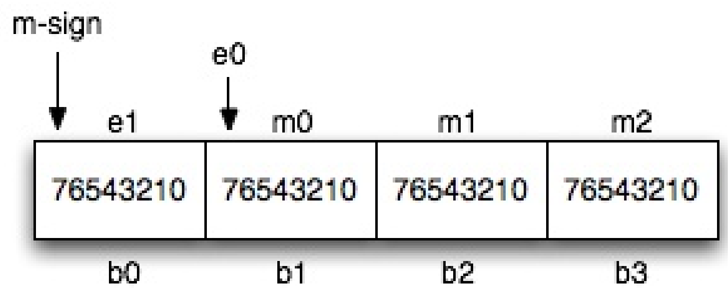
$$1.\textit{mantissa} \text{ x } 2^{(exponent-bias)}$$

Note that the integer part ("1.") is implicit in all formats as described above. In all cases the exponent is stored as an unsigned, biased integer (that is, the stored exponent value - bias value = true exponent).

IEEE754Double



IEEE754Single



7.2.3 Complex

IEEE complex numbers consist of two IEEE_REAL format numbers of the same precision, contiguous in memory. The first number represents the real part and the second the imaginary part of the complex value.

Chapter 8

Astronomical Nomenclature

IAU and USGS

TBD

Chapter 9

Browse Products

TBD

Chapter 10

Calibration Standards

TBD - Will not be completed for first release.

Chapter 11

Cartographic Standards

Pulled fairly verbatim from v3.8 of StdRef. Needs to be adapted to PDS4.

11.1 Introduction

To facilitate use, exchange and integration of its products, the PDS follows accepted planetary cartographic standards for data products where they exist. Because such standards evolve as new data and knowledge are acquired, there are advisory groups charged with developing and periodically updating standards for coordinate systems. All data providers for PDS products should follow accepted standards and be aware of current NASA and international recommendations on cartographic coordinate systems and conventions relevant to their bodies of interest. An absolute requirement for all PDS products is that relevant coordinate systems and frames be clearly specified in product labels and supporting documents. This chapter specifies, as of late 2008, the authoritative sources for international cartographic standards, provides a summary of major cartographic elements to which those standards apply, and identifies the primary standards that PDS has adopted.

11.1.1 International and NASA Advisory Groups for Cartographic Standards

The primary international body for coordinate systems in the Solar System is the International Astronomical Union (IAU). The IAU has recognized the International Celestial Reference System (ICRS) as the defining inertial reference system and its associated International Celestial Reference

Frame (ICRF) (Ma et al., 1998) as the defining frame for that system. The ICRS and ICRF are maintained for the IAU by the International Earth Rotation and Reference Systems Service (IERS, http://www.iers.org/).

For cartographic coordinates and conventions for planets and satellites, the IAU and the International Association of Geodesy (IAG) have established jointly the Working Group on Cartographic Coordinates and Rotational Elements (WGCCRE), which publishes triennial reports, currently in the journal Celestial Mechanics and Dynamical Astronomy (Davies, et al., 1980, 1983, 1986, 1989, 1992, 1996; Seidelmann, et al., 2002, 2005, 2007). This working group includes PDS-affiliated scientists, thus assuring full interaction in defining the standards. Publications and reports issued by the WGCCRE can be found at http://astrogeology.usgs.gov/Projects/WGCCRE/. PDS data providers should refer to these reports for current information and recommendations on rotational elements for Solar System bodies and how these are related to their cartographic coordinates.

The NASA Lunar Geodesy and Cartography Working Group and the Mars Geodesy and Cartography Working Group are sponsored by the NASA Lunar Precursor Robotics Program (LPRP) and Mars Program offices, respectively, and are responsible within NASA for providing additional coordination of cartographic standards and related (e.g., data processing) issues (Archinal et al., 2008a, 2008b; Duxbury et al., 2002). These Working Groups have made additional recommendations regarding coordinate systems (generally with additional detail) beyond those of the WGCCRE.

11.2 Inertial Reference Frame and Time System

The orientation of a body in the Solar System can be calculated using a series of rotation angles to define the directions of the bodys principal axes with respect to an inertial reference frame (i.e., a system that is not rotating or accelerating relative to a specific reference point) which provides a standard frame from which position, velocity, and acceleration can be measured. Such a reference frame is a set of identifiable fiducial points and their positions on the sky, providing a practical realization of a reference system that defines the origin, fundamental planes (or axes), and transformations between observed elements and reference points in the celestial coordinate system. Reference coordinate systems are defined by a system of concepts (e.g., using planetocentric latitude and longitude) while a reference coordinate frame is a specific realization of a coordinate system that is anchored to real data (such as a photogrammetric control network, altimetry crossover solutions, or lunar ephemerides) (Kovalevsky and Mueller, 1981).

For a planetary body in space, position is defined relative to a Z axis (typically the spin vector of the body, or the planetographic north pole), the X axis (defined as the point where the equator of the body crosses the equatorial plane of an inertial frame at a specific epoch), and the Y axis of a right-handed system. The standard units for coordinates are based on the International System of

Units (SI), including decimal degrees. The orientation of Solar System bodies can be calculated from angular position (right ascension ? and declination ??) with respect to the equatorial system of a particular epoch. For example, the orientation of the north pole of a body at a given epoch is specified by its right ascension ? and declination ?, while the location of the prime meridian is specified by the angle W (Davies et al., 1980).

The standard epoch is called J2000.0 and is defined to be 2000 January 1.5 TDB, where TDB is Barycentric Dynamical Time (e.g., Seidelmann et al., 2007). This corresponds to 2000 January 1, 1200 hours TT (Terrestrial Time) or the Julian Date 2451545.0 (NAO, USNO and HMNAO, 1983). This also corresponds to 2000 January 1, 11:58:55.816 UTC (Coordinated Universal Time; Seidelmann et al., 1992). Although the natural system for many applications would be TDB, UTC is considered the fundamental system for all PDS data products. The standard way of expressing UTC is in year, month, day, hour, minute, and decimal seconds. Julian Dates (JD) are supported as a supplementary system for reporting UTC time. However the JD time scale must be specified (e.g., UTC or TDB). See the Planetary Science Data Dictionary (PDS, 2008), chapter 2, for further information on time representation.

The currently accepted orientation of the inertial system (i.e., J2000.0 right ascension and declination) is defined by the International Celestial Reference System (ICRS), which is a particular implementation of the Barycentric Celestial Reference System (BCRS) (IAU, 2000). The ICRS is the fundamental celestial reference system of the IAU, and it has an origin at the barycenter of the Solar System and space fixed (kinematically non-rotating) axis directions. As noted by the IAU, the ICRS is meant to represent the most appropriate coordinate system for expressing reference data on the positions and motions of celestial objects. Specifications for the ICRS include a metric tensor, a prescribed method for establishing and maintaining axis directions, a list of benchmark objects with precise coordinates, and standard algorithms to transform these coordinates into observable quantities for any location and time. The ICRS is derived from the International Celestial Reference Frame (ICRF) comprised of coordinates for a set of fiducial points on the sky. The ICRF is within 0.05 arcseconds (Chapront et al., 2002; Herring et al., 2002) of the Solar System inertial frame based on Earths Mean Equator (EME) at the Equinox of Julian Ephemeris Date (JD) 2451545.0 (i.e., J2000.0). This is consistent with current dynamical practice and spacecraft and planetary ephemerides (e.g., those provided by the NASA Jet Propulsion Laboratory).

Many older data sets, collected before the J2000.0 system and ICRF were defined, are referenced to EME and Equinox of Besselian 1950.0 (B1950.0; JD 2433282.423). While this reference frame should not be used for current data, PDS supports this reference frame for older data. Transformation between the B1950.0 and J2000.0 (and the nearly equivalent ICRF) systems has been well defined by the IAU (NAO, USNO and HMSNAO, 1983; also see http://nedwww.ipac.caltech.edu/forms/calculato

Positions may be expressed in other coordinate systems and associated frames, which can be derived from the fundamental system and frame, when this enhances the use of the data for various applications. These include ecliptic-based coordinates and heliographic coordinates. These coordinates, while possibly ”natural” for many applications, are derivable from the fundamental system and are therefore treated as supplementary data by PDS. In some cases, it is convenient to work in one preferred coordinate system and then to convert to another, more standard system for products. This practice of providing the natural working coordinates in addition to the coordinates in a fundamental system promotes ease of use of PDS products and should be adopted by all data providers who use coordinate systems other than the fundamental system. As noted above, all supplementary coordinate systems must be fully documented in PDS products and must be negotiated with the PDS prior to delivery.

11.3 Spin Axes and Prime Meridians

The spin axis orientations of many Solar System bodies are defined by the WGCCRE in the ICRF inertial reference frame. For historical reasons, the orientation of the spin axis of planets and satellites is defined by the north pole, which is the pole that is on the northern side of the Invariant Plane of the Solar System (close to but not the same as the ecliptic). With this definition of the north pole, it is also necessary to specify whether the rotation is direct or prograde (in the same direction as the Suns rotation or counterclockwise when viewed from above the north pole) or retrograde (opposite to the direction of the Suns rotation).

For small bodies such as comets and asteroids, for which precession due to torques can cause large changes in the angular momentum vector, the orientation is defined by the positive pole, which is the pole determined by the right hand rule for rotation. Since some small bodies can be in excited state rotation, there are numerous complications in application that are addressed in more detail in the WGCCRE reports. Depending on the mode of excited state rotation, the axis may coincide with the maximum moment of inertia. Some cases, particularly the case of chaotic rotation, are considered on a case by case basis by the WGCCRE.

If a body has a solid surface, prime meridians for a given longitude system may be defined by specifying the coordinates of a surface feature on the body (usually a small feature such as a crater in the equatorial region) or by the mean direction relative to the parent body for synchronously rotating bodies (e.g., the Moon, the Galilean moons, and most of the Saturnian moons). Where insufficient observations exist to determine the principal moment of inertia, coordinates of a surface feature will be specified and used to define the prime meridian. In the case of planets without solid surfaces, the definition of the prime meridian is somewhat arbitrary. In any case, the actual definitions are decided by the WGCCRE, not by the PDS. We note that influxes of new data often lead to an iterative process to define (or improve) the orientation of the spin axis or other parameters used to define a coordinate system and in these cases the data providers (e.g., spacecraft mission personnel) and the WGCCRE must maintain close contact regarding the definition.

11.4 Body-Fixed Planetary Coordinate Systems

Two types of coordinate systems are fixed to the body planetocentric and planetographic. Details of the coordinate systems for planets and satellites differ from those for small bodies and rings. This section discusses only the aspects that are common to all applications. The Planetocentric system has an origin at the center of mass of the body. Planetocentric coordinates are defined by a vector from the center of mass of the body (often approximated as the center of figure) to the point of interest, typically but not necessarily a point on the surface (e.g., an impact crater with known position). The planetocentric latitude is the angle between the equatorial plane and the vector, while the planetocentric longitude is the angle between the prime meridian and the projection of the vector onto the equatorial plane.

The Planetographic system also has an origin at the center of mass of the body. Planetographic coordinates, however, are defined by vectors perpendicular to a reference surface, often a biaxial ellipsoid that is centered on the body and chosen to describe the gross shape of the body. Reference surfaces vary from body to body and are defined by the WGCCRE in consultation with the observers who provide the information to define such surfaces. The most common reference surface is an oblate spheroid aligned with the spin axis of the body. However, for certain applications the reference surface may be a triaxial ellipsoid, a gravitational equipotential, or a higher order surface model.

For a biaxial ellipsoid the planetographic latitude is the angle between the equatorial plane and a vector through the point of interest, where the vector is normal to the reference surface. Planetographic longitude is the angle between the prime meridian and the projection of the same vector onto the equatorial plane. In general, the planetographic vector does not pass through the origin. The vector need not pass through the spin axis but in most realistic cases it does. If the reference surface is a sphere, the planetographic and planetocentric vectors are identical.

The WGCCRE allows for the use of either planetographic or planetocentric coordinates for a given body, so data providers may adopt either system. Historically planetographic coordinates have been preferred for cartographic products, while planetocentric coordinates were used for dynamical (i.e. orbit, gravity field, altimetric) observations and calculations. For the planet Mercury, the MESSENGER mission has chosen to use planetocentric coordinates as the primary coordinate system for all products (Seidelmann et al., 2007). For the planet Mars, the MGCWG and all current NASA missions have chosen to use planetocentric coordinates as the primary coordinate system for products (Duxbury et al., 2002). Producers of printed or electronically printed maps (e.g., in PDF format) may wish to show both types of coordinates.

11.4.1 Planets and Satellites

For planets and satellites, the conventions are complicated for historical reasons. In the planetocentric coordinate system, northern latitudes are those in the hemisphere of the body containing the spin pole that points to the northern side of the invariant plane of the Solar System. The bodys rotation direction, either prograde or retrograde, must also be specified. Planetocentric longitude increases eastward (i.e., in the direction defined by the right-hand rule and the north pole) from the prime meridian, from 0 to 360. Thus an external observer sees the longitude decreasing with time if the rotation is prograde but increasing with time if the rotation is retrograde.

North and south planetographic latitude are defined in the same way as for planetocentric latitude, although the numerical values for a given point on the surface, (other than on the equator or at the poles) are different if the reference surface is not a sphere. The definition of planetographic longitude is dependent upon the rotation direction of the body, with the basic definition being that an external observer should see the longitude increasing with time, or that the longitude increases in the direction opposite to the rotation, although there are exceptions due to historical practice for Earth, the Moon, and Sun. That is to say, the longitude increases to the west if the rotation is prograde (or eastward) and vice versa. Whether the rotation direction is prograde or retrograde can be determined from the current WGCCRE report. See Tables 1 and 2 (or their equivalent in any future report), where the sign of the velocity term for W indicates either prograde (positive) or retrograde (negative) rotation. For all bodies a longitude range of 0 to 360 can be used.

For Earth, the Moon, and the Sun, a longitude range of -180 to +180 has been used in the past [including in existing PDS data sets, as defined by the Planetary Science Data Dictionary (PDS, 2002)] and is allowed by the WGCCRE. However, for the Moon, the NASA LGCWG and LRO Mission recommend that in the future, only the 0 to 360 range be used (LGCWG, 2008; LRO Project, 2008). For printed or electronically printed maps (e.g., in PDF format), it may be useful to label the longitude grid both with primary 0 to 360 coordinates and -180 to +180 coordinates.

For the Moon, two slightly different reference systems are commonly used to orient the lunar body-fixed coordinate system. One is the Mean Earth/Polar Axis (ME) system, the preferred system to be used for PDS data products. The other is the axis of figure system, also called the Principal Axis (PA) system, sometimes used internally among instrument teams for specific applications. For computing precise lunar coordinates, the WGCCRE recommends the use of the JPL DE403 ephemeris (which provides lunar orientation in the PA system), rotated into the ME system. The WGCCRE noted in its most recent report that improved versions of the JPL ephemerides were imminent and might be used instead. In fact the JPL DE421 ephemeris is now available and, after rotation into the ME system, is recommended for use (LGCWG, 2008; LRO Project, 2008). The maximum difference between these two frames in the ME system for the period 2000-2019 is only about 6 meters (Archinal, 2008).

11.4.2 Small Bodies

For small bodies (asteroids and comets), both planetographic and planetocentric coordinates follow the same right hand rule that is used to define the positive pole, which can be either above or below the invariant plane of the Solar System. For the simple case of a body with positive pole pointing to the northern hemisphere of the Solar System, this corresponds to longitude, both planetocentric and planetographic, increasing eastward, 0 to 360, which in turn corresponds to the case in which the longitude seen by an outside observer decreases with time.

For some small bodies, coordinates based on latitude and longitude alone can be multi-valued in radius i.e., the vector from the center of the body can intersect the surface in more than one place. There may also be complications (due to the irregular shape) which force special procedures when producing a useful, planar map. Such details are discussed in reports of the WGCCRE.

11.4.3 Rings

There is no international standard for ring coordinate systems. Standards in use for such PDS products were defined by experts in the Rings Node, in consultation with a broad cross-section of interested scientists. Conventions for coordinate systems for rings are similar to those for small bodies, in as much as they are all based on a right-hand rule, with longitude increasing in the direction of orbital motion. Thus longitude increases eastward for the prograde-moving rings (Jupiter, Saturn, and Neptune), but it increases westward for retrograde-moving rings of Uranus. Rings also use a positive pole direction following the right hand rule, analogous to the case for small-body rotation, thus coinciding with the North Pole of Jupiter, Saturn, and Neptune, but the South Pole of Uranus.

Coordinates for rings differ from those for planets and small bodies in not being body-fixed because there are no fixed features to define longitude. They are defined in an inertial system that is co-moving with the center of mass of the parent body. Specifically, longitudes are measured from the ascending node of the plane of the rings in the ICRF, i.e. the point at which the plane of the rings intersects the ICRF equator. In the case of inclined rings, longitudes are measured as a broken angle from the ascending node of the planets equatorial plane in the ICRF, along the equatorial plane to the ring planes ascending node, and thereafter along the ring plane.

11.4.4 Planetary Plasma Interactions

There are no international standards for values or names of coordinate systems of planetary plasma observations. Recommendations for coordinate systems in the near-Earth environment by Russell

(1971) have been generalized for use with plasma observations at other bodies. More recently, other systems have been defined (e.g., Franz and Harper, 2002) and are currently in use. The coordinate systems used for plasma observations and data analysis typically are right-handed. The primary exception to this rule is the left-handed Jovian System III.

Standards for planetary plasma data products for PDS were defined by experts in the Planetary Plasma Interactions Node, following recommendations from Russell (1971) and Franz and Harper (2002) and in consultation with other specialists. Providers and users of PDS data featuring plasma observations are encouraged to use names as defined by these authors where appropriate, and to follow similar name construction when new systems must be defined.

11.5 Surface Models

A standard reference surface model commonly used for hard surfaces is the digital terrain model (DTM). The DTM defines body radius or geometric height above the body reference surface as a function of cartographic latitude and longitude. Spheroids, ellipsoids and harmonic expansions giving analytic expressions for radius as a function of cartographic coordinates are all allowed in PDS. A DTM may also define potential height, i.e., elevation, above an equipotential surface, provided the method is specified, including the specification of appropriate constants and gravity field that is used to convert to/from radii and potential height.

The only internationally recognized DTM is the MOLA model for Mars (Seidelmann, et al., 2007, page 168 in WGCCRE #10). DTMs are also available for other bodies, including the Moon and several small bodies; but their use is not officially recommended and therefore up to the individual user.

The digital image model (DIM) defines body brightness in a specified spectral band or bands as a function of cartographic latitude and longitude. A DIM may be associated with the surface radius, geometric height, or potential height values in a corresponding DTM or it may be registered independently to a spheroid, ellipsoid, or spherical harmonic expansion.

11.6 PDS Keywords for Cartographic Coordinates

NOTE: This section definitely needs updating for PDS4!!!

To support the descriptions of these various reference coordinate systems and frames, the PDS has defined the following set of geometry data elements [see the Planetary Science Data Dictionary (PDS, 2008) for complete definitions and additional data elements].

A_AXIS_RADIUS
B_AXIS_RADIUS
C_AXIS_RADIUS
COORDINATE_SYSTEM_CENTER_NAME
COORDINATE_SYSTEM_DESC
COORDINATE_SYSTEM_ID
COORDINATE_SYSTEM_NAME
COORDINATE_SYSTEM_REF_EPOCH
COORDINATE_SYSTEM_TYPE
EASTERNMOST_LONGITUDE
LATITUDE
LONGITUDE
MAXIMUM_LATITUDE
MAXIMUM_LONGITUDE
MINIMUM_LATITUDE
MINIMUM_LONGITUDE
POSITIVE_LONGITUDE_DIRECTION
WESTERNMOST_LONGITUDE

To support the description of locations in a planetary ring system, the PDS has defined the following data elements:

CENTER_RING_RADIUS
RING_RADIUS
MINIMUM_RING_RADIUS
MAXIMUM_RING_RADIUS

RING_LONGITUDE
MINIMUM_RING_LONGITUDE
MAXIMUM_RING_LONGITUDE

B1950_RING_LONGITUDE
MINIMUM_B1950_RING_LONGITUDE
MAXIMUM_B1950_RING_LONGITUDE

RING_EVENT_TIME
RING_EVENT_START_TIME
RING_EVENT_STOP_TIME

RADIAL_RESOLUTION
MINIMUM_RADIAL_RESOLUTION
MAXIMUM_RADIAL_RESOLUTION

The radius and longitude elements define an inertial location in the rings, and the ring event time elements define the time at the ring plane to which an observation refers. If desired, the radial resolution elements can be used to specify the radial dimensions of ring features that can be resolved in the data. See the Planetary Science Data Dictionary (PSDD; PDS, 2008) for complete definitions of these elements.

Some rings are not circular and/or equatorial. In these cases, the PSDD provides additional elements that can be used to describe a rings shape. The elements are:

RING_SEMIMAJOR_AXIS
RING_ECCENTRICITY
RING_PERICENTER_LONGITUDE
PERICENTER_PRECESSION_RATE
RING_INCLINATION
RING_ASCENDING_NODE_LONGITUDE
NODAL_REGRESSION_RATE
REFERENCE_TIME

Here the value of REFERENCE_TIME indicates the instant at which the LONGITUDE elements are defined. The actual pericenter and ascending node at the time of an observation are determined based on the precession and regression rates as follows:

```
pericenter_longitude = RING_PERICENTER_LONGITUDE +
        PERICENTER_PRECESSION_RATE *
        (observation_time - REFERENCE_TIME) mod 360
  ascending_node_longitude =
        RING_ASCENDING_NODE_LONGITUDE +
        NODAL_REGRESSION_RATE *
        (observation_time - REFERENCE_TIME) mod 360
```

The oscillating modes of a ring can also be specified if necessary:

RING_RADIAL_MODE
RING_RADIAL_MODE_AMPLITUDE
RING_RADIAL_MODE_FREQUENCY
RING_RADIAL_MODE_PHASE

Additional elements should be used to specify the assumed orientation of the planets pole:

POLE_RIGHT_ASCENSION
POLE_DECLINATION
COORDINATE_SYSTEM_ID

The COORDINATE_SYSTEM_ID can be either J2000.0 or B1950.0, with J2000.0 serving as the default. See the PSDD for further details.

11.7 Map Resolution

A uniform set of resolutions is helpful for analyses of multiple datasets and development of map products derived from PDS data, and the selected scale must account for differences in available image resolution and quality. Such map scales are measured against a reference surface that is typically a geometrically defined shape that represents a given planetary body. For global maps, the recommended spatial resolution for a map is 2n pixels per degree of latitude, where a pixel is treated as a finite area and n is an integer. A spatial resolution of 2n pixels per degree allows simple coregistration of multiple datasets by doubling or halving the pixel sizes (typically by averaging or interpolation) and without resampling or otherwise changing the pixels. These recommendations continue a convention established in the 1960s and 1970s by the lunar and Mars research communities (e.g., Batson, 1987; Greeley and Batson, 1990), as advocated by the NASA Planetary Cartography Working Group (PCWG) and its successor the Planetary Cartography and Geologic Mapping Working Group (PCGMWG) (PCWG, 1993, pp. 22-24), and affirmed by the LGCWG (2008).

For polar regions of global maps, the recommendation is also to use the binary map scale or 2n pixels per degree of latitude near the pole. This practice maintains consistency with the global data product.

For working at landing site scales with data that has pixels of tens of centimeters to a few meters in size, spatial resolutions of maps are more convenient if provided at scales of 1 meter per pixel resolution or multiples thereof (LGCWG, 2008). At such human scales this convention is simpler and will preserve inherent details of resolution for applications such as landing site operations, traversing, and surface engineering studies.

For both global and local maps showing elevation or relief, the recommended vertical resolution is 1 x 10m meters, where m is an integer chosen to preserve all the resolution inherent in the data.

11.8 References

(Need to fix references.)

(Note: All WGCCRE reports are listed below for completeness. WGCCRE Report 7 was not issued).

Archinal, B. A. (2008). Summary of Lunar Geodesy and Cartography Working Group Teleconference of Tuesday, 2008 April 16, May 18.

Archinal, B. A. and the Lunar Geodesy and Cartography Working Group (2008a). Lunar Mapping Standards and the NASA LPRP Lunar Geodesy and Cartography Working Group, Scientific Event B01, The Moon: Science, New Results, Ongoing Missions, Future Robotic and Human Exploration,, 37th COSPAR Scientific Assembly, July 13-20, Montreal, Canada, http://www.cospar-assembly.org/home.php. Abstract B01-0050-08 and available at http://www.cospar-assembly.org/user/download.php?

Archinal, B. A. and the Lunar Geodesy and Cartography Working Group (2008b). Lunar Science Support Activities by the NASA LPRP Lunar Geodesy and Cartography Working Group: Recommendations for Lunar Cartographic Standards, NLSI Lunar Science Conference, July 20-23, Moffett Field, CA. Abstract no. 2080, available at http://www.lpi.usra.edu/meetings/nlsc2008/pdf/2080.pdf.

Chapter 12

Context Information

Still needs some rewriting.

Scientific data cannot be properly understood and interpreted without knowledge of the context in which the data was acquired. The PDS requires that contextual information for all archived data be included within the archive or be identified and readily available from external publishers.

This includes such varied information as the mission, spacecraft or other facility within which the instrument operates, the instrument that acquired the data, the target of the investigation, and an overview of the data collection. Additional documentation should be provided on the key personnel associated with the instrument, data, and archiving task. Finally, any external references providing additional, supplementary information about the data should also be included.

PDS maintains a set of context files online for use where relevant.

Much context information evolves during the course of a mission and necessitates update to the context documentation. For example, unexpected events which occur during the course of a mission will necessitate updates documenting the impact of various instrument and spacecraft systems or on the quality of the data. This might include such as 'safing' incidents, instrument anomalies or failures, extensions of the mission, etc.

12.1 Investigations

Investigation is a super-category including sub-classes of Mission, Observing Campaign, and Other Investigation. The information may be included directory in the archive or referenced in an external journal article. It must be provided and updated by the flight project or other coordinating authority.

12.2 Nodes

This information describes the PDS node responsible for curating the archived data. The PDS node is responsible for providing the information.

12.3 Observing Systems

12.3.1 Instrument Hosts

Instrument_Host is a generic term covering any kind of platform on which an instrument may be mounted. The PDS currently recognizes three types of platforms: Laboratory, Rover, and Spacecraft.

12.3.2 Instruments

This information may be either included directly in the PDS archive or located in a journal article published by an external organization. In either case, it is the responsibility of the instrument team or hosting observatory to provide the information.

12.4 Personnel

Indicates PDS, mission, and other personnel involved in the production of PDS archives. Includes two sub-categories: PDS Affiliate and PDS Guest. This information will be provided by the investigator.

12.5 Publications / References

TBD

12.6 Resources

TBD

12.7 Software

TBD

12.8 Targets

PDS now recognizes multiple targets for instrument observations including: calibration objects, dust, features, regions, rings, solar system bodies, solar wind, and stars. This information will be provided by the PDS, gleaned from external sources.

Chapter 13

Documentation

Supplementary or ancillary reference materials are usually included with archive products to improve their short- and long-term utility. These documents augment the internal documentation of the product labels and provide further assistance in understanding the data products and accompanying materials. Typical archive documents include:

- Flight project documents
- Instrument papers
- Science articles
- Volume information
- Software Interface Specifications (SISs)
- Software user manuals

The PDS criteria for inclusion of a document in the archive are:

1. Would this information be helpful to a data user?
2. Is the material necessary?
3. Is the documentation complete?

In general, the PDS seeks to err on the side of completeness.

Each document to be archived must be prepared and saved in a PDS-compliant format, including a PDS label. Documents are delivered in the DOCUMENT directory of an archive volume (see the Volume Organization and Naming chapter of this document).

A flat, human-readable ASCII text version of each document must be included on the volume, although additional versions may be included in other supported formats at the option of the data producer. Flat ASCII text means the file may contain only the standard, 7-bit printable ASCII character set, plus the blank character and the carriage-return and linefeed characters as record delimiters. A file is human-readable if it is not encoded and if any special markup tags which may be included do not significantly interfere with an average users ability to read the file. So, for example, simple HTML files and TeX/LaTeX files with relatively little markup embedded in the text are generally considered human-readable and may, therefore, be used to satisfy the above ASCII text version requirement.

Note that the PDS takes the requirement for complete documentation very seriously. Documents that are essential to the understanding of an archive are considered as important as the data files themselves. Furthermore, including a document in a PDS archive constitutes publication (or re-publication) of that document. Consequently, documents prepared for inclusion in an archive are expected to meet not only the PDS label and format requirements, but also the structural, grammatical and lexical requirements of a refereed journal submission. Documents submitted for archiving which contain spelling errors, poor grammar or illogical organization will be rejected and may ultimately lead to the rejection of the submitted data for lack of adequate documentation.

Chapter 14

Geometry

Text currently pulled from external sources – needs to be re-written

(From PAG 3.3)

The mission is expected to archive complete geometric details from launch through end of mission. These typically include the full ephemeris of the spacecraft and orientation of the spacecraft and all instruments, the relationship of these to coordinate systems on the target, a history of all significant spacecraft events, and other housekeeping data (such as temperatures and power levels) that might be useful in understanding the behavior of instruments.

The mission should also archive ancillary data that are important to either mission planning or interpretation of the data from the mission. These might include contemporaneous, Earth-based observations or key models, such as shape models, used in interpreting the data.

Normally, radiometric tracking of data should be archived even if there is no ”instrument team” for radio science.

(include info on SPICE here)

Chapter 15

Science Data

Text currently pulled from external sources – needs to be re-written

(From APG 2.2.1)

In this document, we use the terms 'primary data' or 'science data' to refer to the primary output from instruments making scientific observations or measurements. Results from the processing of instrument outputcalibrated data, resampled or gridded data, maps, etc. are also included within the rubric of 'primary data'. Tables of summary output from many observations by several instruments may also be 'primary data'.

(From APG 3.2.1)

Data products should be defined based on science requirements in many cases the file structures have been determined during the development of the instrument even before your instrument was selected for the mission. With raw data, the products are usually obvious and are determined by the manner in which the data are read out and transmitted. For partially or fully processed data, there are more choices; you can select parameters that will be more useful (e.g., calibrated radiance) or drop others that seem unimportant (status flags of heaters that only operate during eclipses). If the products have already been defined, you may still have some options regarding file sizes and formats. Note, however, that the more you can use an existing conceptual pipeline to deliver archival products, the simpler and more successful your archiving task will be. Mainstream products will have been well-scrubbed by the time they go to archive, and you will have to implement fewer unique steps to get them there. If you are also designing the primary data pipeline, plan the simplest, most self-explanatory file structures possible (consider accessing the data 20 years later on an alien platform without the team software).

(From APG 3.2.1.1)

Typically, data products are defined based on factors such as their utility in supporting science investigations of the team, the probable way in which the data will be accessed, and the expected frequency of access. What does the team need in order to meet its science goals? What processing schemes have been employed in the past either by your team or others? What types of information are presented at scientific and project meetings? Will any of the results be fed back into future planning? If so, what and how? Is your team centrally located? If not, are there concerns about delivering data to remote sites either because of the volume or poor communications links? Do your data appear in typically small or large blocks? If small, are there logical ways to aggregate several small blocks into a larger product? If large, does it make sense to subdivide the blocks into smaller pieces without affecting the science use?

(From APG 3.2.1.2)

Although PDS has defined a set of data formats or objects (see Appendix A for the complete list), tables and images account for the majority of products. Anticipated use of the science data will often dictate the archiving format e.g., TABLE, IMAGE, SPECTRUM, etc. For example, many spacecraft imaging data sets will be composed of single image data products, each defined as an IMAGE object. Once you have determined the 'logical' definition of your data products, you need to format them according to PDS Standards.

TABLE is a natural choice for numerical data that can be easily visualized in rows and columns, the SPECTRUM being a special case of the TABLE that is defined for spectral data. The PDS TABLE object accommodates both binary and ASCII tables; the latter is encouraged for all but the most massive files. The SPREADSHEET object is similar to TABLE except that it uses FIELD objects with maximum byte counts instead of COLUMN's and may provide more efficient storage when fields are highly irregular and/or many values are missing. TABLE, SPECTRUM, and SPREADSHEET objects can be read into many database applications. It is recommended that you discuss data formats with your PDS rep in the initial steps of archive planning to assist you in choosing the best object for your data.

An example of a TABLE is given in Figures 3.2.1.2a,b – Note: This TABLE contains the 4 typical data types (time, Integer, character and real) that appear in ASCII tables. The data object (which we assume is a single, separate file) is given in Figure 3.2.1.2a; the object definition, which will be part of its detached label, follows in Figure 3.2.1.2b. Note that the value of ROW_BYTES includes delimiters but that the values of START_BYTE and BYTES do not and that the double quotes in Column 3 are considered to be the delimiters of a character string.

The IMAGE object is flexible in the sense that both ASCII and binary samples (pixels) can be accommodated. The image is specified by keywords that give the number of lines and samples and the properties of the latter. It is possible to assign part of the beginning and/or end of each line for storage of non-image data. Figure 3.2.1.2c shows an example of an IMAGE object definition.

Chapter 16

Software

TBD - may not be included in PDS4

Chapter 17

Supplementary Products

TBD

17.1 AAREADME Products

going away?

17.2 Ancillary Database Dumps

17.3 Errata Products

17.4 Modification Histories

17.5 Processing History

17.6 Supplementary Metadata Products

17.7 Update Files

(what to call tables of associated products like anaglyph pairs)

footprint files? (or is this a browse product?)

Chapter 18

Time Standards

ISO 8601

TBD

Chapter 19

XML Schemas

TBD

19.1 Data Dictionaries

19.1.1 Purpose

The primary purpose of the Planetary Science Data Dictionary (PSDD) is to allow members of the planetary science community to benefit from standards work done in the area of data product description. The work that supports it is done at the Jet Propulsion Laboratory by individuals who participate in U.S. and international standards efforts. As a result the PSDD may serve as a guide to other data systems still in development, or to data systems that will eventually be connected with PDS or the IPDA.

19.1.2 Dictionary Format

TBD

19.1.3 Registration Authority and Change Control Procedures

TBD

Chapter 20

Nomenclature Rules

20.1 Identifiers

20.1.1 Logical Identifiers vs. Versioned Identifiers

Each version of each product in the PDS has an identifier which is unique in the world. In the PDS, we refer to this as a *versioned identifier* or *lidvid*.

This identifier, minus the version information, is referred to as a *logical identifier* or *lid*. The logical identifier is used to denote all versions of a product collectively.

lidvid = lid "::" version id

Versioned

length limit on versioned identifiers is 255 characters; there is no specific limit on any individual component of these, but missions must plan well in advance to leave enough characters for the longest LIDVID (the product LIDVID), INCLUDING ALLOWING FOR HIGH LEVEL VERSION IDs!

stay well under the limit in order to accommodate the inevitable and unanticipated changes that will come down the road (higher sol numbers than expected due to extended missions, higher version ids

20.1.2 Product Identifiers

Product identifiers are used to register PDS products and must be **globally** unique.

product logical identifier (lid) = <collection logical identifier> ":" <specific identifier>
product versioned identifier (lidvid) = <product logical identifier> "::" <product version id>

specific identifier = a string, determined by the data provider, that is unique within the collection

Many data providers choose to use the file name for specific identifier. In this case, it shall consist only of the file name base of the label file (i.e., without the file name extension). If the file name base is not unique within the collection, information (such as the relative directory path) shall be pre-pended to it to make it unique.

Rules:

- Must be unique across the PDS.
- (character set is same as for file names)

20.1.3 Collection Identifiers

collection logical identifier (lid) = <bundle logical identifier> ":" <specific identifier>
collection versioned identifier (lidvid) = <collection logical id> "::" <collection version id>

specific identifier = <mission abbrev.> "_" <inst abbrev.> "_" <collection type>
["_" <processing level or collection subtype>]
["_" <target or target system or target type>]
["_" <description>]

Mission abbreviation examples: (May also be the abbreviation for a spacecraft, telescope, observing campaign, or investigation)
CLEM
GLL
MGS
SL9

Instrument abbreviation examples:
MOC

NIMS
UVVIS

Collection types:
BROWSE
CALIBRATION
CONTEXT (for context products delivered as part of non-context bundles)
DATA
DOCUMENT
GEOMETRY (for non-SPICE geometry data)
INSTRUMENT
INSTRUMENT_HOST / OBSERVING_SYSTEM
MISSION / INVESTIGATION / CAMPAIGN
NODE
PUBLICATIONS
SPICE_KERNEL (for SPICE products delivered as part of non-SPICE bundles)
SUPPLEMENTARY
TARGET
XML_SCHEMA

Rules:

- char set =

20.1.4 Bundle Identifiers

bundle logical identifier (lid) = “URN:NASA:PDS:” <specific identifier>
bundle versioned identifier (lidvid) = <bundle logical identifier> “::” <bundle version id>

specific identifier = that portion of the collection specific identifier which is common across all the component collections of the bundle

bundle and collection ids must be unique across pds

in general they will be constructed major to minor order

the bundle id will consist of all those terms that are common across component collections

order of discriminating terms is up to data provider in consultation with node

underscores used within terms, dashes used between terms and = underscore

no forward slashes

LIDs are all uppercase (inside the labels; not enforced if used on file systems)

use ids rather than names for component terms (like “MGS” rather than “MARS GLOBAL SURVEYOR”)

20.2 Other Identifying Information

20.2.1 Version IDs

Version IDs are used for all types of products, including standard products, collections, and bundles.

Note that in all cases, the incrementing of a version identifier implies that a user of that product should use the latest version of the product. In other words, incrementing a version indicates that previous versions of the product have been superseded. (It is, of course, possible that a user may, in specific cases, deliberately choose to use older versions of a product. As long as those products were archived with the PDS, they will still be available, albeit potentially offline.)

Formation Rules:

- Version identifiers are appended to logical identifiers to form versioned identifiers. They are separated from the logical identifier by a double colon (“::”).
- Version IDs must be of the form M.m where “M” and “m” are both integers. “M” is the “major” component of the version and “m” is the “minor” component of the version.
- The major number is initialized to one for archive products. (Zero may be used for sample products or test run products that are not intended for the archive.) The minor number is initialized to zero.
- Whenever the major number is incremented, the minor number is re-set to zero.
- The minor portion of the version is **not** pre-padded with zeros; it is simply incremented as an integer. Thus, “1.1” and “1.10” are different versions; “1.01” is invalid.

It is up to the determination of each PDS discipline node, working in consultation with data preparers, to determine the criteria for how and under what circumstances version numbers will be incremented for a particular archive. Specific versioning conventions for every archive must be detailed in the respective product documentation. Note that depending upon the criteria that a given node/mission combination selects, it is possible in some cases that higher versioned products may be created without their lower versioned counterparts existing.

20.2.2 Local Identifiers

Change to "Local IDs"?

Local identifiers are only required to be unique within an individual product label. They are primarily used for navigating among the different portions of the label and can be used, for example, to tie together a data object with an Object_Statistics class describing it, to to make clear the relationship between two Image_Grayscale classes (a primary and a browse) and the one or two File classes identifying the data they describe.

The construction of these identifiers is therefore largely up to the data provider, with the exception of the character set restrictions. There is also one recommendation, to make these identifiers simpler for humans to find in a label:

Rules:

- Must be unique within the containing label.
- The character set for local identifiers is restricted to ASCII letters, number, the underscore, the colon, the dash, and the space. Numeric codes for these ASCII characters to be added. Period? Forward slash? Semi-colon? - No space

formation rule: use class name with incrementing number appended, underscore, prepended with zeroes so that all local identifiers of the same class type have the same number of digits (This is to preserve the ability to do alphabetical sorting.)

20.2.3 Names

TBD - Is this still needed?

Rules:

•

Recommendations:

•

20.2.4 Titles and Alternate Titles

Titles and alternate titles are intended for human consumption. Their primary use is to be displayed alongside a product when it is shown in some sort of product browser. It is not required to be unique. The specificity of the title is entirely up to the data provider, and is constrained only the manpower and resources necessary to populate it. Both of the following are acceptable for titles:

”LORRI image of Io; lat: 4 degS, lon: 164 degW; time: 2007-03-01T00:35 UTC”

”This image of Io, captured by New Horizon’s Long Range Reconnaissance Imager, captured the active Tvashtar plume. The image was acquired on March 1, 2007.”

The first of these two titles could be generated automatically by data production pipeline software. The second required post-production modification to incorporate observed feature information. Either is acceptable.

UTF-8 printable characters (explicitly disallowing roughly 30 ASCII characters non-printable characters)

limit to 255 **bytes** in length (up to data providers to be careful if using UTF-8 to verify byte length) (character type definition?)

Rules:

- Titles must be no more than 255 characters in length.
- Titles are text strings. The character set is restricted to the US-ASCII? character set.

Recommendations:

- None.

How are alternate titles used?

20.3 Directories

Reserved directory names: browse, calibration, context, data, document, geometry, supplemental (or about), xml_schema

same character set as filenames except no periods

Rules:

-

Recommendations:

-

Although NTFS allows each path component (directory or filename) to be 255 characters long and paths up to about 32767 characters long, the Windows kernel only supports paths up to 259 characters long if no UNC is used for addressing. – Need to follow up on this. See the section on "Maximum Path Length Limitation" on the following web page: http://msdn.microsoft.com/en-us/library/aa365247(VS.85).aspx

20.4 Filenames

Although modern operating systems are extremely permissive when it comes to filenames, best practices argue against taking full advantage of this. For example, many characters that are legal in filenames require escaping and can create numerous problems when moving files from one operating system to another. Similarly, 255 character filenames, while permitted, are often inconvenient to users who can't distinguish between similarly named lengthy filenames in many views on various operating systems.

Reserved names: AUX, COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9, CON, LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, LPT9, NUL, PRN, (and none of the former with any extension), a.out, core, .profile, .history, and .cshrc

Rules:

- Filenames must be unique within directories.
- Filenames must be no longer than 255 characters (although see path length restrictions discussed in the section on directory names).

- Filenames must be case-insensitive. (In other words, “MyFile.txt” and “myfile.txt” are not permitted in the same directory.)
- The character set is restricted to A-Z (ASCII 0x41 through 0x5A), a-z (ASCII 0x61 through 0x7A), 0-9 (ASCII 0x30 through 0x39), dash ”-” (ASCII 0x2D), underscore ”_” (ASCII 0x5F), and period ”.” (ASCII 0x2E). (we may re-consider this later)
- Filenames may not begin or end with a dash, underscore, or period.
- Filenames must begin with an alphabetical character (A-Z or a-z or 0-9).

Recommendations:

- Don’t abuse the permitted 255 character length for filenames. Try to keep them no longer than 30 to 40 characters in general. **Archives can fail peer review for being unnecessarily difficult to use as much as for violating actual PDS standards!**

Still have to discuss file extensions.

filenames must have at least one period followed by an extension. They may have more than one period, but PDS will consider all periods other than the final one to be part of the basename. (file base + file extension)

remember all SPICE file extensions need to be reserved

Need to include version information in filenames.

See: http://msdn.microsoft.com/en-us/library/aa365247(VS.85).aspx (“Naming Files, Paths, and Namespaces” from MSDN) and http://en.wikipedia.org/wiki/File_name (Wikipedia topic “Filename”)

20.5 Classes

The following rules and recommendations for the naming of classes in the PDS data model apply to all levels of the model: common, node, and mission.

Rules:

- The character set is restricted to ASCII letters, numbers, and the underscore. The first character of the class name must be a letter.

- Each component of the class name shall begin with an uppercase character; all other characters shall be lowercase, except when the class name incorporates an acronym. (Ex. Stream_Delimited_Field, SPICE_Kernel, Array_2D)

- The class name must not exceed 255 characters.

- The word order for the components of the class name shall be most significant first (Ex. Product_Table_Character, rather than Character_Table_Product).

Recommendations:

- Whenever possible, when creating subclasses of existing classes, use portions of the parent class name in the child class name to make the relationship between the classes apparent. (Ex. Table_Base, Table_Base_Character or Image_3D, Image_Color, Image_Color_Anaglyph)

- Use widely recognized and accepted terms for class name components, that clearly indicate the nature of a particular class. (Examples to avoid: Image_PanMos_Proj_JMR, Fpu_Bin_Table, Comp_12_8_Parm)

Camel Case - MSB (or do what make sense) follows rules for sub-classes

20.6 Attributes

Rules:

-

Recommendations:

-

rightmost component must correspond to an existing data element concept no articles

tool that provides pull down list of data element concepts for user to select from,

lower case - LSB

rules for common and node dictionaries but not for mission?

20.7 Attribute Values

Rules:

-

enumerated values - case no restrictions, but must match enumerated values (i.e. no limitation on person who created dictionary, but limitation on data preparer is that they must follow DD), UTF-8

Recommendations:

-

Appendices

Digital Object Classes

PDS4 classes can be divided into three types: digital, conceptual, and physical.

Physical classes are those used to describe actual physical objects that one could touch: a spacecraft, a planet, a camera. Conceptual objects are those that describe a concept, such as a mission, a map projection, or a camera model. Digital objects are those that we traditionally think of as data: an image, a binary table, a document, or a software program.

Appendix A of this document deals exclusively with the digital data classes. Appendix B describes the physical classes, and Appendix C details the conceptual classes.

.1 Digital Product Classes

In PDS4, the concept of a product is slightly different than it was in PDS3. In addition to the traditional data products that included things like images and tables, all of the components of an archive are now considered to be products. Thus, what we used to call a catalog object as described in a MISSION.CAT file is now considered to be a product in its own right, described by a Product_Mission class. Similarly, software and documents are now described by the Document_Set and Software_Set classes (problem here???).

Digital products are further divided into three sub-types: data, software, and documents. The reason for this is that the associated information required for each of these sub-types varies. All data products require a the same set of associated information; documents and software require their own associated pieces of information. The following three sections explain the overall structure of these three sub-types of digital products.

.1.1 Data Product Classes

The Data Product classes serve as the top level container for a primary digital data object and all of its associated pieces. It provides a place where the identification information for the product is located. It has a place for specifying the product's relationship to a data set, an instrument, a spacecraft or ground-based observation platform, a mission, a target, and a curating node. It provides the most basic information about the structure of the file containing the digital object (record type, size, and checksum). It also contains some of the descriptive information relating to the circumstances under which the data product was acquired (time, spacecraft clock count, etc.). Most importantly, it contains the data object set for the primary data object. An overview of the structure of the Data Product classes is shown in figure (ref).

are typically built around a primary data object such as an image or table.

.2 Array_Axis

The Array Axis class is used as a component of the array class and defines an axis of the array.

.2.1 Attributes & Associations

Attribute	**Definition**	**Card.**	**Req?**
elements	The elements attribute provides the count of the number of elements in an axis.	1	Y
name_	The name attribute indicates the label or significance of the dimension.	1	Y
sequence_number	The sequence number attribute provides a number that is used to order elements in an array.	1	Y
unit	The unit attribute indicates unit of measurement. **Valid Values:** m s km deg rad ms	1	N
Association	**Definition**	**Card.**	**Req?**

.2.2 Schema

TBD

.2.3 XML Example

TBD

.3 Array_Element

The Array Element class is used as a component of the array class and defines an element of the array.

.3.1 Attributes & Associations

Attribute	**Definition**	**Card.**	**Req?**
data_type	The data_type attribute provides the hardware representation used to store a value. **Valid Values:** IEEE754Single IEEE754Double UnsignedByte SignedMSB2 SignedMSB4 SignedMSB8 SignedLSB4 SignedLSB2 SignedLSB8	1	Y
scaling_factor	The scaling factor attribute is the scaling factor to be applied to each stored value in order to recover the original observation value. The observed value (Ov) is calculated from the stored value (Sv) thus: Ov = (Sv * scaling_factor) + value_offset.	1	N
unit	The unit attribute indicates unit of measurement. **Valid Values:** kg mol dn K C	1	N
value_offset	The value offset attribute is the fixed value to be added to each real value stored in order to recover the original observation value. The observed value (Ov) is calculated from the stored value (Sv) thus: Ov = (Sv * scaling_factor) + value_offset.	1	N
Association	**Definition**	**Card.**	**Req?**

.3.2 Schema

TBD

.3.3 XML Example

TBD

.4 Array_Ngt3D

The Array NGT3D class is the parent class for all n dimensional array base classes where n is greater than three.

.4.1 Attributes & Associations

Attribute	**Definition**	**Card.**	**Req?**
axes	The axes attribute provides a count of the axes.	1	Y
axis_order	The axis order attribute gives the axis index that varies fastest with respect to storage order. **Valid Values:** FIRST_INDEX_FASTEST	1	Y
comment	The comment attribute is a character string expressing one or more remarks or thoughts relevant to the object.	1	N
encoding_type	The encoding type attribute indicates the storage algorithm.	1	Y
local_identifier	The local identifier attribute provides the name of a local object; it is unique within a label. The value of local_identifier should be the class name. If several instances of the same class exist, then a numeric suffix is appended to the class name.	1	Y
Association	**Definition**	**Card.**	**Req?**
data_location	Association for locations.	1	Y

data_object	Composition association for Data Object.	1	Y
has_Array_Axis	Composition association for array axis.		N
has_Array_Element	Composition association for array element.	1	Y

.4.2 Schema

TBD

.4.3 XML Example

TBD

.5 Document_Desc

The Document_Desc class describes a document.

.5.1 Attributes & Associations

Attribute	**Definition**	**Card.**	**Req?**
acknowledgement_text	The acknowledgement_text attribute is a character string which recognizes another's contribution, authority, or right.	1	N
author_list	The author list attribute lists the composers of a work.	1	N
copyright	The copyright attribute is a character string giving information about the exclusive right to make copies, license, and otherwise exploit an object, whether physical or digital.	1	N
description	The description attribute is a character string that provides a statement, picture in words, or account that describes or is otherwise relevant to the object. Dublin Core: An account of the resource.	1	N
document_title	The document_title attribute provides the full name of the published document. This optional attribute is used only if the title in the identification area of the document product is not sufficient.	1	N
doi	The doi attribute gives the Digital Object Identifier for an object, assigned by the appropriate DOI System Registration Agency.	1	N

editor_list	The editor list attribute lists the editors of a work.	1	N
publication_date	The publication_date attribute provides the date when a published item, such as a document or a physical archival voliume, was issued.	1	Y
revision_id	The revision_id attribute indicates the revision level of a document, as distinct from the version_id, which tracks revisions to the files and label that constitute the document product.	1	N
Association	**Definition**	**Card.**	**Req?**
data_object	Composition association for Data Object.	1	Y

.5.2 Schema

TBD

.5.3 XML Example

TBD

.6 Document_File

The Document File class consists of attributes involved in the role of describing a file which is a part of a document.

.6.1 Attributes & Associations

Attribute	Definition	Card.	Req?
comment	The comment attribute is a character string expressing one or more remarks or thoughts relevant to the object.	1	N
creation_date_time	The creation date time attribute provides a date and time when the object was created.	1	N
directory_path_name	The directory path name attribute provides a sequence of names that locates a directory in a hierarchy of directories.	1	N
file_name	The file name attribute provides the name of a file.	1	Y
file_size	The file size attribute provides the size of the file.	1	N
local_identifier	The local identifier attribute provides the name of a local object; it is unique within a label. The value of local_identifier should be the class name. If several instances of the same class exist, then a numeric suffix is appended to the class name.	1	Y

max_record_bytes	The max record bytes attribute provides the maximum number of bytes that may be contained in a record.	1	N
md5_checksum	The md5_checksum attribute is the 32-character hexadecimal number computed for a file using the MD5 algorithm.	1	N
records	The records attribute provides a count of records.	1	N
Association	**Definition**	**Card.**	**Req?**
data_object	Composition association for Data Object.	1	Y

.6.2 Schema

TBD

.6.3 XML Example

TBD

.7 Document_Format

The Document Format provides a description of a variant of a logical document that is stored in a specific format. For example the PDS Standards Reference has HTML and PDF formatted versions.

.7.1 Attributes & Associations

Attribute	**Definition**	**Card.**	**Req?**
description	The description attribute is a character string that provides a statement, picture in words, or account that describes or is otherwise relevant to the object. Dublin Core: An account of the resource.	1	N
format_type	The format type attribute indicates the digital format used. **Valid Values:** HTML TEXT PDF-A	1	Y
starting_point_identifier	The starting point identiifer attribute provides the local_identifier of the file to be accessed first.	1	N
Association	**Definition**	**Card.**	**Req?**

.7.2 Schema

TBD

.7.3 XML Example

TBD

.8 Document_Part

The Document_Part class describes a file used as a component of a document.

.8.1 Attributes & Associations

Attribute	**Definition**	**Card.**	**Req?**
comment	The comment attribute is a character string expressing one or more remarks or thoughts relevant to the object.	1	N
encoding_type	The encoding type attribute indicates the storage algorithm. **Valid Values:** BINARY CHARACTER	1	Y
external_standard_id	The external standard id attribute provides the formal name of a standard not under PDS governance. **Valid Values:** PDF-A ENCAPSULATED_POSTSCRIPT GIF HTML JPG LaTEX MICROSOFT_WORD PNG POSTSCRIPT RICH_TEXT TEXT TIFF PDF	1	Y
local_identifier	The local identifier attribute provides the name of a local object; it is unique within a label. The value of local_identifier should be the class name. If several instances of the same class exist, then a numeric suffix is appended to the class name.	1	Y
Association	**Definition**	**Card.**	**Req?**
data_object	Composition association for Data Object.	1	Y

.8.2 Schema

TBD

.8.3 XML Example

TBD

.9 Encoded_Image

The Encoded Image class, a subclass of Encoded Byte stream is used for ancillary images in standard formats, such as JPEG.

.9.1 Attributes & Associations

Attribute	**Definition**	**Card.**	**Req?**
comment	The comment attribute is a character string expressing one or more remarks or thoughts relevant to the object.	1	N
encoding_type	The encoding type attribute indicates the storage algorithm.	1	Y
external_standard_id	The external standard id attribute provides the formal name of a standard not under PDS governance.	1	Y
local_identifier	The local identifier attribute provides the name of a local object; it is unique within a label. The value of local_identifier should be the class name. If several instances of the same class exist, then a numeric suffix is appended to the class name.	1	Y
Association	**Definition**	**Card.**	**Req?**
data_location	Association for locations.	1	Y
data_object	Composition association for Data Object.	1	Y

.9.2 Schema

TBD

.9.3 XML Example

TBD

.10 File_PDF

The File PDF class describes a PDF encoded byte stream.

.10.1 Attributes & Associations

Attribute	**Definition**	**Card.**	**Req?**
comment	The comment attribute is a character string expressing one or more remarks or thoughts relevant to the object.	1	N
encoding_type	The encoding type attribute indicates the storage algorithm.	1	Y
external_standard_id	The external standard id attribute provides the formal name of a standard not under PDS governance.	1	Y
local_identifier	The local identifier attribute provides the name of a local object; it is unique within a label. The value of local_identifier should be the class name. If several instances of the same class exist, then a numeric suffix is appended to the class name.	1	Y
Association	**Definition**	**Card.**	**Req?**
data_location	Association for locations.	1	Y
data_object	Composition association for Data Object.	1	Y

.10.2 Schema

TBD

.10.3 XML Example

TBD

.11 Header

The Header class describes a data object header.

.11.1 Attributes & Associations

Attribute	Definition	Card.	Req?
bytes	The bytes attribute provides the number of bytes.	1	Y
comment	The comment attribute is a character string expressing one or more remarks or thoughts relevant to the object.	1	N
description	The description attribute is a character string that provides a statement, picture in words, or account that describes or is otherwise relevant to the object. Dublin Core: An account of the resource.	1	N
encoding_type	The encoding type attribute indicates the storage algorithm.	1	Y
external_standard_id	The external standard id attribute provides the formal name of a standard not under PDS governance.	1	Y
local_identifier	The local identifier attribute provides the name of a local object; it is unique within a label. The value of local_identifier should be the class name. If several instances of the same class exist, then a numeric suffix is appended to the class name.	1	Y

name_	The name attribute provides a word or combination of words by which the object is known.	1	N
Association	**Definition**	**Card.**	**Req?**
data_location	Association for locations.	1	Y
data_object	Composition association for Data Object.	1	Y

.11.2 Schema

Fix page run-off.

```
<?xml version="1.0" encoding="UTF-8"?>
  <!-- PDS4 XML/Schema for Header_0.1.1.1.c  Fri Oct 29 06:29:36 PDT 2010 -->
  <!-- Generated from the PDS4 Information Model V0.1.1.1.c -->
  <!-- *** This PDS4 product schema is a preliminary deliverable. *** -->
  <!-- *** It is being made available for review and testing. *** -->
  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://pds.nasa.gov/schema/pds4/pds"
    xmlns:pds="http://pds.nasa.gov/schema/pds4/pds"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified"
    version="1.1.1">

    <xsd:include schemaLocation="Extended_Types_0111c.xsd">
      <xsd:annotation>
        <xsd:documentation>PDS (common) Data Dictionary</xsd:documentation>
      </xsd:annotation>
    </xsd:include>

  <!-- <xsd:import namespace="http://pds.nasa.gov/schema/pds4/anyNS"
    schemaLocation="Any_Described_Data_Object_0111c.xsd"/> -->

  <xsd:complexType name="Header_Type">
    <xsd:sequence>
      <xsd:annotation>
        <xsd:documentation>
          The Header class describes a data object header.
        </xsd:documentation>
      </xsd:annotation>
      <xsd:element name="local_identifier" type="pds:local_identifier_Type" minOccurs="1" maxO
```

```
      <xsd:element name="comment" type="pds:comment_Type" minOccurs="0" maxOccurs="1"> <
      <xsd:element name="description" type="pds:description_Type" minOccurs="0" maxOccu
      <xsd:element name="bytes" type="pds:bytes_Type" minOccurs="1" maxOccurs="1"> </xsd
      <xsd:element name="encoding_type" type="pds:Encoded_Byte_Stream_encoding_type_Type
      <xsd:element name="external_standard_id" type="pds:Header_external_standard_id_Typ
      <xsd:element name="name" type="pds:name_Type" minOccurs="0" maxOccurs="1"> </xsd:e
      <xsd:element name="Data_Location" type="pds:Data_Location_Type" minOccurs="1" max0
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Data_Location_Type">
    <xsd:sequence>
      <xsd:annotation>
        <xsd:documentation>
          The Data Location class provides the location of a digital object.
        </xsd:documentation>
      </xsd:annotation>
      <xsd:element name="file_local_identifier" type="pds:file_local_identifier_Type" mi
      <xsd:element name="offset" type="pds:offset_Type" minOccurs="1" maxOccurs="1"> </x
    </xsd:sequence>
  </xsd:complexType>

  <xsd:element name="Header" type="pds:Header_Type">
    <xsd:annotation>
      <xsd:documentation>
        The Header XML Schema
      </xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:schema>
```

.11.3 XML Example

```
<pds:Header>
  <pds:local_identifier>PDS2_header</pds:local_identifier>
  <pds:comment>Original PDS2 label accompanying this product.</pds:comment>
  <pds:bytes>2068</pds:bytes>
  <pds:external_standard_id>PDS2</pds:external_standard_id>
  <pds:Data_Location>
    <pds:file_local_identifier>c1133641.img</pds:file_local_identifier>
    <pds:offset>0</pds:offset>
  </pds:Data_Location>
</pds:Header>
```

.12 Image_2D_Display

The Image_2D_Display class provides attributes to enable the display of a 2D image.

.12.1 Attributes & Associations

Attribute	**Definition**	**Card.**	**Req?**
first_line	The first_line attribute indicates the line within a source image that corresponds to the first line in a sub-image. Note: For the MPF IMP EDRs, the source image was the complete 256x256 image area within the CCD.	1	Y
first_line_sample	The first_line_sample attribute indicates the sample within a source image that corresponds to the first sample in a sub-image. Note: For the MPF IMP EDRs, the source image was the complete 256x256 image area within the CCD.	1	Y
line_display_direction	The line_display_direction attribute gives the preferred direction for displaying image lines on a display device. The default value is down, meaning lines are displayed top to bottom. When used, ”line_display_direction” must be accompanied by ”sample_display_direction”, which gives the order of sample display within each line. Image rotation attributes such as TWIST_ANGLE, CELESTIAL_NORTH_CLOCK_ANGLE, and BODY_POLE_CLOCK_ANGLE are defined assuming that the image is displayed in its preferred orientation. **Valid Values:** DOWN LEFT RIGHT UP	1	Y

sample_display_direction	The sample_display_direction attribute is the preferred orientation of samples within a line for viewing on a display device. The default is right, meaning samples are viewed from left to right on the display. "sample_display_direction" must be used with "line_display_direction". Image rotation attributes such as TWIST_ANGLE, CELESTIAL_NORTH_CLOCK_ANGLE, and BODY_POLE_CLOCK_ANGLE are defined under the assumption that the image is displayed in its preferred orientation. **Valid Values:** DOWN LEFT RIGHT UP	1	Y
Association	**Definition**	**Card.**	**Req?**

.12.2 Schema

TBD

.12.3 XML Example

TBD

.13 Image_3D

The Image 3D class is an extension of array_base and defines a three dimensional image.

.13.1 Attributes & Associations

Attribute	**Definition**	**Card.**	**Req?**
axes	The axes attribute provides a count of the axes.	1	Y
axis_order	The axis order attribute gives the axis index that varies fastest with respect to storage order. **Valid Values:** FIRST_INDEX_FASTEST	1	Y
comment	The comment attribute is a character string expressing one or more remarks or thoughts relevant to the object.	1	N
encoding_type	The encoding type attribute indicates the storage algorithm.	1	Y
local_identifier	The local identifier attribute provides the name of a local object; it is unique within a label. The value of local_identifier should be the class name. If several instances of the same class exist, then a numeric suffix is appended to the class name.	1	Y
Association	**Definition**	**Card.**	**Req?**
data_location	Association for locations.	1	Y

data_object	Composition association for Data Object.	1	Y
has_Array_Axis	Composition association for array axis.		N
has_Array_Element	Composition association for array element.	1	Y

.13.2 Schema

Fix page run-off.


```
<?xml version="1.0" encoding="UTF-8"?>
  <!-- PDS4 XML/Schema for Image_3D_0.1.1.1.c  Fri Oct 29 06:29:36 PDT 2010 -->
  <!-- Generated from the PDS4 Information Model V0.1.1.1.c -->
  <!-- *** This PDS4 product schema is a preliminary deliverable. *** -->
  <!-- *** It is being made available for review and testing. *** -->
  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://pds.nasa.gov/schema/pds4/pds"
    xmlns:pds="http://pds.nasa.gov/schema/pds4/pds"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified"
    version="1.1.1">

    <xsd:include schemaLocation="Extended_Types_0111c.xsd">
      <xsd:annotation>
        <xsd:documentation>PDS (common) Data Dictionary</xsd:documentation>
      </xsd:annotation>
    </xsd:include>

  <!-- <xsd:import namespace="http://pds.nasa.gov/schema/pds4/anyNS"
    schemaLocation="Any_Described_Data_Object_0111c.xsd"/> -->

  <xsd:complexType name="Image_3D_Type">
    <xsd:sequence>
      <xsd:annotation>
        <xsd:documentation>
          The Image 3D class is an extension of array_base and defines a  three dimensio
        </xsd:documentation>
      </xsd:annotation>
      <xsd:element name="local_identifier" type="pds:local_identifier_Type" minOccurs="1
      <xsd:element name="comment" type="pds:comment_Type" minOccurs="0" maxOccurs="1"> <
      <xsd:element name="axes" type="pds:Array_3D_axes_Type" minOccurs="1" maxOccurs="1"
```

```
    <xsd:element name="axis_order" type="pds:Image_3D_axis_order_Type" minOccurs="1" maxOccu
    <xsd:element name="encoding_type" type="pds:Array_Base_encoding_type_Type" minOccurs="1"
    <xsd:element name="Data_Location" type="pds:Data_Location_Type" minOccurs="1" maxOccurs=
    <xsd:element name="Array_Axis" type="pds:Array_Axis_Type" minOccurs="3" maxOccurs="3"> <
    <xsd:element name="Array_Element" type="pds:Array_Element_Type" minOccurs="1" maxOccurs=
  </xsd:sequence>
  <xsd:attribute name="base_class" type="xsd:string" fixed="Array_Base"> </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="Data_Location_Type">
  <xsd:sequence>
    <xsd:annotation>
      <xsd:documentation>
        The Data Location class provides the location of a digital object.
      </xsd:documentation>
    </xsd:annotation>
    <xsd:element name="file_local_identifier" type="pds:file_local_identifier_Type" minOccur
    <xsd:element name="offset" type="pds:offset_Type" minOccurs="1" maxOccurs="1"> </xsd:ele
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Array_Axis_Type">
  <xsd:sequence>
    <xsd:annotation>
      <xsd:documentation>
        The Array Axis class is used as a component of the array class and defines an axis o
      </xsd:documentation>
    </xsd:annotation>
    <xsd:element name="elements" type="pds:elements_Type" minOccurs="1" maxOccurs="1"> </xsd
    <xsd:element name="name" type="pds:name_Type" minOccurs="1" maxOccurs="1"> </xsd:element
    <xsd:element name="sequence_number" type="pds:sequence_number_Type" minOccurs="1" maxOcc
    <xsd:element name="unit" type="pds:unit_Type" minOccurs="0" maxOccurs="1"> </xsd:element
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Array_Element_Type">
  <xsd:sequence>
    <xsd:annotation>
      <xsd:documentation>
        The Array Element class is used as a component of the array class and defines an ele
      </xsd:documentation>
    </xsd:annotation>
    <xsd:element name="data_type" type="pds:data_type_Type" minOccurs="1" maxOccurs="1"> </x
    <xsd:element name="scaling_factor" type="pds:scaling_factor_Type" minOccurs="0" maxOccur
    <xsd:element name="unit" type="pds:unit_Type" minOccurs="0" maxOccurs="1"> </xsd:element
    <xsd:element name="value_offset" type="pds:value_offset_Type" minOccurs="0" maxOccurs="1
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="Image_3D" type="pds:Image_3D_Type">
  <xsd:annotation>
```

```
      <xsd:documentation>
        The Image_3D XML Schema
      </xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:schema>
```

.13.3 XML Example

TBD

.14 Image_Grayscale

The Image Grayscale class is an extension of array_base and defines a two dimensional grayscale image.

.14.1 Attributes & Associations

Attribute	**Definition**	**Card.**	**Req?**
axes	The axes attribute provides a count of the axes.	1	Y
axis_order	The axis order attribute gives the axis index that varies fastest with respect to storage order. **Valid Values:** FIRST_INDEX_FASTEST	1	Y
comment	The comment attribute is a character string expressing one or more remarks or thoughts relevant to the object.	1	N
encoding_type	The encoding type attribute indicates the storage algorithm.	1	Y
has_image_2d_array	Associated Image_2D_Array	1	N
local_identifier	The local identifier attribute provides the name of a local object; it is unique within a label. The value of local_identifier should be the class name. If several instances of the same class exist, then a numeric suffix is appended to the class name.	1	Y
Association	**Definition**	**Card.**	**Req?**

associated_Object_Statistics	Association for object statistics.	1	N
associated_Special_Constants	Association for special constants.	1	N
data_location	Association for locations.	1	Y
data_object	Composition association for Data Object.	1	Y
has_Array_Axis	Composition association for array axis.		N
has_Array_Element	Composition association for array element.	1	Y

.14.2 Schema

Fix page run-off.

```
<?xml version="1.0" encoding="UTF-8"?>
  <!-- PDS4 XML/Schema for Image_Grayscale_0.1.1.1.c  Fri Oct 29 06:29:36 PDT 2010 -->
  <!-- Generated from the PDS4 Information Model V0.1.1.1.c -->
  <!-- *** This PDS4 product schema is a preliminary deliverable. *** -->
  <!-- *** It is being made available for review and testing. *** -->
  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://pds.nasa.gov/schema/pds4/pds"
    xmlns:pds="http://pds.nasa.gov/schema/pds4/pds"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified"
    version="1.1.1">

    <xsd:include schemaLocation="Extended_Types_0111c.xsd">
      <xsd:annotation>
        <xsd:documentation>PDS (common) Data Dictionary</xsd:documentation>
      </xsd:annotation>
    </xsd:include>

  <!-- <xsd:import namespace="http://pds.nasa.gov/schema/pds4/anyNS"
    schemaLocation="Any_Described_Data_Object_0111c.xsd"/> -->
```

```
<xsd:complexType name="Image_Grayscale_Type">
  <xsd:sequence>
    <xsd:annotation>
      <xsd:documentation>
        The Image Grayscale class is an extension of array_base and defines a  two dimensio
      </xsd:documentation>
    </xsd:annotation>
    <xsd:element name="local_identifier" type="pds:local_identifier_Type" minOccurs="1" max
    <xsd:element name="comment" type="pds:comment_Type" minOccurs="0" maxOccurs="1"> </xsd:
    <xsd:element name="axes" type="pds:Array_2D_axes_Type" minOccurs="1" maxOccurs="1"> </x
    <xsd:element name="axis_order" type="pds:Image_Grayscale_axis_order_Type" minOccurs="1"
    <xsd:element name="encoding_type" type="pds:Array_Base_encoding_type_Type" minOccurs="1
    <xsd:element name="Data_Location" type="pds:Data_Location_Type" minOccurs="1" maxOccurs
    <xsd:element name="Object_Statistics" type="pds:Object_Statistics_Type" minOccurs="0" m
    <xsd:element name="Special_Constants" type="pds:Special_Constants_Type" minOccurs="0" m
    <xsd:element name="Array_Axis" type="pds:Array_Axis_Type" minOccurs="2" maxOccurs="2">
    <xsd:element name="Array_Element" type="pds:Array_Element_Type" minOccurs="1" maxOccurs
    <xsd:element name="Image_2D_Display" type="pds:Image_2D_Display_Type" minOccurs="0" max
  </xsd:sequence>
  <xsd:attribute name="base_class" type="xsd:string" fixed="Array_Base"> </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="Data_Location_Type">
  <xsd:sequence>
    <xsd:annotation>
      <xsd:documentation>
        The Data Location class provides the location of a digital object.
      </xsd:documentation>
    </xsd:annotation>
    <xsd:element name="file_local_identifier" type="pds:file_local_identifier_Type" minOccu
    <xsd:element name="offset" type="pds:offset_Type" minOccurs="1" maxOccurs="1"> </xsd:el
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Object_Statistics_Type">
  <xsd:sequence>
    <xsd:annotation>
      <xsd:documentation>
        The Object Statistics class provides a set of values that provide metrics about the
      </xsd:documentation>
    </xsd:annotation>
    <xsd:element name="local_identifier" type="pds:local_identifier_Type" minOccurs="1" max
    <xsd:element name="description" type="pds:description_Type" minOccurs="0" maxOccurs="1"
    <xsd:element name="maximum" type="pds:maximum_Type" minOccurs="0" maxOccurs="1"> </xsd:
    <xsd:element name="md5_checksum" type="pds:md5_checksum_Type" minOccurs="0" maxOccurs="
    <xsd:element name="mean" type="pds:mean_Type" minOccurs="0" maxOccurs="1"> </xsd:elemen
    <xsd:element name="median" type="pds:median_Type" minOccurs="0" maxOccurs="1"> </xsd:el
    <xsd:element name="minimum" type="pds:minimum_Type" minOccurs="0" maxOccurs="1"> </xsd:
    <xsd:element name="sample_bit_mask" type="pds:sample_bit_mask_Type" minOccurs="0" maxOc
    <xsd:element name="standard_deviation" type="pds:standard_deviation_Type" minOccurs="0"
```

```
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Special_Constants_Type">
    <xsd:sequence>
      <xsd:annotation>
        <xsd:documentation>
          The Special Constants class provides a set of values used to indicate special
        </xsd:documentation>
      </xsd:annotation>
      <xsd:element name="error_constant" type="pds:error_constant_Type" minOccurs="0" ma
      <xsd:element name="invalid_constant" type="pds:invalid_constant_Type" minOccurs="(
      <xsd:element name="missing_constant" type="pds:missing_constant_Type" minOccurs="(
      <xsd:element name="not_applicable_constant" type="pds:not_applicable_constant_Type
      <xsd:element name="saturated_constant" type="pds:saturated_constant_Type" minOccu
      <xsd:element name="unknown_constant" type="pds:unknown_constant_Type" minOccurs="(
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Array_Axis_Type">
    <xsd:sequence>
      <xsd:annotation>
        <xsd:documentation>
          The Array Axis class is used as a component of the array class and defines an
        </xsd:documentation>
      </xsd:annotation>
      <xsd:element name="elements" type="pds:elements_Type" minOccurs="1" maxOccurs="1">
      <xsd:element name="name" type="pds:name_Type" minOccurs="1" maxOccurs="1"> </xsd:e
      <xsd:element name="sequence_number" type="pds:sequence_number_Type" minOccurs="1"
      <xsd:element name="unit" type="pds:unit_Type" minOccurs="0" maxOccurs="1"> </xsd:e
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Array_Element_Type">
    <xsd:sequence>
      <xsd:annotation>
        <xsd:documentation>
          The Array Element class is used as a component of the array class and defines
        </xsd:documentation>
      </xsd:annotation>
      <xsd:element name="data_type" type="pds:data_type_Type" minOccurs="1" maxOccurs="1
      <xsd:element name="scaling_factor" type="pds:scaling_factor_Type" minOccurs="0" ma
      <xsd:element name="unit" type="pds:unit_Type" minOccurs="0" maxOccurs="1"> </xsd:e
      <xsd:element name="value_offset" type="pds:value_offset_Type" minOccurs="0" maxOcc
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Image_2D_Display_Type">
    <xsd:sequence>
      <xsd:annotation>
        <xsd:documentation>
```

```
          The Image_2D_Display class provides attributes to enable the display of a 2D image.
        </xsd:documentation>
      </xsd:annotation>
      <xsd:element name="first_line" type="pds:first_line_Type" minOccurs="1" maxOccurs="1"> <
      <xsd:element name="first_line_sample" type="pds:first_line_sample_Type" minOccurs="1" ma
      <xsd:element name="line_display_direction" type="pds:line_display_direction_Type" minOcc
      <xsd:element name="sample_display_direction" type="pds:sample_display_direction_Type" mi
    </xsd:sequence>
  </xsd:complexType>

  <xsd:element name="Image_Grayscale" type="pds:Image_Grayscale_Type">
    <xsd:annotation>
      <xsd:documentation>
        The Image_Grayscale XML Schema
      </xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:schema>
```

.14.3 XML Example

```
<pds:Image_Grayscale>
  <pds:local_identifier>Image</pds:local_identifier>
  <pds:axes>2</pds:axes>
  <pds:axis_order>FIRST_INDEX_FASTEST</pds:axis_order>
  <pds:Data_Location>
    <pds:file_local_identifier>c1133641.img</pds:file_local_identifier>
    <pds:offset>2068</pds:offset>
  </pds:Data_Location>
  <pds:Object_Statistics>
    <pds:local_identifier>Statistics</pds:local_identifier>
    <pds:maximum>255</pds:maximum>
    <pds:md5_checksum>3501b4cbb0acfb4392f0c8db1825d342</pds:md5_checksum>
    <pds:mean>133.84</pds:mean>
    <pds:median>145</pds:median>
    <pds:minimum>0</pds:minimum>
    <pds:standard_deviation>41.84</pds:standard_deviation>
  </pds:Object_Statistics>
  <pds:Special_Constants>
  </pds:Special_Constants>
  <pds:Array_Axis>
  </pds:Array_Axis>
  <pds:Array_Axis>
  </pds:Array_Axis>
  <pds:Array_Element>
  </pds:Array_Element>
  <pds:Image_2D_Display>
  </pds:Image_2D_Display>
```

```
</pds:Image_Grayscale>
```

.15 Inventory_LID

The Inventory_LID class is used to provide LID product references for a secondary collection.

.15.1 Attributes & Associations

Attribute	**Definition**	**Card.**	**Req?**
comment	The comment attribute is a character string expressing one or more remarks or thoughts relevant to the object.	1	N
encoding_type	The encoding type attribute indicates the storage algorithm.	1	Y
fields	The fields attribute provides a count of the fields.	1	Y
local_identifier	The local identifier attribute provides the name of a local object; it is unique within a label. The value of local_identifier should be the class name. If several instances of the same class exist, then a numeric suffix is appended to the class name.	1	Y
record_bytes	The record bytes attribute provides a count of the bytes in a record.	1	Y
records	The records attribute provides a count of records.	1	Y
Association	**Definition**	**Card.**	**Req?**
data_location	Association for locations.	1	Y

data_object	Composition association for Data Object.	1	Y
has_Record	Composition association for record.	?VARIABLE	Y
reference_association_type	The reference association type attribute provides the name of the association used in a reference. **Valid Value:** has_member	1	Y

.15.2 Schema

TBD

.15.3 XML Example

TBD

.16 Inventory_LIDVID

The Inventory_LIDVID class is used to provide LIDVID product references for a primary collection.

.16.1 Attributes & Associations

Attribute	**Definition**	**Card.**	**Req?**
comment	The comment attribute is a character string expressing one or more remarks or thoughts relevant to the object.	1	N
encoding_type	The encoding type attribute indicates the storage algorithm.	1	Y
fields	The fields attribute provides a count of the fields.	1	Y
local_identifier	The local identifier attribute provides the name of a local object; it is unique within a label. The value of local_identifier should be the class name. If several instances of the same class exist, then a numeric suffix is appended to the class name.	1	Y
record_bytes	The record bytes attribute provides a count of the bytes in a record.	1	Y
records	The records attribute provides a count of records.	1	Y
Association	**Definition**	**Card.**	**Req?**

data_location	Association for locations.	1	Y
data_object	Composition association for Data Object.	1	Y
has_Record	Composition association for record.	?VARIABLE	Y
reference_association_type	The reference association type attribute provides the name of the association used in a reference. **Valid Value:** has_member	1	Y

.16.2 Schema

TBD

.16.3 XML Example

TBD

.17 Manifest

The manifest class defines a table for file references.

.17.1 Attributes & Associations

Attribute	Definition	Card.	Req?
comment	The comment attribute is a character string expressing one or more remarks or thoughts relevant to the object.	1	N
encoding_type	The encoding type attribute indicates the storage algorithm.	1	Y
fields	The fields attribute provides a count of the fields.	1	Y
local_identifier	The local identifier attribute provides the name of a local object; it is unique within a label. The value of local_identifier should be the class name. If several instances of the same class exist, then a numeric suffix is appended to the class name.	1	Y
record_bytes	The record bytes attribute provides a count of the bytes in a record.	1	Y
records	The records attribute provides a count of records.	1	Y
Association	**Definition**	**Card.**	**Req?**
data_location	Association for locations.	1	Y

data_object	Composition association for Data Object.	1	Y
has_Record	Composition association for record.	?VARIABLE	Y

.17.2 Schema

TBD

.17.3 XML Example

TBD

.18 Map_Base_2D

The Map Base 2D class is is the parent class for all two dimensional map classes.

.18.1 Attributes & Associations

Attribute	**Definition**	**Card.**	**Req?**
axes	The axes attribute provides a count of the axes.	1	Y
axis_order	The axis order attribute gives the axis index that varies fastest with respect to storage order. **Valid Values:** FIRST_INDEX_FASTEST	1	Y
comment	The comment attribute is a character string expressing one or more remarks or thoughts relevant to the object.	1	N
encoding_type	The encoding type attribute indicates the storage algorithm.	1	Y
local_identifier	The local identifier attribute provides the name of a local object; it is unique within a label. The value of local_identifier should be the class name. If several instances of the same class exist, then a numeric suffix is appended to the class name.	1	Y
Association	**Definition**	**Card.**	**Req?**
data_location	Association for locations.	1	Y

data_object	Composition association for Data Object.	1	Y
has_Array_Axis	Composition association for array axis.		N
has_Array_Element	Composition association for array element.	1	Y

.18.2 Schema

TBD

.18.3 XML Example

TBD

.19 Movie

The Movie class is an extension of array_base and defines a move as a set of two dimensional images in a time series.

.19.1 Attributes & Associations

Attribute	**Definition**	**Card.**	**Req?**
axes	The axes attribute provides a count of the axes.	1	Y
axis_order	The axis order attribute gives the axis index that varies fastest with respect to storage order. **Valid Values:** FIRST_INDEX_FASTEST	1	Y
comment	The comment attribute is a character string expressing one or more remarks or thoughts relevant to the object.	1	N
encoding_type	The encoding type attribute indicates the storage algorithm.	1	Y
local_identifier	The local identifier attribute provides the name of a local object; it is unique within a label. The value of local_identifier should be the class name. If several instances of the same class exist, then a numeric suffix is appended to the class name.	1	Y
Association	**Definition**	**Card.**	**Req?**
data_location	Association for locations.	1	Y

data_object	Composition association for Data Object.	1	Y
has_Array_Axis	Composition association for array axis.		N
has_Array_Element	Composition association for array element.	1	Y

.19.2 Schema

TBD

.19.3 XML Example

TBD

.20 SPICE_Kernel_Binary

The SPICE Kernel class describes a SPICE file.

.20.1 Attributes & Associations

Attribute	**Definition**	**Card.**	**Req?**
comment	The comment attribute is a character string expressing one or more remarks or thoughts relevant to the object.	1	N
encoding_type	The encoding type attribute indicates the storage algorithm.	1	Y
external_standard_id	The external standard id attribute provides the formal name of a standard not under PDS governance.	1	Y
kernel_type	The kernel type attribute identifies the type of SPICE kernel. **Valid Values:** SPK PCK CK EK DSK DBK	1	Y
local_identifier	The local identifier attribute provides the name of a local object; it is unique within a label. The value of local_identifier should be the class name. If several instances of the same class exist, then a numeric suffix is appended to the class name.	1	Y
Association	**Definition**	**Card.**	**Req?**
data_location	Association for locations.	1	Y

data_object	Composition association for Data Object.	1	Y

.20.2 Schema

Fix page run-off.

```
<?xml version="1.0" encoding="UTF-8"?>
  <!-- PDS4 XML/Schema for SPICE_Kernel_Binary_0.1.1.1.c  Fri Oct 29 06:29:36 PDT 2010 -
  <!-- Generated from the PDS4 Information Model V0.1.1.1.c -->
  <!-- *** This PDS4 product schema is a preliminary deliverable. *** -->
  <!-- *** It is being made available for review and testing. *** -->
  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://pds.nasa.gov/schema/pds4/pds"
    xmlns:pds="http://pds.nasa.gov/schema/pds4/pds"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified"
    version="1.1.1">

    <xsd:include schemaLocation="Extended_Types_0111c.xsd">
      <xsd:annotation>
        <xsd:documentation>PDS (common) Data Dictionary</xsd:documentation>
      </xsd:annotation>
    </xsd:include>

  <!-- <xsd:import namespace="http://pds.nasa.gov/schema/pds4/anyNS"
    schemaLocation="Any_Described_Data_Object_0111c.xsd"/> -->

  <xsd:complexType name="SPICE_Kernel_Binary_Type">
    <xsd:sequence>
      <xsd:annotation>
        <xsd:documentation>
          The SPICE Kernel class describes a SPICE file.
        </xsd:documentation>
      </xsd:annotation>
      <xsd:element name="local_identifier" type="pds:local_identifier_Type" minOccurs="1
      <xsd:element name="comment" type="pds:comment_Type" minOccurs="0" maxOccurs="1"> <
      <xsd:element name="encoding_type" type="pds:Encoded_Byte_Stream_encoding_type_Type
      <xsd:element name="external_standard_id" type="pds:SPICE_Kernel_Binary_external_st
      <xsd:element name="kernel_type" type="pds:kernel_type_Type" minOccurs="1" maxOccu
      <xsd:element name="Data_Location" type="pds:Data_Location_Type" minOccurs="1" max
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Data_Location_Type">
    <xsd:sequence>
```

```
      <xsd:annotation>
        <xsd:documentation>
          The Data Location class provides the location of a digital object.
        </xsd:documentation>
      </xsd:annotation>
      <xsd:element name="file_local_identifier" type="pds:file_local_identifier_Type" minOccu
      <xsd:element name="offset" type="pds:offset_Type" minOccurs="1" maxOccurs="1"> </xsd:ele
    </xsd:sequence>
  </xsd:complexType>

  <xsd:element name="SPICE_Kernel_Binary" type="pds:SPICE_Kernel_Binary_Type">
    <xsd:annotation>
      <xsd:documentation>
        The SPICE_Kernel_Binary XML Schema
      </xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:schema>
```

.20.3 XML Example

TBD

.21 SPICE_Kernel_Text

The SPICE Kernel class describes a SPICE file.

.21.1 Attributes & Associations

Attribute	**Definition**	**Card.**	**Req?**
comment	The comment attribute is a character string expressing one or more remarks or thoughts relevant to the object.	1	N
encoding_type	The encoding type attribute indicates the storage algorithm.	1	Y
external_standard_id	The external standard id attribute provides the formal name of a standard not under PDS governance.	1	Y
kernel_type	The kernel type attribute identifies the type of SPICE kernel. **Valid Values:** LSK SCLK PCK EK IK FK MK	1	Y
local_identifier	The local identifier attribute provides the name of a local object; it is unique within a label. The value of local_identifier should be the class name. If several instances of the same class exist, then a numeric suffix is appended to the class name.	1	Y
Association	**Definition**	**Card.**	**Req?**
data_location	Association for locations.	1	Y

data_object	Composition association for Data Object.	1	Y

.21.2 Schema

Fix page run-off.

```
<?xml version="1.0" encoding="UTF-8"?>
  <!-- PDS4 XML/Schema for SPICE_Kernel_Text_0.1.1.1.c  Fri Oct 29 06:29:36 PDT 2010 -->
  <!-- Generated from the PDS4 Information Model V0.1.1.1.c -->
  <!-- *** This PDS4 product schema is a preliminary deliverable. *** -->
  <!-- *** It is being made available for review and testing. *** -->
  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://pds.nasa.gov/schema/pds4/pds"
    xmlns:pds="http://pds.nasa.gov/schema/pds4/pds"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified"
    version="1.1.1">

    <xsd:include schemaLocation="Extended_Types_0111c.xsd">
      <xsd:annotation>
        <xsd:documentation>PDS (common) Data Dictionary</xsd:documentation>
      </xsd:annotation>
    </xsd:include>

  <!-- <xsd:import namespace="http://pds.nasa.gov/schema/pds4/anyNS"
    schemaLocation="Any_Described_Data_Object_0111c.xsd"/> -->

  <xsd:complexType name="SPICE_Kernel_Text_Type">
    <xsd:sequence>
      <xsd:annotation>
        <xsd:documentation>
          The SPICE Kernel class describes a SPICE file.
        </xsd:documentation>
      </xsd:annotation>
      <xsd:element name="local_identifier" type="pds:local_identifier_Type" minOccurs="1" max(
      <xsd:element name="comment" type="pds:comment_Type" minOccurs="0" maxOccurs="1"> </xsd:e
      <xsd:element name="encoding_type" type="pds:SPICE_Kernel_Text_encoding_type_Type" minOcc
      <xsd:element name="external_standard_id" type="pds:external_standard_id_Type" minOccurs=
      <xsd:element name="kernel_type" type="pds:kernel_type_Type" minOccurs="1" maxOccurs="1">
      <xsd:element name="Data_Location" type="pds:Data_Location_Type" minOccurs="1" maxOccurs=
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Data_Location_Type">
    <xsd:sequence>
```

```
      <xsd:annotation>
        <xsd:documentation>
          The Data Location class provides the location of a digital object.
        </xsd:documentation>
      </xsd:annotation>
      <xsd:element name="file_local_identifier" type="pds:file_local_identifier_Type" mi
      <xsd:element name="offset" type="pds:offset_Type" minOccurs="1" maxOccurs="1"> </
    </xsd:sequence>
  </xsd:complexType>

  <xsd:element name="SPICE_Kernel_Text" type="pds:SPICE_Kernel_Text_Type">
    <xsd:annotation>
      <xsd:documentation>
        The SPICE_Kernel_Text XML Schema
      </xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:schema>
```

.21.3 XML Example

TBD

.22 Special_Constants

The Special Constants class provides a set of values used to indicate special cases that occur in the data.

.22.1 Attributes & Associations

Attribute	**Definition**	**Card.**	**Req?**
error_constant	The error_constant attribute provides a value that indicates the original value was in error.	1	N
invalid_constant	The invalid constant attribute provides a value that indicates the original value was invalid.	1	N
missing_constant	The missing constant attribute provides a value that indicates the original value was missing.	1	N
not_applicable_constant	The not applicable constant attribute provides a value that indicates the original value was not applicable.	1	N
saturated_constant	The saturated constant attribute provides a value that indicates the original value was saturated.	1	N
unknown_constant	The unknown constant attribute provides a value that indicates the original value was unknown.	1	N
Association	**Definition**	**Card.**	**Req?**

.22.2 Schema

TBD

.22.3 XML Example

TBD

.23 Spectrum_2D

The Spectrum 2D class is an extension of array_base and defines a two dimensional spectrum.

.23.1 Attributes & Associations

Attribute	**Definition**	**Card.**	**Req?**
axes	The axes attribute provides a count of the axes.	1	Y
axis_order	The axis order attribute gives the axis index that varies fastest with respect to storage order. **Valid Values:** FIRST_INDEX_FASTEST	1	Y
comment	The comment attribute is a character string expressing one or more remarks or thoughts relevant to the object.	1	N
encoding_type	The encoding type attribute indicates the storage algorithm.	1	Y
local_identifier	The local identifier attribute provides the name of a local object; it is unique within a label. The value of local_identifier should be the class name. If several instances of the same class exist, then a numeric suffix is appended to the class name.	1	Y
Association	**Definition**	**Card.**	**Req?**
data_location	Association for locations.	1	Y

data_object	Composition association for Data Object.	1	Y
has_Array_Axis	Composition association for array axis.		N
has_Array_Element	Composition association for array element.	1	Y

.23.2 Schema

Fix page run-off.

```
<?xml version="1.0" encoding="UTF-8"?>
  <!-- PDS4 XML/Schema for Spectrum_2D_0.1.1.1.c  Fri Oct 29 06:29:36 PDT 2010 -->
  <!-- Generated from the PDS4 Information Model V0.1.1.1.c -->
  <!-- *** This PDS4 product schema is a preliminary deliverable. *** -->
  <!-- *** It is being made available for review and testing. *** -->
  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://pds.nasa.gov/schema/pds4/pds"
    xmlns:pds="http://pds.nasa.gov/schema/pds4/pds"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified"
    version="1.1.1">

    <xsd:include schemaLocation="Extended_Types_0111c.xsd">
      <xsd:annotation>
        <xsd:documentation>PDS (common) Data Dictionary</xsd:documentation>
      </xsd:annotation>
    </xsd:include>

  <!-- <xsd:import namespace="http://pds.nasa.gov/schema/pds4/anyNS"
    schemaLocation="Any_Described_Data_Object_0111c.xsd"/> -->

  <xsd:complexType name="Spectrum_2D_Type">
    <xsd:sequence>
      <xsd:annotation>
        <xsd:documentation>
          The Spectrum 2D class is an extension of array_base and defines a  two dimens
        </xsd:documentation>
      </xsd:annotation>
      <xsd:element name="local_identifier" type="pds:local_identifier_Type" minOccurs="1
      <xsd:element name="comment" type="pds:comment_Type" minOccurs="0" maxOccurs="1"> <
      <xsd:element name="axes" type="pds:Array_2D_axes_Type" minOccurs="1" maxOccurs="1"
```

```
    <xsd:element name="axis_order" type="pds:axis_order_Type" minOccurs="1" maxOccurs="1"> <
    <xsd:element name="encoding_type" type="pds:Array_Base_encoding_type_Type" minOccurs="1"
    <xsd:element name="Data_Location" type="pds:Data_Location_Type" minOccurs="1" maxOccurs=
    <xsd:element name="Array_Axis" type="pds:Array_Axis_Type" minOccurs="2" maxOccurs="2"> <
    <xsd:element name="Array_Element" type="pds:Array_Element_Type" minOccurs="1" maxOccurs=
  </xsd:sequence>
  <xsd:attribute name="base_class" type="xsd:string" fixed="Array_Base"> </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="Data_Location_Type">
  <xsd:sequence>
    <xsd:annotation>
      <xsd:documentation>
        The Data Location class provides the location of a digital object.
      </xsd:documentation>
    </xsd:annotation>
    <xsd:element name="file_local_identifier" type="pds:file_local_identifier_Type" minOccu
    <xsd:element name="offset" type="pds:offset_Type" minOccurs="1" maxOccurs="1"> </xsd:ele
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Array_Axis_Type">
  <xsd:sequence>
    <xsd:annotation>
      <xsd:documentation>
        The Array Axis class is used as a component of the array class and defines an axis o
      </xsd:documentation>
    </xsd:annotation>
    <xsd:element name="elements" type="pds:elements_Type" minOccurs="1" maxOccurs="1"> </xsd
    <xsd:element name="name" type="pds:name_Type" minOccurs="1" maxOccurs="1"> </xsd:element
    <xsd:element name="sequence_number" type="pds:sequence_number_Type" minOccurs="1" maxOcc
    <xsd:element name="unit" type="pds:unit_Type" minOccurs="0" maxOccurs="1"> </xsd:element
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Array_Element_Type">
  <xsd:sequence>
    <xsd:annotation>
      <xsd:documentation>
        The Array Element class is used as a component of the array class and defines an ele
      </xsd:documentation>
    </xsd:annotation>
    <xsd:element name="data_type" type="pds:data_type_Type" minOccurs="1" maxOccurs="1"> </x
    <xsd:element name="scaling_factor" type="pds:scaling_factor_Type" minOccurs="0" maxOccur
    <xsd:element name="unit" type="pds:unit_Type" minOccurs="0" maxOccurs="1"> </xsd:element
    <xsd:element name="value_offset" type="pds:value_offset_Type" minOccurs="0" maxOccurs="1
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="Spectrum_2D" type="pds:Spectrum_2D_Type">
  <xsd:annotation>
```

```
      <xsd:documentation>
        The Spectrum_2D XML Schema
      </xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:schema>
```

.23.3 XML Example

TBD

.24 Spectrum_3D

The Spectrum 3D class is an extension of array_base and defines a three dimensional spectrum.

.24.1 Attributes & Associations

Attribute	Definition	Card.	Req?
axes	The axes attribute provides a count of the axes.	1	Y
axis_order	The axis order attribute gives the axis index that varies fastest with respect to storage order. **Valid Values:** FIRST_INDEX_FASTEST	1	Y
comment	The comment attribute is a character string expressing one or more remarks or thoughts relevant to the object.	1	N
encoding_type	The encoding type attribute indicates the storage algorithm.	1	Y
local_identifier	The local identifier attribute provides the name of a local object; it is unique within a label. The value of local_identifier should be the class name. If several instances of the same class exist, then a numeric suffix is appended to the class name.	1	Y
Association	**Definition**	**Card.**	**Req?**
data_location	Association for locations.	1	Y

data_object	Composition association for Data Object.	1	Y
has_Array_Axis	Composition association for array axis.		N
has_Array_Element	Composition association for array element.	1	Y

.24.2 Schema

Fix page run-off.

```
<?xml version="1.0" encoding="UTF-8"?>
  <!-- PDS4 XML/Schema for Spectrum_3D_0.1.1.1.c  Fri Oct 29 06:29:36 PDT 2010 -->
  <!-- Generated from the PDS4 Information Model V0.1.1.1.c -->
  <!-- *** This PDS4 product schema is a preliminary deliverable. *** -->
  <!-- *** It is being made available for review and testing. *** -->
  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://pds.nasa.gov/schema/pds4/pds"
    xmlns:pds="http://pds.nasa.gov/schema/pds4/pds"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified"
    version="1.1.1">

    <xsd:include schemaLocation="Extended_Types_0111c.xsd">
      <xsd:annotation>
        <xsd:documentation>PDS (common) Data Dictionary</xsd:documentation>
      </xsd:annotation>
    </xsd:include>

  <!-- <xsd:import namespace="http://pds.nasa.gov/schema/pds4/anyNS"
    schemaLocation="Any_Described_Data_Object_0111c.xsd"/> -->

  <xsd:complexType name="Spectrum_3D_Type">
    <xsd:sequence>
      <xsd:annotation>
        <xsd:documentation>
          The Spectrum 3D class is an extension of array_base and defines a  three dimen
        </xsd:documentation>
      </xsd:annotation>
      <xsd:element name="local_identifier" type="pds:local_identifier_Type" minOccurs="1
      <xsd:element name="comment" type="pds:comment_Type" minOccurs="0" maxOccurs="1"> <
      <xsd:element name="axes" type="pds:Array_3D_axes_Type" minOccurs="1" maxOccurs="1"
```

```
    <xsd:element name="axis_order" type="pds:axis_order_Type" minOccurs="1" maxOccurs="1"> <
    <xsd:element name="encoding_type" type="pds:Array_Base_encoding_type_Type" minOccurs="1"
    <xsd:element name="Data_Location" type="pds:Data_Location_Type" minOccurs="1" maxOccurs=
    <xsd:element name="Array_Axis" type="pds:Array_Axis_Type" minOccurs="3" maxOccurs="3"> <
    <xsd:element name="Array_Element" type="pds:Array_Element_Type" minOccurs="1" maxOccurs=
  </xsd:sequence>
  <xsd:attribute name="base_class" type="xsd:string" fixed="Array_Base"> </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="Data_Location_Type">
  <xsd:sequence>
    <xsd:annotation>
      <xsd:documentation>
        The Data Location class provides the location of a digital object.
      </xsd:documentation>
    </xsd:annotation>
    <xsd:element name="file_local_identifier" type="pds:file_local_identifier_Type" minOccu
    <xsd:element name="offset" type="pds:offset_Type" minOccurs="1" maxOccurs="1"> </xsd:ele
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Array_Axis_Type">
  <xsd:sequence>
    <xsd:annotation>
      <xsd:documentation>
        The Array Axis class is used as a component of the array class and defines an axis o
      </xsd:documentation>
    </xsd:annotation>
    <xsd:element name="elements" type="pds:elements_Type" minOccurs="1" maxOccurs="1"> </xsd
    <xsd:element name="name" type="pds:name_Type" minOccurs="1" maxOccurs="1"> </xsd:element
    <xsd:element name="sequence_number" type="pds:sequence_number_Type" minOccurs="1" maxOcc
    <xsd:element name="unit" type="pds:unit_Type" minOccurs="0" maxOccurs="1"> </xsd:element
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Array_Element_Type">
  <xsd:sequence>
    <xsd:annotation>
      <xsd:documentation>
        The Array Element class is used as a component of the array class and defines an ele
      </xsd:documentation>
    </xsd:annotation>
    <xsd:element name="data_type" type="pds:data_type_Type" minOccurs="1" maxOccurs="1"> </x
    <xsd:element name="scaling_factor" type="pds:scaling_factor_Type" minOccurs="0" maxOccur
    <xsd:element name="unit" type="pds:unit_Type" minOccurs="0" maxOccurs="1"> </xsd:element
    <xsd:element name="value_offset" type="pds:value_offset_Type" minOccurs="0" maxOccurs="1
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="Spectrum_3D" type="pds:Spectrum_3D_Type">
  <xsd:annotation>
```

```
      <xsd:documentation>
        The Spectrum_3D XML Schema
      </xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:schema>
```

.24.3 XML Example

TBD

.25 Stream_Delimited

The Stream Delimited class defines a simple spreadsheet.

.25.1 Attributes & Associations

Attribute	Definition	Card.	Req?
comment	The comment attribute is a character string expressing one or more remarks or thoughts relevant to the object.	1	N
encoding_type	The encoding type attribute indicates the storage algorithm.	1	Y
external_standard_id	The external standard id attribute provides the formal name of a standard not under PDS governance.	1	Y
field_delimiter	The field_delimiter provides the character or characters that indicate the end of a character string. **Valid Values:** 0x09 0x3B 0x7C 0x2C	1	Y
fields	The fields attribute provides a count of the fields.	1	Y
local_identifier	The local identifier attribute provides the name of a local object; it is unique within a label. The value of local_identifier should be the class name. If several instances of the same class exist, then a numeric suffix is appended to the class name.	1	Y
maximum_record_length	The maximum record length attribute sets an upper (inclusive) bound on the number of bytes in a record.	1	Y

record_delimiter	The record delimiter attribute provides the character or characters used to indicate the end of a record. **Valid Values:** 0xOA 0xOD 0xOD_0xOA	1	Y
records	The records attribute provides a count of records.	1	Y
Association	**Definition**	**Card.**	**Req?**
data_location	Association for locations.	1	Y
data_object	Composition association for Data Object.	1	Y
has_stream_record	Composition association for record.	?VARIABLE	Y

.25.2 Schema

Fix page run-off.

```
<?xml version="1.0" encoding="UTF-8"?>
  <!-- PDS4 XML/Schema for Stream_Delimited_0.1.1.1.c  Fri Oct 29 06:29:36 PDT 2010 -->
  <!-- Generated from the PDS4 Information Model V0.1.1.1.c -->
  <!-- *** This PDS4 product schema is a preliminary deliverable. *** -->
  <!-- *** It is being made available for review and testing. *** -->
  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://pds.nasa.gov/schema/pds4/pds"
    xmlns:pds="http://pds.nasa.gov/schema/pds4/pds"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified"
    version="1.1.1">

    <xsd:include schemaLocation="Extended_Types_0111c.xsd">
      <xsd:annotation>
        <xsd:documentation>PDS (common) Data Dictionary</xsd:documentation>
      </xsd:annotation>
    </xsd:include>
```

```
<!-- <xsd:import namespace="http://pds.nasa.gov/schema/pds4/anyNS"
  schemaLocation="Any_Described_Data_Object_0111c.xsd"/> -->

<xsd:complexType name="Stream_Delimited_Type">
  <xsd:sequence>
    <xsd:annotation>
      <xsd:documentation>
        The Stream Delimited class defines a simple spreadsheet.
      </xsd:documentation>
    </xsd:annotation>
    <xsd:element name="local_identifier" type="pds:local_identifier_Type" minOccurs="1" max
    <xsd:element name="comment" type="pds:comment_Type" minOccurs="0" maxOccurs="1"> </xsd:
    <xsd:element name="encoding_type" type="pds:Stream_Delimited_encoding_type_Type" minOcc
    <xsd:element name="external_standard_id" type="pds:Stream_Delimited_external_standard_i
    <xsd:element name="field_delimiter" type="pds:field_delimiter_Type" minOccurs="1" maxOc
    <xsd:element name="fields" type="pds:fields_Type" minOccurs="1" maxOccurs="1"> </xsd:el
    <xsd:element name="maximum_record_length" type="pds:maximum_record_length_Type" minOccu
    <xsd:element name="record_delimiter" type="pds:record_delimiter_Type" minOccurs="1" max
    <xsd:element name="records" type="pds:records_Type" minOccurs="1" maxOccurs="1"> </xsd:
    <xsd:element name="Data_Location" type="pds:Data_Location_Type" minOccurs="1" maxOccurs
    <xsd:element name="Stream_Delimited_Record" type="pds:Stream_Delimited_Record_Type" min
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Data_Location_Type">
  <xsd:sequence>
    <xsd:annotation>
      <xsd:documentation>
        The Data Location class provides the location of a digital object.
      </xsd:documentation>
    </xsd:annotation>
    <xsd:element name="file_local_identifier" type="pds:file_local_identifier_Type" minOccu
    <xsd:element name="offset" type="pds:offset_Type" minOccurs="1" maxOccurs="1"> </xsd:el
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Stream_Delimited_Record_Type">
  <xsd:sequence>
    <xsd:annotation>
      <xsd:documentation>
        The Stream Delimited Record class is a component of the stream delimited (spreadshe
      </xsd:documentation>
    </xsd:annotation>
    <xsd:element name="Stream_Delimited_Grouped_Sequence" type="pds:Stream_Delimited_Groupe
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Stream_Delimited_Grouped_Sequence_Type">
  <xsd:sequence>
    <xsd:annotation>
      <xsd:documentation>
```

```
        The Stream Delimited Grouped Sequence class is a component of the grouped stre
      </xsd:documentation>
    </xsd:annotation>
    <xsd:element name="repetitions" type="pds:repetitions_Type" minOccurs="0" maxOccu
    <xsd:element name="Stream_Delimited_Field_Sequence" type="pds:Stream_Delimited_Fie
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Stream_Delimited_Field_Sequence_Type">
  <xsd:sequence>
    <xsd:annotation>
      <xsd:documentation>
        The Stream Delimited Field Sequence class is a component of the grouped strean
      </xsd:documentation>
    </xsd:annotation>
    <xsd:element name="Stream_Delimited_Field" type="pds:Stream_Delimited_Field_Type"
    <xsd:element name="Stream_Delimited_Grouped_Sequence" type="pds:Stream_Delimited_C
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Stream_Delimited_Field_Type">
  <xsd:sequence>
    <xsd:annotation>
      <xsd:documentation>
        The Stream Delimited Field class is a component of the stream delimited  (spre
      </xsd:documentation>
    </xsd:annotation>
    <xsd:element name="field_name" type="pds:field_name_Type" minOccurs="1" maxOccurs=
    <xsd:element name="field_number" type="pds:field_number_Type" minOccurs="0" maxOcc
    <xsd:element name="field_data_type" type="pds:field_data_type_Type" minOccurs="1"
    <xsd:element name="field_format" type="pds:field_format_Type" minOccurs="0" maxOcc
    <xsd:element name="minimum_scaled_value" type="pds:minimum_scaled_value_Type" min(
    <xsd:element name="maximum_scaled_value" type="pds:maximum_scaled_value_Type" min(
    <xsd:element name="field_min_logical" type="pds:field_min_logical_Type" minOccurs=
    <xsd:element name="field_max_logical" type="pds:field_max_logical_Type" minOccurs=
    <xsd:element name="field_scaling_factor" type="pds:field_scaling_factor_Type" min(
    <xsd:element name="field_value_offset" type="pds:field_value_offset_Type" minOccu
    <xsd:element name="field_unit" type="pds:field_unit_Type" minOccurs="0" maxOccurs=
    <xsd:element name="field_description" type="pds:field_description_Type" minOccurs=
    <xsd:element name="field_bytes" type="pds:field_bytes_Type" minOccurs="1" maxOccu
    <xsd:element name="Object_Statistics" type="pds:Object_Statistics_Type" minOccurs=
    <xsd:element name="Special_Constants" type="pds:Special_Constants_Type" minOccurs=
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Object_Statistics_Type">
  <xsd:sequence>
    <xsd:annotation>
      <xsd:documentation>
        The Object Statistics class provides a set of values that provide metrics abou
      </xsd:documentation>
```

```
      </xsd:annotation>
      <xsd:element name="local_identifier" type="pds:local_identifier_Type" minOccurs="1" max
      <xsd:element name="description" type="pds:description_Type" minOccurs="0" maxOccurs="1">
      <xsd:element name="maximum" type="pds:maximum_Type" minOccurs="0" maxOccurs="1"> </xsd:e
      <xsd:element name="md5_checksum" type="pds:md5_checksum_Type" minOccurs="0" maxOccurs="1
      <xsd:element name="mean" type="pds:mean_Type" minOccurs="0" maxOccurs="1"> </xsd:element
      <xsd:element name="median" type="pds:median_Type" minOccurs="0" maxOccurs="1"> </xsd:ele
      <xsd:element name="minimum" type="pds:minimum_Type" minOccurs="0" maxOccurs="1"> </xsd:e
      <xsd:element name="sample_bit_mask" type="pds:sample_bit_mask_Type" minOccurs="0" maxOcc
      <xsd:element name="standard_deviation" type="pds:standard_deviation_Type" minOccurs="0"
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Special_Constants_Type">
    <xsd:sequence>
      <xsd:annotation>
        <xsd:documentation>
          The Special Constants class provides a set of values used to indicate special cases
        </xsd:documentation>
      </xsd:annotation>
      <xsd:element name="error_constant" type="pds:error_constant_Type" minOccurs="0" maxOccu
      <xsd:element name="invalid_constant" type="pds:invalid_constant_Type" minOccurs="0" max
      <xsd:element name="missing_constant" type="pds:missing_constant_Type" minOccurs="0" max
      <xsd:element name="not_applicable_constant" type="pds:not_applicable_constant_Type" min
      <xsd:element name="saturated_constant" type="pds:saturated_constant_Type" minOccurs="0"
      <xsd:element name="unknown_constant" type="pds:unknown_constant_Type" minOccurs="0" max
    </xsd:sequence>
  </xsd:complexType>

  <xsd:element name="Stream_Delimited" type="pds:Stream_Delimited_Type">
    <xsd:annotation>
      <xsd:documentation>
        The Stream_Delimited XML Schema
      </xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:schema>
```

.25.3 XML Example

TBD

.26 Stream_Delimited_Field

The Stream Delimited Field class is a component of the stream delimited (spreadsheet) record class and defines a field of the record.

.26.1 Attributes & Associations

Attribute	Definition	Card.	Req?
field_bytes	The field bytes attribute provides the maximum number of bytes allowed for a field.	1	Y
field_data_type	The field data type indicates the machine representation in which a field value is digitally stored. **Valid Values:** ASCII_Boolean_TF ASCII_Date_YMD ASCII_Integer ASCII_Real ASCII_AnyURI ASCII_Date_DOY ASCII_Date_Time_DOY ASCII_Date_Time_UTC ASCII_Date_Time_YMD ASCII_LID ASCII_LIDVID ASCII_MD5_Checksum ASCII_Short_String_Collapsed ASCII_Text_Preserved ASCII_Short_String_Preserved ASCII_Time ASCII_VID ASCII_DOI ASCII_Integer_Binary ASCII_Integer_Hex ASCII_NonNegative_Integer	1	Y
field_description	The field description attribute provides a statement, picture in words, or account that describes a field.	1	N
field_format	The field format attribute indicates how a field value is to be presented in printable characters. The allowed formats are Fortran and C printing formats.	1	N

field_max_logical	The field max logical attribute provides the maximum valid operating range for an instrument (with the same scaling factors and offsets applied to the data values).	1	N
field_min_logical	The field min logical attribute provides the minimum valid operating range for an instrument (with the same scaling factors and offsets applied to the data values).	1	N
field_name	The field name attribute provides a word or a combination of words by which a field is known.	1	Y
field_number	The field number attribute provides the location of a field in a series of fields.	1	N
field_scaling_factor	The field scaling factor attribute provides a number which scales or multiplies with the value of the field.	1	N
field_unit	The field unit attribute indicates the unit of measurement associated with a field value. **Valid Values:** mol deg rad m**2 m km kg K C s ms m/s L m**3	1	N
field_value_offset	The field value offset attribute provides a number that indicates a displacement from the value in the field.	1	N
maximum_scaled_value	The maximum_scaled_value attribute provides the maximum value after application of ”scaling_factor” and ”offset”.	1	N
minimum_scaled_value	The minimum_scaled_value attribute provides the minimum value after application of ”scaling_factor” and ”offset”.	1	N

Association	Definition	Card.	Req?
associated_Object_Statistics	Association for object statistics.	1	N
associated_Special_Constants	Association for special constants.	1	N

.26.2 Schema

TBD

.26.3 XML Example

TBD

.27 Stream_Delimited_Field_Sequence

The Stream Delimited Field Sequence class is a component of the grouped stream delimited (spreadsheet) class. It defines a set of fields or a nested set of fields.

.27.1 Attributes & Associations

Attribute	**Definition**	**Card.**	**Req?**
Association	**Definition**	**Card.**	**Req?**
has_Stream_Delimited_Field	Composition association for field.		N

.27.2 Schema

TBD

.27.3 XML Example

TBD

.28 Stream_Delimited_Grouped_Sequence

The Stream Delimited Grouped Sequence class is a component of the grouped stream delimited (spreadsheet) class. It defines a set of fields.

.28.1 Attributes & Associations

Attribute	Definition	Card.
repetitions	The repetitions attribute indicates the number of occurrences.	1
Association	**Definition**	**Card.**
has_Stream_Delimited_Field_Sequence	Composition association for field sequence.	?VARIABL

.28.2 Schema

TBD

.28.3 XML Example

TBD

.29 Stream_Delimited_Record

The Stream Delimited Record class is a component of the stream delimited (spreadsheet) class and defines a record of the spreadsheet.

.29.1 Attributes & Associations

Attribute	**Definition**	**Card.**	**Req?**
Association	**Definition**	**Card.**	**Req?**
has_Stream_Deleted_Field_Sequence	Composition association for field sequence.	?VARIABLE	Y

.29.2 Schema

TBD

.29.3 XML Example

TBD

.30 Table_Binary

The Table Binary class is an extension of table base and defines a simple binary table.

.30.1 Attributes & Associations

Attribute	**Definition**	**Card.**	**Req?**
comment	The comment attribute is a character string expressing one or more remarks or thoughts relevant to the object.	1	N
encoding_type	The encoding type attribute indicates the storage algorithm.	1	Y
fields	The fields attribute provides a count of the fields.	1	Y
local_identifier	The local identifier attribute provides the name of a local object; it is unique within a label. The value of local_identifier should be the class name. If several instances of the same class exist, then a numeric suffix is appended to the class name.	1	Y
record_bytes	The record bytes attribute provides a count of the bytes in a record.	1	Y
records	The records attribute provides a count of records.	1	Y
Association	**Definition**	**Card.**	**Req?**
data_location	Association for locations.	1	Y

data_object	Composition association for Data Object.	1	Y
has_Record	Composition association for record.	?VARIABLE	Y

.30.2 Schema

Fix page run-off.

```
<?xml version="1.0" encoding="UTF-8"?>
  <!-- PDS4 XML/Schema for Table_Binary_0.1.1.1.c  Fri Oct 29 06:29:36 PDT 2010 -->
  <!-- Generated from the PDS4 Information Model V0.1.1.1.c -->
  <!-- *** This PDS4 product schema is a preliminary deliverable. *** -->
  <!-- *** It is being made available for review and testing. *** -->
  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://pds.nasa.gov/schema/pds4/pds"
    xmlns:pds="http://pds.nasa.gov/schema/pds4/pds"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified"
    version="1.1.1">

    <xsd:include schemaLocation="Extended_Types_0111c.xsd">
      <xsd:annotation>
        <xsd:documentation>PDS (common) Data Dictionary</xsd:documentation>
      </xsd:annotation>
    </xsd:include>

  <!-- <xsd:import namespace="http://pds.nasa.gov/schema/pds4/anyNS"
    schemaLocation="Any_Described_Data_Object_0111c.xsd"/> -->

  <xsd:complexType name="Table_Binary_Type">
    <xsd:sequence>
      <xsd:annotation>
        <xsd:documentation>
          The Table Binary class is an extension of table base and defines a simple binary tab
        </xsd:documentation>
      </xsd:annotation>
      <xsd:element name="local_identifier" type="pds:local_identifier_Type" minOccurs="1" max(
      <xsd:element name="comment" type="pds:comment_Type" minOccurs="0" maxOccurs="1"> </xsd:e
      <xsd:element name="encoding_type" type="pds:Table_Base_Binary_encoding_type_Type" minOcc
      <xsd:element name="fields" type="pds:fields_Type" minOccurs="1" maxOccurs="1"> </xsd:ele
      <xsd:element name="record_bytes" type="pds:record_bytes_Type" minOccurs="1" maxOccurs="1
      <xsd:element name="records" type="pds:records_Type" minOccurs="1" maxOccurs="1"> </xsd:e
      <xsd:element name="Data_Location" type="pds:Data_Location_Type" minOccurs="1" maxOccurs=
```

```
      <xsd:element name="Table_Record_Binary" type="pds:Table_Record_Binary_Type" minOcc
    </xsd:sequence>
    <xsd:attribute name="base_class" type="xsd:string" fixed="Table_Base"> </xsd:attribu
  </xsd:complexType>

  <xsd:complexType name="Data_Location_Type">
    <xsd:sequence>
      <xsd:annotation>
        <xsd:documentation>
          The Data Location class provides the location of a digital object.
        </xsd:documentation>
      </xsd:annotation>
      <xsd:element name="file_local_identifier" type="pds:file_local_identifier_Type" mi
      <xsd:element name="offset" type="pds:offset_Type" minOccurs="1" maxOccurs="1"> </x
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Table_Record_Binary_Type">
    <xsd:sequence>
      <xsd:annotation>
        <xsd:documentation>
          The Table Record Binary class is a component of the table class and defines a
        </xsd:documentation>
      </xsd:annotation>
      <xsd:element name="Table_Binary_Field" type="pds:Table_Binary_Field_Type" minOccu
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Table_Binary_Field_Type">
    <xsd:sequence>
      <xsd:annotation>
        <xsd:documentation>
          The Table Binary Field class is a component of the table record class and defi
        </xsd:documentation>
      </xsd:annotation>
      <xsd:element name="field_name" type="pds:field_name_Type" minOccurs="1" maxOccurs=
      <xsd:element name="field_number" type="pds:field_number_Type" minOccurs="0" maxOcc
      <xsd:element name="field_data_type" type="pds:Table_Binary_Field_field_data_type_T
      <xsd:element name="field_location" type="pds:field_location_Type" minOccurs="1" ma
      <xsd:element name="field_length" type="pds:field_length_Type" minOccurs="1" maxOcc
      <xsd:element name="field_format" type="pds:Table_Field_Extended_field_format_Type"
      <xsd:element name="minimum_scaled_value" type="pds:minimum_scaled_value_Type" minO
      <xsd:element name="maximum_scaled_value" type="pds:maximum_scaled_value_Type" minO
      <xsd:element name="field_min_logical" type="pds:field_min_logical_Type" minOccurs=
      <xsd:element name="field_max_logical" type="pds:field_max_logical_Type" minOccurs=
      <xsd:element name="field_scaling_factor" type="pds:field_scaling_factor_Type" minO
      <xsd:element name="field_value_offset" type="pds:field_value_offset_Type" minOccu
      <xsd:element name="field_unit" type="pds:field_unit_Type" minOccurs="0" maxOccurs=
      <xsd:element name="field_description" type="pds:field_description_Type" minOccurs=
      <xsd:element name="Object_Statistics" type="pds:Object_Statistics_Type" minOccurs=
      <xsd:element name="Special_Constants" type="pds:Special_Constants_Type" minOccurs=
```

```
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Object_Statistics_Type">
    <xsd:sequence>
      <xsd:annotation>
        <xsd:documentation>
          The Object Statistics class provides a set of values that provide metrics about the
        </xsd:documentation>
      </xsd:annotation>
      <xsd:element name="local_identifier" type="pds:local_identifier_Type" minOccurs="1" max(
      <xsd:element name="description" type="pds:description_Type" minOccurs="0" maxOccurs="1">
      <xsd:element name="maximum" type="pds:maximum_Type" minOccurs="0" maxOccurs="1"> </xsd:e
      <xsd:element name="md5_checksum" type="pds:md5_checksum_Type" minOccurs="0" maxOccurs="1
      <xsd:element name="mean" type="pds:mean_Type" minOccurs="0" maxOccurs="1"> </xsd:element
      <xsd:element name="median" type="pds:median_Type" minOccurs="0" maxOccurs="1"> </xsd:ele
      <xsd:element name="minimum" type="pds:minimum_Type" minOccurs="0" maxOccurs="1"> </xsd:e
      <xsd:element name="sample_bit_mask" type="pds:sample_bit_mask_Type" minOccurs="0" maxOcc
      <xsd:element name="standard_deviation" type="pds:standard_deviation_Type" minOccurs="0"
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Special_Constants_Type">
    <xsd:sequence>
      <xsd:annotation>
        <xsd:documentation>
          The Special Constants class provides a set of values used to indicate special cases
        </xsd:documentation>
      </xsd:annotation>
      <xsd:element name="error_constant" type="pds:error_constant_Type" minOccurs="0" maxOccu
      <xsd:element name="invalid_constant" type="pds:invalid_constant_Type" minOccurs="0" max(
      <xsd:element name="missing_constant" type="pds:missing_constant_Type" minOccurs="0" max(
      <xsd:element name="not_applicable_constant" type="pds:not_applicable_constant_Type" min(
      <xsd:element name="saturated_constant" type="pds:saturated_constant_Type" minOccurs="0"
      <xsd:element name="unknown_constant" type="pds:unknown_constant_Type" minOccurs="0" max(
    </xsd:sequence>
  </xsd:complexType>

  <xsd:element name="Table_Binary" type="pds:Table_Binary_Type">
    <xsd:annotation>
      <xsd:documentation>
        The Table_Binary XML Schema
      </xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:schema>
```

.30.3 XML Example

TBD

.31 Table_Binary_Field

The Table Binary Field class is a component of the table record class and defines a field of the record. This extension defines a binary field.

.31.1 Attributes & Associations

Attribute	Definition	Card.	Req?
field_data_type	The field data type indicates the machine representation in which a field value is digitally stored.	1	Y
field_description	The field description attribute provides a statement, picture in words, or account that describes a field.	1	N
field_format	The field format attribute indicates how a field value is to be presented in printable characters. The allowed formats are Fortran and C printing formats.	1	N
field_length	The field length attribute indicates the maximum number of characters allowed for a value in a field.	1	Y
field_location	The field location attribute indicates the starting position for a field in a record.	1	Y
field_max_logical	The field max logical attribute provides the maximum valid operating range for an instrument (with the same scaling factors and offsets applied to the data values).	1	N

field_min_logical	The field min logical attribute provides the minimum valid operating range for an instrument (with the same scaling factors and offsets applied to the data values).	1	N
field_name	The field name attribute provides a word or a combination of words by which a field is known.	1	Y
field_number	The field number attribute provides the location of a field in a series of fields.	1	N
field_scaling_factor	The field scaling factor attribute provides a number which scales or multiplies with the value of the field.	1	N
field_unit	The field unit attribute indicates the unit of measurement associated with a field value. **Valid Values:** mol deg rad m**2 m km kg K C s ms m/s L m**3	1	N
field_value_offset	The field value offset attribute provides a number that indicates a displacement from the value in the field.	1	N
maximum_scaled_value	The maximum_scaled_value attribute provides the maximum value after application of ”scaling_factor” and ”offset”.	1	N
minimum_scaled_value	The minimum_scaled_value attribute provides the minimum value after application of ”scaling_factor” and ”offset”.	1	N
Association	**Definition**	**Card.**	**Req?**
associated_Object_Statistics	Association for object statistics.	1	N

associated_Special_Constants	Association for special constants.	1	N

.31.2 Schema

TBD

.31.3 XML Example

TBD

.32 Table_Binary_Field_Sequence

The Table Binary Field Sequence class is a component of the grouped table class. It defines a set of fields or a nested set of fields.

.32.1 Attributes & Associations

Attribute	**Definition**	**Card.**	**Req?**
Association	**Definition**	**Card.**	**Req?**
has_Table_Binary_Grouped	Composition association for grouped sequence of fields.		N

.32.2 Schema

TBD

.32.3 XML Example

TBD

.33 Table_Binary_Grouped

The Table Binary Grouped class is an extension of table base and defines a simple binary table that allows repeating groups of fields.

.33.1 Attributes & Associations

Attribute	**Definition**	**Card.**	**Req?**
comment	The comment attribute is a character string expressing one or more remarks or thoughts relevant to the object.	1	N
encoding_type	The encoding type attribute indicates the storage algorithm.	1	Y
fields	The fields attribute provides a count of the fields.	1	Y
local_identifier	The local identifier attribute provides the name of a local object; it is unique within a label. The value of local_identifier should be the class name. If several instances of the same class exist, then a numeric suffix is appended to the class name.	1	Y
record_bytes	The record bytes attribute provides a count of the bytes in a record.	1	Y
records	The records attribute provides a count of records.	1	Y
Association	**Definition**	**Card.**	**Req?**

data_location	Association for locations.	1	Y
data_object	Composition association for Data Object.	1	Y
has_Record	Composition association for record.	?VARIABLE	Y

.33.2 Schema

TBD

.33.3 XML Example

TBD

.34 Table_Binary_Grouped_Bit_Field

The Table Binary Grouped Bit Field class is a component of the table record class and defines a field of the record. This extension defines a binary grouped bit field.

.34.1 Attributes & Associations

Attribute	Definition	Card.	Req?
bit_mask	The bit mask attribute provides a bit mask	1	N
bits	The bits attrribute provides the number of bits	1	Y
field_data_type	The field data type indicates the machine representation in which a field value is digitally stored.	1	Y
field_description	The field description attribute provides a statement, picture in words, or account that describes a field.	1	N
field_format	The field format attribute indicates how a field value is to be presented in printable characters. The allowed formats are Fortran and C printing formats.	1	N
field_length	The field length attribute indicates the maximum number of characters allowed for a value in a field.	1	Y
field_location	The field location attribute indicates the starting position for a field in a record.	1	Y

field_max_logical	The field max logical attribute provides the maximum valid operating range for an instrument (with the same scaling factors and offsets applied to the data values).	1	N
field_min_logical	The field min logical attribute provides the minimum valid operating range for an instrument (with the same scaling factors and offsets applied to the data values).	1	N
field_name	The field name attribute provides a word or a combination of words by which a field is known.	1	Y
field_number	The field number attribute provides the location of a field in a series of fields.	1	N
field_scaling_factor	The field scaling factor attribute provides a number which scales or multiplies with the value of the field.	1	N
field_unit	The field unit attribute indicates the unit of measurement associated with a field value. **Valid Values:** mol deg rad m**2 m km kg K C s ms m/s L m**3	1	N
field_value_offset	The field value offset attribute provides a number that indicates a displacement from the value in the field.	1	N
maximum_scaled_value	The maximum_scaled_value attribute provides the maximum value after application of "scaling_factor" and "offset".	1	N
minimum_scaled_value	The minimum_scaled_value attribute provides the minimum value after application of "scaling_factor" and "offset".	1	N

start_bit	The start bit attribute provides the position of the first bit within an ordered sequence of bits.	1	Y
Association	**Definition**	**Card.**	**Req?**

.34.2 Schema

TBD

.34.3 XML Example

TBD

.35 Table_Binary_Grouped_Field

The Table Binary Grouped Field class is a component of the table record class and defines a field of the record. This extension defines a binary grouped field.

.35.1 Attributes & Associations

Attribute	Definition	Card.	Req?
field_data_type	The field data type indicates the machine representation in which a field value is digitally stored.	1	Y
field_description	The field description attribute provides a statement, picture in words, or account that describes a field.	1	N
field_format	The field format attribute indicates how a field value is to be presented in printable characters. The allowed formats are Fortran and C printing formats.	1	N
field_length	The field length attribute indicates the maximum number of characters allowed for a value in a field.	1	Y
field_location	The field location attribute indicates the starting position for a field in a record.	1	Y
field_max_logical	The field max logical attribute provides the maximum valid operating range for an instrument (with the same scaling factors and offsets applied to the data values).	1	N

field_min_logical	The field min logical attribute provides the minimum valid operating range for an instrument (with the same scaling factors and offsets applied to the data values).	1	N
field_name	The field name attribute provides a word or a combination of words by which a field is known.	1	Y
field_number	The field number attribute provides the location of a field in a series of fields.	1	N
field_scaling_factor	The field scaling factor attribute provides a number which scales or multiplies with the value of the field.	1	N
field_unit	The field unit attribute indicates the unit of measurement associated with a field value. **Valid Values:** mol deg rad m**2 m km kg K C s ms m/s L m**3	1	N
field_value_offset	The field value offset attribute provides a number that indicates a displacement from the value in the field.	1	N
maximum_scaled_value	The maximum_scaled_value attribute provides the maximum value after application of "scaling_factor" and "offset".	1	N
minimum_scaled_value	The minimum_scaled_value attribute provides the minimum value after application of "scaling_factor" and "offset".	1	N
Association	**Definition**	**Card.**	**Req?**

.35.2 Schema

TBD

.35.3 XML Example

TBD

.36 Table_Binary_Grouped_Sequence

The Table Binary Grouped Sequence class is a component of the grouped table class. It defines a set of fields.

.36.1 Attributes & Associations

Attribute	**Definition**	**Card.**	**Req?**
repetitions	The repetitions attribute indicates the number of occurrences.	1	N
Association	**Definition**	**Card.**	**Req?**
has_Table_Binary_Field_Sequence	Composition association for field sequence.	?VARIABLE	Y

.36.2 Schema

TBD

.36.3 XML Example

TBD

.37 Table_Character_Field

The Table Character Field class is a component of the table record class and defines a field of the record. This extension defines a character field.

.37.1 Attributes & Associations

Attribute	Definition	Card.	Req?
field_data_type	The field data type indicates the machine representation in which a field value is digitally stored.	1	Y
field_description	The field description attribute provides a statement, picture in words, or account that describes a field.	1	N
field_format	The field format attribute indicates how a field value is to be presented in printable characters. The allowed formats are Fortran and C printing formats.	1	N
field_length	The field length attribute indicates the maximum number of characters allowed for a value in a field.	1	Y
field_location	The field location attribute indicates the starting position for a field in a record.	1	Y
field_max_logical	The field max logical attribute provides the maximum valid operating range for an instrument (with the same scaling factors and offsets applied to the data values).	1	N

field_min_logical	The field min logical attribute provides the minimum valid operating range for an instrument (with the same scaling factors and offsets applied to the data values).	1	N
field_name	The field name attribute provides a word or a combination of words by which a field is known.	1	Y
field_number	The field number attribute provides the location of a field in a series of fields.	1	N
field_scaling_factor	The field scaling factor attribute provides a number which scales or multiplies with the value of the field.	1	N
field_unit	The field unit attribute indicates the unit of measurement associated with a field value. **Valid Values:** mol deg rad m**2 m km kg K C s ms m/s L m**3	1	N
field_value_offset	The field value offset attribute provides a number that indicates a displacement from the value in the field.	1	N
maximum_scaled_value	The maximum_scaled_value attribute provides the maximum value after application of ”scaling_factor” and ”offset”.	1	N
minimum_scaled_value	The minimum_scaled_value attribute provides the minimum value after application of ”scaling_factor” and ”offset”.	1	N
Association	**Definition**	**Card.**	**Req?**
associated_Object_Statistics	Association for object statistics.	1	N

associated_Special_Constants	Association for special constants.	1	N

.37.2 Schema

TBD

.37.3 XML Example

TBD

.38 Table_Character_Field_Sequence

The Table Character Field Sequence class is a component of the grouped table class. It defines a set of fields or a nested set of fields.

.38.1 Attributes & Associations

Attribute	**Definition**	**Card.**	**Req?**
Association	**Definition**	**Card.**	**Req?**
has_Table_Character_Grouped	Composition association for grouped sequence of fields.		N

.38.2 Schema

TBD

.38.3 XML Example

TBD

.39 Table_Character_Grouped

The Table Character Grouped class is an extension of table base and defines a simple character table that allows repeating groups of fields.

.39.1 Attributes & Associations

Attribute	**Definition**	**Card.**	**Req?**
comment	The comment attribute is a character string expressing one or more remarks or thoughts relevant to the object.	1	N
encoding_type	The encoding type attribute indicates the storage algorithm.	1	Y
fields	The fields attribute provides a count of the fields.	1	Y
local_identifier	The local identifier attribute provides the name of a local object; it is unique within a label. The value of local_identifier should be the class name. If several instances of the same class exist, then a numeric suffix is appended to the class name.	1	Y
record_bytes	The record bytes attribute provides a count of the bytes in a record.	1	Y
records	The records attribute provides a count of records.	1	Y
Association	**Definition**	**Card.**	**Req?**

data_location	Association for locations.	1	Y
data_object	Composition association for Data Object.	1	Y
has_Record	Composition association for record.	?VARIABLE	Y

.39.2 Schema

TBD

.39.3 XML Example

TBD

.40 Table_Character_Grouped_Field

The Table Character Grouped Field class is a component of the table record class and defines a field of the record. This extension defines a character grouped field.

.40.1 Attributes & Associations

Attribute	Definition	Card.	Req?
field_data_type	The field data type indicates the machine representation in which a field value is digitally stored.	1	Y
field_description	The field description attribute provides a statement, picture in words, or account that describes a field.	1	N
field_format	The field format attribute indicates how a field value is to be presented in printable characters. The allowed formats are Fortran and C printing formats.	1	N
field_length	The field length attribute indicates the maximum number of characters allowed for a value in a field.	1	Y
field_location	The field location attribute indicates the starting position for a field in a record.	1	Y
field_max_logical	The field max logical attribute provides the maximum valid operating range for an instrument (with the same scaling factors and offsets applied to the data values).	1	N

field_min_logical	The field min logical attribute provides the minimum valid operating range for an instrument (with the same scaling factors and offsets applied to the data values).	1	N
field_name	The field name attribute provides a word or a combination of words by which a field is known.	1	Y
field_number	The field number attribute provides the location of a field in a series of fields.	1	N
field_scaling_factor	The field scaling factor attribute provides a number which scales or multiplies with the value of the field.	1	N
field_unit	The field unit attribute indicates the unit of measurement associated with a field value. **Valid Values:** mol deg rad m**2 m km kg K C s ms m/s L m**3	1	N
field_value_offset	The field value offset attribute provides a number that indicates a displacement from the value in the field.	1	N
maximum_scaled_value	The maximum_scaled_value attribute provides the maximum value after application of ”scaling_factor” and ”offset”.	1	N
minimum_scaled_value	The minimum_scaled_value attribute provides the minimum value after application of ”scaling_factor” and ”offset”.	1	N
Association	**Definition**	**Card.**	**Req?**

.40.2 Schema

TBD

.40.3 XML Example

TBD

.41 Table_Character_Grouped_Sequence

The Table Character Grouped Sequence class is a component of the grouped table class. It defines a set of fields.

.41.1 Attributes & Associations

Attribute	Definition	Card.	Req?
repetitions	The repetitions attribute indicates the number of occurrences.	1	N
Association	**Definition**	**Card.**	**Req?**
has_Table_Character_Field_Sequence	Composition association for field sequence.	?VARIABLE	Y

.41.2 Schema

TBD

.41.3 XML Example

TBD

.42 Table_Field_File_Specification_Name

The Table_Field_File_Specification_Name class defines a table field that provides a file name, file extension, and relative directory path to a product label.

.42.1 Attributes & Associations

Attribute	Definition	Card.	Req?
field_data_type	The field data type indicates the machine representation in which a field value is digitally stored.	1	Y
field_description	The field description attribute provides a statement, picture in words, or account that describes a field.	1	N
field_format	The field format attribute indicates how a field value is to be presented in printable characters. The allowed formats are Fortran and C printing formats.	1	N
field_length	The field length attribute indicates the maximum number of characters allowed for a value in a field.	1	Y
field_location	The field location attribute indicates the starting position for a field in a record.	1	Y
field_name	The field name attribute provides a word or a combination of words by which a field is known.	1	Y
field_number	The field number attribute provides the location of a field in a series of fields.	1	N

Association	Definition	Card.	Req?

.42.2 Schema

TBD

.42.3 XML Example

TBD

.43 Table_Field_LID

The Table_Field_LID class defines a table field that provides the logical identifier for a product.

.43.1 Attributes & Associations

Attribute	Definition	Card.	Req?
field_data_type	The field data type indicates the machine representation in which a field value is digitally stored.	1	Y
field_description	The field description attribute provides a statement, picture in words, or account that describes a field.	1	N
field_format	The field format attribute indicates how a field value is to be presented in printable characters. The allowed formats are Fortran and C printing formats.	1	N
field_length	The field length attribute indicates the maximum number of characters allowed for a value in a field.	1	Y
field_location	The field location attribute indicates the starting position for a field in a record.	1	Y
field_name	The field name attribute provides a word or a combination of words by which a field is known.	1	Y
field_number	The field number attribute provides the location of a field in a series of fields.	1	N

Association	Definition	Card.	Req?

.43.2 Schema

TBD

.43.3 XML Example

TBD

.44 Table_Field_LIDVID

The Table Field LIDVID class defines a table field that provides the logical identifier and version identifier for a product.

.44.1 Attributes & Associations

Attribute	Definition	Card.	Req?
field_data_type	The field data type indicates the machine representation in which a field value is digitally stored.	1	Y
field_description	The field description attribute provides a statement, picture in words, or account that describes a field.	1	N
field_format	The field format attribute indicates how a field value is to be presented in printable characters. The allowed formats are Fortran and C printing formats.	1	N
field_length	The field length attribute indicates the maximum number of characters allowed for a value in a field.	1	Y
field_location	The field location attribute indicates the starting position for a field in a record.	1	Y
field_name	The field name attribute provides a word or a combination of words by which a field is known.	1	Y
field_number	The field number attribute provides the location of a field in a series of fields.	1	N

Association	Definition	Card.	Req?

.44.2 Schema

TBD

.44.3 XML Example

TBD

.45 Table_Record_Binary

The Table Record Binary class is a component of the table class and defines a record of the table. This extension defines a binary record.

.45.1 Attributes & Associations

Attribute	**Definition**	**Card.**	**Req?**
Association	**Definition**	**Card.**	**Req?**
has_Table_Field	Composition association for field.	?VARIABLE	Y

.45.2 Schema

TBD

.45.3 XML Example

TBD

.46 Table_Record_Binary_Grouped

The Table Record Binary Grouped class is a component of the table class and defines a record of the table. This extension defines a binary record with grouped fields.

.46.1 Attributes & Associations

Attribute	Definition	Card.	Req?
Association	**Definition**	**Card.**	**Req?**
has_Table_Field	Composition association for field.	?VARIABLE	Y

.46.2 Schema

TBD

.46.3 XML Example

TBD

.47 Table_Record_Character

The Table Record Character class is a component of the table class and defines a record of the table. This extension defines a character record.

.47.1 Attributes & Associations

Attribute	**Definition**	**Card.**	**Req?**
Association	**Definition**	**Card.**	**Req?**
has_Table_Field	Composition association for field.	?VARIABLE	Y

.47.2 Schema

TBD

.47.3 XML Example

TBD

.48 Table_Record_Character_Grouped

The Table Record Character Grouped class is a component of the table class and defines a record of the table. This extension defines a character record with grouped fields.

.48.1 Attributes & Associations

Attribute	**Definition**	**Card.**	**Req?**
Association	**Definition**	**Card.**	**Req?**
has_Table_Field	Composition association for field.	?VARIABLE	Y

.48.2 Schema

TBD

.48.3 XML Example

TBD

.49 Table_Record_Inventory_LID

The Table_Record_Inventory_LID class defines the record of an inventory for a secondary collection.

.49.1 Attributes & Associations

Attribute	**Definition**	**Card.**	**Req?**
has_Field_File_Specification_Name	Associated File_Specification_Name	1	Y
has_Field_LID	Associated LID	1	Y
Association	**Definition**	**Card.**	**Req?**

.49.2 Schema

TBD

.49.3 XML Example

TBD

.50 Table_Record_Inventory_LIDVID

The Table_Record_Inventory_LIDVID class defines the record of an inventory for a primary collection.

.50.1 Attributes & Associations

Attribute	**Definition**	**Card.**	**Req?**
has_Field_File_Specification_Name	Associated File_Specification_Name	1	Y
has_Field_LIDVID	Associated LIDVID	1	Y
Association	**Definition**	**Card.**	**Req?**

.50.2 Schema

TBD

.50.3 XML Example

TBD

.51 XML_Schema

The XML Schema class defines a text stream file containing XML Schema.

.51.1 Attributes & Associations

Attribute	**Definition**	**Card.**	**Req?**
comment	The comment attribute is a character string expressing one or more remarks or thoughts relevant to the object.	1	N
encoding_type	The encoding type attribute indicates the storage algorithm.	1	Y
external_standard_id	The external standard id attribute provides the formal name of a standard not under PDS governance.	1	Y
local_identifier	The local identifier attribute provides the name of a local object; it is unique within a label. The value of local_identifier should be the class name. If several instances of the same class exist, then a numeric suffix is appended to the class name.	1	Y
Association	**Definition**	**Card.**	**Req?**
data_location	Association for locations.	1	Y
data_object	Composition association for Data Object.	1	Y

.51.2 Schema

TBD

.51.3 XML Example

TBD

Non-Digital Object Classes

Copied from Digital Object Classes – needs to be updated

PDS4 classes can be divided into three types: digital, conceptual, and physical.

Physical classes are those used to describe actual physical objects that one could touch: a spacecraft, a planet, a camera. Conceptual objects are those that describe a concept, such as a mission, a map projection, or a camera model. Digital objects are those that we traditionally think of as data: an image, a binary table, a document, or a software program.

Appendix A of this document deals exclusively with the digital data classes. Appendix B describes the physical classes, and Appendix C details the conceptual classes.

.52 Context Classes

In PDS4, the concept of a product is slightly different than it was in PDS3. In addition to the traditional data products that included things like images and tables, all of the components of an archive are now considered to be products. Thus, what we used to call a catalog object as described in a MISSION.CAT file is now considered to be a product in its own right, described by a Product_Mission class. Similarly, software and documents are now described by the Document_Set and Software_Set classes (problem here???).

Digital products are further divided into three sub-types: data, software, and documents. The reason for this is that the associated information required for each of these sub-types varies. All data products require a the same set of associated information; documents and software require their own associated pieces of information. The following three sections explain the overall structure of these three sub-types of digital products.

.52.1 Supplemental Classes

The Data Product classes serve as the top level container for a primary digital data object and all of its associated pieces. It provides a place where the identification information for the product is located. It has a place for specifying the product's relationship to a data set, an instrument, a spacecraft or ground-based observation platform, a mission, a target, and a curating node. It provides the most basic information about the structure of the file containing the digital object (record type, size, and checksum). It also contains some of the descriptive information relating to the circumstances under which the data product was acquired (time, spacecraft clock count, etc.). Most importantly, it contains the data object set for the primary data object. An overview of the structure of the Data Product classes is shown in figure (ref).

are typically built around a primary data object such as an image or table.

Acronyms

PDS Planetary Data System. 13–15, 17

Index