JPL D-7669, Part 2

Planetary Data System Standards Reference

September 5, 2011
Version 4.0.4 (beta)

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

PDS4 build 1d, based on:

Information Model Specification
Version 0.4.1.1.f
generated Sat Aug 27 06:57:19 PDT 2011

schema files
built from the PDS4 ontology model V0.4.1.1.f
generated Sat Aug 20 07:14:14 PDT 2011

Protegé ".pont" file
version 3.4.6
build 613
generated Sat Aug 27 07:07:28 PDT 2011

Contents

Change Log

Version	Section (old/new)	Change
4.0.1	Appendix A	Added examples for Image_Grayscale
4.0.2.	Appendix A	Temporarily dropped appendices
		(other changes not fully documented)
4.0.3	*throughout*	Changed references to external documentation to clickable links.
		Updated format of citations and made them clickable to bibliography
		Introduced global acronym handling and made acronyms clickable to definitions listed in appendices
		Made hidden hyperlinks explicit and therefore accessible in printed versions of the document, except where links were non-essential and provided solely for convenience of online readers
		Fixed text wrapping on long URLs that ran off page edge
	1.1/*	**PDS Data Policy** section removed.
	1.2/1.1	List of documents dropped; text updated to refer to complete list in bibliography.
	1.4/1.3	Deleted "standards" from third line.
	1.5/1.4	Completely re-written.
	1.6/*	All references in section 1.6 were transferred to the bibliography and the section was deleted.
	2 & 3	Switched the orders of chapters 2 and 3.
	2/3	Made numerous changes to opening paragraphs of chapter.

2.1/3.1	Significant changes throughout section, including dropping "errata.txt", changing name of "aareadme.xml" and changing cardinality values of files and subdirectories in root directory.
2.2/3.2	Changed "standard product label values" to "product label values"
2.5/3.5	Changed "standard data products" to "regular data products"
2.6/3.6	Changed "standard data products" to "basic data products"
2.7/3.7	Added required files table and completed text.
3.4/2.4	Changed "collections and standard products" to "collections, which in turn identify all of the basic products".
4	Changed "classes, elements, and standard values for the elements" to "classes, attributes, and standard values".
	Changed "classes, elements, and standard values" to "classes, attributes, and standard values."
	Significantly re-wrote introductory section.
	Updated label structure figure.
*/4.1	Added new section **The XML Declaration and Schema Reference**
4.1/4.2	Added text, label snippet, and table of attributes.
4.3/4.4	Added text and label snippet.
4.3.1/4.4.1	Added text and label snippet.
4.3.2/4.4.2	Added text and label snippet.
4.5.1.1/4.6.1.1	Deleted section **The Data_Location Class**.
*/4.6	Added new section **The Closing Tag**.
*/4.7	Added placeholder section **Local Data Dictionaries**.
5	Added "base classes" to end of last sentence.
5.4	Changed two instances of "encoded stream base" to "encoded byte stream".
6.1	Changed title from **Attribute Data Types** to **Character Data Types**.
	Modified text to include character tables.
	Re-organized data_types into subsections.
	Deleted ASCII_Date_Time_UTC.
	Changed ASCII_Integer_Base* to ASCII_Numeric_Base*.
	Added ASCII_Mask.

	Updated max value of ASCII_NonNegative_Integer from 2147483647 to 4294967295.
6.2/*	Deleted section **Character Data Types**.
6.3/6.2	Added text.
6.3.1.1/6.2.1.1	Changed "Least Significant Bit (LSB)" to "Least Significant Byte (LSB) first (also known as *little-endian*)"
	Dropped the SignedByte data type.
6.3.1.2/6.2.1.2	Changed "Least Significant Byte first (LSB) format" to "LSB format"
6.3.1.3/6.2.1.3	Changed "Most Significant Byte first (MSB)" to "Most Significant Byte (MSB) first (also known as *big-endian*)"
	Dropped the SignedByte data type.
6.3.1.4/6.2.1.4	Changed "Most Significant first (MSB) format" to "MSB format"
6.3.3/6.2.3	Added figures for complex types.
*/6.2.4	Added subsection heading for Boolean binary type.
7.4	Sixth bullet under **Rules**, changed "alphabetical" to "alphanumeric".
8/*	**Astronomical Nomenclature** chapter removed.
10.1/9.1	Last sentence: changed "late 2008" to "early 2011"
10.1.1/9.1.1	Second paragraph: updated WGCCRE list to include 2011 report
10.2/9.2	Second line: changed "bodys" to "body's"
	Second paragraph: fixed broken Greek symbols
	Third paragraph, last sentence: changed "See the Planetary Science Data Dictionary (PDS, 2008), chapter 2" to "See chapter 12 of this document"
	Fourth paragraph, first sentence: fixed incorrect citation; changed "(IAU, 2000)" to "(IAU, 2002)"
	Fourth paragraph: changed "Earths" to "Earth's"
10.3/9.3	First paragraph: changed "Suns" to "Sun's" (twice)
10.4.1/9.4.1	Throughout: fixed missing degree symbols
	First paragraph: changed "bodys" to "body's"
	Second paragraph: added "the" before "Sun"
	Second paragraph: fixed Greek symbol

	Third paragraph: changed “existing PDS data sets” to “existing sets of data archived with the PDS”
10.7/9.7	First paragraph: fixed superscripted text
13/12	Added note about SPICE time systems after final paragraph.
13.3.1/12.3.1	Changed case of attributes from upper to lower.
	Changed “elements” to “attributes”.
	Removed “and catalog files”.
13.3.2/12.3.2	Changed “elements” to “attributes”.
13.3.2.1/12.3.2.1	Changed case of attributes from upper to lower.
	Changed “elements” to “attributes”.
13.3.2.4.1/12.3.2.4.1	Replaced all instances of “areocentric” with “planetocentric’.
	Replaced all instances of “ARA” with “RA”
	Changed case of attributes from upper to lower.
	Changed “element” to “attribute”.
	Changed “Zaxis” to “Z-axis”
	Added line breaks and indenting in LTST equations.
13.3.2.4.2/12.3.2.4.2	Replaced “areocentric” with “planetocentric”
appendices	Fixed chapter and section numbering in appendices
Appendix A	Re-attached Appendix A, “Digital Object Classes”
	Completely re-wrote introductory section.
	Fixed automation code to include only terminal user classes
	Fixed automation code to include inherited attributes and associations
	In “Attributes & Associations” sections, changed redundant “Req?” column to “Values” column
	Dropped schema samples
	Improved automated generation of XML examples
Acronyms	Added list of acronyms
Bibliography	Added bibliography

Responses to Build 1B Assessment

Reviewer	Section	Comment	Response
Acton	1.2	*SPICE Archive Preparation Guide* insert "and other SPICE archive producers" after (NAIF) and drop "that NAIF follows"	I've dropped the list of documents from this section so this point is moot.
	1.6	Drop Davies, 1991 reference. Replace with Seidelmann, et al., 2007.	
	1.7	Questions inclusion of CCSDS PVL Specification	
	2.1	Add "Sub-directories are indicated by use of bold font." to end of first paragraph.	
		Try to avoid splitting directory / file contents tables across page boundaries.	
	2.2	How does errata file mentioned here relate to the "errata.xml" file mentioned earlier?	
	2.8	Add, "The geometry directory contains any observation geometry data other than 'SPICE'-type data files."	
		(In response to question about relationship between cartographic info and geometry directory) add "cartographic information can be found (generally is found) within some SPICE files, but could also be found–explicitly or implicitly–in other 'geometry files.' "	
	2.9	After final paragraph, add "Note that it is common practice for a flight project to produce and archive a SPICE bundle, usually containing all SPICE data used on a project. As a consequence, for science data archive bundles this directory is often not used."	

3.1	First paragraph, switch order to "digital, physical or conceptual"	
3.3	What is this? The title of this section says it will discuss something called "collection project."	
4	First paragraph, last line, change "element" to "elements".	
4	Second paragraph, "one label file for every product" – is this true for software? Is a collection of software treated as a single product, or a set of products?	
4.4	"the system files" - what is meant by this?	
5	Add "base classes." to the end of the last sentence.	Done.
5.2	Array base figure has overwritten some text.	
5.3	First paragraph doesn't make sense.	
5.4	Two instances of "encoded stream base" need to be updated to "encoded byte stream".	Fixed.
	Might be useful to provide one or two examples here?	
6.4	In response to question about place of section on multi-valued data types, "Acton doesn't think description of quaternions really belongs in a cartography section."	
6.4.2	Final paragraph, 'Rotations Required Reading' and 'SPICE Quaternion White Paper' – should there be more formal references for these document?	
7.1	Taken as a whole section 7.1.1 is very unclear! Needs to be totally rewritten.	
7.1.1	The use of "vs." (in the title) helps make the following discussion confusing.	
	Third paragraph, what are "HIGH LEVEL VERSION IDs"	
	Fourth paragraph, e.g.	Not sure what's being suggested here.

7.1.2	What is meant by the term "product identifiers"; should this be "product logical identifiers"?	
	Fourth paragraph, change "name for specific" to "name as the specific".	
	Second rule, insert "Allowed" before "character"	
7.1.3	Description of specific identifier: What is this stuff?	Backus-Naur Form (BNF) notation. (This was also used in PDS3 StdRef.)
7.2	Is this the "VID" mentioned earlier? Or something else? (Make this clear.) (The use of "Other" makes it sound as if it is unrelated to VID.)	
7.2.1	Are version IDs required or optional? Suggest changing "are" to "can be".	
7.2.2	Local_identifiers are also used in section 4.4	
	The second paragraph mentions a recommendation to follow; where is it?	
7.2.6	Insert "by" after "constrained only"	
	Fifth paragraph, remove "characters" after "ASCII".	
7.3	Information here is inconsistent with section 2.1.	
7.4	Under **Recommendations**, question about "must have at least one period followed by an extension"	
7.5	For what purpose is the reference at the bottom of section 7.4 provided? Why should a user look here?	
8	I suggest elimination of this chapter. Or replace it with a simple pointer to the IAU website.	Chapter has been deleted.
10	Seems a little odd that there is no mention of "SPICE" within this chapter.	Currently working with C. Acton to address this.
10.2	Third paragraph, at the end of the fourth sentence, put ", with the exception of SPICE products, for which TDB is used."	

10.3	Sixth line, "Suns" missing an apostrophe.	Fixed.
10.4.1	Fourth paragraph, replace everything from JPL DE403 on to "current JPL developmental ephemeris (e.g. DExxx).	
11.3.1	Where does a lander fit in? Consider adding this.	
12	Add a fourth point under "PDS criteria for inclusion of a document...": "4. Is the material free of ITAR restrictions, or can it be modified to be so?"	
	Second paragraph on page 82, contradicted by *PDS4 Concepts Document* which says, "PDS has adopted the PDF/A standard as the archival format for complex documents. UTF-8 encoded text files may be used for simple documents that do not require graphics or special formatting control."	
13	Note that the "SPICE" system, generally used by planetary missions for computing observation geometry, uses additional time systems and time representations.	Added.
13.3.2.4.1	Replace instances of "areocentric" with "planetocentric", and "ARA" with "RA (right ascension)"	Done.
	Middle of p. 87, insert "and" after "in deg".	Inserted missing line breaks after "hours" and "deg" and indented equations, which fixes problem.
13.3.2.4.2	Replace "areocentric" with "planetocentric".	Done.

Bell	general	Once more, formatting / spelling / etc. inconsistencies. Many fragmentary sentences, paragraphs, sections. One problem that this document has over others is that it often refers to URLs with a hyperlink around a word (as in "for examples see here", where "here" is hyperlinked). If one reads the document by printing it (yes, I still like to edit / comment longhand), you have no idea to what the link is pointing (if you can even tell it's a link). If references are to off-line documents (e.g., last line of last paragraph of Sec. 6.4.2?), then these should have real citations, even a bibliographic list of such references.	Most of the worst of these "here" links were deleted with the removal of the chapter on Astronomical Nomenclature. In the remaining cases, I have provided the external link explicitly where it was important. In places where it remains hidden, it is provided solely as a convenience to the online reader, and is not critical to understanding the material presented.
	1	In the second paragraph to chapter 1, it seems to me that the second sentence is backwards. It states "All version 4 data can be described using version 3 labels, but the converse is not true." Shouldn't the reverse hold, that all V3 data can be described with V4 labels, but not all V4 data can be described with V3 labels? Isn't that part of the goal of PDS4?	Actually, the sentence is correct, as written. In response to the Management Council's directive to reduce the complexity of the data storage formats used in PDS4, the DDWG determined that no data interleaving between digital objects would be permitted. Thus, any PDS3 data products that included interleaved data are not compliant with PDS4 and cannot be described using PDS4 classes.
	1.1	PDS Data Policy. I don't know what "safed" data are. I couldn't find the term defined elsewhere, but it was referred to elsewhere as a status type.	Section deleted.
	2.10	I don't understand the point of the XML schema directory. If all schema are registered and in a central location, why does one need a directory for them as well? My understanding was that all versions of the schema should be available via the registry as well, so even that would make this unnecessary.	
	6.3.1.2	Text says "Least Significant Byte", should be "Bit"	No, "Byte" is correct in this case.
	6.3.1.3	Text says "Most Significant Byte", should be "Bit"	Ditto.
	6.3.1.4	Text says "Most Significant", should have "Bit" added.	Changed to an acronym.

	7.1	In general, once again, some lists of items (e.g., mission abbreviations) seem exemplary, others (collection types?) seem exhaustive with no indication either way, leaving it subject to interpretation.	
	7.1.4	Bundle Identifiers. States that LIDs are all uppercase in labels, but that this is not enforced if used on the file system(s). If true, this seems a general statement (LIDs in general) and not specific to Bundle LIDs.	
	7.4	States "remember all SPICE file extensions need to be reserved". Should either list them here or refer back to earlier list (at least by section number). Also states that "Need to include version information in filenames". In what sense? This can't be in the form "::version" as used elsewhere as colons aren't allowed in the file names (by stated convention).	
	10.2	Third paragraph, last line states "See the Planetary Science Data Dictionary (PDS, 2008), chapter 2....". Shouldn't this refer to Chapter 13 of this document? Or it this something else entirely?	Corrected.
	12	Second paragraph on page 82 states "A flat, human-readable ASCII text version of each document must be included on the volume....". This seems at odds with the stated desire (DPH?) to have all documents in PDS/A format with other versions optional.	
	all docs	As an aside, there are clearly a lot of holes left in some of the documents. While that doesn't need pointing out, it did make it difficult in some cases [to] review content.	
Chanover	general	I found this document to be EXCELLENT even if preliminary. It was not until I looked at this document that I finally had an idea of what a PDS4 dataset should actually LOOK like. It does achieve its purposes as outlined by the bulleted items above.	
Gordon	3.2	"...standard product can only belong to one primary collection". Is this how we want to say it? I have an expansion for this comment which I'll introduce directly with the DDWG.	

	general	There are still a lot of TBDs in the SR – primarily because the issues have not been addressed in the Information Model.	
Isbell	1.2	APG..."Is this being replaced by the DPH?" Answer?	The document list in this section has been replaced by a text reference to the bibliography.
		PDS Policies and Processes. Document name and existence?	Ditto.
		Proposers Archive Guide. In progress?	Ditto.
		SPICE Archive Preparation Guide. In progress?	Ditto.
	1.7	As asked within document..."ASCII and UTF-8 standards?"	
	2.10	Final line in section is incomplete..."There is one required subdirectory to the XML schema directory. It contains".	
	5.1	"Table Base" - (ASCII characters xxx and yyy, respectively)." replace with ..."(ASCII character codes 13 and 10, respectively)."	
		Although this section shows an example of fastest-varying order for sample, line, band, I assume other orders are allowed? True? We should probably show example of other order (e.g. band, sample, line) $b_1s_1l_1, b_2s_1l_1, b_3s_1l_1 ... b_1s_2l_1, b_2s_2l_1, b_3s_2l_1 ...$ $b_1s_3l_1, b_2s_3l_1, b_3s_3l_1 ...$	
		Graphical note: The 3-dimensional conceptual image covers (is in front of) part of the "Array_Base" image.	
	5.2	"Array Base" - Needs expanded information along with directive for adequate documentation by data provider.	
	5.3	"Parseable Byte Stream" - Needs expanded information along with directive for adequate documentation by data provider.	

6.3	“Binary Data Types” - Perhaps some added clarity is worthwhile throughout 6.3 regarding *byte* vs *bit* within LSB and MSB acronyms. For example, 6.3.1.1 (Signed LSB integers) states *bit* while 6.3.1.3 (Signed MSB integers) states *byte*. And, 6.3.1.4 does not indicate either *bit* or *byte*.	“Byte” is the correct version of the acronym. Fixed throughout (and hopefully clarified).
7.1.4	Will the bundle identifiers form (xxx:xxx:xxx) cause any string encoding / parsing problems within XML document readers / parsers? Specific example within Image_Grayscale: <logical_identifier> URN:NASA:PDS:example.DPH.Product-ImageGrayscale.MPFL-M-IMP-2-EDR-V1.0 </logical_identifier>	
7.4	“Filenames must be unique within directories.” ...Is now the time for? ... “Filenames must be unique within an archive.”?	
8	Does this entire chapter need to exist? Why have a “Astronomical Nomenclature” section within a PDS standards document? Yes, the first paragraph has some mention of naming convention within archives. Seems we need only this small statement along with the existing IAU reference.	Chapter has been removed.
10	Cartographic Standards: Clearly needs updating for PDS4. I expect to be involved in this process. Already working with Elizabeth, others. For this process, will consider Federal Geographic Data Committee (FGDC) data standards along with input from ISIS cartography development. Larger topics are yet to be address. Side thought: Rather than try to define all anticipated map projections, should we build a base structure to which projections are added as included (or migrated from PDS3) within archives?	
14.1	“The following summary of SI unit information is extracted from The International System of Units.” Include reference of extraction source.	

general	Per various "TBD"s throughout documents, these should be updated (or drafts added) ahead of external review.
	Minor oddity: For some pages, the 'heading' at top of each page can be confusing. That is, the heading title and section numbering there can 'conflict' with the actual upcoming section number. Small example, the start of page 19 is still part of 3.1 even though the heading shows section 3.2 (indicating 3.2 starts on this page). A more glaring example would be the "2.5 Context Directory" table on page 13. It appears to have a "2.6 Data Directory" context (due to the page header). Suggest reducing heading font or heading location (center?). Or, force headings to reflect current chapter rather than section?
all docs	Archive Structure: Some traditional directory names have changed from PDS3 to PDS4. The "negative" of this is familiarity with names within PDS3. However, the new names readily identify a new PDS4 archive (good). So, with the new directory names, I think we need additional clarity on what belongs in each directory (e.g. index files in About directory per my comment a few lines below).
	For the upcoming external review, the sample structure provided should contain all directories (and sample file content) as listed in the standards reference. (e.g. sample is missing "about", "calibration", "geometry", "spice" directories.
	As related, sample aareadme.txt and errata.txt are in PDS3 form vs PDS4 (intended to be PDS4?)
	Directory names should be consistent within documents and sample archive (e.g. xml_schema vs schema, document vs documents, etc).
	Per Standards Ref "2.2 About Directory", I assume the PDS4 equivalent of the PDS3 index.tab file goes here? True? If so, a sample table should be provided.

King	general	The Standards Reference is well written, but has some vagueness which needs to be addressed. The directory structure for a generic "collection" is not described. The directory structure for a "bundle" may not be a Data Provider can deliver data, especially for incremental deliveries which would typically be single collections. This document seems to be at least three separate documents. One covering the PDS specified directory layout, one summarizing external standards which apply to PDS content and usage and one on Nomenclature Rules. having three documents would be preferred. Chapter 3 might be more appropriate in the Concepts document, though a separate document dedicated to describing products might be better focused. That is the intent of the current DPH, this chapter would be a good starting point for a re-write of the DPH.	
	1.1	Data Policy: This section can be omitted. It will be part of the Policies and Processes document. If it is kept it should be adopted data policy verbatim.	Agreed; section deleted.
	1.2	Purpose: The list of PDS documents is nearly orthogonal. The Archive Preparation Guide and Data Providerś Handbook are really tutorials. I would include them as such.	Not sure what you mean by "orthogonal"; however, I have modified the text of the paragraph and dropped the list of documents in lieu of a reference to the bibliography.
	1.4	Audience: Replace "(Those new to the PDS should first read the DPH.)" with something like. The reader should be familiar with the contents of the "PDS Concepts", "PDS Policies and Procedures" and have a general understanding of the "PDS Data Dictionary".	
	1.5	Document Organization: I know this is a standard JPL template for a document, but this section is superfluous. Do I need to know introductory information is in the "Introduction". This section could be removed.	Section has been completely re-written. Hopefully it's a little more useful now.
	1.6	References to "Planetary Science Data Dictionary" and "Planetary Data System Data Preparation Workbook Version 3.1" are outdated. This is for PDS4 so aren't these obsolete?.	

2 [First paragraph] states that "Data Providers must deliver data in the following directory structure". Deliver to who? Presumably to PDS. Operationally, Data Providers would typically deliver Collections and not whole "archive" bundles where multiple related collections are assembled. While PDS may always create archives with this layout, it is not practical for a Data Provider. Conversely, a user may only want individual collections, so when PDS becomes the Data Provider (through user services) it too may not need to deliver a bundle as described.

A brief overview of what information is conveyed in the directory/file layout tables would be useful. I can guess that "fixed" means that the name must be as shown. Also, what does "Class" mean. I can guess that the XML document must contain at least the specified class.

2.1 The description of the contents for each file would be easier to read and scan if they were formatted like the following:

aareadme.xml
A brief overview of the contents...

2.2 A standard heading for the section containing the directory layout table would help to differentiate the narrative from the table. Perhaps "Layout" or "Directory Layout".

Describe the expected files with the same detail as under "Root Directory".

Now I'm a little confused. In "root" there is a "bundle.xml" which describes a bundle. A bundle can refer to "Collections". In "about" is a "collection_misc.xml". Is this directory structure for a Collection? If so, why is "bundle.xml" present? If this is a for a bundle, who (how?) would I specify more than one collection?

2.3 Note in the text, perhaps at the beginning of the paragraph or as the second sentence the cardinality of the directory. In this case its optional. This would be true for each directory section.

Change “facilitate use of the archive” to “facilitate use of the collection”. I’m assuming this is a collection and that both ad hoc and archive collections would adhere to the same structure. Hence the generic reference to “collection”.

Footnote (6) implies that multiple collections can exist. So this overall directory structure is for a “bundle”? This would make much more sense if it was for a “collection”. Since the “root” directory can have any name, then it corresponds to the “collection” name. Adopting a class-style directory layout would be much simpler and would avoid compound name for folders and files.

2.6 The statement “The data directory contains one or more data collections.” means this really a bundle, but what is the standard template (or layout) for a collection? It appears to be

```
root
    collection.xml
    manifest.tab
    products
        product.xml
```

This raises a fundamental question. If “root” sets the context, why does ”products” have a context prefix? Using “browse” as an example the path reference to the products would be “browse/products” which is semantically as rich and does not contain redundant information. The current specification is have semantic overloading, for example “browse/browse_products”.

2.10 States that “There is only one XML schema collection per archive.” Really? What if there is a version change in the schema and some collections adhere to a previous version.

3 Paragraph starting with “As has been stated elsewhere...” is incomplete and unnecessary. It could be stated in one the previous paragraphs that “Any type of object with a label; including images, documents, and spacecraft; is considered a product.”

3.1	The statement “Within the PDS archive, three types of products are recognized: standard products,...” now overloads the term “product”. Another higher order term is needed. Perhaps “entity”, “resource” or “grouping”.
3.2-3.4	Isn’t there a hierarchy (as described in the Concepts document) that Bundles contain Collections and Collections contain Products (see figure below). In section 3.4 that statement “bundles consist of a list of references to products” implies a bundle may contain just “standard products”, in which case, a “bundle” would be no different than a “collection”. This is a clear example why “products” is an overloaded term in this context.
4	Is it “PSDD” or “PDD”? There is no “section IV”.
4.4	Example should be in XML.
5	I need to check the DD, but it would make more sense the name of base classes have a more consistent naming convention. Use “Table” instead of “Table_Base”, “Array” instead of “Array_Base”.
5.1	Refers to “Unencoded Stream Base” which is not mentioned in intro. Delimiters are required to be fixed width. This should be mentioned.
5.2	Change “has two associated classes” to “is composed of two classes” because there is a life-cycle dependency of the two associated classes. That is, Array_Base cannot exist without the Array_Axis and Array_Element members.
5.3	Based on the descriptions it’s hard to tell the difference between Parsable and Encoded. It seems it’s either text (Parsable) or binary (Encoded). Shouldn’t this be Unicode_Byte_Stream and Binary_Byte_Stream?

6.1 Why is there “ASCII_AnyURI” when there is no other instance of “AnyURI” like there is with “ASCII_Date_Time_*”. Using just “ASCII_URI” would suffice.

Why no “ASCII_Text_Collapses”? There are many examples of text (more than 255 characters) where line control and spaces are irrelevant. For example, this is true with most content of HTML pages.

There are “UTF8” versions for strings. Since UTF8 is backward compatible with 8-bit ASCII why not just use “UTF8” everywhere?

We do this because there are situations where we are deliberately limiting the character set to ASCII. In some cases, we don’t want any special characters.

7.1 More introductory text is needed. Some information about how to read the format specification would be helpful. For example, how to read something like:

product logical identifier (lid) = <collection logical identifier> “:” <specific identifier>

7.1.1 Why a 255 character limit? Nice binary number, but unnecessary unless there’s a technical reason. One consideration might be whether the LIDVID can be expressed in a URL. While HTTP does not specify a limit, Internet Explorer limits URLs to 2083 characters. Even with a generous allowance for protocol, host and service strings in a URL length of 1024 would be more friendly.

7.1.2 These rules do [not?] match those in the Concepts document.

7.1.3 Why use underscore (“_”) as a delimiter when “:” is used elsewhere?

Examples would be helpful.

Taken as a whole the specification for an identifier is:

specific identifier = a string, determined by the data provider, that is unique within the collection
product logical identifier (lid) = <collection logical identifier> “:” <specific identifier>
product versioned identifier (lidvid) = <product logical identifierr> “::” <product version id>
collection logical identifier (lid) = <bundle logical identifier> “:” <specific identifier>
collection versioned identifier (lidvid) = <collection logical id> “”::” <collection version id>
bundle logical identifier (lid) = “URN:NASA:PDS:” <specific identifier>
bundle versioned identifier (lidvid) = <bundle logical identifier> “::” <bundle version id>

		This means that a product identifier is dependent on a pre-determined grouping. Since some products (like documents) may be members of multiple groups this can present problems when forming alternate collections and bundles in accordance with the directory naming conventions (described in this same document). While this approach may result in a natural uniqueness (provided bundle names are unique), it's unclear how to reconcile the identifier to directory structure mapping.	
	7.2.3	Digital Object Identifiers: Wouldn't these be local IDs?	
	7.4	Rules: Yippee! PDS finally got it right on file names.	☺
	8.1	Strike sentence "Questions have been asked about the proper English..."	Chapter has been removed in response to other reviewers' comments.
	8.4	Strike sentence "Contrary to some recent media reports..."	Ditto.
	11	Not clear what is the purpose of this chapter. Currently it reads like dictionary definitions, but should provide guidelines for the content. If it remains it should follow and be included in the same document as the directory layout information.	
	12	The requirement that a flat ASCII version be included (I hope) is just un-edited, non-updated text and won't be part of the PDS4 requirements. It should follow and be included in the same directory as the directory layout information.	
Law	general	The Standards Reference has a good start. As it is a "work in progress", lots more effort still needed to complete and clean it up. It has a good layout. Excellent questions and notes are included in this draft. It points out a number of issues have yet to be resolved by the DDWG before the document can become a solid reference.	

Martin	2.1	I think you should standardize on index.html as the entry to a PDS data set. Should provide a standard html template for introducing a data collection. I don't think aareadme.xml is a good idea.	I'm open to changing it, but this is the filename decided upon after a discussion of the topic by the DDWG.
	2.2	Does "About" directory replace ERRATA directory?	To my knowledge, we never had an ERRATA directory. The ERRATA *file* is now included in the "About" directory.
	2.5	Why are some names the same between the file name and the class name but some are different (e.g. context_publication.xml vs Product_Citation)? The context discussion starts talking about system bundles and secondary bundles, neither of which were discussed in the concept document.	
	2.10	"XML Schema Directory" stops prematurely.	
	3	PDS Product consists of one or more objects. This seems to be embracing the "explicit file" architecture of PDS3 where a single label file can point to multiple object files. This design was and is a complete failure. As far as I know no one has ever written any software to work with explicit file objects. "There is always a one-to-one correspondence" for products, but not for objects?	
	3.1	Files are not really discussed in the ensuing paragraph. Objects = digital, physical, conceptual Objects = digital, nondigital Objects = tagged, what is the alternative? Object = class Product = single label + tagged object(s) Types of Products = standard product, collection, bundle	
	4	"PDS product labels are required for describing the contents and format of each individual product" but products are made up of objects, maybe multiple objects stored in different files.	
	5	The Encoded_Byte_Stream is an option for the storage format for PDS4 objects. It is not a "fundamental data structure".	

5.2	Part of the array_base contents listing along the right side of the page is overwritten by the array diagram.	
	Ordering of elements. I think it is odd to require a certain ordering (e.e. "FIRST_INDEX_FASTEST") but still have a keyword for it. Having a keyword makes it seem like an option.	We chose in PDS4 to make everything explicit, so that a data user would not need to have assumed knowledge about the PDS format to understand the data.
5.3	Parsable Byte Stream is MIA.	
5.4	Encoded Byte Stream. This doesn't tell me much. Can I provide data products in PNG, PDF, FITS, GIF, JPEG, CDF, NETCDF, HDF5, TIFF, GEOTIFF?	
6	Chapter 6 is not very useful the way it is presented.	The content of this chapter is necessary to preparing data in compliance with the PDS4 standards. How would you prefer to see it presented?
6.3	Do you really want to allow LSB int's.	A significant proportion of the data already in PDS archives is in LSB format; yes, we do want to allow it.
6.4.1	What about vectors?	
6.4.2	Quaternions. I'm not sure I see the point of the segment on quaternions. There is not enough information to be useful.	
7.1	Identifiers. I don't find this section to be very useful.	
7.2.1	"The minor portion of the version is not-prepadded with zeros; it is simply incremented as an integer..." So a list of items sorted by version number may not come out in sequential order.	
11	Context Information. I would question whether the old delivery mechanism is still appropriate for [this component]. I think we should have an on-line content management system that handles all of this.	That online content management system is included in the system and data model design. However, we still intend to archive the basic components of the system.
12	Documentation. Same as above.	Ditto.

	14	The units in PDS3 were mainly for annotation. There were only a couple cases of non-default units being used in products. This section needs to discuss the rules for using units in PDS4 XML labels.	
McLaughlin	general	I only had time to skim the Standards Reference so my comments are quite limited. First the document is presently organized in a reasonably comprehensible and useful manor.	
	2	However the use of the data modeling terms such as "class" and "cardinality" so early in Chapter 2 seems premature. Cardinality determines if an entity is required or optional in the root structure of an archive and how many occurrences are allowed (such as one-and-only-one or many), so why not use words in place of encoded strings such as "0..1" and "1..*"? Many users of the standards will probably *not* be data modelers.	
	8	Finally Chapter 8 states that PDS requires the use of IAU nomenclature, so why not simply direct the reader to the appropriate IAU and USGS web sites and publications instead of duplicating some of that content in the standards?	The section on Astronomical Nomenclature has been removed.
Simpson	1	These are the standards for PDS Version 4, not version 4 of the PDS standards. It is PDS4 that is not backwards compatible.	"version 4 of the PDS standards" vs. "the standards for PDS, version 4"; not sure I agree with you on this one.
	1.1	Why would PDS funding be used to publish non-compliant data?	Under PDS3, this occurred specifically in the instance of "safed" data. Anyway, the section has been dropped.
		"(Do we wish to drop the statements about waivers?)" – Yes!	Section dropped.
	1.2	"as a reference" – replace "a" with "the" Other documents provide supplementary info, examples, etc. SR is the primary REF doc.	Done.
		Delete *Archive Preparation Guide*; APG is not a REF	This section doesn't describe references, but rather other documents to be used in conjunction with the StdRef. However, I've decided to drop the entire list in lieu of a reference to the bibliography.
		Delete *PDS Policies and Processes*	Ditto

	Delete *Proposer's Archiving Guide*; Not a ref.	Ditto
	Uncertain about inclusion of *SPICE Archive Preparation Guide*	Ditto
	Delete tutorials; not ref.	Ditto
1.4	Third line, delete "standards".	Done
1.5	Outline isn't even close to what follows.	Section has been completely re-written.
1.6	Most of these are NOT cited.	All referenced or other important documents transferred to bibliography; section deleted.
	There are a ton of Refs in Chapter 10 that are missing from this list.	The references in Chapter 10 have been listed in the bibliography.
	IAU Style Manual (SR p.57)	Still need to add to bibliography.
1.8	Need to define "data engineer".	
2	This chapter needs a few words at the beginning to explain format of tables and terms being used (class, cardinality, volume, ...).	A full paragraph of explanatory text has been added.
	What is a directory in the context of PDS4?	
	First paragraph: If the PDS DN changes the structure, what happens to all the URIs? 7.2.4 needs to be very clean and rigorous.	
	First paragraph: is this relevant?	
	Second paragraph: Need to explain what a product is and then identify the possible product types.	
	Footnote to this section: remove the word "multiple".	Done.
2.1	"Root" in the section title should be all uppercase.	
	Is ROOT a named directory, is it implicit, or can it be either?	
	First paragraph: "bundle" is undefined.	Chapters 2 and 3 have been switched; should solve this problem.

		The file names listed here are not in the Information Model.	True that they're not there yet. I believe Steve intends to add them in the future.
		Make sure that tables don't get split across pages.	

Not all comments entered yet.

Slavney	2.1	Root Directory: Specify which subdirectories are required and which are optional.	
		It says one bundle.xml file per root directory. Does that mean one bundle = one data set per root directory? That is, no more putting multiple "data sets" on a "volume"?	
		Do we still need a Geometry directory? I have seldom seen it used. Doesn't SPICE suffice?	
		Re SPICE: Don't we still want to store SPICE data separately at NAIF? Why is this part of the directory structure? What if there are different versions of the SPICE kernels at NAIF, which are the right ones to use? Etc.	
	3	"Data Objects and Products" "Thus, in addition to traditional data products such as images, tables, and histograms, missions and spacecraft are now also treated as products." I can't understand how a mission or a spacecraft can be a product. Products must be digital. What is the benefit of considering physical objects to be products?	
	6.1	"Attribute Data Types" Does the ASCII_Real type include numbers in scientific notation? (e.g. 1.0E-16) If so, say so; if not, make another type.	
	7.1	"Identifiers" Include realistic examples of each type.	
	7.4	"Filenames" Are spaces allowed in file names? Windows allows them but I wish they didn't. Web browsers will try to escape them. Let's say explicitly that they're not allowed.	
		There's a link to a Wikipedia article on file names. Do we have any qualms about using Wikipedia as a reference?	

8	"Astronomical Nomenclature" Why is this here if it's already on the IAU web site? Just provide the link. This detailed information does not seem relevant to PDS archiving.	This section has been removed.
10	I did not review Chapter 10 on Cartographic Standards as it all needs to be re-written. Let's get expert help to do this.	This section has been updated for version 4.0.3 (with the exception of 10.6, which is waiting on definition of the cartographic classes). Lisa Gaddis was consulted on several points of ambiguity.
12	Can we get rid of the requirement that all documents must exist as plain ASCII text? I thought the use of PDS/A would relieve us of this. The Concepts documents seem to say so (section 11).	

Responses to Build 1C Assessment

Reviewer	Section	Comment	Response

Pending completion of 1c assessment.

Chapter 1

Introduction

The Planetary Data System (PDS) Standards Reference is a complete specification for version 4 of the PDS standards. These standards are used to design data storage formats and encode descriptive labels for data stored in the PDS.

Note that version 4 of the PDS standards is *not* backwards compatible with version 3. All version 4 data can be described using version 3 labels, but the converse is not true.

This document *does not* provide a formal definition of the grammar of the eXtensible Markup Language (XML), which is used to encode the PDS4 standards; guidance on this aspect of PDS labels is provided in the Data Provider's Handbook (DPH).

1.1 Purpose

This document is intended to serve as the reference document detailing PDS standards used in the preparation of PDS compliant data. It is to be used within the context of the PDS4 document suite, described in the *Introduction to the PDS4 Document Set*. (Please see the bibliography at the end of this document for a full listing of PDS documents available.) Would like to insert a `\ref` to the bibliography but can't figure out how to assign a target label to it. - EDR

1.2 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.

1.3 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, 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 DPH.)

1.4 Document Organization

This document is divided into three main sections. The first of these, “Archive Structure Standards”, provides detailed information about the structure of all of the components of a PDS archive: labels, data, bundles, collections, and overall organization. The second major section, “Data Content Standards”, is more focused on the nature of *what* is stored in PDS archives. This includes guidance on necessary calibration information, contextual information, and documentation, as well as how to populate cartographic information, date and time values, and various nomenclature rules. Finally, the Appendices contain detailed listings of specific classes used in describing PDS data.

1.5 External Standards

External standards which apply to the content of this document:

What about ASCII and UTF-8 standards? – EDR

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)
- International Standards Organization / International Electrotechnical Commission (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*
- International Standards Organization / Technical Standard (ISO/TS) 15000-3:2004 *electronic business eXtensible Markup Language (ebXML) – Part 3: Registry information model specification (ebRIM)*
- ISO/TS 15000-4:2004 *ebXML – Part 4: Registry services specification (ebRS)*

National Institute of Standards and Technology (NIST):

- NIST Special Publication 330 *The International System of Units (SI)*, United States version of the English text of the eighth edition (2006), Issued March 2008

World Wide Web Consortium (W3C):

- *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

1.6 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/

Part I

Archive Structure Standards

Chapter 2

Data Objects and Products

This chapter still needs a lot of work.

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.

As has been stated elsewhere, in Planetary Data System, standards version 4 (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

2.1 Terminology: Products, Objects, Classes, and Files

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 PDS.

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.

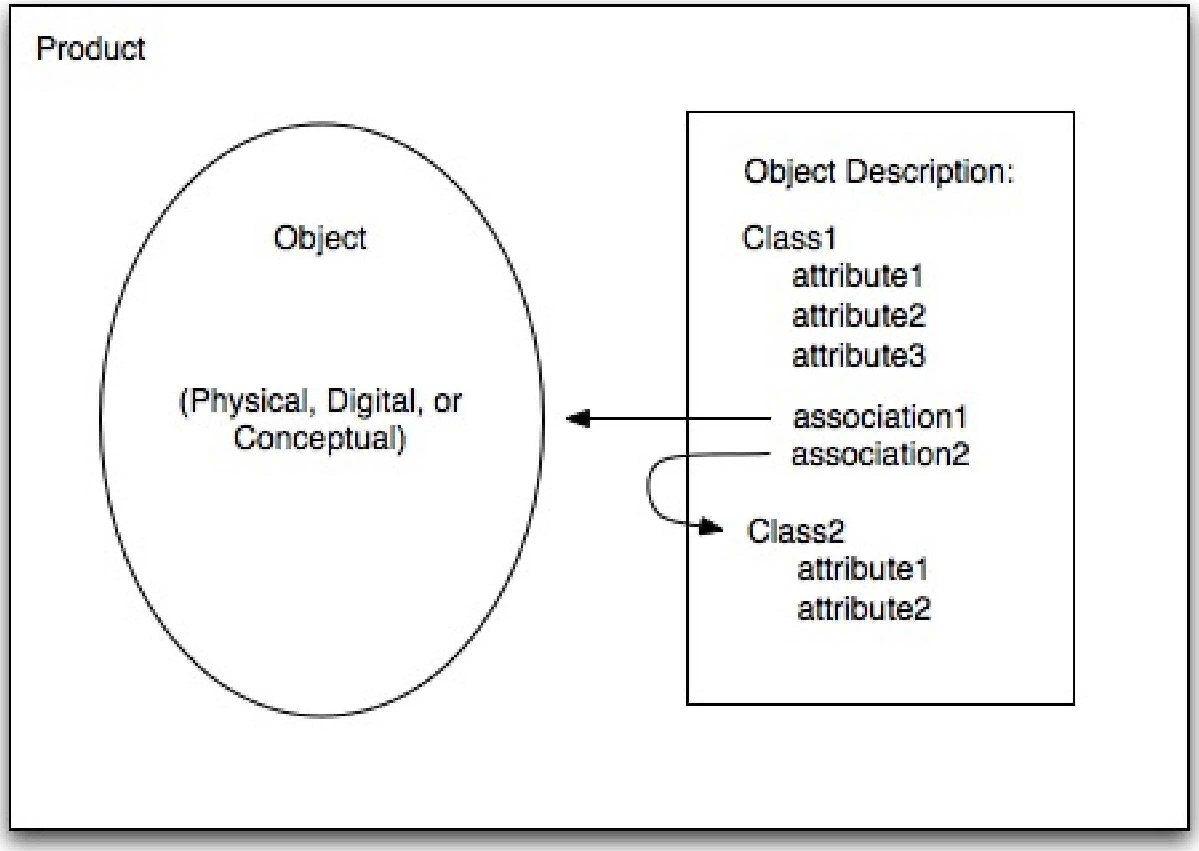
Furthermore, a *list* of products (accompanied by a description of the list) is also considered to be a product. Thus one may identify and track logical sets of products.

Within the PDS archive, three types of products are recognized: *standard* products, *collections*, and *bundles*. Each of these are described 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.



2.2 Standard Products

The lowest level product in the 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.

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

2.3 Collection Products

The collection manifest table shall consist of one record for each product in the collection. Each record shall contain a product LIDVID (see 7.1.1), followed by the file_specification_name for the product's label.

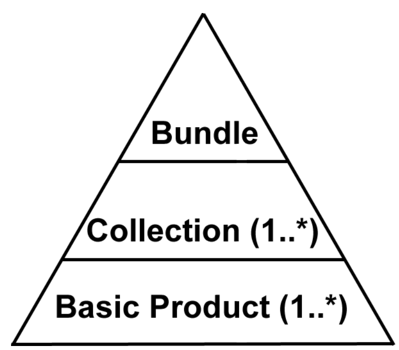
2.4 Bundle Products

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, which in turn identify all of the basic products necessary to perform useful science analysis on the data contained therein.

Chapter 3

Archive Organization and Directory Contents

Archive bundles have a simple hierarchical structure. A bundle has several member collections, each of which has several member basic products.



Data providers will typically deliver data to the PDS in a directory structure that matches the organization of the bundle. PDS discipline nodes may or may not choose to retain that structure in their online repositories.

Data delivered to the PDS shall be assigned to directories based on product type[1] (i.e., observational data of a particular processing level, browse data, documentation, SPICE data, etc.).

The following sections describe the contents of the root directory, followed by the contents of the subdirectories in alphabetical order. In the tables that follow, a **bold-faced** font is used for directories, while plain type is used for files. The "class" identifies the PDS4 product class (insert ref here) used in the specified label to describe associated objects. Files for which no class is identified are described in separate label files. "Cardinality" refers to the number of instances of a particular file or directory type that may occur in an archive. Where a single digit appears, the archive must include precisely that number of files (or directories). Where a range of numbers is specified (using the notation m..n), then any number of files in that range may be included. A minimum of "0" means that the file or directory is optional; a maximum of "*" means that there is no limit on the number of files that may be included.

[1]The one exception is the **about** directory, which may contain products of different types.

3.1 Root Directory

The root directory corresponds to the top level of an archive bundle. It contains several directories and one or two files.

File or Directory Name	Class	Cardinality	Name Fixed?
root	–	1	not fixed
bundle.xml	Product_Bundle	1	fixed
readme.[html,txt,xml]	Product_Document	0..1	fixed
about	–	1	fixed
browse	–	0..1	fixed
calibration	–	0..1	fixed
context	–	1[2]	fixed
data	–	1	fixed
document	–	1[2]	fixed
geometry	–	0..1	fixed
SPICE	–	0..1	fixed
xml_schema	–	1	fixed

The most critical of the files is the XML-formatted bundle label. This label provides the identifying information for the bundle and contains a list, in XML format, identifying all of the component collections of which the bundle is comprised.

The bundle product shall be implemented using the Product_Bundle class. There must be one, and only one, of these files included in each bundle. The filename, "bundle.xml", is fixed.

A second, optional, "readme" file provides a general overview of the bundle contents and organization in human readable format. It may also contain general instructions for use of the archive and contact information for data preparer or discipline node personnel. This file may be formatted either as plain text, HTML, or XML.

The directories have a one-to-one correspondence with the bundle's collections. These directories are described in greater detail in the following sections.

The aareadme shall be implemented using the Product_Document class, and shall contain an Identification_Area_Document class with the product_subclass attribute set to "AAREADME". There must be one, and only one, of these files included in each bundle. The filename, "aareadme.xml", is fixed.

The bundle.xml file identifies all of the component collections of which the archive bundle is comprised.

The errata.xml file identifies and describes errors and/or anomalies found in the archive.

The errata shall be implemented using the Product_Document class, and shall contain an Identification_Area_Document class with the product_subclass attribute set to ”ERRATA”. There must be one, and only one, of these files included in each bundle. The filename, ”errata.xml”, is fixed.

3.2 About Directory

The about directory contains 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. Any updates made to product label values after the archive is ingested may be included in the about directory in tabular form. Examples of the types of information contained in the about directory include:

modification history
errata
updates
tables anaglyph pairs
footprint files
database dumps

File or Directory Name	Class	Cardinality	Name Fixed?
about	–	1	fixed
collection_misc.xml	Collection_Miscellaneous	1	fixed
collection_misc_manifest.tab	–	1	fixed
about_products	–	1[1]	fixed
⋮	(as appropriate)	⋮	⋮

[3]Note that DPH permits multiple **about_products** directories; I’m not sure why this is necessary, rather than having multiple subdirectories underneath it. - EDR

The about collection shall be implemented using a Collection_Miscellaneous class.

3.3 Browse Directory

The browse directory contains one or more browse collections. Each browse collection contains “quick-look” products designed to facilitate use of the archive. Browse products and the browse directory are optional.[2]

File or Directory Name	Class	Cardinality	Name Fixed?
browse	–	0..1	fixed
collection_browse.xml	Collection_Browse	1..*	fixed[3]
collection_browse_manifest.tab	–	1..*	fixed[3]
browse_products	–	1..*	fixed[3]
browse_product1.xml	(as appropriate)	1..*	not fixed
browse_product1.jpg	–	1..*	not fixed
⋮	⋮	⋮	⋮

[3] In the case of multiple browse collections these fixed names may be modified by adding a prefix to distinguish among the collections. For example: edr_collection_browse.xml and rdr_collection_browse.xml.

For each browse collection present in the archive, there shall be one collection_browse_manifest.tab file, one collection_browse.xml file and one browse_products sub-directory. If only one browse collection is present, these names are fixed. If more than one browse collection is present, these names may be prefixed with an additional identifying word terminated in an underscore, to distinguish the collections. In all such cases, the collection label, manifest, and directory must all be clearly identified with one another by utilizing the same prefix.

The browse collection shall be implemented using the Collection_Browse class.

If appropriate, each browse_products directory may contain multiple sub-directories. The structure of these sub-directories is at the discretion of the data provider. A structure that parallels the structure of the data subdirectories is frequently utilized.

3.4 Calibration Directory

The calibration directory contains calibration data and files necessary for the calibration of science data products.

File or Directory Name	Class	Cardinality	Name Fixed?
calibration	–	0..1	fixed

[4] Can there be multiple collections in the **calibration** directory? DPH says "no"; I suspect we might want the possibility. -EDR

[2] Note that there are two alternative methods for including browse objects in an archive. The first is described in this section, namely, to include stand-alone browse products in a browse collection stored in a browse directory. The second method is to store the browse object alongside the primary data object in the science data product. (An example would be to stored a low-resolution JPEG formatted image in a separate file, but in the same directory, as a full-resolution raster-formatted image, and describe them both as a single product using a common PDS label. The latter primarily works in the case of a one-to-one correspondence between browse objects and primary data objects.

File or Directory Name	**Class**	**Cardinality**	**Name Fixed?**
collection_calibration.xml	Collection_Calibration	1..*[4]	fixed
collection_calibration_manifest.tab	–	1..*[4]	fixed
calibration_products	–	1..*[4]	fixed
calibration_product1.xml	(as appropriate)	1..*	not fixed
calibration_product1.tab	–	1..*	not fixed
⋮	⋮	⋮	⋮

[4]Can there be multiple collections in the **calibration** directory? DPH says “no”; I suspect we might want the possibility. - EDR

TBD - Will not be completed for the first release.

3.5 Context Directory

The context 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.

File or Directory Name	**Class**	**Cardinality**	**Name Fixed?**
context	–	1	fixed
collection_context.xml	Collection_Context	1	fixed
collection_context_manifest.tab	–	1	fixed
context_products	–	1	fixed
context_inst.xml	Product_Instrument	0..1[5]	fixed
context_insthost.xml	Product_Instrument_Host	0..1[5]	fixed
context_investigation.xml	Product_Investigation	0..1[5]	fixed
context_node.xml	Product_Node	0..1[5]	fixed
context_personnel.xml	Product_PDS_Affiliate or Product_PDS_Guest[6]	0..1[5]	fixed
context_publication.xml	Product_Citation[6]	0..1[5]	fixed
context_resource.xml	Product_Resource	0..1[5]	fixed
context_target.xml	Product_Target	0..1[5]	fixed

[5]We need to settle on which of these context files are required and which are optional. - EDR

[6]I’m not certain exactly how this is supposed to be implemented. Will probably need to be updated. - EDR

The context products are members of a context collection, the primary copy of which is archived as part of the PDS system bundle. Is this correct? – No. - EDR Thus, in the context of an archive bundle containing regular data products, the context collection is a *secondary* bundle. The context collection shall be implemented using the Collection_Context class.

(Description information about instruments, etc., must be published (whether via the PDS or external publisher. Space Sciences Review permits inclusion of these documents in PDS archives.

instrument, instrument_host, investigation

data set description is required in the document collection; the bundle or collection label will serve the purpose of the context product for the data set.x

data set description should include: citation_desc, acknowledgement text (get samples from Mitch, including acknowledgement of PDS)

data set description summary / abstract will be used in ADS as abstract (probably appear both in document as text and in bundle/collection label in abstract field)

3.6 Data Directory

The data directory contains one or more data collections. Each data collection contains basic data products.

For regular science archives, the data directory is required. For special archives (like a SPICE bundle or a system bundle), the directory is optional.

File or Directory Name	Class	Cardinality	Name Fixed?
data	–	0..1	fixed
collection_data.xml	Collection_Data	1..*	fixed[6]
collection_data_manifest.tab	–	1..*	fixed[6]
data_products	–	1..*	fixed[6]
subdir1	–	1..*	not fixed
data_product1.xml	(as appropriate)	1..*	not fixed
data_product1.tab	–	1..*	not fixed
⋮	⋮	⋮	⋮
subdir2	–	1..*	not fixed
data_productn.xml	(as appropriate)	1..*	not fixed
data_productn.tab	–	1..*	not fixed
⋮	⋮	⋮	⋮

[6]In the case of multiple data collections these fixed names may be modified by adding a prefix to distinguish among the collections. For example: edr_collection_data.xml and rdr_collection_data.xml labels would correspond to edr_data_products and rdr_data_products directories.

There shall be one collection This directory may be further sub-divided to prevent over-crowding of directories and facilitate archive organization.

3.7 Document Directory

The document directory ...

File or Directory Name	Class	Cardinality	Name Fixed?
document	–	1	fixed
collection_document.xml	Collection_Document	1	fixed
collection_document_manifest.tab[7]	–	1	fixed
document_products	–	1	fixed
document_product1.xml	(as appropriate)	1..*	not fixed
document_product1.pdf	–	1..*	not fixed
⋮	⋮	⋮	⋮

[7] DPH has "collection_document_manifest.tex"; need to change. - EDR

3.8 Geometry Directory

The geometry directory contains non-SPICE geometry files (for example, Supplementary Experiment Data Record (SEDR)).

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

3.9 SPICE Kernel Directory

(all SPICE data inclu

(same directory structure as for SPICE Bundles)

The "spice" directory in a PDS bundle can contain individual SPICE kernel files, their XML labels, and XML labels describing primary and/or secondary SPICE collections.

If the PDS bundle contains any SPICE kernel files, these kernel files and their XML labels must be placed in the following subdirectories based on the kernel type:

ck	CK files (spacecraft and instrument orientation data)
dbk	DBK files (databases in SPICE format)
dsk	DSK files (digital shape data for natural bodies)
ek	EK files (events information)
fk	FK files (reference frames definitions)
ik	IK files (instrument parameters and FOV definitions)
lsk	LSK files (leapsecond information)
mk	MK files (meta-kernels listing kernels to be used together)
pck	PCK files (natural body rotation and size/shape constants)
sclk	SCLK files (spacecraft clock correlation data)
spk	SPK files (trajectory and ephemeris data)

All SPICE kernel files provided in the ”spice” directory in the Bundle must be members of a single primarySPICE collection. The XML label describing this SPICE collection must reside in the ”spice” directory.

The ”spice” directory in a Bundle may also contain additional XML labels for secondary SPICE collections, pointing to the kernels in the Bundle and/or kernels in other Bundles or SPICE collections provided in other Bundles.

If SPICE kernel files of a particular type are not included in the Bundle, the subdirectory for that kernel type must be omitted.

If no SPICE kernel files are included in the Bundle, all subdirectories and the primary collection XML label must be omitted. In this case the ”spice” directory must contain only the XML labels for secondary SPICE collections.

If no SPICE kernel files and primary and secondary SPICE collections are included in the Bundle, the ”spice” directory must not be present in the Bundle.

3.10 XML Schema Directory

The XML schema collection contains the XML schema files included in or referenced by XML labels in the archive.

There is one XML schema collection per archive.

collection_xml_schema.xml

collection_xml_schema_manifest.tab

xml_schema_products

There is one required subdirectory to the XML schema directory. It contains

Chapter 4

Labels

PDS product labels are required for describing the contents and format of each individual product within an archive. Specifically, the labels contain the *description objects* that describe corresponding *data objects*. (See chapter 2 for a more detailed explanation.) These description objects are populated using a standard set of classes, attributes, and standard values. These classes, attributes, and standard values are defined in the Planetary Science Data Dictionary (PSDD) (PDS, 2011a).

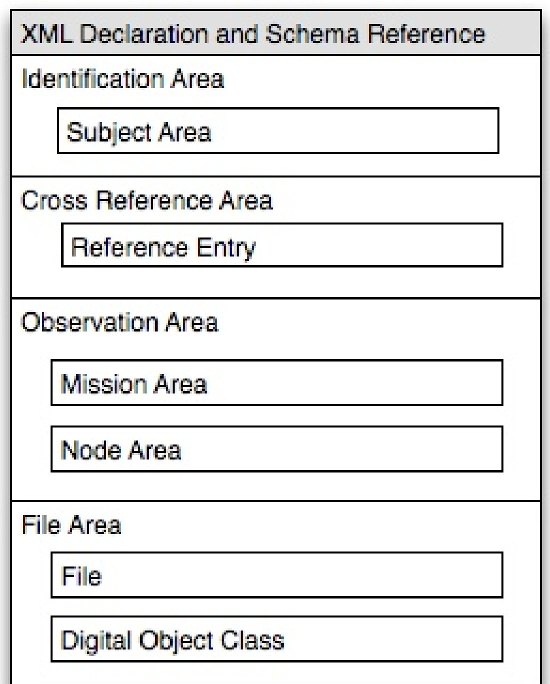


Figure 4.1: Generic PDS label structure

PDS product labels have a general structure, shown in Figure 4.1, that is used for the vast majority of products. The following sections, organized in accordance with this structure, provide general descriptions of the classes used in PDS labels, plus details about variations seen in specific types of products. Additionally, appendices A through C of this document provide detailed descriptions of the data object classes, a complete listing of their attributes and associations, and examples of XML label snippets to demonstrate how the classes are used in labeling PDS products. (Note: not all appendices completed yet.)

Under the PDS4 standard, all product labels are detached from the data files they describe. Label files must end with the file extension ".xml". There is one label for every product. Each label may describe one or more data files.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Product_Array_2D_Image xmlns="http://pds.nasa.gov/schema/pds4/pds"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://pds.nasa.gov/schema/pds4/pds Product_Array_2D_Image_0300e.xsd ">
```

Figure 4.2: The XML Declaration and Schema Reference

```xml
  <Identification_Area>
    <logical_identifier>urn:nasa:pds:mpfl-m-imp-raw:data:1246943630-r-0074051101</logical_identifier>
    <version_id>2.0</version_id>
    <product_class>Product_Array_2D_Image</product_class>
    <title>Filter 5 in 4 tiers - fourth quad monster pan</title>
    <alternate_id>IMP_EDR-1246943630-REGULAR-0074051101</alternate_id>
    <last_modification_date_time>1998-07-14T00:36:08.000Z</last_modification_date_time>
    <Subject_Area>
      <target_name>MARS</target_name>
      <data_set_name>MPFL-M-IMP-2-EDR-V1.0</data_set_name>
      <instrument_name>IMAGER FOR MARS PATHFINDER</instrument_name>
      <instrument_host_name>MARS PATHFINDER LANDER</instrument_host_name>
      <investigation_name>MARS PATHFINDER</investigation_name>
    </Subject_Area>
  </Identification_Area>
```

Figure 4.3: The Identification_Area

4.1 The XML Declaration and Schema Reference

As stated earlier in this document, the PDS4 data model is expressed in XML. In order for a PDS label to adequately meet the XML standard, it must be both “well formed” and “valid”. A well formed label must have correct XML syntax; a valid label must conform to the rules of an XML schema document (XSD). XSDs provide the rules governing the content of a specific XML document class; they are provided by the PDS and may only be modified through extensions to the classes and with the approval of a PDS discipline node.

The first few lines of each label (see Figure 4.2) include an XML declaration statement and a reference to the schema with which the label complies.

The first line is the XML declaration. It indicates that this is an XML document, that it adheres to version 1.0 of the XML standard, and that it is encoded using the UTF-8 character set.

The following lines identify the PDS namespace, indicate the standard with which the schema complies, and provide the name of the relevant schema document.

4.2 The Identification Area

The purpose of the Identification Area in a PDS label is to provide the unique identifier for the product the label describes. Since this identifier is used to track the product in the PDS registry, every PDS product must have an Identification Area.

The two most critical attributes in the Identification Area are the logical_identifier and the version_id. Together these constitute the product’s LIDVID, an identifier for this product that is unique in the world. The LIDVID is used to ingest the product into the PDS registry and is used by other products to provide a reference to this product. For example, a browse version of this image would indicate its full resolution source image by providing the LIDVID of the source product. (For details on how to construct LIDVIDs and other identifiers, consult sections 7.1 and 7.2 of this document.)

The product_class attribute provides the name of the product class used to describe the product. This should be the same as the name of the reference schema describing this product that was identified in the top few lines of the label. This information is used by the PDS registry to assist in product tracking and reporting.

The title attribute is a name for the product. It is a simple text string, with no specific format requirements. It is not necessary for the title to be unique for each product.

These four required attributes constitute the core of the Identification Area. Different types of products utilize different Identification_Area subclasses, which may have additional requirements beyond that of the generic Identification Area. Table 4.1 provides a complete list of Identification_Area attributes and indicates which of these are restricted in the various subclasses.

4.2.1 The Subject Area

The "Subject_Area" of the label (see Figure 4.3) pro

4.3 The Cross_Reference Area

4.4 The Observation Area

The "Observation_Area" of the label is the area where the observation parameters are listed. The top portion of this area includes a relatively small set of attributes that common to the entire PDS. These are attributes that are deemed to be widely enough used that it makes sense for them to have common definitions across all PDS disciplines. More specific classes and attributes are located in the "Mission_Area" and "Node_Area" (see below).

```
<Observation_Area>
  <comment>comment</comment>
  <start_date_time>1997-07-07T05:13:42.763Z</start_date_time>
  <stop_date_time>not applicable</stop_date_time>
  <local_true_solar_time>13:39:12</local_true_solar_time>
  <mission_phase_name>token</mission_phase_name>
  <planet_day_number>3</planet_day_number>
  <spacecraft_clock_cnt_partition>0</spacecraft_clock_cnt_partition>
  <spacecraft_clock_start_count>1246943630</spacecraft_clock_start_count>
  <Mission_Area>
  ...
  </Mission_Area>
  <Node_Area>
  ...
  </Node_Area>
</Observation_Area>
```

Attribute or Association	**Definition**	**Cardinality**	**Identification_Area_Bundle**	**Identification_Area_Collection**	**Identification_Area_Document**	**Identification_Area_Manifest**	**Identification_Area_Product**	**Identification_Area_System**
logical_identifier	A logical identifier identifies the set of all versions of an object. It is an object identifier without a version.	1						
version_id	The version id attribute gives the version.	1						
product_class	The product_class attribute identifies the generic class of the product. For example, the product Product_Table_Character would have product_class=Product_Table_Character.	1						**R**
title	The name given to the resource. Typically, a Title will be a name by which the resource is formally known. - Dublin Core - The title is used to refer to an object in a version independent manner.	1						
alternate_title	The alternate title attribute provides one or more alternate names for a product.	0..*						
alternate_id	The alternate id attribute is an additional identifier supplied by the data provider. This identifier has no role in any system function.	0..*						
last_modification_date_time	The last modification date time attribute gives the most recent date and time that a change was made.	0..1						
product_subclass	The product_subclass attribute provides the name of a subclass under a product_class. For example, User_Manual is a subclass of Product_Document.	0..1			**R**			
contains_primary_member	The contains_primary_member attribute indicates whether a collection contains products that are primary members of the collection.	-		**R**				
home	The home attribute indicates where an object resides. It provides sufficient information to be able to access the object.	-						**R**
subject_area	The subject_area association is a relationship to Subject_Area.	0..1			**R**		**R**	

Table 4.1: Identification_Area

4.4.1 The Mission Area

The “Mission_Area” of the label provides the container for all mission-specific classes and attributes. These classes and attributes are defined in a local data dictionary (see section 4.7), and must be prefixed with the namespace identifier applicable to that local dictionary.

```
<Mission_Area>
  <mpf:application_packet_id>34</mpf:application_packet_id>
  <mpf:application_packet_name>SCI_IMG_3</mpf:application_packet_name>
  <mpf:auto_exposure_data_cut>3000</mpf:auto_exposure_data_cut>
  <mpf:auto_exposure_pixel_fraction>1.0000</mpf:auto_exposure_pixel_fraction>
  <mpf:frame_id>BOTH</mpf:frame_id>
</Mission_Area>
```

4.4.2 The Node Area

The “Node_Area” of the label provides the container for all discipline-specific classes and attributes. These are the classes and attributes that are defined in various PDS node-level local data dictionaries (see section 4.7); they must be prefixed with the relevant node’s namespace identifier.

```
<Node_Area>
  <img:Camera_Parameters>
    <img:exposure_duration>46.0</img:exposure_duration>
    <img:exposure_type>AUTO</img:exposure_type>
    <img:filter_name>L670_R670</img:filter_name>
    <img:filter_number>5</img:filter_number>
  </img:Camera_Parameters>
</Node_Area>
```

Note that it is possible to utilize classes and attributes pulled from multiple node dictionaries. In this case, a separate Node_Area would be used for each node’s elements.

Schemas and dictionaries for node level classes and attributes are currently only available in the Information Model Specification and in the http://pds.nasa.gov/schema/pds4/generic/[node] directories. They may be included as an appendix in a future version of this document.

4.5 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.

```
<File_Area_Observational>
  <File>
    <local_identifier>FILE</local_identifier>
    <creation_date_time>1998-07-14T00:36:08.000Z</creation_date_time>
    <file_name>i943630r.img</file_name>
    <file_size>138240</file_size>
    <max_record_bytes>512</max_record_bytes>
    <md5_checksum>fa1db0cfb1cea71e438ab791a6ee766d</md5_checksum>
    <records>270</records>
  </File>
  <Array_2D_Image base_class="Array_Base">
    ...
  </Array_2D_Image>
  <Header>
    ...
  </Header>
</File_Area_Observational>
```

4.5.1 The Digital Object Classes

4.6 The Closing Tag

In order to for the XML label to be well formed, it must end with a closing tag for the class opened in the second line of the label (see section 4.1, above).

```
</Product_Array_2D_Image>
```

4.7 Local Data Dictionaries

TBD; I wonder if this needs to be an independent chapter. This will include information about the correct format of local data dictionaries and how to reference them in schemas. (Nomenclature rules for classes and attributes are discussed in the chapter on nomenclature.)

Chapter 5

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 base classes.

5.1 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 American Standard Code for Information Interchange (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.

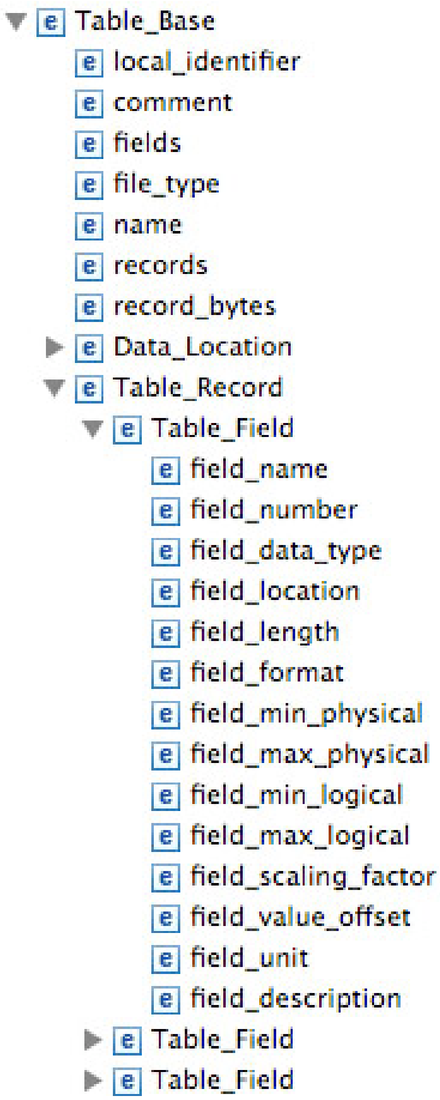


Figure 5.1: Table_Base

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 Unicode Transformation Format (UTF)-8 values.

(include picture here)

5.2 Array Base

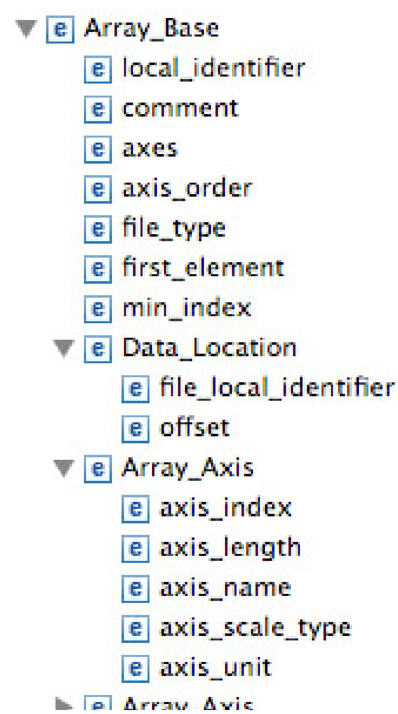
The second of the four basic structures consists solely of homogeneous pixels in fixed-length arrays in any number of dimensions, although two- and three-dimensional data are the most common. These arrays are described using a subclass of the Array_Base class, which has two associated classes (besides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

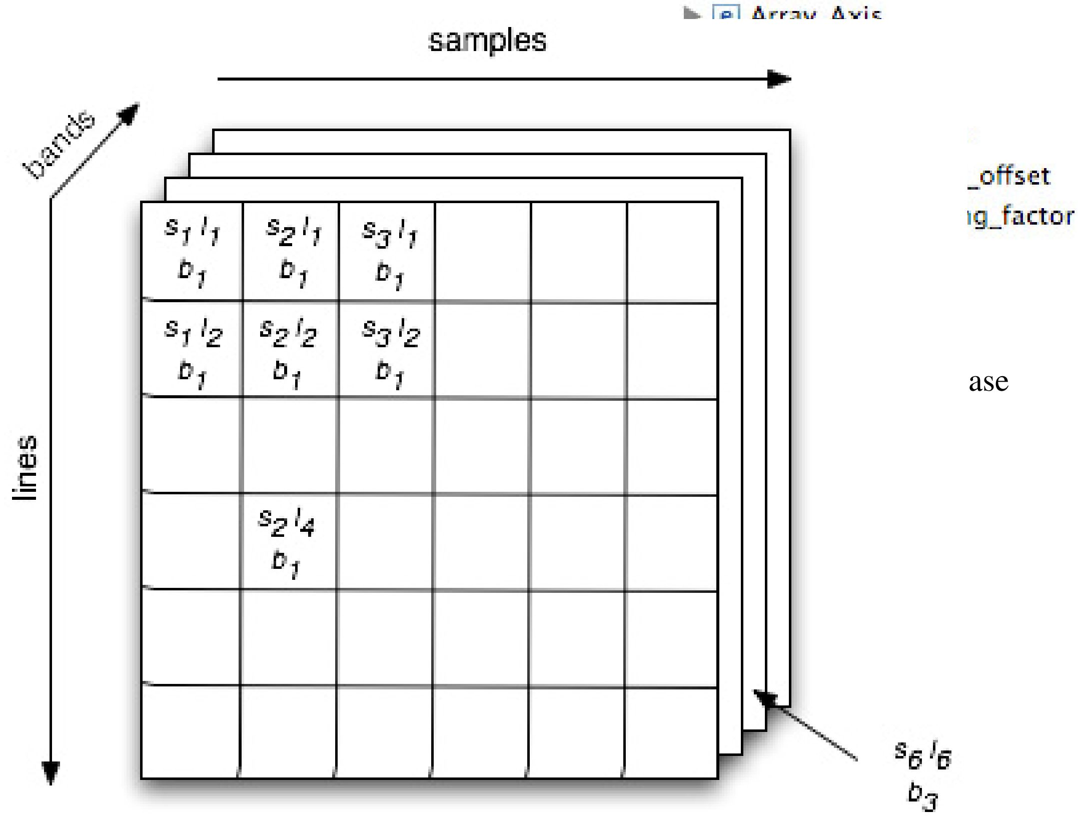
true value = (stored value x scaling factor) + offset value

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:

$s_1 l_1 b_1, s_2 l_1 b_1, s_3 l_1 b_1 ... s_1 l_2 b_1, s_2 l_2 b_1, s_3 l_2 b_1 ... s_1 l_1 b_2, s_2 l_1 b_2, s_3 l_1 b_2 ...$

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)

5.3 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”.

5.4 Encoded Byte Stream

The encoded byte stream 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 PDF.

In order to interpret an encoded byte stream, 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 6

Data Storage Types

6.1 Character Data Types

Character data types are used to classify attributes in class descriptions (i.e., in product labels) and to describe the data formats of fields in character tables.

6.1.1 Boolean Types

Value	Description	Values
ASCII_Boolean_TF	True / False indicator	`T` or `F`[1]. Limit 1 character.

[1] Are other boolean values permitted? – EDR

6.1.2 Date and Time Types

Value	Description	Values
ASCII_Date[1]	A date string in either Day Of Year (DOY) or Year Month Day (YMD) format.	Date value in either of the following two forms: `yyyy-doy` `yyyy-mm-dd`.
ASCII_Date_DOY	A date string in DOY format.	Date value of the form: `yyyy-doy`.

[1] Note: this data_type is not currently in the information model; it is being considered for adoption. – EDR

[2] Has been deleted from model – why? – EDR

Value	Description	Values
ASCII_Date_YMD	A date string in YMD format.	Date value of the form: `yyyy-mm-dd`.
ASCII_Date_Time[2]	A date/time string where the date may be in either DOY or YMD format.	Date/time value in either of the following two forms: `yyyy-doyThh:mm:ss[.fff]` `yyyy-mm-ddThh:mm:ss[.fff]`. If Coordinated Universal Time (UTC), `Z` should be appended.
ASCII_Date_Time_DOY	A date/time string where the date is in DOY format.	Date/time value of the form: `yyyy-doyThh:mm:ss[.fff]`. If UTC, `Z` should be appended.
ASCII_Date_Time_YMD	A date/time string where the date is in YMD format.	Date/time value of the form: `yyyy-mm-ddThh:mm:ss[.fff]`. If UTC, `Z` should be appended.
ASCII_Time	An ASCII time string. May be used for local times.	ASCII string of the form: `hh:mm:ss[.fff]`. If UTC, `Z` should be appended.

[1] Note: this data_type is not currently in the information model; it is being considered for adoption. – EDR

[2] Has been deleted from model – why? – EDR

6.1.3 Numeric Types

Value	Description	Values
ASCII_Integer	An ASCII character representation of a decimal integer.	ASCII string consisting of the characters 0 through 9, plus the negative sign "-".
ASCII_Mask[1]	An ASCII representation, in binary, octal, or hexadecimal notation, of a bit mask.	Limit 255 characters.
ASCII_MD5_Checksum	A 128-bit hash value calculated using the MD5 algorithm.	An ASCII string consisting of the characters 0 through 9 and A through F. Must be exactly 32 characters in length.

[1] As currently defined, there appears to be no difference between ASCII_Mask and ASCII_Numeric_Base2 / ASCII_Numeric_Base16. Should it be redefined to become the catch-all for all ASCII representations of binary, octal, and hexadecimal numbers? – EDR

[2] Note that version 0.3.0.0.e of the Information Model incorrectly set the maximum value of this data_type to 2147483647; it has since been corrected. – EDR

[3] Note: this data_type is not currently in the information model; it is being considered for adoption. – EDR

Value	Description	Values
ASCII_NonNegative_Integer	An ASCII character representation of a decimal integer greater than or equal to zero.	ASCII number in the range 0 to 4294967295.[2]
ASCII_Numeric_Base2	An ASCII representation of a number in binary format.	An ASCII string consisting of the characters 0 and 1.
ASCII_Numeric_Base8[3]	An ASCII representation of a number in octal format.	An ASCII string consisting of the characters 0 through 7.
ASCII_Numeric_Base16	An ASCII representation of a number in hexadecimal format.	An ASCII string consisting of the characters 0 through 9 and A through F.
ASCII_Real	An ASCII character representation of a real number.	ASCII number in the range of negative infinity to positive infinity.

[1]As currently defined, there appears to be no difference between ASCII_Mask and ASCII_Numeric_Base2 / ASCII_Numeric_Base16. Should it be redefined to become the catch-all for all ASCII representations of binary, octal, and hexadecimal numbers? – EDR

[2]Note that version 0.3.0.0.e of the Information Model incorrectly set the maximum value of this data_type to 2147483647; it has since been corrected. – EDR

[3]Note: this data_type is not currently in the information model; it is being considered for adoption. – EDR

6.1.4 String Types

Value	Description	Values
ASCII_AnyURI	A URI or its subclasses URN and URL (See section 7.2.4 for details.)	An ASCII string in URI format. Limit 255 characters.
ASCII_DOI	A Digital Object Identifier (DOI). (See section 7.2.3 for details.)	ASCII string of the form: `nn.nnnn/nnn` Limit 255 characters.
ASCII_Identifier	A PDS identifier. (See section 7.1 for details.)	ASCII string beginning with an alphabetic character. Limit 100 characters.
ASCII_LID	A PDS logical identifier. (See section 7.1.1 for details.)	An ASCII string of the form `URN:NASA:PDS:xxxx` Limit 255 characters.

[1]Please note that a UTF-8 character string close to 255 *characters* in length may exceed 255 *bytes* in length.

Value	Description	Values
ASCII_LIDVID	A PDS versioned identifier (logical identifier plus version id). (See section 7.1.1 for details.)	An ASCII string of the form `URN:NASA:PDS:xxxx::M.n`. Limit 255 characters.
ASCII_VID	A PDS version id. (See section 7.2.1 for details.)	An ASCII string of the form `M.m`, where `M` and `m` are both integers.
ASCII_Directory_Path_Name	A system directory path in UNIX format.	ASCII string of the form: `dir1/dir2/` Limit 255 characters.
ASCII_File_Name	A system file name. (See section 7.4 for details.)	ASCII string of the form: `file_name.file_ext` Limit 255 characters.
ASCII_File_Specification_Name	A system file including directory path, file name, and file extension in UNIX format.	ASCII string of the form: `dir1/dir2/file_name.ext` Limit 255 characters.
ASCII_Short_String_Collapsed	An ASCII-encoded text string of limited length with whitespace collapsed. (I.e. multiple spaces, new lines, tabs, and carriage returns are not significant.)	An ASCII string. Limit 255 characters.
ASCII_Short_String_Preserved	An ASCII-encoded text string of limited length with whitespace preserved. (I.e. multiple spaces, new lines, tabs, and carriage returns are significant.)	An ASCII string. Limit 255 characters.
ASCII_Text_Preserved	An ASCII-encoded text string of unlimited length with whitespace preserved.	An ASCII string.
UTF8_Short_String_Collapsed	A UTF-8 encoded text string of limited length with whitespace collapsed.	A UTF-8 string. Limit 255 characters.[1]
UTF8_Short_String_Preserved	A UTF8-encoded text string of limited length with whitespace preserved.	A UTF-8 string. Limit 255 characters.[1]

[1]Please note that a UTF-8 character string close to 255 *characters* in length may exceed 255 *bytes* in length.

Value	Description	Values
UTF8_Text_Preserved	A UTF8-encoded text string of unlimited length with whitespace preserved.	A UTF-8 string.

[1]Please note that a UTF-8 character string close to 255 *characters* in length may exceed 255 *bytes* in length.

6.2 Binary Data Types

Binary data types are used to describe data formats of fields in binary tables and array element formats in arrays.

6.2.1 Integers

6.2.1.1 Signed LSB Integers

This section describes signed integers stored in Least Significant Byte (LSB) first (also known as *little-endian*) 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}

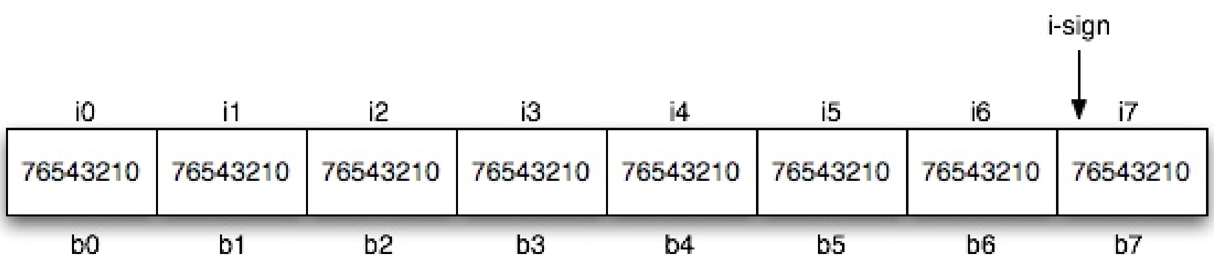
4-byte integers:

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}

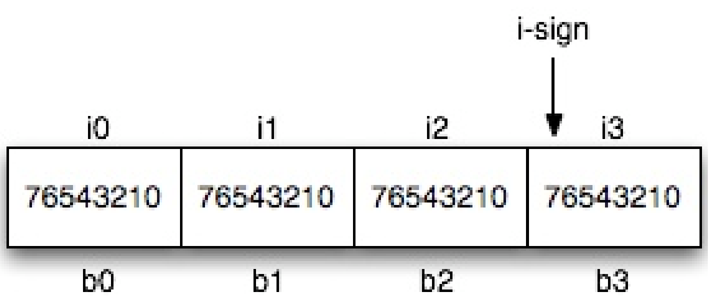
2-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7
In i1, bits 0-6 represent 2^8 through 2^{14}

All negative values are represented in two's complement.

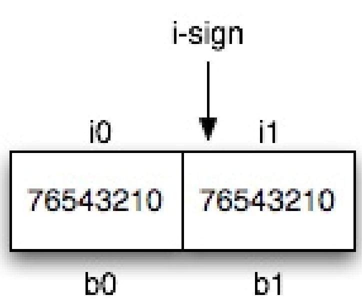
SignedLSB8



SignedLSB4



SignedLSB2



6.2.1.2 Unsigned LSB Integers

This section describes unsigned integers stored in 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)

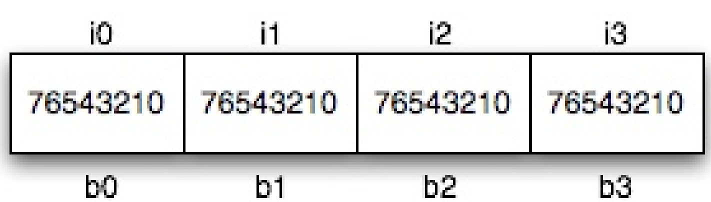
i0 – i3 Arrangement of bytes in the integer, from lowest order to highest order. The bits within each byte are interpreted from right to left (e.g., lowest value = bit 0, highest value = bit 7), in the following way:

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}

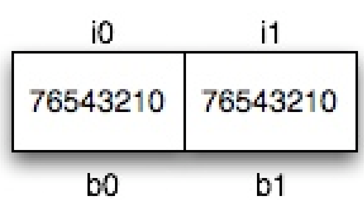
2-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7
In i1, bits 0-7 represent 2^8 through 2^{15}

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

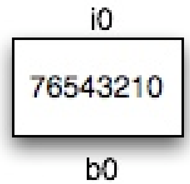
UnsignedLSB4



UnsignedLSB2



Unsigned Byte



6.2.1.3 Signed MSB Integers

This section describes the signed integers stored in Most Significant Byte (MSB) first (also known as *big-endian*) 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

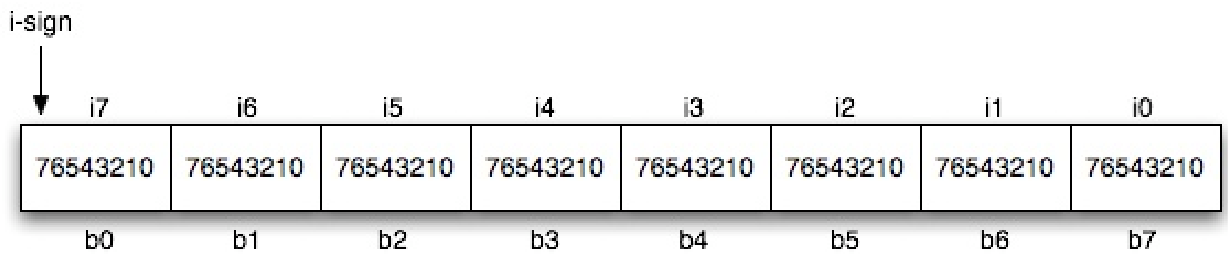
i0 – i7 Arrangement of bytes in the integer, from lowest order to highest order. The bits within each byte are interpreted from right to left (e.g., lowest value = bit 0, highest value = bit 7) in the following way:

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}

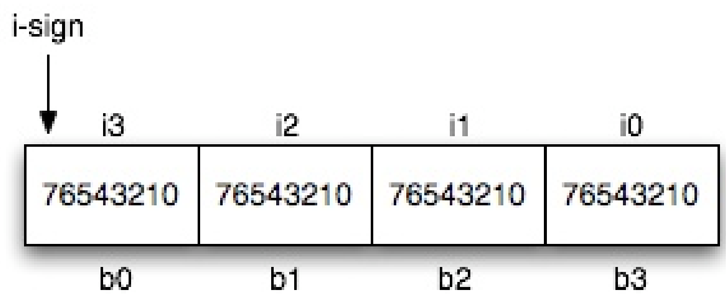
4-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7
In i1, bits 0-7 represent 2^8 through 2^{15}
In i2, bits 0-7 represent 2^{16} through 2^{23}
In i3, bits 0-6 represent 2^{24} through 2^{30}

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}

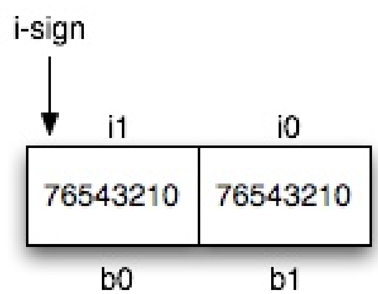
SignedMSB8



SignedMSB4



SignedMSB2



6.2.1.4 Unsigned MSB Integers

This section describes unsigned integers stored in MSB format. In this section the following definitions apply:

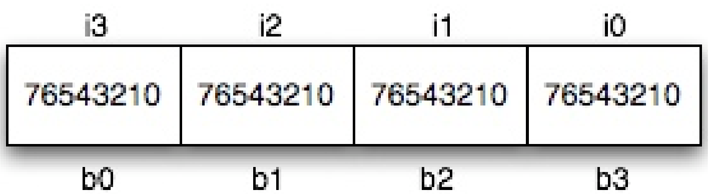
b0 – b3 Arrangement of bytes as they appear when read from a file (e.g., read b0 first, then b1, b2, and b3) *i0 – i3* Arrangement of bytes in the integer, from lowest order to highest order. The bits within each byte are interpreted from right to left (e.g., lowest value = bit 0, highest value = bit 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}

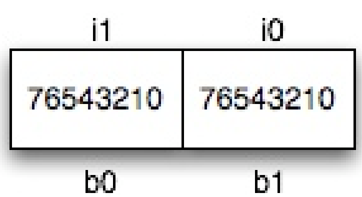
2-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7
In i1, bits 0-7 represent 2^8 through 2^{15}

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

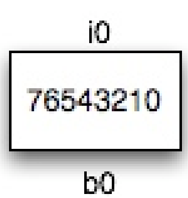
UnsignedMSB4



UnsignedMSB2



UnsignedByte



6.2.2 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 $1/2^{52}$

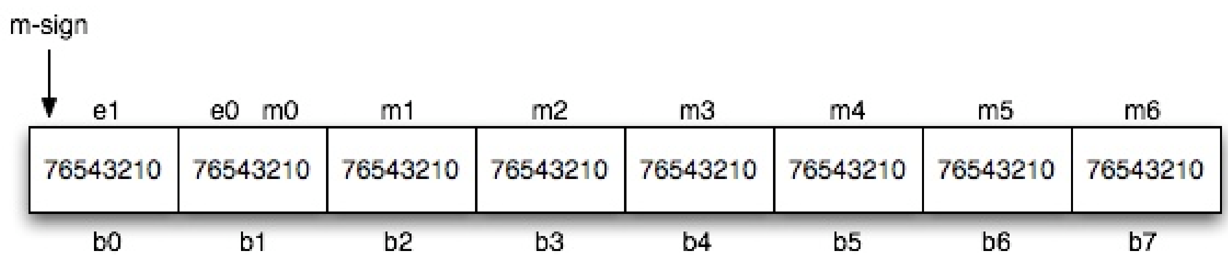
4-bytes (single precision):
In m0, bits 6-0 represent $1/2^{1}$ through $1/2^{7}$
In m1, bits 7-0 represent $1/2^{8}$ through $1/2^{15}$
In m2, bits 7-0 represent $1/2^{16}$ through $1/2^{23}$

The following representations all follow this format:

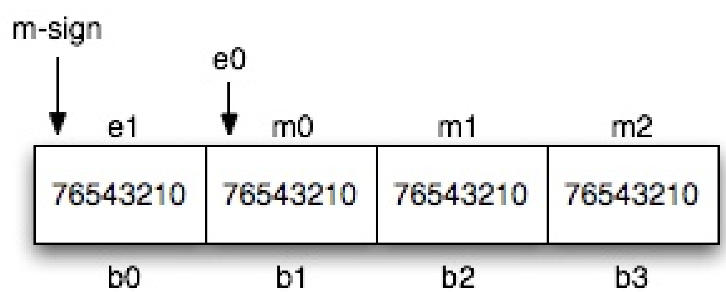
$$1.mantissa \text{ x } 2^{(exponent-bias)}$$

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



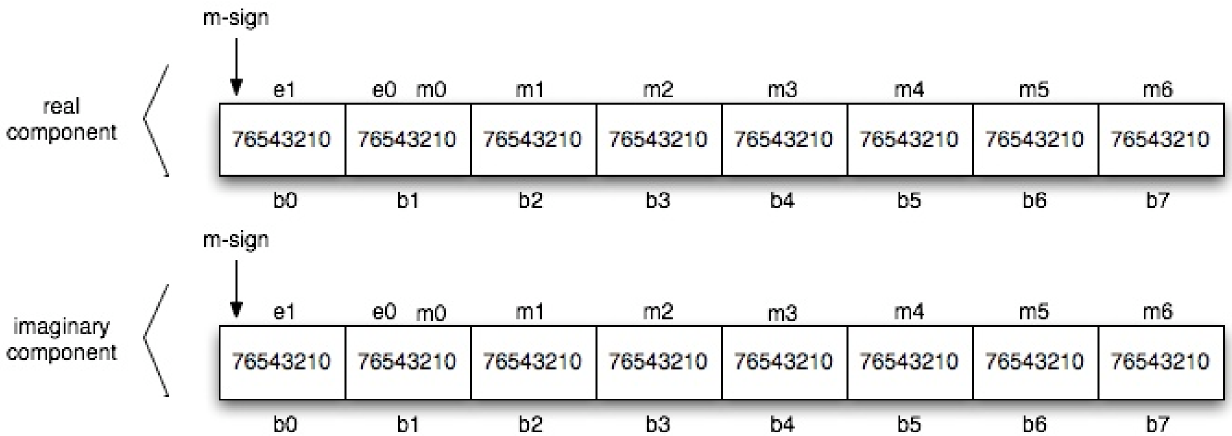
IEEE754Single



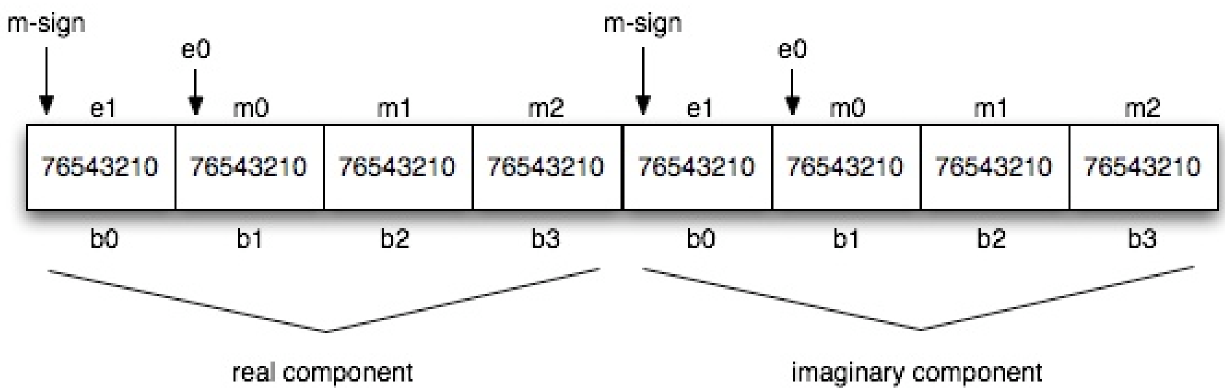
6.2.3 Complex

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.

ComplexB16



ComplexB8



6.2.4 Boolean

This binary data type is listed in the Information Model Specification, but is, as yet, insufficiently defined there to describe here. – EDR

6.3 Multi-Valued Data Types

Where do these definitions belong? This section is basically a placeholder until I come up with a better idea. – EDR.

6.3.1 Vectors

6.3.2 Quaternions

A quaternion is a four-component representation of a rotation matrix. This particular definition is focused on the PDS use of quaternions; one should refer to other sources for a more complete discourse on quaternion math.

A quaternion may be used to specify the rotation of one Cartesian reference frame–sometimes referred to as the base frame or the 'From' frame–into coincidence with a second Cartesian reference frame–sometimes referred to as the target reference frame or the 'To' frame. Unlike an Euler rotation where three sequential rotations about primary axes are used, a quaternion rotation is a single action, specified by a Cartesian vector used as the positive axis of the rotation (right hand rule) and the magnitude (an angle) of rotation about that axis.

The quaternion may be thought of as defining the instantaneous orientation–sometimes called 'pointing'–of a structure such as an instrument, antenna, solar array or spacecraft bus, given relative to a specified reference frame (the base frame), at an epoch of interest.

Perhaps of more use is the concept that a quaternion may be used to rotate an arbitrary Cartesian 3-vector defined in one reference frame (e.g. an instrument's reference frame) to an equivalent vector defined in another reference frame (e.g. the frame tied to a spacecraft or the J2000 inertial reference frame).

A quaternion has four components. One of the components is a scalar, a function of the angle of rotation (cosine of half the rotation angle), while the remaining three components are used to specify a vector, given in the base reference frame, about which the rotation will be made. In the PDS context a quaternion has a magnitude of one, and so may be treated as a unit quaternion.

In many cases a time tag (epoch) must be associated with the quaternion because the orientation varies over time. A time tag is not needed if the 'To' and 'From' frames have a fixed offset.

The QUATERNION_DESC element is always to be paired with the QUATERNION element, and will contain a complete description of the formation and rotational sense of the quaternion specified with the QUATERNION keyword, and the structure (organization of the four components) of the quaternion.

In the lingo of the NASA 'SPICE' ancillary information system a rotation matrix is synonymous with a C-matrix–that which may be obtained from a C-kernel. The SPICE Toolkit provides an assortment of routines that deal with quaternions. The SPICE system also provides information about specification of reference frames and time tags suitable for use with quaternions in the SPICE context. The NAIF Node of the PDS can provide additional documentation on quaternions in a spacecraft ancillary data context ('Rotations Required Reading' and 'SPICE Quaternion White Paper').

Chapter 7

Nomenclature Rules

Make sure to specify permitted character sets!

7.1 Identifiers

7.1.1 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

7.1.2 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)

7.1.3 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 =

7.1.4 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) (changed)

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

7.2 Other Identifying Information

7.2.1 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.

7.2.2 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.)

7.2.3 Digital Object Identifiers

TBD – Probably need to add a description of these. – EDR

7.2.4 Uniform Resource Identifiers

TBD – Probably need to add a description of these. – EDR

7.2.5 Names

TBD - Is this still needed?

Rules:

•

Recommendations:

•

7.2.6 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?

7.3 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

7.4 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: "Naming Files, Paths, and Namespaces" from MSDN and Wikipedia topic "Filename".)

7.5 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

7.6 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?

7.7 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:

-

Part II

Data Content Standards

Chapter 8

Calibration Standards

TBD - This section is targeted for completion in the second half of calendar year 2011.

Chapter 9

Cartographic Standards

9.1 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 National Aeronautics and Space Administration (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 early 2011, 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.

9.1.1 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; Archinal et al., 2011). 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 (LGCWG) and the Mars Geodesy and Cartography Working Group (MGCWG) 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; Archinal et al., 2008b; Duxbury et al. 2002). These Working Groups have made additional recommendations regarding coordinate systems (generally with additional detail) beyond those of the WGCCRE.

9.2 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 body's 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 α and declination δ) 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 α and declination δ, 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 Barycentric Dynamical Time (TDB) (e.g., Seidelmann et al., 2007). This corresponds to 2000 January 1, 1200 hours Terrestrial Time (TT) or the Julian Date 2451545.0 (NAO, USNO and HMNAO, 1983). This also corresponds to 2000 January 1, 11:58:55.816 Coordinated Universal Time (UTC) (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 chapter 12 of this document 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, 2002). 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 Earth's 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 HMNAO, 1983; also see http://nedwww.ipac.caltech.edu/forms/calculator.html).

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.

9.3 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 Sun's rotation or counterclockwise when viewed from above the north pole) or retrograde (opposite to the direction of the Sun's 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.

9.4 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.

9.4.1 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 body's 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 the 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 sets of data archived with the PDS, as defined by the Planetary Science Data Dictionary (PDS, 2002) (fix reference - EDR) and is allowed by the WGCCRE. However, for the Moon, the NASA LGCWG and Lunar Reconnaissance Orbiter (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 Jet Propulsion Laboratory (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).

9.4.2 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.

9.4.3 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.

9.4.4 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 (PPI) 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.

9.5 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). (WGCCRE #10 is (Archinal et al., 2011), which has only been published online to date, so I'm not sure where the "page 168" comes from. Need to clarify this with Lisa G. - EDR) 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.

9.6 PDS Keywords for Cartographic Coordinates

NOTE: Work is currently ongoing on defining the cartographic classes to be used in PDS4; drafts of these are expected to be available roughly mid-year, 2011. When they become available, this section will be updated. - EDR

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) (update ref - EDR) 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.

9.7 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 2^n pixels per degree of latitude, where a pixel is treated as a finite area and n is an integer. A spatial resolution of 2^n pixels per degree allows simple co-registration 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 2^n 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 10^m meters, where m is an integer chosen to preserve all the resolution inherent in the data.

Need to determine desired format for references in Bibliography and design or acquire a .bst file to accomplish that. - EDR

Need to determine how to output dois, if available. (Already done for some doc. types, but not others?) - EDR

How to include statement at beginning of section 2.8 in StdRefv3.8 about WGCCRE Report 7 not being issued? This needs to be inserted after the ”Bibliography” statement, but before the list of references. - EDR

I’ve dropped the ”chair”, ”vice-chair” and ”consultant” parenthetical statements from the WGCCRE report references; how important is it to have them? (Can be done, but messes up the citations. Probably fixable with research.) - EDR

Need to fix Archinaletal2008a reference; currently missing name of the ”event”. Also need to determine how to include URLs in references. - EDR

Need to fix Archinaletal2008b reference to include abstract number and URL. - EDR

For all WGCCRE reports, need to determine how to append report number. - EDR

Need to fix Duxburyetal2002a reference to include meeting name and URL. - EDR

Need to fix GreeleyBatson1990a reference to include number of book pages. - EDR

Need to include URL for IAU2002a. - EDR

I’ve added information for KovalevskyMueller1981a; is this okay? - EDR

LGCWG2008a - need to add URL; also, how to indicate draft (or find published version)? - EDR

LRO2008a - need to include verion. - EDR

PCWG1993a only seems to be available as one used copy on Amazon for $100; is this still publicly available? - EDR

Need to add PDS4 PSDD reference. - EDR

In Seidelmanetal1992a, BibTex doesn’t want to use both author and editor fields for Seidelmann; need to fix? - EDR

Chapter 10

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.

10.1 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.

10.2 Nodes

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

10.3 Observing Systems

10.3.1 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.

10.3.2 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.

10.4 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.

10.5 Publications / References

TBD

10.6 Resources

TBD

10.7 Software

TBD

10.8 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 11

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.

11.1 Data Set Description

Data Set Overview A high level description of the characteristics and properties of a data set

Parameters The primary parameters (measured or derived quantities) included in the data set, with units and sampling intervals

Processing The overall processing used to produce the data set, including a description of the input data (and source), processing methods or software, and primary parameters or assumptions used to produce the data set

Data Detailed description of each data type identified in the Data Set Overview, (e.g., image data, table data, etc.)

Ancillary Data Description of the ancillary information needed to interpret the data set. The ancillary information may or may not be provided along with the data set. If not, this description should include sources or references for locating the ancillary data.

Coordinate System Description of the coordinate system(s) or frame(s) of reference to be used for proper interpretation of the data set

Software Description of software relevant to the data, including software supplied with the data set as well as external software or systems that may be accessed independently to assist in visualization or analysis of the data

11.2 Instrument Description

11.3 Instrument Host Description

11.4 Investigation / Mission Description

Chapter 12

Time Standards

TBD from ISO 8601

The following is a place holder and needs to be updated.

PDS has adopted a subset of the International Standards Organization Standard (ISO/DIS) 8601 standard entitled Data Element and Interchange Formats - Representations of Dates and Times, and applies the standard across all disciplines in order to give the system generality.

It is important to note that the ISO/DIS 8601 standard covers only ASCII representations of dates and times.

Note that the "SPICE" system, generally used by planetary missions for computing observation geometry, uses additional time systems and time representations.

12.1 Date/Times

In the PDS there are two recognized date/time formats: Now incorrect!

CCYY-MM-DDTHH:MM:SS.sss (preferred format)
CCYY-DDDTHH:MM:SS.sss

Each format represents a concatenation of the conventional date and time expressions with the two parts separated by the letter T:

CC - century (00-99)
YY - year (00-99)
MM - month (01-12)
DD - day of month (01-31)
DDD - day of year (001-366)
T - date/time separator
HH - hour (00-23)
MM - minute (00-59)
SS - second (00-59)
sss - fractions of second (000-999)

Note: See Section 7.4 Midnight and Leap Seconds for special cases involving the indication of midnight and leap seconds.

The preferred date/time format is: CCYY-MM-DDTHH:MM:SS.sss.

Date/Time Precision

The above date/time formats may be truncated on the right to match the precision of the date/time value in any of the following forms:

1998
1998-12
1998-12-01
1998-12-01T23
1998-12-01T23:59
1998-12-01T23:59:58
1998-12-01T23:59:58.1
1998-12-01T23:59:58.12

12.2 Dates

Dates should be expressed in the conventional ISO/DIS 8601 format. On those rare occasions when dates cannot be expressed in the conventional format, a native format may be used.

12.2.1 Conventional Dates

Conventional dates are represented in ISO/DIS 8601 format as either year (including century), month, day-of-month (CCYY-MM-DD), or as year, day-of-year (CCYY-DDD). The hyphen character (-) is used as the field separator in this format. The year, month, day-of month format is the preferred format for use in PDS labels and catalog files and is referred to as PDS standard date format, but either format is acceptable.

12.2.2 Native Dates

Dates in any format other than the ISO/DIS 8601 format described above are considered to be in a format native to the specific data set, thus native dates. Native date formats are specified by the data preparer in conjunction with the PDS data engineer. Mission-elapsed days and time-to-encounter are both examples of native dates.

12.3 Times

The PDS allows times to be expressed in conventional and native (alternate) formats.

12.3.1 Conventional Times

Conventional times are represented as hours, minutes and seconds according to the ISO/DIS 8601 time format standard: HH:MM:SS[.sss]. Note that the hours, minutes, and integral seconds fields must contain two digits. The colon (':') is used as a field separator. Fractional seconds consisting of a decimal point (the European-style comma may not be used) and up to three digits (thousandths of a second) may be included if appropriate.

Coordinated Universal Time (UTC) is the PDS time standard and must be formatted in the previously described ISO/DIS 8601 standard format. The letter ”Z”, indicating the civil time zone at Greenwich (i.e., GMT), should never be appended to a UTC time. The relationship between UTC and GMT has varied historically and with observer context. Note that in PDS data sets created under earlier versions of the Standards, an appended Z is taken as indicating UTC.

The start_time and stop_time attributes required in data product labels are in UTC. For data collected by spacecraft-mounted instruments, the date/ time must be a time that corresponds to spacecraft event time. For data collected by instruments not located on a spacecraft, this time shall be an earth-based event time.

Adoption of UTC (rather than spacecraft-clock-count, for example) as the standard facilitates comparison of data from a particular spacecraft or ground-based facility with data from other sources.

12.3.2 Native Times

Times in any format other than the ISO/DIS 8601 format described above are considered to be in a format native to the data set, and thus native times. Data preparers should consult a PDS data engineer for assistance in selecting appropriate PDS attributes for expressing native times.

Examples of quantities that may be expressed in native time formats are included below.

12.3.2.1 Spacecraft Clock Count (sclk)

There is one native time of particular interest, however, which has specific keywords associated with it. The spacecraft clock reading (that is, the count) often provides the essential timing information for a space-based observation. Therefore, the attributes spacecraft_clock_start_count and spacecraft_clock_stop_count are required in labels describing space-based data. This value is formatted as a string to preserve precision.

12.3.2.2 Ephemeris Time

Need to talk to Chuck and/or Boris to determine if they want to say anything here.

12.3.2.3 Relative Time

12.3.2.4 Local (Solar) Time

12.3.2.4.1 Local True Solar Time

The local_true_solar_time attribute describes the local true solar time (LTST). It is one of two types of solar time used to express the time of day at a point on the surface of a planetary body. LTST is measured relative to the true position of the Sun as seen from a point on the planet's surface. The coordinate system used to define LTST has its origin at the center of the planet. Its Z-axis is the north pole vector (or spin axis) of the planet. The X-axis is chosen to point in the direction of the vernal equinox of the planet's orbit. (The vernal or autumnal equinox vectors are found by searching the planetary ephemeris for those times when the vector from the planet's center to the Sun is perpendicular to the planet's north pole vector. The vernal equinox is the time when the Sun appears to rise above the planet's equator.) Positions of points in this frame can be expressed as a radius and planetocentric 'right ascension' and 'declination' angles. The planetocentric right ascension angle, or RA, is measured positive eastward in the equatorial plane from the vernal equinox vector to the intersection of the meridian containing the point with the equator. Similarly, the planetocentric declination is the angle between the equatorial plane and the vector to the point. LTST is a function of the difference between the RAs of the vectors to the Sun and to the point on the planet's surface. Specifically,

LTST = (a(P) - a(TS)) * (24 / 360) + 12

where,

LTST = the local true solar time in true solar hours
a(P) = RA of the point on the planet's surface in deg
a(TS) = RA of the true sun in deg

The conversion factor of 24/360 is applied to transform the angular measure in decimal degrees into hours-minutes-seconds of arc. This standard representation divides 360 degrees into 24 hours, each hour into 60 minutes, and each minute into 60 seconds of arc. The hours, minutes, and seconds of arc are called 'true solar' hours, minutes, and seconds when used to measure LTST. The constant offset of 12 hours is added to the difference in RAs to place local noon (12:00:00 in hours, minutes, seconds) at the point where the Sun is directly overhead; at this time, the RA of the true sun is the same as that of the surface point so that a(P) - a(TS) = 0. The use of 'true solar' time units can be extended to define a true solar day as 24 true solar hours. Due to the eccentricity of planetary orbits and the inclination of orbital planes to equatorial planes (obliquity), the Sun does not move at a uniform rate over the course of a planetary year. Consequently, the number of SI seconds in a true solar day, hour, minute or second is not constant. (Definition adapted from Vaughan (1995).)

12.3.2.4.2 Local Mean Solar Time

The desire to work with solar days, hours, minutes, and seconds of uniform length led to the concept of the fictitious mean Sun (FMS). The FMS is defined as a point that moves on the celestial equator of a planetary body at a constant rate that represents the average mean motion of the Sun over a planetary year. local mean solar time (LMST), is defined, by analogy with LTST, as the difference between the planetocentric right ascensions of a point on the surface and of the FMS.

The difference between LTST and LMST varies over time. The length of a mean solar day is constant and can be computed from the mean motion of the FMS and the rotation rate of a planet. The mean solar day is also called a 'sol'. Mean solar hours, minutes, and seconds are defined in the same way as the true solar units.

The acceptable range of values for local_mean_solar_time is '00:00:00.000' to '23:59:59.999'. (Definition adapted from Vaughan (1995).)

12.4 Midnight and Leap Seconds

The ISO/DIS 8601 standard for representation of midnight and leap seconds are also used in PDS time fields.

12.4.1 Midnight

Midnight may be indicated in one of two ways: as 00:00:00 or 24:00:00. The usual precision modifications apply as well i.e. 24:00 is also recognized as midnight.

The 00:00:00 notation is used to indicate midnight at the beginning of a date. 24:00:00 is used to indicate midnight at the end of a date. So, for example, the following two date/time strings refer to precisely the same moment:

2007-04-07T24:00:00 = 2007-04-08T00:00:00

When the hours field has the value 24, any and all subsequent time fields must be zero.

12.4.2 Leap Seconds

Leap seconds may be positive or negative, but in either case are always applied at the end of the day in question. A positive leap second is indicated with a time value of 23:59:60. A negative leap second is indicated by the omission of the time 23:59:59. That is, on the day of a negative leap second, the sequence leading through midnight is:

23:59:57
23:59:58
00:00:00
00:00:01

And on the day of a positive leap second, the sequence through midnight is:

23:59:58
23:59:59
23:59:60
00:00:00
00:00:01

Note that the only time when the seconds value of a time string may contain the value 60 is when this represents a positive leap second.

Chapter 13

Units of Measurement

The uniform use of units of measure facilitates broad catalog searches across archive systems.The PDS standard system for units, where applicable, is the SI. The default units for data elements in the PSDD are determined as each element is defined and added to the dictionary. Specific unit definitions are also included in the PSDD.

The PDS allows exceptions to the SI unit requirement when common usage conflicts with the SI standard (e.g., angles which are measured in degrees rather than radians).

Both singular and plural unit names, as well as unit symbols, are allowed. The double asterisk (**) is used, rather than the caret (ˆ), to indicate exponentiation. When the units associated with a value of a PDS element are not the same as the default units specified in the PSDD (or when explicit units are preferred), a unit expression is used with the value. These unit expressions are enclosed in angular brackets (¡ ¿) and follow the value to which they apply.

Update to indicate implementation using attributes.

Examples Needs to be udpated

EXPOSURE_DURATION = 10 seconds
DECLINATION = -14.2756 degrees
MASS = 123 kg
MASS_DENSITY = 123 g/cm^3
MAP_RESOLUTION = 123 pixel/degree
MAP_SCALE = 123 km/pixel

Note that in the above example, MASS_DENSITY is not expressed in the SI default unit of measurement for density (kg/m^3).

PDS recommends (in order of preference) that measurements be expressed using the default SI units of measurements, as defined in the following paragraphs. If it is not desirable to use the default SI unit of measurement, then the unit of measurement should be expressed using the SI nomenclature defined in the following paragraphs. If a unit of measurement is not defined by the SI standard, then a unit of measurement can be derived (e.g., pixels per degree, kilometers per pixel, etc.).

13.1 SI Units

The following summary of SI unit information is extracted from The International System of Units.

Base units As the system is currently used, there are seven fundamental SI units, termed base units:

QUANTITY	NAME OF UNIT	SYMBOL
length	meter	m
mass	kilogram	kg
time	second	s
electric current	ampere	A
thermodynamic temperature	kelvin	K
amount of substance	mole	mol
luminous intensity	candela	cd

SI units are all written in mixed case; symbols are also mixed case except for those derived from proper names. No periods are used in any of the symbols in the international system.

Derived units In addition to the base units of the system, a host of derived units, which stem from the base units, are also employed. One class of these is formed by adding a prefix, representing a power of ten, to the base unit. For example, a kilometer is equal to 1,000 meters, and a millisecond is .001 (that is, 1/1,000) second. The prefixes in current use are as follows:

SI PREFIXES

Factor	Prefix	Symbol	Factor	Prefix	Symbol
10^{18}	exa	E	10^{-1}	deci	d
10^{15}	peta	P	10^{-2}	centi	c
10^{12}	tera	T	10^{-3}	milli	m
10^{9}	giga	G	10^{-6}	micro	
10^{6}	mega	M	10^{-9}	nano	n

10^3	kilo	k	10^{-12}	pico	p
10^2	hecto	h	10^{-15}	femto	f
10^1	deka	da	10^{-18}	atto	a

Note that the kilogram (rather than the gram) was selected as the base unit for mass for historical reasons. Notwithstanding, the gram is the basis for creating mass units by addition of prefixes.

Another class of derived units consists of powers of base units and of base units in algebraic relationships. Some of the more familiar of these are the following:

QUANTITY	**NAME OF UNIT**	**SYMBOL**
area	square meter	m^2
volume	cubic meter	m^3
density	kilogram per cubic meter	kg/m^3
velocity	meter per second	m/s
angular velocity	radian per second	rad/s
acceleration	meter per second squared	m/s^2
angular acceleration	radian per second squared	rad/s^2
kinematic viscosity	square meter per second	m^2/s
dynamic viscosity	newton-second per square meter	$N * s/m^2$
luminance	candela per square meter	cd/m^2
wave number	1 per meter	m^{-1}
activity (of a radioactive source)	1 per second	s^{-1}

Many derived SI units have names of their own:

QUANTITY	**NAME OF UNIT**	**SYMBOL**	**EQUIVALENT**
frequency	hertz	Hz	s^-1
force	newton	N	$kg * m/s^2$
pressure (mechanical stress)	pascal	Pa	N/m^2
work, energy, quantity of heat	joule	J	$N * m$
power	watt	W	J/s
quantity of electricity potential difference	coulomb	C	$A * s$
electromotive force	volt	V	W/A
electrical resistance	ohm		V/A
capacitance	farad	F	$A * s/V$
magnetic flux	weber	Wb	$V * s$
inductance	henry	H	$V * s/A$
magnetic flux density	tesla	T	Wb/m^2

luminous flux	lumen	lm	$cd * sr$
illuminance	lux	lx	lm/m^2

Supplementary units are as follows:

QUANTITY	**NAME OF UNIT**	**SYMBOL**
plane angle	radian	rad
solid angle	steradian	sr

Use of figures with SI units In the international system it is considered preferable to use only numbers between 0.1 and 1,000 in expressing the quantity associated with any SI unit. Thus the quantity 12,000 meters is expressed as 12 km, not 12,000 m. So too, 0.003 cubic centimeters is preferably written 3 mm3, not 0.003 cm3.

Appendices

Appendix A

Digital Object Classes

The classes shown in this appendix describe the digital objects used in basic products such as images, tables, SPICE kernels, documents, and native headers. All of these classes (with the exception of the Document_Format class; see chapter 11) appear in a File_Area of a PDS label.

In the tables shown below, the "attributes" are the meta-data words used to describe each digital object. The "associations" provide a relationship between the current class and a related class. When an association is present, it indicates that the specified class either may or must be present within the current class, in the PDS label.

The "Definition" column provides a brief description of the attribute, or specifies the type of association.

The "Card." (cardinality) column indicates the number of instances of a particular attribute or associated class that may occur in the current class. Where a single digit appears, the class must include precisely that number of attributes (or associated classes). Where a range of numbers is specified (using the notation m..n), then any number of attributes in that range may be included. A minimum of "0" means that the attribute or associated class is optional; a maximum of "*" means that there is no limit on the number of attributes or associated classes that may be included.

In those cases where an attribute has either a single allowed value, or a list of permissible values, those values will be listed in the "Values" column. In cases where there is no specified value, the data type will be listed instead. For associations, the name (or names) of the class (or classes) that fit the specified association type are listed. These class names are linked to the sections in this appendix where they are described. (Not yet implemented.)

Need to add Tagged_Digital_Child classes.
Need to complete numerous examples.

A.1 Array_2D_Image

The Array 2D Image class is an extension of array_base and defines a two dimensional image.

Attributes & Associations

Attribute	**Definition**	**Card.**	**Values**
encoding_type	The encoding_type attribute provides the storage format (binary or character).	1	BINARY
local_identifier	The local_identifier attribute provides a character string which uniquely identifies the containing object within the label.	1	[string]
axes	The axes attribute provides a count of the axes.	1	2
offset	The offset attribute provides the displacement of the object starting position from the beginning of the parent structure (file, record, etc.). If there is no displacement, offset=0.	1	[integer]
axis_order	The axes order attribute indicates the axis index that varies the fastest with respect to axis element storage order.	1	FIRST_INDEX_FASTEST
Association	**Definition**	**Card.**	**Values**
has_Array_Element	The has_Array_Element association is a relationship to array element.	1	Array_Element
has_Array_Axis	The has_Array_Axis association is a relationship to array axis.	2	Array_Axis
associated_Statistics	The associated_Object_Statistics association is a relationship to object statistics.	0..1	Object_Statistics
associated_Special_Constants	The associated_Special_Constants association is a relationship to special constants.	0..1	Special_Constants

has_image_2d_display	The has_image_2d_display association is a relationship to Image_2D_Array.	0..1	Image_2D_Display

XML Example

```
    <Array_2D_Image base_class="Array_Base">
      <local_identifier>IMAGE</local_identifier>
      <axes>2</axes>
      <axis_order>FIRST_INDEX_FASTEST</axis_order>
      <encoding_type>BINARY</encoding_type>
      <offset unit_of_measure_type="byte">0</offset>
      <Object_Statistics>
        <local_identifier>STATISTICS</local_identifier>
        <maximum>4095</maximum>
        <md5_checksum>d220dac0d1fe312f3f3b9c824f6ac294</md5_checksum>
        <mean>1385.3000</mean>
        <median>894</median>
        <minimum>145</minimum>
        <sample_bit_mask>2#0000111111111111#</sample_bit_mask>
        <standard_deviation>538.0290</standard_deviation>
      </Object_Statistics>
      <Special_Constants>
        <error_constant>token</error_constant>
        <invalid_constant>token</invalid_constant>
        <missing_constant>token</missing_constant>
        <not_applicable_constant>token</not_applicable_constant>
        <saturated_constant>token</saturated_constant>
        <unknown_constant>token</unknown_constant>
      </Special_Constants>
      <Array_Axis>
        <name>SAMPLES
Array_2D_Image__Array_Axis::name1|LINES</name>
        <elements>256</elements>
        <sequence_number>1</sequence_number>
      </Array_Axis>
      <Array_Axis>
        <name>LINES</name>
        <unit>AU</unit>
        <elements>0</elements>
        <sequence_number>0</sequence_number>
      </Array_Axis>
      <Array_Element>
        <data_type>UnsignedMSB2</data_type>
        <scaling_factor>0.0</scaling_factor>
        <value_offset>0</value_offset>
        <unit>DN</unit>
      </Array_Element>
```

```
    <Image_2D_Display>
      <first_line>0</first_line>
      <first_line_sample>0</first_line_sample>
      <line_display_direction>DOWN</line_display_direction>
      <sample_display_direction>RIGHT</sample_display_direction>
    </Image_2D_Display>
  </Array_2D_Image>
```

A.2 Array_2D_Spectrum

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

Attributes & Associations

Attribute	**Definition**	**Card.**	**Values**
encoding_type	The encoding_type attribute provides the storage format (binary or character).	1	BINARY
local_identifier	The local_identifier attribute provides a character string which uniquely identifies the containing object within the label.	1	[string]
axes	The axes attribute provides a count of the axes.	1	2
offset	The offset attribute provides the displacement of the object starting position from the beginning of the parent structure (file, record, etc.). If there is no displacement, offset=0.	1	[integer]
axis_order	The axis order attribute provides the axis index that varies fastest with respect to storage order.	1	FIRST_INDEX_FASTEST
Association	**Definition**	**Card.**	**Values**
has_Array_Element	The has_Array_Element association is a relationship to array element.	1	Array_Element
has_Array_Axis	The has_Array_Axis association is a relationship to array axis.	2	Array_Axis

XML Example

```
<Array_2D_Spectrum base_class="Array_Base">
  <local_identifier>token</local_identifier>
  <comment>token</comment>
```

```
  <axes>2</axes>
  <axis_order>FIRST_INDEX_FASTEST</axis_order>
  <encoding_type>BINARY</encoding_type>
  <offset unit_of_measure_type="byte">0</offset>
  <Array_Axis>
    <name>token</name>
    <unit>AU</unit>
    <elements>0</elements>
    <sequence_number>0</sequence_number>
  </Array_Axis>
  <Array_Axis>
    <name>token</name>
    <unit>AU</unit>
    <elements>0</elements>
    <sequence_number>0</sequence_number>
  </Array_Axis>
  <Array_Element>
    <data_type>IEEE754Double</data_type>
    <scaling_factor>0.0</scaling_factor>
    <value_offset>0</value_offset>
    <unit>AU</unit>
  </Array_Element>
</Array_2D_Spectrum>
```

A.3 Array_2D_Map

The Array 2D Map class is an extension of array_base and defines a two dimensional map.

Attributes & Associations

Attribute	Definition	Card.	Values
encoding_type	The encoding_type attribute provides the storage format (binary or character).	1	BINARY
local_identifier	The local_identifier attribute provides a character string which uniquely identifies the containing object within the label.	1	[string]
axes	The axes attribute provides a count of the axes.	1	2
offset	The offset attribute provides the displacement of the object starting position from the beginning of the parent structure (file, record, etc.). If there is no displacement, offset=0.	1	[integer]
axis_order	The axis order attribute provides the axis index that varies fastest with respect to storage order.	1	FIRST_INDEX_FASTEST
Association	**Definition**	**Card.**	**Values**
has_Array_Element	The has_Array_Element association is a relationship to array element.	1	Array_Element
has_Array_Axis	The has_Array_Axis association is a relationship to array axis.	2	Array_Axis

XML Example

```
<Array_2D_Map base_class="Array_Base">
  <local_identifier>token</local_identifier>
  <comment>token</comment>
  <axes>2</axes>
```

```
    <axis_order>FIRST_INDEX_FASTEST</axis_order>
    <encoding_type>BINARY</encoding_type>
    <offset unit_of_measure_type="byte">0</offset>
    <Array_Axis>
      <name>token</name>
      <unit>AU</unit>
      <elements>0</elements>
      <sequence_number>0</sequence_number>
    </Array_Axis>
    <Array_Axis>
      <name>token</name>
      <unit>AU</unit>
      <elements>0</elements>
      <sequence_number>0</sequence_number>
    </Array_Axis>
    <Array_Element>
      <data_type>IEEE754Double</data_type>
      <scaling_factor>0.0</scaling_factor>
      <value_offset>0</value_offset>
      <unit>AU</unit>
    </Array_Element>
  </Array_2D_Map>
```

A.4 Array_3D_Image

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

Attributes & Associations

Attribute	**Definition**	**Card.**	**Values**
encoding_type	The encoding_type attribute provides the storage format (binary or character).	1	BINARY
local_identifier	The local_identifier attribute provides a character string which uniquely identifies the containing object within the label.	1	[string]
axes	The axes attribute provides a count of the axes.	1	3
offset	The offset attribute provides the displacement of the object starting position from the beginning of the parent structure (file, record, etc.). If there is no displacement, offset=0.	1	[integer]
axis_order	The axis order attribute provides the axis index that varies fastest with respect to storage order.	1	FIRST_INDEX_FASTEST
Association	**Definition**	**Card.**	**Values**
has_Array_Element	The has_Array_Element association is a relationship to array element.	1	Array_Element
has_Array_Axis	The has_Array_Axis association is a relationship to array axis.	3	Array_Axis

XML Example

```
<Array_3D_Image base_class="Array_Base">
  <local_identifier>token</local_identifier>
  <comment>token</comment>
```

```
    <axes>3</axes>
    <axis_order>FIRST_INDEX_FASTEST</axis_order>
    <encoding_type>BINARY</encoding_type>
    <offset unit_of_measure_type="byte">0</offset>
    <Array_Axis>
      <name>token</name>
      <unit>AU</unit>
      <elements>0</elements>
      <sequence_number>0</sequence_number>
    </Array_Axis>
    <Array_Axis>
      <name>token</name>
      <unit>AU</unit>
      <elements>0</elements>
      <sequence_number>0</sequence_number>
    </Array_Axis>
    <Array_Axis>
      <name>token</name>
      <unit>AU</unit>
      <elements>0</elements>
      <sequence_number>0</sequence_number>
    </Array_Axis>
    <Array_Element>
      <data_type>IEEE754Double</data_type>
      <scaling_factor>0.0</scaling_factor>
      <value_offset>0</value_offset>
      <unit>AU</unit>
    </Array_Element>
  </Array_3D_Image>
```

A.5 Array_3D_Spectrum

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

Attributes & Associations

Attribute	Definition	Card.	Values
encoding_type	The encoding_type attribute provides the storage format (binary or character).	1	BINARY
local_identifier	The local_identifier attribute provides a character string which uniquely identifies the containing object within the label.	1	[string]
axes	The axes attribute provides a count of the axes.	1	3
offset	The offset attribute provides the displacement of the object starting position from the beginning of the parent structure (file, record, etc.). If there is no displacement, offset=0.	1	[integer]
axis_order	The axis order attribute provides the axis index that varies fastest with respect to storage order.	1	FIRST_INDEX_FASTEST
Association	**Definition**	**Card.**	**Values**
has_Array_Element	The has_Array_Element association is a relationship to array element.	1	Array_Element
has_Array_Axis	The has_Array_Axis association is a relationship to array axis.	3	Array_Axis

XML Example

```
<Array_3D_Spectrum base_class="Array_Base">
  <local_identifier>token</local_identifier>
  <comment>token</comment>
```

```
  <axes>3</axes>
  <axis_order>FIRST_INDEX_FASTEST</axis_order>
  <encoding_type>BINARY</encoding_type>
  <offset unit_of_measure_type="byte">0</offset>
  <Array_Axis>
    <name>token</name>
    <unit>AU</unit>
    <elements>0</elements>
    <sequence_number>0</sequence_number>
  </Array_Axis>
  <Array_Axis>
    <name>token</name>
    <unit>AU</unit>
    <elements>0</elements>
    <sequence_number>0</sequence_number>
  </Array_Axis>
  <Array_Axis>
    <name>token</name>
    <unit>AU</unit>
    <elements>0</elements>
    <sequence_number>0</sequence_number>
  </Array_Axis>
  <Array_Element>
    <data_type>IEEE754Double</data_type>
    <scaling_factor>0.0</scaling_factor>
    <value_offset>0</value_offset>
    <unit>AU</unit>
  </Array_Element>
</Array_3D_Spectrum>
```

A.6 Array_3D_Movie

The Array 3D Movie class is an extension of array_base and defines a movie as a set of two dimensional images in a time series.

Attributes & Associations

Attribute	Definition	Card.	Values
encoding_type	The encoding_type attribute provides the storage format (binary or character).	1	BINARY
local_identifier	The local_identifier attribute provides a character string which uniquely identifies the containing object within the label.	1	[string]
axes	The axes attribute provides a count of the axes.	1	3
offset	The offset attribute provides the displacement of the object starting position from the beginning of the parent structure (file, record, etc.). If there is no displacement, offset=0.	1	[integer]
axis_order	The axis order attribute provides the axis index that varies fastest with respect to storage order.	1	FIRST_INDEX_FASTEST
Association	**Definition**	**Card.**	**Values**
has_Array_Element	The has_Array_Element association is a relationship to array element.	1	Array_Element
has_Array_Axis	The has_Array_Axis association is a relationship to array axis.	3	Array_Axis

XML Example

```
<Array_3D_Movie base_class="Array_Base">
  <local_identifier>token</local_identifier>
  <comment>token</comment>
```

```
  <axes>3</axes>
  <axis_order>FIRST_INDEX_FASTEST</axis_order>
  <encoding_type>BINARY</encoding_type>
  <offset unit_of_measure_type="byte">0</offset>
  <Array_Axis>
    <name>token</name>
    <unit>AU</unit>
    <elements>0</elements>
    <sequence_number>0</sequence_number>
  </Array_Axis>
  <Array_Axis>
    <name>token</name>
    <unit>AU</unit>
    <elements>0</elements>
    <sequence_number>0</sequence_number>
  </Array_Axis>
  <Array_Axis>
    <name>token</name>
    <unit>AU</unit>
    <elements>0</elements>
    <sequence_number>0</sequence_number>
  </Array_Axis>
  <Array_Element>
    <data_type>IEEE754Double</data_type>
    <scaling_factor>0.0</scaling_factor>
    <value_offset>0</value_offset>
    <unit>AU</unit>
  </Array_Element>
</Array_3D_Movie>
```

A.7 Table_Character

The Table Character class is an extension of table base and defines a simple character table.

Attributes & Associations

Attribute	Definition	Card.	Values
encoding_type	The encoding_type attribute provides the storage format (binary or character).	1	CHARACTER
local_identifier	The local_identifier attribute provides a character string which uniquely identifies the containing object within the label.	1	[string]
record_bytes	The record_bytes attribute provides a count of the bytes in a record, including a record delimiter, if present.	1	[integer]
fields	The fields attribute provides a count of the fields	1	[integer]
records	The records attribute provides a count of records.	1	[integer]
offset	The offset attribute provides the displacement of the object starting position from the beginning of the parent structure (file, record, etc.). If there is no displacement, offset=0.	1	[integer]
Association	**Definition**	**Card.**	**Values**
has_Record	The has_Record association is a relationship to record.	1	Table_Record_Character
uniformly_sampled	The uniformly_sampled association is a relationship to Uniformly_Sampled.	0..1	Uniformly_Sampled

XML Example

```
<Table_Character base_class="Table_Base">
```

```
  <local_identifier>token</local_identifier>
  <comment>token</comment>
  <encoding_type>CHARACTER</encoding_type>
  <fields>0</fields>
  <offset unit_of_measure_type="byte">0</offset>
  <record_bytes>0</record_bytes>
  <records>0</records>
  <Table_Record_Character>
    <Table_Character_Field>
      <name>token</name>
      <description>token</description>
      <field_number>0</field_number>
      <data_type>ASCII_AnyURI</data_type>
      <field_location>0</field_location>
      <field_length>0</field_length>
      <field_format>token</field_format>
      <minimum_scaled_value>token</minimum_scaled_value>
      <maximum_scaled_value>token</maximum_scaled_value>
      <scaling_factor>0.0</scaling_factor>
      <value_offset>0</value_offset>
      <unit>AU</unit>
      <Special_Constants>
        <error_constant>token</error_constant>
        <invalid_constant>token</invalid_constant>
        <missing_constant>token</missing_constant>
        <not_applicable_constant>token</not_applicable_constant>
        <saturated_constant>token</saturated_constant>
        <unknown_constant>token</unknown_constant>
      </Special_Constants>
      <Field_Statistics>
        <local_identifier>token</local_identifier>
        <description>token</description>
        <maximum>0.0</maximum>
        <mean>0.0</mean>
        <median>0.0</median>
        <minimum>0.0</minimum>
        <sample_bit_mask>sample_bit_mask</sample_bit_mask>
        <standard_deviation>0.0</standard_deviation>
      </Field_Statistics>
    </Table_Character_Field>
  </Table_Record_Character>
  <Uniformly_Sampled>
    <first_sampling_parameter_value>0.0</first_sampling_parameter_value>
    <last_sampling_parameter_value>0.0</last_sampling_parameter_value>
    <sampling_parameter_interval>0.0</sampling_parameter_interval>
    <sampling_parameter_name>along_track_distance</sampling_parameter_name>
    <sampling_parameter_scale>EXPONENTIAL</sampling_parameter_scale>
    <sampling_parameter_unit>...</sampling_parameter_unit>
  </Uniformly_Sampled>
</Table_Character>
```

A.8 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.

Attributes & Associations

Attribute	Definition	Card.	Values
encoding_type	The encoding_type attribute provides the storage format (binary or character).	1	CHARACTER
local_identifier	The local_identifier attribute provides a character string which uniquely identifies the containing object within the label.	1	[string]
record_bytes	The record_bytes attribute provides a count of the bytes in a record, including a record delimiter, if present.	1	[integer]
fields	The fields attribute provides a count of the fields	1	[integer]
records	The records attribute provides a count of records.	1	[integer]
offset	The offset attribute provides the displacement of the object starting position from the beginning of the parent structure (file, record, etc.). If there is no displacement, offset=0.	1	[integer]
Association	**Definition**	**Card.**	**Values**
has_Record	The has_Record association is a relationship to record.	1	Table_Record_Character_Grouped

XML Example

```
<Table_Character_Grouped base_class="Table_Base">
  <local_identifier>token</local_identifier>
  <comment>token</comment>
```

```
  <encoding_type>CHARACTER</encoding_type>
  <fields>0</fields>
  <offset unit_of_measure_type="byte">0</offset>
  <record_bytes>0</record_bytes>
  <records>0</records>
  <Table_Record_Character_Grouped>
    <Table_Character_Grouped_Sequence>
      <repetitions>0</repetitions>
      <Table_Character_Field_Sequence>
        <Table_Character_Grouped_Field>
          <name>token</name>
          <description>token</description>
          <field_number>0</field_number>
          <data_type>ASCII_AnyURI</data_type>
          <field_location>0</field_location>
          <field_length>0</field_length>
          <field_format>token</field_format>
          <minimum_scaled_value>token</minimum_scaled_value>
          <maximum_scaled_value>token</maximum_scaled_value>
          <scaling_factor>0.0</scaling_factor>
          <value_offset>0</value_offset>
          <unit>AU</unit>
        </Table_Character_Grouped_Field>
        <Table_Character_Grouped_Sequence/>
      </Table_Character_Field_Sequence>
    </Table_Character_Grouped_Sequence>
  </Table_Record_Character_Grouped>
</Table_Character_Grouped>
```

A.9 Inventory_LIDVID_Primary

The Inventory LIDVID Primary class defines the inventory for primary members of a collection.

Attributes & Associations

Attribute	Definition	Card.	Values
encoding_type	The encoding_type attribute provides the storage format (binary or character).	1	CHARACTER
local_identifier	The local_identifier attribute provides a character string which uniquely identifies the containing object within the label.	1	[string]
record_bytes	The record_bytes attribute provides a count of the bytes in a record, including a record delimiter, if present.	1	[integer]
fields	The fields attribute provides a count of the fields	1	2
records	The records attribute provides a count of records.	1	[integer]
offset	The offset attribute provides the displacement of the object starting position from the beginning of the parent structure (file, record, etc.). If there is no displacement, offset=0.	1	[integer]
reference_association_type	The reference_association_type attribute describes the type of association.	1	has_member_LIDVID_Primary
Association	**Definition**	**Card.**	**Values**

has_Record	The has_Record association is a relationship to record.	1	Table_Record_Inventory_LIDVID_Primary

XML Example

```
<Inventory_LIDVID_Primary base_class="Table_Base">
  <local_identifier>token</local_identifier>
  <comment>token</comment>
  <encoding_type>CHARACTER</encoding_type>
  <fields>2</fields>
  <offset unit_of_measure_type="byte">0</offset>
  <record_bytes>0</record_bytes>
  <records>0</records>
  <reference_association_type>has_member</reference_association_type>
  <Table_Record_Inventory_LIDVID_Primary>
    <Table_Field_LIDVID>
      <name>LIDVID</name>
      <description>token</description>
      <field_number>1</field_number>
      <data_type>ASCII_LIDVID</data_type>
      <field_location>1</field_location>
      <field_length>0</field_length>
      <field_format>urn:nasa:pds:xxxx::M.n</field_format>
    </Table_Field_LIDVID>
    <Table_Field_File_Specification_Name>
      <name>file_specification_name</name>
      <description>token</description>
      <field_number>2</field_number>
      <data_type>ASCII_File_Specification_Name</data_type>
      <field_location>0</field_location>
      <field_length>0</field_length>
      <field_format>dir1/dir2/file_name.file_extension</field_format>
    </Table_Field_File_Specification_Name>
  </Table_Record_Inventory_LIDVID_Primary>
</Inventory_LIDVID_Primary>
```

A.10 Inventory_LIDVID_Secondary

The Inventory LIDVID Secondary class defines the inventory for secondary members of a collection. The references are LIDVIDs.

Attributes & Associations

Attribute	Definition	Card.	Values
encoding_type	The encoding_type attribute provides the storage format (binary or character).	1	CHARACTER
local_identifier	The local_identifier attribute provides a character string which uniquely identifies the containing object within the label.	1	[string]
record_bytes	The record_bytes attribute provides a count of the bytes in a record, including a record delimiter, if present.	1	[integer]
fields	The fields attribute provides a count of the fields	1	1
records	The records attribute provides a count of records.	1	[integer]
offset	The offset attribute provides the displacement of the object starting position from the beginning of the parent structure (file, record, etc.). If there is no displacement, offset=0.	1	[integer]
reference_association_type	The reference_association_type attribute describes the type of association.	1	has_member_LIDVID_Secondary
Association	**Definition**	**Card.**	**Values**

has_Record	The has_Record association is a relationship to record.	1	Table_Record_Inventory_LIDVID_Secondary

XML Example

```
<Inventory_LIDVID_Secondary base_class="Table_Base">
  <local_identifier>token</local_identifier>
  <comment>token</comment>
  <encoding_type>CHARACTER</encoding_type>
  <fields>1</fields>
  <offset unit_of_measure_type="byte">0</offset>
  <record_bytes>0</record_bytes>
  <records>0</records>
  <reference_association_type>has_member</reference_association_type>
  <Table_Record_Inventory_LIDVID_Secondary>
    <Table_Field_LIDVID>
      <name>LIDVID</name>
      <description>token</description>
      <field_number>1</field_number>
      <data_type>ASCII_LIDVID</data_type>
      <field_location>1</field_location>
      <field_length>0</field_length>
      <field_format>urn:nasa:pds:xxxx::M.n</field_format>
    </Table_Field_LIDVID>
  </Table_Record_Inventory_LIDVID_Secondary>
</Inventory_LIDVID_Secondary>
```

A.11 Inventory_LID_Secondary

The Inventory LID Secondary class defines the inventory for secondary members of a collection. The references are LIDs.

Attributes & Associations

Attribute	Definition	Card.	Values
encoding_type	The encoding_type attribute provides the storage format (binary or character).	1	CHARACTER
local_identifier	The local_identifier attribute provides a character string which uniquely identifies the containing object within the label.	1	[string]
record_bytes	The record_bytes attribute provides a count of the bytes in a record, including a record delimiter, if present.	1	[integer]
fields	The fields attribute provides a count of the fields	1	1
records	The records attribute provides a count of records.	1	[integer]
offset	The offset attribute provides the displacement of the object starting position from the beginning of the parent structure (file, record, etc.). If there is no displacement, offset=0.	1	[integer]
reference_association_type	The reference_association_type attribute describes the type of association.	1	has_member_LID_Secondary
Association	**Definition**	**Card.**	**Values**

has_Record	The has_Record association is a relationship to record.	1	Table_Record_Inventory_LID_Secondary

XML Example

```
<Inventory_LID_Secondary base_class="Table_Base">
  <local_identifier>token</local_identifier>
  <comment>token</comment>
  <encoding_type>CHARACTER</encoding_type>
  <fields>1</fields>
  <offset unit_of_measure_type="byte">0</offset>
  <record_bytes>0</record_bytes>
  <records>0</records>
  <reference_association_type>has_member</reference_association_type>
  <Table_Record_Inventory_LID_Secondary>
    <Table_Field_LID>
      <name>LID</name>
      <description>token</description>
      <field_number>1</field_number>
      <data_type>ASCII_LID</data_type>
      <field_location>1</field_location>
      <field_length>0</field_length>
      <field_format>urn:nasa:pds:xxxx</field_format>
    </Table_Field_LID>
  </Table_Record_Inventory_LID_Secondary>
</Inventory_LID_Secondary>
```

A.12 Delivery_Manifest

The Delivery_Manifest class defines a two column table for file references. The table structure is compatible with the output from an MD5 checksum utility.

Attributes & Associations

Attribute	**Definition**	**Card.**	**Values**
encoding_type	The encoding_type attribute provides the storage format (binary or character).	1	CHARACTER
local_identifier	The local_identifier attribute provides a character string which uniquely identifies the containing object within the label.	1	[string]
record_bytes	The record_bytes attribute provides a count of the bytes in a record, including a record delimiter, if present.	1	[integer]
fields	The fields attribute provides a count of the fields	1	2
records	The records attribute provides a count of records.	1	[integer]
offset	The offset attribute provides the displacement of the object starting position from the beginning of the parent structure (file, record, etc.). If there is no displacement, offset=0.	1	[integer]
reference_association_type	The reference_association_type attribute describes the type of association.		has_member
Association	**Definition**	**Card.**	**Values**
has_Record	The has_Record association is a relationship to record.	1	Table_Record_Manifest

XML Example

A.13 Table_Binary

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

Attributes & Associations

Attribute	**Definition**	**Card.**	**Values**
encoding_type	The encoding_type attribute provides the storage format (binary or character).	1	BINARY
local_identifier	The local_identifier attribute provides a character string which uniquely identifies the containing object within the label.	1	[string]
record_bytes	The record_bytes attribute provides a count of the bytes in a record, including a record delimiter, if present.	1	[integer]
fields	The fields attribute provides a count of the fields	1	[integer]
records	The records attribute provides a count of records.	1	[integer]
offset	The offset attribute provides the displacement of the object starting position from the beginning of the parent structure (file, record, etc.). If there is no displacement, offset=0.	1	[integer]
Association	**Definition**	**Card.**	**Values**
has_Record	The has_Record association is a relationship to record.	1	Table_Record_Binary
uniformly_sampled	The uniformly_sampled association is a relationship to Uniformly_Sampled.	0..1	Uniformly_Sampled

XML Example

```
<Table_Binary base_class="Table_Base">
```

```
  <local_identifier>token</local_identifier>
  <comment>token</comment>
  <encoding_type>BINARY</encoding_type>
  <fields>0</fields>
  <offset unit_of_measure_type="byte">0</offset>
  <record_bytes>0</record_bytes>
  <records>0</records>
  <Table_Record_Binary>
    <Table_Binary_Field>
      <name>token</name>
      <description>token</description>
      <field_number>0</field_number>
      <data_type>ASCII_AnyURI</data_type>
      <field_location>0</field_location>
      <field_length>0</field_length>
      <field_format>token</field_format>
      <minimum_scaled_value>token</minimum_scaled_value>
      <maximum_scaled_value>token</maximum_scaled_value>
      <scaling_factor>0.0</scaling_factor>
      <value_offset>0</value_offset>
      <unit>AU</unit>
      <Special_Constants>
        <error_constant>token</error_constant>
        <invalid_constant>token</invalid_constant>
        <missing_constant>token</missing_constant>
        <not_applicable_constant>token</not_applicable_constant>
        <saturated_constant>token</saturated_constant>
        <unknown_constant>token</unknown_constant>
      </Special_Constants>
      <Field_Statistics>
        <local_identifier>token</local_identifier>
        <description>token</description>
        <maximum>0.0</maximum>
        <mean>0.0</mean>
        <median>0.0</median>
        <minimum>0.0</minimum>
        <sample_bit_mask>sample_bit_mask</sample_bit_mask>
        <standard_deviation>0.0</standard_deviation>
      </Field_Statistics>
    </Table_Binary_Field>
  </Table_Record_Binary>
  <Uniformly_Sampled>
    <first_sampling_parameter_value>0.0</first_sampling_parameter_value>
    <last_sampling_parameter_value>0.0</last_sampling_parameter_value>
    <sampling_parameter_interval>0.0</sampling_parameter_interval>
    <sampling_parameter_name>along_track_distance</sampling_parameter_name>
    <sampling_parameter_scale>EXPONENTIAL</sampling_parameter_scale>
    <sampling_parameter_unit>...</sampling_parameter_unit>
  </Uniformly_Sampled>
</Table_Binary>
```

A.14 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.

Attributes & Associations

Attribute	Definition	Card.	Values
encoding_type	The encoding_type attribute provides the storage format (binary or character).	1	BINARY
local_identifier	The local_identifier attribute provides a character string which uniquely identifies the containing object within the label.	1	[string]
record_bytes	The record_bytes attribute provides a count of the bytes in a record, including a record delimiter, if present.	1	[integer]
fields	The fields attribute provides a count of the fields	1	[integer]
records	The records attribute provides a count of records.	1	[integer]
offset	The offset attribute provides the displacement of the object starting position from the beginning of the parent structure (file, record, etc.). If there is no displacement, offset=0.	1	[integer]
Association	**Definition**	**Card.**	**Values**
has_Record	The has_Record association is a relationship to record.	1	Table_Record_Binary_Grouped

XML Example

```
<Table_Binary_Grouped base_class="Table_Base">
  <local_identifier>token</local_identifier>
  <comment>token</comment>
```

```xml
    <encoding_type>BINARY</encoding_type>
    <fields>0</fields>
    <offset unit_of_measure_type="byte">0</offset>
    <record_bytes>0</record_bytes>
    <records>0</records>
    <Table_Record_Binary_Grouped>
      <Table_Binary_Grouped_Sequence>
        <repetitions>0</repetitions>
        <Table_Binary_Field_Sequence>
          <Table_Binary_Grouped_Bit_Field>
            <name>token</name>
            <description>token</description>
            <field_number>0</field_number>
            <data_type>Bit</data_type>
            <field_location>0</field_location>
            <field_length>0</field_length>
            <field_format>token</field_format>
            <minimum_scaled_value>token</minimum_scaled_value>
            <maximum_scaled_value>token</maximum_scaled_value>
            <scaling_factor>0.0</scaling_factor>
            <value_offset>0</value_offset>
            <unit>AU</unit>
            <bit_mask>bit_mask</bit_mask>
            <bits>0</bits>
            <start_bit>0</start_bit>
          </Table_Binary_Grouped_Bit_Field>
          <Table_Binary_Grouped_Field>
            <name>token</name>
            <description>token</description>
            <field_number>0</field_number>
            <data_type>ASCII_File_Name</data_type>
            <field_location>0</field_location>
            <field_length>0</field_length>
            <field_format>token</field_format>
            <minimum_scaled_value>token</minimum_scaled_value>
            <maximum_scaled_value>token</maximum_scaled_value>
            <scaling_factor>0.0</scaling_factor>
            <value_offset>0</value_offset>
            <unit>AU</unit>
          </Table_Binary_Grouped_Field>
          <Table_Binary_Grouped_Sequence/>
        </Table_Binary_Field_Sequence>
      </Table_Binary_Grouped_Sequence>
    </Table_Record_Binary_Grouped>
  </Table_Binary_Grouped>
```

A.15 Stream_Delimited

The Stream Delimited class defines a simple spreadsheet.

Attributes & Associations

Attribute	**Definition**	**Card.**	**Values**
encoding_type	The encoding_type attribute provides the storage format (binary or character).	1	CHARACTER
local_identifier	The local_identifier attribute provides a character string which uniquely identifies the containing object within the label.	1	[string]
external_standard_id	The external_standard_id attribute provides the formal name of a standard not under PDS governance.	1	CSV OTHER
offset	The offset attribute provides the displacement of the object starting position from the beginning of the parent structure (file, record, etc.). If there is no displacement, offset=0.	1	[integer]
fields	The fields attribute provides a count of the fields	1	[integer]
record_delimiter	The record delimiter attribute provides the character or characters used to indicate the end of a record.	1	0xOA 0xOD 0xOD_0xOA
records	The records attribute provides a count of records.	1	[integer]
maximum_record_length	The maximum_record_length attribute provides the upper, inclusive bound on the length of a record, including any record delimiter.	1	[string]
field_delimiter	The field_delimiter attribute provides the character or characters that indicate the end of a character string.	1	0x09 0x3B 0x7C 0x2C
Association	**Definition**	**Card.**	**Values**

has_stream_record	The has_stream_record association is a relationship to record.	1..*	Stream_Delimited_Record

XML Example

```
<Stream_Delimited>
  <local_identifier>token</local_identifier>
  <comment>token</comment>
  <encoding_type>CHARACTER</encoding_type>
  <external_standard_id>CSV</external_standard_id>
  <field_delimiter>0x09</field_delimiter>
  <fields>0</fields>
  <maximum_record_length>0</maximum_record_length>
  <offset unit_of_measure_type="byte">0</offset>
  <record_delimiter>0xOA</record_delimiter>
  <records>0</records>
  <Stream_Delimited_Record>
    <Stream_Delimited_Grouped_Sequence>
      <repetitions>0</repetitions>
      <Stream_Delimited_Field_Sequence>
        <Stream_Delimited_Field>
          <name>token</name>
          <description>token</description>
          <field_number>0</field_number>
          <data_type>IEEE754Double</data_type>
          <field_length>0</field_length>
          <field_format>token</field_format>
          <minimum_scaled_value>token</minimum_scaled_value>
          <maximum_scaled_value>token</maximum_scaled_value>
          <scaling_factor>0.0</scaling_factor>
          <value_offset>0</value_offset>
          <unit>AU</unit>
          <Special_Constants>
            <error_constant>token</error_constant>
            <invalid_constant>token</invalid_constant>
            <missing_constant>token</missing_constant>
            <not_applicable_constant>token</not_applicable_constant>
            <saturated_constant>token</saturated_constant>
            <unknown_constant>token</unknown_constant>
          </Special_Constants>
          <Field_Statistics>
            <local_identifier>token</local_identifier>
            <description>token</description>
            <maximum>0.0</maximum>
            <mean>0.0</mean>
            <median>0.0</median>
            <minimum>0.0</minimum>
```

```
            <sample_bit_mask>sample_bit_mask</sample_bit_mask>
            <standard_deviation>0.0</standard_deviation>
          </Field_Statistics>
        </Stream_Delimited_Field>
        <Stream_Delimited_Grouped_Sequence/>
      </Stream_Delimited_Field_Sequence>
    </Stream_Delimited_Grouped_Sequence>
  </Stream_Delimited_Record>
</Stream_Delimited>
```

A.16 XML_Schema

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

Attributes & Associations

Attribute	Definition	Card.	Values
encoding_type	The encoding_type attribute provides the storage format (binary or character).	1	CHARACTER
local_identifier	The local_identifier attribute provides a character string which uniquely identifies the containing object within the label.	1	[string]
external_standard_id	The external_standard_id attribute provides the formal name of a standard not under PDS governance.	1	XML_Schema
offset	The offset attribute provides the displacement of the object starting position from the beginning of the parent structure (file, record, etc.). If there is no displacement, offset=0.	1	[integer]

XML Example

```
<XML_Schema>
  <local_identifier>token</local_identifier>
  <comment>token</comment>
  <encoding_type>CHARACTER</encoding_type>
  <external_standard_id>XML_Schema</external_standard_id>
  <offset unit_of_measure_type="byte">0</offset>
</XML_Schema>
```

A.17 Service_Description

The Service Description class defines a file that contains a standardized service specification.

Attributes & Associations

Attribute	Definition	Card.	Values
encoding_type	The encoding_type attribute provides the storage format (binary or character).	1	CHARACTER
local_identifier	The local_identifier attribute provides a character string which uniquely identifies the containing object within the label.	1	[string]
external_standard_id	The external_standard_id attribute provides the formal name of a standard not under PDS governance.	1	WSDL WADL
offset	The offset attribute provides the displacement of the object starting position from the beginning of the parent structure (file, record, etc.). If there is no displacement, offset=0.	1	[integer]

XML Example

A.18 SPICE_Kernel_Text

The SPICE Kernel class describes a SPICE file.

Attributes & Associations

Attribute	Definition	Card.	Values
encoding_type	The encoding_type attribute provides the storage format (binary or character).	1	CHARACTER
local_identifier	The local_identifier attribute provides a character string which uniquely identifies the containing object within the label.	1	[string]
external_standard_id	The external_standard_id attribute provides the formal name of a standard not under PDS governance.	1	SPICE
offset	The offset attribute provides the displacement of the object starting position from the beginning of the parent structure (file, record, etc.). If there is no displacement, offset=0.	1	[integer]
kernel_type	The kernel_type attribute identifies the type of SPICE kernel.	1	LSK SCLK PCK EK IK FK MK

XML Example

```
<SPICE_Kernel_Text>
  <local_identifier>token</local_identifier>
  <comment>token</comment>
  <encoding_type>CHARACTER</encoding_type>
  <external_standard_id>SPICE</external_standard_id>
  <kernel_type>EK</kernel_type>
  <offset unit_of_measure_type="byte">0</offset>
```

```
</SPICE_Kernel_Text>
```

A.19 Header

The Header class describes a data object header.

Attributes & Associations

Attribute	Definition	Card.	Values
encoding_type	The encoding_type attribute provides the storage format (binary or character).	1	CHARACTER
local_identifier	The local_identifier attribute provides a character string which uniquely identifies the containing object within the label.	1	[string]
external_standard_id	The external_standard_id attribute provides the formal name of a standard not under PDS governance.	1	VICAR ODL FITS ISIS
offset	The offset attribute provides the displacement of the object starting position from the beginning of the parent structure (file, record, etc.). If there is no displacement, offset=0.	1	[integer]
bytes	The bytes attribute provides the number of bytes in the object.	1	[string]
description	The description attribute provides a statement, picture in words, or account that describes or is otherwise relevant to the object.	0..1	[string]
name	The name attribute provides a word or combination of words by which the object is known.	0..1	[string]

XML Example

```
<Header>
  <local_identifier>token</local_identifier>
  <comment>token</comment>
```

```
    <name>token</name>
    <description>token</description>
    <bytes>0</bytes>
    <encoding_type>CHARACTER</encoding_type>
    <external_standard_id>FITS</external_standard_id>
    <offset unit_of_measure_type="byte">0</offset>
  </Header>
```

A.20 File_PDF

The File PDF class describes a PDF encoded byte stream.

Attributes & Associations

Attribute	Definition	Card.	Values
encoding_type	The encoding_type attribute provides the storage format (binary or character).	1	BINARY
local_identifier	The local_identifier attribute provides a character string which uniquely identifies the containing object within the label.	1	[string]
external_standard_id	The external_standard_id attribute provides the formal name of a standard not under PDS governance.	1	PDF
offset	The offset attribute provides the displacement of the object starting position from the beginning of the parent structure (file, record, etc.). If there is no displacement, offset=0.	1	[integer]

XML Example

```
<File_PDF>
  <local_identifier>token</local_identifier>
  <comment>token</comment>
  <encoding_type>BINARY</encoding_type>
  <external_standard_id>PDF</external_standard_id>
  <offset unit_of_measure_type="byte">0</offset>
</File_PDF>
```

A.21 SPICE_Kernel_Binary

The SPICE Kernel class describes a SPICE file.

Attributes & Associations

Attribute	Definition	Card.	Values
encoding_type	The encoding_type attribute provides the storage format (binary or character).	1	BINARY
local_identifier	The local_identifier attribute provides a character string which uniquely identifies the containing object within the label.	1	[string]
external_standard_id	The external_standard_id attribute provides the formal name of a standard not under PDS governance.	1	SPICE
offset	The offset attribute provides the displacement of the object starting position from the beginning of the parent structure (file, record, etc.). If there is no displacement, offset=0.	1	[integer]
kernel_type	The kernel_type attribute identifies the type of SPICE kernel.	1	SPK PCK CK EK DSK DBK

XML Example

```
<SPICE_Kernel_Binary>
  <local_identifier>token</local_identifier>
  <comment>token</comment>
  <encoding_type>BINARY</encoding_type>
  <external_standard_id>SPICE</external_standard_id>
  <kernel_type>EK</kernel_type>
  <offset unit_of_measure_type="byte">0</offset>
</SPICE_Kernel_Binary>
```

A.22 Header_Binary

The Header Binary class describes a data object header.

Attributes & Associations

Attribute	**Definition**	**Card.**	**Values**
encoding_type	The encoding_type attribute provides the storage format (binary or character).	1	BINARY
local_identifier	The local_identifier attribute provides a character string which uniquely identifies the containing object within the label.	1	[string]
external_standard_id	The external_standard_id attribute provides the formal name of a standard not under PDS governance.	1	VICAR ODL FITS ISIS
offset	The offset attribute provides the displacement of the object starting position from the beginning of the parent structure (file, record, etc.). If there is no displacement, offset=0.	1	[integer]
bytes	The bytes attribute provides the number of bytes in the object.	1	[string]
description	The description attribute provides a statement, picture in words, or account that describes or is otherwise relevant to the object.	0..1	[string]
name	The name attribute provides a word or combination of words by which the object is known.	0..1	[string]

XML Example

A.23 Encoded_Image

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

Attributes & Associations

Attribute	Definition	Card.	Values
encoding_type	The encoding_type attribute provides the storage format (binary or character).	1	BINARY
local_identifier	The local_identifier attribute provides a character string which uniquely identifies the containing object within the label.	1	[string]
external_standard_id	The external_standard_id attribute provides the formal name of a standard not under PDS governance.	1	JPEG GIF TIFF PDF
offset	The offset attribute provides the displacement of the object starting position from the beginning of the parent structure (file, record, etc.). If there is no displacement, offset=0.	1	[integer]

XML Example

```
<Encoded_Image>
  <local_identifier>token</local_identifier>
  <comment>token</comment>
  <encoding_type>BINARY</encoding_type>
  <external_standard_id>GIF</external_standard_id>
  <offset unit_of_measure_type="byte">0</offset>
</Encoded_Image>
```

A.24 Encoded_Binary

The Encoded Binary class describes a binary encoded byte stream.This class is used to describe files in the repository that are being registered using Product_File_Repository.

Attributes & Associations

Attribute	Definition	Card.	Values
encoding_type	The encoding_type attribute provides the storage format (binary or character).	1	BINARY
local_identifier	The local_identifier attribute provides a character string which uniquely identifies the containing object within the label.	1	[string]
external_standard_id	The external_standard_id attribute provides the formal name of a standard not under PDS governance.	1	System
offset	The offset attribute provides the displacement of the object starting position from the beginning of the parent structure (file, record, etc.). If there is no displacement, offset=0.	1	[integer]

XML Example

Acronyms

ASCII American Standard Code for Information Interchange. 29, 30, 34, 35

BCRS Barycentric Celestial Reference System. 65

CCSDS Consultative Committee for Space Data Systems. 2, 3

DIM digital image model. 70

DOI Digital Object Identifier. 35

DOY Day Of Year. 33, 34

DPH Data Provider's Handbook. 1, 2

DTM digital terrain model. 70

ebXML electronic business eXtensible Markup Language. 3

EME Earth's Mean Equator. 65

FMS fictitious mean Sun. 87, 88

FOV Field of View. 18

HTML Hypertext Markup Language. 32

IAG International Association of Geodesy. 64

IAU International Astronomical Union. 63–65

ICRF International Celestial Reference Frame. 63, 65, 66, 69

ICRS International Celestial Reference System. 63, 65

IEEE Institute of Electrical and Electronics Engineers. 3

IERS International Earth Rotation and Reference Systems Service. 63

ISO International Standards Organization. 3

ISO/IEC International Standards Organization / International Electrotechnical Commission. 3

ISO/TS International Standards Organization / Technical Standard. 3

JD Julian Date. 65

JPL Jet Propulsion Laboratory. 68

LGCWG Lunar Geodesy and Cartography Working Group. 64, 68, 73

LMST local mean solar time. 87, 88

LPRP Lunar Precursor Robotics Program. 64

LRO Lunar Reconnaissance Orbiter. 68

LSB Least Significant Byte. 37, 39

LTST local true solar time. 86–88

MD5 Message-Digest algorithm 5. 34

ME Mean Earth/Polar Axis. 68

MESSENGER MErcury Surface, Space ENvironment, GEochemistry, and Ranging. 67

MGCWG Mars Geodesy and Cartography Working Group. 64, 67

MOLA Mars Orbiter Laser Altimeter. 70

MSB Most Significant Byte. 40, 41

NASA National Aeronautics and Space Administration. 63–65, 67, 68, 73

NIST National Institute of Standards and Technology. 3

PA Principal Axis. 68

PCGMWG Planetary Cartography and Geologic Mapping Working Group. 73

PCWG Planetary Cartography Working Group. 73

PDF Portable Document Format. 32, 67

PDS Planetary Data System. 1–4, 7–9, 11, 21, 23, 29, 32, 63–66, 68–70, 73, 91, 97

PDS4 Planetary Data System, standards version 4. 7, 21, 23

PPI Planetary Plasma Interactions. 70

PSDD Planetary Science Data Dictionary. 21, 91

RA right ascension. 87

SEDR Supplementary Experiment Data Record. 17

SI International System of Units / Systeme Internationale d'Unites. 64, 91, 92, 94

SPICE Spacecraft, Planet, Instrument, C-matrix (pointing), and Events kernels (see the PDS NAIF node for more details). 16–18, 97

TDB Barycentric Dynamical Time. 65

TT Terrestrial Time. 65

URL Uniform Resource Locator. 3, 4

UTC Coordinated Universal Time. 34, 65

UTF Unicode Transformation Format. 30

W3C World Wide Web Consortium. 3

WGCCRE Working Group on Cartographic Coordinates and Rotational Elements. 64, 66–69

XML eXtensible Markup Language. 1, 3, 7, 12, 23, 32

XSD XML schema document. 23

YMD Year Month Day. 33, 34

Bibliography

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

B. A. Archinal and the Lunar Geodesy and Cartography Working Group. Lunar Mapping Standards and the NASA LPRP Lunar Geodesy and Cartography Working Group. 37th COSPAR Scientific Assembly, July 13-20, Montreal, Canada, July 2008a.

B. A. Archinal and the Lunar Geodesy and Cartography Working Group. 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, July 2008b.

B. A. Archinal, M. F. A'Hearn, E. Bowell, A. Conrad, G. J. Consolmagno, R. Courtin, T. Fukushima, D. Hestroffer, J. L. Hilton, G. A. Krasinsky, G. Neumann, J. Oberst, P. K. Seidelmann, P. Stooke, D. J. Tholen, P. C. Thomas, and I. P. Williams. Report of the IAU Working Group on Cartographic Coordinates and Rotational Elements: 2009. *Celestial Mechanics and Dynamical Astronomy*, 109(2):101–135, January 2011. doi: 10.1007/s10569-010-9320-4.

R. M. Batson. Digital Cartography of the Planets: New Methods, its Status and Future. *Photogrammetric Engineering and Remote Sensing*, 53:1211–1218, 1987.

J. Chapront, M. Chapront-Touzé, and G. Francou. A new determination of lunar orbital parameters, precession constant and tidal acceleration from LLR measurements. *Astronomy and Astrophysics*, 387:700–709, 2002.

Committee on Data Management and Computation, Space Science Board, Commission on Physical Sciences, Mathematics, and Resources, and National Research Council. *Issues and Recommendations Associated with Distributed Computation and Data Management Systems for the Space Sciences*. National Academy Press, Washington, D.C., 1986.

M. E. Davies, V. K. Abalakin, C. A. Cross, R. L. Duncombe, H. Masursky, B. Morando, T. C. Owen, P. K. Seidelmann, A. T. Sinclair, G. A. Wilkins, and Y. S. Tjuflin. Report of the IAU Working Group on Cartographic Coordinates and Rotation Elements of the Planets and Satellites. *Celestial Mechanics*, 22:205–230, October 1980. doi: 10.1007/BF01229508.

M. E. Davies, V. K. Abalakin, J. H. Lieske, P. K. Seidelmann, A. T. Sinclair, A. M. Sinzi, B. A.

Smith, and Y. S. Tjuflin. Report of the IAU Working Group on Cartographic Coordinates and Rotation Elements of the Planets and Satellites: 1982. *Celestial Mechanics*, 29:309–321, April 1983. doi: 10.1007/BF01228525.

M. E. Davies, V. K. Abalakin, M. Bursa, T. Lederle, J. H. Lieske, R. H. Rapp, P. K. Seidelmann, A. T. Sinclair, V. G. Teifel, and Y. S. Tjuflin. Report of the IAU/IAG/COSPAR Working Group on Cartographic Coordinates and Rotation Elements of the Planets and Satellites: 1985. *Celestial Mechanics*, 39:103–113, May 1986. doi: 10.1007/BF01232291.

M. E. Davies, V. K. Abalakin, M. Bursa, G. E. Hunt, J. H. Lieske, B. Morando, R. H. Rapp, P. K. Seidelmann, A. T. Sinclair, and Y. S. Tjuflin. Report of the IAU/IAG/COSPAR Working Group on Cartographic Coordinates and Rotation Elements of the Planets and Satellites: 1988. *Celestial Mechanics and Dynamical Astronomy*, 46:187–204, June 1989. doi: 10.1007/BF00053048.

M. E. Davies, V. K. Abalakin, A. Brahic, M. Bursa, B. H. Chovitz, P. K. Seidelmann, A. T. Sinclair, and Y. S. Tjuflin. Report of the IAU/IAG/COSPAR Working Group on Cartographic Coordinates and Rotation Elements of the Planets and Satellites: 1991. *Celestial Mechanics and Dynamical Astronomy*, 53:377–397, December 1992. doi: 10.1007/BF00051818.

M. E. Davies, V. K. Abalakin, M. Bursa, J. H. Lieske, B. Morando, D. Morrison, P. K. Seidelmann, A. T. Sinclair, B. Yallop, and Y. S. Tjuflin. Report of the IAU/IAG/COSPAR Working Group on Cartographic Coordinates and Rotational Elements of the Planets and Satellites: 1994. *Celestial Mechanics and Dynamical Astronomy*, 63(2):127–148, June 1996. doi: 10.1007/BF00693410.

T. C. Duxbury, R. L. Kirk, B. A. Archinal, and G. A. Neumannn. Mars Geodesy/Cartography Working Group Recommendations on Mars Cartographic Constants and Coordinate Systems. In *Commission IV, WG IV/9: Extraterrestrial Mapping Workshop from the Joint International Symposium on Geospatial Theory, Processing and Applications*, July 2002.

M. Franz and D. Harper. Heliospheric Coordinate Systems. *Planetary and Space Science*, 50(2): 217–233, 2002.

R. Greeley and R. M. Batson. *Planetary Mapping*. Cambridge University Press, Cambridge, 1990.

T. A. Herring, P. M. Mathews, and B. A. Buffett. Modeling of nutation-precession: Very long baseline interferometry results. *J. Geophys. Res.*, 107(B4):2069, 2002. doi: 10.1029/2001JB000165.

International Astronomical Union. Proceedings of the Twenty-Fourth General Assembly, Manchester 2000. In *Transactions of the IAU*, volume XXIV-B, pages 33–57, 2002.

J. Kovalevsky and I. I. Mueller. Comments on Conventional Terrestrial and Quasi-Intertial Reference Systems. In E. M. Gaposchkin and B. Kolaczek, editors, *Reference Coordinate Systems for Earth Dynamics, Proceedings of IAU Colloq. 56, Warsaw, 8-12 September 1980.*, volume 86 of *Astrophysics and Space Science Library*, pages 375–384, Dordrecht, Holland, 1981. Reidel Publishing.

Lunar Geodesy and Cartography Working Group. Recommendations for Formatting Large Lunar Datasets, May 2008.

Lunar Reconnaissance Orbiter Project. A Standardized Lunar Coordinate System for the Lunar Reconnaissance Orbiter, May 2008.

C. Ma, E. F. Arias, T. M. Eubanks, A. L. Fey, A. M. Gontier, C. S. Jacobs, O. J. Sovers, B. A. Archinal, and P. Charlot. The International Celestial Reference Frame As Realized by Very Long Baseline Interferometry. *Astronomical Journal*, 116:516–546, July 1998.

Nautical Almanac Office, U. S. Naval Observatory, and H. M. Nautical Alamanac Office, Royal Greenwich Observatory. The Introduction of the Improved IAU System of Astronomical Constants, Time Scales and Reference Frame Into the Astronomical Almanac. In *Supplement to the Astronomical Almanac 1984*. U. S. Government Printing Office, Washington and Her Majesty's Stationary Office, London, 1983.

Navigation and Ancillary Information Facility (NAIF). *SPICE Archive Preparation Guide (PDS4 Version)*. Jet Propulsion Laboratory, Pasadena, CA, TBD.

Judith J. Newton. *Guide on Data Entity Naming Conventions*. National Bureau of Standards (NBS) Special Publication 500-149, October 1987.

Planetary Cartography Working Group. *Planetary Cartography 1993-2003*. NASA, Washington, D.C., 1993.

Planetary Data System. *Archive Preparation Guide (APG), Version 1.4*. JPL D-31224. Jet Propulsion Laboratory, Pasadena, CA, April 2010a.

Planetary Data System. *Data Providers' Handbook - Archiving Guide to the PDS4 Data Standards, Version 0.22.3*. Jet Propulsion Laboratory, Pasadena, CA, October 2010b.

Planetary Data System. *Proposer's Archiving Guide (PAG), Version 1.4*. JPL D-26359. Jet Propulsion Laboratory, Pasadena, CA, March 2010c.

Planetary Data System. *PDS4 Data Dictionary - Unabridged - V.0.3.0.0.e*. Jet Propulsion Laboratory, Pasadena, CA, March 2011a.

Planetary Data System. *Introduction to the PDS4 Document Set*. Jet Propulsion Laboratory, Pasadena, CA, March 2011b.

Planetary Data System. *PDS4 Data Standards Concepts*. Jet Propulsion Laboratory, Pasadena, CA, March 2011c.

Planetary Data System. *PDS Policies and Procedures*. Jet Propulsion Laboratory, Pasadena, CA, TBD.

C. T. Russell. Geophysical Coordinate Transformations. *Cosmic Electrodynamics*, 2:184–196, 1971.

P. K. Seidelmann, B. Guinot, and L. E. Dogget. *Time*, chapter 2. Explanatory Supplement to the Astronomical Almanac. U. S. Naval Observatory, University Science Books, Mill Valley, CA, 1992.

P. K. Seidelmann, V. K. Abalakin, M. Bursa, M. E. Davies, C. De Bergh, J. H. Lieske, J. Oberst, J. L. Simon, E. M. Standish, P. Stooke, and P. C. Thomas. Report of the IAU/IAG Working Group on Cartographic Coordinates and Rotational Elements of the Planets and Satellites: 2000. *Celestial Mechanics and Dynamical Astronomy*, 82(1):83–110, January 2002. doi: 10.1023/A: 1013939327465.

P. K. Seidelmann, B. A. Archinal, M. F. A'Hearn, D. P. Cruikshank, J. L. Hilton, H. U. Keller, J. Oberst, J. L. Simon, P. Stooke, D. J. Tholen, and P. C. Thomas. Report of the IAU/IAG Working Group on Cartographic Coordinates and Rotational Elements: 2003. *Celestial Mechanics and Dynamical Astronomy*, 91(3-4):203–215, March 2005. doi: 10.1007/s10569-004-3115-4.

P. K. Seidelmann, B. A. Archinal, M. F. A'Hearn, A. Conrad, G. J. Consolmagno, D. Hestroffer, J. L. Hilton, G. A. Krasinsky, G. Neumann, J. Oberst, P. Stooke, E. F. Tedesco, D. J. Tholen, P. C. Thomas, and I. P. Williams. Report of the IAU/IAG Working Group on cartographic coordinates and rotational elements: 2006. *Celestial Mechanics and Dynamical Astronomy*, 98(3):155–180, July 2007. doi: 10.1007/s10569-007-9072-y.

R. Vaughan. *Mars Pathfinder Project Planetary Constants and Models*. JPL D-12947, PF-100-PMC-01. Jet Propulsion Laboratory, December 1995.

Index