Filling Out the Geometry Dictionary Classes

The classes in this dictionary define the observational geometry relevant to the data objects presented in the product.

If you are not already well-versed in geometry as it relates to observational meta-data, the Geometry Discipline Dictionary can be fairly intimidating. This page is here to help orient you to terminology and structure so you can navigate the Geometry Dictionary more easily. It is written for the non-specialist.

Terminology and Concepts

To start with, here are some key terms and concepts you'll find throughout the Geometry Dictionary.

Central Body
Sometimes the field of view contains one thing that is orbiting around another thing. When the smaller thing is the target of interest, *Central Body* is the term used to refer to the larger thing that the target is orbiting.

Clock Angle
Clock angles are a way of specifying the direction of something, like North or the Sun, from the center of an image. They are measured clockwise from "up" - a vertical line running from the middle of the image to the center of the top edge. In order to correctly interpret a clock angle, you *must* be displaying the image correctly. "Correctly", in the PDS case, means according to the explicit display directions included in the *<geom:Image_Display_Geometry>* class.

Coordinate System *vs* Reference Frame
In the geometry dictionary, these two terms are distinct, and they are used here the same way they are used in the documentation for the NAIF SPICE toolkit (the software most missions and PDS nodes are using to calculate geometric values for PDS4 labels). A *Reference Frame* is defined by three orthogonal axes and an orientation in space; a *Coordinate System* is the result of fixing a *Reference Frame* to a specific origin. So (loosely speaking), "celestial coordinates" is a *reference frame*, "celestial coordinates centered on the Sun" is a *coordinate system*.

Coordinate System *vs* Coordinate Space
For data obtained by orbiter or flyby spacecraft, labels will generally only need to refer to one or two reference frames, and one or two coordinate systems. But for landed missions, and in particular rovers, the situation is much more complicated, as coordinate systems are needed to describe in detail the location and orientation of individual pieces of hardware (sampling arms, drills, etc.) that move, as well as the lander/rover itself. New coordinate systems are defined pretty much at every stop, with each of these bootstrapping off of the previously defined coordinate system. Typically, an index is used to identify the individual coordinate system steps in these sequences. The term *coordinate space* is used in classes that refer to these indexed sequences of coordinate systems.

North Pole
For solar system objects, the IAU defines the "north" pole of a planet as the pole that is on the same side of the invariant plane of the solar system as the Earth's north pole. For larger planets and satellites, the poles are commonly referred to as "north" and "south" and you will see classes with "north pole" in their name or description. Only use these classes for things that have a well-defined "north". For small bodies, which tend to tumble and have complicated rotational states, "north" is not an applicable concept. For these cases, use classes with the "positive pole" notation, instead.

Object

In the Geometry Dictionary, the word "object" is used strictly to refer to the digital data object in the data file(s) referenced by the label. It will *not* be used to refer to physical objects like planets, comets, rings, dust, spacecraft, instruments, or any other thing that might be found in the field of view or involved in actually recording the observation. So if you are writing a label for an image of Titan, "object" will always mean the image, not Titan.

Observer

The designation of *observer* is used in the Geometry dictionary as a general way of identifying the origin of a vector or other quantity with a sense of direction. So, for example, in the vector class <*Vector_Cartesian_Position_Sun_to_Spacecraft*>, which defines a position vector from the center of the Sun to the center of the spacecraft, the Sun is in the *observer* role and the spacecraft is in the *target* role.

Positive Pole

This term is used to indicate the direction of positive angular momentum (according to the "right-hand rule") for a body that is rotating. The positive pole may or may not be considered the "north" pole, depending on a number of things including the overall rotational state of the body. Typically, asteroids and comets will be described in terms of their "positive" poles, while larger planets and their satellites will be described in terms of their "north" poles.

Specific *vs* Generic

Certain geometric quantities recur in data from widely varying sources, and are of particular interest to users, researchers, and analysts. In order to make those data quickly recognizable to both humans and programs, many classes and attributes are defined with "specific" names - names that indicate the observer and/or target. For example, <*spacecraft_target_center_distance*> is the attribute that contains the scalar distance between the spacecraft and the center of the target. "Generic" classes, on the other hand, allow a data preparer to define distances between arbitrary points, but require an explicit specification of start and end point. You should use the defined specific classes and attributes wherever they are applicable to support correlative studies across data sources, and use the generic classes only when there is no specific alternative.

SPICE

"SPICE" is an acronym representing a system of data files (called "kernels") and the associated software toolkit that supports them. SPICE is widely used by missions and end-users to calculate observational geometry for spacecraft data, and is useful for ground-based observers as well. When you see this acronym used in a class or attribute name it indicates that the associated concepts are mapped directly from the SPICE toolkit and documentation. See the NAIF website for details: http://naif.jpl.nasa.gov/naif. In particular, the **_spice_name* attributes should contain the NAIF-issued SPICE identifier for the thing (spacecraft, instrument, reference frame, etc.) being referenced, if any/known.

Target

The term *target* is used in this dictionary primarily to refer to the thing of interest in the enclosing geometry class. It may or may not be the same thing named in the <*Target_Identification*> area elsewhere in your label. In fact, if you have multiple things of interest in your field of view, you may well have distinct <*Geometry*> classes in your label, each one of which provides geometry for one specific thing of interest ("target"). So *target* will be defined locally in your Geometry classes and subclasses. The <*Geometry_Target_Identification*> class is used throughout the Geometry dictionary structures to identify a thing of interest as the local *target* of reference.

"*Target*" is also employed in a much broader sense when generic classes are used to define vectors that are outside of the specific classes provided in this dictionary. In these cases *observer* indicates the origin or starting point, while *target* indicates the destination or ending point. Context should make the distinction in meaning clear.

<Geometry>

REQUIRED

This is the wrapper class for the Geometry Dictionary. It must be used whenever you want or need to include geometry information in a label. Any mission using NAIF software to calculate geometry should include the <*SPICE_Kernel_Files*> class; you'll likely need to include the <*Image_Display_Geometry*> class for anything image-like; and one of either <*Geometry_Orbiter*> or <*Geometry_Lander*> is required.

> **Note:** *As of this writing, spacecraft-oriented flyby and orbiter geometry classes are provided in this dictionary, as well as some classes that support ground-based geometry values. If you have other types of geometry to document in your labels, or don't find what you expect, please do notify your PDS node consultant and ask for the preferred method of recording this information.*

These classes are necessarily fairly complex, so the descriptions have been broken into separate pages corresponding to the major subclasses. You may, if you need to, have more than one <*Geometry*> class in your label.

<SPICE_Kernel_Files>

OPTIONAL

This class is used to cite the SPICE kernel files used in calculating the associated geometric values (for missions using the NAIF *SPICE* Toolkit). Your PDS node will likely require you to use this class if the NAIF software was used. It may appear only once in each <*Geometry*> class.

- Filling out the SPICE_Kernel_Files Class

<Expanded_Geometry>

OPTIONAL

This class is used when geometry is supplied as a separate data object or product either in addition to or in place of specific values in the other <*Geometry*> subclasses. So if, for example, you have geometry calculated for each plane of an image cube as a separate table object, use this class to identify the table that holds the related geometry.

- Filling out the Expanded_Geometry Class

<Image_Display_Geometry>

OPTIONAL

This class is used to provide orientation to fundamental directions (North, East, etc.) with respect to a declared display orientation. If your data object is an image or image-like thing (a 2D spectrum, say), you will be required to include this class in your geometry info. You may repeat this class if you have more than one image-like thing to describe in your label.

- Filling out the Image_Display_Geometry Class

<Geometry_Orbiter>

OPTIONAL

This class provides detailed geometry (positions, orientations, velocities, and so on) appropriate for orbiting and fly-by spacecraft. One of this class or *Geometry_Lander* is required; you may have both. You may also repeat this class if appropriate.

- Filling out the Geometry_Orbiter Class

<Geometry_Lander>

OPTIONAL

This class provides geometric information relevant to a landed spacecraft, including rovers. One of either this class or *Geometry_Orbiter* is required; you may have both. You may also repeat this class if appropriate.

This class does not yet appear to be ready for use. If you have lander geometry to describe, including instrument articulation angles and motion tracking information, please contact your PDS node consultant for assistance.

- *Filling out the Geometry_Lander Class* - still in development