Filling Out the Spectral Dictionary Classes

The classes in this dictionary describe the spectral characteristics of the data objects presented in the product.

The *Spectral Discipline Dictionary* is used to provide metadata that describes how a light spectrum is presented in a data object. This guide defines the terminology used in the Spectral Dictionary and provides a brief overview of the various methods available for defining the relevant metadata in your product labels.

The reserved abbreviation for the Spectral Dictionary namespace is **sp:**.

Preliminaries

First off, note that the Spectral Dictionary is designed only for light spectra. More specifically, it is designed to describe wavelength, frequency, and wavenumber spectra. It does not describe mass spectra, time-of-flight spectra, or any spectra that are not the direct result of gathering photons.

Further, the Spectral Dictionary can only handle spectra that are in one of the following data structures:

- A table in which each row of the table contains a complete spectrum ("Tabulated Spectra")
- A table in which each row of the table contains a single point in a spectrum, and the entire table comprises a single spectrum ("1D Spectrum")
- An image array (specifically, an *<Array_2D_Spectrum>* object) , where either the axes of the array are aligned with the spectral dimensions, or the array contains an image which presents a recorded spectrum
- A stack of images (*<Array_3D_Spectrum>*), where each plane is a 2D spectrum (as in the previous bullet)

If you have spectral data that doesn't fit those models, talk to your PDS Node consultant about the best course of action.

Terminology and Concepts

The bulk of the Spectral Dictionary is concerned with defining the quantization of the spectrum - in other words, what range of wavelengths (or frequencies, or wavenumbers, as appropriate) is represented by each pixel (or row, or column) along the spectral dimension of the data structure.

Spectral "Bins"

The term *binning* is used to refer to the spectral quantization throughout the Spectral Dictionary, irrespective of data structure. The *bin* corresponds to the spectral range associate with a single point:

- In tabulated spectra, individual fields represent *bins*.
- In 1D spectra, each row is a *bin*.
- In 2D spectra, each pixel along the spectral axis is a *bin*.
- In 3D spectra, each pixel along the spectral axis (which corresponds to a plane in the perpendicular axes) is a *bin*

These data structure bins may or may not correspond to the bins of the original detector, either because of re-binning of digital data or digitization of analog data. Regardless, the relationship between detector bins and data structure bins should be well-documented, and in every case when the Spectral Dictionary refers to "bins", it is the data structure bins that are intended.

Field of View (FOV)

In the Spectral Dictionary, field of view or *FOV* is the generalized terminology used to refer to what is traditionally thought of as the slit or aperture that governs the light path producing the spectrum, which is then recorded by the detector. "Slits" can be complex in structure, so the metadata descriptions are intended to provide good descriptions of traditional slits and apertures, and a heads-up and a reference to details for non-traditional "slits".

If you are working with a complex slit/aperture, you will need a reference (ideally, a reference to something also in the PDS4 archive) to a document that describes it in some detail. You may have to write this document if none exists; including pictures and diagrams is greatly appreciated by users.

Look-up

It is common for the values defining the spectral bins to be located either in their own data object within the product, or in another product entirely. When a spectral bin parameter is not explicitly stated in the label metadata (and must instead be read from a data file), the term "lookup" is used in the class name even if the data to be "looked up" are in the same data file as the observational data. Details of the look-up class will make it clear where the data reside; PDS4 LID references will be used, when needed, to identify other archive products containing bin parameters.

Label Structures

<Spectral_Characteristics>

The Spectral Dictionary is a discipline dictionary. As such, its classes are included in the <*Discipline_Area*> of your label. You *must* use the <*Spectral_Characteristics*> class in order to use any of the other classes and attributes in the Spectral Dictionary. In general, you will only have one <*sp:Spectral_Characteristics*> class in your label even for data structures (like tabulated spectra and spectral cubes) that contain multiple spectra. For very complex cases, it is (just barely) conceivable that you might need more than one <*sp:Spectral_Characteristics*> class in a single label, but check with your PDS Node consultant first.

(For this and the rest of this discussion, I'm going to elide the "sp:" prefix in the interest of clarity for explanations, but bear in mind that all the attributes and classes described below must *be placed in the Spectral namespace in PDS4 labels. If that sounds like gibberish to you, see the* Using Local Dictionaries *page.)*

At the top of the *Spectral_Characteristics* class is a <*pds:Local_Internal_Reference*> class that identifies the data object containing the spectral data described by the rest of the class. Following that, the *Spectral_Characteristics* class has four major sections (in order):

1. **Spectrum type parameters:** These attributes at the top of the *Spectral_Characteristics* class identify the data structure (tabulated, 1D, 2D, etc.), the spectral bin type (wavelength, frequency, or wavenumber), and for tabulated spectra the location of the recorded values for each spectral bin.
2. **<Observation_Parameters>:** This class contains attributes related to: how the data were collected (integration time, number of exposures); the detector (resolution limit); and calibration, where appropriate (calibration star names).
3. **<Field_of_View>:** This class is for describing the field of view. There are subclasses for the simple cases (circular and rectangular), and a more general class for complex FOVs. If you have a complex FOV, this is where you provide a test description for users and a link to the more detailed documentation.

4. **<Bin_Description>:** This is the class where you define how the spectral data is binned. The first field is a text description in which you should describe *how* the bin parameters were determined.

The <Bin_Description> Subclass

Bin description being the primary function of the Spectral Dictionary, the <*Bin_Description*> class provides three different methods for specifying the bin centers and widths: explicit definition; calculation; and look-up.

<Axis_Bin_Set> - Explicit Definition

The <*Axis_Bin_Set*> class and its subclasses are used to provide specific parameters for each bin along the spectral dimension. The bins must be labeled with a sequence number, and at a minimum the center and bin width must be provided. In this case you may also identify a specific filter, grating setting, or detector associated with the bin.

<Uniformly_Sampled> - Calculation

The <*Uniformly_Sampled*> class lets you define your bins mathematically, by indicating the first and last bin centers and defining the intervening interval. In this case bin width is assumed to be uniform as well, and you specify it as a constant.

<Spectral_Lookup> - Look-up

The <*Spectral_Lookup*> class is used to direct a user (or program) to a file location where the bin center and width data can be programmatically read. The parameters can be referenced as fields within a spectral table, as data objects within the same file via a <*pds:Local_Internal_Reference*> class, or as data objects in another PDS archive product via a <*pds:Internal_Reference*> class.

Unknown Bin Widths

As of Spectral Dictionary version 1.2.0.0:

- In those rare cases where the bin widths are not known and cannot be determined, the bin width attributes within the *Axis_Bin_Set* and *Uniformly_Sampled* options can be declared to be nil with a reason of "unknown". Examples of how to do this are shown further down on this page under headings <Uniformly_Sampled_[unit]> and <Axis_Ben_Set_[unit]>
- For the *Spectral_Lookup* case where no widths are available to look up, a *Bin_Width_Constant* option has been added to the <*Spectral_Lookup*> class that can either be used to indicate a constant nominal bin width, or to explicitly declare the bin widths as nil (and unknown).

Usage Notes

The *Spectral Discipline Dictionary* is now posted on GitHub, where users can see the source file and build products for all versions of the dictionary, as well as using the "Issues" feature of the GitHub interface to report problems or request changes (you can also contact the steward directly). The GitHub repository is https://github.com/pds-data-dictionaries/ldd-spectral. You do not need to have a GitHub account to simply download the schema files.

In order to accommodate data designers at various stages of preparation, each version of the Spectral Dictionary is built against multiple versions of the core PDS4 Information Model (IM). The content is the same in each case - only the core namespace reference has changed. Select the version of the Spectral Dictionary that matches the version of the core dictionary that you are using. If you have questions, please contact the steward for this dictionary, or your local PDS node.

Steward

The steward for this dictionary is Anne Raugh, at the Small Bodies Node, University of Maryland.

<Spectral_Characteristics>

REQUIRED

This is the wrapper class for spectral characteristics. Repeat it once for each spectral-type object you have in your observational product. The Spectral Discipline Dictionary can currently accommodate spectra measured in wavelength, wavenumber, and frequency that are presented as either image cubes, images, or tables (either linear spectra where each row presents one point in the spectrum, or tabulated spectra where each row contains an entire spectrum).

If you have a different type of spectrum, contact Anne Raugh at the Small Bodies Node at the University of Maryland for assistance.

<pds:Local_Internal_Reference>

REQUIRED

Use this class to identify the spectral-type data object in this product for which this class provides the characteristics. Typically, an observational product will contain only a single spectral object (though it might be a spectral cube). This class is required even in that case.

Note that this class and its elements are in the *pds:* core namespace, rather than the Spectral Dictionary namespace, so you will need to make the appropriate change of namespace for this class. See "Schema Referencing in PDS4 Labels: Namespace References" for details.

<pds:comment>

OPTIONAL

This is a free-format text field that you can use if there is something about the relationship between the referenced spectrum and its characteristics that requires additional explanation.

<pds:local_identifier_reference>

REQUIRED

This attribute must contain a *<local_identifier>* string of a spectral data object described elsewhere in this label. Spelling and capitalization count.

<pds:local_reference_type>

REQUIRED

This attribute must have one of these values:

- **spectral_characteristics_to_array_object**
- **spectral_characteristics_to_table_object**

The value you use must correspond to the type of the data object you are linking to. So if your *local_identifier* points to an *Array_2D_Spectrum* then the *local_reference_type* must be **spectral_characteristics_to_array_object**. If your *local_identifier* points to a linear spectrum in a *Table_Character* object, then the *local_reference_type* must be **spectral_characteristics_to_table_object**.

<description>

OPTIONAL

This is a free-format text field you can use to provide any additional explanation or caveats about the information following that would be helpful to an end user.

<spectrum_format>

REQUIRED

This attribute indicates the dimensionality of the spectrum, and must have one of the values described, following. When the *<Primary_Result_Summary>* is present, the *<Science_Facets>/<discipline>* value should be **Spectroscopy** and the *<Science_Facets>/<facet1>* value should correspond to the value here as indicated:

1D
The spectrum is linear; presented as a table object where each row contains the information about one point in the spectrum. *<facet1>* should have a value of **Linear**.

2D
The spectrum is in an *Array_2D_Spectrum* object. In this case *<facet1>* might contain either *2D* or *Spectral Image*, depending on whether or not, respectively, the image axes align with the spectral axes.

3D
The spectrum is in an *Array_3D_Spectrum* object. The value of *<facet1>* should be **Spectral Cube**.

Tabulated-Flat
Multiple spectra are presented as a table in which each row contains a single spectrum *and* the individual spectral points and related parameters (bin center, bin width, etc.) are each defined by unique *Field* definitions. In other words, there are no *Group_Field* groups within the table definition. In this case, all fields containing spectral data or bin parameters ***must*** have unique *<field_number>* attributes. Bin parameters may also be referenced in a different data object or product, if appropriate. In this case, *<facet1>* should be **Tabulated**.

Tabulated-Parameter Groups
Multiple spectra are presented as a table in which each row contains a single spectrum, but *Group_Field* classes are used to define groups corresponding to the spectral values, bin centers, and bin widths. That is, the spectral values are stored contiguously with the record, the bin centers are stored contiguously within the record, etc. In this case, the *Group_Field* classes containing the spectral values, bin centers, and bin widths (when present) *must* contain only a single *Field* that is repeated. Bin parameters may also be referenced in a different data object or product, if appropriate. In this case, *<facet1>* should be **Tabulated**.

Tabulated-Point Group
Multiple spectra are presented as a table in which each row contains a single spectrum, *and* a single *Group_Field* class is used to define the value and all related parameters at a single spectral point. In this case, the same *Group_Field* class must contain the spectral value, bin center, and bin width (when present) for each point in the spectrum; the number of times that group is repeated is the number of points in a single spectrum. Bin parameters may also be referenced in a different data object or product, if appropriate. In this case, *<facet1>* should be **Tabulated**.

Note that the difference between **Tabulated-Parameter Groups** and **Tabulated-Point Group** is nil in terms of metadata, but the specific designation is potentially useful to a program trying to access the data in the most efficient manner possible, as well as helping a human reader to understand what can be a complex and extensive data structure description in the label.

> The relationship between this value and the (optional) *<facet1>* in *Primary_Result_Summary* is not currently validated. Type carefully.

<value_field_name>

REQUIRED for **1D**, **Tabulated-Parameter Groups**, and **Tabulated-Point Group** format spectra; otherwise *PROHIBITED*

This attribute is required if <*spectrum_format*> is **1D**, **Tabulated-Parameter Groups**, or **Tabulated-Point Group**, and forbidden otherwise. For these spectra, you must indicate which field in the table or group actually contains the measured spectral value. The value of the <*name*> attribute from the *Field_Character* (or *Field_Binary*) definition is specified as the value of this attribute.

> The existence of the given *name* value in the referenced table is validated, but because field <*name*> values are not defined as identifiers, this sort of reference is somewhat risky and schematic validation can't cover all the bases. So if you are using this attribute, please ensure the following:
>
> 1. Your value field name is unique in the table.
> 2. Your value field name doesn't contain problematic characters, like '>' or '&'.

<value_field_number_list>

REQUIRED for **Tabulated-Flat** spectra; otherwise *PROHIBITED*

This attribute is required if <*spectrum_format*> is *Tabulated-Flat*, and forbidden otherwise. The value should be formatted as a parenthesized, comma-separated list of *field_number* values corresponding to those *Field* classes in the table which contain the spectral value measurements.

Format example:

<sp:value_field_number_list>(3, 7, 11, 15, 19, 23, 27)</value_field_number_list>

The intention is that this list can be parsed by software and mapped to field definitions in order to plot the spectral points.

<spectral_bin_type>

REQUIRED

This attribute indicates the unit of measure for the spectral bin dimensions. It must be one of the following values:

- **frequency**
- **wavelength**
- **wavenumber**

The bin descriptions you provide in the subsequent classes and attributes must correspond to the *spectral_bin_type* specified here. In particular, units of measure must be appropriate.

<Observation_Parameters>

OPTIONAL

This class includes various attributes related to the observing circumstances. While this class is optional, as are its attributes, it should be present and complete in all cases where the attributes are applicable. That would include the great majority of spectral data types. Note that there is a <comment> attribute in this class that can be used to provide caveats for the attribute values (or lack of same) as needed, and in complex cases a reference to a description document can and should be included in the label <Reference_List>. When in doubt, check with your PDS node for guidance.

<number_of_exposures>

OPTIONAL

This attribute indicates the number of individual exposures or distinct integrations that were combined to produce the spectrum in the data object. It should be present even if the value is "1". To omit this attribute implies that "number of exposures" is not an applicable concept for the data presented, which is not a typical situation.

<net_integration_time>

OPTIONAL

The net_integration_time is the total actual integration time contributing to each pixel. In most cases, all pixels are exposed simultaneously and this is equivalent to the net observation time. For instruments that sweep across pixels, rows, or planes, though, this value will differ from the observation time. It is hard to imagine a case where this concept is not applicable, but if you have such a case, please include a <comment> attribute to make that clear.

<resolution_limit_frequency> *or* <resolution_limit_wavelength> *or* <resolution_limit_wavenumber>

OPTIONAL

One of these attributes may be provided to indicate the smallest possible distinguishable spectral interval for the instrument in units corresponding to the *spectral_bin_type* indicated in the parent class. If the spectral resolution is unknown, omit the attribute.

<solar_analog_star_name>

OPTIONAL

The solar_analog_star_name attribute provides a name in a standard format (i.e., one that is recognized by the SIMBAD system) for the solar analog star used to reduce the raw spectral data to the reflectance units presented in this spectrum. You may repeat this attribute if needed for additional stars. *Do not* repeat this attribute to provide alternate names for the same star.

<absolute_calibration_star_name>

OPTIONAL

This attribute provides a name in a standard format (i.e., one that is recognized by the SIMBAD system) for the star used to reduce the raw spectral data to the (non-reflectance) units presented in this spectrum. You may repeat this attribute if needed for additional stars. *Do not* repeat this attribute to provide alternate names for the same star.

<comment>

OPTIONAL

This is an optional free-format text field to provide any additional information or caveats about attributes and values in this class.

<Field_of_View>

REQUIRED

This class describes the field of view of the spectral data object. This is often the aperture or slit used to make the observation, but may be the same as the field of view of the instrument some cases. One of the "*_FOV" sub-classes *must* be present.

<description>

REQUIRED

This is a text description of the slit or aperture. Often this will be a simple sentence. In the case of a complex aperture shape, though, this should be more expansive.

<Circular_FOV>

OPTIONAL

One of Circular_FOV*,* Rectangular_FOV*, and* Complex_FOV *must be specified.*

This class indicates a circular aperture and defines the angular diameter of the field.

<diameter_angle>
REQUIRED

The angle subtended on the sky by the diameter of the aperture. You must specify units of angle.

For those rare cases where the FOV is unknown, or varies from record to record within a set of tabulated spectra, you can use *Circular_FOV* and declare its one parameter to be "nil" with a reason of "unknown" or "inapplicable", respectively. To indicate that a single FOV is not an applicable concept at the whole-product level, for example, you would code this:

```
<diameter_angle unit="arcsec" xsi:nil="true" nilReason="inapplicable"/>
```

The required *<description>* attribute, above, should explain to a reader what is known about field of view and where additional details, if any, can be found.

<Rectangular_FOV>

OPTIONAL

One of Circular_FOV*,* Rectangular_FOV*, and* Complex_FOV *must be specified.*

This class is used to describe a rectangular field of view, as for a traditional simple slit.

<width_angle>
REQUIRED

The width_angle is the angle subtended on the sky by the short dimension of a simple slit. On those extremely rare occasions where this value is unknown, it may be defined as "nil" with a reason of "unknown":

```
<width_angle unit="arcsec" xsi:nil="true" nilReason="unknown"/>
```

In this case, as much as is known about the slit should be included in the *<description>* attribute of the containing class.

Note that a valid unit of measure must be supplied for this attribute even when it is declared to be nil.

<length_angle>
REQUIRED

The length_angle attribute provides the angle subtended on the sky by the long dimension of a simple slit. When it is not known, it can be defined as "nil" (see *width_angle*, above).

<celestial_north_position_angle>
OPTIONAL

The celestial_north_position_angle attribute provides the angle the long axis of the slit makes with respect to celestial north, measured eastward of celestial north in the range 0-180 degrees.

In general, either this or <*body_positive_pole_position_angle*> should be present, or the preceding <*description*> should make it clear why there is no known or relevant "position angle"-type attribute.

<body_positive_pole_position_angle>
OPTIONAL

The body_positive_pole_position_angle attribute provides the angle the long axis of the slit makes with respect to the positive (or "north") pole of the target body, measure clockwise from the pole in the range 0-180 degrees.

In general, either this or <*celestial_north_position_angle*> should be present, or the preceding <*description*> should make it clear why there is no known or relevant "position angle"-type attribute.

<Complex_FOV>

OPTIONAL

One of Circular_FOV*,* Rectangular_FOV*, and* Complex_FOV *must be specified.*

This class is used to provide some level of description for a complex aperture. As much as makes sense to include should be included.

<width_angle>
OPTIONAL

If there is an applicable "width" for the aperture, it can be specified here as a subtended angle. This value can be repeated, if that makes sense.

<length_angle>
OPTIONAL

If there is an applicable "length" for the aperture, it can be specified here as a subtended angle. This value can be repeated, if that makes sense.

<celestial_north_position_angle>
OPTIONAL

The celestial_north_position_angle attribute provides the angle the long axis of the slit makes with respect to celestial north, measured eastward of celestial north in the range 0-180 degrees.

In general, either this or <*body_positive_pole_position_angle*> should be present, or the preceding <*description*> should make it clear why there is no known or relevant "position angle"-type attribute.

<body_positive_pole_position_angle>
OPTIONAL

The body_positive_pole_position_angle attribute provides the angle the long axis of the slit makes with respect to the positive (or "north") pole of the target body, measure clockwise from the pole in the range 0-180 degrees.

In general, either this or <*celestial_north_position_angle*> should be present, or the preceding <*description*> should make it clear why there is no known or relevant "position angle"-type attribute.

<pds:Internal_Reference>

OPTIONAL

Use this class to link to a document that provides a detailed description of the slit or field of view. This could be a text description, a graphic representation, or a larger document (an instrument SIS for example) that includes this information. Use the *pds:comment* field to indicate a specific section or page if you are referencing a part of a larger document. This class may be repeated, if needed.

While technically optional, it is hard to imagine a data product with a complex aperture passing review without a good description of the aperture shape and size being included.

Note that, like the *<pds:Local_Internal_Reference>* above, this class and its attributes are in the *pds:* namespace.

<pds:lid_reference> or *<pds:lidvid_reference>*
REQUIRED

Either a logical identifier without a version number (a "LID"), or a logical identifier with a version number (a "LIDVID"), must be provided for the product you wish to reference. You may not provide both, or repeat this attribute.

<pds:reference_type>
REQUIRED

This attribute must have the value **spectral_characteristics_to_document**.

<pds:comment>
OPTIONAL

This is a free-format text field in which you can provide additional explanation for why you are referencing the previously mentioned product. If the relevant information is actually in a subsection of the document (rather than comprising the whole document) a section number or page range reference would be an excellent thing to include here.

<Bin_Description>

REQUIRED

This class provides three different ways to define the spectral bin characteristics: implicitly, as a uniformly sampled set; explicitly, as individual bin descriptions provided in the label; or externally, in another data object in this product or even in a separate product altogether. You are required to provide at least a minimal text description of the bin profiles (how the bin widths were determined), and you must use one of the three methods (*Uniformly_Sampled_[unit]*, *Axis_Bin_Set_[unit]*, or *Spectral_Lookup*) to describe the spectral bins. Note that the "Axis" in *Axis_Bin_Set* can also refer to rows and columns (fields) in tabular data.

To pass review, most data products will require a fairly precise description of the spectral binning. Check with your PDS node consultant if you have any questions or doubts.

<bin_profile_description>

REQUIRED

This is a free-format text field for describing the shape of the spectral bins and how their "widths", defined in the following attributes, were determined. It is required to be present and non-null. It does not need to be exhaustive - it just needs to provide a user with the essential details needed to understand the spectrum. If the situation is complicated but well-documented, the *<pds:Reference_List>* class in the upper part of the product label should include an explicit reference to the explanatory document elsewhere in the archive or in the literature. If the situation is complicated but not well-documented, please consider writing that document, archiving it, and then referencing it.

<Uniformly_Sampled_[unit]>

OPTIONAL

This class occurs in three flavors, one for each possible value of <*spectral_bin_type*>. Use the version of the class and its attributes corresponding to that value - indicated by the string "[unit]" in the descriptions that follow. The class provides the parameters needed to calculate the bin dimensions, in the specified unit, at each point along the spectral dimension of the data object.

<axis_name>

REQUIRED

This attribute identifies which dimension of the data object corresponds to the spectral dimension in the data. For array-type objects, the value here must be the same as that of an <*axis_name*> attribute from the referenced data object. For *1D* spectra, this must contain the value **Row**; for all flavors of *Tabulated* spectra, this must contain the value *Field*.

Note: *Prior to version 1.2.0.0 of the* Spectral Dictionary*, this attribute was required only for array-type data objects.*

<sampling_interval_[unit]>

REQUIRED

This attribute provides the spacing from bin center to bin center in the indicated unit of measure.

<sampling_scale>

REQUIRED

This attribute indicates the type of scale used for sampling. It must have one of two values: **Linear** or **Logarithmic**. In the latter case, you must specify the base of the log using <*sampling_base*>.

<sampling_base>

see description

This attribute is required to be present if the preceding *sampling_scale* attribute is **Logarithmic**, otherwise it is forbidden. The value should be a number, so for the natural log a numeric approximation of *e* to an appropriate precision (e.g., "2.7182818") should be used.

<bin_width_[unit]>

REQUIRED

The width of one spectral bin in the appropriate units. How "width" is determined should have been indicated by the required *bin_width_desc* at the top of the *Uniformly_Sampled_[unit]* class.

If the bin width is unknown, this attribute can be nilled with a reason of "unknown". In the case of a frequency spectrum, for example, that would look like this:

```
<bin_width_frequency unit="Hz" xsi:nil="true" nilReason="unknown"/>
```

Note that it is required to indicate some appropriate value for unit, even when the value is declared to be nil.

Note: *Prior to Spectral Dictionary version 1.2.0.0,* bin_width_[unit] *attributes were not nillable.*

<first_center_[unit]>

REQUIRED

This attribute provides the value, in the corresponding unit, at the center of the first spectral bin. It must be numerically less than the *last_center_[unit]* value, following.

> Note that there is a validation check to make sure that the value of the "first_center" is less than that of the "last_center", but this check is only applied to the numeric values - it ignores any possible change in units of measure.
>
> On the whole, it would be better for both users and programs if you use the same unit of measure for all spectral parameters values in this class.

<last_center_[unit]>
REQUIRED

This attributes provides the value, in the corresponding unit, at the center of the last spectral bin. It must be numerically greater than the *first_center_[unit]* value, preceding.

<comment>
OPTIONAL

This free-format text field is provided as a place to note any peculiarities or caveats not otherwise noted about the bins and their definition.

<Axis_Bin_Set_[unit]>

OPTIONAL

This is the class to use if your spectral bins are not uniform but it is reasonable to provide individual bin parameters in the product label. This is often the case, for example, if your spectrum is a composite of points obtained by discrete detectors or observations through a series of filters, or has been resampled or selected from another (hopefully archival) spectrum product. If your bins are so numerous or complex that describing them individually in the label is not practical, you should probably be looking into the <*Spectral_Lookup*> class, following.

The *Axis_Bin_Set* class occurs in three flavors, one for each possible value of <*spectral_bin_type*>. Use the version of the class and its attributes corresponding to that value - indicated by the string "[unit]" in the descriptions that follow. This class provides parameters for defining each individual bin along the spectral axis. For array-type objects, the spectral axis is one of the array axes. For *1D*-type spectra, the spectral axis corresponds to the row dimension; for all *Tabulated*-type spectra, the spectral dimension corresponds to the column (field) dimension.

<axis_name>
REQUIRED

This attribute identifies which dimension of the data object corresponds to the spectral dimension in the data. For array-type objects, the value here must be the same as that of an <*axis_name*> attribute from the referenced data object. For *1D* spectra, this must contain the value **Row**; for all flavors of *Tabulated* spectra, this must contain the value *Field*.

Note:""" *Prior to version 1.2.0.0 of the* Spectral Dictionary*, this attribute was required only for array-type data objects.*

<Bin_[unit]>
REQUIRED

This sub-class is repeated once for each spectral bin along the named axis. It defines the dimensions of a single bin in the corresponding spectral units.

Note that it is difficult, if not impossible, to validate schematically that bins are being described in their physical sequence along the axis, but this should certainly be the case and reviewers and end-users will expect it.

<bin_sequence_number>
REQUIRED

Detailed bin descriptions make for a long and difficult-to-read string of XML. This attribute is used to help map bin descriptions to sequential positions along the spectral axis. It should start with "1" for the first bin and increment for each successive bin described.

<center_[unit]>
REQUIRED

This attribute contains the spectral value at the center of the bin being described.

<bin_width_[unit]>
REQUIRED

This is the width of the particular bin. How the bin width was measured or otherwise determined should be described at least briefly in the *<bin_width_desc>* attribute at the top of the *Bin_Description* class.

If the bin width is unknown, this can be specified by setting this attribute to nil with a reason of "unknown". For a wavelength spectrum, for example, that would look like this:

```
<bin_width_wavelength unit="nm" xsi:nil="true" nilReason="unknown"/>
```

Note: *For versions of the Spectral Dictionary prior to 1.2.0.0, the* bin_width_[unit] *attributes are not nillable.*

<detector_number>
OPTIONAL

If you spectrum is a composite from several detectors, this attribute can be used to provide an identifying sequence number. Documentation elsewhere in the label or in the archive submission should map the sequence numbers here to specific detectors.

<grating_position>
OPTIONAL

Use this attribute to indicate that the spectral bin corresponds to a specific grating position. Typically, grating positions begin at zero and increase with wavelength (or equivalent). Documentation elsewhere in the label or in the archive submission should decode the grating position as needed for users.

<original_bin_number>
FORBIDDEN

This attribute was removed from the Spectral Dictionary as of version 1.2.0.0. It was a hold-over from PDS3 that proved to be mission-specific, and thus not appropriate for a discipline dictionary. If you have a different opinion about this, please contact the Spectral Dictionary Steward with examples.

<Filter>
OPTIONAL

If your spectral bin corresponds to the bandpass of a specific filter, use this class to identify the filter. Filters specific to missions, projects, or observatories should be described in documentation also included in the PDS archive (and referenced in the *Reference_List* section of this product label). Standard ground-based filter sets should be identified using the full, formal name wherever possible (i.e., "Johnson V", not just "V").

<filter_name>
OPTIONAL

Standard ground-based filter sets should be identified using the full, formal name wherever possible (i.e., "Johnson V", not just "V" or "V(J)" or similar). Other filters should be referenced by the formal name defined in project or instrument documentation. If no formal documentation of the name exists, colloquial names can be used as a last resort.

<filter_number>
OPTIONAL

If there is a list of filters identified by number somewhere in the referenced documentation for the data, then here's the place to put the number of a specific filter (typically a filter wheel position, for example). If you are making use of this as part of a mission data submission, please consider including validation tests in your mission dictionary that will ensure that *filter_number* and *filter_name* (for example, and among others) have the correct correspondence.

<comment>
OPTIONAL

This text field is for caveats and explanations as needed to clarify the filter situation.

<Spectral_Lookup>

OPTIONAL

Often the spectral bin parameters for 2D and 3D spectral arrays are supplied as parallel arrays with explicit values for each pixel. These parameter arrays may be included in the same data product, or as one or more separate products. It is also common for spectra provided as tables to include bin parameters as additional fields in the table object. In either event, this class is provided to link to the data objects that provide the essential bin parameters.

<Bin_Center_Lookup>
REQUIRED

This class provides a link to the values at the center of each spectral bin. Note that exactly one method for specifying the location of the bin centers *must* be provided.

<bin_center_field_name>
OPTIONAL

This attribute is used with spectra that have a <*spectrum_format*> value of **1D**, **Tabulated-Parameter Groups**, or **Tabulated-Point Group** to identify the *Field* class that defines the centers of the spectral bins. It must correspond to the value of the <*name*> attribute in a *Field* definition (**not** the *Field_Group* definition, if any).

> This is kind of a weird case, and it is not currently possible to schematically validate everything that should be validated. So if you are going to use this method for specifying bin centers for 1D and tabulated spectra, please ensure the following:
>
> 1. Your bin center field name is unique.
> 2. Your bin center field name doesn't contain problematic characters, like '>' or '&'.
> 3. The bin center field has a <*unit*> attribute.
> 4. That <*unit*> attribute in that field is consistent with the <*spectral_bin_type*> value for the spectrum.

<bin_center_field_number_list>
OPTIONAL

This attribute is used only with spectra that have a <*spectrum_format*> value of **Tabulated-Flat**, and in this case it is necessary that every *Field* class contains a <*field_number*> attribute. The value of *bin_center_field_number_list* is a parenthesized, comma-separated list of numbers used to select *Fields* containing the bin center data by their field numbers.

Format Example:

<bin_center_field_number_list>(4, 9, 14, 19, 24, 29, 34, 39, 44, 49, 54)</
bin_center_field_number_list>

<pds:Local_Internal_Reference>

OPTIONAL

If your spectral object is a 2D or 3D array, *and* the bin centers are provided as a separate array but are included as part of this data product (could be in the same data file as the spectrum or a separate one - as long as they are both described by this label), the you should use this class to indicate the <*local_identifier*> of the array corresponding to the bin centers. You will, of course, have to include a unique <*local_identifier*> attribute in that array object to reference.

This class is filled out similarly to the *pds:Local_Internal_Reference* class mentioned at the top of this page, except that the <*reference_type*> value must be **spectral_characteristics_to_bin_center_values**.

<pds:Internal_Reference>

OPTIONAL

If the bin centers are provided as a data object in a different PDS archive product, use this class to reference that other product. This is filled out similarly to the *pds:Internal_Reference* class detailed for the *Field_of_View* class, except that the <*reference_type*> value must be **spectral_characteristics_to_bin_center_values**.

> Note that it is not possible to schematically validate that the object you reference in another product has the same dimensions as an object in this product. Please confirm that that is the case when you use this form of bin center specification.

<Bin_Width_Constant>

OPTIONAL

One of <*Bin_Width_Constant*> and <*Bin_Width_Lookup*> must be provided.

This class is used for those cases where the bin width is truly constant, where it is not well known and thus approximated by a constant, or where it is unknown. This class should contain exactly one attribute, corresponding to the spectral type. If the bin width is unknown, then that attribute should be declared "nil" with a reason of "unknown", and the <*bin_profile_description*> attribute should provide as much information as is known (or an explanation of why nothing is known) about the spectral bin profiles.

<bin_width_frequency> *or* <bin_width_wavelength> *or* <bin_width_wavenumber>

REQUIRED

Exactly one of these attributes is required. The attribute must correspond to the <*spectral_type*> value. When bin width is unknown, declare the attribute to be nil with a reason of "unknown". For example, in the case of a wavenumber spectrum, it would look like this:

```
        <Bin_Width_Constant>
            <bin_width_wavenumber unit="1/m" xsi:nil="true" nilReason="unknown"/>
        <Bin_Width_Constant>
```

Note that it is necessary to supply appropriate units of measure for this attribute even when the value is declared to be nil.

Note: *Prior to version 1.2.0.0 of the Spectral Dictionary, the* bin_width_[unit] *attributes were not nillable, and the* <Bin_Width_Constant> *class was not available as an option in* <Spectral_Lookup>*. (Although it appeared at a higher level of the* <Spectral_Characteristics> *class, it was not logically usable.)*

<Bin_Width_Lookup>
OPTIONAL

One of <*Bin_Width_Constant*> and <*Bin_Width_Lookup*> must be provided.

This class provides a link to the values defining the width of each spectral bin. Note that exactly one method for specifying the location of the bin widths *must* be provided. If bin widths are unknown, use the preceding <*Bin_Width_Constant*> class to declare them "nil".

<bin_width_field_name>
OPTIONAL

This attribute is used with spectra that have a <*spectrum_format*> value of **1D**, **Tabulated-Parameter Groups**, or **Tabulated-Point Group** to identify the *Field* class that defines the widths of the spectral bins. It must correspond to the value of the <*name*> attribute in a *Field* definition (**not** the *Field_Group* definition, if any).

> This is kind of a weird case, and it is not currently possible to schematically validate everything that should be validated. So if you are going to use this method for specifying bin centers for 1D spectra, please ensure the following:
>
> 1. Your bin width field name is unique.
> 2. Your bin width field name doesn't contain problematic characters, like '>' or '&'.
> 3. The bin width field has a <*unit*> attribute.
> 4. That <*unit*> attribute in that field is consistent with the <*spectral_bin_type*> value for the spectrum.

<bin_width_field_number_list>
OPTIONAL

This attribute is used only with spectra that have a <*spectrum_format*> value of **Tabulated-Flat**, and in this case it is necessary that every *Field* class contains a <*field_number*> attribute. The value of *bin_width_field_number_list* is a parenthesized, comma-separated list of numbers used to select *Fields* containing the bin width data by their field numbers.

Format Example:

<bin_width_field_number_list>(5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55)</bin_width_field_number_list>

<pds:Local_Internal_Reference>
OPTIONAL

If your spectral object is a 2D or 3D array, *and* the bin widths are provided as a separate array but are included as part of this data product (could be in the same data file as the spectrum or a separate one - as long as they are both described by this label), the you should use this class to indicate the <*local_identifier*> of the array corresponding to the bin widths. You will, of course, have to include a unique <*local_identifier*> attribute in that array object to reference.

This class is filled out similarly to the *pds:Local_Internal_Reference* class mentioned at the top of this page, except that the *<reference_type>* value must be **spectral_characteristics_to_bin_width_values**.

<pds:Internal_Reference>
OPTIONAL

If your spectral object is a 2D or 3D array and the bin widths are provided as a data object in a different PDS archive product, use this class to reference that other product. This is filled out similarly to the *pds:Internal_Reference* class detailed for the *Field_of_View* class, except that the *<reference_type>* value must be **spectral_characteristics_to_bin_width_values**.

> Note that it is not possible to schematically validate that the object you reference in another product has the same dimensions as an object in this product. Please confirm that that is the case when you use this form of bin width specification.

<comment>
OPTIONAL

This attribute, which occurs at the bottom of the *<Spectral_Lookup>* class (in case you've lost track), is for any notes or caveats about the lookup process that you'd like to provide for users.