Array 3D Structures

Filling Out the Array 3D Image Data Structure

The ***<Array_3D_Image>*** class is a specialization of the *<Array_3D>* generic class. Use it if your 3D array should be considered as a single image in which the individual pixels comprise a series of values (as in an RGB color image, for example).

For an *Array_3D_Image*, follow the instructions for Filling Out the Array_3D Data Structure in the section below, with the following additional constraints/requirements:

- Use **<Array_3D_Image>** and **</Array_3D_Image>** rather than **<Array_3D>** and **</Array_3D>**
- You absolutely ***must*** include a *<Display_Settings>* class in the *<Discipline_Area>* of the label. This class must include *Display_Direction* for the primary plane (however you define it), and *Movie_Display_Settings* if your third axis is time. If your third axis is wavelength, please consider including a *Color_Display_Settings* class for generating color previews of your image cube. See Filling Out the Display Dictionary Class for more information.

Filling Out the Array 3D Movie Data Structure

The ***<Array_3D_Movie>*** class is a specialization of the *<Array_3D>* generic class. Use it if your 3D array should be considered as a set of 2D images taken in chronological sequence.

For an *Array_3D_Movie*, follow the instructions for Filling Out the Array_3D Data Structure in the section below, with the following additional constraints/requirements:

- Use **<Array_3D_Movie>** and **</Array_3D_Movie>** rather than **<Array_3D>** and **</Array_3D>**
- You absolutely ***must*** include a *<Display_Settings>* class, with a *<Movie_Display_Settings>* subclass, in the *Discipline_Area* of your label to describe the correct way to orient and run the movie. See Filling Out the Display Dictionary Class for more information.

Filling Out the Array 3D Spectrum Data Structure

The ***<Array_3D_Spectrum>*** class is a specialization of the *<Array_3D>* generic class. Use it if your 3D array should be considered as a series of *Array_2D_Spectrum* objects.

For an *Array_3D_Spectrum*, follow the instructions for Filling Out the Array_3D Data Structure in the section below, with the following additional constraints/requirements:

- Use **<Array_3D_Spectrum>** and **</Array_3D_Spectrum>** rather than **<Array_3D>** and **</Array_3D>**
- Include a *<Display_Settings>* class from the Display Discipline Dictionary to define the correct way to orient and display the planes of the cube. Also, consider using a *Color_Display_Settings* class from the same dictionary if it makes sense to, for use in generating preview or thumbnail images of your spectral cube. See Filling Out the Display Dictionary Class for more information.
- You absolutely ***must*** include a *<Spectral_Characteristics>* class from the Spectral Discipline Dictionary to describe things like binning and the location of the spectral axis or axes in your data. See Filling Out the Spectral Dictionary Class for more information.

Filling Out the Array 3D Data Structure

The ***<Array_3D>*** class is the generic base class that underlies all the other *<Array_3D_* >* classes. Use it when one of the more specialized classes is not appropriate and you don't need access to any software developed specifically for them.

For an *Array_3D*, follow the instructions for Filling Out the Array_2D Data Structure, with the following additional constraints/requirements:

- Use **<Array_3D>** and **</Array_3D>** rather than **<Array_2D>** and **</Array_2D>**
- **<axes>** must have a value of **3**, rather than **2**.
- There must be three **<Axis_Array>** classes, rather than two.
- You should include a *<Display_Settings>* class from the Display Discipline Dictionary in your *Discipline_Area* to define the correct way to display at least whatever constitutes a "plane" in your structure. If you think this does not apply to your data, please contact your node consultant. See Filling Out the Display Dictionary Class for additional information.