



*International*  
*Virtual*  
*Observatory*  
*Alliance*

# IVOA Spectral Data Model

## Version 2.0

### IVOA Proposed Recommendation 20130425

#### **This version**

[PR-SpectralDM-2.0-20130425](#)

#### **Previous versions:**

[WD-SpectralDM-2.0-20130308](#)

[WD-SpectralDM-2.0-20120907](#)

<http://www.ivoa.net/Documents/SpectrumDM/20111020/SpectrumDM-20111020.pdf>

#### **Editors:**

Jonathan McDowell, Mark Cresitello-Dittmar

#### **Contributors:**

Jonathan McDowell, Jesus Salgado, Carlos Rodrigo Blanco, Pedro Osuna, Doug Tody, Enrique Solano, Joe Mazzarella, Raffaele D'Abrusco, Mireille Louys, Tamas Budavari, Markus Dolensky, Inga Kamp, Kelly McCusker, Pavlos Protopapas, Arnold Rots, Randy Thompson, Frank Valdes, Petr Skoda, Bruno Rino, Jim Cant, Omar Laurino and the IVOA Data Access Layer and Data Model Working Groups.

#### **Abstract**

We present a data model describing the structure of spectrophotometric datasets with spectral and temporal coordinates and associated metadata. This data model may be used to represent spectra, time series data, segments of SED (Spectral Energy Distributions) and other spectral or temporal associations.

## Status of this document

This is an IVOA Proposed Recommendation made available for public review. It is appropriate to reference this document only as a recommended standard that is under review and which may be changed before it is accepted as a full Recommendation.

The **Virtual Observatory (VO)** is general term for a collection of federated resources that can be used to conduct astronomical research, education, and outreach.

The **International Virtual Observatory Alliance (IVOA)** (<http://www.ivoa.net>) is a global collaboration of separately funded projects to develop standards and infrastructure that enable VO applications. A list of current IVOA Recommendations and Standards can be found at (<http://www.ivoa.net/Documents>).

## Acknowledgements

This document has been developed with support from NSF and NASA under the Virtual Astronomical Observatory (VAO) project, the National Science Foundation's <http://www.nsf.gov> Information Technology Research Program under Cooperative Agreement AST0122449 with The Johns Hopkins University, from the UK Particle Physics and Astronomy Research Council (PPARC) <http://www.pparc.ac.uk>, and from the European Commission's Sixth Framework Program <http://fp6.cordis.lu/fp6/home.cfm> via the Optical Infrared Coordination Network (OPTICON), <http://www.astro-opticon.org>.

## Change Log

2011 Dec 21: Initial draft of version 2.0

- Grouped several top (Segment) level attributes into a new Dataset object.
- Organizational overhaul, consolidated objects defined in external models (Char & STC) to separate sections.

2012 Feb 08

- Update ucd strings with recommendations from Spectrum\_1.1 review

2012 Mar 14

- Updates from working group discussion (Segment, CoordSys, CoordFrame).

2012 Apr

- Changed from SpectralCharacterizationAxis extension of CharacterizationAxis containing ResPower attribute, to SpectralResolution extension of Resolution containing ResolvingPower attribute. This is more consistent with the Characterization and ObsCore model definitions.
- Previous versions of this model defined Data.\*Axis.Quality and Data.\*Axis.Quality.n utypes with little definition in the text. In this model, we move the Quality value to Data.\*Axis.Accuracy.QualityStatus, and the definition items to Char.\*Axis.QualityDefs.
- Added FITS serialization.

2012 Sep 09

- General cleanup of open issues.
- add UTypes for model nodes.
- clarify DataSet.DataModel value for Spectrum vs PhotometryPoint.

2013 Feb 07: Comments from Draft review period with working group.

- remove use of xmlns in VOTable example
- expanded definition of DataSet to include XSD pointer, etc.
- Utype updates to correlate with ObsCore and Char.

# Contents

<b>1</b>	<b>Introduction</b>	<b>9</b>
1.1	Motivation . . . . .	9
1.2	Requirements . . . . .	9
1.3	Note . . . . .	9
1.4	Use Cases . . . . .	10
1.5	IVOA Architecture Context . . . . .	11
<b>2</b>	<b>Spectral Data Model</b>	<b>12</b>
2.1	Spectral . . . . .	12
2.2	BaseSPS . . . . .	12
2.2.1	BaseSPS.Dataset . . . . .	12
2.2.2	BaseSPS.Characterization . . . . .	12
2.2.3	BaseSPS.CoordSys . . . . .	13
2.2.4	BaseSPS.Curation . . . . .	13
2.2.5	BaseSPS.Data . . . . .	13
2.2.6	BaseSPS.DataID . . . . .	13
2.2.7	BaseSPS.Derived . . . . .	13
2.2.8	BaseSPS.Target . . . . .	13
2.3	Characterization . . . . .	14
2.4	CharacterizationAxis . . . . .	14
2.4.1	CharacterizationAxis.CoordSys . . . . .	14
2.4.2	CharacterizationAxis.QualityDefs . . . . .	14
2.4.3	CharacterizationAxis.SpectralResolution . . . . .	15
2.5	CoordSys . . . . .	16
2.5.1	CoordSys.ID:String . . . . .	16
2.5.2	CoordSys.CoordFrame . . . . .	16
2.5.3	CoordSys.SpaceFrame . . . . .	16
2.5.4	CoordSys.SpectralFrame . . . . .	17
2.5.5	CoordSys.TimeFrame . . . . .	17
2.5.6	CoordSys.RedshiftFrame . . . . .	17
2.5.7	CoordSys.FluxFrame . . . . .	17
2.6	Corrections . . . . .	18
2.7	CorrectionItem . . . . .	18
2.7.1	CorrectionItem.Name: String . . . . .	18
2.7.2	CorrectionItem.Applied: Boolean . . . . .	18
2.8	GenericCorr . . . . .	18
2.8.1	GenericCorr.Value: Double . . . . .	18
2.9	ApFrac . . . . .	19
2.9.1	ApFrac.Name: String . . . . .	19
2.9.2	ApFrac.Value: Double . . . . .	19
2.9.3	ApFrac.Applied: Boolean . . . . .	19
2.10	Curation . . . . .	20
2.10.1	Curation.Publisher:String . . . . .	20
2.10.2	Curation.PublisherID:URI . . . . .	20
2.10.3	Curation.PublisherDID:String . . . . .	20
2.10.4	Curation.Date:String . . . . .	20

2.10.5	Curation.Version:String	20
2.10.6	Curation.Rights:Enum	20
2.10.7	Curation.Reference:String	20
2.10.8	Curation.Contact.Name:String	20
2.10.9	Curation.Contact.Email:String	21
2.11	Data	22
2.11.1	Data.DataAxis	22
2.12	DataAxis	23
2.12.1	Mapping to CharAxis	23
2.12.2	Domain subclasses	23
2.12.3	DataAxis.SystemID:String	24
2.12.4	DataAxis.name: String	24
2.12.5	DataAxis.ucd: String	24
2.12.6	DataAxis.unit: String	24
2.12.7	DataAxis.Value:Double	24
2.12.8	DataAxis.Accuracy	24
2.12.9	DataAxis.Correction	24
2.12.10	DataAxis.Resolution	24
2.13	DataID	25
2.13.1	DataID.Title:String	25
2.13.2	DataID.Creator:String	25
2.13.3	DataID.Collection:String	25
2.13.4	DataID.DatasetID:URI	25
2.13.5	DataID.CreatorDID:URI	25
2.13.6	DataID.Date:String	25
2.13.7	DataID.Version:String	25
2.13.8	DataID.Instrument:String	26
2.13.9	DataID.Bandpass:String	26
2.13.10	DataID.CreationType:Enum	26
2.13.11	DataID.Logo:URL	26
2.13.12	DataID.Contributor:String	26
2.13.13	DataID.DataSource:String	26
2.14	DataModel	27
2.14.1	DataModel.Name:String	27
2.14.2	DataModel.Prefix:String	27
2.14.3	DataModel.url:URL	27
2.15	Dataset	27
2.15.1	Dataset.DataModel	27
2.15.2	Dataset.Type:Enum	27
2.15.3	Dataset.Length:Integer	27
2.15.4	Dataset.SpectralSI:String	27
2.15.5	Dataset.TimeSI:String	28
2.15.6	Dataset.FluxSI:String	28
2.16	Derived	29
2.16.1	Derived.SNR:Double	29
2.16.2	Derived.Redshift.Value:Double	29
2.16.3	Derived.Redshift.StatError:Double	29
2.16.4	Derived.Redshift.Confidence:Double	29

2.16.5	Derived.VarAmpl:Double	29
2.17	FluxFrame	30
2.17.1	FluxFrame.refID:URI	30
2.17.2	FluxFrame.PhotCal	30
2.18	PhotCal	31
2.19	QualityCode	32
2.19.1	QualityCode.value: Integer	32
2.19.2	QualityCode.description: String	32
2.20	SpectralResolution	33
2.20.1	SpectralResolution.ResolPower	33
2.20.2	SpectralResolution.ResolPower.refVal: Double	33
2.21	Target	34
2.21.1	Target.Name:String	34
2.21.2	Target.Description:String	34
2.21.3	Target.Pos:(Double, Double)	34
2.21.4	Target.Class:String	34
2.21.5	Target.SpectralClass:String	34
2.21.6	Target.Redshift:Double	34
2.21.7	Target.VarAmpl:Double	34
<b>3</b>	<b>Characterization Data Model Elements</b>	<b>35</b>
3.1	Characterization (Char)	35
3.1.1	Char.CharacterizationAxis	35
3.2	CharacterizationAxis (CharAxis)	35
3.2.1	CharAxis.name: String	36
3.2.2	CharAxis.ucd: String	36
3.2.3	CharAxis.unit: String	36
3.2.4	CharAxis.CalibStatus: Enum	36
3.2.5	CharAxis.Accuracy	37
3.2.6	CharAxis.CoordSys	37
3.2.7	CharAxis.Coverage	37
3.2.8	CharAxis.Resolution	37
3.2.9	CharAxis.SamplingPrecision	37
3.3	Accuracy	37
3.3.1	Accuracy.BinSize: Double	38
3.3.2	Accuracy.BinLow: Double	38
3.3.3	Accuracy.BinHigh: Double	38
3.3.4	Accuracy.StatErrSize: Double	38
3.3.5	Accuracy.StatErrLow: Double	38
3.3.6	Accuracy.StatErrHigh: Double	38
3.3.7	Accuracy.SysError: Double	38
3.3.8	Accuracy.QualityStatus: Integer	39
3.4	Coverage	39
3.4.1	Coverage.Location	39
3.4.2	Coverage.Bounds	39
3.4.3	Coverage.Support	39
3.5	Location	39
3.5.1	Location.Value: Double	39

3.6	Bounds . . . . .	39
3.6.1	Bounds.Extent: Double . . . . .	39
3.6.2	Bounds.Start: Double . . . . .	39
3.6.3	Bounds.Stop: Double . . . . .	40
3.7	Support . . . . .	40
3.7.1	Support.Extent: Double . . . . .	40
3.7.2	Support.Area: String . . . . .	40
3.8	Resolution . . . . .	40
3.8.1	refVal: Double . . . . .	40
3.9	SamplingPrecision . . . . .	40
3.9.1	SampleExtent: Double . . . . .	40
3.9.2	SamplingPrecision.SamplingPrecisionRefVal.FillFactor: Double . . . . .	40
3.10	SpatialCharAxis . . . . .	41
3.11	SpectralCharAxis . . . . .	42
3.12	TimeCharAxis . . . . .	43
3.13	FluxCharAxis . . . . .	44
<b>4</b>	<b>STC Data Model Elements</b>	<b>45</b>
4.1	Frame . . . . .	45
4.1.1	Frame.Name:String . . . . .	45
4.1.2	Frame.ID:String . . . . .	45
4.1.3	Frame.UCD:String . . . . .	45
4.2	CoordFrame . . . . .	46
4.2.1	CoordFrame.ReferencePosition:String . . . . .	46
4.3	RedshiftFrame . . . . .	46
4.3.1	RedshiftFrame.DopplerDefinition: Enum . . . . .	46
4.4	SpaceFrame . . . . .	46
4.4.1	SpaceFrame.Equinox: Double . . . . .	47
4.5	SpectralFrame . . . . .	47
4.5.1	SpectralFrame.Redshift: Double . . . . .	47
4.6	TimeFrame . . . . .	47
4.6.1	TimeFrame.Zero: Double . . . . .	47
<b>5</b>	<b>IVOA Conventions</b>	<b>49</b>
5.1	Model Reuse and UTypes . . . . .	49
5.1.1	Model Prefix (Namespace) . . . . .	49
5.1.2	Model Reuse . . . . .	49
5.1.3	UTypes . . . . .	49
5.2	Units . . . . .	49
5.3	UCDs . . . . .	50
5.4	Extensibility . . . . .	52
5.4.1	Namespace . . . . .	52
5.4.2	UTypes . . . . .	52
5.4.3	Scope . . . . .	52
5.4.4	Support . . . . .	52
5.4.5	Note . . . . .	53
<b>6</b>	<b>Tables</b>	<b>54</b>

<b>7</b>	<b>Spectrum Data Model</b>	<b>56</b>
7.1	Spectrum . . . . .	56
7.1.1	Model Prefix . . . . .	56
7.1.2	Axis Requirements . . . . .	56
7.1.3	UTypes . . . . .	56
7.1.4	UCDs and Units . . . . .	57
7.1.5	Required Fields . . . . .	57
<b>8</b>	<b>Photometry Point Data Model</b>	<b>58</b>
8.1	PhotometryPoint . . . . .	58
8.1.1	Model Prefix . . . . .	58
8.1.2	Axis Requirements . . . . .	58
8.1.3	UTypes . . . . .	58
8.1.4	UCDs and Units . . . . .	59
8.1.5	Required Fields . . . . .	59
8.1.6	Restrictions . . . . .	59
<b>A</b>	<b>Spectral Data Model UTypes</b>	<b>60</b>
<b>B</b>	<b>Spectral Data Model Axis UCDs</b>	<b>66</b>
B.1	Flux Axis UCDs . . . . .	66
B.2	Spectral Axis UCDs . . . . .	66
<b>C</b>	<b>Sample Serializations</b>	<b>67</b>
C.1	VOTable Serialization . . . . .	67
C.1.1	Basic Spectrum Instance . . . . .	67
C.1.2	Basic PhotometryPoint Instance . . . . .	70
C.2	FITS Serialization . . . . .	73
C.2.1	Basic Spectrum Instance . . . . .	78
C.2.2	Basic PhotometryPoint Instance . . . . .	79
	<b>References</b>	<b>80</b>



# 1 Introduction

## 1.1 Motivation

Data Models in the VO aim to define the common elements of astronomical data and metadata collections and to provide a framework for describing their relationships so these become interoperable in a transparent manner. Spectra and photometric data are stored in many different ways within the astronomical community. In this document we present a proposed abstraction for such data, which can be used for describing spectrum and photometry datasets and also reused by other standards.

The Spectral Data Model (SpectralDM) standard presents a core data model describing the structure of spectrophotometric datasets with coordinate axes and associated metadata. This data model may be used to represent spectra, time series data, generic segments of a Spectral Energy Distribution (SED) and other spectral or temporal associations.

We provide sample serializations for the Spectrum and PhotometryPoint dataset use cases in VOTABLE and FITS formats.

This version incorporates elements of the Photometry model needed to represent PhotometryPoint data. We are also addressing comments resulting from the implementation of previous versions of this model and making adjustments which bring the Characterization and STC components in better agreement with their parent model. Most changes are minor and do not represent any fundamental changes with respect to the previous Recommendation. Modifications to UTypes and UCDs (Appendix A) can have an impact on implementations and/or validators.

## 1.2 Requirements

The primary goals of this document are:

- to provide a specification of a generic spectrophotometric segment.
- to represent a single 1-dimensional spectrum in sufficient detail to understand the differences between two spectra of the same object and between two spectra of different objects.
- to represent photometric measurements in magnitudes, together with their calibration.
- to enable the representation of time series photometry, with many photometry points of the same object at different times.

## 1.3 Note

We would like to note that while the diagrams in this document use many of the UML notations and structure, they are NOT UML format. Readers should use the diagrams as a guide to object relations and usage, but should not interpret them as strict UML representations.

## 1.4 Use Cases

## 1.5 IVOA Architecture Context

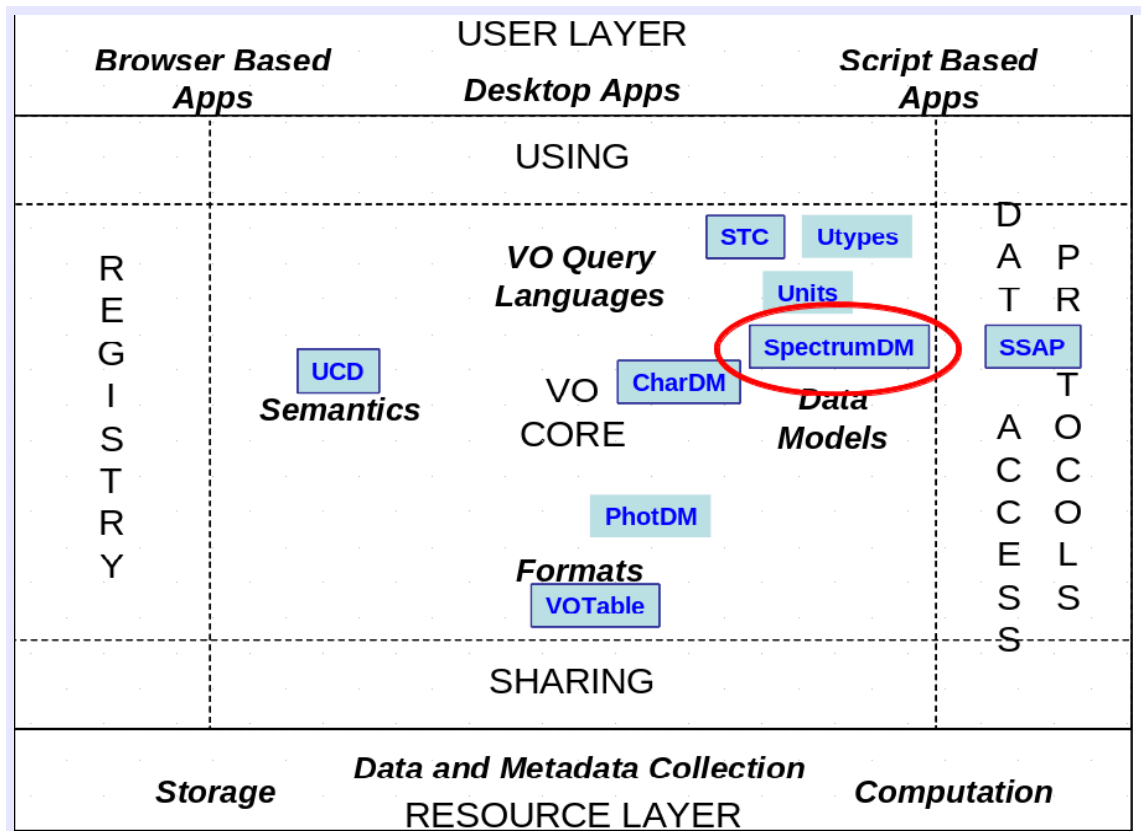


Figure 1: Spectral DM in IVOA architecture

## 2 Spectral Data Model

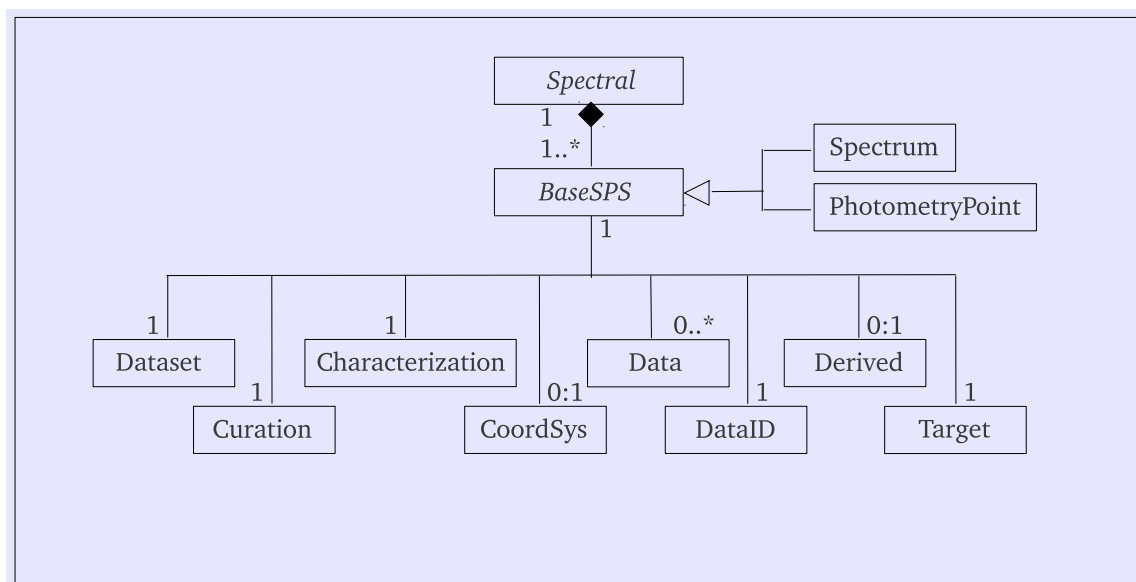


Figure 2: High level class diagram for the Spectral Data Model.

### 2.1 Spectral

Abstract container class for holding multiple instances of the BaseSPS class. This object cannot be instantiated on its own. It merely serves as a hook for defining more complex models.

### 2.2 BaseSPS

Abstract object defining the Base SpectroPhotometric Sequence (Sequence). This object contains attributes common to all SpectroPhotometric data, but cannot be instantiated on its own. Specific types must be defined as extensions of this object. This document includes the definition of the Spectrum (section 7) and PhotometryPoint (section 8) extensions.

#### 2.2.1 BaseSPS.Dataset

See Section 2.15 for content definition. The Dataset object encapsulates metadata describing what sort of Sequence this is. Its primary purpose is in support of data discovery.

#### 2.2.2 BaseSPS.Characterization

See Section 2.3 for content definition.

The Characterization object provides a 'characteristic' view of the coordinate space for the Sequence as a whole.

There is strong correlation between the information provided here, with that provided under the Data element. The distinction is that Characterization provides a broad scale description of the coordinate space, while the Data section provides specific information relevant to an individual Data 'point'. For example, the Accuracy fields in Characterization represent typical accuracy for the dataset, while those within Data provide per-data-point errors.

Where overlap exists, the values provided within Characterization should be considered the default value for the corresponding element of the Data element.

### **2.2.3 BaseSPS.CoordSys**

See Section 2.5 for content definition.

CoordSys is a container object providing the 'default' coordinate frames used throughout the Sequence. In this context, the object should be used as a simple container of coordinate frames. The additional fields for the specialized frames should not be utilized since here we provide no restrictions on the number of instances of any particular frame type.

### **2.2.4 BaseSPS.Curation**

See Section 2.10 for content definition.

Curation object associated with the Sequence.

### **2.2.5 BaseSPS.Data**

See Section 2.11 for content definition.

Set of zero (0) or more Data objects associated with the Sequence. Each holds all information related to a single data value (or point).

### **2.2.6 BaseSPS.DataID**

See Section 2.13 for content definition.

DataID provides high level identification metadata for the Sequence, and any associations with various collections.

### **2.2.7 BaseSPS.Derived**

See Section 2.16 for content definition.

Provides a high level summary of certain properties of the Sequence. Its primary purpose is to support high level filtering of datasets during data discovery.

### **2.2.8 BaseSPS.Target**

See Section 2.21 for content definition.

Target object associated with the Sequence.

## 2.3 Characterization

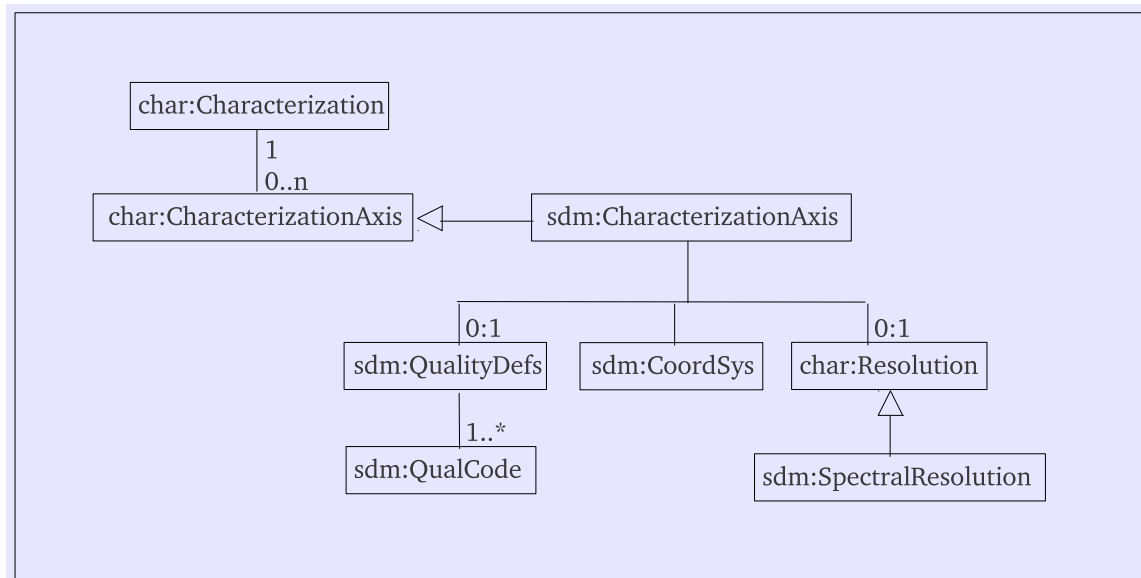


Figure 3: Characterization object showing extensions to Characterization model elements.

This model uses a Characterization object which is consistent with the IVOA Characterization Data Model Standard[4]. However, this model imposes certain restrictions and simplifications on several elements of that model. Section 3.1 of this document provides details on the definitions used in this model. In this section, we provide definitions for extensions to those elements which are required for this model.

## 2.4 CharacterizationAxis

The content of CharacterizationAxis is defined within the Characterization model. Refer to Section 3.2 of this document for details on the definition used in this model.

### 2.4.1 CharacterizationAxis.CoordSys

In Section 2.5 we define a modified CoordSys object for this model to accommodate the Spectral specific FluxFrame. This object overrides the definition given in the Characterization model. Its usage under CharacterizationAxis is unchanged, and is described in Section 3.2.

### 2.4.2 CharacterizationAxis.QualityDefs

We extend the CharacterizationAxis object with an optional QualityDefs container for a set of one (1) or more Quality code definitions (QualityCode).

The Accuracy model defines a numerical QualityStatus 'flag' indicating the quality of the associated value. This object provides a mapping of the possible QualityStatus values to a string representation of their meaning. We define it at the Characterization level since the definitions are valid for the Sequence as a whole, and on the CharacterizationAxis, since each axis may have independent definitions.

### **2.4.3 CharacterizationAxis.SpectralResolution**

Extension of the CharacterizationAxis.Resolution object with spectrum specific content. This object is to be associated with the SpectralCharAxis type in place of the parent object.

## 2.5 CoordSys

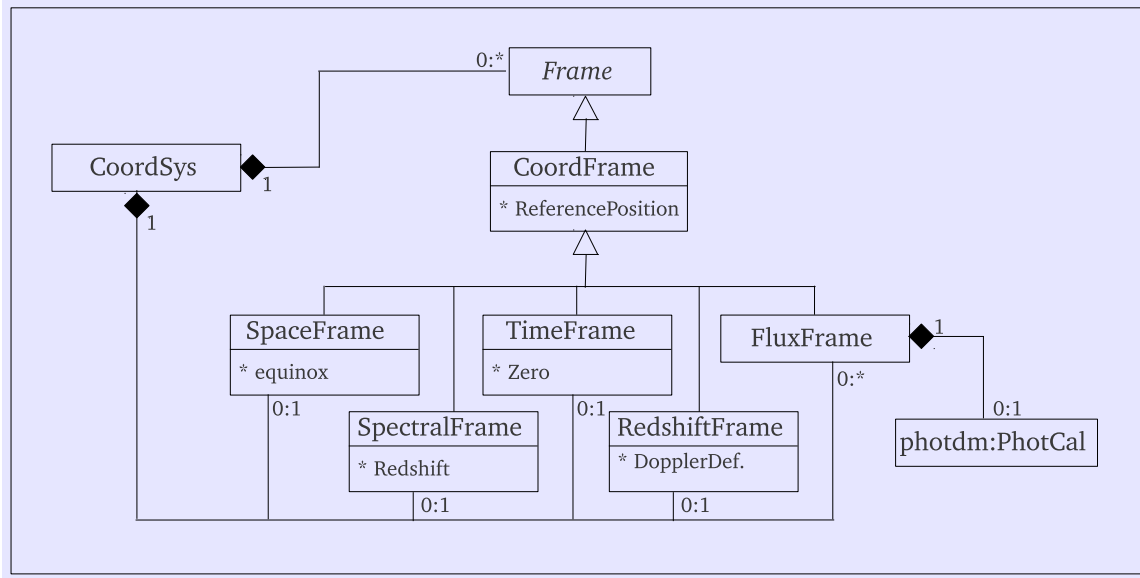


Figure 4: Diagram for CoordSys object.

The CoordSys object is derived from the AstroCoordSystem object of the STC model. This is the same object used by the Characterization model for this element, but here we modify it for Spectral specific requirements.

- The set of zero or more generic frames refers to the base Frame object, rather than the derive CoordFrame. This enables us to include the specialized coordinate frames (SpaceFrame, TimeFrame, etc. ).
- In addition to holding zero or one of each of the specialized frames, (SpaceFrame, TimeFrame, etc. ), we add a set of zero or more FluxFrames.

See "STC Model Elements" in Section 4 for descriptions of the Frame hierarchy used in this model.

### 2.5.1 CoordSys.ID:String

User defined free form string giving the overall ID for the coordinate system.

### 2.5.2 CoordSys.CoordFrame

CoordSys contains one (1) or more Coordinate Frames, each of which describes a particular set of coordinate axes.

### 2.5.3 CoordSys.SpaceFrame

Zero or one specialized SpaceFrame object.



#### **2.5.4 CoordSys.SpectralFrame**

Zero or one specialized SpectralFrame object.

#### **2.5.5 CoordSys.TimeFrame**

Zero or one specialized TimeFrame object.

#### **2.5.6 CoordSys.RedshiftFrame**

Zero or one specialized RedshiftFrame object.

#### **2.5.7 CoordSys.FluxFrame**

Zero or more specialized FluxFrame objects.

## 2.6 Corrections

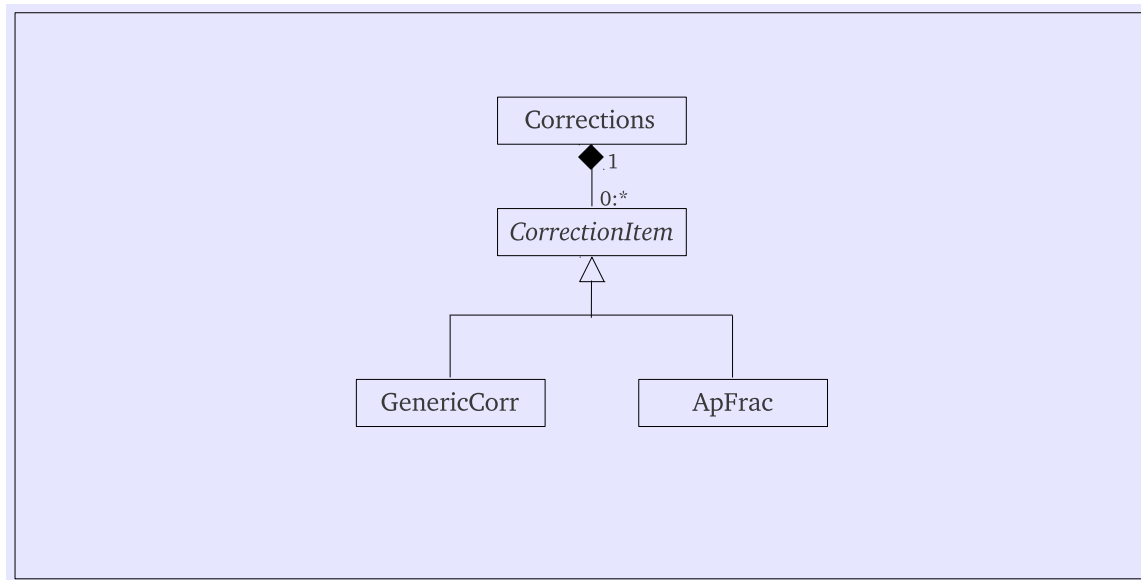


Figure 5: Diagram for Correction object

Holds a set of zero (0) or more **CorrectionItem** objects. Corrections are associated with values and provide a mechanism to define adjustments made, or to be made.

## 2.7 CorrectionItem

This is an abstract object which defines the basic interface to correction objects. Conceptually we can generalize these objects as examples of 'corrections to apply' and 'corrections applied'.

### 2.7.1 CorrectionItem.Name: String

free format, user defined name of the correction type.

### 2.7.2 CorrectionItem.Applied: Boolean

Boolean flag indicating whether or not the correction has been applied.

## 2.8 GenericCorr

Extension of **CorrectionItem** which allows for a user-defined correction which has no modeled definition.

### 2.8.1 GenericCorr.Value: Double

Size or scale of the correction.

## 2.9 ApFrac

Extension of CorrectionItem specifying an Aperture Fraction correction.

This particular correction is conceptually quite close to the Spatial Filling Factor defined in the IVOA Characterization model to account for e.g. gaps between pixels. However, the filling factor is thought of as a statistical correction, while the aperture fraction is a single modeled correction.

Typically a photometric measurement captures only part of the flux from the object. For a point source, the measurement aperture (physical or software) may include only a finite fraction of the instrument point spread function. For an extended source such as a galaxy, we may wish to guess the total object brightness based on the fraction actually observed. In each case this correction depends on comparing the observation aperture to a spatial model of the light distribution of the source. It is therefore a function not of the photometric band, but of the individual data points. If such a correction has been applied, we will wish to record the amount of the correction. If no correction has been applied, one could record the appropriate point-source correction, bearing in mind that it may not be relevant.

We define the 'adopted Aperture Fraction Correction' (i.e. what you think the right correction is) as

$$\text{Measured flux} = \text{Ap Fraction} \times \text{Total flux}$$

### 2.9.1 ApFrac.Name: String

fixed value of "ApFrac".

### 2.9.2 ApFrac.Value: Double

Value between 0.0 and 1.0

### 2.9.3 ApFrac.Applied: Boolean

Boolean flag indicating whether or not the correction has been applied.

## 2.10 Curation

Curation is an object which provides information about the entity responsible for the support of the dataset content. It is assembled from definitions provided by the IVOA standard document, "Resource Metadata for the Virtual Observatory Version 1.12" [1] (Resource Metadata). Here, we provide a brief description of each field for easy reference, along with a notation of its mapping to the Resource Metadata document (RM:field), where the reader may find more detailed information.

### 2.10.1 Curation.Publisher:String

The entity making the data available. (RM:Curation.Publisher)

### 2.10.2 Curation.PublisherID:URI

The identifier for the publisher, provided according to the syntax defined in "IVOA Identifiers" [2]. (RM:Curation.PublisherID)

### 2.10.3 Curation.PublisherDID:String

This is a publisher specified dataset ID (RM:Resource.Identifier) and may be an internal ID used by the archive.

### 2.10.4 Curation.Date:String

Date the curated dataset was last modified. The preferred format is ISO8601 (YYYY-MM-DD). (RM:Curation.Date)

### 2.10.5 Curation.Version:String

Version is provided by the publisher or creator and may be any string. (RM:Curation.Version)

### 2.10.6 Curation.Rights:Enum

Indicates access privileges to the content. (RM:Collection.Rights)  
Allowed values are:

- "public"
- "proprietary"
- "mixed"

### 2.10.7 Curation.Reference:String

A set of one (1) or more bibliographic or documentation references associated with this dataset. (RM:General.Source)

### 2.10.8 Curation.Contact.Name:String

Name of a contact responsible for the content of the dataset. (RM:Curation.Contact.Name)

### **2.10.9 Curation.Contact.Email:String**

e-mail address of the contact. (RM:Curation.Contact.Email)

## 2.11 Data

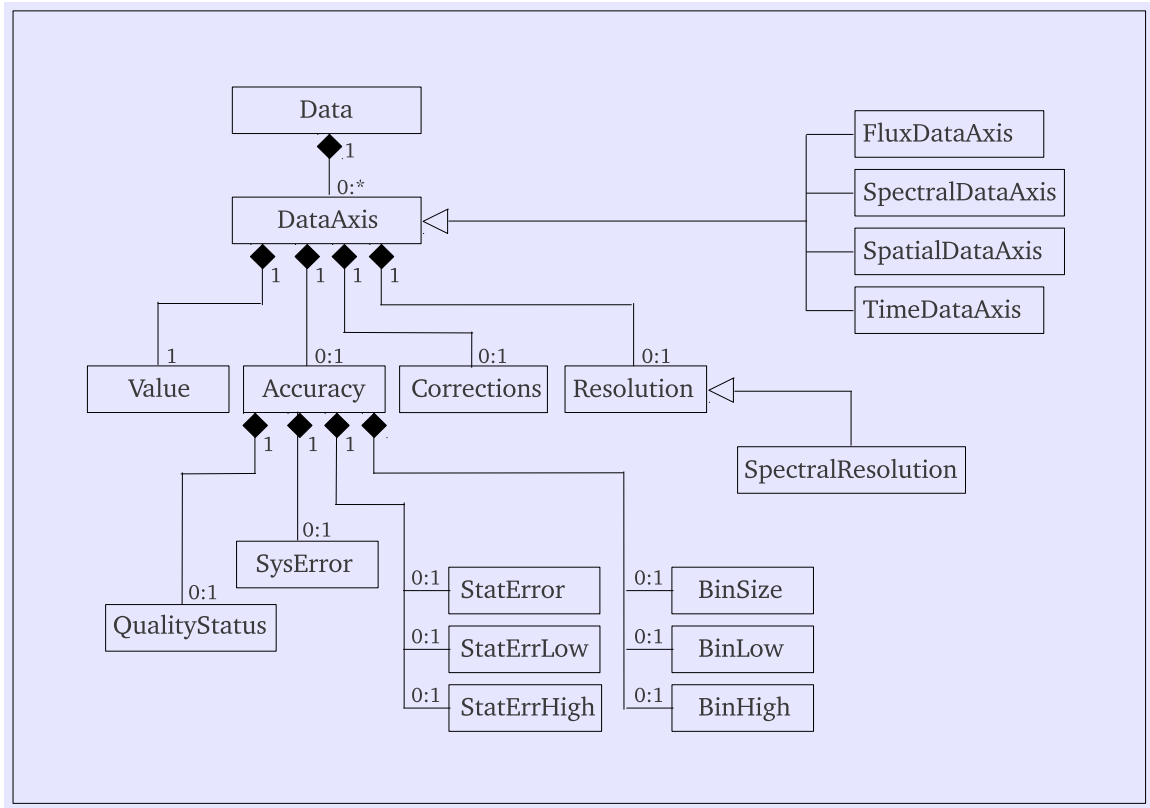


Figure 6: Diagram for Data object

The Data object is a container holding a set of DataAxis objects for the dataset. The number of, and definition of specific axes depends on the use case. It is for the higher level components to define the axis requirements for that case. Since the Characterization axes provide global defaults for the Data axes, they should have consistent axis requirements.

This object has a very similar construct to the Characterization object. The distinction is that here we provide specific information related to a particular data point, while the Characterization object holds broad scale information for the Sequence as a whole.

### 2.11.1 Data.DataAxis

This object contains a complete definition of an individual data point. This includes the coordinate space, accuracy, quality, and the data value.

## 2.12 DataAxis

The DataAxis object reuses many elements which are described in the Characterization model for the support of the CharacterizationAxis object. This reflects the nature of these objects as both being extensions of a more general Axis type object. We reuse these elements here with the same definitions as under CharacterizationAxis (see Section 3.2).

### 2.12.1 Mapping to CharAxis

The following elements are defined in both CharacterizationAxis and DataAxis. The CharacterizationAxis values serve as the default value for the corresponding element in DataAxis and are considered to have the definitions as stated. For example, Char.SpectralAxis.Accuracy.StatError constitutes the default value for each Data.SpectralAxis.Accuracy.StatError field.

Mapping of CharAxis and DataAxis common elements	
CharAxis field	Mapping definition to DataAxis
*Axis.name	Data and CharAxis share the same name unless otherwise stated.
*Axis.ucd	All data values of this axis are represented by this ucd unless specifically stated otherwise.
*Axis.unit	All data values of this axis are represented in this unit unless specifically stated otherwise.
*Axis.Accuracy.BinSize	Average width of all bins along this axis as determined from either BinSize or (BinHigh - BinLow).
*Axis.Accuracy.StatError	Average statistical error of all values along this axis as determined from either StatError or (StatErrHigh - StatErrLow ).
*Axis.Accuracy.SysError	Average systematic error of all values along this axis.
*Axis.Resolution	Average resolution of all values along this axis.

### 2.12.2 Domain subclasses

We allow the same set of domain specializations and named instances as described in Section 3.2 for the CharacterizationAxis, namely;

**FluxAxis** - instance of FluxDataAxis.

**SpatialAxis** - instance of SpatialDataAxis.

**SpectralAxis** - instance of SpectralDataAxis.

**TimeAxis** - instance of TimeDataAxis.

**BackgroundModel** - additional instance of type FluxDataAxis, which shares the same domain restrictions, and draws default values from the Char.FluxAxis instance.

The domain restrictions for the Accuracy and Resolution objects (the only overlapping objects) are the same, except that here, we also allow access to the various Low/High fields.

Their usage in the UType list is identical.

### **2.12.3 DataAxis.SystemID:String**

ID string of the coordinate system associated with this data axis. This field provides a mechanism for linking a DataAxis to a particular CharacterizationAxis via its coordinate system (COORDSYS) ID. If not provided, the linkage is implied through the axis specialization type (SpectralAxis, TimeAxis, etc).

### **2.12.4 DataAxis.name: String**

Free form name of the axis.

### **2.12.5 DataAxis.ucd: String**

Defines the specific physical quantity being covered by the axis. There are many different physical quantities which may be covered by a single axis definition. It is this UCD field which is used to unambiguously distinguish them.

### **2.12.6 DataAxis.unit: String**

Unit for the axis.

### **2.12.7 DataAxis.Value:Double**

Value of the data point along that particular axis.

### **2.12.8 DataAxis.Accuracy**

Accuracy object as described in Section 3.3

Note: The width values (e.g. StatError, BinSize), are suitable for Characterization accuracy (the global accuracy) while the low/high values have meaning within Data (the per-point values).

### **2.12.9 DataAxis.Correction**

Holds a set of zero (0) or more CorrectionItems associated with this axis. These define various corrections which may or may not have been applied to the data value.

### **2.12.10 DataAxis.Resolution**

The Resolution object as described in Section 3.8



## 2.13 DataId

The Data Identification object (DataID) gives the dataset ID for a particular Sequence, and its membership of larger collections.

The Dataset IDs in this object must comply with the syntax for dataset identifiers defined in the "IVOA Identifiers" [2] document, including the use of 'stop' characters to identify specific datasets that are not individually in the registry.

Much of the content of this object is assembled from various definitions in the IVOA Resource Metadata document. Here, we provide a brief description of each field for easy reference, along with a notation of its mapping to the Resource Metadata document (RM:field), where the reader may find more detailed information.

### 2.13.1 DataID.Title:String

A free form string giving a title for the Sequence. (RM:Identity.Title)

### 2.13.2 DataID.Creator:String

A free form string giving the name of the creator of the document. (RM:Curation.Creator)

### 2.13.3 DataID.Collection:String

A free form string, generally defined and set by the creating entity. The Sequence is associated with one or more Collections (instrument name, survey name. etc.) indicating some degree of compatibility with other data sharing the same Collection properties. Examples: "WFC", "Sloan", "BFS Spectrograph", "MSX Galactic Plane Survey".

### 2.13.4 DataID.DatasetID:URI

The value in this field may be the same as in Curation.PublisherDID.

For this field we recommend a journal-based URI such as the IVOA/ADEC/ADS dataset identifier. By agreement between the AAS journals, the ADS and the ADEC (NASA data centers), dataset identifiers, described in <http://vo.ads.harvard.edu/dv/>, will be used to link journal articles back to the archival datasets containing the relevant observational data. If analogous but independent systems of URI designation are later adopted by other centers (e.g. by European journals) and accepted by IVOA, they will be suitable in this field.

### 2.13.5 DataID.CreatorDID:URI

The Dataset ID as defined internally by the creator. It may be entirely different from the DatasetID field. It is used to identify a particular original exposure in an archive and will not necessarily change even if the VO object in question is a cutout or is otherwise further processed.

### 2.13.6 DataID.Date:String

Data processing or creation date (RM:Curation.Date). The preferred format is ISO8601 (YYYY-MM-DD).

### 2.13.7 DataID.Version:String

Version of the creator-produced dataset.

### 2.13.8 DataID.Instrument:String

This field identifies the instrument used to create the data. (RM:Collection.Instrument)  
This can be a specific instrument name, general type or something else, e.g., a program in the case of theoretical data. We restrict this field to a single value.

### 2.13.9 DataID.Bandpass:String

A string describing the spectral range. This field corresponds to both the Coverage.Spectral and Coverage.Spectral.Bandpass fields of the Resource Metadata document. The value may be any one of the strings listed for Coverage.Spectral (e.g. "Optical") or Coverage.Spectral.Bandpass (e.g. "B" ). Note that at this time there is no fixed list of values for this latter group.

### 2.13.10 DataID.CreationType:Enum

The dataset creation type describes the nature or genre of the content. (RM:General.Type)  
It may have one of the following values:

**Archival** Indicates that it is one of a collection of datasets generated in a systematic, homogeneous way and stored statically (or at least versioned). It will be possible to regenerate this dataset at a later date. The remaining types imply on-the-fly manipulation.

**Cutout** Indicates that the dataset was created 'on-the-fly', by subsetting, but not by modifying values.

**Filtered** May involve excluding data prior to binning into samples, also 'on-the-fly'.

**Mosaic** Combines multiple original datasets "on-the-fly".

**spectral extraction** Has been extracted, for example, from a spectral data cube

**catalog extraction** Has been extracted from a catalog.

### 2.13.11 DataID.Logo:URL

URL pointer to a graphical logo associated with the creator of the document content. -  
(RM:Curation.Creator.Logo)

### 2.13.12 DataID.Contributor:String

This field contains a set of one (1) or more strings identifying entities which contributed to the content of the document. (RM:Curation.Contributor)

### 2.13.13 DataID.DataSource:String

Describes the original source of the data (e.g. 'survey').

## 2.14 DataModel

### 2.14.1 DataModel.Name:String

Free format string provides the name and version of the model document. Recommended format is "[name]-[version].[subversion]"

### 2.14.2 DataModel.Prefix:String

Prefix string used in UType string for elements associated with this model. For IVOA standard models, this will be a predetermined string (e.g. 'stc' or 'spec'). Extended, or user defined content can specify a unique prefix for their own content.

### 2.14.3 DataModel.url:URL

URL pointer to the XML Schema associated with the model.

## 2.15 Dataset

This object contains metadata related to the overall dataset definition. It specifies what kind of dataset it is, the model(s) which describe it, and other qualifying information useful in support of data discovery and access. It provides a mechanism for linking data providers and users to a file via data access protocol. It is extremely generic, and may be useful for any type of data.

### 2.15.1 Dataset.DataModel

Provided metadata related to the data model(s) used within the document. If present, the first instance must describe the primary model used (e.g. the Spectrum model). Additional instances may be added to describe the models for any extended content.

### 2.15.2 Dataset.Type:Enum

Indicates the makeup of the data content. May have one of the following values which we are listed here as though they were a subset of spectrophotometric related values extracted from a controlled value list in a generic Dataset document:

- "Mixed" - Some combination of the other options.
- "Photometry" - Data have different spectral coord with irregular gaps.
- "Spectrum" - Data have different spectral coord in contiguous bins.
- "TimeSeries" - Data share the same spectral coord, varying times.

### 2.15.3 Dataset.Length:Integer

The number of points in the data.

### 2.15.4 Dataset.SpectralSI:String

This field gives the default units for the Spectral axis represented in the Osuna-Salgado dimensional format. It represents a translation between the default units as given in Char.SpectralAxis.unit to base SI units. See Section 5.2 for more information on that format.

### **2.15.5 Dataset.TimeSI:String**

This field gives the default units for the Time axis represented in the Osuna-Salgado dimensional format. It represents a translation between the default units as given in Char.TimeAxis.unit to base SI units. See Section 5.2 for more information on that format.

### **2.15.6 Dataset.FluxSI:String**

This field gives the default units for the Flux axis represented in the Osuna-Salgado dimensional format. It represents a translation between the default units as given in Char.FluxAxis.unit to base SI units. See Section 5.2 for more information on that format.

## 2.16 Derived

The Derived (short for Derived Data) object holds summary information about the Sequence.

### 2.16.1 Derived.SNR:Double

The signal-to-noise (SNR) is provided mainly as a way for searches to exclude data whose quality is insufficient for a particular study. Data providers may use their own definition, as we do not prescribe a uniform method to calculate it. A suitable method, set forth by the STScI/ST-ECF/CADC Spectral Container Working Group, is to define the signal as the median of the flux values in the spectrum and the noise as the median absolute third-order difference of flux values spaced two pixels apart. This noise value is then multiplied by  $1.482602 / \sqrt{6}$ . A detailed description and discussion of the algorithm can be found in the "ST-ECF newsletter; issue #42" [6]. Implementations of the algorithm can be obtained from "stecf.org" [7]

This method describes the high-spectral-frequency noise but does not take into account intermediate-spectral-frequency background 'noise'. Projects which are background dominated may wish to include this in the noise definition. Furthermore, most spectra vary in SNR across their waveband; users should therefore only use this single SNR as a crude selection parameter.

### 2.16.2 Derived.Redshift.Value:Double

This field represents a measurement of the redshift from the data. We provide fields for both the redshift measured value and statistical error. We define the redshift to be  $\Delta\lambda/\lambda$  and it may be positive or negative.

NOTE: There are two other redshifts in our model:

- the Target redshift: stores the Target redshift as determined by other means.
- the SpectralFrame redshift: used only if a "rest frame" spectrum is presented and represents the assumed redshift used to shift the spectrum.

### 2.16.3 Derived.Redshift.StatError:Double

Error on measured redshift.

### 2.16.4 Derived.Redshift.Confidence:Double

Confidence is an additional measure of accuracy, which expresses a probability between 0 and 1 that the quoted errors do apply. Example: This measure is used in the Sloan spectral service to provide a way of describing the estimated probability that the redshift is completely in error because the lines have been misidentified. Its default value is 1.0.

### 2.16.5 Derived.VarAmpl:Double

This field is a dimensionless value indicating the variability amplitude as a fraction of the mean. The value must be positive, but is otherwise unbound. It is a characteristic amplitude, a precise value is not required. (e.g. a value of 0.2 implies a 20 percent variation around the mean).

## 2.17 FluxFrame

Extension of the STC CoordFrame object used to provide a photometric calibration, for the case of photometric data, especially those in magnitude units. The photometric band fits naturally into the CoordSys model as it defines a coordinate frame for the flux axis.

- The FluxFrame.Name field is used to label the photometric band. This label is intended for user information and purposes such as the labeling of plots. A short name such as 'B' or 'SDSS g' is appropriate.

### 2.17.1 FluxFrame.refID:URI

This field is an IVOA resource identifier pointing to a photometric calibration which conforms to the PhotCal data model defined in the "IVOA Photometry Data Model standard" [3].

### 2.17.2 FluxFrame.PhotCal

PhotCal object associated with the FluxFrame. While it is permitted, it is not recommended that users include both the refID to an external calibration (above) and an embedded definition at this pointer. If both are defined, this pointer should be considered definitive. It is left to the user to verify the consistency of the reference.

## 2.18 PhotCal

Photometric data expressed in magnitudes refer to a photometric calibration described using the PhotCal object. This object is fully defined in the "IVOA Photometry Data Model" [3].

The filter reference coordinate and its metadata are already well described by the Char.SpectralAxis.Coverage.Location object of this model. Implementors should note that while for optical photometry, we expect the unit and UCD will normally be angstrom and 'em.wl', this will likely not be true for non-optical wavebands.

Note that in the Photometry standard, the PhotometryFilter model has its own Characterization, but this is not relevant for the PhotometryPoint use case; here we will use the Characterization object as described in this model. The semantics are subtly different: the time axis for the PhotometryFilter characterization describes when the filter was in use on a given telescope, which is not relevant to our use case of a particular observation.

The filter transmission curve should be instantiated as a separate Spectrum instance and pointed to with a resource identifier in the utype

```
PhotCal.PhotometryFilter.transmissionCurve.
```

The full list of utypes for PhotCal is given in the Photometry standard and should be considered the normative document for them. Please refer to Section 5.1 - Model Reuse and UTypes for a description of how these model components are incorporated into this model.

## **2.19 QualityCode**

The field provides a mapping between a particular Quality value (eg. Accuracy.QualityStatus) and its associated meaning. The values of 0 and 1 are reserved for indicating generic "good" and "bad" data respectively.

### **2.19.1 QualityCode.value: Integer**

Numeric value of the quality code.

### **2.19.2 QualityCode.description: String**

Free form string description of what the numeric value means.



## 2.20 SpectralResolution

Extension of the Characterization model Resolution object with spectrum specific content.

### 2.20.1 SpectralResolution.ResolPower

This element is extracted from the ObsCore model where it is used in exactly this manner. The definition here is a simplified version of that object.

### 2.20.2 SpectralResolution.ResolPower.refVal: Double

We allow an alternative field for quantifying the resolution, called the spectral resolving power. This is the dimensionless  $\lambda/\Delta\lambda$  which is often preferred for spectra because it is often more constant across the spectrum than the resolution.

ResPower and Resolution can be interchanged by dividing out Coverage.Location.

## **2.21 Target**

In spectral data it is particularly important to be able to specify the target of the observation, which may be an astronomical source or some other target (calibration, diffuse background, etc.). By explicitly including a target model we can not only facilitate searches on particular types of target but also support archives of model spectra for which the Coverage fields may not be relevant.

### **2.21.1 Target.Name:String**

Target name.

### **2.21.2 Target.Description:String**

Free form description of target.

### **2.21.3 Target.Pos:(Double, Double)**

This field gives a nominal RA and Dec for the target. For example, the catalog position of the source. Note: This is distinct from the Coverage.Location fields which indicate the actual telescope pointing position for that Sequence.

### **2.21.4 Target.Class:String**

Target or object class There is no international standard list of valid values for Target class. Nevertheless, an initial deployment of the VO would gain some benefit from using archive-specific classes, and provide a framework for converging on a standard list.

### **2.21.5 Target.SpectralClass:String**

Spectral class of object. There is no international standard list of valid values for Target spectral class.

### **2.21.6 Target.Redshift:Double**

This field gives the actual redshift of the astronomical object. It is normally used to store the cosmological redshift of extragalactic objects, although it may also be used to store the observed redshift of Galactic sources if that information is felt by the data provider to be useful.

Note: This is distinct from the Derived.Redshift which indicates a measured redshift value.

### **2.21.7 Target.VarAmpl:Double**

Typical target variability amplitude.

### 3 Characterization Data Model Elements

#### 3.1 Characterization (Char)

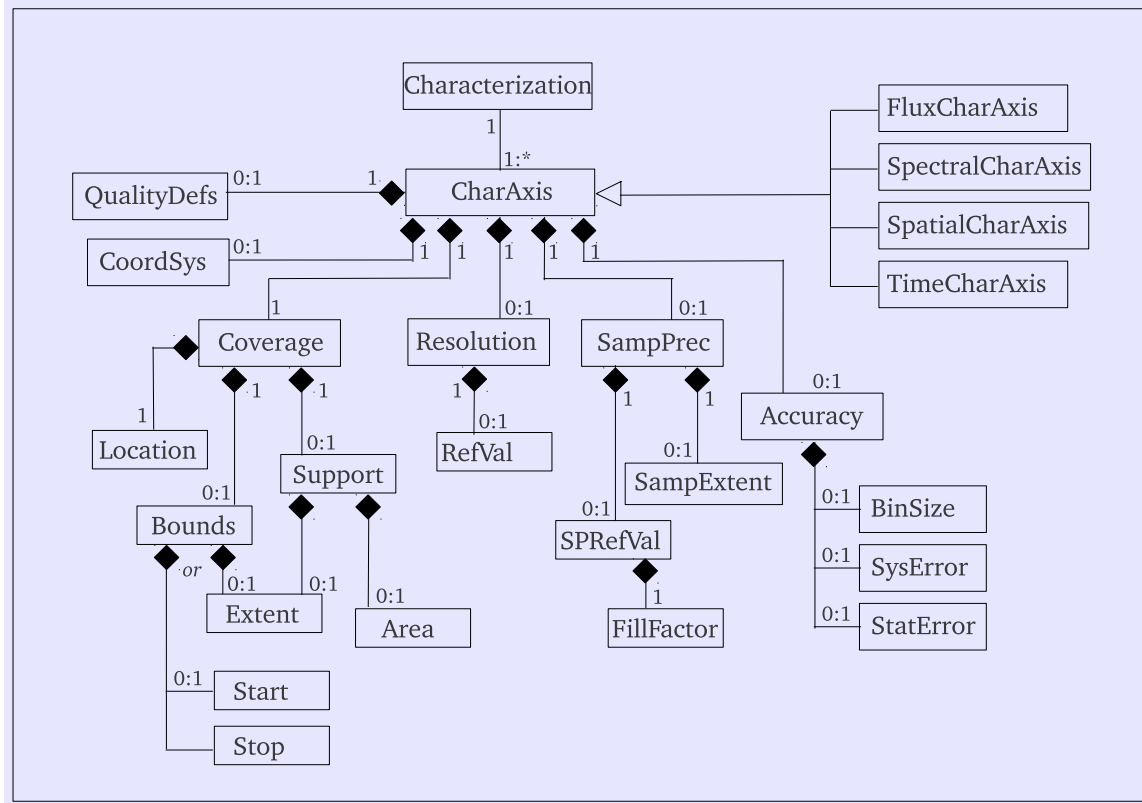


Figure 7: Diagram for Characterization object

The Characterization object and components described in this document are consistent with the "Data Model for Astronomical DataSet Characterisation" [4] (Characterization Model). Here, we describe a subset of Objects and Fields which are to be supported within this model. Note that there may be differences in interpretation of certain elements of the Characterization model and it's components.

Please refer to Section 5.1, Model Reuse and UTypes for a description of how these model components are incorporated into this model.

The Characterization object is a container holding a set of CharacterizationAxis objects for the data set. The number of and definition of specific axes depends on the use case, and it is for the higher level components to define the axis requirements for that case.

##### 3.1.1 Char.CharacterizationAxis

The Characterization object contains a set of one or more CharacterizationAxis objects.

#### 3.2 CharacterizationAxis (CharAxis)

Each CharacterizationAxis describes the axis, and contains other objects providing a full description of the axis properties.

We define a set of specialized CharacterizationAxis objects which provide a view into the object tree which is appropriate for a particular 'domain'. These specializations establish default units, denote interpretation nuances of various elements, and restrict the set of elements which are appropriate for that domain.

The following named instances may be used for defining axis requirements for specific use cases.

**FluxAxis** - instance of FluxCharAxis.

**SpatialAxis** - instance of SpatialCharAxis.

**SpectralAxis** - instance of SpectralCharAxis.

**TimeAxis** - instance of TimeCharAxis.

This listing is roughly equivalent to the Characterization Model "Property versus Axis" Table. Of these, SpatialAxis, TimeAxis, and SpectralAxis are consistent with the Characterization model. FluxAxis is added for this use case.

### 3.2.1 CharAxis.name: String

Free form name of the axis.

### 3.2.2 CharAxis.ucd: String

Defines the specific physical quantity being covered by the axis. There are many different physical quantities which may be covered by a single axis definition. It is this UCD field which is used to unambiguously distinguish them.

### 3.2.3 CharAxis.unit: String

Default unit for the axis.

### 3.2.4 CharAxis.CalibStatus: Enum

Calibration flag for the axis. Allowed values are:

- "UNCALIBRATED" - indicates that although the values reflect a measurement of the given UCD, they are not in units which can be directly compared with other data.  
Such values may be useful in the case of a spectrum with ABSOLUTE calibration on the wavelengths but UNCALIBRATED fluxes; the wavelengths of discontinuous features such as spectral lines can be measured on the assumption that the missing calibration function has no sharp discontinuities in the region of interest.
- "CALIBRATED" - indicates that values are expected to be correct within the given uncertainty.
- "ABSOLUTE" - same as CALIBRATED; include for backward compatibility with previous versions of this document.
- "RELATIVE" - indicates that although an unknown systematic error is present, the ratio of any two values will be correct.

- "NORMALIZED" - indicates that the data have been divided by a reference data set.

NOTE: This is consistent with the Characterisation model element ChAxis.CalibrationStatus.

### 3.2.5 CharAxis.Accuracy

Accuracy object associated with this axis.

### 3.2.6 CharAxis.CoordSys

We override the Characterization model definition of this element with that provided in this document.

There is significant overlap in content between this element, and the CoordSys element defined on the top level Sequence. We, therefore, allow this element to be optional.

If present, it must be defined independently from the top level CoordSys, and contain only Frames which are relevant to the particular axis to which it is associated. The Frame definitions may be simple references to a corresponding definition in the top level CoordSys.

### 3.2.7 CharAxis.Coverage

Coverage object associated with this axis.

### 3.2.8 CharAxis.Resolution

Resolution object associated with this axis.

### 3.2.9 CharAxis.SamplingPrecision

SamplingPrecision object associated with this axis.

## 3.3 Accuracy

Accuracy model. This is a simplified version of the Accuracy object described in the Characterization model.

NOTE: Omitting accuracy information indicates that the errors are unknown. Data providers are STRONGLY encouraged to provide explicit error measures whenever possible.

### Coordinate Bins

We can express the bandpass for each spectral bin as a low and high value for the spectral coordinate, or as a width. The same is done for photometry points, which amounts to approximating a filter by a rectangular bandpass. Time bins are also given as low and high values or as a width.

### Uncertainties

In addition to the binning, we allow the model to express uncertainties (which may be larger than the bin width), both statistical and systematic.

We allow one or two-sided statistical errors but only one-sided systematic errors. You can specify StatErr, or StatErrHigh/StatErrLow, but not both.

Statistical errors have the same units as the data, while systematic errors are dimensionless fractions (e.g. a 5 percent systematic error is expressed as 0.05).

For position, we have a single statistical error - a two-sided error doesn't make sense for a 2D coordinate.

For fluxes, we include plus and minus flux errors. The errors are understood as 1 sigma gaussian errors which are uncorrelated for different points in the spectrum.

**CLARIFICATION:** the two-sided errors StatErrLow and StatErrHigh are the plus/minus ERRORS, not the (value+error, value-error). In other words, if Value = 10 and there is a symmetric uncertainty of 3, the ErrorLow and ErrorHigh are both +3.0, and NOT 7.0, 13.0. This is different from the sampling description BinLow and BinHigh, which give the VALUES at the low and high end of the bin. Thus if the central wavelength of the bin is 4200.0, and the bin size is 10, then the BinLow and BinHigh values are 4195.0, 4205.0 and NOT 10.0, 10.0. Note that because of this, 0.0 is NOT an acceptable default for BinLow and BinHigh, while it IS acceptable (albeit unlikely) for StatErrLow and StatErrHigh.

### Expressing Upper Limits

If the data provider has only upper limit information, it should be represented by setting the value and the lower error value equal to the limit, and the upper error value equal to zero (e.g. 5 (+0,-5)).

#### 3.3.1 Accuracy.BinSize: Double

Bin width

#### 3.3.2 Accuracy.BinLow: Double

Minimum bin limit

#### 3.3.3 Accuracy.BinHigh: Double

Maximum bin limit

#### 3.3.4 Accuracy.StatErrSize: Double

Error width

#### 3.3.5 Accuracy.StatErrLow: Double

Low Error

#### 3.3.6 Accuracy.StatErrHigh: Double

Upper Error

#### 3.3.7 Accuracy.SysError: Double

systematic error value, assumed constant across a given spectrum and fully correlated (so that, e.g. it does not enter into estimating spectral slopes).

### 3.3.8 Accuracy.QualityStatus: Integer

This field represents the quality of the corresponding value as a positive integer, with the following meanings:

**0** - The value is considered good.

**1** - The value is considered bad for an unspecified reason. (e.g., no data in the sample interval).

**other** - Other positive integers greater than one (1) may be used to flag data which is bad or dubious for specific reasons. NOTE: The QualityCode object may be used to provide a meaning for the quality value.

Note: Bitmasks, used in some archives, such as SDSS, should be remapped to such independent Quality fields.

## 3.4 Coverage

Describes the region of the axis domain from which the data were taken.

These regions are defined by progressively more accurate descriptions. For this model, we support the following subset of (3 of 4) levels defined in the Characterization model.

### 3.4.1 Coverage.Location

### 3.4.2 Coverage.Bounds

### 3.4.3 Coverage.Support

## 3.5 Location

Gives a single characteristic point This object has been simplified from the Characterization model description where it is an stc:astroCoordType. Here, Location is collapsed to a simple set of values.

### 3.5.1 Location.Value: Double

Defines the center point of the coverage along an axis. The number of values depends on the dimensionality of the axis type. For example, a SpatialAxis should provide two (2) values, while TimeAxis need only provide one (1).

## 3.6 Bounds

Specifies the range of the axis domain which the data spans.

### 3.6.1 Bounds.Extent: Double

Width of the coverage, centered on Coverage.Location

### 3.6.2 Bounds.Start: Double

Lower boundary of the coverage. Represents the minimum value along that axis. Fore example, the observation start time, or minimum spectral value.

### 3.6.3 **Bounds.Stop: Double**

Upper boundary of the coverage. Represents the maximum value along that axis. For example, the observation end time, or maximum spectral value.

## 3.7 **Support**

Gives the detailed spatial field of view footprint, on/off time ranges (including gaps) and spectral ranges. NOTE: The Characterization model provides for an array of start-stop pairs indicating data accumulated over a series of intervals. This is not supported in this model.

### 3.7.1 **Support.Extent: Double**

### 3.7.2 **Support.Area: String**

Full and accurate description of the footprint. We do not allow a full STC region description. Our simplified model allows for the area to be expressed as either:

circle: circle x0 y0 r

polygon: polygon x1 y1 x2 y2 x3 y3 ...

The aperture field is important to determine what part of an extended object is contributing to the spectrum;

For a slit spectrum, the effective aperture on the sky is usually the slit width in the cross-dispersion direction, while for a fiber it may be a circular region. Note that since the goal of the VO Spectrum description is to describe the data as it is now, not to describe where it came from, our 'aperture' is always the effective extraction aperture, not the original instrument aperture if that is different.

## 3.8 **Resolution**

A simplified resolution model consisting of a single number characterizing the resolution of the axis.

### 3.8.1 **refVal: Double**

A typical or average value for the resolution.

## 3.9 **SamplingPrecision**

### 3.9.1 **SampleExtent: Double**

### 3.9.2 **SamplingPrecision.SamplingPrecisionRefVal.FillFactor: Double**

Gives the filling factor, a statistical way of indicating that an axis is only partly sampled. We define this as a simple numerical value between 0 and 1, where 1 means fully sampled.

The Characterization model provides a more detailed SamplingPrecision tree. While we use only part of this, we retain the full length field names for compatibility.



### 3.10 SpatialCharAxis

Denoted by the UType node "SpatialAxis", this specialization of the general CharacterizationAxis provides a view of the object space appropriate for the Spatial domain. Elements which are not visible in this domain are indicated by '(n/a)' for "Not Applicable". These elements do not have a corresponding UType listed in Appendix A and should not be expected to be supported by implementations.

Model Element	Note
Name	Axis name, default 'Sky'
unit	default unit 'deg'
ucd	default ucd 'pos.eq'
Accuracy	
BinSize	(n/a)
BinLow	(n/a)
BinHigh	(n/a)
StatError	astrometric statistical error
StatErrLow	(n/a)
StatErrHigh	(n/a)
SysError	systematic error
CalibStatus	level of coordinate calibration
Coverage	
Location.Value	central position
Bounds	
Extent	aperture angular size in decimal deg.
Start	(n/a)
Stop	(n/a)
Support	
Extent	field of view area.
Area	full description of aperture. positions must be decimal degrees. in the same coordinate system as Spatial Coord.
Resolution	spatial resolution
SamplingPrecision	
SampleExtent	pixel size in decimal degrees.
RefVal.FillFactor	statistical correction for gaps between active pixels

### 3.11 SpectralCharAxis

Denoted by the UType node "SpectralAxis", this specialization of the general Characterization-Axis provides a view of the object space appropriate for the Spectral domain. Elements which are not visible in this domain are indicated by '(n/a)' for "Not Applicable". These elements do not have a corresponding UType listed in Appendix A and should not be expected to be supported by implementations.

Model Element	Note
Name	Axis name
unit	must be defined by user
ucd	must be defined by user
Accuracy	
BinSize	bin size of Spectral Coord.
BinLow	(n/a)
BinHigh	(n/a)
StatError	measurement error in Spectral Coord.
StatErrLow	(n/a)
StatErrHigh	(n/a)
SysError	measurement error in Spectral Coord.
CalibStatus	level of coordinate calibration
Coverage	
Location.Value	central wavelength
Bounds	
Extent	same units as Spectral Coord.
Start	same units as Spectral Coord.
Stop	same units as Spectral Coord.
Support	
Extent	Effective width; same units as Spectral Coord.
Area	(n/a)
Resolution	FWHM in same units as Spectral Coord.
SamplingPrecision	
SampleExtent	bin size of Spectral Coord.
RefVal.FillFactor	unknown usage

### 3.12 TimeCharAxis

Denoted by the UType node "TimeAxis", this specialization of the general CharacterizationAxis provides a view of the object space appropriate for the Temporal domain. Elements which are not visible in this domain are indicated by '(n/a)' for "Not Applicable". These elements do not have a corresponding UType listed in Appendix A and should not be expected to be supported by implementations.

Model Element	Note
Name	Axis name, default 'Time'
unit	default unit 'd'
ucd	default ucd 'time'
Accuracy	
BinSize	bin size of Time Coord.
BinLow	(n/a)
BinHigh	(n/a)
StatError	measurement error in Time Coord.
StatErrLow	(n/a)
StatErrHigh	(n/a)
SysError	measurement error in Time Coord.
CalibStatus	level of coordinate calibration
Coverage	
Location.Value	Exposure midpoint
Bounds	
Extent	Total elapsed time (Stop - Start)
Start	Start Time
Stop	Stop Time
Support	
Extent	effective exposure time. The total length of all observing intervals times any statistical dead-time filling factor.
Area	(n/a)
Resolution	Time resolution.
SamplingPrecision	
SampleExtent	bin size of Time Coord
RefVal.FillFactor	dead time correction

### 3.13 FluxCharAxis

Denoted by the UType node "FluxAxis", this specialization of the general CharacterizationAxis provides a view of the object space appropriate for the Observable domain. Elements which are not visible in this domain are indicated by '(n/a)' for "Not Applicable". These elements do not have a corresponding UType listed in Appendix A and should not be expected to be supported by implementations.

Model Element	Note
Name	Axis name, default 'Flux'
unit	must be defined by user
ucd	must be defined by user
Accuracy	
BinSize	(n/a)
BinLow	(n/a)
BinHigh	(n/a)
StatError	measurement error
StatErrLow	(n/a)
StatErrHigh	(n/a)
SysError	measurement error
CalibStatus	level of coordinate calibration
Coverage	
Location.Value	(n/a)
Bounds	
Extent	(n/a)
Start	(n/a)
Stop	(n/a)
Support	
Extent	(n/a)
Area	(n/a)
Resolution	(n/a)
SamplingPrecision	
SampleExtent	(n/a)
RefVal.FillFactor	(n/a)

## 4 STC Data Model Elements

**Notes on compatibility with, and differences from, STC 1.33:** The CoordSys and Frame hierarchy used by the Characterisation model stem from the STC model. The Frame object hierarchy described here closely follows that model. The definitions are generally simplifications of the corresponding object of the STC model. Notable differences include:

- The specialized Frames are all extensions of the derived CoordFrame object rather than the base Frame object. So the specialized frames have a parent-child relationship with the generic frame, rather than a sibling relationship. This enables the desired, (and intended), usage in the CoordSys object.
- We add the extra Redshift attribute in the SpectralFrame, instead of the more complex CustomReferencePosition approach used in STC.
- In STC's XML serialization, the frame types and reference positions are enumerated defined elements. Here they are strings, and we require that the frame name is the frame type.
- We don't explicitly include the coordinate flavor.

**All CoordSys values are optional** , but data providers should take special care to check whether or not the defaults are appropriate for their data. The implications of the defaults are:

- Positions are given in ICRS RA,Dec in degrees and are heliocentric values (i.e. corrected for annual parallax and aberration, as normally found in source catalogs).
- Times are given in MJD days and represent times of photon arrival at the telescope.
- Spectral coordinates are as observed at the telescope, and not corrected for redshift, the motion of the Earth, etc.

### 4.1 Frame

This is an abstract base for the Coordinate Frame objects which must be extended to provide particular context and add attributes relevant a specific coordinate frame. It corresponds to the Frame object (`stc::coordFrameType`) of the STC model and has a similar definition.

#### 4.1.1 Frame.Name:String

A free form string 'label' for the frame.

#### 4.1.2 Frame.ID:String

A free form string ID enabling references to a particular frame instance from other objects.

#### 4.1.3 Frame.UCD:String

UCD string providing contextual meaning to the frame.

## 4.2 CoordFrame

The CoordFrame object is a simple extension to the base Frame object. It is used for specifying non-specialized coordinate frame types. It corresponds to the STC model CoordFrame object (stc::genericCoordFrameType), but with a simplified content definition.

### 4.2.1 CoordFrame.ReferencePosition:String

The origin of the coordinate system (and thus also its rest frame). The value of this field is restricted to the set of values shown in Table 1.

## 4.3 RedshiftFrame

Extension of the CoordFrame object used to provide information regarding the conversion of the spectral coordinate to velocity or redshift. Also known as the Velocity frame.

We emphasize that the name redshift does not imply that blueshifts are excluded, merely that, in both galactic and extragalactic astronomy, when a shift is interpreted as a velocity a positive value indicates a shift to the red. The concept of Redshift frame includes both cosmological and local Doppler velocities.

Note: RedshiftFrame should only be used if measuring things in velocities; a rest-frame spectrum of a redshifted quasar whose spectral axis is in Angstroms will be described by a SpectralFrame. The reason we have BOTH SpectralFrame and RedshiftFrame is to support certain data products, particularly used in spectral line radioastronomy, in which a spectrum (possibly obtained in piecewise spectral regions) is refactored into a set of separate spectral segments centered on different spectral lines; each segment is assigned a velocity axis centered on that line (and the same pixel from the original spectrum can appear in multiple segments each with a different velocity coordinate); you then consider the data as a 2D array with a spectral axis (indexing the segments) and a velocity axis (for each segment/spectral line).

Other coordinate system information needed for velocity spectral coordinates include the observation-fixed spectral frame, the observatory location, the source redshift, and the velocity zero point (in Greisen et al, SSYSOBS, OBSGEO, VELOSYS, RESTFRQ/RESTWAV). However, we omit these in the current model.

### 4.3.1 RedshiftFrame.DopplerDefinition: Enum

This is the Doppler Definition, allowed values are:

- "optical"
- "radio"
- "relativistic"
- "pseudo-relativistic" - [synonym for relativistic]

Comparisons to these values should not be case sensitive.

## 4.4 SpaceFrame

Extension of the CoordFrame object used to provide a spatial coordinate frame.

- The `SpaceFrame.Name` field is restricted to the set of values shown in Table 3. NOTE: This model does not support the use of `CUSTOM` or `GEO_D` as it does not accommodate their parameters.

#### 4.4.1 `SpaceFrame.Equinox`: Double

This field provides the equinox of the frame and only has meaning for frames in Table 3 which indicate "Equinox" as a parameter.

### 4.5 SpectralFrame

Extension of the `CoordFrame` object used to provide a spectral coordinate frame.

The spectral frame is defined by its `ReferencePosition` field. Once the choice of wavelength versus frequency or energy has been made, the only free parameter is the location at which the spectrum would have the given spectral coordinates. For directly observed data this is the topocenter (location of the observation); spectra may be velocity-corrected to a given velocity frame, which may be defined by the location which is at rest in that velocity frame (e.g. the heliocenter). Strictly, the correction may not be just a velocity shift, but any kind of spectral shift including e.g. gravitational redshifts; it is still true that such a shift corresponds to a location (e.g. surface of a white dwarf star) that can be quoted as a reference position.

- The `SpectralFrame.Name` field will not be considered significant for the Spectral Frame since the frame is defined by its `ReferencePosition`. We suggest that it may be filled by the name of the spectral coordinate, using names such as 'WAVE', 'FREQ' or 'ENER'.

#### 4.5.1 `SpectralFrame.Redshift`: Double

The spectral frame has a `Redshift` attribute to specify a rest frame. It is used only if the frame's `ReferencePosition` is "CUSTOM". This redshift is measured in dimensionless units, defined as  $\Delta\lambda/\lambda$  and may be negative. No specific interpretation of the shift as a cosmological or velocity shift effect is implied. [We note for the record that some co-authors object to using the word 'redshift' in this generic sense.]

### 4.6 TimeFrame

Extension of the `CoordFrame` object used to provide a Time frame.

The `TimeFrame` is defined by the frame `Name` and `ReferencePosition` fields.

- The `TimeFrame.Name` field is restricted to the set of allowed values shown in Table 2.

#### 4.6.1 `TimeFrame.Zero`: Double

Zero point of timescale, in MJD.

The STC document allows other date format possibilities, including JD and the string format ISO-8601. This model requires the use of MJD as the time type for absolute times. Relative times may be given in other units, relative to the `TimeFrame.Zero` value.

One standard reference time in astronomy is the origin of Julian Day Number on the TT (Terrestrial Time) timescale, BC 4713 Nov 24 at 11:59:27.81 (Gregorian). Using TT is preferable to UTC because it does not contain leap seconds, so the elapsed time in days is just equal to the difference in JD values.

However, the IVOA preferred format for representing Dates (ISO-8601) does not support dates before AD 1, so cannot express this reference time. Therefore, we feel the ISO standard is not a suitable format for representations of reference times.

Non-default choices of reference time may be specified in external serializations by a date in ISO-8601 format, e.g. "2004-11-30T11:59:00.01".



## 5 IVOA Conventions

### 5.1 Model Reuse and UTypes

In lieu of IVOA standards defining the mechanism for combining elements from multiple models in a single document, we include this section describing the policy used for this model.

#### 5.1.1 Model Prefix (Namespace)

A single "namespace" is to be used for all elements of the Spectral Data Model. This includes elements which have been incorporated from other IVOA models. This namespace is identified by the use of a specific prefix on the UType strings for model elements.

This document defines a DataModel object which enables data providers to declare the model(s) used, and any associated metadata such as the URL at which the corresponding XML schema may be found.

Extensions to the Spectral Data Model should define a unique prefix to be used to tag elements from those models.

Client software must, at a minimum, support the specified prefix string for the primary model (e.g. the 'spec' prefix for Spectrum files).

#### 5.1.2 Model Reuse

This model incorporates elements from the Characterization Data Model and the Photometry Data Model. The policy used for encapsulating these elements is to extend the UTypes for elements defined in this model with those of the source model. The extended UType string will be tagged with the appropriate namespace from this document.

For example, the FluxFrame object defined in this model contains a PhotCal object which is defined in the Photometry Data Model. The UType to specify the "identifier" field of the PhotCal object in a Spectrum instance is "spec:CoordSys.FluxFrame.PhotCal.identifier".

#### 5.1.3 UTypes

Characterization: The named instance tags (FluxAxis, TimeAxis, etc.) denote specific instances of CharacterizationAxis. These tag names are used as synonyms for the CharacterizationAxis portion of the relevant UType.

### 5.2 Units

This model requires the use of the IVOA VOUnits Standard[11] for representing units of physical quantities. This standard reconciles common practices and current standards for use within the IVOA community.

The OGIP convention recommended by previous versions of this document is highly compatible with this standard.

Example unit strings provided in this document (Appendix B), are given according to "best practices" recommendations in the standard. These represent the minimum set of units which users can expect applications to support. Applications are required to support these unit strings in any style allowed by the standard. Applications should support the use of alternate prefixes (e.g. "mJy" vs "Jy"), and component order.

## Osuna-Sagado representation

Pedro Osuna and Jesus Salgado have proposed a representation in the spirit of dimensional analysis (Osuna-Salgado 2005)[12]. This system uses the symbols M, L, T to signify kg, m, s respectively and omits the syntax for powers, (e.g. \*\*) so that:

10\*\*3 Jy Hz

which is equivalent to

10\*\*-23 kg s\*\*-2

is written compactly as

1.E-23 MT-2

This alternate representation is supported only for the set of 'SI' fields of the Sequence object, namely, TimeSI, FluxSI, SpectralSI.

## Recommendations

Until IVOA generic unit conversion software is mature and widely deployed, it is helpful to interoperable applications to include a representation of the units in "base SI form", including only the base units kg, m, s (and possibly A, sr) with a numeric prefix.

Although the spectral model is flexible enough to permit different units for each field, as a matter of style we strongly recommend that whenever possible the same units should be used for compatible fields (e.g. flux and error on flux).

## Notes

We note the distinction between the unit **count** (an instrumental value) and the unit **photon** (used in the photon number flux, i.e. the number of photons incident; photon number flux = energy flux divided by photon energy).

## 5.3 UCDs

UCDs or Unified Content Descriptors are the IVOA's standardized vocabulary for astronomical concepts. In this document we use UCDs as field attributes to distinguish alternate physics within the same data model roles. For example, to distinguish frequency versus wavelength on the spectral coordinate axis.

Appendix A includes UCD values for most model elements. For each axis, there are several fields whose UCDs depend on that of the axis itself. For example, the FluxAxis value and the statistical error on that value. Since there are multiple options for the axis UCD, we use the placeholder notation "[Axis.ucd]" in the dependent UCDs. For these fields, the UCD for the Axis value should be substituted in that place.

For example, if the FluxAxis value uses the UCD "phot.flux.density;em.wl", the UCD for the statistical error on that value would be "stat.error;phot.flux.density;em.wl". if the FluxAxis value uses "phot.count", then the UCD for the statistical error would be "stat.error;phot.count".

**The literal string "[FluxAxis.ucd]", for example, in a UCD field is invalid.**

The list of valid UCDs used in this model is taken from "The UCD1+ controlled vocabulary" [9] (UCD List), with syntax defined in "An IVOA Standard for Unified Content Descriptors" [10].

The concept of the " $\nu L_\nu$ " or " $\lambda L_\lambda$ " luminosity flux, or equivalently the luminosity per logarithmic energy interval  $L(\log \nu)$ , is a distinct concept in the world of spectral energy distributions - and it's a different concept from the bolometric luminosity, which has the same units. The UCD board has not yet approved a UCD expressing this concept; we have to use `phys.luminosity` and infer the concept from the units.

## 5.4 Extensibility

There is no IVOA wide convention for defining how users may extend models with their own content. Here, we define the mechanism by which users may add content to an instance of the Spectral Data Model.

### 5.4.1 Namespace

All user-defined content, (not modeled by this document), must be contained under a separate unique prefix tag. The full description should be stored in a DataModel object on Dataset. Users should take care not to make use of prefix tags which are associated with current IVOA standards, (e.g. 'cha', 'spec', 'ssa', 'stc'). At the time of this writing, there is no central repository of reserved namespace strings.

### 5.4.2 UTypes

As the content is user-defined, we impose no restriction on the UType content. Users may choose to extend Spectral UTypes as described in Section 5.1, or generate an independent UType hierarchy.

### 5.4.3 Scope

We permit any object modeled in this document to be extended with user-defined content, with the following restrictions:

- Extended content must respect the inherent nature of the object being extended. That is, for example, users may not extend objects which are defined to be constant for a Sequence (e.g. DataID) with content which varies on a per-point basis.
- Values of extended content must be consistent with the content of modeled data. That is, scalar values of standard base data types (e.g. String, Double, Integer, Boolean).
- Since extended content, by definition, does not follow this model, it is not possible for general Spectral Model applications to interpret complex structures within that content. It is, therefore, recommended that users define extended content in such a way as to avoid ambiguity between its components. For example, defining multiple instances of a complex object with identical UType sets.

### 5.4.4 Support

General Spectral Model applications should, but are not required to, provide the following support for extended content:

- Retain existence of extended content, including namespace and UTypes.
- Retain association with modeled component.
- Provide access to extended content by users.

#### **5.4.5 Note**

The above mechanism could also be used for applications to retain and 'support' versions of the Spectral model other than their primary version. Unrecognized content under a Spectral Model namespace (e.g. 'spec') could be treated in the same manner as user-defined content as described above.

## 6 Tables

Token	Meaning	Note
UNKNOWN	Unknown origin	
RELOCATABLE	Relative origin	Suitable for simulations
CUSTOM	Origin specified wrt another system	
TOPOCENTER	Location of the observing device	(telescope)
BARYCENTER	Solar system barycenter	
HELIOCENTER	Center of the Sun	
GEOCENTER	Center of the Earth	
EMBARYCENTER	Earth-Moon barycenter	
MOON	Center of the Moon	
MERCURY	Center of Mercury	
VENUS	Center of Venus	
MARS	Center of Mars	
JUPITER	Center of Jupiter	
SATURN	Center of Saturn	
URANUS	Center of Uranus	
NEPTUNE	Center of Neptune	
PLUTO	Center of Pluto	
LSRK	Kinematic local standard of rest	Redshift frame only
LSRD	Dynamic local standard of rest	Redshift frame only
GALACTIC_CENTER	Center of the Galaxy	
LOCAL_GROUP_CENTER	Barycenter of the Local Group	

Table 1: Allowed values for CoordFrame.ReferencePosition

Token	Meaning	Note
LOCAL	Relocatable (simulation) time	
TT	Terrestrial Time	
UTC	Coordinated Universal Time	
ET	Ephemeris Time	
TDB	Barycentric dynamical time	
TCG	Terrestrial Coordinate Time	
TCB	Barycentric Coordinate Time	
TAI	International Atomic Time	
LST	Local Sidereal Time	

Table 2: Allowed values for TimeFrame.Name

Token	Meaning	Parameter(s)
UNKNOWN	Unknown frame	
CUSTOM	Custom frame	Pole, axis
AZ.EL	Azimuth and elevation	
BODY	Generic body (eg planet)	
ICRS	The ICRS frame	
FK4	FK4	Equinox
FK5	FK5	Equinox
ECLIPTIC	Ecliptic I,b	Equinox
GALACTIC_I	Old galactic LI,BI	
GALACTIC_II	Galactic LII,BII	
SUPER_GALACTIC	SGL, SGB	
MAG	Geomagnetic ref frame	
GSE	Geocentric Solar Ecliptic	
GSM	Geocentric Solar Magnetic	
SM	Solar Magnetic	
HGC	Heliographic	
HEE	Heliocentric Earth Ecliptic	
HEEQ	Heliocentric Earth Equatorial	
HCI	Heliocentric Inertial	
HCD	Heliocentric of Date	
GEO_C	Geocentric corotating	
GEO_D	Geodetic ref frame	Spheroid
MERCURY_C	Corotating planetocentric	
VENUS_C	Corotating planetocentric	
LUNA_C	Corotating planetocentric	
MARS_C	Corotating planetocentric	
JUPITER_C_III	Corotating planetocentric	
SATURN_C_III	Corotating planetocentric	
URANUS_C_III	Corotating planetocentric	
NEPTUNE_C_III	Corotating planetocentric	
PLUTO_C	Corotating planetocentric	
MERCURY_G	Corotating planetographic	
VENUS_G	Corotating planetographic	
LUNA_G	Corotating planetographic	
MARS_G	Corotating planetographic	
JUPITER_G_III	Corotating planetographic	
SATURN_G_III	Corotating planetographic	
URANUS_G_III	Corotating planetographic	
NEPTUNE_G_III	Corotating planetographic	
PLUTO_G	Corotating planetographic	

Table 3: Allowed values for SpaceFrame.Name

## 7 Spectrum Data Model

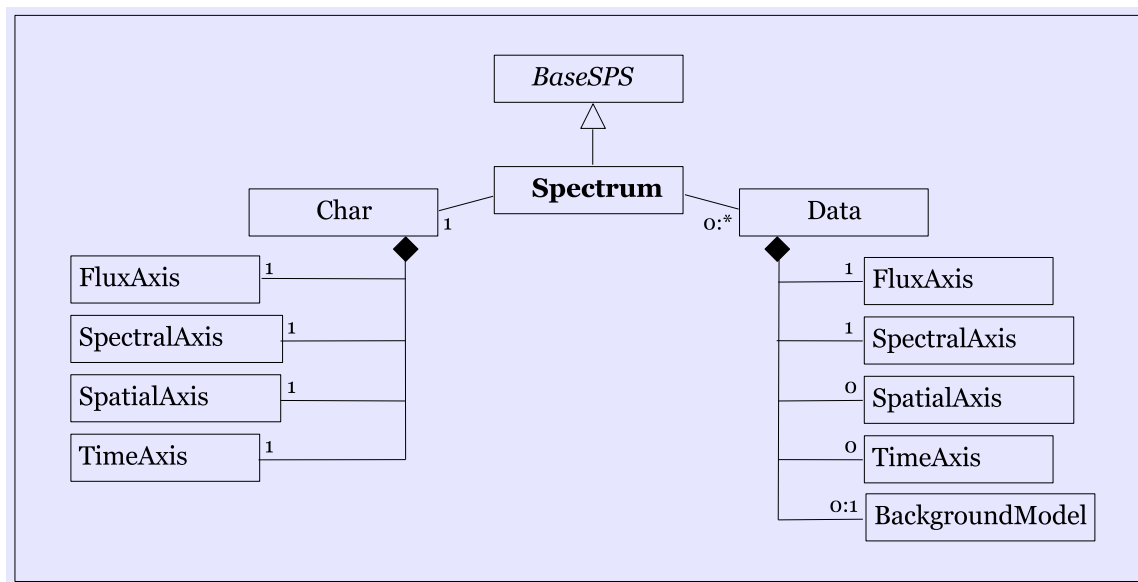


Figure 8: Spectrum Data Model Extension Axis Requirements.

### 7.1 Spectrum

An extension of BaseSPS class. This object has no additional attributes, but does provide context which allows the contents to be interpreted as a Spectrum object.

#### 7.1.1 Model Prefix

The following model prefix is to be associated with objects following this model. Its usage should conform to the current IVOA Convention as given in Section 5.1.1

Model Prefix Specification - Spectrum Data Model	
"spec"	Spectrum Data Model instance

#### 7.1.2 Axis Requirements

Here we define the axis requirements for a Spectrum instance in terms of the named specializations described in the Spectral model text for CharAxis and DataAxis.

Flavor	Char Requirement	Data Requirement
SpectralAxis	Exactly one must be defined.	Exactly one must be defined.
FluxAxis	Exactly one must be defined.	Exactly one must be defined.
TimeAxis	Exactly one must be defined.	None may be defined.
SpatialAxis	Exactly one must be defined.	None may be defined.
BackgroundModel	N/A	Exactly one may be defined.

#### 7.1.3 UTypes

We define the UType "Spectrum" for use, as needed, to tag the top level container node in serializations.



For all other fields, the Spectrum Model uses the same UType list as the Spectral Model. See Appendix A for the complete list.

#### 7.1.4 UCDs and Units

In addition to UTypes, Appendix A provides the appropriate UCD string to use for various model fields. For FluxAxis and SpectralAxis, there are several possible UCDs which apply. Appendix B defines the set of acceptable UCDs for those axes along with a set of units associated with each UCD option. These are the minimal set of units which users should expect to be handled by applications.

#### 7.1.5 Required Fields

The Spectrum model uses the base field requirements and default values given in Appendix A except as noted below.

Field	Req	Default
Dataset.DataModel.Name	MAN	"Spectrum-2.0"
Dataset.DataModel.Prefix	OPT	"spec"
Dataset.Type	OPT	"Spectrum"
Target.Name	MAN	
Char.SpectralAxis.ucd	MAN	
Char.SpectralAxis.unit	MAN	
Char.TimeAxis.Coverage.Bounds.Extent	MAN	
Char.TimeAxis.Coverage.Bounds.Start	MAN	
Char.TimeAxis.Coverage.Bounds.Stop	MAN	
Data.SpectralAxis.Value	MAN	

NOTE: For TimeAxis.Coverage.Bounds, it is mandatory to supply EITHER the extent OR start/stop pair.

## 8 Photometry Point Data Model

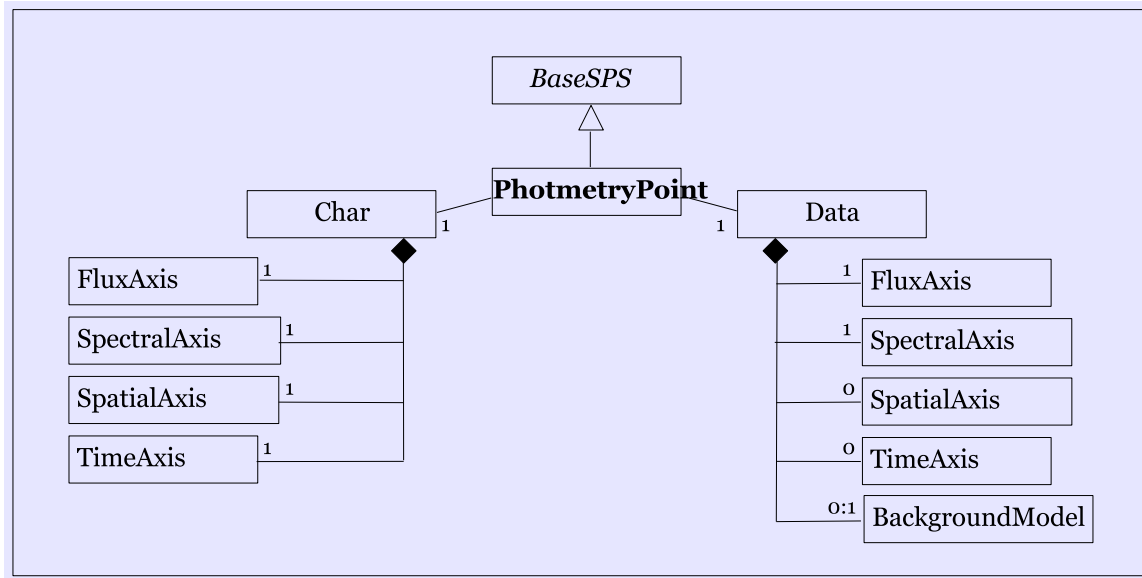


Figure 9: PhotometryPoint Data Model Extension Axis Requirements.

### 8.1 PhotometryPoint

An extension of BaseSPS class to represent photometry point data.

#### 8.1.1 Model Prefix

The following model prefix is to be associated with objects following this model. Its usage should conform to the current IVOA Convention as given in Section 5.1.1

Model Prefix Specification - PhotometryPoint Data Model	
"phot"	Photometry Point Data Model instance

#### 8.1.2 Axis Requirements

Here we define the axis requirements for a PhotometryPoint instance in terms of the named specializations described in the Spectral model text for CharAxis and DataAxis.

Flavor	Char Requirement	Data Requirement
SpectralAxis	Exactly one must be defined.	Exactly one must be defined.
FluxAxis	Exactly one must be defined.	Exactly one must be defined.
TimeAxis	Exactly one must be defined.	None may be defined.
SpatialAxis	Exactly one must be defined.	None may be defined.
BackgroundModel	N/A	Exactly one may be defined.

#### 8.1.3 UTypes

We define the UType "PhotometryPoint" for use, as needed, to tag the top level container object in serializations.

For all other fields, the PhotometryPoint Model uses the same UType list as the Spectral Model. See Appendix A for the complete list.

### 8.1.4 UCDs and Units

In addition to UTypes, Appendix A provides the appropriate UCD string to use for various model fields. For FluxAxis and SpectralAxis, there are several possible UCDs which apply. Appendix B defines the set of acceptable UCDs for those axes along with a set of units associated with each UCD option. These are the minimal set of units which users should expect to be handled by applications.

### 8.1.5 Required Fields

The PhotometryPoint model uses the same requirement set and default values as the Spectral model except as noted below:

Field	Req	Default
Dataset.DataModel.Name	MAN	"PhotometryPoint-1.0"
Dataset.DataModel.Prefix	OPT	"phot"
Dataset.Type	OPT	"PhotometryPoint"
Target.Name	MAN	
Char.SpectralAxis.ucd	MAN	
Char.SpectralAxis.unit	MAN	
CoordSys.FluxFrame.PhotCal.ZeroPoint.Flux.Value	MAN	
CoordSys.FluxFrame.PhotCal.ZeroPoint.Flux.Unit	MAN	
Data.SpectralAxis.Value	MAN	

### 8.1.6 Restrictions

For PhotometryPoint, there may be only 1 Data point per Sequence.

## A Spectral Data Model UTypes

Spectral Data Model UType List

Field	Datatype	UCD1+	Meaning	Req	Default
Char			node		
Char.FluxAxis			node		
Char.FluxAxis.Name	String		name for flux axis	OPT	Flux
Char.FluxAxis.ucd	String	meta.ucd	ucd for flux	MAN	(N/A)
Char.FluxAxis.unit	String	meta.unit	Unit for flux	MAN	(N/A)
Char.FluxAxis.CalibStatus	Enum String	meta.code.qual	Type of coord calibration	OPT	CALIBRATED
Char.FluxAxis.Accuracy			node		
Char.FluxAxis.Accuracy.StatError	Double	stat.error:[FluxAxis.ucd]	error	REC	NaN
Char.FluxAxis.Accuracy.SysError	Double	stat.error.sys:[FluxAxis.ucd]	Systematic error	REC	NaN
Char.FluxAxis.CoordSys			node		
Char.FluxAxis.CoordSys.*			Axis coordinate system	OPT	
Char.FluxAxis.QualityDefs			node		
Char.FluxAxis.QualityDefs.QualityCode			node		
Char.FluxAxis.QualityDefs.QualityCode.value	Integer		Quality code numeric value	OPT	(None)
Char.FluxAxis.QualityDefs.QualityCode.description	String		Meaning of quality value	OPT	(None)
Char.SpatialAxis			node		
Char.SpatialAxis.Name	String	meta.id	name for spatial axis	OPT	Sky
Char.SpatialAxis.ucd	String	meta.ucd	ucd for spatial coord	REC	pos.eq
Char.SpatialAxis.unit	String	meta.unit	Unit for spatial coord	REC	deg
Char.SpatialAxis.CalibStatus	Enum String	meta.code.qual	Type of coord calibration	OPT	CALIBRATED
Char.SpatialAxis.Accuracy			node		
Char.SpatialAxis.Accuracy.StatError	Double	stat.error:[SpatialAxis.ucd]	Astrometric statistical error	OPT	NaN
Char.SpatialAxis.Accuracy.SysError	Double	stat.error.sys:[SpatialAxis.ucd]	Systematic error	OPT	NaN
Char.SpatialAxis.CoordSys			node		
Char.SpatialAxis.CoordSys.*			Axis coordinate system	OPT	
Char.SpatialAxis.Coverage			node		
Char.SpatialAxis.Coverage.Location			node		
Char.SpatialAxis.Coverage.Location.Value	Double	[SpatialAxis.ucd]	Position	MAN	(N/A)
Char.SpatialAxis.Coverage.Bounds			node		
Char.SpatialAxis.Coverage.Bounds.Extent	Double	pos.angDistance;instr.fov	Aperture angular size	MAN	(N/A)
Char.SpatialAxis.Coverage.Support			node		
Char.SpatialAxis.Coverage.Support.Area	String		Aperture region	REC	UNKNOWN
Char.SpatialAxis.Coverage.Support.Extent	Double	pos.angDistance;instr.fov	Field of view area	OPT	NaN
Char.SpatialAxis.QualityDefs			node		
Char.SpatialAxis.QualityDefs.QualityCode			node		
Char.SpatialAxis.QualityDefs.QualityCode.value	Integer		Quality code numeric value	OPT	(None)
Char.SpatialAxis.QualityDefs.QualityCode.description	String		Meaning of quality value	OPT	(None)
Char.SpatialAxis.Resolution			node		
Char.SpatialAxis.Resolution.refVal	Double	pos.angResolution	Spatial resolution of data	OPT	NaN
Char.SpatialAxis.SamplingPrecision			node		
Char.SpatialAxis.SamplingPrecision.SampleExtent	Double	phys.angSize;instr.pixel	spatial bin size	OPT	(pixel size in deg)
Char.SpatialAxis.SamplingPrecision.SamplingPrecisionRefVal			node		
Char.SpatialAxis.SamplingPrecision.SamplingPrecisionRefVal.FillFactor	Double	stat.filling:[SpatialAxis.ucd]	Sampling Filling factor	OPT	1.0
Char.SpectralAxis			node		
Char.SpectralAxis.Name	String		name for spectral axis	OPT	SpectralCoord
Char.SpectralAxis.ucd	String	meta.ucd	ucd for spectral coord	REC	(N/A)
Char.SpectralAxis.unit	String	meta.unit	Unit for spectral coord	REC	(N/A)
Char.SpectralAxis.CalibStatus	Enum String	meta.code.qual	Type of coord calibration	OPT	CALIBRATED
Char.SpectralAxis.Accuracy			node		
Char.SpectralAxis.Accuracy.BinSize	Double	[SpectralAxis.ucd];spect.binSize	Wavelength bin size	OPT	NaN
Char.SpectralAxis.Accuracy.StatError	Double	stat.error:[SpectralAxis.ucd]	Spectral coord measurement error	REC	0.0
Char.SpectralAxis.Accuracy.SysError	Double	stat.error.sys:[SpectralAxis.ucd]	Spectral coord measurement error	REC	0.0
Char.SpectralAxis.CoordSys			node		
Char.SpectralAxis.CoordSys.*			Axis coordinate system	OPT	
Char.SpectralAxis.Coverage			node		

Spectral Data Model UType List (Continued)

Field	Datatype	UCD1+	Meaning	Req	Default
Char.SpectralAxis.Coverage.Location			node		
<b>Char.SpectralAxis.Coverage.Location.Value</b>	Double	[SpectralAxis.ucd];instr.bandpass	Spectral coord value	MAN	(N/A)
Char.SpectralAxis.Coverage.Bounds			node		
<b>Char.SpectralAxis.Coverage.Bounds.Extent</b>	Double	instr.bandwidth	Width in spectral coordinate	MAN	(N/A)
<b>Char.SpectralAxis.Coverage.Bounds.Start</b>	Double	[SpectralAxis.ucd];stat.min	Minimum on spectral coordinate	MAN	(N/A)
<b>Char.SpectralAxis.Coverage.Bounds.Stop</b>	Double	[SpectralAxis.ucd];stat.max	Maximum on spectral coordinate	MAN	(N/A)
Char.SpectralAxis.Coverage.Support			node		
Char.SpectralAxis.Coverage.Support.Extent	Double	instr.bandwidth	Effective width of spectrum	OPT	(N/A)
Char.SpectralAxis.QualityDefs			node		
Char.SpectralAxis.QualityDefs.QualityCode			Quality code numeric value	OPT	(None)
Char.SpectralAxis.QualityDefs.QualityCode.value	Integer		Meaning of quality value	OPT	(None)
Char.SpectralAxis.QualityDefs.QualityCode.description	String		node		
Char.SpectralAxis.Resolution			node		
Char.SpectralAxis.Resolution.refVal	Double	spect.resolution:[SpectralAxis.ucd]	Spectral resolution FWHM	OPT	self.Accuracy.BinSize
Char.SpectralAxis.Resolution.ResolPower			node		
Char.SpectralAxis.Resolution.ResolPower.refVal	Double	spect.resolution	Spectral resolving power	OPT	NaN
Char.SpectralAxis.SamplingPrecision			node		
Char.SpectralAxis.SamplingPrecision.SampleExtent	Double	[SpectralAxis.ucd];spect.binSize	Wavelength bin size	OPT	self.Accuracy.BinSize
Char.SpectralAxis.SamplingPrecision.SamplingPrecisionRefVal			node		
Char.SpectralAxis.SamplingPrecision.SamplingPrecisionRefVal.FillFactor	Double	stat.filling:[SpectralAxis.ucd]	Sampling Filling factor	OPT	1.0
Char.TimeAxis			node		
Char.TimeAxis.Name	String		name for time axis	OPT	Time
Char.TimeAxis.ucd	String	meta.ucd	ucd for time	REC	time
Char.TimeAxis.unit	String	meta.unit	Unit for time	REC	d
Char.TimeAxis.CalibStatus	Enum String	meta.code.qual	Type of coord calibration	OPT	CALIBRATED
Char.TimeAxis.Accuracy			node		
Char.TimeAxis.Accuracy.BinSize	Double	time.interval	Time bin size	OPT	NaN
Char.TimeAxis.Accuracy.StatError	Double	stat.error:[TimeAxis.ucd]	Time coord measurement statistical error	OPT	NaN
Char.TimeAxis.Accuracy.SysError	Double	stat.error.sys:[TimeAxis.ucd]	Time coord measurement systematic error	OPT	NaN
Char.TimeAxis.CoordSys			node		
Char.TimeAxis.CoordSys.*			Axis coordinate system	OPT	
Char.TimeAxis.Coverage			node		
Char.TimeAxis.Coverage.Location			node		
<b>Char.TimeAxis.Coverage.Location.Value</b>	Double	time.epoch;obs	Exposure midpoint	MAN	(N/A)
Char.TimeAxis.Coverage.Bounds			node		
<b>Char.TimeAxis.Coverage.Bounds.Extent</b>	Double	time.duration;obs.exposure	Total elapsed time	MAN	(N/A)
Char.TimeAxis.Coverage.Bounds.Start	Double	time.start;obs.exposure	Start time	REC	NaN
Char.TimeAxis.Coverage.Bounds.Stop	Double	time.end;obs.exposure	Stop time	REC	NaN
Char.TimeAxis.Coverage.Support			node		
Char.TimeAxis.Coverage.Support.Extent	Double	time.duration;obs.exposure	Effective exposure time	OPT	NaN
Char.TimeAxis.QualityDefs			node		
Char.TimeAxis.QualityDefs.QualityCode			Quality code numeric value	OPT	(None)
Char.TimeAxis.QualityDefs.QualityCode.value	Integer		Meaning of quality value	OPT	(None)
Char.TimeAxis.QualityDefs.QualityCode.description	String		node		
Char.TimeAxis.Resolution			node		
Char.TimeAxis.Resolution.refVal	Double	time.resolution	Temporal resolution FWHM	OPT	self.Accuracy.BinSize
Char.TimeAxis.SamplingPrecision			node		
Char.TimeAxis.SamplingPrecision.SampleExtent	Double	time.interval	time bin size	OPT	self.Accuracy.BinSize
Char.TimeAxis.SamplingPrecision.SamplingPrecisionRefVal			node		
Char.TimeAxis.SamplingPrecision.SamplingPrecisionRefVal.FillFactor	Double	stat.filling;time	Sampling Filling factor	OPT	NaN
CoordSys			node		
CoordSys.ID	String	meta.id	ID string for coordinate system	OPT	(None)
CoordSys.CoordFrame			node		
CoordSys.CoordFrame.ID	String	meta.id	ID string for coordinate frame	OPT	(None)
CoordSys.CoordFrame.Name	String		Name of coordinate frame	OPT	(None)
CoordSys.CoordFrame.UCD	String	meta.ucd	UCD	OPT	(None)
CoordSys.CoordFrame.RefPos	String		Frame origin	OPT	(None)

Spectral Data Model UType List (Continued)

Field	Datatype	UCD1+	Meaning	Req	Default
CoordSys.FluxFrame			node		
CoordSys.FluxFrame.ID	String	meta.id	ID string for coordinate frame	OPT	(None)
CoordSys.FluxFrame.Name	String		Name of photometric band	OPT	(None)
CoordSys.FluxFrame.UCD	String	meta.ucd	UCD for photometric calibration	OPT	phot.mag
CoordSys.FluxFrame.refID	String	meta.ref.ivorn	URI for photometric calibration	OPT	(None)
CoordSys.FluxFrame.PhotCal			node		
CoordSys.FluxFrame.PhotCal.*			Photometric Calibration UTypes		
CoordSys.FluxFrame.PhotCal.PhotmetryFilter			node		
CoordSys.FluxFrame.PhotCal.PhotmetryFilter.*			Photometry Filter UTypes		
CoordSys.RedshiftFrame			node		
CoordSys.RedshiftFrame.ID	String	meta.id	ID string for coordinate frame	OPT	(None)
CoordSys.RedshiftFrame.Name	String		Redshift frame name	OPT	(None)
CoordSys.RedshiftFrame.UCD	String	meta.ucd	Redshift frame UCD	OPT	(None)
CoordSys.RedshiftFrame.RefPos	String		Redshift frame origin	OPT	UNKNOWN
CoordSys.RedshiftFrame.DopplerDefinition	Enum String		Doppler definition	OPT	UNKNOWN
CoordSys.SpaceFrame			node		
CoordSys.SpaceFrame.ID	String	meta.id	ID string for coordinate frame	OPT	(None)
CoordSys.SpaceFrame.Name	String	pos.frame	Name of the spatial coord frame	REC	ICRS
CoordSys.SpaceFrame.UCD	String	meta.ucd	Space frame UCD	OPT	Char.SpatialAxis.ucd
CoordSys.SpaceFrame.RefPos	String		Origin of SpaceFrame	OPT	UNKNOWN
CoordSys.SpaceFrame.Equinox	Double	time.equinox;pos.frame	Equinox	OPT	2000.0
CoordSys.SpectralFrame			node		
CoordSys.SpectralFrame.ID	String	meta.id	ID string for coordinate frame	OPT	(None)
CoordSys.SpectralFrame.Name	String		Spectral frame name	OPT	(None)
CoordSys.SpectralFrame.UCD	String	meta.ucd	Spectral frame UCD	OPT	Char.SpectralAxis.ucd
CoordSys.SpectralFrame.RefPos	String	sdm:spect.frame	Spectral frame origin	OPT	TOPOCENTER
CoordSys.SpectralFrame.Redshift	Double		Spectral frame shift	OPT	0.0
CoordSys.TimeFrame			node		
CoordSys.TimeFrame.ID	String	meta.id	ID string for coordinate frame	OPT	(None)
CoordSys.TimeFrame.Name	String	sdm:time.frame	Timescale	OPT	TT
CoordSys.TimeFrame.UCD	String	meta.ucd	Time frame UCD	OPT	Char.TimeAxis.ucd
CoordSys.TimeFrame.RefPos	String	time.scale	Time frame origin	OPT	TOPOCENTER
CoordSys.TimeFrame.Zero	Double	arith.zp;time	Zero point of timescale in MJD	OPT	0.0
Curation			node		
<b>Curation.Publisher</b>	String	meta.curation	Publisher	MAN	(N/A)
Curation.PublisherID	String	meta.ref.url;meta.curation	URI for VO Publisher	OPT	UNKNOWN
Curation.Date	String		Date curated dataset last modified	OPT	UNKNOWN
Curation.Version	String	meta.version;meta.curation	Version info	OPT	UNKNOWN
Curation.Rights	Enum String	meta.code	Proprietary restrictions level	REC	public
Curation.Reference	String	meta.bib.bibcode	URL or Bibcode for documentation	REC	UNKNOWN
Curation.Contact			node		
Curation.Contact.Name	String	meta.bib.author;meta.curation	Contact name	OPT	UNKNOWN
Curation.Contact.Email	String	meta.ref.url;meta.email	Contact email	OPT	UNKNOWN
Curation.PublisherDID	URI	meta.ref.url;meta.curation	Publisher specified dataset ID	REC	DataID.DatasetID
Data			node		
Data.*Axis.Corrrections.GenericCorr			node		
Data.*Axis.Corrrections.GenericCorr.Name	String		Correction name	OPT	(None)
Data.*Axis.Corrrections.GenericCorr.Value	Double		Correction value	OPT	(None)
Data.*Axis.Corrrections.GenericCorr.Applied	Boolean	meta.code	Correction applied flag	OPT	FALSE
Data.*Axis.SystemID	String	meta.id	associated coord system id.	OPT	(None)
Data.FluxAxis			node		

Spectral Data Model UType List (Continued)

Field	Datatype	UCDI+	Meaning	Req	Default
<b>Data.FluxAxis.Value</b>		[see Appendix B]	Flux values for points	MAN	(N/A)
Data.FluxAxis.ucd		meta.ucd	ucd for flux	OPT	Char.FluxAxis.ucd
Data.FluxAxis.unit		meta.unit	Unit for flux	OPT	Char.FluxAxis.unit
Data.FluxAxis.Accuracy			node		
Data.FluxAxis.Accuracy.QualityStatus	Integer	meta.code.qual;[FluxAxis.ucd]	Quality	OPT	0
Data.FluxAxis.Accuracy.StatError	Double	stat.error;[FluxAxis.ucd]	symmetric error	OPT	Char.FluxAxis...
Data.FluxAxis.Accuracy.StatErrLow	Double	stat.error;[FluxAxis.ucd];stat.min	Lower error	OPT	NaN
Data.FluxAxis.Accuracy.StatErrHigh	Double	stat.error;[FluxAxis.ucd];stat.max	Upper error	OPT	NaN
Data.FluxAxis.Accuracy.SysError	Double	stat.error.sys;[FluxAxis.ucd]	Systematic error	OPT	Char.FluxAxis...
Data.FluxAxis.Corrections			node		
Data.FluxAxis.Corrections.ApFrac	Double	arith.ratio	Aperture fraction	OPT	1.0
Data.FluxAxis.Corrections.ApFrac.Value	Double				
Data.FluxAxis.Corrections.ApFrac.Applied	Boolean	meta.code	Aperture fraction applied	OPT	FALSE
Data.SpectralAxis			node		
Data.SpectralAxis.Value		[see Appendix B]	Spectral coordinates for points	REC	NaN
Data.SpectralAxis.ucd		meta.ucd	ucd for spectral coord	OPT	Char.SpectralAxis.ucd
Data.SpectralAxis.unit		meta.unit	Unit for spectral coord	OPT	Char.SpectralAxis.unit
Data.SpectralAxis.Accuracy			node		
Data.SpectralAxis.Accuracy.BinSize	Double	[SpectralAxis.ucd];spect.binSize	Wavelength bin size	OPT	Char.SpectralAxis...
Data.SpectralAxis.Accuracy.BinLow	Double	[SpectralAxis.ucd];stat.min	Spectral coord bin lower end	OPT	NaN
Data.SpectralAxis.Accuracy.BinHigh	Double	[SpectralAxis.ucd];stat.max	Spectral coord bin upper end	OPT	NaN
Data.SpectralAxis.Accuracy.StatError	Double	stat.error;[SpectralAxis.ucd]	Spectral coord measurement error	OPT	NaN
Data.SpectralAxis.Accuracy.StatErrLow	Double	stat.error;[SpectralAxis.ucd];stat.min	Spectral coord measurement lower error	OPT	NaN
Data.SpectralAxis.Accuracy.StatErrHigh	Double	stat.error;[SpectralAxis.ucd];stat.max	Spectral coord measurement upper error	OPT	NaN
Data.SpectralAxis.Accuracy.SysError	Double	stat.error.sys;[SpectralAxis.ucd]	Spectral coord systematic error	OPT	Char.SpectralAxis...
Data.SpectralAxis.Resolution			node		
Data.SpectralAxis.Resolution.refVal	Double	spect.resolution;[SpectralAxis.ucd]	Spectral resolution FWHM	OPT	Char.SpectralAxis...
Data.TimeAxis			node		
Data.TimeAxis.Value			Time coordinates for points	OPT	Char.TimeAxis.Location
Data.TimeAxis.ucd		meta.ucd	ucd for time	OPT	Char.TimeAxis.ucd
Data.TimeAxis.unit		meta.unit	Unit for time	OPT	Char.TimeAxis.unit
Data.TimeAxis.Accuracy			node		
Data.TimeAxis.Accuracy.BinSize	Double	time.interval	Time bin size	OPT	Char.TimeAxis...
Data.TimeAxis.Accuracy.BinLow	Double	time.start;time.interval	Time bin start	OPT	NaN
Data.TimeAxis.Accuracy.BinHigh	Double	time.end;time.interval	Time bin stop	OPT	NaN
Data.TimeAxis.Accuracy.StatError	Double	stat.error;[TimeAxis.ucd]	Time coord measurement statistical error	OPT	Char.TimeAxis...
Data.TimeAxis.Accuracy.StatErrLow	Double	stat.error;[TimeAxis.ucd];stat.min	Time coord measurement lower error	OPT	NaN
Data.TimeAxis.Accuracy.StatErrHigh	Double	stat.error;[TimeAxis.ucd];stat.max	Time coord measurement upper error	OPT	NaN
Data.TimeAxis.Accuracy.SysError	Double	stat.error.sys;[TimeAxis.ucd]	Time coord systematic error	OPT	Char.TimeAxis...
Data.TimeAxis.Resolution			node		
Data.TimeAxis.Resolution.refVal	Double	time.resolution	Temporal resolution FWHM	OPT	Char.TimeAxis...
Data.BackgroundModel			node		
Data.BackgroundModel.Value			Flux values for points	OPT	(None)
Data.BackgroundModel.ucd		meta.ucd	ucd for background flux	OPT	Char.FluxAxis.ucd
Data.BackgroundModel.unit		meta.unit	Unit for background flux	OPT	Char.FluxAxis.unit
Data.BackgroundModel.Accuracy			node		
Data.BackgroundModel.Accuracy.QualityStatus	Integer	meta.code.qual;[FluxAxis.ucd]	Quality	OPT	0
Data.BackgroundModel.Accuracy.StatError	Double	stat.error;[FluxAxis.ucd]	Symmetric error	OPT	Char.FluxAxis...
Data.BackgroundModel.Accuracy.StatErrLow	Double	stat.error;[FluxAxis.ucd];stat.min	Lower error	OPT	NaN
Data.BackgroundModel.Accuracy.StatErrHigh	Double	stat.error;[FluxAxis.ucd];stat.max	Upper error	OPT	NaN
Data.BackgroundModel.Accuracy.SysError	Double	stat.error.sys;[FluxAxis.ucd]	Systematic error	OPT	Char.FluxAxis...
DataID			node		
<b>DataID.Title</b>	String	meta.title;meta.dataset	Dataset Title	MAN	(N/A)
DataID.Creator	String	meta.curation	VO Creator ID	OPT	UNKNOWN
DataID.Collection	String		Collection name(s)	OPT	(None)



Spectral Data Model UType List (Continued)

Field	Datatype	UCDI+	Meaning	Req	Default
DataID.DatasetID	URI	meta.id;meta.dataset	IVOA Dataset ID	OPT	UNKNOWN
DataID.CreatorDID	URI	meta.id	Creator defined Dataset ID	OPT	(None)
DataID.Date	String	time.epoch;meta.dataset	Data processing/creation date	OPT	UNKNOWN
DataID.Version	String	meta.version;meta.dataset	Version of dataset	OPT	UNKNOWN
DataID.Instrument	String	meta.id;instr	Instrument ID	OPT	UNKNOWN
DataID.Bandpass	String	instr.bandpass	Band	OPT	UNKNOWN
DataID.CreationType	Enum String		Dataset creation type	OPT	Archival
DataID.Logo	URL	meta.ref.url	URL for creator logo	OPT	UNKNOWN
DataID.Contributor	String		Contributor(s)	OPT	UNKNOWN
DataID.DataSource	String		Original data type	OPT	UNKNOWN
Dataset			node		
Dataset.Type	Enum String		Dataset or segment type	OPT	Spectral
<b>Dataset.DataModel.Name</b>	String		Data model name and version	MAN	Spectral-2.0
Dataset.DataModel.Prefix	String		Data model prefix tag	REC	(None)
Dataset.DataModel.url	String	meta.ref.url	Reference URL for model	OPT	(None)
Dataset.Length	Long	meta.number	Number of points	OPT	(must be derived)
Dataset.TimeSI	String		SI factor and dimensions	REC	(None)
Dataset.SpectralSI	String		SI factor and dimensions	REC	(None)
Dataset.FluxSI	String		SI factor and dimensions	REC	(None)
Derived			node		
Derived.SNR	Double	stat.snr	Signal-to-noise for spectrum	OPT	NaN
Derived.Redshift			node		
Derived.Redshift.Value	Double		Measured redshift for spectrum	OPT	NaN
Derived.Redshift.StatError	Double	stat.error;src.redshift	Error on measured redshift	OPT	NaN
Derived.Redshift.Confidence	Double		Confidence value on redshift	OPT	NaN
Derived.VarAmpl	Double	src.var.amplitude;arith.ratio	Variability amplitude as fraction of mean	OPT	NaN
Target			node		
Target.Name	String	meta.id;src	Target name	REC	UNKNOWN
Target.Description	String	meta.note;src	Target descriptive text	OPT	UNKNOWN
Target.Class	String	src.class	Target or object class	OPT	UNKNOWN
Target.SpectralClass	String	src.spType	Object spectral class	OPT	UNKNOWN
Target.Redshift	Double	src.redshift	Target redshift	OPT	UNKNOWN
Target.Pos	Double	pos.eq;src	Target RA and Dec	REC	NaN
Target.VarAmpl	Double	src.var.amplitude	Target variability amplitude, typical	OPT	NaN

## B Spectral Data Model Axis UCDs

### B.1 Flux Axis UCDs

Flux Axis UCD Options (FluxAxis.ucd)		
UCD1+	Meaning	Unit (VOUnit style)
phot.flux.density;em.wl	Flux density per unit wave.	erg.cm**(-2).s**(-1).angstrom**(-1), W.m**(-2).m**(-1), keV.cm**(-2).s**(-1).angstrom**(-1)
phot.flux.density;em.freq	Flux density per unit freq.	erg.cm**(-2).s**(-1).Hz**(-1), Jy, W.m**(-2).Hz**(-1)
phot.flux.density;em.energy	Flux density per energy interval	keV.cm**(-2).s**(-1).keV**(-1)
phot.flux.density;em.energy; meta.number	Photons per unit area, time, energy	photon.cm**(-2).s**(-1).keV**(-1)
phot.flux.density;em.wl	Flux density per log wave interval ( $\nu F(\nu)$ )	erg.cm**(-2).s**(-1), Jy.Hz
phot.flux.density.sb;em.wl	Surface brightness per unit wavelength	erg.cm**(-2).s**(-1).angstrom**(-1).arcsec**(-2)
phot.flux.density.sb;em.freq	Surface brightness per unit frequency	Jy.sr**(-1)
phot.count	Counts in spectral channel	count, adu
arith.rate;phot.count	Count rate in spectral channel	count/s
phot.flux.density;arith.ratio	Flux ratio of two spectra	(empty string)
phys.luminosity;em.wl	Luminosity per unit wave	erg.s**(-1).angstrom**(-1), W/m
phys.luminosity;em.freq	Luminosity per unit freq	erg.s**(-1).Hz**(-1), W/Hz
phys.luminosity;em.energy	Luminosity per unit energy	erg.s**(-1).keV**(-1)
phys.luminosity;em.energy	Luminosity per log frequency	erg.s**(-1), W
phys.energy.density	Radiation energy density per unit volume, per unit wave etc.	erg.cm**(-3), J.m**(-3)
phot.fluence;em.wl	Photon number flux per unit wave.	photon.cm**(-2).s**(-1).angstrom**(-1)
phot.flux.density;em.wl; phys.polarization	Polarized flux per unit wavelength	erg.cm**(-2).s**(-1).angstrom**(-1)
phys.polarization	Polarized fraction vs spectral coord	(empty string)
phys.luminosity; phys.angArea;em.wl	Flux per unit solid angle (at source)	erg.cm**(-2).s**(-1).sr**(-1).angstrom**(-1)
phot.antennaTemp	Antenna temperature	K
phot.flux.density; phys.temperature	Brightness temperature	K
phot.mag	Magnitude in defined band	mag
phot.mag	AB (spectrophotometric) magnitude	mag
phot.flux.density;instr.beam	Flux per resolution element (e.g. Jy/beam)	Jy/beam
phot.mag.sb	Surface brightness in magnitudes	mag.arcsec**(-2)
phys.transmission	Filter transmission, 0.0 to 1.0	(empty string)
phys.area;phys.transmission	Effective area	cm**2
phot.flux.density;em.wl; spect.continuum	Continuum only	erg.cm**(-2).s**(-1).angstrom**(-1).arcsec**(-2)

### B.2 Spectral Axis UCDs

Spectral Axis UCD Options		
UCD1+	Meaning	Unit (VOUnit style)
PREFERRED		
em.wl	Wavelength	angstrom, m
em.freq	Frequency of photon	Hz
em.energy	Photon energy	erg, eV, J
instr.pixel;em.wl	Instrumental spectral bin	chan
ALTERNATIVE		
em.wavenumber	Wavenumber	m**(-1)
em.wl;obs.atmos	Air wavelength	angstrom, m
em.wl	Log wavelength	(empty string)
em.freq	Log frequency of photon	(empty string)
em.energy	Log photon energy	(empty string)
spect.dopplerVeloc	Apparent radial velocity	m/s
spect.dopplerVeloc.radio	Radio velocity	m/s
spect.dopplerVeloc.opt	Optical velocity	m/s
spect.dopplerVeloc	Velocity (c=1)	(empty string)

## C Sample Serializations

### C.1 VOTable Serialization

These sample serializations of Spectrum Datasets use a single TABLE block. Sequence level elements are serialized using PARAM elements while the per-point elements use VOTable FIELDS. We use the VOTable GROUPS construct to aid readability. It is not a requirement for users to make use of this construct for all elements of the model. They are, however, required in order to differentiate multiple instances of the same type of object. For example, GenericCoord, so that the attributes are associated correctly.

Names of fields and parameters are left to the data provider, as are any descriptions. The utype and ucd attributes of these VOTable elements are used to denote the datamodel UType and UCD tags. Note: this may cause some confusion for elements which have 'name', 'ucd', and 'utype' attributes which each have their own UType defined, (e.g. TimeFrame.name, TimeFrame.unit, TimeFrame.ucd).

We have made up arbitrary NAME attributes for the VOTable elements and these are not to be considered standard. The name fields are free to be whatever the data provider wants, allowing for compatibility with local archive nomenclature. It is the UType which is standardized.

#### C.1.1 Basic Spectrum Instance

This example shows a Basic Spectrum Dataset, it contains only the required fields.

```
<?xml version="1.0" encoding="UTF-8"?>
<VOTABLE version="1.2" xmlns="http://www.ivoa.net/xml/VOTable/v1.2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ivoa.net/xml/VOTable/v1.2 http://www.ivoa.net/xml/VOTable/VOTable-1.2.xsd"
>
<RESOURCE utype="spec:Spectrum">
<TABLE name="Sample Spectrum instance" utype="spec:Spectrum">

<FIELD ID="DataFluxValue" name="DataFluxValue"
  datatype="double" ucd="phot.flux.density;em.freq" utype="spec:Data.FluxAxis.Value" unit="Jy"/>
<FIELD ID="DataSpectralValue" name="DataSpectralValue"
  datatype="double" ucd="em.freq" utype="spec:Data.SpectralAxis.Value" unit="Hz"/>

<GROUP name="Dataset">
  <GROUP name="DataModel">
    <PARAM name="ModelName"
      datatype="char" utype="spec:Dataset.DataModel.Name" value="Spectrum-2.0" arraysize="*">
      <DESCRIPTION>Datamodel name and version</DESCRIPTION>
    </PARAM>
  </GROUP>
</GROUP>

<GROUP name="Curation">
  <PARAM name="Publisher"
    datatype="char" ucd="meta.curation" utype="spec:Curation.Publisher" value="Database" arraysize="*">
    <DESCRIPTION>Dataset publisher</DESCRIPTION>
  </PARAM>
</GROUP>

<GROUP name="DataID">
  <PARAM name="Title"
    datatype="char" ucd="meta.title;meta.dataset" utype="spec:DataID.Title"
    value="Sample Spectrum Data" arraysize="*">
    <DESCRIPTION>Dataset Title</DESCRIPTION>
  </PARAM>
```

```

</GROUP>

<GROUP name="Target">
  <PARAM name="TargetName"
    datatype="char" ucd="meta.id;src" utype="spec:Target.Name" value="3C 273" arraysize="*">
    <DESCRIPTION>Target name</DESCRIPTION>
  </PARAM>
</GROUP>

<GROUP name="Characterization">
  <GROUP name="Char.FluxAxis" utype="spec:Char.FluxAxis">
    <PARAM name="FluxAxisUcd"
      datatype="char" utype="spec:Char.FluxAxis.ucd" value="phot.flux.density;em.freq" arraysize="*">
      <DESCRIPTION>UCD for flux axis</DESCRIPTION>
    </PARAM>
    <PARAM name="FluxAxisUnit"
      datatype="char" utype="spec:Char.FluxAxis.unit" value="Jy" arraysize="*">
      <DESCRIPTION>Unit for flux axis</DESCRIPTION>
    </PARAM>
  </GROUP>

  <GROUP name="Char.SpatialAxis">
    <PARAM name="SpatialAxisCoverageLocation"
      datatype="double" ucd="pos.eq" utype="spec:Char.SpatialAxis.Coverage.Location.Value"
      value="187.277915 2.052388" arraysize="2">
      <DESCRIPTION>Position</DESCRIPTION>
    </PARAM>
    <PARAM name="SpatialAxisCoverageBoundsExtent"
      datatype="double" ucd="instr.fov" utype="spec:Char.SpatialAxis.Coverage.Bounds.Extent" value="0.00555">
      <DESCRIPTION>Aperture angular size</DESCRIPTION>
    </PARAM>
  </GROUP>

  <GROUP name="Char.SpectralAxis">
    <PARAM name="SpectralAxisUcd"
      datatype="char" utype="spec:Char.SpectralAxis.ucd" value="em.freq" arraysize="*">
      <DESCRIPTION>UCD for spectral axis</DESCRIPTION>
    </PARAM>
    <PARAM name="SpectralAxisUnit"
      datatype="char" utype="spec:Char.SpectralAxis.unit" value="Hz" arraysize="*">
      <DESCRIPTION>Unit for spectral axis</DESCRIPTION>
    </PARAM>
    <PARAM name="SpectralAxisCoverageLocation"
      datatype="double" ucd="instr.bandpass" utype="spec:Char.SpectralAxis.Coverage.Location.Value" value="">
      <DESCRIPTION>Spectral coord value.</DESCRIPTION>
    </PARAM>
    <PARAM name="SpectralAxisCoverageBoundsExtent"
      datatype="double" ucd="instr.bandwidth" utype="spec:Char.SpectralAxis.Coverage.Bounds.Extent" value="">
      <DESCRIPTION>Width of spectral coordinate</DESCRIPTION>
    </PARAM>
    <PARAM name="SpectralAxisCoverageBoundsStart"
      datatype="double" ucd="em.freq;stat.min" utype="spec:Char.SpectralAxis.Coverage.Bounds.Start" value="">
      <DESCRIPTION>Start in spectral coord</DESCRIPTION>
    </PARAM>
    <PARAM name="SpectralAxisCoverageBoundsStop"
      datatype="double" ucd="em.freq;stat.max" utype="spec:Char.SpectralAxis.Coverage.Bounds.Stop" value="">
      <DESCRIPTION>Stop in spectral coord</DESCRIPTION>
    </PARAM>
  </GROUP>

```

```

<GROUP name="Char.TimeAxis">
  <PARAM name="TimeAxisCoverageLocation"
    datatype="double" ucd="time.epoch" utype="spec:Char.TimeAxis.Coverage.Location.Value"
    value="5199.54882939815">
    <DESCRIPTION>Exposure midpoint</DESCRIPTION>
  </PARAM>
  <PARAM name="TimeAxisCoverageBoundsExtent"
    datatype="double" ucd="time.duration" utype="spec:Char.TimeAxis.Coverage.Bounds.Extent" unit="s"
    value="2701.0">
    <DESCRIPTION>Total elapsed time</DESCRIPTION>
  </PARAM>
</GROUP>

</GROUP>

<GROUP name="Data">
  <FIELDref ref="DataFluxValue"/>
  <FIELDref ref="DataSpectralValue"/>
</GROUP>

<DATA>
<TABLEDATA>
  <TR><TD>8.32826233e+14</TD> <TD>3.72981229e-30</TD></TR>
  <TR><TD>8.32190479e+14</TD> <TD>2.58023996e-30</TD></TR>
  <TR><TD>8.31555695e+14</TD> <TD>3.49485448e-30</TD></TR>
  <TR><TD>8.30921878e+14</TD> <TD>3.53532448e-30</TD></TR>
  <TR><TD>8.30289027e+14</TD> <TD>3.53108340e-30</TD></TR>
</TABLEDATA>
</DATA>
</TABLE>
</RESOURCE>
</VOTABLE>

```

## C.1.2 Basic PhotometryPoint Instance

This example shows a Basic PhotometryPoint Dataset, it contains only the required fields.

In this case, since there is only 1 data point to serialize, we use the VOTable PARAM element for all of the Model fields. Note: that it would be equally valid to define a set of VOTable FIELD elements and fill 1 row of VOTable TABLEDATA.

```
<?xml version="1.0" encoding="UTF-8"?>
<VOTABLE version="1.2" xmlns="http://www.ivoa.net/xml/VOTable/v1.2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ivoa.net/xml/VOTable/v1.2 http://www.ivoa.net/xml/VOTable/VOTable-1.2.xsd"
>
<RESOURCE utype="phot:PhotometryPoint">
<TABLE name="Sample Photometry Point instance" utype="phot:PhotometryPoint">

<GROUP name="Dataset">
  <GROUP name="DataModel">
    <PARAM name="ModelName"
      datatype="char" utype="phot:Dataset.DataModel.Name" value="PhotometryPoint-1.0" arraysize="*">
      <DESCRIPTION>Datamodel name and version</DESCRIPTION>
    </PARAM>
  </GROUP>
</GROUP>

<GROUP name="CoordSys">
  <PARAM name="ZeroPoint.Flux.Value"
    datatype="double" ucd="phot.flux.density;em.freq"
    utype="phot:CoordSys.FluxFrame.PhotCal.ZeroPoint.Flux.value" value="2.72E-11">
    <DESCRIPTION>Flux value at Zero point associated to this filter.</DESCRIPTION>
  </PARAM>
  <PARAM name="ZeroPoint.Flux.Unit"
    datatype="char" ucd="meta.ucd" utype="phot:CoordSys.FluxFrame.Photcal.ZeroPoint.Flux.unit"
    value="Jy" arraysize="*">
    <DESCRIPTION>unit for Zero Point flux value.</DESCRIPTION>
  </PARAM>
</GROUP>

<GROUP name="Curation">
  <PARAM name="Publisher"
    datatype="char" ucd="meta.curation" utype="phot:Curation.Publisher" value="Database" arraysize="*">
    <DESCRIPTION>Dataset publisher</DESCRIPTION>
  </PARAM>
</GROUP>

<GROUP name="DataID">
  <PARAM name="Title"
    datatype="char" ucd="meta.title;meta.dataset" utype="phot:DataID.Title"
    value="Sample Spectrum Data" arraysize="*">
    <DESCRIPTION>Dataset Title</DESCRIPTION>
  </PARAM>
</GROUP>

<GROUP name="Target">
  <PARAM name="TargetName"
    datatype="char" ucd="meta.id;src" utype="phot:Target.Name" value="3C 273" arraysize="*">
    <DESCRIPTION>Target name</DESCRIPTION>
  </PARAM>
</GROUP>
```

```

<GROUP name="Characterization">
  <GROUP name="Char.FluxAxis" utype="phot:Char.FluxAxis">
    <PARAM name="FluxAxisUcd"
      datatype="char" utype="phot:Char.FluxAxis.ucd" value="phot.flux.density;em.freq" arraysize="*">
      <DESCRIPTION>UCD for flux axis</DESCRIPTION>
    </PARAM>
    <PARAM name="FluxAxisUnit"
      datatype="char" utype="phot:Char.FluxAxis.unit" value="Jy" arraysize="*">
      <DESCRIPTION>Unit for flux axis</DESCRIPTION>
    </PARAM>
  </GROUP>

  <GROUP name="Char.SpatialAxis">
    <PARAM name="SpatialAxisCoverageLocation"
      datatype="double" ucd="pos.eq" utype="phot:Char.SpatialAxis.Coverage.Location.Value"
      value="187.277915 2.052388" arraysize="2">
      <DESCRIPTION>Position</DESCRIPTION>
    </PARAM>
    <PARAM name="SpatialAxisCoverageBoundsExtent"
      datatype="double" ucd="instr.fov" utype="phot:Char.SpatialAxis.Coverage.Bounds.Extent"
      value="0.00555">
      <DESCRIPTION>Aperture angular size</DESCRIPTION>
    </PARAM>
  </GROUP>

  <GROUP name="Char.SpectralAxis">
    <PARAM name="SpectralAxisUcd"
      datatype="char" utype="phot:Char.SpectralAxis.ucd" value="em.freq" arraysize="*">
      <DESCRIPTION>UCD for spectral axis</DESCRIPTION>
    </PARAM>
    <PARAM name="SpectralAxisUnit"
      datatype="char" utype="phot:Char.SpectralAxis.unit" value="Hz" arraysize="*">
      <DESCRIPTION>Unit for spectral axis</DESCRIPTION>
    </PARAM>
    <PARAM name="SpectralAxisCoverageLocation"
      datatype="double" ucd="instr.bandpass" utype="phot:Char.SpectralAxis.Coverage.Location.Value"
      value="">
      <DESCRIPTION>Spectral coord value.</DESCRIPTION>
    </PARAM>
    <PARAM name="SpectralAxisCoverageBoundsExtent"
      datatype="double" ucd="instr.bandwidth" utype="phot:Char.SpectralAxis.Coverage.Bounds.Extent"
      value="">
      <DESCRIPTION>Width of spectral coordinate</DESCRIPTION>
    </PARAM>
    <PARAM name="SpectralAxisCoverageBoundsStart"
      datatype="double" ucd="em.freq;stat.min" utype="phot:Char.SpectralAxis.Coverage.Bounds.Start"
      value="">
      <DESCRIPTION>Start in spectral coord</DESCRIPTION>
    </PARAM>
    <PARAM name="SpectralAxisCoverageBoundsStop"
      datatype="double" ucd="em.freq;stat.max" utype="phot:Char.SpectralAxis.Coverage.Bounds.Stop"
      value="">
      <DESCRIPTION>Stop in spectral coord</DESCRIPTION>
    </PARAM>
  </GROUP>

  <GROUP name="Char.TimeAxis">
    <PARAM name="TimeAxisCoverageLocation"

```

```

    datatype="double" ucd="time.epoch" utype="phot:Char.TimeAxis.Coverage.Location.Value"
    value="5199.54882939815">
    <DESCRIPTION>Exposure midpoint</DESCRIPTION>
  </PARAM>
  <PARAM name="TimeAxisCoverageBoundsExtent"
    datatype="double" ucd="time.duration" utype="phot:Char.TimeAxis.Coverage.Bounds.Extent" unit="s"
    value="2701.0">
    <DESCRIPTION>Total elapsed time</DESCRIPTION>
  </PARAM>
</GROUP>

</GROUP>

<GROUP name="Data">
  <PARAM name="DataFluxValue"
    datatype="double" ucd="phot.flux.density;em.freq" utype="phot:Data.FluxAxis.Value" unit="Jy"
    value="2.72E-11">
    <DESCRIPTION>Photometry point flux value.</DESCRIPTION>
  </PARAM>
  <PARAM name="DataSpectralValue"
    datatype="double" ucd="em.freq" utype="phot:Data.SpectralAxis.Value" unit="Hz"
    value="6.17E+23">
    <DESCRIPTION>Photometry point spectral value.</DESCRIPTION>
  </PARAM>
</GROUP>

</TABLE>
</RESOURCE>
</VOTABLE>

```



## C.2 FITS Serialization

Here we define the requirements for serializing an instance of the Spectral model as a single extension of a FITS binary table.

### Metadata Elements

Sequence level elements are serialized as FITS keywords. Since there is no IVOA convention available for mapping elements (by UType) to individual keywords, users are REQUIRED to use the names given below in the FITS Keyword mapping table. In addition, there is no mechanism for associating a UCD to each key. Applications are REQUIRED to assign appropriate UCD values to these elements as defined in the UType list, when interpreting these files.

**Keyword arrays:** Some model elements are expressed as keyword arrays (e.g. DataID.Collection). These map to FITS array keywords (COLLECTn) where 'n' is the index of the array. If the field contains only one value, the element may be expressed as a scalar keyword, omitting the counter portion of the name (COLLECT).

**Expressing spatial coordinates in FITS:** FITS employs the WCS conventions to express celestial coordinates for image axes or table columns. This convention does not apply to a single celestial position. For this, we propose a local convention leveraging the existing WCS conventions:

- The keyword names for the coordinates are those used in the first four characters of the CTYPE values for the WCS paper: e.g. RA, DEC, GLON, GLAT.
- The coordinate system is identified by the keyword SKY\_UCD with values such as pos.eq, etc.
- The RADECSYS and EQUINOX keywords should be used when appropriate.
- Values are always in degrees.
- The VOCSID optional keyword is provided to allow VO coordinate system names to be propagated.

**The SPECSYS keyword:** We note the allowed values of the SPECSYS keyword from Greisen et al and the corresponding values for the CoordFrame reference position noted in Table 1:

FITS	Reference Position	Meaning
TOPOCENT	TOPOCENTER	Topocenter
GEOCENTR	GEOCENTER	Geocenter
BARYCENT	BARYCENTER	Solar System Barycenter
HELIOCEN	HELIOCENTER	Heliocenter
LSRK	LSRK	Kinematic local standard of rest
LSRD	LSRD	Dynamic local standard of rest
GALACTOC	GALACTIC_CENTER	Galactic center
LOCALGRP	LOCAL_GROUP_CENTER	Local group barycenter
CMBDIPOL	-	Frame of the Cosmic Microwave Background dipole
SOURCE	-	Source rest frame

### Data Elements

Model elements under the Data node are serialized as table columns. We extend the FITS standard column definition keyword set to allow the assignment of UType and UCD to these elements. As such, users are not required to use the specified names. However, users are strongly encouraged to do so in order to provide consistency, and reduce confusion in employing the 'Greenbank' convention.

**Column Definition Keys:** The following keywords must be defined for each column mapping to a Spectral model element. Here, 'n' refers to the column number.

**TTYPEn** - field name.

**TFORMn** - datatype of field.

**TUNITn** - units for field. (optional: define as needed)

**TUTYPEn** - UType string mapping the column to a model field.

**TUCDn** - UCD associated with the field.

**Greenbank Convention:** In some cases, the column data arrays may have the same value for each data point. In this case, users may employ the 'Greenbank' convention in which the column is omitted, and replaced by a keyword whose name is the same as the column.

Employing this convention removes the UType and UCD mapping associated with the field. It is, therefore, REQUIRED that the keyword replacement use the name provided in the keyword mapping table, even if different from the original column name.

To support this convention, implementors should check both keywords and columns for appropriate tokens.

**Data-Char Overlap:** Section 2.12.1 describes several model fields from Char which serve as a default for the corresponding field under Data. We assign the same name to both fields, since, generally, only one need be present. FITS standards strongly recommend uniqueness of names between keys and columns. Therefore, if both are present, users should select a unique name for the table column.

**WCS table keywords:** The spectral coordinate may be identified by optional  $1S_{n,1}$  and  $1CTYP_n$  keywords as per WCS Paper 3. Table 9 of that paper implies that each data column which is a function of the spectral coord needs a pair of such keywords. Applications which implement the spectrum data model may ignore the WCS keys and interpret the file by recognizing the  $TUTYP_n$  value identifying which column is the spectral coordinate and that FLUX, etc. are functions of it, but the WCS keys give a general FITS application a chance at making sense of the file. For example,  $TTYPE_5='ERR\_LO'$  and  $TUTYP_5='Data.FluxAxis.Accuracy.StatErrLow'$ ; the WCS keyword  $1CTYPE_5='WAVE-TAB'$  indicates that the data in column 5 is a function of wavelength, and that the wavelengths are in a lookup table. The WCS keyword  $1S_{5,1}='WAVE'$  indicates that the lookup table for the x-axis of column 5 is in the column with  $TTYPEn='WAVE'$ .

### Recommendations:

- The order of keywords and columns is not significant. However, it is strongly recommended that RA and Dec be in adjacent columns or keywords.
- The  $TTYPEn$  keyword values for the x and y columns are unrestricted, but it is strongly recommended that (for consistency of style with WCS Paper 3) the values for the x axis have for their first 4 characters 'WAVE', 'FREQ' and 'ENER' for the case of wavelength, frequency and energy respectively. We also recommend the value 'FLUX' for the y axis, where appropriate.
- For metadata fields, we use SPEC\_ keywords to denote the spectral axis generically, but in the table columns, we use the terms WAVE\_, FREQ\_, and ENER\_ as appropriate. Thus if the Data.SpectralAxis.Value field is WAVE, the Data.SpectralAxis.Accuracy.BinLow field should be WAVE\_LO; if Value is FREQ, BinLow should be FREQ\_LO. We believe the small extra parsing overhead is worth it for the readability and interoperability (since these names have been used in existing FITS files) of the crucial main data table.

- To express the RedshiftFrame, we recommend using a FITS WCS system with suffix 'Z' applied to the spectral coordinate axis, when appropriate. CoordSys.RedshiftFrame.DopplerDefinition is represented by the first four characters of TCTYPnZ and should have the values VRAD, VOPT, ZOPT or VELO, as per the convention for spectral CTYPE keywords in Paper III of the FITS WCS system. CoordSys.RedshiftFrame.RefPos is represented by SPECSYSZ and should have values as listed in Paper III of the FITS WCS system.
- Photometry metadata may be referred to externally via the PHID keyword, which can contain a URI to a photometry calibration resource, or explicitly provided in the header using the PH series of keywords tabulated below. The spectral coordinate (e.g. effective wavelength) of the photometric filter is stored in the SPEC\_VAL, SPEC\_UNI, and SPEC\_UCD keywords.
- Additional keywords and columns which are not part of this model (including other conventions such as e.g. TDMINn) are allowed to be present, but are not guaranteed to be propagated by implementations.

### Limitations:

- The mechanism described here does not allow for the definition of more than one of any complex object. This is noted especially for the FluxFrame object, for which we may have more than one defined for a given photometric sequence.
- A Photometry Filter may not be serialized in the same file as the photometric sequence. There is no mechanism for serializing a reference to another extension within the same file. It may be referred to via the keyword holding the reference URI.

### Mapping

This table gives the mapping of Spectral data model elements to FITS columns and keywords. For each model UType, the corresponding FITS keyword is defined. The recommended value to use for the standardized keywords (TTYPEEn, TUNITn, etc) are also given.

FITS keyword mapping for Spectral Data Model	
Spectral Model UType	FITS Keyword/Column definition
Char.FluxAxis.Name	FLUX_NAM
Char.FluxAxis.ucd	FLUX_UCD
Char.FluxAxis.unit	FLUX_UNI
Char.FluxAxis.CalibStatus	FLUX_CAL
Char.FluxAxis.Accuracy.StatError	STAT_ERR
Char.FluxAxis.Accuracy.SysError	SYS_ERR
Char.FluxAxis.QualityDefs.QualityCode.value	QUALn
Char.FluxAxis.QualityDefs.QualityCode.description	QUALn value
Char.SpatialAxis.Name	SKY_NAM
Char.SpatialAxis.ucd	SKY_UCD
Char.SpatialAxis.unit	SKY_UNI
Char.SpatialAxis.CalibStatus	SKY_CAL
Char.SpatialAxis.Accuracy.StatError	SKY_ERR
Char.SpatialAxis.Accuracy.SysError	SKY_SYE
Char.SpatialAxis.Coverage.Location.Value	RA,DEC;etc
Char.SpatialAxis.Coverage.Bounds.Extent	APERTURE
Char.SpatialAxis.Coverage.Support.Area	REGION
Char.SpatialAxis.Coverage.Support.Extent	AREA
Char.SpatialAxis.QualityDefs.QualityCode.value	
Char.SpatialAxis.QualityDefs.QualityCode.description	
Char.SpatialAxis.Resolution.refVal	SKY_RES
Char.SpatialAxis.SamplingPrecision.SampleExtent	TCDLTn
Char.SpatialAxis.SamplingPrecision.SamplingPrecisionRefVal.FillFactor	SKY_FILL
Char.SpectralAxis.Name	SPEC_NAM
Char.SpectralAxis.ucd	SPEC_UCD
Char.SpectralAxis.unit	SPEC_UNI
Char.SpectralAxis.CalibStatus	SPEC_CAL
Char.SpectralAxis.Accuracy.BinSize	SPEC_BIN
Char.SpectralAxis.Accuracy.StatError	SPEC_ERR
Char.SpectralAxis.Accuracy.SysError	SPEC_SYE
Char.SpectralAxis.Coverage.Location.Value	SPEC_VAL
Char.SpectralAxis.Coverage.Bounds.Extent	SPEC_BW
Char.SpectralAxis.Coverage.Bounds.Start	TDMINn
Char.SpectralAxis.Coverage.Bounds.Stop	TDMAXn
Char.SpectralAxis.Coverage.Support.Extent	SPECWID
Char.SpectralAxis.QualityDefs.QualityCode.value	

FITS keyword mapping (Continued)	
Spectral Model UType	FITS Keyword/Column definition
Char.SpectralAxis.QualityDefs.QualityCode.description	
Char.SpectralAxis.Resolution.refVal	SPEC_RES
Char.SpectralAxis.ResolPower.refVal	SPEC_RP
Char.SpectralAxis.SamplingPrecision.SampleExtent	SPEC_BIN
Char.SpectralAxis.SamplingPrecision.SamplingPrecisionRefVal.FillFactor	SPEC_FIL
Char.TimeAxis.Name	TIME_NAM
Char.TimeAxis.ucd	TIME_UCD
Char.TimeAxis.unit	TIMEUNIT
Char.TimeAxis.CalibStatus	TIME_CAL
Char.TimeAxis.Accuracy.BinSize	TIMEDEL
Char.TimeAxis.Accuracy.StatError	TIME_ERR
Char.TimeAxis.Accuracy.SysError	TIME_SYE
Char.TimeAxis.Coverage.Location.Value	T MID
Char.TimeAxis.Coverage.Bounds.Extent	TELAPSE
Char.TimeAxis.Coverage.Bounds.Start	TSTART
Char.TimeAxis.Coverage.Bounds.Stop	TSTOP
Char.TimeAxis.Coverage.Support.Extent	EXPOSURE
Char.TimeAxis.QualityDefs.QualityCode.value	
Char.TimeAxis.QualityDefs.QualityCode.description	
Char.TimeAxis.Resolution.refVal	TIME_RES
Char.TimeAxis.SamplingPrecision.SampleExtent	TIMEDEL
Char.TimeAxis.SamplingPrecision.SamplingPrecisionRefVal.FillFactor	DTCOR
CoordSys.ID	VOCSID
CoordSys.CoordFrame.ID	
CoordSys.CoordFrame.Name	
CoordSys.CoordFrame.UCD	
CoordSys.CoordFrame.RefPos	
CoordSys.FluxFrame.ID	
CoordSys.FluxFrame.Name	PHBAND
CoordSys.FluxFrame.UCD	PHUCD
CoordSys.FluxFrame.refID	PHID
CoordSys.FluxFrame.PhotCal.*	(see below)
CoordSys.RedshiftFrame.ID	
CoordSys.RedshiftFrame.Name	ZNAME
CoordSys.RedshiftFrame.UCD	
CoordSys.RedshiftFrame.RefPos	SPECSYSZ
CoordSys.RedshiftFrame.DopplerDefinition	TCTYPnZ
CoordSys.SpaceFrame.ID	
CoordSys.SpaceFrame.Name	RADECSYS
CoordSys.SpaceFrame.UCD	SKY_UCD
CoordSys.SpaceFrame.RefPos	SKY_REF
CoordSys.SpaceFrame.Equinox	EQUINOX
CoordSys.SpectralFrame.ID	
CoordSys.SpectralFrame.Name	SPECNAME
CoordSys.SpectralFrame.UCD	TUCDn
CoordSys.SpectralFrame.RefPos	SPECSYS
CoordSys.SpectralFrame.Redshift	REST_Z
CoordSys.TimeFrame.ID	
CoordSys.TimeFrame.Name	TIMESYS
CoordSys.TimeFrame.UCD	
CoordSys.TimeFrame.RefPos	
CoordSys.TimeFrame.Zero	MJDREF
Curation.Date	VODATE
Curation.Publisher	VOPUB
Curation.PublisherID	VOPUBID
Curation.PublisherDID	DS_IDPUB
Curation.Reference	VOREF
Curation.Rights	VORIGHTS
Curation.Version	VOVER
Curation.Contact.Name	CONTACT
Curation.Contact.Email	EMAIL
DataID.Title	TITLE
DataID.Creator	AUTHOR
DataID.Collection	COLLECTn
DataID.DatasetID	DS_IDENT
DataID.CreatorDID	CR_IDENT
DataID.Date	DATE
DataID.Version	VERSION
DataID.Instrument	INSTRUME
DataID.Bandpass	SPECBAND
DataID.CreationType	CRETYP
DataID.Logo	VOLOGO
DataID.Contributor	CONTRIBn
DataID.DataSource	DSSOURCE
Dataset.DataModel.Name	VOCLASS
Dataset.DataModel.Prefix	VOPRFX
Dataset.Type	VOSEGT
Dataset.Length	DATALEN
Dataset.TimeSI	TIMESDIM
Dataset.SpectralSI	SPECSDIM

FITS keyword mapping (Continued)

Spectral Model UType	FITS Keyword/Column definition
Dataset.FluxSI	FLUXSDIM
Derived.SNR	DER_SNR
Derived.Redshift.Value	DER_Z
Derived.Redshift.StatError	DER_ZERR
Derived.Redshift.Confidence	DER_ZCNF
Derived.VarAmpl	DER_VAR
Target.Name	OBJECT
Target.Description	OBJDESC
Target.Class	SRCLASS
Target.SpectralClass	SPECTYPE
Target.Redshift	REDSHIFT
Target.Pos	RA/DEC_TARG
Target.VarAmpl	TARGVAR
PhotCal.identifier	PHUID
PhotCal.ZeroPoint.referenceMagnitude.value	PHMAGZ
PhotCal.ZeroPoint.referenceMagnitude.unit	PHMUNI
PhotCal.ZeroPoint.referenceMagnitude.ucd	PHMUCD
PhotCal.ZeroPoint.Flux.value	PHZERO
PhotCal.ZeroPoint.Flux.unit	PHFUNI
PhotCal.ZeroPoint.Flux.ucd	PHFUCD
PhotCal.ZeroPoint.type	PHFTYPE
PhotCal.AsinhZeroPoint.softeningParameter	PHFLUPB
PhotCal.MagnitudeSystem.ReferenceSpectrum.uri	PHREFS
PhotCal.MagnitudeSystem.type	PHMSTY
PhotCal.PhotometryFilter.name	PHNAME
PhotCal.PhotometryFilter.bandName	PHGID
PhotCal.PhotometryFilter.description	PHDESC
PhotCal.PhotometryFilter.fpsIdentifier	FPSID
PhotCal.PhotometryFilter.transmissionCurve	PHTRANS
Data.FluxAxis.Value	TTYPEn='FLUX'
Data.FluxAxis.ucd	TUCDn
Data.FluxAxis.unit	TUNITn
Data.FluxAxis.Accuracy.QualityStatus	TTYPEn='QUALITY'
Data.FluxAxis.Accuracy.StatError	TTYPEn='ERR'
Data.FluxAxis.Accuracy.StatErrLow	TTYPEn='ERR_LO'
Data.FluxAxis.Accuracy.StatErrHigh	TTYPEn='ERR_HI'
Data.FluxAxis.Accuracy.SysError	TTYPEn='SYS_ERR'
Data.FluxAxis.Corrections.ApFrac.Value	
Data.FluxAxis.Corrections.ApFrac.Applied	
Data.SpectralAxis.Value	TTYPEn='WAVE','ENER' or 'FREQ'
Data.SpectralAxis.ucd	TUCDn
Data.SpectralAxis.unit	TUNITn
Data.SpectralAxis.Accuracy.BinSize	TTYPEn='[WAVE,ENER,FREQ]_BIN'
Data.SpectralAxis.Accuracy.BinLow	TTYPEn='[WAVE,ENER,FREQ]_LO'
Data.SpectralAxis.Accuracy.BinHigh	TTYPEn='[WAVE,ENER,FREQ]_HI'
Data.SpectralAxis.Accuracy.StatError	TTYPEn='[WAVE,ENER,FREQ]_ERR'
Data.SpectralAxis.Accuracy.StatErrLow	TTYPEn='[WAVE,ENER,FREQ]_ELO'
Data.SpectralAxis.Accuracy.StatErrHigh	TTYPEn='[WAVE,ENER,FREQ]_EHI'
Data.SpectralAxis.Accuracy.SysError	TTYPEn='[WAVE,ENER,FREQ]_SYE'
Data.SpectralAxis.Resolution.refVal	TTYPEn='[WAVE,ENER,FREQ]_RES'
Data.TimeAxis.Value	TTYPEn='TIME'
Data.TimeAxis.ucd	TUCDn
Data.TimeAxis.unit	TUNITn
Data.TimeAxis.Accuracy.BinSize	TTYPEn='TIMEDEL'
Data.TimeAxis.Accuracy.BinLow	TTYPEn='TIME_LO'
Data.TimeAxis.Accuracy.BinHigh	TTYPEn='TIME_HI'
Data.TimeAxis.Accuracy.StatError	TTYPEn='TIME_ERR'
Data.TimeAxis.Accuracy.StatErrLow	TTYPEn='TIME_ELO'
Data.TimeAxis.Accuracy.StatErrHigh	TTYPEn='TIME_EHI'
Data.TimeAxis.Accuracy.SysError	TTYPEn='TIME_SYE'
Data.TimeAxis.Resolution.refVal	TTYPEn='TIME_RES'
Data.BackgroundModel.Value	TTYPEn='BGFLUX'
Data.BackgroundModel.ucd	TUCDn
Data.BackgroundModel.unit	TUNITn
Data.BackgroundModel.Accuracy.QualityStatus	TTYPEn='BGQUAL'
Data.BackgroundModel.Accuracy.StatError	TTYPEn='BG_ERR'
Data.BackgroundModel.Accuracy.StatErrLow	TTYPEn='BG_ELO'
Data.BackgroundModel.Accuracy.StatErrHigh	TTYPEn='BG_EHI'
Data.BackgroundModel.Accuracy.SysError	TTYPEn='BG_SYE'
Data.*Axis.Corrections.GenericCorr.Name	
Data.*Axis.Corrections.GenericCorr.Value	
Data.*Axis.Corrections.GenericCorr.Applied	

## C.2.1 Basic Spectrum Instance

This is an example serialization of a Basic Spectrum Dataset, it contains only required elements.

```

XTENSION= 'BINTABLE'           / binary table extension
BITPIX   =                    8 / 8-bit bytes
NAXIS    =                    2 / 2-dimensional binary table
NAXIS1   =                   16 / width of table in bytes
NAXIS2   =                  1501 / number of rows in table
PCOUNT   =                    0 / size of special data area
GCOUNT   =                    1 / one data group (required keyword)
TFIELDS  =                    2 / number of fields in each row
EXTNAME  = 'SPECTRUM'         / name of this binary table extension
HDUNAME  = 'SPECTRUM'         / ASCDM block name
TTYPE1   = 'FREQ'             / Spectral column name
TFORM1   = '1D'               / format of field
TUNIT1   = 'Hz'               /
TTYPE2   = 'FLUX'             / Flux column name
TFORM2   = '1D'               / format of field
TUNIT2   = 'W m**(-2) Hz**(-1)'
COMMENT  ----- Dataset -----
VOCLASS  = 'Spectrum-2.0'     / Datamodel name and version
COMMENT  ----- Curation -----
VOPUB    = 'Database'         / Dataset publisher
COMMENT  ----- DataID -----
TITLE    = 'Sample Spectrum Data' / Dataset Title
COMMENT  ----- Target -----
OBJECT   = '3C 273'           / Target name
COMMENT  ----- Char.FluxAxis -----
FLUX_UCD= 'phot.flux.density;em.freq' / UCD for flux axis
FLUX_UNI= 'W m**(-2) Hz**(-1)' / Unit for flux axis
COMMENT  ----- Char.SpatialAxis -----
RA       = 1.8727791500000E+02 / [deg] Pointing position
DEC      = 2.0523880000000E+00 / [deg] Pointing position
APERTURE= 5.5500000000000E-04 / [deg] Aperture angular size
COMMENT  ----- Char.SpectralAxis -----
SPEC_UCD= 'em.freq'           / UCD for spectral axis
SPEC_UNI= 'Hz'                 / Unit for spectral axis
SPEC_VAL= 5.7786304400000E+14 / [Hz] Characteristic spectral coord.
SPEC_BW  = 4.2054702500000E+14 / [Hz] Width of spectral coordinate
TDMIN2   = 4.3600285500000E+14 / [Hz] Start in spectral coord
TDMAX2   = 8.5654988300000E+14 / [Hz] Stop in spectral coord
COMMENT  ----- Char.TimeAxis -----
TMID     = 5.1995488293981E+03 / [d] MJD of exposure midpoint
TELAPSE  = 7.5000000000000E+02 / [s] Total elapsed time
COMMENT  ----- Data Fields -----
TUTYP1   = 'spec:Data.SpectralAxis.Value'
TUCD1    = 'em.freq'
TUTYP2   = 'spec:Data.FluxAxis.Value'
TUCD2    = 'phot.flux.density;em.freq'
*****
    The data would look like this..
#      FREQ                FLUX
    8.56549883E+14         1108157630.0
    8.55999451E+14         1092354030.0
    8.55449726E+14         1081555890.0
...
    4.36145619E+14         333332200.0
    4.36002855E+14         330290796.0

```

## C.2.2 Basic PhotometryPoint Instance

This is an example serialization of a Basic PhotometryPoint Dataset, it contains only required elements.

```

XTENSION= 'BINTABLE'           / binary table extension
BITPIX  =           8 / 8-bit bytes
NAXIS   =           2 / 2-dimensional binary table
NAXIS1  =          16 / width of table in bytes
NAXIS2  =           1 / number of rows in table
PCOUNT  =           0 / size of special data area
GCOUNT  =           1 / one data group (required keyword)
TFIELDS =           2 / number of fields in each row
EXTNAME = 'PHOTOMETRYPOINT'   / name of this binary table extension
HDUNAME = 'PHOTOMETRYPOINT'   / ASCDM block name
TTYPE1  = 'FREQ'              / Spectral column name
TFORM1  = '1D'                / format of field
TUNIT1  = 'Hz'                /
TTYPE2  = 'FLUX'              / Flux column name
TFORM2  = '1D'                / format of field
TUNIT2  = 'Jy'                /
COMMENT ----- Dataset -----
VOCLASS = 'PhotometryPoint-1.0' / Datamodel name and version
COMMENT ----- Curation -----
VOPUB   = 'Database'          / Dataset publisher
COMMENT ----- DataID -----
TITLE   = 'Sample Spectrum Data' / Dataset Title
COMMENT ----- Target -----
OBJECT  = '3C 273'            / Target name
COMMENT ----- CoordSys -----
PHZERO  = 1.6144526095200E+03 / Flux value at filter Zero point.
PHFUNI  = 'Jy'                / Unit for zero point flux value.
COMMENT ----- Char.FluxAxis -----
FLUX_UCD= 'phot.flux.density;em.freq' / UCD for flux axis
FLUX_UNI= 'Jy'                / Unit for flux axis
COMMENT ----- Char.SpatialAxis -----
RA      = 1.8727791500000E+02 / [deg] Pointing position
DEC     = 2.0523880000000E+00 / [deg] Pointing position
APERTURE= 5.5500000000000E-04 / [deg] Aperture angular size
COMMENT ----- Char.SpectralAxis -----
SPEC_UCD= 'em.freq'          / UCD for spectral axis
SPEC_UNI= 'Hz'                / Unit for spectral axis
SPEC_VAL=                    / [Hz] Characteristic spectral coord.
SPEC_BW  =                    / [Hz] Width of spectral coordinate
TDMIN2   =                    / [Hz] Start in spectral coord
TDMAX2   =                    / [Hz] Stop in spectral coord
COMMENT ----- Char.TimeAxis -----
TMID     = 5.1995488293981E+03 / [d] MJD of exposure midpoint
TELAPSE  = 2.7010000000000E+03 / [s] Total elapsed time
COMMENT ----- Data Fields -----
TUTYP1   = 'spec:Data.SpectralAxis.Value'
TUCD1    = 'em.freq'
TUTYP2   = 'spec:Data.FluxAxis.Value'
TUCD2    = 'phot.flux.density;em.freq'
*****
    The data would look like this..
#    FREQ                FLUX
    2.4E+14              0.03350

```

## References

- [1] Resource Metadata for the Virtual Observatory: Version 1.12, 02 March 2007  
<http://www.ivoa.net/Documents/latest/RM.html>
- [2] "IVOA Identifiers": Version 1.12, 14 March 2007  
<http://www.ivoa.net/Documents/latest/IDs.html>
- [3] "IVOA Photometry Data Model": Version 1.0, 2011  
<http://www.ivoa.net/Documents/PHOTDM/20111013/WD-PhotDM-1.0-20111013.pdf>
- [4] "Data Model for Astronomical DataSet Characterisation": Version 1.13  
<http://www.ivoa.net/Documents/REC/DM/CharacterisationDM-20080325.pdf>
- [5] "Space-Time Coordinate Metadata for the Virtual Observatory": Version 1.33, 2007  
<http://www.ivoa.net/Documents/REC/DM/STC-20071030.pdf>
- [6] "ST-ECF newsletter, issue #42", June 2007  
[http://www.spacetelescope.org/about/further\\_information/newsletters/html/newsletter\\_42.html](http://www.spacetelescope.org/about/further_information/newsletters/html/newsletter_42.html)
- [7] "DER\_SNR: A simple and general spectroscopic signal-to-noise measurement algorithm";  
[http://www.stecf.org/software/ASTROsoft/DER\\_SNR](http://www.stecf.org/software/ASTROsoft/DER_SNR)
- [8] "OGIP 93-001; "Specification of Physical Units within OGIP FITS files", 1995 May 04  
[http://heasarc.gsfc.nasa.gov/docs/heasarc/ofwg/docs/general/ogip\\_93\\_001/ogip\\_93\\_001.html](http://heasarc.gsfc.nasa.gov/docs/heasarc/ofwg/docs/general/ogip_93_001/ogip_93_001.html)
- [9] "The UCD1+ controlled vocabulary": Version 1.23  
<http://www.ivoa.net/Documents/REC/UCD/UCDlist-20070402.pdf>  
<http://cdsweb.u-strasbg.fr/UCD/ucd1p-words.txt>
- [10] "An IVOA Standard for Unified Content Descriptors": Version 1.10  
<http://www.ivoa.net/Documents/REC/UCD/UCD-20050812.pdf>  
<http://www.ivoa.net/Documents/latest/UCD.html>
- [11] "Units in the VO": Version 1.0  
<http://www.ivoa.net/Documents/VOUnits/20120820/PR-VOUnits-1.0-20120820.pdf>
- [12] "Dimensional Analysis applied to Spectrum Handling in Virtual Observatory Context", Pedro Osuna, Jesus Selgado; Nov. 2005



## Open Issues

- UTypes specification is inadequate for supporting multiple instances of a complex object within a node. There is no mechanism for unambiguously associating the various fields. In this model, this issue effects several components, and becomes a bigger issue with multiband photometry and time series.
  - multiple FluxAxis within Data or Characterization;
  - multiple FluxFrame within CoordSys;
  - multiple QualityCode within QualityDefs;
- The FITS support described in this document is carried from earlier versions. The static keyword mapping is extremely rigid and does not allow for standards migration (RADECSYS/RADESYS), multiple object definitions (FluxAxis), or user extensibility. It would be far more flexible and generic to define a mapping mechanism which users can ADD to their data products which would map the FITS information (keys) to specific model concepts (UTypes). This would be a major change from earlier versions of the document, so is not included here. It should be discussed as a possibility for future enhancement to IVOA FITS serialization conventions.