International
Virtual
Observatory
Alliance

IVOA Spectral Data Model

Version 2.0

IVOA Proposed Recommendation 20150206

This version:

PR-SpectralDM-2.0-20150206

Previous version(s):

PR-SpectralDM-2.0-20140730
PR-SpectralDM-2.0-20140309
PR-SpectralDM-2.0-20130425
WD-SpectralDM-2.0-20130308
WD-SpectralDM-2.0-20120907
SpectrumDM-20111020

Editor(s):

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.

2014 Mar 09:

Updates resulting from observation-dataset review and reference implementation notes:
- Replace figures with UML diagrams.
- Fold in Observation/Dataset hierarchy. Define generic Dataset as separate entity.
- Added ObsConfig and Proposal objects to Dataset tree.
- Moved Instrument, Bandpass, Datasource from DataID to ObsConfig object.
- Define SpectralDataset as extension of generic Dataset, replacing BaseSPS
- Allow multiple CoordSys objects, restore Frame hierarchy to STC form
  NOTE: Frames themselves still simplified.
- Replaced unused "Data" container with SPPoint object used by Spectrum/PP.
- Datatypes created for Enumerated values
- Date type created with format comments consistent with DAL (not quite ISO8601)
- Converted source document from LaTex to OpenOffice odt.

2014 Jul 30:

Updates from RFC comments:
- Small changes to language describing relation of 'domain' axes to CharacterisationAxis.
- Change UType of CalibStatus -> CalibrationStatus.

2015 Feb 06:

Updates from TCG comments:
- update architecture diagram
- add content to use-cases section

- removed numerical clarifiers (e.g. "Zero (0) or more" => "Zero or more")
- expanded/clarified the descriptions for several fields, notably DatasetID related fields, Collections, Target attributes..
- fields with multiplicity > 1 should retain singular name (e.g. DataID.collection)
- update diagrams for elements experiencing a name change from above.
- sweep document correcting capitalization of attributes.

Contents

1 Introduction

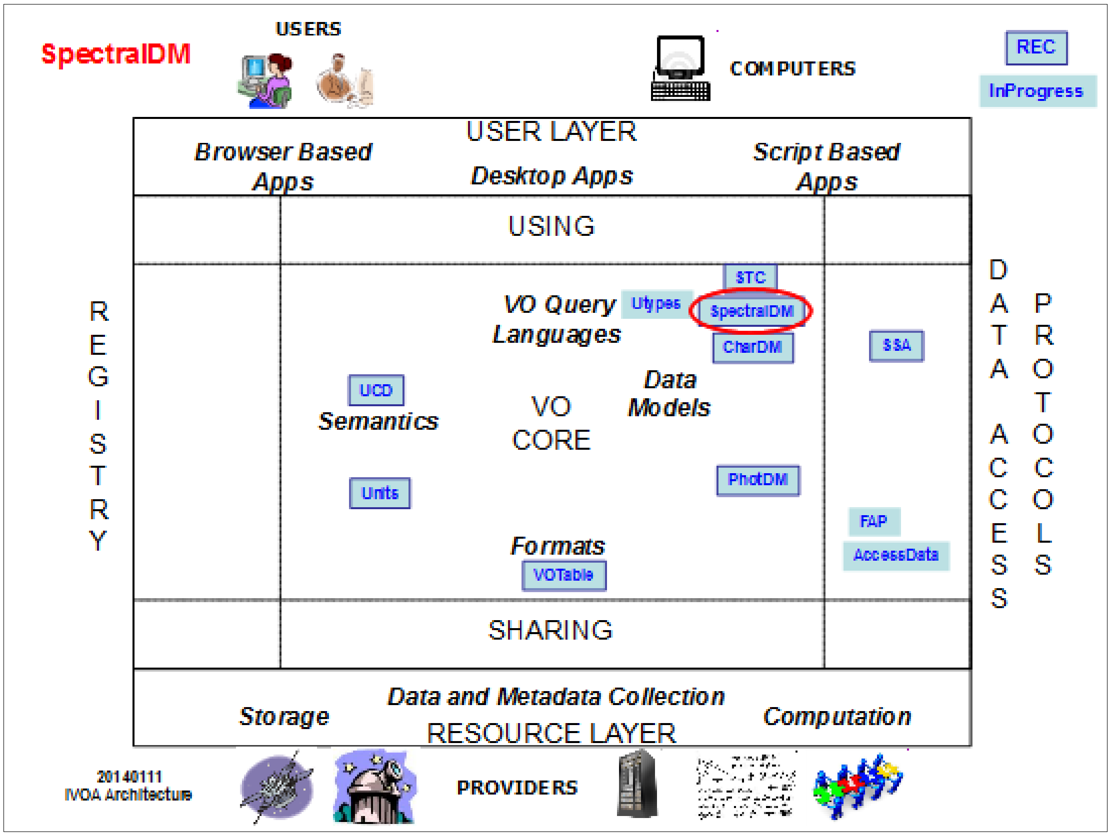
1.1 Motivation

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 characterisation 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 Role in the IVOA Architecture



SpectralDM incorporates and extends elements from other core IVOA Data Models, namely; STC, and Characterisation. It also provides a connection to the PhotDM, enabling the specification of photometry metadata. As with most IVOA Data Models, SpectralDM makes use of the standard IVOA Semantics for UCDs and Units, as well as the de-facto Utypes convention. SpectralDM instances are intended to be discovered and accessed via IVOA protocols such as SSA (Simple Spectral Access).

1.3 Use Cases

The scope of this model is to define the framework for representing any of a wide range of spectro-photmetric datasets within the VO. In this document, we include specific models for the Spectrum and Photometry point cases. As such, our primary use cases focus on these.

1) Facilitate interoperability of spectro-photometric datasets within the VO.

By providing a generalized framework for spectro-photometric data, and their representation, analysis tools may be developed which seamlessly integrate data from a variety of wavebands, missions and data centers, and interactively share the content with other VO/SpectralDM aware applications.

For example, an application can combine an Astro-1 spectrum dataset extracted from the MAST archive, with photometry data extracted from NED, to generate an SED for a particular object. This can then be fit, and compared against a theoretical spectrum obtained from yet another provider.

2) Facilitate the VO-compliant serialization of spectro-photometric data.

Data providers wishing to publish Spectrum or Photometry point data to the VO may make use of this model to translate their holdings to the generalized model. Then, by applying the serialization conventions for the desired format, produce VO-compliant datasets ready for discovery by VO services such as TAP or SSA.

a) Spectrum:

The Spectrum model in this document supports a single spectral-flux coordinate axis pair. To provide both flux-vs-wavelength and flux-vs-frequency for a single dataset requires the generation of two separate instances.

b) Photometry point:

The Photometry point model in this document represents a single measurement, expressed as a spectral-flux coordinate axis pair.

These models may be used to generate stand-alone products (ie. a Spectrum dataset), or combined as components in more complex datasets such as Spectral Energy Distributions (SED), or Photometry catalogues. The standardized models for these datatsets are beyond the scope of this document.

3) Support the discovery of spectro-photometric datasets within the VO.

Development of specialized access protocols such as Simple Spectral Access (SSA), may make use of the objects and metadata definitions provided by this model as a basis for their own components, such as:

- definition of query parameters for various properties
- definition of query response objects which describe instances of these datasets.

Doing so provides a certain consistency between the access protocol and the products they are serving.

1.4 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.

1.5 Structure of this Documentation

+ Major sections for each model area (Dataset, Spectral, Characterisation, etc).

+ First subsection in each section is the primary element within that model

+ Subsequent subsections for secondary elements, in alphabetical order.

+ Each subsection has sub-subsections for each attribute/relation

  - attributes show the full definition including datatype and usage.

  - relations describe the usage of the object in that context, and a reference to the full definition.

+ Capitalization convention

  - Objects and complex data types are expressed in PascalCase

  - attributes are camelCase

  - primitive data types (string, double, etc) are lower case

2 Observation-Dataset

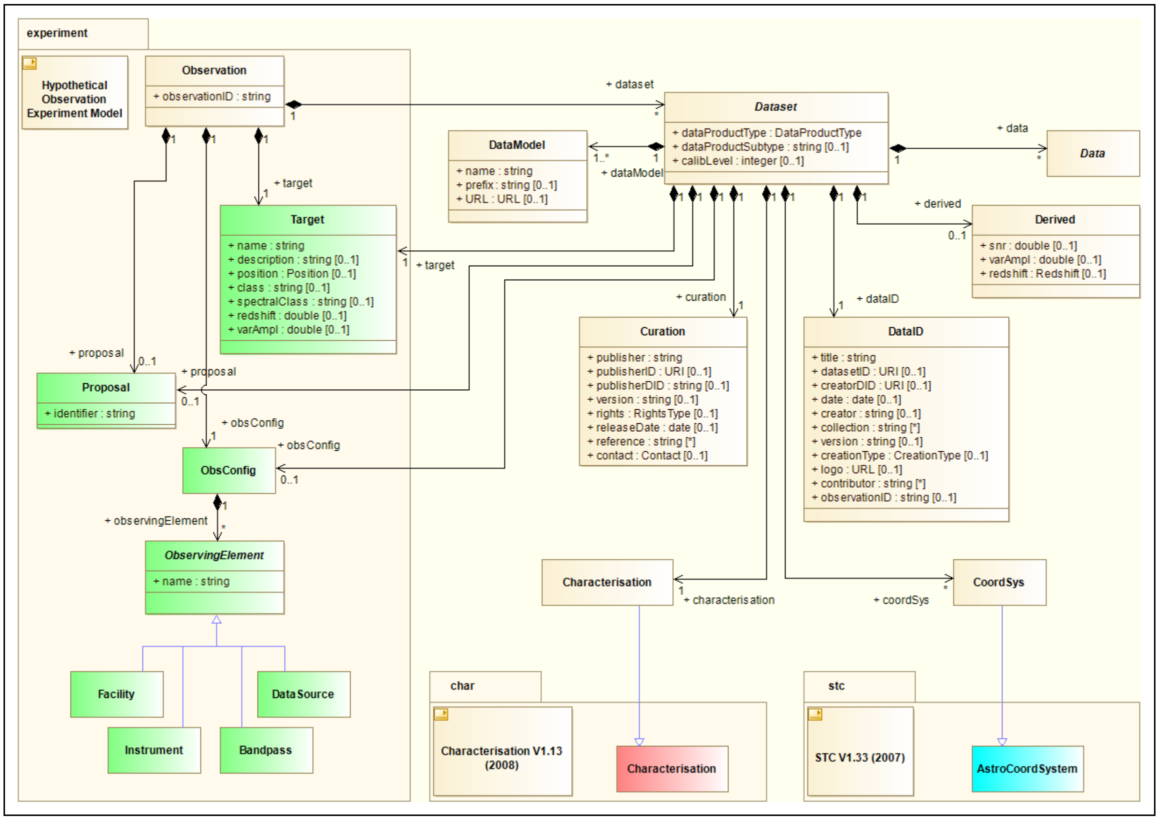


Illustration 1: Observation-Dataset detail diagram

This section describes the general Astronomical Dataset derived from an Observation. It provides the general framework for all types of datasets within the IVOA models. The diagram shows several elements to be drawn from an Experiment/Observation model. At the time of this writing, there is no such model. We define these placeholder elements here, with the expectation that they will be replaced by equivalent objects when that model is created.

2.1 Dataset

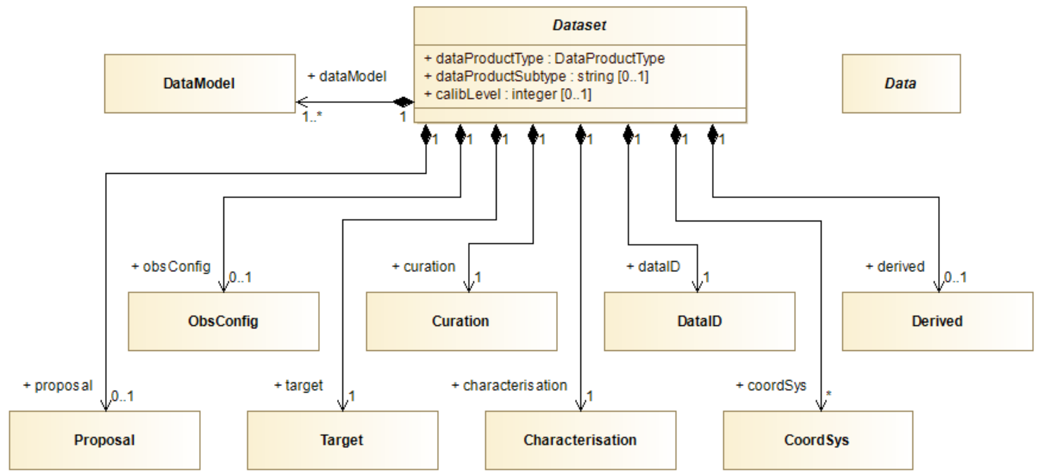


Illustration 2: Dataset detail

Abstract object for the generic IVOA Dataset. It is intended to be useful for any type of data. Specific dataset models should extend this object, providing detailed definitions and additional content as appropriate for that type of dataset.

2.1.1 Dataset.dataProductType:DataProductType

Indicates the makeup of the data content. Allowed values are provided in the enumeration list DataProductType in Section 6.4.2.1.

2.1.2 Dataset.dataProductSubtype:string

Secondary type identifier for the dataset. This field is intended to precisely specify the scientific nature of the data product, possibly in terms relevant only to a specific archive or data collection. For example, dataProductType="image" could have associated dataProductSubtype="src.image", "bkg.image", "pixelMask", etc. Values are unrestricted strings.

2.1.3 Dataset.calibLevel:integer

High level classification for the calibration level of a particular dataset as a whole. The calibration level concept conveys to the user information on how much data reduction/processing has been applied to the data. It is up to the data providers to consider how to map their own internal classification to the scale defined here.

Scale:

- 0 - Raw instrumental, in a proprietary or internal data-provider defined format.
- 1 - Instrumental data in a standard format (FITS, VOTable, etc)
- 2 - Calibrated, science ready data with the instrument signature removed.
- 3 - Enhanced data products like mosaics, re-sampled or drizzled images, or heavily processed survey fields. Level 3 data products may represent the combination of data from multiple primary observations.

2.1.4 Dataset.characterisation:Characterisation

See Section 2.2 for content definition.

Characterisation provides a 'characteristic' view of the coordinate space for the dataset as a whole.

2.1.5 Dataset.coordSys:CoordSys

See Section 2.3 for content definition.

Zero or more coordinate system definitions associated with the dataset.

2.1.6 Dataset.curation:Curation

See Section 2.4 for content definition.

Curation object associated with the dataset.

2.1.7 Dataset.dataID:DataID

See Section 2.6 for content definition.

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

2.1.8 Dataset.dataModel:DataModel

See Section 2.7 for content definition.

Provides metadata related to the data model(s) represented within a serialization of a dataset. 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.

2.1.9 Dataset.derived:Derived

See Section 2.8 for content definition.

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

2.1.10 Dataset.obsConfig:ObsConfig

See Section 2.9 for content definition.

ObsConfig provides some high-level metadata related to the observation configuration.

2.1.11 Dataset.proposal:Proposal

See Section 2.10 for content definition.

Metadata identifying any proposal related to the observation which produced the dataset.

2.1.12 Dataset.target:Target

See Section 2.11 for content definition.

Target object associated with the dataset.

2.2 Characterisation

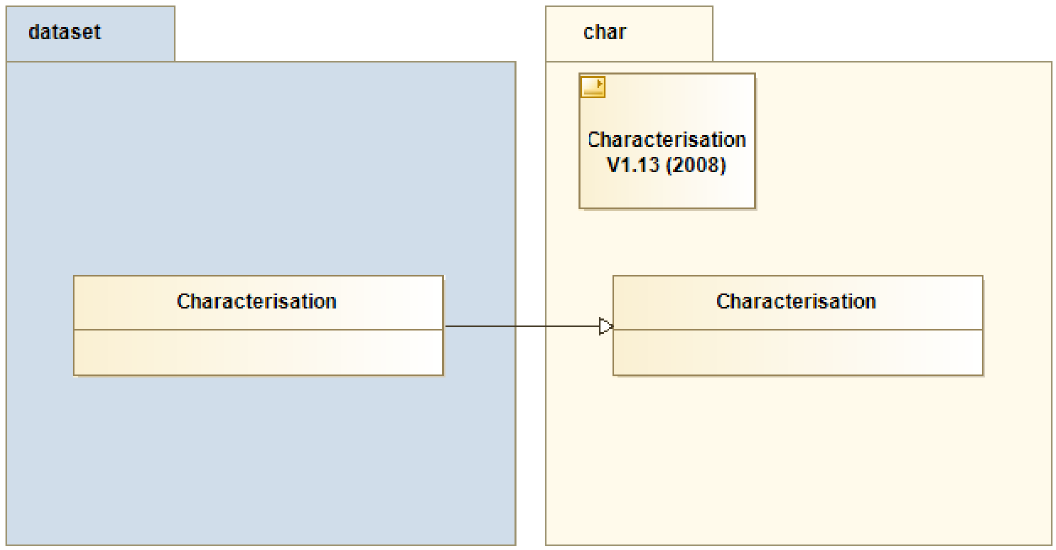


Illustration 3: Characterisation detail

Generic dataset Characterisation is a direct extension of the Characterisation object from the IVOA Characterisation Data Model Standard [4]. Specific dataset models may extend and/or override this model as appropriate for that model.

There is a strong correlation between the information provided here, with that provided under the Data element. The distinction is that characterisation 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 characterisation represent typical accuracy for the dataset, while those within Data provide per-data-point errors.

2.3 CoordSys

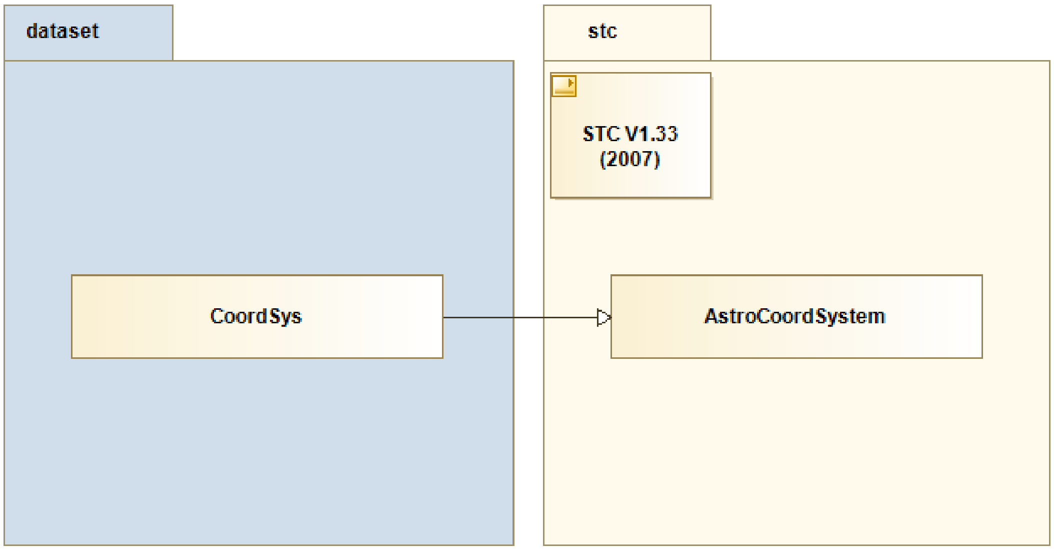


Illustration 4: CoordSys detail

The CoordSys object in the Dataset model is a direct extension of the AstroCoordSystem object defined by the STC Model [5]. Specific dataset models may extend and/or override this model as appropriate for that model.

2.4 Curation

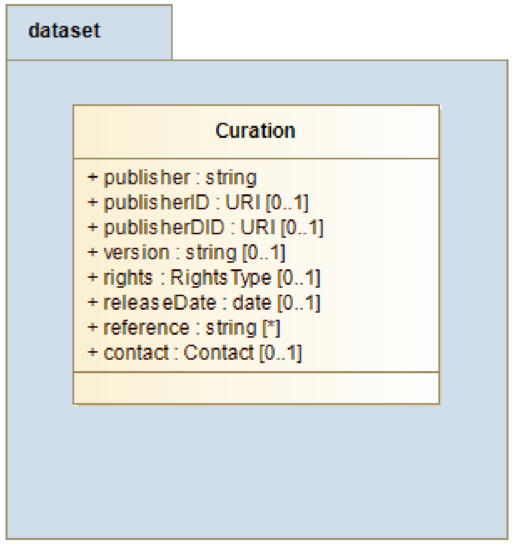


Illustration 5: Curation detail

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.

2.4.1 Curation.publisher:string

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

2.4.2 Curation.publisherID:URI

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

2.4.3 Curation.publisherDID:URI

IVOA dataset identifier assigned by the publisher and unique within the namespace controlled by the publisher (ie data center). Typically, this would be the same as the DataID.DatasetID, but may differ in order to distinguish the same dataset held by multiple data centers. Values are to be expressed as dataset identifiers using the syntax described in "IVOA Identifiers"[2]. (RM:Resource.Identifier)

2.4.4 Curation.releaseDate:Date

Date the curated dataset was last modified. (RM:Curation.Date)

Date type and format restrictions are described in Section 6.4.1.1.

2.4.5 Curation.version:string

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

2.4.6 Curation.rights:RightsType

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

RightsType enumerated value list is detailed in Section 6.4.2.3.

2.4.7 Curation.reference:string

Zero or more bibliographic or documentation references associated with the dataset. This element may occur multiple times, one for each instance. Each instance contains a forward link to a major publication which references the dataset. Values should be expressed as a URL, or bibcode (discernable as a 19 character string beginning with 4 four digits). Free text references are allowed, but discouraged. (RM:General.Source)

2.4.8 Curation.contact:Contact

Contact information of the person/entity responsible for the content of the dataset. (RM:Curation.Contact) The Contact datatype is described in Section 6.4.2.4.

2.5 Data

The Data object is an abstract container providing access to the data content of the dataset. Specific dataset types should extend and populate this object, and define the relation to it.

2.6 DataID

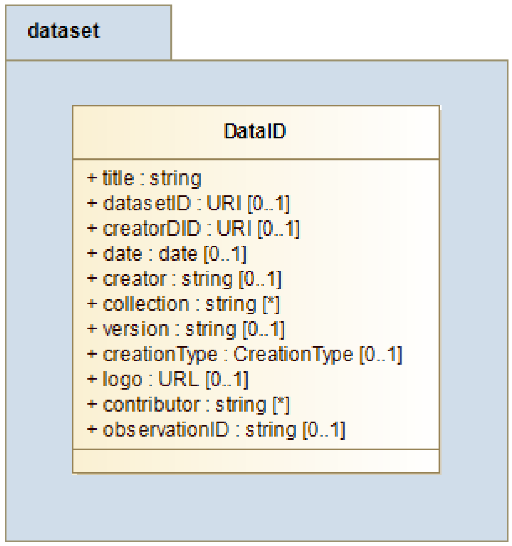


Illustration 6: DataID detail

The Data Identification object (DataID) stores the dataset identifiers and its membership within 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. e.g., ivo://example.net/aservice?2013/5/2342.

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.

2.6.1 DataID.title:string

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

2.6.2 DataID.creator:string

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

2.6.3 DataID.collection:string

The dataset is associated with zero or more Collections (instrument name, survey name, etc.) indicating some degree of compatibility with other data sharing the same Collection properties. This element may occur multiple times, once for each collection, expressed as a free form string. Specific values are generally defined and set by the creating entity. Examples: "WFC", "Sloan", "BFS Spectrograph", "MSX Galactic Plane Survey".

2.6.4 DataID.datasetID:URI

Public dataset identifier assigned by the publisher. While the Curation.PublisherDID field is unique to the data center and used to identify the dataset within it's holdings, this identifier is unique within the VO, and suitable for linking journal articles back to the corresponding archival dataset. We recommend a journal-based URI such as the IVOA/ADEC/ADS dataset identifier.

2.6.5 DataID.creatorDID:URI

The dataset identifier assigned by the creator. Here, the authority-id of the identifier must be that of the creator. The value may be entirely different from that of the DatasetID field as it is used to identify the 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.

2.6.6 DataID.date:Date

Data processing or creation date (RM:Curation.Date). See Section 6.4.1.1 for Date format specification.

2.6.7 DataID.version:string

Version of the creator-produced dataset.

2.6.8 DataID.creationType:CreationType

The dataset creation type describes the nature or genre of the content. (RM:General.Type). The creationType enumeration set is defined in Section 6.4.2.2.

2.6.9 DataID.logo:URL

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

2.6.10 DataID.contributor:string

The dataset is associated with zero or more entities which contributed to the content of the document. This element may be occur multiple times, once for each entity, expressed as a free form string. (RM:Curation.Contributor)

2.6.11 DataID.observationID:string

Internal ID determined by the data provider to identify the observation from which the dataset was produced.

2.7 DataModel

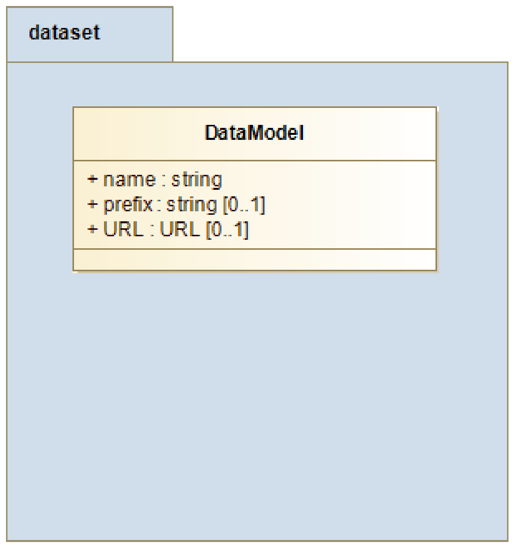


Illustration 7: DataModel detail

This object provides specific information regarding the data model and version thereof that a dataset represents. This information is primarily an aide to (de)serialization of the dataset. As such, it would better be defined by serialization conventions. In lieu of such conventions, we include this object in the model proper.

2.7.1 DataModel.name:string

Free format string provides the name and version of the model document. Recommended format is "[name]-[version].[subversion]". Each IVOA standard model defines this value and serializations of these models MUST match this defined string. Extended, or user defined model can specify a unique name for their own content. See sections 7.1.5 and 8.1.5 for values defined for the Spectrum and PhotometryPoint models respectively.

2.7.2 DataModel.prefix:string

Prefix string used in UType strings for elements associated with this model. For IVOA standard models, this will be a predetermined string (e.g. 'stc' or 'spec') defined within that model. Serializations of these models may provide the string here, or rely on the default. Either way, the value MUST match that defined in the model. Extended, or user defined model can specify a unique prefix for their own content. See sections 7.1.5 and 8.1.5 for values defined for the Spectrum and PhotometryPoint models respectively.

2.7.3 DataModel.URL:URL

URL pointer to the XML Schema associated with the model.

2.8 Derived

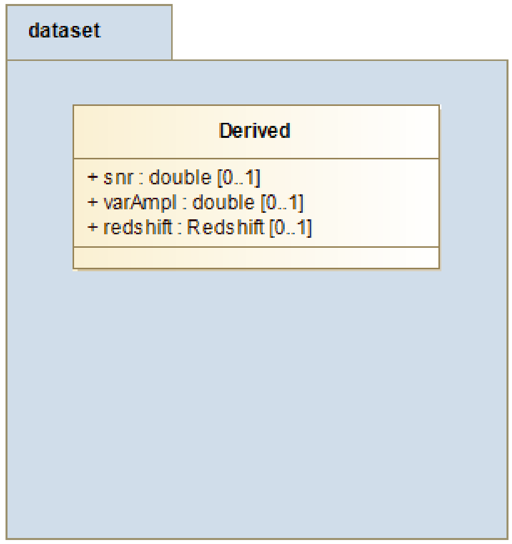


Illustration 8: Derived Data detail

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

2.8.1 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/STECF/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.

2.8.2 Derived.redshift:Redshift

This field represents a measurement of the redshift from the data. We provide fields for both the redshift measured value and statistical error. See Section 6.4.2.5 for details on the Redshift datatype.

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.

2.8.3 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).

2.9 ObsConfig

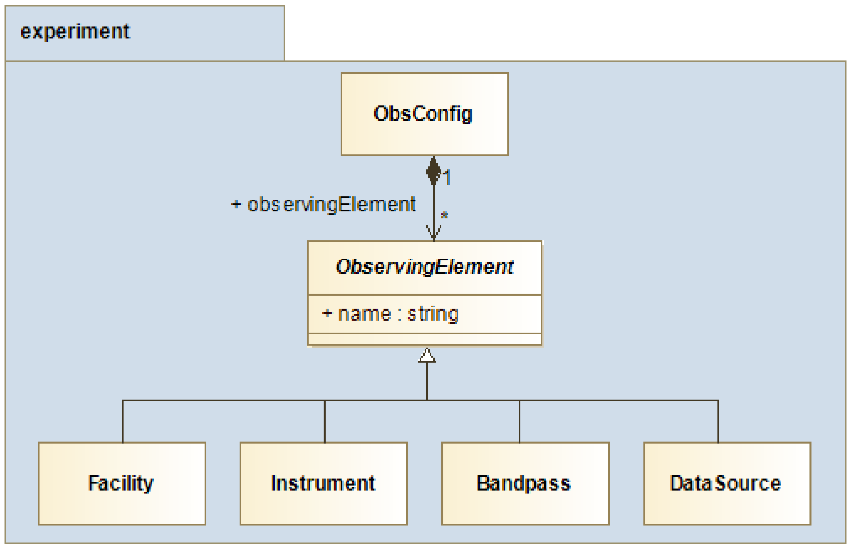


Illustration 9: ObsConfig detail

2.9.1 ObsConfig.observingElement:ObservingElement

Collection of zero or more ObservingElements which define observation parameters.

2.9.2 ObsConfig.Facility.name:string

Name of the facility performing the observation.

2.9.3 ObsConfig.Instrument.name: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.

2.9.4 ObsConfig.DataSource.name:string

Describes the original source of the data in a very general fashion. In other words, "What sort of observation originally generated the data?". Suggested values include:

- **survey**: Survey data typically covers some region of observational parameter space with as complete as possible coverage within that region.
- **pointed**: Pointed data of a particular object or field.
- **theory**: Theory data, generated based on a theoretical model.
- **artificial**: Artificial, or simulated data. Similar to 'theory', but not necessarily based on a theoretical model.
- **custom**: Custom data, as part of a specific research product.

2.9.5 ObsConfig.BandPass.name: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.

2.10 Proposal

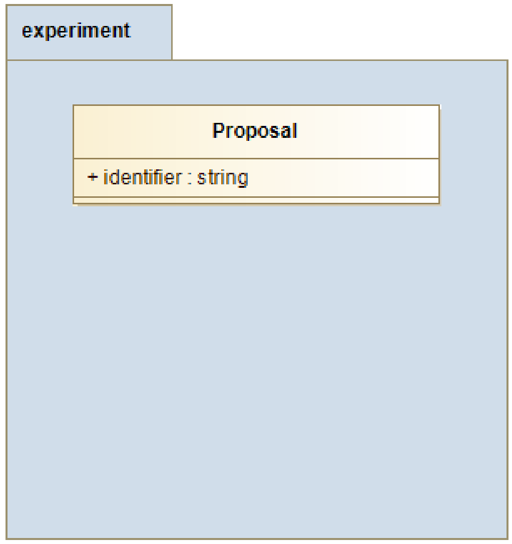


Illustration 10: Proposal detail

2.10.1 Proposal.identifier:string

Identifier of the proposal to which the observation belongs.

2.11 Target

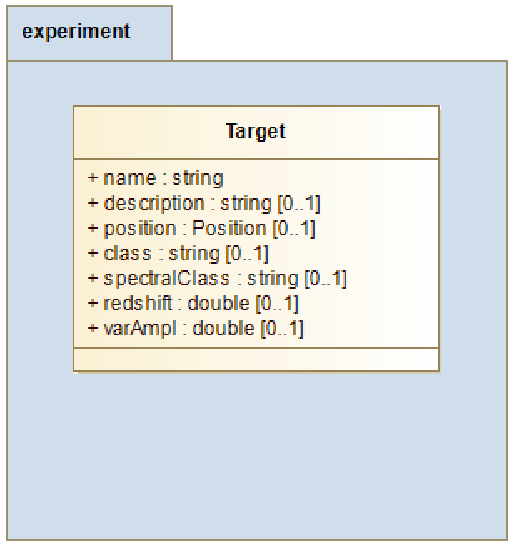


Illustration 11: Target detail

In astronomical 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 data for which the Coverage fields may not be relevant.

2.11.1 Target.name:string

Target name. The primary purpose of this field is to provide the user with a recognizable identity of the 'subject'. When referring to an astronomical target, one may specify a particular object, or a more general target such as the name of a survey field. When specifying a particular object, it is highly recommended to use a name suitable for input to a domain-specific resolution service, e.g. SIMBAD or NED.

2.11.2 Target.description:string

Free form description of target.

2.11.3 Target.position:Position2

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. See Section 6.4.2.8 for definition of Position2 type.

2.11.4 Target.class:string

General classification or type of the target. This field supports the discovery of data pertaining to a common class. e.g. "Star", "Galaxy", "AGN". At the time of this writing, there is no IVOA recommended vocabulary for this field. The SIMBAD and NED databases use defined vocabularies for astronomical object classifications which may serve as the basis for such.

2.11.5 Target.spectralClass:string

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

2.11.6 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.

2.11.7 Target.varAmpl:double

Typical target variability amplitude. Range 0:1.

3 Spectral Model

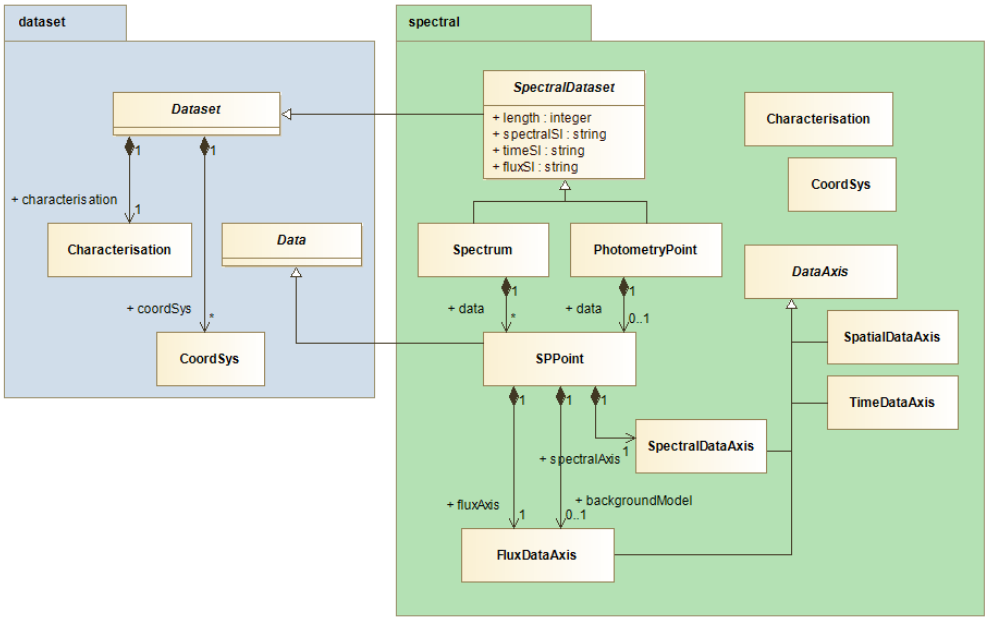


Illustration 12: Spectral Dataset Model

This section describes the dataset model for Spectral regime data products.

3.1 SpectralDataset

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 PhotmetryPoint (Section 8) extensions.

3.1.1 SpectralDataset.length:integer

The number of points in the data.

3.1.2 SpectralDataset.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 6.2 for more information on that format.

3.1.3 SpectralDataset.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 6.2 for more information on that format.

3.1.4 SpectralDataset.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 6.2 for more information on that format.

3.2 characterisation

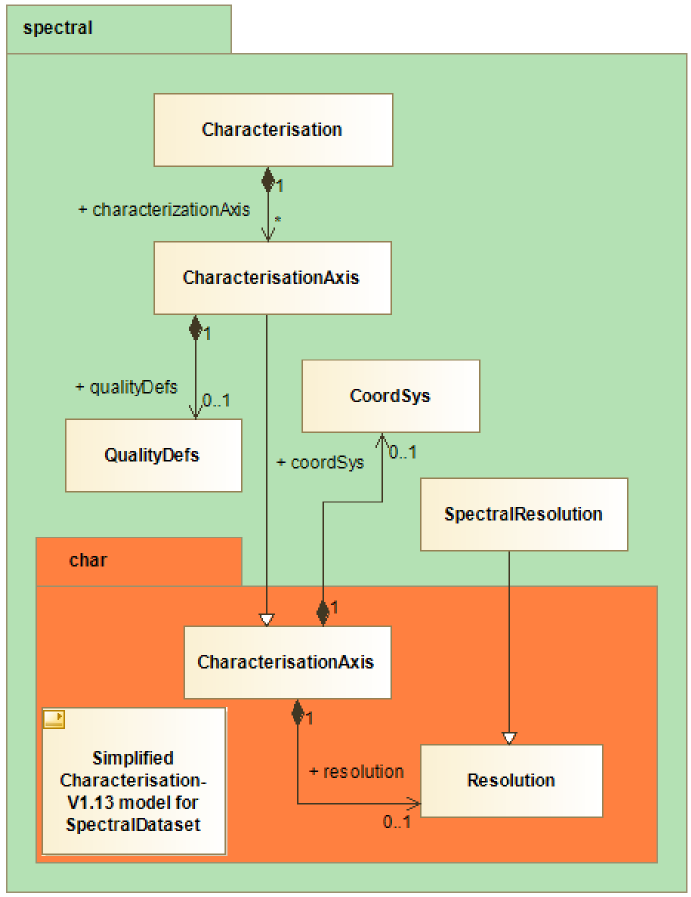


Illustration 13: SpectralDataset Characterisation

This SpectralDataset model overrides the generic Datatset Characterisation object, and extends the definition for objects required by this model. It defines a set of objects which are consistent with the IVOA Characterisation Data Model Standard[4], but which impose certain restrictions and simplifications on several elements of that model. Section 4 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.

3.3 CharacterizationAxis

The primary content of CharacterizationAxis is defined within the Characterization model. Refer to Section 4.2 of this document for details on the definition used in this model. We extend that definition with additional content described below.

3.3.1 CharacterizatonAxis.coordSys:CoordSys

In Section 3.4 we define a modified CoordSys object for this model to accommodate the Spectral specific FluxFrame. This object overrides the definition given in the Characterzation model. Its usage under CharacterizatonAxis is unchanged, and is described in Section 4.2.

3.3.2 CharacterizationAxis.qualityDefs:QualityDefs

We extend the CharacterizationAxis object with an optional QualityDefs container for a set of one 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.

3.3.3 CharacterizationAxis.spectralResolution: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.

3.4 CoordSys

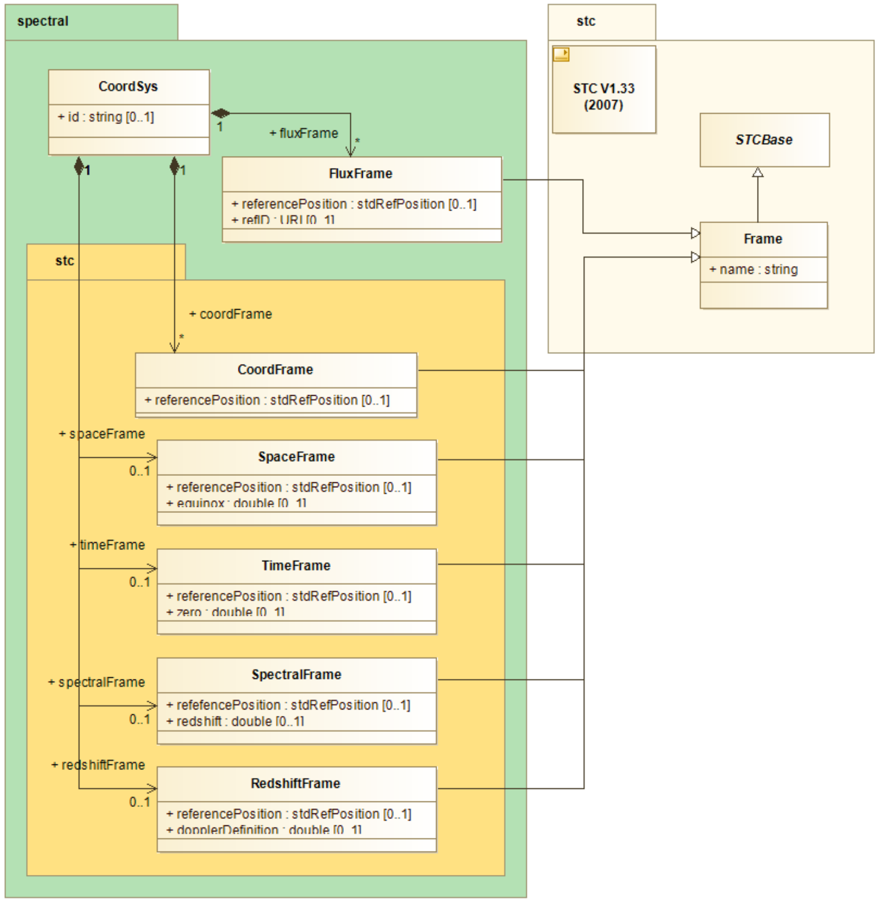


Illustration 14: SpectralDataset CoordSys detail

The SpectralDataset overrides the CoordSys definition given in the base Dataset model as being a direct extension of the STC:AstroCoordSystem object. Here, the CoordSys object is consistent with the AstroCoordSystem object, but modified and extended for Spectral specific requirements.

+ The Frame hierarchy is a simplified set of specialized frame definitions.

+ Addition of zero or more FluxFrames.

See "STC Model Elements" in Section 5 for descriptions of the Frame hierarchy used in this model.

3.4.1 CoordSys.ID:string

User defined free form string giving the ID for the coordinate system as a whole.

3.4.2 CoordSys.coordFrame:CoordFrame

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

3.4.3 CoordSys.spaceFrame:SpaceFrame

Zero or one specialized SpaceFrame object.

3.4.4 CoordSys.spectralFrame:SpectralFrame

Zero or one specialized SpectralFrame object.

3.4.5 CoordSys.timeFrame:TimeFrame

Zero or one specialized TimeFrame object.

3.4.6 CoordSys.redshiftFrame:RedshiftFrame

Zero or one specialized Redshift object.

3.4.7 CoordSys.fluxFrame:FluxFrame

Zero or more specialized FluxFrame objects. This object is an extension of the STC Frame set required for Spectral class data. It is described in Section 3.9.

3.5 Corrections

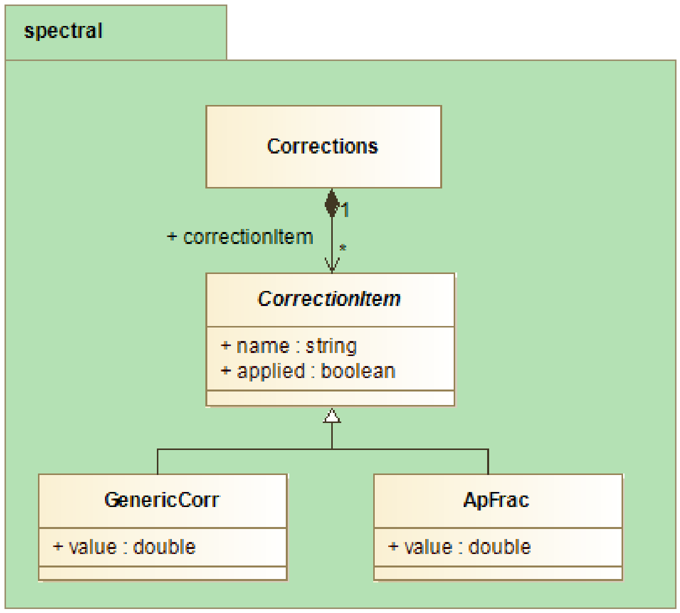


Illustration 15: Corrections detail

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

3.6 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".

3.6.1 CorrectionItem.name:string

Free format, user defined name of the correction type.

3.6.2 CorrectionItem.applied:boolean

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

3.7 GenericCorr

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

3.7.1 GenericCorr.value:double

Size of scale of the correction.

3.8 ApFrac

Extension of CorrectionItem specifying an Aperture Fraction correction.

This particular correction is conceptually quite close to the Spatial Filling Factor defined in the IVAO 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

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

3.8.1 ApFrac.name:string

Attribute value fixed as "ApFrac"

3.8.2 ApFrac.value:double

Value between 0.0 and 1.0

3.8.3 ApFrac.applied:boolean

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

3.9 DataAxis

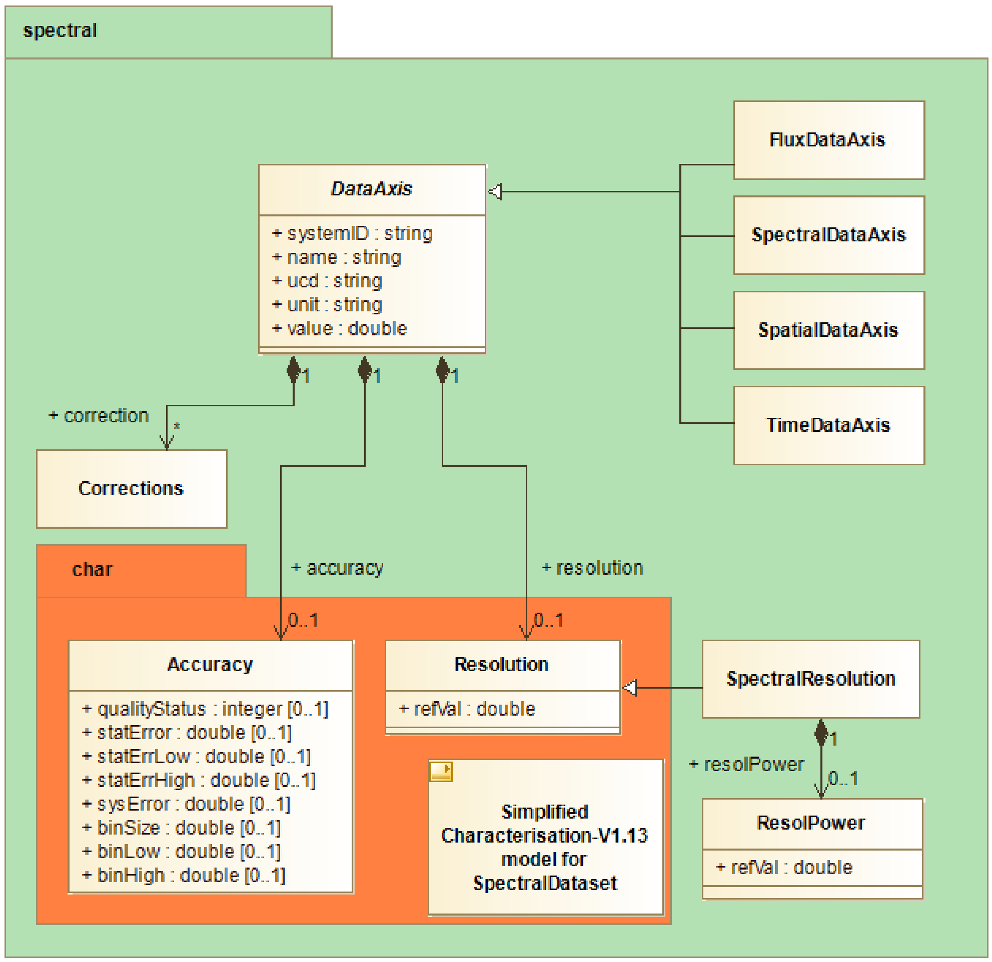


Illustration 16: DataAxis detail

The DataAxis object reuses many elements which are described in the Characterization model for the support of the ChacterizationAxis 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 ChacterizationAxis (see Section 4.2).

3.9.1 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.

3.9.2 Domain subclasses

We allow the same set of domain specializations and associated labels as described in Section 4.2 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.

3.9.3 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).

3.9.4 DataAxis.name:string

Free form name of the axis.

3.9.5 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.

3.9.6 DataAxis.unit:string

Unit for the axis, must comply with VOUnit standard.

3.9.7 DataAxis.value:double

Value of the data point along that particular axis.

3.9.8 DataAxis.accuracy:Accuracy

Accuracy object as described in Section 4.3.

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).

3.9.9 DataAxis.correction:Correction

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

3.9.10 DataAxis.resolution:Resolution

The Resolution object as described in Section 4.8

3.10 FluxFrame

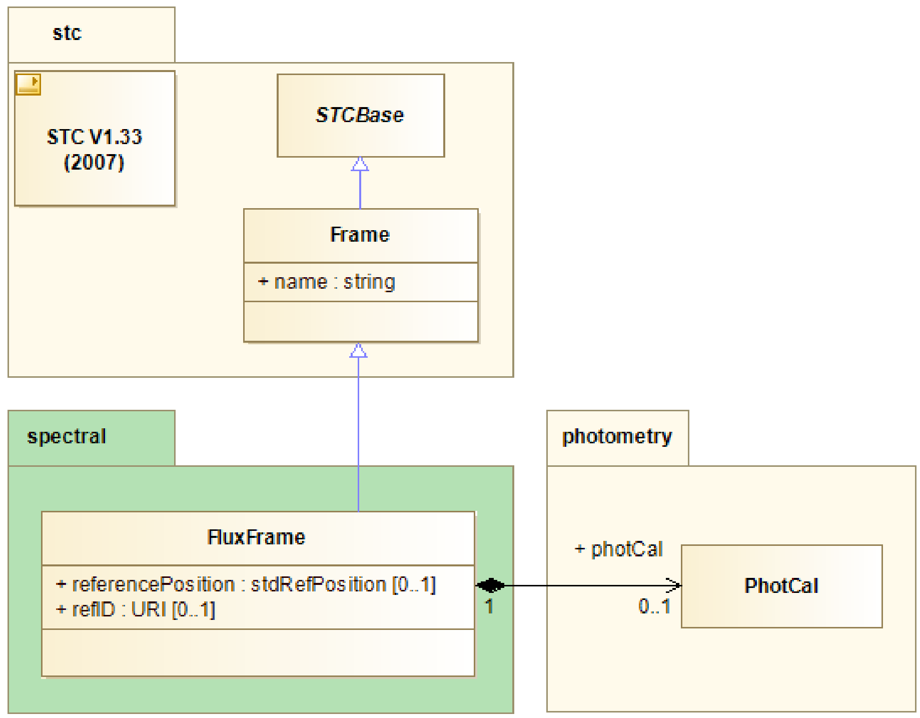


Illustration 17: FluxFrame detail

Extension of the STC Frame 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.

3.10.1 FluxFrame.name:string

This 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.

3.10.2 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].

3.10.3 FluxFrame.photCal: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.

3.11 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 characterisation 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 6.1 - Model Reuse and UTypes for a description of how these model components are incorporated into this model.

3.12 QualityCode

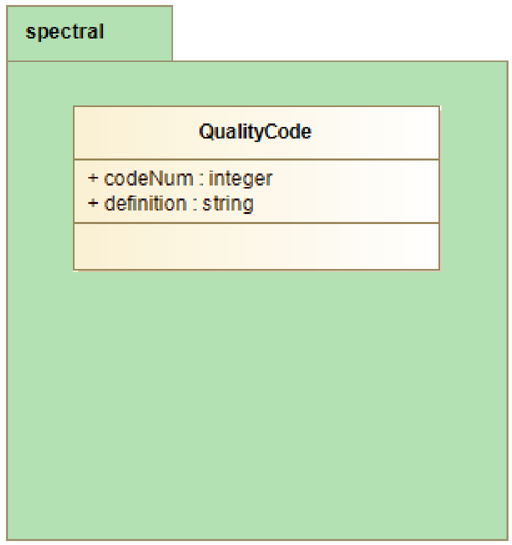


Illustration 18: QualityCode detail

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.

3.12.1 QualityCode.codeNum:integer

Numeric value of the quality code.

3.12.2 QualityCode.definition:string

Free form string description of what the numeric value means.

3.13 SpectralResolution

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

3.13.1 SpectralResolution.resolPower: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.

3.13.2 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.

3.14 SPPoint

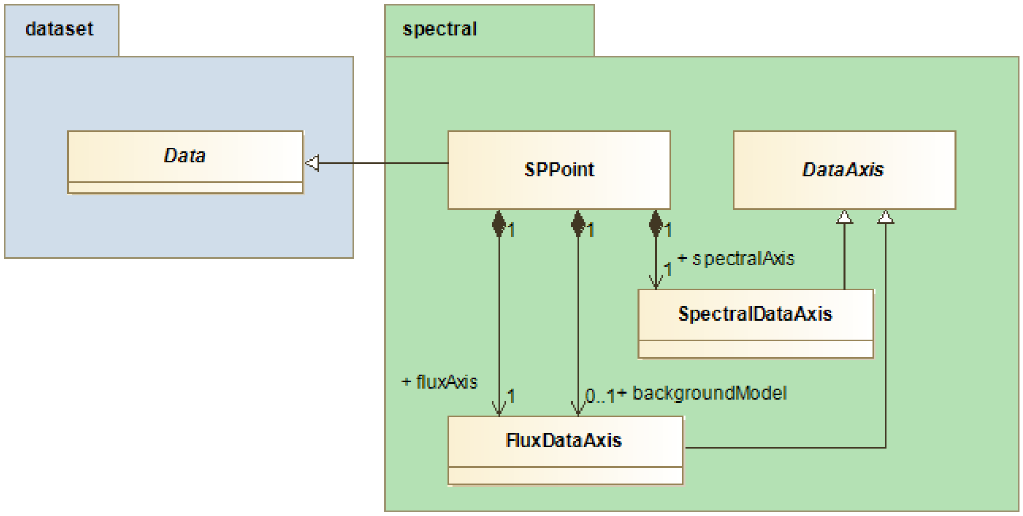


Illustration 19: Spectro-Photometric Point detail

The SPPoint object is a container holding a set of data values related to a Spectro-Photometric Point. It is comprised of a specific set of specialized DataAxis objects, providing values, errors, and other associated data along that axis. DataAxis hierarchy is detailed in Section 3.9.

3.14.1 Point.spectralAxis:SpectralDataAxis

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

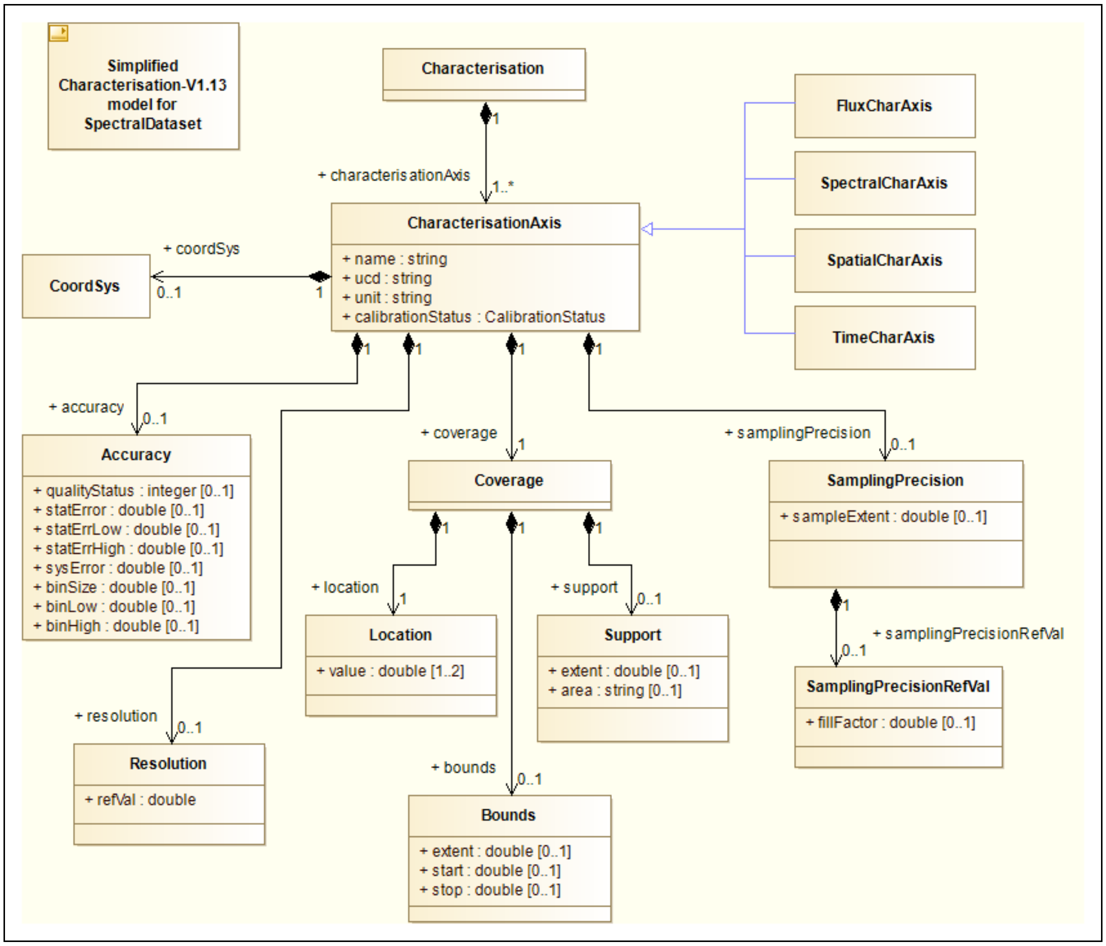
3.14.2 Point.fluxAxis:FluxDataAxis

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

3.14.3 Point.backgroundModel:FluxDataAxis

An additional FluxDataAxis instance to provide background flux data.

4 Characterisation Data Model Elements



4.1 Characterisation (Char)

The Characterisation object and components described in this document are consistent with the "Data Model for Astronomical DataSet Characterisation"[4] (Characterisation 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 Characterisation 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 Characterisation object is a container holding a set of CharacterisationAxis 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.

4.1.1 Char.characterisationAxis

The Characterisation object contains a set of one or more CharacterisationAxis objects.

4.2 CharacterisationAxis (CharAxis)

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

We define a set of specialized CharacterisationAxis 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 labels 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 Characterisation Model "Property versus Axis" Table. Of these, SpatialAxis, TimeAxis, and SpectralAxis are consistent with the Characterisation model. FluxAxis is added for this use case.

4.2.1 CharAxis.name:string

Free form name of the axis.

4.2.2 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.

4.2.3 CharAxis.unit:string

Default unit for the axis.

4.2.4 CharAxis.calibrationStatus:CalibrationStatus

Calibration flag for the axis. The CalibrationStatus enumeration is detailed in Section 6.4.3.4.

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.

4.2.5 CharAxis.accuracy:Accuracy

Accuracy object associated with this axis.

4.2.6 CharAxis.coordSys:CoordSys

We override the Characterisation 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.

4.2.7 CharAxis.coverage:Coverage

Coverage object associated with this axis.

4.2.8 CharAxis.resolution:Resolution

Resolution object associated with this axis.

4.2.9 CharAxis.samplingPrecision:SamplingPrecision

SamplingPrecision object associated with this axis.

4.3 Accuracy

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

4.3.1 Accuracy.binSize:double

Bin width. NOTE: Added for DataAxis support, not part of Characterisation model.

4.3.2 Accuracy.binLow:double

Minimum bin limit. NOTE: Added for DataAxis support, not part of Characterisation model.

4.3.3 Accuracy.binHigh:double

Maximum bin limit. NOTE: Added for DataAxis support, not part of Characterisation model.

4.3.4 Accuracy.statError:double

Error width

4.3.5 Accuracy.statErrLow:double

Low Error

4.3.6 Accuracy.statErrHigh:double

Upper Error

4.3.7 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).

4.3.8 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 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.

4.4 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 Characterisation model.

4.4.1 Coverage.location:Location

4.4.2 Coverage.bounds:Bounds

4.4.3 Coverage.support:Support

4.5 Location

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

4.5.1 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 values, while TimeAxis need only provide one.

4.6 Bounds

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

4.6.1 Bounds.extent:double

Width of the coverage, centered on Coverage.Location

4.6.2 Bounds.start:double

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

4.6.3 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.

4.7 Support

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

4.7.1 Support.extent:double

4.7.2 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.

4.8 Resolution

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

4.8.1 Resolution.refVal:double

A typical or average value for the resolution.

4.9 SamplingPrecision

4.9.1 SamplingPrecision.sampleExtent:double

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

4.10 SpatialCharAxis

Denoted by the UType node "SpatialAxis", this specialization of the general CharacterisationAxis 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
CalibrationStatus	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.

4.11 SpectralCharAxis

Denoted by the UType node ”SpectralAxis”, this specialization of the general CharacterisationAxis 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.
CalibrationStatus	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.

4.12 TimeCharAxis

Denoted by the UType node "TimeAxis", this specialization of the general CharacterisationAxis 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.
CalibrationStatus	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

4.13 FluxCharAxis

Denoted by the UType node "FluxAxis", this specialization of the general CharacterisationAxis 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	
StatErrLow	(n/a)
StatErrHigh	(n/a)
SysError	measurement error
CalibrationStatus	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)

5 STC Data Model Elements

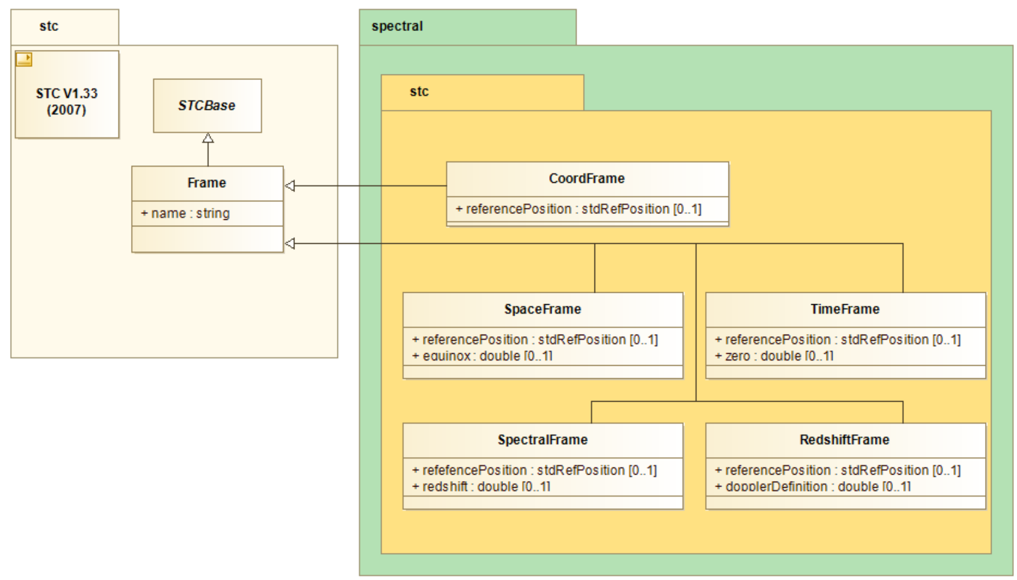


Illustration 20: Spectral Model Frame detail

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:

• 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 defined as object elements. Here they are enumerated strings.

• We don't explicitly include the coordinate flavor (stc:CoordFlavor).

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.

5.1 Frame

The STC Frame object remains the base for the Coordinate Frame hierarchy. It corresponds to the stc::coordFrameType of the STC model. By extension of the STCBase class, this object has the following relevant attributes.

5.1.1 Frame.name:string

A free form string 'label' for the frame.

5.1.2 Frame.ID:string

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

5.1.3 Frame.UCD:string

UCD string providing contextual meaning to the frame.

5.2 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.

5.2.1 CoordFrame.referencePosition:stdRefPosition

The origin of the coordinate system (and thus also its rest frame). The value of this field is restricted to the enumerated set provided by the stdRefPosition type.

5.3 RedshiftFrame

Extension of the Frame 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.

5.3.1 RedshiftFrame.referencePosition:stdRefPosition

The origin of the coordinate system (and thus also its rest frame). The value of this field is restricted to the enumerated set provided by the stdRefPosition type.

5.3.2 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.

5.4 SpaceFrame

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

5.4.1 SpaceFrame.name:string

This field is restricted to the stdSpaceRefFrame enumeration list shown in Section 6.4.3.2. NOTE: This model does not support the use of CUSTOM or GEO_D as it does not accommodate their parameters.

5.4.2 SpaceFrame.referencePosition:stdRefPosition

The origin of the coordinate system (and thus also its rest frame). The value of this field is restricted to the enumerated set provided by the stdRefPosition type.

5.4.3 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.

5.5 SpectralFrame

Extension of the Frame 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.

5.5.1 SpectralFrame.name:string

The 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’.

5.5.2 SpectralFrame.referencePosition:stdRefPosition

The origin of the coordinate system (and thus also its rest frame). The value of this field is restricted to the enumerated set provided by the stdRefPosition type.

5.5.3 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.]

5.6 TimeFrame

Extension of the Frame object used to provide a Time frame.

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

5.6.1 TimeFrame.name:string

This field is restricted to the TimeScale enumeration list, Section 6.4.3.3.

5.6.2 TimeFrame.referencePosition:stdRefPosition

The origin of the coordinate system (and thus also its rest frame). The value of this field is restricted to the enumerated set provided by the stdRefPosition type.

5.6.3 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 does not support dates before AD 1, so cannot express this reference time. Therefore, we feel that 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 format compliant with the Date datatype format (see Section 6.4.1.1, e.g. ”2004-11-30T11:59:00.01”.

6 IVOA Conventions

6.1 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.

6.1.1 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).

6.1.2 Model Reuse

This model incorporates elements from the Characterisation 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”.

6.1.3 UTypes

Characterisation: The labels (FluxAxis, TimeAxis, etc.) are used to identify the specialized realizations of CharacterisationAxis. These labels are used in place of the CharacterisationAxis portion of the relevant UType, and express the role of the axis with respect to the Characterisation class.

6.2 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-Selgado 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).

6.3 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.

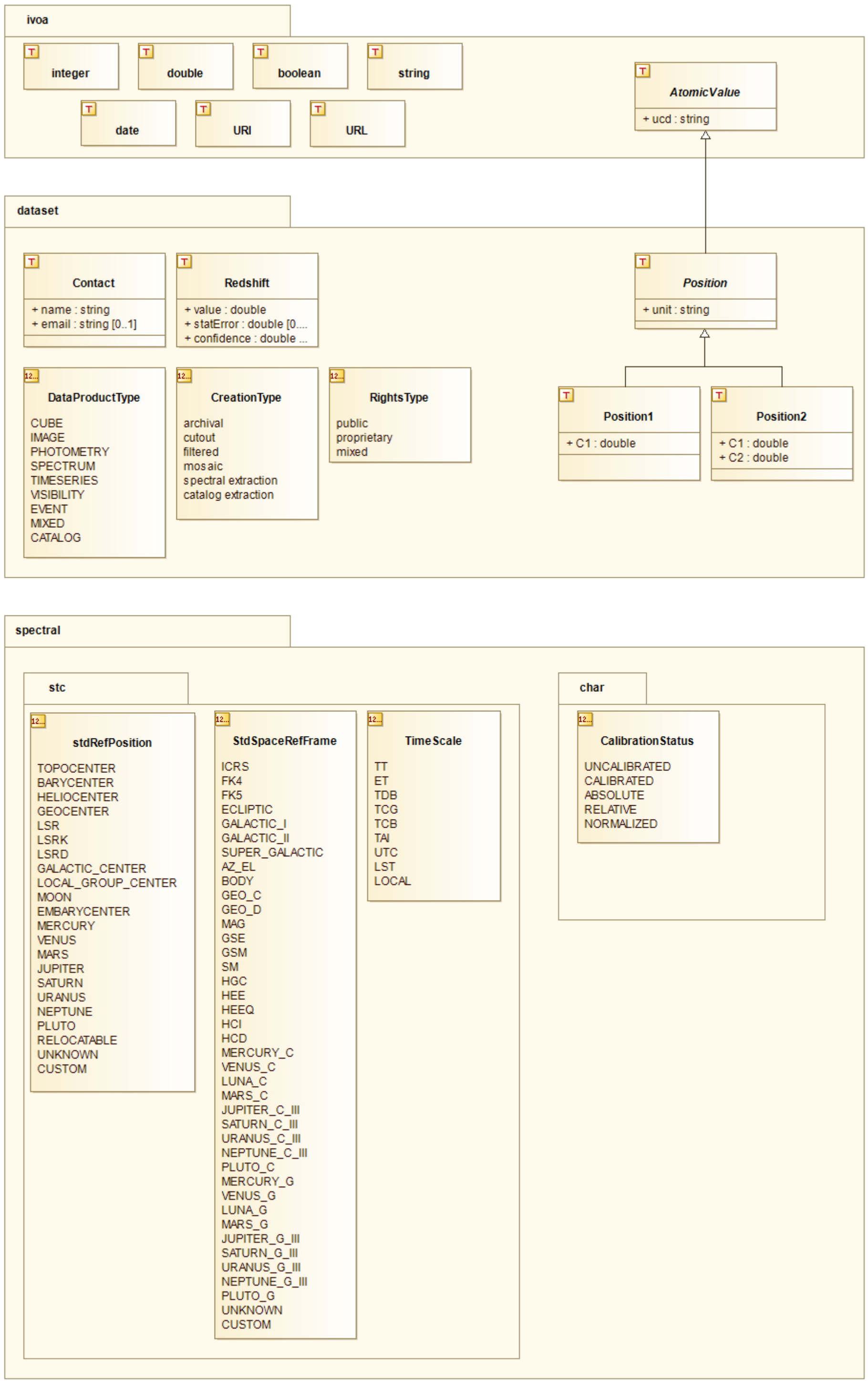
Appendix A includes UCD values for most model elements. For each axis, there are several fields whose UCDs depend on that of the axis itself. For example, the FluxAxis value and the statistical error on that value. Since there are multiple options for the axis UCD, we use the placeholder notation "[Axis.ucd]'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

6.4 Datatypes



This Section describes the datatypes used in this document.

6.4.1 ivoa

The ivoa set of datatypes include primitive types and other generic types applicable for a broad range of objects. Content is inspired by and consistent with the VO-DML ivoa datatypes model.

6.4.1.1 ivoa.date

This datatype is for expressing date-time values. The value is nominally expressed as a string following the FITS standard for the "DATE" keyword. NOTE: The FITS standard is effectively ISO8601 format without the "Z" tag to indicate UTC. Values are nominally expressed in UTC.

6.4.1.2 ivoa.AtomicValue

This is an abstract base class for a range of Value based objects.

6.4.2 Observation - Dataset types

Additional datatypes defined for Dataset model objects.

6.4.2.1 DataProductType

Enumeration identifying the type of data product. Allowed values are:

Token	**Meaning**
CUBE	A multidimensional astronomical image of three or more axes.
IMAGE	A two dimensional astronomical image
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.
VISIBILITY	A visibility (radio) dataset of some sort.
EVENT	An event counting dataset of some sort (e.g. X-ray)
MIXED	Some combination of the other options.
CATALOG	A catalog.

6.4.2.2 CreationType

Enumeration of dataset creation types. Allowed values are:

Token	Meaning
archival	Indicates that it is one of a collection of datasets generated in a systematic, homogeneous way and is 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.

6.4.2.3 RightsType

Enumeration indicating property rights levels. Allowed values are:

Token	Meaning
public	
proprietary	
mixed	

6.4.2.4 Contact

Datatype providing contact information for an individual or entity

+ **name**: string - name of individual or entity

+ **email**: string - e-mail address for individual or entity

6.4.2.5 Redshift

Measured redshift.

+ **value**: double - defined as $\Delta\lambda/\lambda$ and may be positive or negative

+ **statError**: double - error on measured redshift

+ **confidence**: double - confidence is an additional measure of accuracy.

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.

6.4.2.6 Position

Extension of the ivoa:AtomicValue type, this abstract type provides a basis for N-dimensional Position types whose values may be obtained via component attributes Cn.

+ **unit**: string - unit in which all Cn position values are given

6.4.2.7 Position1

Extension of Position for one dimensional positions.

+ **C1**: double - position value

6.4.2.8 Position2

Extension of Position for two dimensional positions.

+ **C1**: double - position value in first dimension.

+ **C2**: double - position value in second dimension.

6.4.3 Spectral Model Types

6.4.3.1 stdRefPosition

Enumeration of Standard reference positions.

Token	Meaning	Note
TOPOCENTER	Location of the observing device	(telescope)
BARYCENTER	Solar system barycenter	
HELIOCENTER	Center of the Sun	
GEOCENTER	Center of the Earth	
LSR	Local Standard of Rest	
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	
MOON	Center of the Moon	
EMBARYCENTER	Earth-Moon barycenter	
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	

RELOCATABLE	Relative origin	Suitable for simulations
UNKNOWN	Unknown origin	
CUSTOM	Origin specified wrt another system	

6.4.3.2 stdSpaceRefFrame

Enumeration of standard Space Reference Frames.

Token	Meaning	Parameter(s)
ICRS	The ICRS frame	
FK4	FK4	Equinox
FK5	FK5	Equinox
ECLIPTIC	Ecliptic l,b	Equinox
GALACTIC_I	Old galactic LI, BI	
GALACTIC_II	Galactic LII, BII	
SUPER_GALACTIC	SGL, SGB	
AZ_EL	Azimuth and elevation	
BODY	Generic body (eg. planet)	
GEO_C	Geocentric corotating	
GEO_D	Geodetic ref frame	Spheroid
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	
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	
UNKNOWN	Unknown frame	
CUSTOM	Custom frame	Pole, axis

6.4.3.3 TimeScale

Enumeration of time scales.

Token	**Meaning**
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

6.4.3.4 CalibrationStatus

Enumeration of calibration levels.

Token	Meaning
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.
CALIBRATED	indicates that values are expected to be correct within the given uncertainty.
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.

6.5 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.

6.5.1 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.

6.5.2 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.

6.5.3 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.

6.5.4 Support

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

• Retain existance of extended content, including namespace and UTypes.

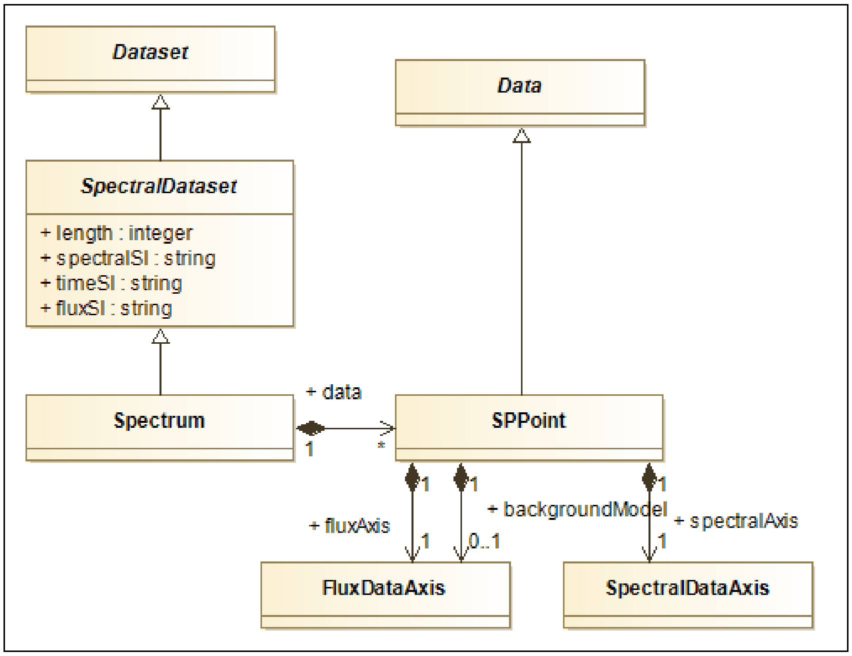
• Retain association with modeled component.

• Provide access to extended content by users.

6.5.5 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.

7 Spectrum Data Model



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 6.1.1.

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 Reqirement**
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 minimial set of units which users should expect to be handled by applications.

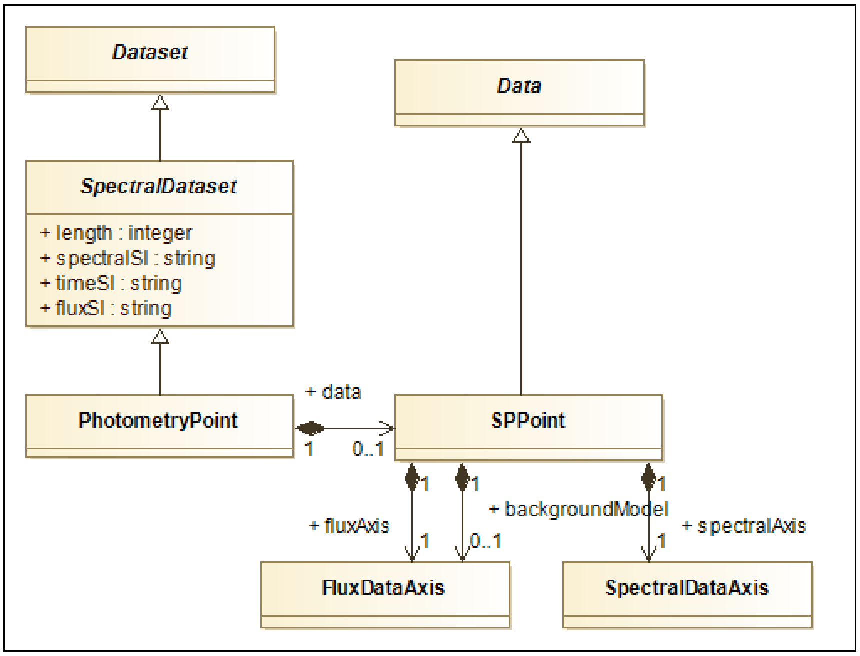
7.1.5 Required Fields

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



8.1 PhotometryPoint

An extension of SpectralDataset 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 6.1.1.

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 Reqirement**
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 node 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 minimial 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 Dataset.DataModel.Prefix Dataset.Type	MAN OPT OPT	"PhotometryPoint-1.0" "phot" "PhotometryPoint"
Target.Name	MAN	
Char.SpectralAxis.ucd Char.SpectralAxis.unit	MAN MAN	
CoordSys.FluxFrame.PhotCal.ZeroPoint.Flux.Value CoordSys.FluxFrame.PhotCal.ZeroPoint.Flux.Unit	MAN MAN	
Data.SpectralAxis.Value	MAN	

8.1.6 Restrictions

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

Appendix 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.CalibrationStatus	Enum String	meta.code.qual	Type of coord calibration	OPT	CALIBRATED
Char.FluxAxis.Accuracy			node		
Char.FluxAxis.Accuracy.StatError	Double	stat.error;[FluxAxis.ucd]	Statistical 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.CodeNum	Integer		Quality code numeric value	OPT	(None)
Char.FluxAxis.QualityDefs.QualityCode.Definition	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.CalibrationStatus	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)

Spectral Data Model UType List					
Field	**Datatype**	**UCD1+**	**Meaning**	**Req**	**Default**
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.CodeNum	Integer		Quality code numeric value	OPT	(None)
Char.SpatialAxis.QualityDefs.QualityCode.Definition	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
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.CalibSatus	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
Char.SpectralAxis.Accuracy.SysError	Double	stat.error.sys;[SpectralAxis.ucd]	Spectral coord measurement error	REC	0
Char.SpectralAxis.CoordSys			node		
Char.SpectralAxis.CoordSys.*			Axis coordinate system	OPT	
Char.SpectralAxis.Coverage			node		
Char.SpectralAxis.Coverage.Location			node		
Char.SpectralAxis.Coverage.Location.Value	Double	[SpectralAxis.ucd];instr.bandpass	Spectral coord value	MAN	(N/A)
Char.SpectralAxis.Coverage.Bounds			node		
Char.SpectralAxis.Coverage.Bounds.Extent	Double	instr.bandwidth	Width in spectral coordinate	MAN	(N/A)
Char.SpectralAxis.Coverage.Bounds.Start	Double	[SpectralAxis.ucd];stat.min	Minimum on spectral coordinate	MAN	(N/A)
Char.SpectralAxis.Coverage.Bounds.Stop	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			node		
Char.SpectralAxis.QualityDefs.QualityCode.CodeNum	Integer		Quality code numeric value	OPT	(None)
Char.SpectralAxis.QualityDefs.QualityCode.Definition	String		Meaning of quality value	OPT	(None)

Spectral Data Model UType List					
Field	**Datatype**	**UCD1+**	**Meaning**	**Req**	**Default**
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
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.CalibrationStatus	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		
Char.TimeAxis.Coverage.Location.Value	Double	time.epoch;obs	Exposure midpoint	MAN	(N/A)
Char.TimeAxis.Coverage.Bounds			node		
Char.TimeAxis.Coverage.Bounds.Extent	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			node		
Char.TimeAxis.QualityDefs.QualityCode.CodeNum	Integer		Quality code numeric value	OPT	(None)
Char.TimeAxis.QualityDefs.QualityCode.Definition	String		Meaning of quality value	OPT	(None)
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		

Spectral Data Model UType List					
Field	**Datatype**	**UCD1+**	**Meaning**	**Req**	**Default**
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)
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
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

Spectral Data Model UType List					
Field	**Datatype**	**UCD1+**	**Meaning**	**Req**	**Default**
CoordSys.SpectralFrame.RefPos	String	sdm:spect.frame	Spectral frame origin	OPT	TOPOCENTER
CoordSys.SpectralFrame.Redshift	Double		Spectral frame shift	OPT	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
Curation			node		
Curation.Publisher	String	meta.curation	Publisher	MAN	(N/A)
Curation.PublisherID	URI	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	String	meta.ref.url;meta.curation	Publisher specified dataset ID	REC	DataID.DatasetID
Data			node		
Data.*Axis.Correction.GenericCorr			node		
Data.*Axis.Correction.GenericCorr.Name	String		Correction name	OPT	(None)
Data.*Axis.Correction.GenericCorr.Value	Double		Correction value	OPT	(None)
Data.*Axis.Correction.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		
Data.FluxAxis.Value		[see Appendix B]	Flux values for points	MAN	(N/A)
Data.FluxAxis.ucd	String	meta.ucd	ucd for flux	OPT	Char.FluxAxis.ucd
Data.FluxAxis.unit	String	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

Spectral Data Model UType List					
Field	**Datatype**	**UCD1+**	**Meaning**	**Req**	**Default**
Data.FluxAxis.Accuracy.SysError	Double	stat.error.sys;[FluxAxis.ucd]	Systematic error	OPT	Char.FluxAxis...
Data.FluxAxis.Corrections			node		
Data.FluxAxis.Correction.ApFrac			node		
Data.FluxAxis.Correction.ApFrac.Value	Double	arith.ratio	Aperture fraction	OPT	1
Data.FluxAxis.Correction.ApFrac.Applied	Boolean	meta.code	Aperture fraction applied	OPT	FALSE
Data Spectral Axis					
Data.SpectralAxis			node		
Data.SpectralAxis.Value		[see Appendix B]	Spectral coordinates for points	REC	NaN
Data.SpectralAxis.ucd	String	meta.ucd	ucd for spectral coord	OPT	Char.SpectralAxis.ucd
Data.SpectralAxis.unit	String	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	String	meta.ucd	ucd for time	OPT	Char.TimeAxis.ucd
Data.TimeAxis.unit	String	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		

Spectral Data Model UType List					
Field	**Datatype**	**UCD1+**	**Meaning**	**Req**	**Default**
Data.BackgroundModel.Value			Flux values for points	OPT	(None)
Data.BackgroundModel.ucd	String	meta.ucd	ucd for background flux	OPT	Char.FluxAxis.ucd
Data.BackgroundModel.unit	String	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		
DataID.Title	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)
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.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
Dataset			node		
Dataset.Type	Enum String		Dataset or segment type	OPT	Spectral
Dataset.DataProductSubType	String	meta.id	Dataset subtype	OPT	UNKNOWN
Dataset.DataModel.Name	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	src.redshift	Measured redshift for spectrum	OPT	NaN
Derived.Redshift.StatError	Double	stat.error;src.redshift	Error on measured redshift	OPT	NaN

Spectral Data Model UType List					
Field	**Datatype**	**UCD1+**	**Meaning**	**Req**	**Default**
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
ObsConfig.Facility.Name	String	meta.id;instr.tel	Facility name	OPT	UNKNOWN
ObsConfig.Instrument.Name	String	meta.id;instr	Instrument ID	OPT	UNKNOWN
ObsConfig.Bandpass.Name	String	instr.bandpass	Band	OPT	UNKNOWN
ObsConfig.DataSource.Name	String		Original data type	OPT	UNKNOWN
Proposal.Identifier	String	meta.id;obs.proposal	Proposal ID	OPT	UNKNOWN
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	Position2	pos.eq;src	Target RA and Dec	REC	NaN
Target.VarAmpl	Double	src.var.amplitude	" Target variability amplitude	typical "	OPT

Appendix 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 (vF(v))	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)

Appendix 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 readablility. 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 is not a comprehensive serialization of the model, but includes all required fields, and an example of the various datatypes (string, double, array, enum, uri, url), as well as complex attribute (Derived.Redshift) and element with multiplicity greater than one (DataID.Collection).

```
<?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 name="Curation">
                <PARAM name="Publisher"
                      datatype="char" ucd="meta.curation" utype="spec:Curation.Publisher" value="Database" arraysize="*">
                      <DESCRIPTION>Dataset publisher</DESCRIPTION>
                </PARAM>
                <PARAM name="Rights"
                      datatype="char" ucd="meta.code" utype="spec:Curation.Rights" value="public" arraysize="*">
                      <DESCRIPTION>Proprietary restrictions level</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>
            <PARAM name="Collection1"
                datatype="char" ucd="" utype="spec:DataID.Collection" value="Chandra" arraysize="*"
                <DESCRIPTION>Collection name</DESCRIPTION>
            </PARAM>
            <PARAM name="Collection2"
                datatype="char" ucd="" utype="spec:DataID.Collection" value="X-Ray" arraysize="*"
                <DESCRIPTION>Collection name</DESCRIPTION>
            </PARAM>
            <PARAM name="Collection3"
                datatype="char" ucd="" utype="spec:DataID.Collection"
                value="Third Cambridge Catalogue of Radio Sources" arraysize="*"
                <DESCRIPTION>Collection name</DESCRIPTION>
            </PARAM>
            <PARAM name="DatasetID"
                datatype="char" ucd="meta.id;meta.dataset" utype="spec:DataID.DatasetID"
                value="ivo://ADS/Sa.CXO#obs/12345" arraysize="*"
                <DESCRIPTION>IVOA Dataset ID</DESCRIPTION>
            </PARAM>
            <PARAM name="CreationDate"
                datatype="char" ucd="time.epoch;meta.dataset" utype="spec:DataID.Date"
                value="2001-12-08T03:11:11" arraysize="*"
                <DESCRIPTION>Data processing/creation date</DESCRIPTION>
            </PARAM>
            <PARAM name="Logo"
                datatype="char" ucd="meta.ref.url" utype="spec:DataID.Logo"
                value="http://www.cfa.harvard.edu/common/images/left/cfa-logo.gif" arraysize="*"
                <DESCRIPTION>URL for creator logo</DESCRIPTION>
            </PARAM>
        </GROUP>

        <GROUP name="Derived">
            <GROUP name="Redshift">
                <PARAM name="MeasuredRedshift"
                    datatype="double" ucd="src.redshift" utype="spec:Derived.Redshift.Value" value="0.159" >
                    <DESCRIPTION>Measured redshift for spectrum</DESCRIPTION>
                </PARAM>
                <PARAM name="MeasuredRedshiftError"
                    datatype="double" ucd="stat.error;src.redshift" utype="spec:Derived.Redshift.StatError" value="0.002" >
                    <DESCRIPTION>Error on measured redshift</DESCRIPTION>
                </PARAM>
                <PARAM name="MeasuredRedshiftConfidence"
                    datatype="double" ucd="" utype="spec:Derived.Redshift.Confidence" value="1.0" >
                    <DESCRIPTION>Confidence value on redshift</DESCRIPTION>
                </PARAM>
            </GROUP>
        </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="Characterisation">
            <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"  utype="spec: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"  utype="spec: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"  utype="spec: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>
</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. See the spectrum serialization for examples of other data and object types.

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="Characterisation">
    <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"  utype="phot: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"  utype="phot: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"  utype="phot: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 standard CoordFrame reference position:

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 consistancy, 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, implementers should check both keywords and columns for appropriate tokens.

Data-Char Overlap: Section 3.9.1 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 1Sn_1 and 1CTYPn 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 TUTYPn 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, TTYPE5=’ERR_LO’ and TUTYP5=’Data.FluxAxis.Accuracy.StatErrLow’; the WCS keyword 1CTYPE5=’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 1S5_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 metatdata 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 (TTYPEn, 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.CalibrationStatus	FLUX_CAL
Char.FluxAxis.Accuracy.StatError	STAT_ERR
Char.FluxAxis.Accuracy.SysError	SYS_ERR
Char.FluxAxis.QualityDefs.QualityCode.CodeNum	QUALn
Char.FluxAxis.QualityDefs.QualityCode.Definition	QUALn value
Char.SpatialAxis.Name	SKY_NAM
Char.SpatialAxis.ucd	SKY_UCD
Char.SpatialAxis.unit	SKY_UNI
Char.SpatialAxis.CalibrationStatus	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.CodeNum	
Char.SpatialAxis.QualityDefs.QualityCode.Definition	
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.CalibrationStatus	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.CodeNum	
Char.SpectralAxis.QualityDefs.QualityCode.Definition	
Char.SpectralAxis.Resolution.refVal	SPEC_RES
Char.SpectralAxis.ResolPower.refVal	SPEC_RP
Char.SpectralAxis.SamplingPrecision.SampleExtent	SPEC_BIN

FITS keyword mapping for Spectral Data Model	
Spectral Model UType	**FITS Keyword/Column definition**
Char.SpectralAxis.SamplingPrecision.SamplingPrecisionRefVal.FillFactor	SPEC_FIL
Char.TimeAxis.Name	TIME_NAM
Char.TimeAxis.ucd	TIME_UCD
Char.TimeAxis.unit	TIMEUNIT
Char.TimeAxis.CalibrationStatus	TIME_CAL
Char.TimeAxis.Accuracy.BinSize	TIMEDEL
Char.TimeAxis.Accuracy.StatError	TIME_ERR
Char.TimeAxis.Accuracy.SysError	TIME_SYE
Char.TimeAxis.Coverage.Location.Value	TMID
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.CodeNum	
Char.TimeAxis.QualityDefs.QualityCode.Definition	
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

FITS keyword mapping for Spectral Data Model	
Spectral Model UType	**FITS Keyword/Column definition**
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	CRETYPE
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
Dataset.FluxSI	FLUXSDIM
Derived.SNR	DER_SNR
Derived.Redshift.Value	DER_Z
Derived.Redshift.StatError	DER_ZERR
Derived.Redshift.Confidence	DER_ZCNF

FITS keyword mapping for Spectral Data Model	
Spectral Model UType	**FITS Keyword/Column definition**
Derived.VarAmpl	DER_VAR
Target.Name	OBJECT
Target.Description	OBJDESC
Target.Class	SRCCLASS
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.Correction.ApFrac.Value	
Data.FluxAxis.Correction.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’

FITS keyword mapping for Spectral Data Model	
Spectral Model UType	**FITS Keyword/Column definition**
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.Correction.GenericCorr.Name	
Data.*Axis.Correction.GenericCorr.Value	
Data.*Axis.Correction.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/newsletter42.html

[7] ”DER SNR: A simple and general spectroscopic signal-to-noise measurement algorithm”;
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 93001.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

Serialization Issues

This model contains elements which are complex objects with multiplicity greater than one. Serialization conventions to date, do not provide a means of unambiguously associating the various elements of one instance versus another. In this model, this issue effects several components, and becomes a bigger issue with multiband photometry and time series.

– multiple FluxAxis within Data or Characterisation;

– multiple FluxFrame within CoordSys;

– multiple QualityCode within QualityDefs;

• For VOTable serializations, we recommend the use of the GROUP element to provide the necessary structure for each instance.

* For FITS serializations, it is simply not possible. 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.