*I*nternational

*V*irtual

*O*bservatory

*A*lliance

Space-Time Coordinate (STC) Metadata Model

Version 1.31

IVOA Note 5 December 2007

This version:
http://www.ivoa.net/Documents/Notes/STC/STC-Model-20071205.html
Latest version:
http://www.ivoa.net/Documents/latest/STC-Model.html
Previous version(s):
http://www.ivoa.net/Documents/Notes/STC/STC-Model-20071030.html

Author(s):
A. H. Rots

Abstract

This document provides a summary of the model for the description of the VO Space-Time Coordinate metadata. It shows the hierarchical relation between the various components with brief descriptions where needed. The Tables that are referred to may be found in the STC document. XML element names are shown in **`bold Courier`** font.
The objective is to provide a framework that allows a complete and internally consistent specification of coordinate metadata – primarily the intertwined temporal, spatial, spectral, and redshift coordinates – that is extensible to accommodate all future applications.

Status of This Document

This is a Note. The first release of this document was 2007-12-04.

This is an IVOA Note expressing suggestions from and opinions of the authors. It is intended to share best practices, possible approaches, or other perspectives on interoperability with the Virtual Observatory. It should not be referenced or otherwise interpreted as a standard specification.

A list of current IVOA Recommendations and other technical documents can be found at http://www.ivoa.net/Documents/.

Table of Contents

1 STC Metadata Description

An STC metadata description element consists of three components:

1. Coordinate System
2. Coordinates
3. Coordinate area

Missing components or subcomponents shall be considered UNKNOWN; this is to be interpreted as: it is up to the client to decide whether or not to accept the metadata, and to assign a sensible default value if the component is relevant.

1.1 Coordinate System (`CoordSys`, `AstroCoordSystem`, `PixelCoordSystem`)

A Coordinate System consists of one or more Coordinate Frames.

A Coordinate Frame typically consists of a Reference Frame (orientation) and a Reference Position (or Origin), but some Frames contain more, or less, information. There are six kinds of Frames.

The three kinds of Coordinate System are:

1. (Generic) **`CoordSys`**; contains:
   a. One or more Generic Coordinate Frames
2. **`AstroCoordSystem`**; *may* contain:
   a. One or more Generic Coordinate Frames
   b. One Time Frame
   c. One Space Frame
   d. One Spectral Frame
   e. One Redshift Frame
3. **`PixelCoordSystem`**; *may* contain:
   a. One or more Generic Coordinate Frames
   b. One or more Pixel Coordinate Frames

1.1.1 Generic Coordinate Frame (`CoordFrame`)

Allows specifying metadata for non-STC coordinates.

A **`CoordFrame`** contains a Coordinate Flavor and *may* contain a Coordinate Reference Frame and/or a Coordinate Reference Position; it requires a **`frame_id`**.

1.1.1.1 Reference Position (`CoordRefPos`)

A CoordRefPos consists of a Coordinate.

1.1.1.2 Generic Coordinate Reference Frame (`CoordRefFrame`)

There are four types of Coordinate Reference Frame:

(All Transformation elements contain a **`projection`** attribute, using values from Table 4)

1.1.1.2.1 `ScalarRefFrame`

Contains:

1. **`Scale`**

1.1.1.2.2 `Cart2DRefFrame`

Contains one of:

1. **`Transform2`**
2. **`Transform2Matrix`**

1.1.1.2.3 `Cart3DRefFrame`

Contains one of:

1. **`Transform3`**
2. **`Transform3Matrix`**

1.1.1.2.4 `SphericalRefFrame`

Contains:

1. **`Frame`**
2. **`Pole_Zaxis`**
3. **`Xaxis`**

1.1.1.3 Coordinate Flavor (`CoordFlavor`)

`CARTESIAN`, **`SPHERICAL`**, **`UNITSPHERE`**, **`POLAR`**, **`CYLINDRICAL`**, **`HEALPIX`**, or **`STRING`**; contains: dimensionality.

1.1.2 Time Frame (`TimeFrame`)

A **`TimeFrame`** contains a Reference Position and a Time Scale, and *may* contain a Time Reference Direction.

1.1.2.1 Reference Position (`ReferencePosition`)

A Reference Position may be a Standard or a Custom one:

1.1.2.1.1 Standard Reference Position

`TOPOCENTER`, **`GEOCENTER`**, **`BARYCENTER`**, etc., as in Table 1, noting the exceptions.
All Solar System positions, except **`GEOCENTER`**, should include a Planetary Ephemeris.

1.1.2.1.1.1 Planetary Ephemeris (`PlanetaryEphem`; conditional)

JPL-DE200 or JPL-DE405; needed when such am ephemeris is used to transform times.

1.1.2.1.2 Custom Reference Position (`CoordRefPos`)

A Coordinate Reference Position consists of a **`Coordinate`**.

1.1.2.2 Time Scale

`TT` (**`TDT`**), **`ET`**, **`TAI`** (**`IAT`**), **`UTC`**, **`GPS`**, **`TDB`** (**`TEB`**), **`TCG`**, **`TCB`**, **`LST`**, **`LOCAL`** (Table 2).

1.1.2.3 Time Reference Direction (`TimeRefDirection`; conditional)

If Time Reference Position is not **`TOPOCENTER`**, this direction (a **`Coordinate`** element) needs to be provided.

1.1.3 Space Frame (`SpaceFrame`)

A **`SpaceFrame`** includes a Reference Position, Spatial Reference Frame, and Coordinate Flavor, and *may* include an Offset Center.

1.1.3.1 Reference Position (`ReferencePosition`)

A Reference Position may be a Standard or a Custom one:

1.1.3.1.1 Standard Reference Position

`TOPOCENTER`, **`GEOCENTER`**, **`BARYCENTER`**, etc., as in Table 1, noting the exceptions (but including **`RELOCATABLE`**).
All Solar System positions, except **`GEOCENTER`**, should include a Planetary Ephemeris.

1.1.3.1.1.1 Planetary Ephemeris (`PlanetaryEphem`; conditional)

JPL-DE200 or JPL-DE405; needed when such an ephemeris is used to transform times.

1.1.3.1.2 Custom Reference Position (`CoordRefPos`)

A Coordinate Reference Position consists of a **`Coordinate`**.

1.1.3.2 Space Reference Frame

A Space Reference Frame may be a Standard or a Custom one:

1.1.3.2.1 Standard Reference Frame

`FK4`, **`FK5`**, **`ICRS`**, **`ECLIPTIC`**, **`GALACTIC`**, etc., as in Table 3, including **`UNKNOWNFrame`**.

1.1.3.2.1.1 `Equinox` (conditional)

`B`*`nnnn`* or **`J`*`nnnn`***, for **`FK`*`i`*** only

1.1.3.2.2 (Custom) Coordinate Reference Frame (`CoordRefFrame`)

There are four types of Coordinate Reference Frame:

(All Transformation elements contain a **`projection`** attribute, using values from Table 4)

1.1.3.2.2.1 `ScalarRefFrame`

Contains:

1. **`Scale`**

1.1.3.2.2.2 `Cart2DRefFrame`

Contains one of:

1. **`Transform2`**
2. **`Transform2Matrix`**

1.1.3.2.2.3 `Cart3DRefFrame`

Contains one of:

1. **`Transform3`**
2. **`Transform3Matrix`**

1.1.3.2.2.4 `SphericalRefFrame`

Contains:

1. **`Frame`**
2. **`Pole_Zaxis`**
3. **`Xaxis`**

1.1.3.3 Coordinate Flavor (`CoordFlavor`)

`CARTESIAN`, **`SPHERICAL`**, **`UNITSPHERE`**, **`POLAR`**, **`CYLINDRICAL`**, or **`HEALPIX`**; contains: dimensionality.

1.1.3.4 Offset Center (`OffsetCenter`; optional)

For offset coordinates; consists of a **`Coordinate`**.

1.1.4 Spectral Frame (SpectralFrame)

A SpectralFrame only contains a Reference Position

1.1.4.1 Reference Position

Note that the Spectral Reference Position requires position as well as a velocity vector. A Reference Position may be a Standard or a Custom one:

1.1.4.1.1 Standard Reference Position

`TOPOCENTER`, **`GEOCENTER`**, **`BARYCENTER`**, etc., as in Table 1, noting the exceptions (but including **`LSRx`**).
All Solar System positions, except **`GEOCENTER`**, should include a Planetary Ephemeris.

1.1.4.1.1.1 Planetary Ephemeris (`PlanetaryEphem`; conditional)

JPL-DE200 or JPL-DE405; needed when such am ephemeris is used to transform times.

1.1.4.1.2 Custom Reference Position (`CoordRefPos`)

A Coordinate Reference Position consists of a **`Coordinate`**.

1.1.5 Redshift Frame (RedshiftFrame)

A RedshiftFrame contains a Reference Position and a Doppler Definition.
It also contains an attribute that indicates whether the coordinate values represent redshifts or Doppler velocities.

1.1.5.1 Reference Position

Note that the Redshift Reference Position also requires position as well as a velocity vector. A Reference Position may be a Standard or a Custom one:

1.1.5.1.1 Standard Reference Position

TOPOCENTER, **GEOCENTER**, **BARYCENTER**, etc., as in Table 1, noting the exceptions (but including **LSR*x***).
All Solar System positions, except **GEOCENTER**, should include a Planetary Ephemeris.

1.1.5.1.1.1 Planetary Ephemeris (PlanetaryEphem; conditional)

JPL-DE200 or JPL-DE405; needed when such am ephemeris is used to transform times.

1.1.5.1.2 Custom Reference Position (CoordRefPos)

A Coordinate Reference Position consists of a **Coordinate**.

1.1.5.2 Doppler Definition (DopplerDefinition)

Allowed values: **OPTICAL**, **RADIO**, **RELATIVISTIC**.

1.1.6 Pixel Frame (`PixelFrame`)

A **`PixelFrame`** contains a Coordinate Flavor and *may* contain a Coordinate Reference Frame, a Coordinate Reference Position, and/or a Reference Pixel. It contains one or more **`axis_order`** attributes.

1.1.6.1 Reference Position (`CoordRefPos`)

A **`CoordRefPos`** consists of a **`Coordinate`**.

1.1.6.2 Coordinate Reference Frame (`CoordRefFrame`)

There are four types of Coordinate Reference Frame:

(All Transformation elements contain a **`projection`** attribute, using values from Table 4)

1.1.6.2.1 `ScalarRefFrame`

Contains:

1. **`Scale`**

1.1.6.2.2 `Cart2DRefFrame`

Contains one of:

1. **`Transform2`**
2. **`Transform2Matrix`**

1.1.6.2.3 `Cart3DRefFrame`

Contains one of:

1. **`Transform3`**
2. **`Transform3Matrix`**

1.1.6.2.4 `SphericalRefFrame`

Contains:

1. **`Frame`**
2. **`Pole_Zaxis`**
3. **`Xaxis`**

1.1.6.3 Coordinate Flavor (`CoordFlavor`)

`CARTESIAN`, **`SPHERICAL`**, **`UNITSPHERE`**, **`POLAR`**, **`CYLINDRICAL`**, **`HEALPIX`**, or **`STRING`**; contains: dimensionality.

1.1.6.4 Reference Pixel (`ReferencePixel`)

`ReferencePixel` contains a 1, 2, or 3-dimensional **`Pixel`**.

1.2 Coordinates (Coords, AstroCoords, PixelCoords)

A Coordinates element consists of up to six **Coordinate** elements and contains a reference to a Coordinate System. Each **Coordinate** has five components. Each component *may* occur once (a definite or typical value) or twice (indicating a range of values):

1. Value
2. Error
3. Resolution
4. Size
5. Pixel size

All components are optional; in addition, there *may* be a name. One or more **Coordinate** elements *may* be provided through a binary FITS table.

A variety of error types may be defined; statements below on the data types of errors are to be considered preliminary, contingent on additional definitions.

There are nine types of **Coordinate** elements, each of which is a composite of the above-mentioned components, except where noted.

Each Coordinates object needs to refer to a Coordinate System through a **coord_system_id** attribute.

There are three types of Coordinates elements:

1. (Generic) **Coords**; contains one or more Generic Coordinates and refers to a **CoordSys**; individual **Coordinate**s need to refer to a specific Coordinate Frame through a **frame_id**
2. **AstroCoords**; refers to an **AstroCoordSystem** and *may* contain:
    a. One or more Generic Coordinates; individual **Coordinate**s need to refer to a specific Coordinate Frame through a **frame_id**
    b. One Time Coordinate
    c. One Spatial Position Coordinate
    d. One Spatial Velocity Coordinate
    e. One Orbit Coordinate
    f. One Spectral Coordinate
    g. One Redshift Coordinate
    h. One Coordinate File
3. **PixelCoords**; refers to an **PixelCoordSystem** and *may* contain one or more:
    a. Generic Coordinate
    b. Pixel Coordinate

    Individual Coordinates need to refer to a specific Coordinate Frame through a **frame_id**.

1.2.1 Generic Coordinate (`GenCoordinate`)

The **`Coordinate`** needs to refer to a specific Coordinate Frame through a **`frame_id`**. There are four types of **`GenCoordinate`**:

1.2.1.1 `ScalarCoordinate`

Optional units; all components are scalar doubles.

1.2.1.2 `Vector2DCoordinate`

Optional units; **`Error`**, **`Resolution`**, **`Size`**, **`PixSize`** may consist of:

1. Two doubles
2. 2x2 matrix
3. Radius

1.2.1.3 `Vector3DCoordinate`

Optional units; **`Error`**, **`Resolution`**, **`Size`**, **`PixSize`** may consist of:

1. Three doubles
2. 3x3 matrix
3. Radius

1.2.1.4 `StringCoordinate`

Only contains a (string) **`Value`**, no other components.

1.2.2 Time (`Time`)

The value (**`TimeInstant`**) is an instant of the **`AstronTime`** class; all other components (**`Error`**, **`Resolution`**, **`Size`**, **`PixSize`**) are doubles and need a unit (s, d, a or yr, cy). **`AstronTime`** contains an Absolute Time element and an optional relative time element. It also *may* have a **`TimeScale`** attribute. **`AstronTime`** contains the following:

1.2.2.1 Absolute Time (`AbsoluteTime`)

Absolute Time may be of type **`ISOTime`** (ISO-8601 format), **`JDTime`**, or **`MJDTime`**. This should be a decimal for JD and MJD since a double is not guaranteed to have sufficient accuracy. In addition, for use with simulations, it may be of the type **`TimeOrigin`** which currently may only have the value **`RELOCATABLE`**.

1.2.2.2 Time Offset (`TimeOffset`; optional)

Decimal; offset with respect to Absolute Time; unit required.

1.2.3 Spatial Position (`Position`)

Units are required for all components, angular or linear.
There are three types of **`Position`**:

1.2.3.1 `Position1D`

All components are scalar doubles.

1.2.3.2 `Position2D`

`CError2`, **`CResolution2`**, **`CSize2`**, **`CPixSize2`** may consist of:

1. Two doubles
2. 2x2 matrix
3. Radius

1.2.3.3 `Position3D`

`CError3`, **`CResolution3`**, **`CSize3`**, **`CPixSize3`** may consist of:

1. Three doubles
2. 3x3 matrix
3. Radius

1.2.4 Spatial Velocity (`Velocity`)

Same model as Spatial Position; needs a spatial as well as a time unit.
Note that this item is for true space velocities, not for derived Doppler velocities.
There are three types of **`Velocity`**:

1.2.4.1 `Velocity1D`

All components are scalar doubles.

1.2.4.2 `Velocity2D;`

`CError2`, **`CResolution2`**, **`CSize2`**, **`CPixSize2`** may consist of:

1. Two doubles
2. 2x2 matrix
3. Radius

1.2.4.3 `Velocity3D;`

`CError3`, **`CResolution3`**, **`CSize3`**, **`CPixSize3`** may consist of:

1. Three doubles
2. 3x3 matrix
3. Radius

1.2.5 Orbital Elements (`Orbit`; alternative)

Spatial **`Position`** and **`Velocity`** (values only, of course) may also be provided through orbital elements:

1.2.5.1 Semi-major axis (`a`; conditional)

Semi-major axis to be used for elliptical orbits (0 ≤ **`e`** < 1); for parabolic and hyperbolic orbits use periapsis distance **`q`**.

1.2.5.2 Periapsis Distance (`q;` conditional)

Required for open orbits (1 ≤ **`e`**); *may* be used *instead of* **`a`** for closed orbits.

1.2.5.3 Eccentricity (`e`)

1.2.5.4 Inclination (`i`)

1.2.5.5 Longitude of Ascending Node (`Node`)

1.2.5.6 Argument of Periapsis (`Aop`)

1.2.5.7 Mean Anomaly (`M;` optional)

Mean anomaly at time **`T`**; if absent, **`T`** will refer to pericenter.

1.2.5.8 Orbital Period (`P`; optional)

Redundant, but optional for closed orbits.

1.2.5.9 Epoch (`T`)

Epoch of mean anomaly, if **`M`** is present, or of periapsis if **`M`** is absent.

1.2.6 Spectral Coordinate (`Spectral`)

All components are scalar doubles; spectral unit required.

1.2.7 Redshift Coordinate (`Redshift`)

All components are scalar doubles; position and time unit required for Doppler velocities.

1.2.8 Coordinate File (`CoordFile`)

Astronomical coordinate components may be specified in FITS files; this element provides an unambiguous description and reference. It contains the following elements:

1.2.8.1 `FITSFile`

Provides a URI and, optionally, one or two attributes: **`hdu_num`** and **`hdu_name`**. The HDU that is referred to should be a FITS Binary (or ASCII) Table.

1.2.8.2 `FITSTime` (optional)

Contains the FITS TTYPE (name) of the column containing the data for each of the Time coordinate components (value, error, resolution, size, pixel size) present.

1.2.8.3 `FITSPosition` (optional)

Contains the FITS TTYPE (name) of the column containing the data for each of the Time coordinate components (value, error, resolution, size, pixel size) present. For vector coordinates **`FITSPosition`** contains the TTYPEs for the appropriate number of columns, separated by commas.

1.2.8.4 `FITSVelocity` (optional)

Contains the FITS TTYPE (name) of the column containing the data for each of the Time coordinate components (value, error, resolution, size, pixel size) present. For vector coordinates **`FITSVelocity`** contains the TTYPEs for the appropriate number of columns, separated by commas.

1.2.8.5 `FITSSpectral` (optional)

Contains the FITS TTYPE (name) of the column containing the data for each of the Time coordinate components (value, error, resolution, size, pixel size) present.

1.2.8.6 `FITSRedshift` (optional)

Contains the FITS TTYPE (name) of the column containing the data for each of the Time coordinate components (value, error, resolution, size, pixel size) present.

1.2.9 Pixel Coordinate (`Pixel`)

The **`Pixel`** needs to refer to a specific Coordinate Frame through a **`frame_id`**. The elements only contain a **`Value`** component and, optionally, a **`Name`**.
There are three types of **`Pixel`**s:

1.2.9.1 `Pixel1D`

One double.

1.2.9.2 `Pixel2D`

Two doubles.

1.2.9.3 `Pixel3D`

Three doubles.

1.3 Coordinate Area (`CoordArea`, `AstroCoordArea`, `PixelCoordArea`)

Specifies the coordinate volume occupied by the object that the metadata refer to. The Coordinate Area shall contain a reference to a Coordinate System. There is one Area object for each Coordinate Frame that shall indicate whether or not limits are included (**`lo_include`** and **`hi_include`**) and that contains an optional fill factor (**`fill_factor`**; default=1.0). The typical form is one or more intervals consisting of a lower limit, an upper limit, or both (i.e., open intervals are allowed). The selection for 2-dimensional spatial coordinates in particular is more elaborate.

Each Coordinate Area object needs to refer to a Coordinate System through a **`coord_system_id`** attribute.

There are three types of Coordinate Area elements:

1. (Generic) **`CoordArea`**; contains one or more Generic Coordinate Intervals and refers to a **`CoordSys`**; individual **`CoordInterval`**s need to refer to a specific Coordinate Frame through a **`frame_id`**
2. **`AstroCoordArea`**; refers to an **`AstroCoordSystem`** and *may* contain any number of:
   a. Generic Coordinate Intervals; individual **`CoordInterval`**s need to refer to a specific Coordinate Frame through a **`frame_id`**
   b. Time Intervals
   c. Spatial Position Intervals
   d. Spatial Velocity Intervals
   e. Spectral Intervals
   f. Redshift Intervals
3. **`PixelCoordArea`**; refers to an **`PixelCoordSystem`** and *may* contain any number of:
   a. Generic Coordinate Intervals
   b. Pixel Coordinate Intervals

   Individual Coordinate Intervals need to refer to a specific Coordinate Frame through a **`frame_id`**.

1.3.1 Generic Scalar Range (`CoordInterval`)

The **`CoordInterval`** needs to refer to a specific Coordinate Frame through a **`frame_id`**. There are three types of **`CoordInterval`**:

1.3.1.1 `CoordScalarInterval`

Optional units; **`LoLimit`** and **`HiLimit`** are scalar doubles.

1.3.1.2 `Coord2VecInterval`

Optional units; **`LoLimit2Vec`** and **`HiLimit2Vec`** are arrays of two doubles.

1.3.1.3 `Coord3VecInterval`

Optional units; **`LoLimit3Vec`** and **`HiLimit3Vec`** are arrays of three doubles.

1.3.2 Time Interval (`TimeInterval`)

`StartTime` and **`StopTime`** are of class AstronTime (see Section 1.2.2).

1.3.3 Spatial Area (`PositionInterval`)

1.3.3.1 1-Dimensional Intervals (`PositionScalarInterval`)

Spatial position units required; **`LoLimit`** and **`HiLimit`** are scalar doubles.

1.3.3.2 2-Dimensional Rectangles (`Coord2VecInterval`)

One or more intervals defined by vector pairs (BLC/TRC).
Spatial position units required; **`LoLimit2Vec`** and **`HiLimit2Vec`** are arrays of two doubles.

1.3.3.3 2-Dimensional `Region`

A **`Region`** *may* be a shape or the result of an operation on one or more Regions. It shall optionally contain its **`Area`** and associated unit. The supported shapes are: **`Allsky`**, **`Circle`**, **`Ellipse`**, **`Polygon`**, **`Box`**, **`Sector`**, **`Convex`**, **`ConvexHull`**, and **`SkyIndex`**. The supported operations are **`Union`**, **`Intersection`**, **`Negation`**, and **`Difference`**.

1.3.3.3.1 `AllSky`

For convenience.

1.3.3.3.2 `Circle`

Defined by a 2-D Position and a radius.

1.3.3.3.3 `Ellipse`

Defined by a 2-D Position, semi-major and semi-minor axes, and position angle.

1.3.3.3.4 `Polygon`

Defined by two or more vertices connected by straight lines (Cartesian) or great or small circles (Spherical). Small circle sides may be specified by adding a **`SmallCircle`** element to a **`Vertex`** which specifies the pole that defines the curvature of the small circle between that vertex and its predecessor.

1.3.3.3.5 `Box`

For convenience; it is a special case of polygon, allowing only great circle sides.

1.3.3.3.6 `Sector`

The area swept by position angle between two half lines starting in a common position and defined by a position angle.

1.3.3.3.7 `Convex`

Convex polygon on unit sphere, defined by two or more **`HalfSpace`**s.

1.3.3.3.8 `ConvexHull`

The smallest convex polygon that contains all its points; in spherical coordinates all points have to be contained within a hemisphere

1.3.3.3.9 `SkyIndex`

Hook for future use, allowing the specification of a Region through a sky indexing scheme.

1.3.3.3.10 `Union`

Union of two or more Regions.

1.3.3.3.11 `Intersection`

Intersection of two or more Regions.

1.3.3.3.12 `Negation`

Negation of one Region.

1.3.3.3.13 `Difference`

Difference of two Regions; for convenience, since it can, in principle, be constructed through a combination of Intersection and Negation.

1.3.3.4 3-Dimensional Intervals (`Coord3VecInterval`)

One or more brick-shaped intervals defined by vector pairs (BLC/TRC).
Spatial position units required; **`LoLimit3Vec`** and **`HiLimit3Vec`** are arrays of three doubles.

1.3.3.5 3-Dimensional Sphere (`Sphere`)

Defined by a 3-D Position and a radius.
Spatial position units required

1.3.4 Spatial Velocity

Defined by one or more 1-D, 2-D, or 3-D intervals.

1.3.4.1 1-Dimensional Intervals (`VelicityScalarInterval`)

Spatial position and time units required; **`LoLimit`** and **`HiLimit`** are scalar doubles.

1.3.4.2 2-Dimensional Rectangles (`Velocity2VecInterval`)

One or more intervals defined by vector pairs (BLC/TRC).
Spatial position and time units required; **`LoLimit2Vec`** and **`HiLimit2Vec`** are arrays of two doubles.

1.3.4.3 3-Dimensional Intervals (`Velocity3VecInterval`)

One or more brick-shaped intervals defined by vector pairs (BLC/TRC).
Spatial position and time units required; **`LoLimit3Vec`** and **`HiLimit3Vec`** are arrays of three doubles.

1.3.4.4 3-Dimensional Sphere (`VelocitySphere`)

Defined by a 3-D Velocity and a radius.
Spatial position and time units required.

1.3.5 Spectral Interval (`SpectralInterval`)

Spectral units required; **`LoLimit`** and **`HiLimit`** are scalar doubles.

1.3.6 Redshift Interval (`RedshiftInterval`)

Spatial position and time units required if expressed in terms of Doppler velocity; **`LoLimit`** and **`HiLimit`** are scalar doubles.

1.3.7 Pixel Interval (`PixelCoordInterval`)

The **`PixelCoordInterval`** needs to refer to a specific Coordinate Frame through a **`frame_id`**. There are three types of **`PixelCoordInterval`**:

1.3.7.1 `PixelCoordScalarInterval`

Optional units; **`LoLimit`** and **`HiLimit`** are scalar doubles.

1.3.7.2 `PixelCoord2VecInterval`

Optional units; **`LoLimit2Vec`** and **`HiLimit2Vec`** are arrays of two doubles.

1.3.7.3 `PixelCoord3VecInterval`

Optional units; **`LoLimit3Vec`** and **`HiLimit3Vec`** are arrays of three doubles.

2 Units

This section list all units strings as they are recognized in the STC schema.

2.1 Time Units

s (second)
h (hour = 3600 **s**)
d (day = 86400 **s**)
a or **yr** (Julian year = 365.25 **d**)
cy (Julian century = 36525 **d**)
'' (empty, i.e., dimensionless; for ISO-8601 format)

2.2 Spectral Coordinate Units

2.2.1 Frequency

Hz
kHz
MHz
GHz

2.2.2 Wavelength

m
mm
um (micrometer)
nm
Angstrom

2.2.3 Energy

eV
keV
MeV
GeV
TeV

2.3 Spatial Coordinate Units

2.3.1 Angular Units

deg (degree)
rad (radian)
h (hour)
arcmin
arcsec

2.3.2 Linear Units

m (meter)
km
mm
AU
pc (parsec)
kpc
Mpc
lyr (lightyear)
'' (empty, i.e., dimensionless, for unit sphere)

2.3.3 Three-units Strings

These strings are for special 3-D vectors where the components do not share the same unit; one may prefer to give each component its own unit, instead:

'**deg deg m**'
'**deg deg Mpc**'

2.3.4 Position Angles

Then there is the position angle definition (not strictly a unit). Position angles may be counted from North (through East), (positive) X (to positive Y), or (positive) Y (to positive X) axis:

North
X
Y

Velocity units are constructed as **posUnitType** / **velTimeType** and the position and velocityTime units are specified separately in order to reduce the size of the enumeration list the latter may be second, day, hour, year (**a** or **yr**), century, but not empty.

3 Usage

3.1 Referencing

It shall be possible to reference the same object multiple times in a metadataset.
It shall be possible to assign values to metadata components through references.
It shall be possible to refer to components within and outside the same metadataset.

3.2 Use Context

The STC metadata description may be used in a number or contexts. Listed here are the four comprehensive cases, but these are not exclusive.

3.2.1 Resource Profile (`STCResourceProfile`)

Describe coverage, coordinate systems, and properties (e.g., resolution, field-of-view size, accuracy) of data in a repository.

3.2.2 Search (`SearchLocation`)

Describe coverage, coordinate systems, and properties (e.g., resolution, field-of-view size, accuracy, pixel size) of data requested in a query.

3.2.3 Catalog Data (`CatalogEntryLocation`)

STC metadata for a data table: coverage, accuracy, resolution, coordinate values – optionally pointed to through a reference.

3.2.4 Observational Data (`ObsDataLocation`)

In this case there shall be STC metadata for the observer's location (**`ObservatoryLocation`**) as well as for the observational data (**`ObservationLocation`**) themselves.

4 Global Elements

The following table provides a list of all global elements and their substitution group hierarchy.

Column 1:	Name of element
Column 2:	If the element is the head element of a substitution group, this column contains an *s*
Column 3:	If the element is *abstract*, this column contains an *a*
Column 4:	If the element is *nillable*, this column contains an *n*; elements that are not *nillable* cannot be linked to
Column 5:	If the element cannot consist of a link (through an *Xlink* reference or an *ID*/*IDREF* pair), this column contains an *x*; if the element is not linkable, it should not be *nillable*, either
Column 6:	If the element is of type *xs:anyType*, this column contains a *t*
Column 7:	Name of substitution group that the element is the head element of (first level)
Column 8:	Name of substitution group that the element is a member of (first level)
Column 9:	Name of substitution group that the element is the head element of (second level)
Column 10:	Name of substitution group that the element is a member of (second level)

Elements	SubstGr	Abstract	Nillable	No link	AnyType	Head Elem of SubstGr	Member of SubstGr	Head Elem of SubstGr	Member of SubstGr
STCmetadata	s		n			STCmetaData			
CatalogEntryLocation			n				STCmetaData		
ObsDataLocation			n				STCmetaData		
SearchLocation			n				STCmetaData		
STCDescription			n				STCmetaData		
STCResourceProfile			n				STCmetaData		
CoordSys	s		n			CoordSys			
AstroCoordSystem			n				CoordSys		
PixelCoordSystem			n				CoordSys		
CoordFrame			n						
TimeFrame			n						
SpaceFrame			n						
SpectralFrame			n						
RedshiftFrame			n						
PixelCoordFrame			n						
SpaceRefFrame	s	a		x		SpaceRefFrame			
AZ_EL				x			SpaceRefFrame		
BODY				x			SpaceRefFrame		
ECLIPTIC				x			SpaceRefFrame		
FK4				x			SpaceRefFrame		
FK5				x			SpaceRefFrame		
GALACTIC_I				x			SpaceRefFrame		
GALACTIC_II				x			SpaceRefFrame		
GEO_C				x			SpaceRefFrame		
GEO_D				x			SpaceRefFrame		
GSE				x			SpaceRefFrame		
GSM				x			SpaceRefFrame		
HEE				x			SpaceRefFrame		
HEEQ				x			SpaceRefFrame		
HGC				x			SpaceRefFrame		
HGI				x			SpaceRefFrame		
HGS				x			SpaceRefFrame		
HPC				x			SpaceRefFrame		
HPR				x			SpaceRefFrame		
HRTN				x			SpaceRefFrame		
ICRS				x			SpaceRefFrame		
JUPITER_C_III				x			SpaceRefFrame		
JUPITER_G_III				x			SpaceRefFrame		
LUNA_C				x			SpaceRefFrame		
LUNA_G				x			SpaceRefFrame		
MAG				x			SpaceRefFrame		
MARS_C				x			SpaceRefFrame		
MARS_G				x			SpaceRefFrame		
MERCURY_C				x			SpaceRefFrame		
MERCURY_G				x			SpaceRefFrame		
NEPTUNE_C_III				x			SpaceRefFrame		
NEPTUNE_G_III				x			SpaceRefFrame		
PLUTO_C				x			SpaceRefFrame		
PLUTO_G				x			SpaceRefFrame		

Elements	SubstGr	Abstract	Nillable	No link	AnyType	Head Elem of SubstGr	Member of SubstGr	Head Elem of SubstGr	Member of SubstGr
SATURN_C_III				x			SpaceRefFrame		
SATURN_G_III				x			SpaceRefFrame		
SM				x			SpaceRefFrame		
SUPER_GALACTIC				x			SpaceRefFrame		
URANUS_C_III				x			SpaceRefFrame		
URANUS_G_III				x			SpaceRefFrame		
VENUS_C				x			SpaceRefFrame		
VENUS_G				x			SpaceRefFrame		
UNKNOWNFrame				x			SpaceRefFrame		
CoordRefFrame	s	a		x			SpaceRefFrame	CoordRefFrame	
Cart2DRefFrame				x					CoordRefFrame
Cart3DRefFrame				x					CoordRefFrame
ScalarRefFrame				x					CoordRefFrame
SphericalRefFrame				x					CoordRefFrame
CustomSpaceRefFrame	s	a		x	t	CustomSpaceRefFrame			
Cart1DSpaceRefFrame				x			CustomSpaceRefFrame		
Cart2DSpaceRefFrame				x			CustomSpaceRefFrame		
Cart3DSpaceRefFrame				x			CustomSpaceRefFrame		
SphericalSpaceRefFrame				x			CustomSpaceRefFrame		
CTransform2	s	a		x	t	CTransform2			
Transform2			n				CTransform2		
Transform2Matrix			n				CTransform2		
CTransform3	s	a		x	t	CTransform3			
Transform3			n				CTransform3		
Transform3Matrix			n				CTransform3		
ReferencePosition	s	a		x		ReferencePosition			
CoordRefPos				x			ReferencePosition		
BARYCENTER				x			ReferencePosition		
EMBARYCENTER				x			ReferencePosition		
GALACTIC_CENTER				x			ReferencePosition		
GEOCENTER				x			ReferencePosition		
HELIOCENTER				x			ReferencePosition		
JUPITER				x			ReferencePosition		
LOCAL_GROUP_CENTER				x			ReferencePosition		
LSR				x			ReferencePosition		
LSRD				x			ReferencePosition		
LSRK				x			ReferencePosition		
MARS				x			ReferencePosition		
MERCURY				x			ReferencePosition		
MOON				x			ReferencePosition		
NEPTUNE				x			ReferencePosition		
PLUTO				x			ReferencePosition		
RELOCATABLE				x			ReferencePosition		
SATURN				x			ReferencePosition		
TOPOCENTER				x			ReferencePosition		
URANUS				x			ReferencePosition		
VENUS				x			ReferencePosition		
UNKNOWNRefPos				x			ReferencePosition		
CoordFlavor	s	a		x		CoordFlavor			

Elements	SubstGr	Abstract	Nillable	No link	AnyType	Head Elem of SubstGr	Member of SubstGr	Head Elem of SubstGr	Member of SubstGr
CARTESIAN				x			**CoordFlavor**		
CYLINDRICAL				x			**CoordFlavor**		
HEALPIX				x			**CoordFlavor**		
POLAR				x			**CoordFlavor**		
SPHERICAL				x			**CoordFlavor**		
UNITSPHERE				x			**CoordFlavor**		
STRING				x			**CoordFlavor**		
Coords	s		n			**Coords**			
AstroCoords			n				**Coords**		
PixelCoords			n				**Coords**		
Coordinate	s		n			**Coordinate**			
Redshift			n				**Coordinate**		
Spectral			n				**Coordinate**		
GenCoordinate	s		n				**Coordinate**	**GenCoordinate**	
ScalarCoordinate			n						**GenCoordinate**
StringCoordinate			n						**GenCoordinate**
Vector2DCoordinate			n						**GenCoordinate**
Vector3DCoordinate			n						**GenCoordinate**
Position	s		n				**Coordinate**	**Position**	
Position1D			n						**Position**
Position2D			n						**Position**
Position3D			n						**Position**
Velocity	s		n				**Coordinate**	**Velocity**	
Velocity1D			n						**Velocity**
Velocity2D			n						**Velocity**
Velocity3D			n						**Velocity**
Pixel	s		n				**Coordinate**	**Pixel**	
Pixel1D			n						**Pixel**
Pixel2D			n						**Pixel**
Pixel3D			n						**Pixel**
Time			n						
AbsoluteTime	s	a		x	t	**AbsoluteTime**			
ISOTime			n				**AbsoluteTime**		
JDTime			n				**AbsoluteTime**		
MJDTime			n				**AbsoluteTime**		
TimeOrigin				x			**AbsoluteTime**		
TimeOffset			n						
CoordValue	s	a		x	t	**CoordValue**			
Value			n				**CoordValue**		
CValue2	s	a		x	t		**CoordValue**	**CValue2**	
Curve2			n						**CValue2**
Value2			n						**CValue2**
CValue3	s	a		x	t		**CoordValue**	**CValue3**	
Curve3			n						**CValue3**
Value3			n						**CValue3**
CError	s	a		x	t	**CError**			
Error			n				**CError**		
CError2	s	a		x	t		**CError**	**CError2**	
Error2			n						**CError2**

Elements	SubstGr	Abstract	Nillable	No link	AnyType	Head Elem of SubstGr	Member of SubstGr	Head Elem of SubstGr	Member of SubstGr
Error2Matrix			n						CError2
Error2Radius			n						CError2
CError3	s	a		x	t		CError	CError3	
Error3			n						CError3
Error3Matrix			n						CError3
Error3Radius			n						CError3
CResolution	s	a		x	t	CResolution			
Resolution			n				CResolution		
CResolution2	s	a		x	t		CResolution	CResolution2	
Resolution2			n						CResolution2
Resolution2Matrix			n						CResolution2
Resolution2Radius			n						CResolution2
CResolution3	s	a		x	t		CResolution	CResolution3	
Resolution3			n						CResolution3
Resolution3Matrix			n						CResolution3
Resolution3Radius			n						CResolution3
CSize	s	a		x	t	CSize			
Size			n				CSize		
CSize2	s	a		x	t		CSize	CSize2	
Size2			n						CSize2
Size2Matrix			n						CSize2
Size2Radius			n						CSize2
CSize3	s	a		x	t		CSize	CSize3	
Size3			n						CSize3
Size3Matrix			n						CSize3
Size3Radius			n						CSize3
CPixSize	s	a		x	t	CPixSize			
PixSize			n				CPixSize		
CPixSize2	s	a		x	t		CPixSize	CPixSize2	
PixSize2			n						CPixSize2
PixSize2Matrix			n						CPixSize2
CPixSize3	s	a		x	t		CPixSize	CPixSize3	
PixSize3			n						CPixSize3
PixSize3Matrix			n						CPixSize3
CoordArea	s		n			CoordArea			
AstroCoordArea			n				CoordArea		
PixelCoordArea			n				CoordArea		
CoordInterval	s		n			CoordInterval			
Coord2VecInterval			n				CoordInterval		
Coord3VecInterval			n				CoordInterval		
CoordScalarInterval			n				CoordInterval		
PositionInterval	s		n			PositionInterval			
PositionScalarInterval			n				PositionInterval		
Position2VecInterval			n				PositionInterval		
Position3VecInterval			n				PositionInterval		
RegionFile			n				PositionInterval		
Sphere			n				PositionInterval		
Region	s		n				PositionInterval	Region	
AllSky			n						Region

Elements	SubstGr	Abstract	Nillable	No link	AnyType	Head Elem of SubstGr	Member of SubstGr	Head Elem of SubstGr	Member of SubstGr
Box			**n**						**Region**
Circle			**n**						**Region**
Convex			**n**						**Region**
ConvexHull			**n**						**Region**
Difference			**n**						**Region**
Ellipse			**n**						**Region**
Intersection			**n**						**Region**
Negation			**n**						**Region**
Polygon			**n**						**Region**
Sector			**n**						**Region**
SkyIndex			**n**						**Region**
Union			**n**						**Region**
VelocityInterval	**s**		**n**			**VelocityInterval**			
Velocity2VecInterval			**n**				**VelocityInterval**		
Velocity3VecInterval			**n**				**VelocityInterval**		
VelocityScalarInterval			**n**				**VelocityInterval**		
VelocitySphere			**n**				**VelocityInterval**		
PixelCoordInterval	**s**		**n**			**PixelCoordInterval**			
PixelCoord2VecInterval			**n**				**PixelCoordInterval**		
PixelCoord3VecInterval			**n**				**PixelCoordInterval**		
PixelCoordScalarInterval			**n**				**PixelCoordInterval**		