International
Virtual
Observatory
Alliance

IVOA STC Model: Coordinate Metadata for the Virtual Observatory
III. Coordinate Transformations

Version 2.0

IVOA Working Draft 20170919

This version:
WD-STC-2.0-20170919
Previous version(s):

Editor(s):
Arnold Rots

Authors:
Arnold Rots and Mark Cresitello-Dittmar

Abstract

STC2 is version 2 of the Data Model for the metadata describing Space-Time, related, and other Coordinates. These metadata are to be used for specifying coordinate-related information for datasets, catalogs, and queries. It consists of five sub-models that have uni-directional dependencies.

1. **Coordinates** provides the metadata for the coordinate frames and coordinate locations. It consists of three packages:
   1.1. CoordSystems provides the basic model for constructing coordinate frames and for collecting them in coordinate systems
   1.2. Coordinates defines the general model for specifying coordinate values
   1.3. Domain sub-packages provide the concrete model design for the five astronomical domains (temporal, spatial, spectral, redshift, and polarization), generic coordinates (any other – usually dependent – coordinate variable), and pixel coordinates
2. **Transforms** models the specification of coordinate frame transformations and are defined as a mapping from one coordinate frame into another
3. **Measurement** extends the Coordinate concept to associate uncertainties and resolution measures with particular coordinate values
4. **CoordinateArea** allows the specification of the volume in coordinate space that a particular dataset occupies
5. **Region** is a specialization of CoordinateArea specifically for two-dimensional spatial coordinates

This document describes the **STC2 Coordinates** model.

Status of This Document

This is an IVOA Working Draft for review by IVOA members and other interested parties. It is a draft document and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use IVOA Working Drafts as reference materials or to cite them as other than "work in progress".

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 Euro-VO projects (European Commission 7th program): Euro-VO Aida, VO-ICE and CoSADIE.

Change Log:

2017-09-14

Contents

1 Introduction

1.1 Motivation

Astronomy, being primarily being a science that crucially depends on observations, has a very basic need for complete, accurate, and unambiguous metadata regarding coordinate information, meaning all coordinates of the observable space and noting that several of these are intertwined. The Data Model described in this document aims to provide a model for such metadata, satisfying the requirements.

1.2 Requirements

The primary goal of this document is the specification of a Data Model for coordinate metadata that satisfies the following requirements; the Data Model *shall*:

1. Cover all coordinate axes of observable space: Time, Space, Electro-magnetic Spectrum, Redshift (or Doppler Velocity), Polarization, and have the ability to cover any other incidental coordinates (e.g., temperature)
2. Provide metadata that are complete
3. Provide metadata that are unambiguous
4. Provide metadata that are accurate
5. Be expressed in VO-DML
6. Allow usage of only relevant subsets of the metadata, with the proviso that they must satisfy Requirements 2, 3, and 4
7. Be extensible

1.3 Context and Scope

STC2 is version 2 of the Data Model for the metadata describing Space-Time, related, and other Coordinates. These metadata are to be used for specifying coordinate-related information for datasets, catalogs, and queries. It consists of five sub-models that have uni-directional dependencies.

1. **Coordinates** provides the metadata for the coordinate frames and coordinate locations. It consists of three packages:
   1.1. CoordSystems provides the basic model for constructing coordinate frames and for collecting them in coordinate systems
   1.2. Coordinates defines the general model for specifying coordinate values
   1.3. Domain sub-packages provide the concrete model design for the five astronomical domains (temporal, spatial, spectral, redshift, and polarization), generic coordinates (any other – usually dependent – coordinate variable), and pixel coordinates
2. **Transforms** models the specification of coordinate frame transformations and are defined as a mapping from one coordinate frame into another
3. **Measurement** extends the Coordinate concept to associate uncertainties and resolution measures with particular coordinate values
4. **CoordinateArea** allows the specification of the volume in coordinate space that a particular dataset occupies
5. **Region** is a specialization of CoordinateArea specifically for two-dimensional spatial coordinates

This document describes the **STC2 Transforms** model.

1.4 Model Representation

The model is represented in this document in the Uniform Modeling Language (UML):

- Classes have a orange header
- Data types have a yellow header
- Generalizations (inheritance) are indicated by red lines
- Associations are represented by green lines
- Compositions are represented by blue lines
- Elements with red headers represent parent elements imported from another package
- Elements with green headers represent associated elements imported from another package

1.5 Role in the IVOA Architecture

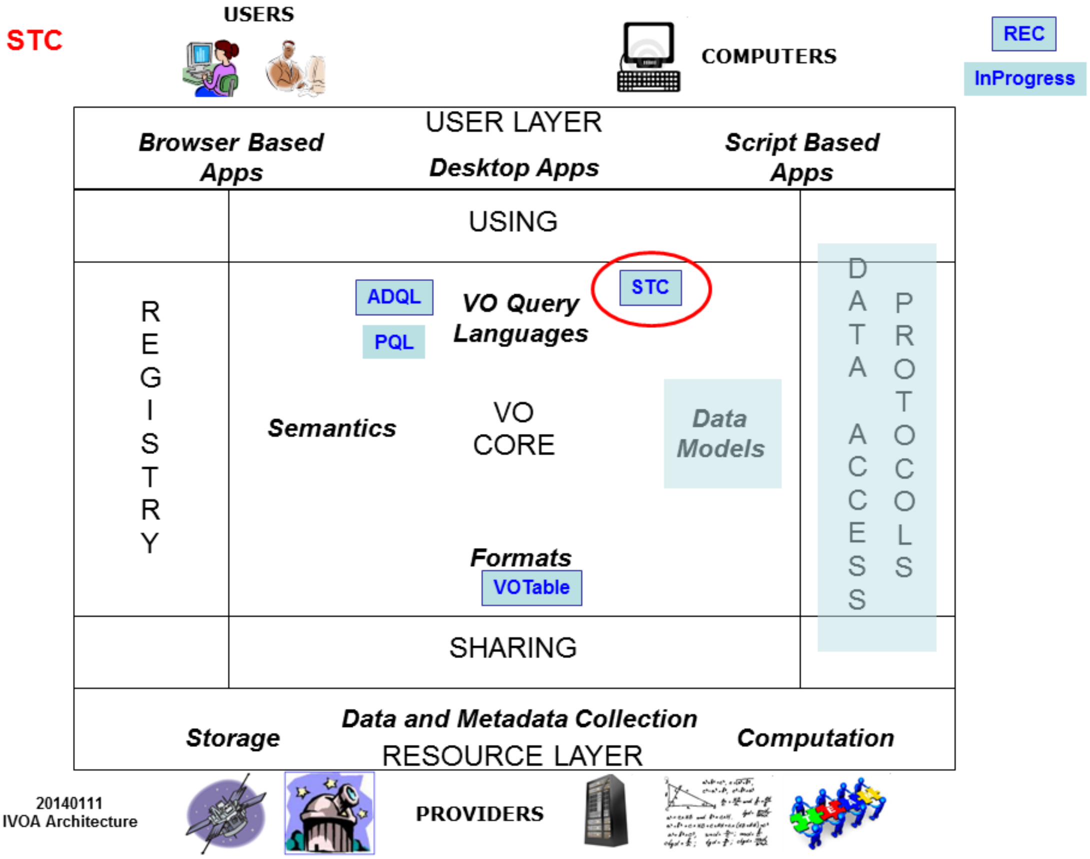


Figure 1 IVOA Standards Context

2 Index of Elements

Index of elements:

Class	CoordFrame	Enum2D	Enum3D	EnumElement1
	EnumElement2	EnumElement3	EnumScalar	FrameTransform
	PolCircular	PolFrameTransform	PolLinear	PolStokes
	PolVector	PolyCoeff1D	PolyCoeff2D	PolyCoeff3D
	Polynomial1D	Polynomial2D	Polynomial3D	Projection1D
	Projection2D	Projection3D	Rotate2D	Rotate3D
	Scale1D	Scale2D	Scale3D	TransformMatrix2D
	TransformMatrix3D	Xform	Xform1D	Xform2D
	Xform3D	Xlate1D	Xlate2D	Xlate3D
DataType	complex	IntegerQuantity	Matrix	Matrix2x2
	Matrix3x3	Quantity	rational	RealQuantity
Enumeration	PolCircularEnum	PolEnum	PolLinearEnum	PolStokesEnum
	PolVectorEnum	Projection		
Package	frametransforms			

Model **stc2_transforms**

owner	Root
properties	qualified name stc2_transforms «model» true author ahr title STC2 Transforms version 2.0 uri http://ivoa.net/vodml/stc2_meas.vo-dml
ownedMember	frametransforms ivoa stc2_coordinates

2.1 *Imported Models*

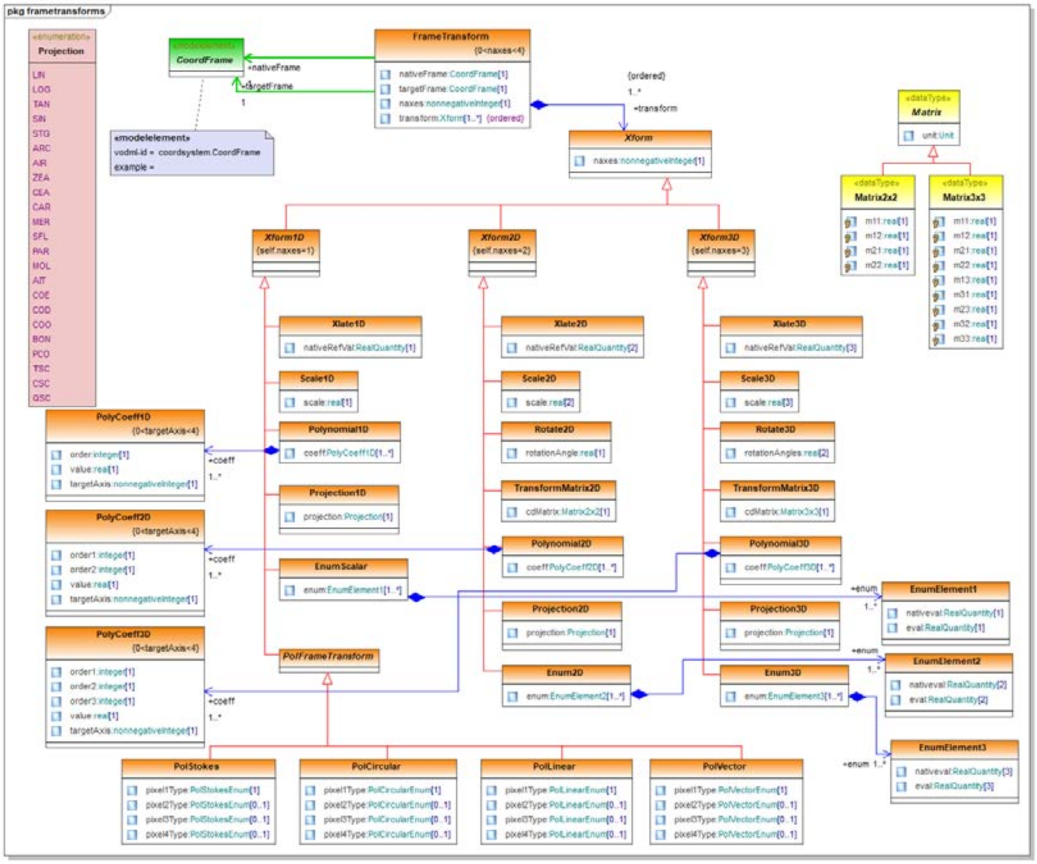
The standard IVOA model, defining primitives and quantities, is imported. Also, a number of elements from the STC2 Coordinates model are imported; these are detailed in Section 5.

3 Package Frame Transforms

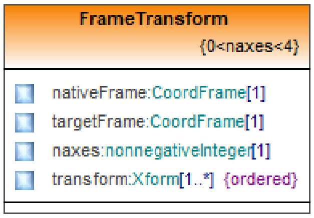
Package **frametransforms**

owner	stc2_transforms
properties	qualified name stc2_transforms::frametransforms
ownedDiagrams	FrameTransforms
ownedMember	Enum2D Enum3D EnumElement1 EnumElement2 EnumElement3 EnumScalar FrameTransform Matrix Matrix2x2 Matrix3x3 PolCircular PolFrameTransform PolLinear PolStokes PolVector PolyCoeff1D PolyCoeff2D PolyCoeff3D Polynomial1D Polynomial2D Polynomial3D Projection Projection1D Projection2D Projection3D Rotate2D Rotate3D Scale1D Scale2D Scale3D TransformMatrix2D TransformMatrix3D Xform Xform1D Xform2D Xform3D Xlate1D Xlate2D Xlate3D
source of relation	Dependency ivoa **IVOA_UML_Profile**

Class Diagram **FrameTransforms** (frametransforms)



3.1.1 Class FrameTransform

diagram	 FrameTransform {0<naxes<4} nativeFrame:CoordFrame[1] targetFrame:CoordFrame[1] naxes:nonnegativeInteger[1] transform:Xform[1..*] {ordered}
owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::FrameTransform abstract false
ownedMember	**Constraint1** **nativeFrame** **naxes** **targetFrame** **transform**
shown on diagram	**FrameTransforms**
constraints	Constraint1: 0<naxes<4
documentation	This mapping class provides a transformation from the nativeFrame to the targetFrame. The transform is specified by one or more Xform objects. The Xform objects specify atomic transform operations and may be chained together (in a specific order) to define more complicated transformations. naxes specifies the dimensionality of the transformation and should reflect the dimensionality of the native and target frames.

Constraint **FrameTransform::Constraint1**

owner	**FrameTransform**	
properties	qualified name	stc2_transforms::frametransforms::FrameTransform::Constraint1
	specification	0<naxes<4
	constrained elements	FrameTransform

Property **FrameTransform::nativeFrame**

owner	**FrameTransform**	
properties	qualified name	stc2_transforms::frametransforms::FrameTransform::nativeFrame
	multiplicity	1
	type	CoordFrame

Property **FrameTransform::naxes**

owner	**FrameTransform**	
properties	qualified name	stc2_transforms::frametransforms::FrameTransform::naxes
	multiplicity	1
	type	nonnegativeInteger

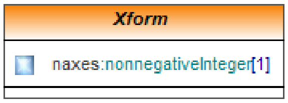
Property **FrameTransform::targetFrame**

owner	**FrameTransform**	
properties	qualified name	stc2_transforms::frametransforms::FrameTransform::targetFrame
	multiplicity	1
	type	CoordFrame

Property **FrameTransform::transform**

owner	**FrameTransform**	
properties	qualified name	stc2_transforms::frametransforms::FrameTransform::transform
	multiplicity	1..*
	type	Xform
	aggregation	composite

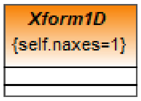
3.1.2 Class Xform

diagram	
owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::Xform abstract true
ownedMember	**naxes**
specific	**Xform1D** **Xform2D** **Xform3D**
typedElements	Class **FrameTransform** Property **transform**
shown on diagram	**FrameTransforms**
documentation	The abstract Xform class. At the next level it is split into subclasses for 1-, 2-, and 3-dimensional transforms.

Property **Xform::naxes**

owner	**Xform**	
properties	qualified name	stc2_transforms::frametransforms::Xform::naxes
	multiplicity	1
	type	nonnegativeInteger

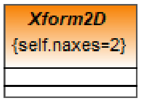
3.1.3 Class Xform1D

diagram	
owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::Xform1D abstract true
ownedMember	**Constraint1**
general	**Xform**
specific	**EnumScalar** **PolFrameTransform** **Polynomial1D** **Projection1D** **Scale1D** **Xlate1D**
shown on diagram	**FrameTransforms**
constraints	Constraint1: self.naxes=1
documentation	The abstract class for atomic 1-dimensional transformations.

Constraint **Xform1D::Constraint1**

owner	**Xform1D**
properties	qualified name stc2_transforms::frametransforms::Xform1D::Constraint1 specification self.naxes=1 constrained elements Xform1D

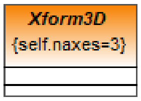
3.1.4 Class Xform2D

diagram	
owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::Xform2D abstract true
ownedMember	**Constraint1**
general	**Xform**
specific	**Enum2D** **Polynomial2D** **Projection2D** **Rotate2D** **Scale2D** **TransformMatrix2D** **Xlate2D**
shown on diagram	**FrameTransforms**
constraints	Constraint1: self.naxes=2
documentation	The abstract class for atomic 2-dimensional transformations.

Constraint **Xform2D::Constraint1**

owner	**Xform2D**
properties	qualified name stc2_transforms::frametransforms::Xform2D::Constraint1 specification self.naxes=2 constrained elements Xform2D

3.1.5 Class Xform3D

diagram	
owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::Xform3D abstract true
ownedMember	**Constraint1**
general	**Xform**
specific	**Enum3D** **Polynomial3D** **Projection3D** **Rotate3D** **Scale3D** **TransformMatrix3D** **Xlate3D**
shown on diagram	**FrameTransforms**
constraints	Constraint1: self.naxes=3
documentation	The abstract class for atomic 3-dimensional transformations.

Constraint **Xform3D::Constraint1**

owner	**Xform3D**
properties	qualified name stc2_transforms::frametransforms::Xform3D::Constraint1 specification self.naxes=3 constrained elements Xform3D

3.1.6 Class Xlate1D

diagram	
owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::Xlate1D abstract false
ownedMember	**nativeRefVal**
general	**Xform1D**
shown on diagram	**FrameTransforms**
documentation	Simple 1-dimensional translation: targetValue = nativeValue - nativeRefVal.

Property **Xlate1D::nativeRefVal**

owner	**Xlate1D**
properties	qualified name stc2_transforms::frametransforms::Xlate1D::nativeRefVal multiplicity 1 type RealQuantity

3.1.7 Class Xlate2D

diagram	Xlate2D nativeRefVal:RealQuantity[2]
owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::Xlate2D abstract false
ownedMember	**nativeRefVal**
general	**Xform2D**
shown on diagram	**FrameTransforms**
documentation	Simple 2-dimensional translation: vector targetValue = nativeValue - nativeRefVal.

Property **Xlate2D::nativeRefVal**

owner	**Xlate2D**
properties	qualified name stc2_transforms::frametransforms::Xlate2D::nativeRefVal ordered true multiplicity 2 type RealQuantity

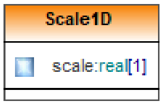
3.1.8 Class Xlate3D

diagram	Xlate3D nativeRefVal:RealQuantity[3]
owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::Xlate3D abstract false
ownedMember	**nativeRefVal**
general	**Xform3D**
shown on diagram	**FrameTransforms**
documentation	Simple 3-dimensional translation: vector targetValue = nativeValue - nativeRefVal.

Property **Xlate3D::nativeRefVal**

owner	**Xlate3D**
properties	qualified name stc2_transforms::frametransforms::Xlate3D::nativeRefVal ordered true multiplicity 3 type RealQuantity

3.1.9 Class Scale1D

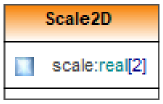
diagram	

owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::Scale1D abstract false
ownedMember	**scale**
general	**Xform1D**
shown on diagram	**FrameTransforms**
documentation	Simple 1-dimensional scaling: targetValue = scale * nativeValue

Property **Scale1D::scale**

owner	**Scale1D**
properties	qualified name stc2_transforms::frametransforms::Scale1D::scale multiplicity 1 type real

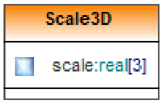
3.1.10 Class Scale2D

diagram	
owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::Scale2D abstract false
ownedMember	**scale**
general	**Xform2D**
shown on diagram	**FrameTransforms**
documentation	Simple 2-dimensional scaling: vector elements targetValue[i] = scale[i] * nativeValue[i]

Property **Scale2D::scale**

owner	**Scale2D**
properties	qualified name stc2_transforms::frametransforms::Scale2D::scale ordered true multiplicity 2 type real

3.1.11 Class Scale3D

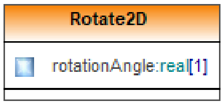
diagram	
owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::Scale3D abstract false
ownedMember	**scale**
general	**Xform3D**

shown on diagram	FrameTransforms
documentation	Simple 3-dimensional scaling: vector elements targetValue[i] = scale[i] * nativeValue[i]

Property **Scale3D::scale**

owner	Scale3D
properties	qualified name stc2_transforms::frametransforms::Scale3D::scale ordered true multiplicity 3 type real

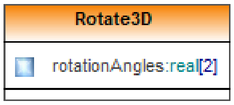
3.1.12 Class Rotate2D

diagram	
owner	frametransforms
properties	qualified name stc2_transforms::frametransforms::Rotate2D abstract false
ownedMember	rotationAngle
general	Xform2D
shown on diagram	FrameTransforms
documentation	Simple 2-dimensional rotation in degrees CCW.

Property **Rotate2D::rotationAngle**

owner	Rotate2D
properties	qualified name stc2_transforms::frametransforms::Rotate2D::rotationAngle multiplicity 1 type real

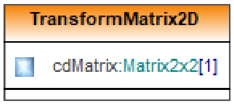
3.1.13 Class Rotate3D

diagram	
owner	frametransforms
properties	qualified name stc2_transforms::frametransforms::Rotate3D abstract false
ownedMember	rotationAngles
general	Xform3D
shown on diagram	FrameTransforms
documentation	Simple 3-dimensional rotation in degrees. The first rotation angle indicates CCW rotation along the Z-axis, the second CCW rotation around the X-axis.

Property **Rotate3D::rotationAngles**

owner	**Rotate3D**
properties	qualified name stc2_transforms::frametransforms::Rotate3D::rotationAngles ordered true multiplicity 2 type real

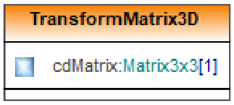
3.1.14 Class TransformMatrix2D

diagram	
owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::TransformMatrix2D abstract false
ownedMember	**cdMatrix**
general	**Xform2D**
shown on diagram	**FrameTransforms**
documentation	CD-matrix transformation: targetVector = cdMatrix . nativeVector (scale and rotate)

Property **TransformMatrix2D::cdMatrix**

owner	**TransformMatrix2D**
properties	qualified name stc2_transforms::frametransforms::TransformMatrix2D::cdMatrix multiplicity 1 type Matrix2x2

3.1.15 Class TransformMatrix3D

diagram	
owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::TransformMatrix3D abstract false
ownedMember	**cdMatrix**
general	**Xform3D**
shown on diagram	**FrameTransforms**
documentation	CD-matrix transformation: targetVector = cdMatrix . nativeVector (scale and rotate)

Property **TransformMatrix3D::cdMatrix**

owner	**TransformMatrix3D**
properties	qualified name stc2_transforms::frametransforms::TransformMatrix3D::cdMatrix multiplicity 1 type Matrix3x3

3.1.16 DataType Matrix

owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::Matrix abstract true
ownedMember	**unit**
specific	**Matrix2x2** **Matrix3x3**
shown on diagram	**FrameTransforms**
documentation	Matrix is the abstract parent data type for 2-D and 3-D transformation matrices. It contains an optional unit that allows for a change in units between native and target values. If unit is absent targetValue SHALL retain the units of the nativeValue.

Property **Matrix::unit**

owner	**Matrix**
properties	qualified name stc2_transforms::frametransforms::Matrix::unit type Unit

3.1.17 DataType Matrix2x2

owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::Matrix2x2 abstract false
ownedMember	**m11** **m12** **m21** **m22**
general	**Matrix**
typedElements	Class **TransformMatrix2D** Property **cdMatrix**
shown on diagram	**FrameTransforms**
documentation	Matrix2x2 is data type for 2-D transformation matrices. It inherits an optional unit from Matrix that allows for a change in units between native and target values. If unit is absent targetValue SHALL retain the units of the nativeValue.

Property **Matrix2x2::m11,m12,m21,m22**

owner	**Matrix2x2**
properties	qualified name stc2_transforms::frametransforms::Matrix2x2::m11 multiplicity 1

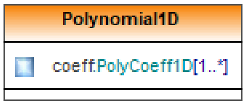
3.1.18 DataType Matrix3x3

owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::Matrix3x3 abstract false
ownedMember	**m11** **m12** **m13** **m21** **m22** **m23** **m31** **m32** **m33**
general	**Matrix**
typedElements	Class **TransformMatrix3D** Property **cdMatrix**
shown on diagram	**FrameTransforms**
documentation	Matrix3x3 is data type for 3-D transformation matrices. It inherits an optional unit from Matrix that allows for a change in units between native and target values. If unit is absent targetValue SHALL retain the units of the nativeValue.

Property **Matrix3x3::m11,m12,m13,m21,m22,m23,m31,m32,m33**

owner	**Matrix3x3**
properties	qualified name stc2_transforms::frametransforms::Matrix3x3::m11 multiplicity 1 type real

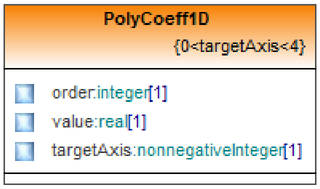
3.1.19 Class Polynomial1D

diagram	
owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::Polynomial1D abstract false
ownedMember	**coeff**
general	**Xform1D**
shown on diagram	**FrameTransforms**
documentation	Polynomial transformation from a 1-dimensional coordinate value: targetValue = sum (value[i] * nativeValue**order[i]), where the scalars and exponents are provided by the PolyCoeff1D objects. The targetValue may be n-dmensional (0<n<4); its axis numbers are also included in the PolyCoeff1D object.

Property **Polynomial1D::coeff**

owner	**Polynomial1D**
properties	qualified name stc2_transforms::frametransforms::Polynomial1D::coeff multiplicity 1..* type PolyCoeff1D aggregation composite

3.1.20 Class PolyCoeff1D

diagram	 PolyCoeff1D {0<targetAxis<4} order:integer[1] value:real[1] targetAxis:nonnegativeInteger[1]
owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::PolyCoeff1D abstract false
ownedMember	**Constraint1** **order** **targetAxis** **value**
typedElements	Class **Polynomial1D** Property **coeff**
shown on diagram	**FrameTransforms**
constraints	Constraint1: 0<targetAxis<4
documentation	PolyCoeff1D contains the parameters for one individual term in a polynomial transformation from a 1-dimensional coordinate to an n-dimensional one (0<n<4); the target axis index is provided by attribute targetAxis.

Constraint **PolyCoeff1D::Constraint1**

owner	**PolyCoeff1D**
properties	qualified name stc2_transforms::frametransforms::PolyCoeff1D::Constraint1 specification 0<targetAxis<4 constrained elements PolyCoeff1D

Property **PolyCoeff1D::order**

owner	**PolyCoeff1D**
properties	qualified name stc2_transforms::frametransforms::PolyCoeff1D::order multiplicity 1 type integer

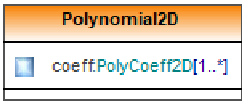
Property **PolyCoeff1D::targetAxis**

owner	**PolyCoeff1D**
properties	qualified name stc2_transforms::frametransforms::PolyCoeff1D::targetAxis multiplicity 1 type nonnegativeInteger

Property **PolyCoeff1D::value**

owner	**PolyCoeff1D**
properties	qualified name stc2_transforms::frametransforms::PolyCoeff1D::value multiplicity 1

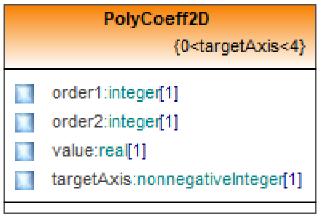
3.1.21 Class Polynomial2D

diagram	
owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::Polynomial2D abstract false
ownedMember	**coeff**
general	**Xform2D**
shown on diagram	**FrameTransforms**
documentation	2-dimensional to n-dimensional polynomial transformation: targetValue = sum (value[i] * nativeValue[1]**order1[i] * nativeValue[2]**order2[i]), where the scalars and exponents are provided by the PolyCoeff2D objects. The targetValue may be n-dmensional (0<n<4); its axis numbers are also included in the PolyCoeff1D object.

Property **Polynomial2D::coeff**

owner	**Polynomial2D**
properties	qualified name stc2_transforms::frametransforms::Polynomial2D::coeff multiplicity 1..* type PolyCoeff2D aggregation composite

3.1.22 Class PolyCoeff2D

diagram	 PolyCoeff2D {0<targetAxis<4} order1:integer[1] order2:integer[1] value:real[1] targetAxis:nonnegativeInteger[1]
owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::PolyCoeff2D abstract false
ownedMember	**Constraint1** **order1** **order2** **targetAxis** **value**
typedElements	Class **Polynomial2D** Property **coeff**
shown on diagram	**FrameTransforms**
constraints	Constraint1: 0<targetAxis<4
documentation	PolyCoeff2D contains the parameters for one individual term in a polynomial transformation from a 2-dimensional coordinate to an n-dimensional one (0<n<4); the target axis index is provided by attribute targetAxis.

Constraint **PolyCoeff2D::Constraint1**

owner	**PolyCoeff2D**
properties	qualified name stc2_transforms::frametransforms::PolyCoeff2D::Constraint1 specification 0<targetAxis<4 constrained elements PolyCoeff2D

Property **PolyCoeff2D::order1**

owner	**PolyCoeff2D**
properties	qualified name stc2_transforms::frametransforms::PolyCoeff2D::order1 multiplicity 1 type integer

Property **PolyCoeff2D::order2**

owner	**PolyCoeff2D**
properties	qualified name stc2_transforms::frametransforms::PolyCoeff2D::order2 multiplicity 1 type integer

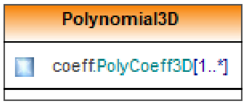
Property **PolyCoeff2D::targetAxis**

owner	**PolyCoeff2D**
properties	qualified name stc2_transforms::frametransforms::PolyCoeff2D::targetAxis multiplicity 1 type nonnegativeInteger

Property **PolyCoeff2D::value**

owner	**PolyCoeff2D**
properties	qualified name stc2_transforms::frametransforms::PolyCoeff2D::value multiplicity 1 type real

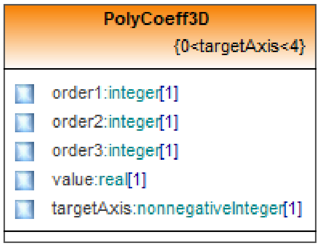
3.1.23 Class Polynomial3D

diagram	
owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::Polynomial3D abstract false
ownedMember	**coeff**
general	**Xform3D**
shown on diagram	**FrameTransforms**
documentation	3-dimensional to n-dimensional polynomial transformation: targetValue = sum (value[i] * nativeValue[1]**order1[i] * nativeValue[2]**order2[i] * nativeValue[3]**order3[i]), where the scalars and exponents are provided by the PolyCoeff3D objects. The targetValue may be n-dmensional (0<n<4); its axis numbers are also included in the PolyCoeff1D object.

Property **Polynomial3D::coeff**

owner	**Polynomial3D**
properties	qualified name stc2_transforms::frametransforms::Polynomial3D::coeff multiplicity 1..* type PolyCoeff3D aggregation composite

3.1.24 Class PolyCoeff3D

diagram	 PolyCoeff3D {0<targetAxis<4} order1:integer[1] order2:integer[1] order3:integer[1] value:real[1] targetAxis:nonnegativeInteger[1]
owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::PolyCoeff3D abstract false
ownedMember	**Constraint1** **order1** **order2** **order3** **targetAxis** **value**
typedElements	Class **Polynomial3D** Property **coeff**
shown on diagram	**FrameTransforms**
constraints	Constraint1: 0<targetAxis<4
documentation	PolyCoeff3D contains the parameters for one individual term in a polynomial transformation from a 3-dimensional coordinate to an n-dimensional one (0<n<4); the target axis index is provided by attribute targetAxis.

Constraint **PolyCoeff3D::Constraint1**

owner	**PolyCoeff3D**
properties	qualified name stc2_transforms::frametransforms::PolyCoeff3D::Constraint1 specification 0<targetAxis<4 constrained elements PolyCoeff3D

Property **PolyCoeff3D::order1**

owner	**PolyCoeff3D**	
properties	qualified name	stc2_transforms::frametransforms::PolyCoeff3D::order1
	multiplicity	1
	type	integer

Property **PolyCoeff3D::order2**

owner	**PolyCoeff3D**	
properties	qualified name	stc2_transforms::frametransforms::PolyCoeff3D::order2
	multiplicity	1
	type	integer

Property **PolyCoeff3D::order3**

owner	**PolyCoeff3D**	
properties	qualified name	stc2_transforms::frametransforms::PolyCoeff3D::order3
	multiplicity	1
	type	integer

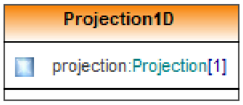
Property **PolyCoeff3D::targetAxis**

owner	**PolyCoeff3D**	
properties	qualified name	stc2_transforms::frametransforms::PolyCoeff3D::targetAxis
	multiplicity	1
	type	nonnegativeInteger

Property **PolyCoeff3D::value**

owner	**PolyCoeff3D**	
properties	qualified name	stc2_transforms::frametransforms::PolyCoeff3D::value
	multiplicity	1
	type	real

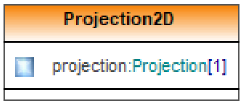
3.1.25 Class Projection1D

diagram		
owner	**frametransforms**	
properties	qualified name	stc2_transforms::frametransforms::Projection1D
	abstract	false
ownedMember	**projection**	
general	**Xform1D**	
shown on diagram	**FrameTransforms**	
documentation	The attribute projection specifies a specific standard projection transformation. One should keep in mind that most of the values of the enumeration literal Projection are intended for 2-dimensional spherical coordinate frames.	

Property **Projection1D::projection**

owner	**Projection1D**	
properties	qualified name	stc2_transforms::frametransforms::Projection1D::projection
	multiplicity	1
	type	Projection

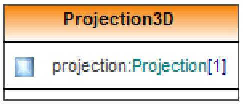
3.1.26 Class Projection2D

diagram	
owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::Projection2D abstract false
ownedMember	**projection**
general	**Xform2D**
shown on diagram	**FrameTransforms**
documentation	The attribute projection specifies a specific standard projection transformation taken from the list of values in the enumeration type Projection.

Property **Projection2D::projection**

owner	**Projection2D**
properties	qualified name stc2_transforms::frametransforms::Projection2D::projection multiplicity 1 type Projection

3.1.27 Class Projection3D

diagram	
owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::Projection3D abstract false
ownedMember	**projection**
general	**Xform3D**
shown on diagram	**FrameTransforms**
documentation	The attribute projection specifies a specific standard projection transformation taken from the list of values in the enumeration type Projection.

Property **Projection3D::projection**

owner	**Projection3D**
properties	qualified name stc2_transforms::frametransforms::Projection3D::projection multiplicity 1 type Projection

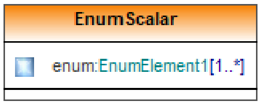
3.1.28 Enumeration Projection

diagram

Projection Code	Description
"" (blank)	Planar (i.e., linear cartesian-to-cartesian) projection
LOG	Linear-to-logarithmic cartesian-to-cartesian projection
TAN	Tangent plane projection
SIN	Sine projection
STG	Stereographic projection
ARC	Zenithal equidistant projection
ZEA	Zenithal equal-area projection
AIR	Airy projection
CEA	Cylindrical equal-area projection
CAR	Plate Carree projection
MER	Mercator projection
SFL	Sanson-Flamsteed projection
PAR	Parabolic projection
MOL	Mollweide projection
AIT	Hammer-Aitoff projection
COE	Conic equal-area projection
COD	Conic equidistant projection
COO	Conic orthomorphic projection
BON	Bonne equal-area projection
PCO	Polyconic projection
TSC	Tangential spherical cube projection
CSC	COBE quadrilateralized spherical cube projection
QSC	Quadrilateralized spherical cube projection

owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::Projection abstract false
ownedMember	**AIR AIT ARC BON CAR CEA COD COE COO CSC LIN LOG MER MOL PAR PCO QSC SFL SIN STG TAN TSC ZEA**
typedElements	Class **Projection1D** Property **projection** Class **Projection2D** Property **projection** Class **Projection3D** Property **projection**
shown on diagram	**FrameTransforms**
documentation	This is th enumerated list of allowed standard mapping (transformation) projections, largely taken from the FITS WCS list of projections.

3.1.29 Class EnumScalar

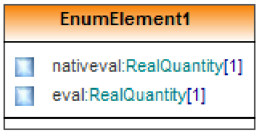
diagram	
owner	**frametransforms**

properties	qualified name stc2_transforms::frametransforms::EnumScalar abstract false
ownedMember	**enum**
general	**Xform1D**
shown on diagram	**FrameTransforms**
documentation	For enumerated coordinates EnumScalar contains the list of enumerated values in EnumElement1 that provide the target values for each of the native values. The number of EnumElement1 in the composition needs to equal the number of native elements.

Property **EnumScalar::enum**

owner	**EnumScalar**
properties	qualified name stc2_transforms::frametransforms::EnumScalar::enum multiplicity 1..* type EnumElement1 aggregation composite

3.1.30 Class EnumElement1

diagram	 EnumElement1 nativeval:RealQuantity[1] eval:RealQuantity[1]
owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::EnumElement1 abstract false
ownedMember	**eval** **nativeval**
typedElements	Class **EnumScalar** Property **enum**
shown on diagram	**FrameTransforms**
documentation	The enumerated targetValue for the enumerated nativeValue is eval.

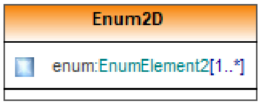
Property **EnumElement1::eval**

owner	**EnumElement1**
properties	qualified name stc2_transforms::frametransforms::EnumElement1::eval multiplicity 1 type RealQuantity

Property **EnumElement1::nativeval**

owner	**EnumElement1**
properties	qualified name stc2_transforms::frametransforms::EnumElement1::nativeval multiplicity 1 type RealQuantity

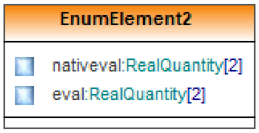
3.1.31 Class Enum2D

diagram	
owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::Enum2D abstract false
ownedMember	**enum**
general	**Xform2D**
shown on diagram	**FrameTransforms**
documentation	For enumerated coordinates Enum2D contains the list of enumerated values in EnumElement2 that provide the target values for each of the native values. The number of EnumElement2 in the composition needs to equal the number of native elements.

Property **Enum2D::enum**

owner	**Enum2D**
properties	qualified name stc2_transforms::frametransforms::Enum2D::enum multiplicity 1..* type EnumElement2 aggregation composite

3.1.32 Class EnumElement2

diagram	 EnumElement2 nativeval:RealQuantity[2] eval:RealQuantity[2]
owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::EnumElement2 abstract false
ownedMember	**eval** **nativeval**
typedElements	Class **Enum2D** Property **enum**
shown on diagram	**FrameTransforms**
documentation	The enumerated targetValue vector for the enumerated nativeValue vector is the vector eval.

Property **EnumElement2::eval**

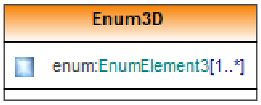
owner	**EnumElement2**
properties	qualified name stc2_transforms::frametransforms::EnumElement2::eval ordered true multiplicity 2 type RealQuantity

Property **EnumElement2::nativeval**

owner	**EnumElement2**
properties	qualified name stc2_transforms::frametransforms::EnumElement2::nativeval ordered true

	multiplicity 2 type RealQuantity

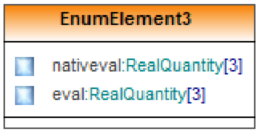
3.1.33 Class Enum3D

diagram	
owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::Enum3D abstract false
ownedMember	**enum**
general	**Xform3D**
shown on diagram	**FrameTransforms**
documentation	For enumerated coordinates Enum3D contains the list of enumerated values in EnumElement3 that provide the target values for each of the native values. The number of EnumElement3 in the composition needs to equal the number of native elements.

Property **Enum3D::enum**

owner	**Enum3D**
properties	qualified name stc2_transforms::frametransforms::Enum3D::enum multiplicity 1..* type EnumElement3 aggregation composite

3.1.34 Class EnumElement3

diagram	 EnumElement3 nativeval:RealQuantity[3] eval:RealQuantity[3]
owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::EnumElement3 abstract false
ownedMember	**eval** **nativeval**
typedElements	Class **Enum3D** Property **enum**
shown on diagram	**FrameTransforms**
documentation	The enumerated targetValue vector for the enumerated nativeValue vector is the vector eval.

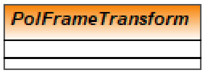
Property **EnumElement3::eval**

owner	**EnumElement3**
properties	qualified name stc2_transforms::frametransforms::EnumElement3::eval ordered true multiplicity 3 type RealQuantity

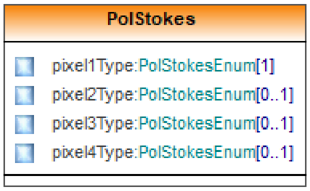
Property **EnumElement3::nativeval**

owner	**EnumElement3**
properties	qualified name stc2_transforms::frametransforms::EnumElement3::nativeval ordered true multiplicity 3 type RealQuantity

3.1.35 Class PolFrameTransform

diagram	
owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::PolFrameTransform abstract true
general	**Xform1D**
specific	**PolCircular** **PolLinear** **PolStokes** **PolVector**
shown on diagram	**FrameTransforms**
documentation	Polarization mappings are a special case of enumerated transformations, since Polarization is by definition an enumerated (scalar) coordinate. This is the abstract class.

3.1.36 Class PolStokes

diagram	 PolStokes pixel1Type:PolStokesEnum[1] pixel2Type:PolStokesEnum[0..1] pixel3Type:PolStokesEnum[0..1] pixel4Type:PolStokesEnum[0..1]
owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::PolStokes abstract false
ownedMember	**pixel1Type** **pixel2Type** **pixel3Type** **pixel4Type**
general	**PolFrameTransform**
shown on diagram	**FrameTransforms**
documentation	The enumerated mapping from pixels to Stokes polarization coordinates.

Property **PolStokes::pixel1Type**

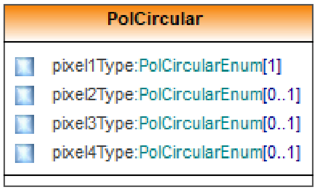
owner	**PolStokes**
properties	qualified name stc2_transforms::frametransforms::PolStokes::pixel1Type multiplicity 1 type PolStokesEnum

Property **PolStokes::pixel2Type,pixel3Type,pixel4Type**

owner	**PolStokes**
properties	qualified name stc2_transforms::frametransforms::PolStokes::pixel2Type multiplicity 0..1

	type PolStokesEnum

3.1.37 Class PolCircular

diagram	 PolCircular pixel1Type:PolCircularEnum[1] pixel2Type:PolCircularEnum[0..1] pixel3Type:PolCircularEnum[0..1] pixel4Type:PolCircularEnum[0..1]
owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::PolCircular abstract false
ownedMember	**pixel1Type** **pixel2Type** **pixel3Type** **pixel4Type**
general	**PolFrameTransform**
shown on diagram	**FrameTransforms**
documentation	The enumerated mapping from pixels to circular polarization coordinates.

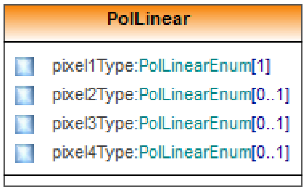
Property **PolCircular::pixel1Type**

owner	**PolCircular**
properties	qualified name stc2_transforms::frametransforms::PolCircular::pixel1Type multiplicity 1 type PolCircularEnum

Property **PolCircular::pixel2Type,pixel3Type,pixel4Type**

owner	**PolCircular**
properties	qualified name stc2_transforms::frametransforms::PolCircular::pixel2Type multiplicity 0..1 type PolCircularEnum

3.1.38 Class PolLinear

diagram	 PolLinear pixel1Type:PolLinearEnum[1] pixel2Type:PolLinearEnum[0..1] pixel3Type:PolLinearEnum[0..1] pixel4Type:PolLinearEnum[0..1]
owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::PolLinear abstract false
ownedMember	**pixel1Type** **pixel2Type** **pixel3Type** **pixel4Type**
general	**PolFrameTransform**
shown on diagram	**FrameTransforms**
documentation	The enumerated mapping from pixels to linear polarization coordinates.

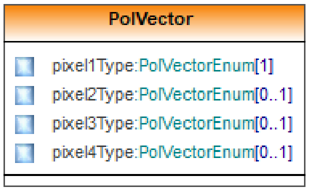
Property **PolLinear::pixel1Type**

owner	**PolLinear**
properties	qualified name stc2_transforms::frametransforms::PolLinear::pixel1Type multiplicity 1 type PolLinearEnum

Property **PolLinear::pixel2Type,pixel3Type,pixel4Type**

owner	**PolLinear**
properties	qualified name stc2_transforms::frametransforms::PolLinear::pixel2Type multiplicity 0..1 type PolLinearEnum

3.1.39 Class PolVector

diagram	 PolVector pixel1Type:PolVectorEnum[1] pixel2Type:PolVectorEnum[0..1] pixel3Type:PolVectorEnum[0..1] pixel4Type:PolVectorEnum[0..1]
owner	**frametransforms**
properties	qualified name stc2_transforms::frametransforms::PolVector abstract false
ownedMember	**pixel1Type** **pixel2Type** **pixel3Type** **pixel4Type**
general	**PolFrameTransform**
shown on diagram	**FrameTransforms**
documentation	The enumerated mapping from pixels to polarization vector coordinates.

Property **PolVector::pixel1Type**

owner	**PolVector**
properties	qualified name stc2_transforms::frametransforms::PolVector::pixel1Type multiplicity 1 type PolVectorEnum

Property **PolVector::pixel2Type,pixel3Type,pixel4Type**

owner	**PolVector**
properties	qualified name stc2_transforms::frametransforms::PolVector::pixel2Type multiplicity 0..1 type PolVectorEnum

4 Imported Coordinate Elements

Model **stc2_coordinates**

owner	**stc2_transforms**
properties	qualified name stc2_transforms::stc2_coordinates «modelimport» true namespaceURI stc2_coordinates url https://volute.g-vo.org/svn/trunk/projects/dm/STC-2.0/vo-dml/STC_coords-v2.0.vo-dml.xml
ownedMember	**CoordinateDomains** **coordsystem**

4.1 Package Coordinate System

Package **coordsystem**

owner	**stc2_coordinates**
properties	qualified name stc2_transforms::stc2_coordinates::coordsystem «modelelement» true vodml-id stc2_coordinates:coordsystem
ownedDiagrams	**CoordSystem**
ownedMember	**CoordFrame**

4.1.1 Class CoordFrame

owner	**coordsystem**
properties	qualified name stc2_transforms::stc2_coordinates::coordsystem::CoordFrame abstract true «modelelement» true vodml-id coordsystem.CoordFrame
typedElements	Class **FrameTransform** Property **nativeFrame** **targetFrame**
shown on diagram	**CoordSystem** **FrameTransforms**
documentation	This is the abstract empty base class for all coordinate frames. It is imported from the tcs2)coordinates model. A Frame SHALL contain one corresponding Coordinate Space. Note: this class is not part of this package, but imported from stc2_coordinates.

4.2 Package PolarizationDomain

Package **CoordinateDomains**

owner	**stc2_coordinates**
properties	qualified name stc2_transforms::stc2_coordinates::CoordinateDomains «modelelement» true vodml-id stc2_coordinates:CoordinateDomains
ownedMember	**polarizationDomain**

Package **polarizationDomain**

owner	**CoordinateDomains**
properties	qualified name stc2_transforms::stc2_coordinates::CoordinateDomains::polarizationDomain «modelelement» true vodml-id stc2_coordinates:CoordinateDomains.polarizationDomain
ownedDiagrams	**PolarizationDomain**
ownedMember	**PolCircularEnum** **PolEnum** **PolLinearEnum** **PolStokesEnum** **PolVectorEnum**

4.2.1 Enumeration PolEnum

owner	**polarizationDomain**
properties	qualified name stc2_transforms::stc2_coordinates::CoordinateDomains::polarizationDomain::PolEnum abstract false «modelelement» true vodml-id CoordinateDomains.polarizationDomain.PolEnum
specific	**PolCircularEnum** **PolLinearEnum** **PolStokesEnum** **PolVectorEnum**
shown on diagram	**PolarizationDomain**
documentation	PolEnum contains all polarization coordinate values. The values for each of the four types of polarization coordinates are contained in separate derived (by restriction) enumeration types. Note: this enumerator is not part of this package, but imported from stc2_coordinates.

4.2.2 Enumeration PolStokesEnum

owner	**polarizationDomain**
properties	qualified name stc2_transforms::stc2_coordinates::CoordinateDomains::polarizationDomain::PolStokesEnum abstract false «modelelement» true vodml-id CoordinateDomains.polarizationDomain.PolStokesEnum
general	**PolEnum**
typedElements	Class **PolStokes** Property **pixel1Type** **pixel2Type** **pixel3Type** **pixel4Type**
shown on diagram	**PolarizationDomain**
documentation	The four Stokes coordinate values for a polarization coordinate. Note: this enumerator is not part of this package, but imported from stc2_coordinates.

4.2.3 Enumeration PolCircularEnum

owner	**polarizationDomain**
properties	qualified name stc2_transforms::stc2_coordinates::CoordinateDomains::polarizationDomain::PolCircularEnum abstract false «modelelement» true vodml-id CoordinateDomains.polarizationDomain.PolCircularEnum
general	**PolEnum**
typedElements	Class **PolCircular** Property **pixel1Type** **pixel2Type** **pixel3Type** **pixel4Type**
shown on diagram	**PolarizationDomain**
documentation	The four circular polarization coordinate values for a polarization coordinate. Note: this enumerator is not part of this package, but imported from stc2_coordinates.

4.2.4 Enumeration PolLinearEnum

owner	**polarizationDomain**
properties	qualified name stc2_transforms::stc2_coordinates::CoordinateDomains::polarizationDomain::PolLinearEnum abstract false «modelelement» true vodml-id CoordinateDomains.polarizationDomain.PolLinearEnum
general	**PolEnum**
typedElements	Class **PolLinear** Property **pixel1Type** **pixel2Type** **pixel3Type** **pixel4Type**

shown on diagram	**PolarizationDomain**
documentation	The four linear polarization coordinate values for a polarization coordinate. Note: this enumerator is not part of this package, but imported from stc2_coordinates.

4.2.5 Enumeration PolVectorEnum

owner	**polarizationDomain**
properties	qualified name stc2_transforms::stc2_coordinates::CoordinateDomains::polarizationDomain::PolVectorEnum abstract false «modelelement» true vodml-id CoordinateDomains.polarizationDomain.PolVectorEnum
general	**PolEnum**
typedElements	Class **PolVector** Property **pixel1Type** **pixel2Type** **pixel3Type** **pixel4Type**
shown on diagram	**PolarizationDomain**
documentation	The four polarization vector coordinate values for a polarization coordinate. Note: this enumerator is not part of this package, but imported from stc2_coordinates.