LOFAR Data Format ICD
Representations of World Coordinates

Document ID: LOFAR-USG-ICD-002

Version 2.05.06

SVN Repository Revision: 9369

L. Bähren, A. Alexov, K. Anderson, J.-M. Grießmeier

SVN Date: 2012-01-10

Contents

Change record

VERSION	DATE	SECTIONS	DESCRIPTION OF CHANGES
0.1	2010-06-02	all	Document creation
0.2	2010-06-04	3.2, 3.3, 4	Imported coordinate description tables from the other ICDs. — Sec. **??**: Basic description of the coordinates group, including examples. — Sec. **??**: Started tracking list of pending issues.
0.3	2010-06-06	all	Collection of examples. Removed *Time* and *Length* coordinate, renamed *Frequency* to *Spectral*. Moved description of "Coordinates Group" to the begin of section **??**. Start filling section **??**.
0.4	2010-06-07	all	Added references on FITS standard and representation of physical units. New section "Specification of Units". Added table with the set of recognized Polarization values (Tab. **??**).
0.5	2010-06-08	all	Including comments by JM. Included basic conversion process figures from [**?**, **?**]. Included common glossary of terms.
0.6	2010-06-09	all	Reorganization of sections; dropping previous distinction between storage containers and physical coordinates – this might be more considered an issue of implementation. Added extra section to review some of the basic concepts as presented in [**?**, **?**, **?**]
0.7	2010-06-16	all	New table with codes and parameters for spherical map projections (Tab. **??**). Renamed keyword: `SYSTEM_RADEC` → `RADEC_SYS`; new table with allowed values for `RADEC_SYS` (Tab. **??**). Moved specification of units to Sec. **??**. Imported comments from Jean-Mathias and Anastasia.
0.8	2010-06-30	**??**	Filling in description for `AXIS_NAMES`.
2.00.00	2010-07-08	Cover	Changed 'revision' to 'version'; updated this version number to 2.00.00 for LOFAR ICDs 1 through 7 to put them on the same version numbering scheme.
2.00.01	2010-10-26	**??**	Added description and storage ordering specification for `PC` linear transformation matrix.
2.00.02	2010-10-27	**??**	Splitting off sections describing the individual coordinates from the main document source, in order allow import into all of the ICDs.
2.00.03	2010-11-15	**??**	Update table describing the attributes attached to coordinate groups
2.00.04	2010-11-19	**??**	Updating table describing attributes attached to the group storing a spectral coordinate
2.01.00	2010-11-23	**??**	New section on the *Separation between physical interpretation and storage mechanism*, laying out the reasoning between establishing two groups of coordinate containers.
2.01.01	2010-11-29	**??**	Applying separation between physical quantity and storage container to Stokes coordinates – now internally using Tabular coordinate for storage.
2.01.02	2010-11-30	all	Using LaTeX package `hyperref` for references, enabling better navigation through the document and access to external resources.

continued on next page

Change record *continued from previous page*

Version	Date	Sections	Description of changes
2.02.0	2010-11-30	??	Updating attributes attached to Stokes coordinate; corrected earlier error in how to store information regarding the Stokes components.
2.02.01	2010-12-06	Changes	Added note on version numbering scheme.
2.03.00	2010-12-07	??	Cleaning up of external sources containing specification of the individual coordinates. New section on time coordinate. Cleaned up tables with the recognized values for time and location reference frames.
2.04.00	2010-12-09	all	Correcting errors in version numbering. Added tables with recognized values for time and location reference frames. Cleaning up of attributes associated with Tabular coordinate (Sec. ??): • `PIXEL_VALUES` → `AXIS_VALUES_PIXEL` • `WORLD_VALUES` → `AXIS_VALUES_WORLD`
2.04.01	2011-03-01	all	Adding listings for hierarchical structure of groups.
2.04.02	2011-03-07	Header	Use variable for document title and list of authors; as these are inserted at a number of places it makes sence to define them once and then reuse the information.
2.04.03	2011-03-10	all	Maintain list of references through BibLaTeX database.
2.05.00	2011-03-11	all	Renamed *Stokes coordinate* → *Polarization coordinate*; updating examples for coordinates representation.
2.05.01	2011-03-30	all	Context and motivation; Update of figures; Table with spectral transformation equations.
2.05.02	2011-04-04	??	Added examples for representation of spectral coordinate.
2.05.03	2011-04-04	??	Clean-up of description for spectral coordinate.
2.05.04	2011-04-26	??	Removed obsolete attribute `CONVERSION_SYSTEM`, which was adopted from the CASA image format instead from [?]; cleanup for `EQUINOX` and `RADEC_SYS` attributes, which are part of a direction coordinate.
2.05.05	2011-05-11	all	Added paragraph with notation conventions; adjusted coordinate descriptions according to these conventions.
2.05.06	2012-01-10	title page	Changed the svnInfoRevision to svnInfoMaxRevision, in order to take the sub-tex file changes into account for the latex compile.

Version numbering scheme In order to track the evolution of the format specification documents the following numbering scheme has been adopted:

```
<major version>.<minor version>.<patch version>
  [0..]         .  [0..99]       .  [0..99]
```

where

- the `<patch version>` is getting incremented on changes to the document, which do not affect the actual contents of the file (such as when changing attribute names and such), e.g. correcting/augmenting descriptions, adding examples, etc.
- The `<minor version>` tracks minor changes to the actual content of the file, such as renaming, adding or removing attributes.
- The `<major version>` indicates major changes with in the file format, such as reorganization of the internal hierarchical structure or official release to the public.

Notation.

Symbol	Description
a, A	Italic lower and upper case chracters denote scalars.
$\mathbf{a}$	Bold lower case characters denote column vectors.
$\mathbf{A}_{[L,M]}$	Bold upper case characters denote matrices; (optional) if given $[L, M]$ denotes the shape.
a_i	Element i from vector $\mathbf{a}$.
A_{ij}	Element (i, j) from matrix $\mathbf{A}$.
$[name_0] \equiv$ `['Time']`	Array of rank 1, storing a single string-type value

Acknowledgements

1 Introduction

1.1 Purpose and scope

This document sets forth a formal data interface specification for LOFAR data products. The specification applies to data structures produced by various LOFAR processing pipelines that will be called COORDINATES GROUP. This is a specification for COORDINATES GROUP data products only and in no way implies, and should not be inferred as, a specification for any data structures the project may use during *in situ* processing by way of producing a final standard COORDINATES GROUP.

This document is intended to be the formal interface control agreement between the LOFAR project, observers/users of LOFAR data products, and the eventual LOFAR science archive facility.

1.2 Context and motivation

Already at a rather early stage in the discussion on requirements for the storage of LOFAR data produts it was realized, that existing data formats would not suffice in dealing with the expected volume and complexity of the data as being generated by LOFAR. With datasets growing to sizes in the multi-Terabyte regime (see e.g. [?]), solutions such as the Flexible Image Transport System (FITS) [?] would now longer scale and deliver the needed performance. Also – as to some some degree alluded to by the name itself – FITS very much is geared towards the storage of image data (though not restricted to it); given the fact that LOFAR will be generating a wide range of data products to be delivered to the scientific community, a very flexible data model is required, which allows for the representation of the complex system configuration for an individual observation leading up to the exported data product.

While the other ICDs [?, ?, ?, ?, ?, ?] describe hierarchical storage structures for data generated by subsystems or scientific pipelines of the LOFAR systems, this ICD concentrates on defining how to represent and store a specific type of metadata: world coordinates. By WORLD COORDINATES, we mean coordinates that serve to locate a measurement in some multidimensional parameter space. Coordinates include, for example, a measurable quantity such as the frequency or wavelength associated with a point in a spectrum, or more abstractly, the longitude and latitude in a conventional spherical coordinate system which define a direction in space. World coordinates may also include enumerations such as “Stokes parameters”, which do not form an image axis in the normal sense interpolation along such axes is not meaningful.

While the issue of representing coordinate information has been convered extensively for the FITS format (see references [?, ?, ?], which have been adopted as part of the FITS standard itself), no comparable description is available for other formats – especially not for the HDF5 file format [?, ?] as adopted for the LOFAR telescope. The main aim of this document therefore is to describe and establish a standard for the encapsulation and representation of world coordinates as part of the data format specifications.

1.2.1 Applicable documents

Table ?? lists all the LOFAR ICDs. Most of the ICDs are for the various LOFAR data types, while ICD numbers 002 and 005 are general and applicable to all the data-format-oriented ICDs. Please note that the data and header information is written in Little-endian format within the HDF5 files.

Reference	Title	Description
ICD-001 [?]	TBB Time-Series Data	Digitized voltage output, as received by the individual LOFAR dipoles.
ICD-002 [?]	Representations of World Coordinates	Definition of how to represent and store meta-data that serve to locate a measurement in some multidimensional parameter space.

continued on next page

Applicable documents *continued from previous page*

Reference	Title	Description
`ICD-003` [?]	Beam-Formed Data	Hosting structure for LOFAR Beam-Formed data.
`ICD-004` [?]	Radio Sky Image Cubes	Primary data product of the imaging pipeline.
`ICD-005` [?]	File Naming Conventions	Conventions for the naming scheme applied to LOFAR standard data products.
`ICD-006` [?]	Dynamic Spectrum Data	Hosting structure for dynamic spectrum data, i.e. intensity as function of time and frequency.
`ICD-007` [?]	Visibility Data	Hosting structure for LOFAR UV Visibility data, primary output of interferometer operations.
`ICD-008` [?]	RM Synthesis Cubes	Hosting structure for LOFAR Rotation Measure Synthesis Cubes output data.

Table 1.1: List of all the LOFAR Interface Control Documents. ICDs 001, 003, 004, 006, 007 and 008 describe different LOFAR data formats, while ICDs 002 and 005 are general and applicable to add the other ICDs.

2 Overview

[**Comment:**
Provide a basic overview of the document, its internal organisation and the overall goal it is supposed to fullfil.]

3 Organization of the data

3.1 High-level structure of the coordinates representation

[**Comment:**
Provide some basic overview of how the creation of a coordinates group is motivated; provide figures showing example layout of coordinate groups and examples how the coordinates group is embedded into the various LOFAR standard data products [?, ?].]

3.2 Overview of coordinate groups

When comparing the representation of World Coordinates with the data models for the LOFAR Standard data products (see [?, ?, ?, ?, ?]) the main difference is, that the present ICD describes a data structure – or actually metadata structure – which can reside at any hierarchical level of any of the other data structures (at least to the degree as this is technically allowed by the chosen implementation).

1. **Coordinates Group** (Sec. ??) In our data model this is the top-level container for coordinate-related metadata. A coordinates group will contain one or more coordinate objects, which together define a coordinate system attached to the actual data.
2. **(Primary) Storage containers**, which serve as underlying building blocks to store coordinate information.
   a) **Direction Coordinate** (Sec. ??)
   b) **Linear Coordinate** (Sec. ??)
   c) **Tabular Coordinate** (Sec. ??)
3. **Composite containers** provide a representation of world coordinates which can be represented by
   a) **Time Coordinate** (Sec. ??)
   b) **Spectral Coordinate** (Sec. ??) defines the parameters and conventions needed to specify spectral information including frequency, wavelength and velocity.
   c) **Polarization Coordinate** (Sec. ??)

4 Detailed data specification

4.1 Basic concepts

4.2 WCS-Formalism

As explained in [?, ?], the conversion of pixel coordinates to world coordinates is regarded as a multi-step process; this is shown conceptually in Fig. ??.

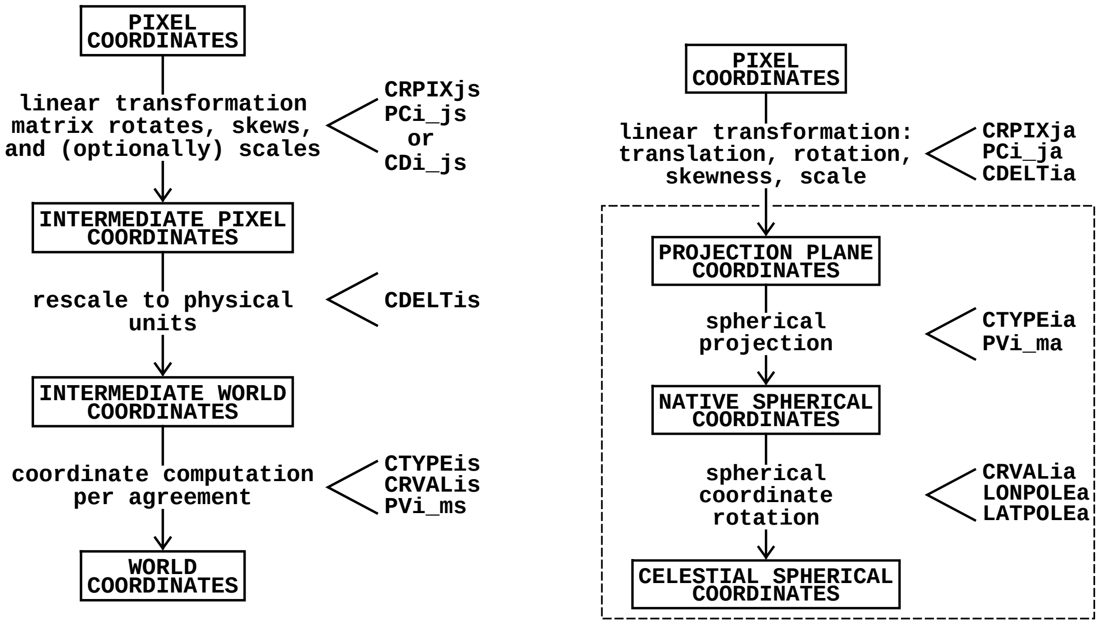


Figure 4.1: Conversion of pixel coordinates to world coordinates shown as a multi-step process. (left) In the first step a linear transformation is applied via matrix multiplication of the pixel coordinate vector. This linear transformation may be restricted to the geometrical effects of rotation and skewness with scaling to physical units deferred until the second step (PCi_j plus CDELTi formalism). Alternatively, scaling may be applied via the matrix with the second step omitted (CDi_j formalism). The final step applies a possibly non-linear transformation to produce the final world coordinates. Although generic keywords for this step are defined in this paper, the mathematical details, including the interpretation of the INTERMEDIATE WORLD COORDINATES, are deferred to later papers which may also interpose additional steps in the algorithm chain. (right) Conversion of pixel coordinates to celestial coordinates. The INTERMEDIATE WORLD COORDINATES of figure on the left are here interpreted as PROJECTION PLANE COORDINATES, i.e. Cartesian coordinates in the plane of projection, and the multiple steps required to produce them have been condensed into one.

For all coordinate types, the first step is a linear transformation applied via matrix multiplication to the vector of *pixel coordinate* elements, p_j:

$$q_i = \sum_{j=1}^{N} M_{ij} (p_j - r_j) \tag{4.1}$$

where r_j are the pixel coordinate elements of the reference point given by the `REFERENCE_PIXEL`. Henceforth we will use j for pixel axis indexing and i for the world axes. The M_{ij} matrix is a non-singular square matrix of dimensions $N \times N$. The elements, q_i, of the resulting *intermediate pixel coordinate* vector are offsets, in dimenionless pixel units, from the reference point along axes coincident with thosewith those of the *intermediate world coordinates*. Thus the conversion of q_i to the corresponding intermediate world coordinate element, x_i, is a simple scale:

$$x_i = s_i q_i \tag{4.2}$$

In the `PC` formalism, the matrix elements M_{ij} are encoded through the `PC` attribute and the s_i as `INCREMENT`. The default values for M_{ij} are

$$M_{ij} = \begin{cases} 1.0 & i = j \\ 0.0 & i \neq j \end{cases} \tag{4.3}$$

The `PC` matrix must not be singular; it must have an inverse.

4.2.1 Specification of units

Unless agreed otherwise, units should conform with the recommendations of the IAU Style Manual [?], though rather appearing in plain character form [?, ?] instead using the notation typically used in a published paper. An overview of the encoding of the basic units is shown on Tab. ?? below.

> **Comment:**
> Check definitions in table against table from [?], p. 106.

Quantity	Unit string	Meaning
length	`m`	meter
mass	`kg`	kilogram
time	`s`	second
plane angle	`rad`	radian
solid angle	`sr`	steradian
temperature	`K`	kelvin
electric current	`A`	ampere
amount of substance	`mol`	mole
luminosity intensity	`cd`	candela

Table 4.1: IAU-recommended basic units (table adopted from [?]).

4.2.2 Separation between physical interpretation and storage mechanism

Both from a technical and a conceptual point of it makes sense to separate the physical interpretation of a coordinate from the underlying storage mechanism (i.e. the container used to hold the metadata). In order to illustrate the motivation for this type of abstraction, consider coordinate information required for the various types of data products as listed in Tab. ?? below:

- ICD-003 (Beam-Formed Data) records stokes values as function of time and frequency

  $$I = I(t, \nu)$$

  Due to the fact that the frequency values are spread across multiple frequency band, the coordinate axis is non-contiguous, thereby requiring storage of the frequencies in tabulated form; as a result of this the following combination or basic storage containers is employed:

IMAGE	ICD	QUANTITY	AXES	UNITS
TBB time-series	001 / [?]	$I(t)$	Time	s
BF data	003 / [?]	$I(p,\nu,\mathrm{Dec},\mathrm{RA})$	Pol/Freq/Dir/Dir	.. /Hz/deg/deg
Sky image	004 / [?]	$I(p,\nu,\mathrm{Dec},\mathrm{RA})$	Pol/Freq/Dir/Dir	.. /Hz/deg/deg
Dyn. Spectrum	006 / [?]	$I(p,\nu,t)$	Pol/Freq/Time	.. /Hz/s
RMSC	008 / [?]	$DF(p,\mathrm{Dec},\mathrm{RA},\phi)$	Pol/Dir./Dir./Faraday Depth	.. /deg/deg/rad m^{-2}
RM map	—	$RM(\mathrm{Dec},\mathrm{RA})$	Dir./Dir.	/deg/deg
CR image	—	$I(p,\nu,r,\mathrm{El},\mathrm{Az})$	Pol/Freq/Dist/Dir./Dir./	.. /p/Hz/m/deg/deg
CR image	—	$I(p,t,\nu,\xi_3,\xi_2,\xi_1)$	Pol/Time/Freq/Pos/Pos/Pos	.. /s/Hz/m/m/m

Table 4.2: Overview of the various data arrays types, associated coordinates and dimensions. Where possible a reference for the data format specification is provided.

```
.
|- Linear
`- Tabular
```

However in order to allow easier interpretation of the coordinates in terms of the encoded physical quantities, the following seems more favorable:

```
.
|- Time
`- Spectral
```

- ICD-004 (Radio Sky Image Cube) records data in the form

$$I = I(p,\nu,\mathrm{Dec},\mathrm{RA})$$

which translates into the following set of physical coordinates:

```
.
|- Polarization        {Tabular}
|- Spectral            {Linear|Tabular}
`- Direction           {Direction}
```

Depending on the character of the spectral axis, the internal representation can be done either using a linear or a tabular coordinate.

- ICD-008 (Rotation Measure Synthesis Cube) records data in the form

$$I = I(p,\phi,\mathrm{Dec},\mathrm{RA})$$

which translates into the following set of physical coordinates:

```
.
|- Polarization
|- FaradayDepth
`- Direction
```

Depending on the distribution of the Faraday depth values, the internal representation can be done either using a linear or a tabular coordinate:

```
.                                 .
|- Polarization                   |- Polarization
|- Linear                         |- Tabular
`- Direction                      `- Direction
```

- Consider the possible representations for the coordinates attached to a (total intensity) dynamic spectrum:

```
.                   .                   .                    .
`- Linear  [2]      |- Linear   [1]     |- Linear   [1]      |- Tabular  [1]
                    `- Linear   [1]     `- Tabular  [1]      `- Tabular  [1]
```

All of the above are valid representation, given how the values along the coordinate axes are distributed. On the other hand looking at this from the perspective of the physical quantities to be described,

```
.
|- Time         [1]
`- Spectral     [1]
```

it becomes clear that separating the underlying storage structure from the physical interpretation results in a much clearer and unified picture.

4.3 Coordinates Group

The Coordinates Group acts as a container to take up a collection of coordinates, as described in the subsequent sections below. Besides this function as a container – grouping together embedded coordinate objects – the Coordinates Group also provides basic reference frame information, which is required for the proper transformation of quantities to other reference systems.

Field/Keyword	Type	Value	Description
GROUPTYPE	string	'Coordinates'	Group type descriptor
REF_LOCATION_VALUE	array<double,1>		Numerical value(s) of the reference location
REF_LOCATION_UNIT	array<string,1>		Physical unit(s) for the reference location
REF_LOCATION_FRAME	string		Identifier for the reference system of the location
REF_TIME_VALUE	double		Numerical value of the reference time
REF_TIME_UNIT	string		Physical unit of the reference time
REF_TIME_FRAME	string		Identifier for the reference time system used
NOF_COORDINATES	int	N_{Coord}	Number of embedded coordinate groups
NOF_AXES	int	$N_{\text{Axes}} = \sum_n^{N_{\text{Coord}}} N_{n,\text{Axes}}$	Cummulative number of coordinate axes, as from adding up the coordinate axes of the embedded coordinate objects/groups.
COORDINATE_TYPES	array<string,1>		Coordinate types of the embedded coordinates.
COORDINATE_{N}	Group		coordinate object container

Table 4.3: Components of a Coordinates group.

```
.
`- COORDINATES                Group
   |- GROUPTYPE               Attr.     string
   |- REF_LOCATION_VALUE      Attr.     array<double,1>
   |- REF_LOCATION_UNIT       Attr.     array<string,1>
   |- REF_LOCATION_FRAME      Attr.     array<string,1>
   |- REF_TIME_VALUE          Attr.     double
   |- REF_TIME_UNIT           Attr.     string
   |- REF_TIME_FRAME          Attr.     string
```

```
|- NOF_COORDINATES          Attr.     int
|- NOF_AXES                 Attr.     int
|- COORDINATE_TYPES         Attr.     array<string,1>
|- COORDINATE_0             Group
|    ...
`- COORDINATE_{N}           Group
```

Comment:
Do we need description of the quantity to which the coordinates are attached to (see e.g. FITS keywords 'BUNIT' and 'BSCALE')?

- GROUPTYPE is the group type descriptor with the fixed value 'Coordinates'.
- Specification of the reference frame/system within which the location is recorded is done through the combination of REF_LOCATION_VALUE, REF_LOCATION_UNIT and REF_LOCATION_FRAME; recognized values for the specification of the reference frame are listed in Tab. ?? below.

Reference Position	Description	Comments
GEOCENTER	Center of the Earth.	
BARYCENTER	Center of the solar system barycenter.	
HELIOCENTER	Center of the Sun.	
TOPOCENTER	"Local"; in most cases this will mean: the location of the telescope.	
LSRK	Kinematic Local Standard of Rest: 20 km s^{-1} in the direction of GALACTIC_II $(56, +23)$.	Only to be used for redshifts and Doppler velocities, and spectral coordinate.
LSRD	Dynamic Local Standard of Rest: 16.6 km s^{-1} in the direction of GALACTIC_II $(53, +25)$.	
GALACTIC	Center of the Galaxy: 220 km s^{-1} in the direction of GALACTIC_II $(90, 0)$ w.r.t. LSRD.	
LOCAL_GROUP	Center of the Local Group: 300 km s^{-1} in the direction of GALACTIC_II $(90, 0)$ w.r.t. BARYCENTER.	
RELOCATABLE	Relocatable center; for simulations.	Only to be used for spatial coordinates.

Table 4.4: Recognized values for the reference frame to specify a location; values and descriptions have been adopted from the "Space-Time Coordinate Metadata for the Virtual Observatory" [?], as produced by the IVOA Data Model Working Group.

- Specification of the reference frame/system within which the time/epoch is recorded is done through the combination of REF_TIME_VALUE, REF_TIME_UNIT and REF_TIME_FRAME; recognized values for the specification of the reference frame are listed in Tab. ?? below.

  For the SI-based time scales, the event tagged 1977 January 1, 00:00:00 TAI (JD 2443144.5 TAI) at the geocenter is special. At that event, the time scales TT, TCG, and TCB all read 1977 January 1, 00:00:32.184 (JD 2443144.5003725). (The $32^s.184$ offset is the estimated difference between TAI and the old Ephemeris Time scale.) This event will be designated t_0 in the following; it can be represented in any of the time scales, and the context will dictate which time scale is appropriate.

  Comment:
  Get reference for definition of time reference frames.

Time	Description
GAST	Greenwich Apparent Sidereal Time
GMST	Greenwich Mean Sidereal Time
LAST	Local Apparent Sidereal Time
LMST	Local Mean Sidereal Time
TAI	International Atomic Time
TCB	Barycentric Coordinate Time
TCG	Geocentric Coordinate Time
TDB	Barycentric Dynamical Time
TT	Terrestrial Time
UT1	Universal Time (affected by variations in length of day)
UTC	Coordinated Universal Time (an atomic tim scale)

Table 4.5: Recognized values for the reference frame to specify a time; descriptions adopted from [?]

From the perspective of a user, the starting point for computing all the time scales is Coordinated Universal Time (UTC). From UTC, we can immediately get International Atomic Time (TAI):

$$\mathrm{TAI} = \mathrm{UTC} + \Delta\mathrm{AT}$$

where ΔAT, an integral number of seconds, is the accumulated number of leap seconds applied to UTC.

- Since the coordinates group acts as a container for multiple coordinate (objects), `NOF_COORDINATES` accounts for the number of such coordinates.
- Since a coordinate can be composed of multiple axes (e.g. a Direction Coordinate consists of two direction angles), `NOF_AXES` accounts for the total number of cordinates axes.
- `COORDINATE_TYPES`

4.4 Basic storage containers

4.4.1 Direction coordinate

The Direction Coordinate consists of a set of two coupled coordinate axes, describing a direction in space; it therefore includes information such as the equinox of the observation, the system of equatorial coordinates on the sphere of the sky, as well as parameters for the spherical map projection.

- `GROUPTYPE` is the group type descriptor with the fixed value `'DirectionCoord'`.
- `COORDINATE_TYPE` is the is the descriptor for the coordinate type, of value `'Direction'`.
- `STORAGE_TYPE` is the descriptor for the underlying storage type for this coordinate, of value `'Direction'`.
- `NOF_AXES` is the number of coordinate axes; keep in mind that a coordinate can consist of multiple axes. For the the DirectionCoordinate we have `NOF_AXES=2`.
- `AXIS_NAMES` are the world axis names connected with the coordinate axes,most commonly

  `AXIS_NAME=['Longitude','Latitude']`
- `AXIS_UNITS` are the physical units along each coordinate axis (corresponding to the FITS keyword `CUNITn`, see [?]). Restrictions on the nature and range of units, if any, will be determined by agreements applying to the specific axis. If they are not so limited, units should conform to the IAU Style Manual [?].

Field/Keyword	Type	Value	Description
GROUPTYPE	string	'DirectionCoord'	Group type descriptor
COORDINATE_TYPE	string	'Direction'	Coordinate Type descriptor
STORAGE_TYPE	string	'Direction'	Descriptor for the underlying storage type for this coordinate
NOF_AXES	int	$N \equiv 2$	Number of coordinate axes
AXIS_NAMES	array<string,1>	$[name_0, name_1]$	World axis names
AXIS_UNITS	array<string,1>	$[unit_0, unit_1]$	Physical units along each coordinate axis.
REFERENCE_VALUE	array<double,1>	$[val_0, val_1]$	Coordinate value at the reference point
REFERENCE_PIXEL	array<double,1>	$[pix_0, pix_1]$	Array location of the reference point in pixels.
INCREMENT	array<double,1>	$[incr_0, incr_1]$	Coordinate increment at reference point.
PC	array<double,1>	$[pc_{00}, pc_{01}, pc_{10}, pc_{11}]$	Non-singular square matrix, for the transformation from intermediate pixel coordinates to intermediate world coordinates.
EQUINOX	string		Equinox of the observation
RADEC_SYS	string		System of equatorial coordinates
PROJECTION	string		Spherical map projection
PROJECTION_PARAM	array<double,1>		Spherical projection parameters
LONPOLE	double		Native longitude of the celestial pole, ϕ_p
LATPOLE	double		Native latitude of the celestial pole, θ_p.

Table 4.6: Attributes/keywords attached to a group describing a direction coordinate.

- REFERENCE_VALUE is the coordinate value at the reference point (corresponding to the FITS keyword CRVALn, see [?]).
- REFERENCE_PIXEL is the array location of the reference point in pixels (corresponding to the FITS keyword CRPIXn, see [?]).
- INCREMENT is the coordinate increment at the reference point (corresponding to the FITS keyword CDELTn, see [?]).
- PC is a non-singular square matrix, for the transformation from intermediate pixel coordinates to intermediate world coordinates. The individual matrix elements are stored as a linear array, ordered as follows:

$$\mathbf{M}_{[N,N]} = \begin{pmatrix} M_{00} & M_{01} & ... & M_{0N} \\ M_{10} & M_{11} & ... & M_{1N} \\ \vdots & & & \vdots \\ M_{N0} & & & M_{NN} \end{pmatrix} \rightarrow [M_{00}, M_{01}, ..., M_{10}, M_{11}, ..., M_{N0}, ..., M_{NN}]$$

- EQUINOX applies to ecliptic as well as to equatorial coordinates (e.g. J2000 or B1950) of the source position.
- RADEC_SYS Several systems of equatorial coordinates (right ascension and declination) are in common use. Apart from the International Celestial Reference System (ICRS, IAU, 1984), the axes of which are by definition fixed with respect to the celestial sphere, each system is parameterized by time. In particular, mean equatorial coordinates are defined in terms of the epoch (i.e. instant of time) of the mean equator and equinox (i.e. pole and origin of right ascension). The same applies for ecliptic

```
.
`- COORDINATE_{N}           Group
   |- GROUPTYPE             Attr.       string
   |- COORDINATE_TYPE       Attr.       string
   |- STORAGE_TYPE          Attr.       string
   |- NOF_AXES              Attr.       int
   |- AXIS_NAMES            Attr.       array<string,1>
   |- AXIS_UNITS            Attr.       array<string,1>
   |- REFERENCE_VALUE       Attr.       array<double,1>
   |- REFERENCE_PIXEL       Attr.       array<double,1>
   |- INCREMENT             Attr.       array<double,1>
   |- PC                    Attr.       array<double,1>
   |- EQUINOX               Attr.       string
   |- RADEC_SYS             Attr.       string
   |- PROJECTION            Attr.       string
   |- PROJECTION_PARAM      Attr.       array<double,1>
   |- LONGPOLE              Attr.       double
   `- LATPOLE               Attr.       double
```

Listing 4.1: Structure of the direction coordinate group.

coordinate systems. The keyword `RADEC_SYS` is used to specify the particular system; recognized values are given in Tab. **??** below.

RADEC_SYS	DESCRIPTION
ICRS	International Celestial Reference System
FK5	mean place, new (IAU 1984) system
FK4	mean place, old (Bessell-Newcomb) system
FK4-NO-E	meanplace, old system but without e-terms
GAPPT	Geocentric Apparent Place, IAU 1984 system

Table 4.7: Allowed values of `RADEC_SYS`

– `PROJECTION` holds the reference code for the spherical map projection, e.g. `AIT`, `SIN`, `STG`, etc. As some of these projections require (or at least allow) additional parameters, the `PROJECTION_PARAM` keyword is used to store these additional parameters. Recognized values are given in Table **??** below.

– `LONPOLE` is the native longitude of the celestial pole, ϕ_p.

– `LATPOLE` is the native latitude of the celestial pole, θ_p.

4.4.2 Linear coordinate

As already indicated by the name, this group encodes the properties of a simple linear coordinate (or a number thereof, as multiple axes are permitted).

– `GROUPTYPE` is the group type descriptor with the fixed value `'LinearCoord'`.

– `COORDINATE_TYPE` is the is the descriptor for the coordinate type, of value `'Linear'`.

– `STORAGE_TYPE` is the descriptor for the underlying storage type for this coordinate, of value `'Linear'`.

– `NOF_AXES` is the number of coordinate axes represented by this coordinate.

– `AXIS_NAMES` are the world axis names connected with the coordinate axes, e.g.

	Projection	ϕ_0	θ_0	Projection parameters
AZP	Zenithal perspective	0°	90°	$[\mu, \gamma]$
SZP	Slant zenithal perspective	0°	90°	$[\mu, \phi_c, \theta_c]$
TAN	Gnomonic	0°	90°	
STG	Stereographic	0°	90°	
SIN	Slant orthographic	0°	90°	$[\xi, \eta]$
ARC	Zenithal equidistant	0°	90°	
ZPN	Zenithal polynomial	0°	90°	$[P_0, P_1, .., P_m]$ for $m = 0, \dots 29$
ZEA	Zenithal equal-area	0°	90°	
AIR	Airy	0°	90°	$[\theta_b]$
CYP	Cylindrical perspective	0°	0°	$[\mu, \lambda]$
CEA	Cylindrical equal area	0°	0°	$[\lambda]$
CAR	Plate carrée	0°	0°	
MER	Mercator	0°	0°	
SFL	Sanson-Flamsteed	0°	0°	
PAR	Parabolic	0°	0°	
MOL	Mollweide	0°	0°	
AIT	Hammer-Aitoff	0°	0°	
COP	Conic perspective	0°	θ_a	$[\theta_a, \eta]$
COE	Conic equal-area	0°	θ_a	$[\theta_a, \eta]$
COD	Conic equidistant	0°	θ_a	$[\theta_a, \eta]$
COO	Conic orthomorphic	0°	θ_a	$[\theta_a, \eta]$
BON	Bonne's equal area	0°	0°	$[\theta_1]$
PCO	Polyconic	0°	0°	
TSC	Tangential Spherical Cube	0°	0°	
CSC	COBE Quadrilateralized Spherical Cube	0°	0°	
QSC	Quadrilateralized Spherical Cube	0°	0°	

Table 4.8: Summary of projection codes, full name, default values of ϕ_0 and θ_0, and required parameters. Values and descriptions have been adopted from [?].

```
AXIS_NAMES=['Distance']
AXIS_NAMES=['Time']
AXIS_NAMES=['Azimuth','Elevation']
```

- `AXIS_UNITS` are the physical units along each coordinate axis (corresponding to the FITS keyword `CUNITi`, see [?]). Restrictions on the nature and range of units, if any, will be determined by agreements applying to the specific axis. If they are not so limited, units should conform to the IAU Style Manual [?].
- `REFERENCE_VALUE` is the coordinate value at the reference point (corresponding to the FITS keyword `CRVAL`n, see [?]).
- `REFERENCE_PIXEL` is the array location of the reference point in pixels (corresponding to the FITS keyword `CRPIX`n, see [?]).
- `INCREMENT` is the coordinate increment at the reference point (corresponding to the FITS keyword `CDELT`n, see [?]).
- `PC` is a non-singular square matrix, for the transformation from intermediate pixel coordinates to intermediate world coordinates. The individual matrix elements are stored as a linear array, ordered

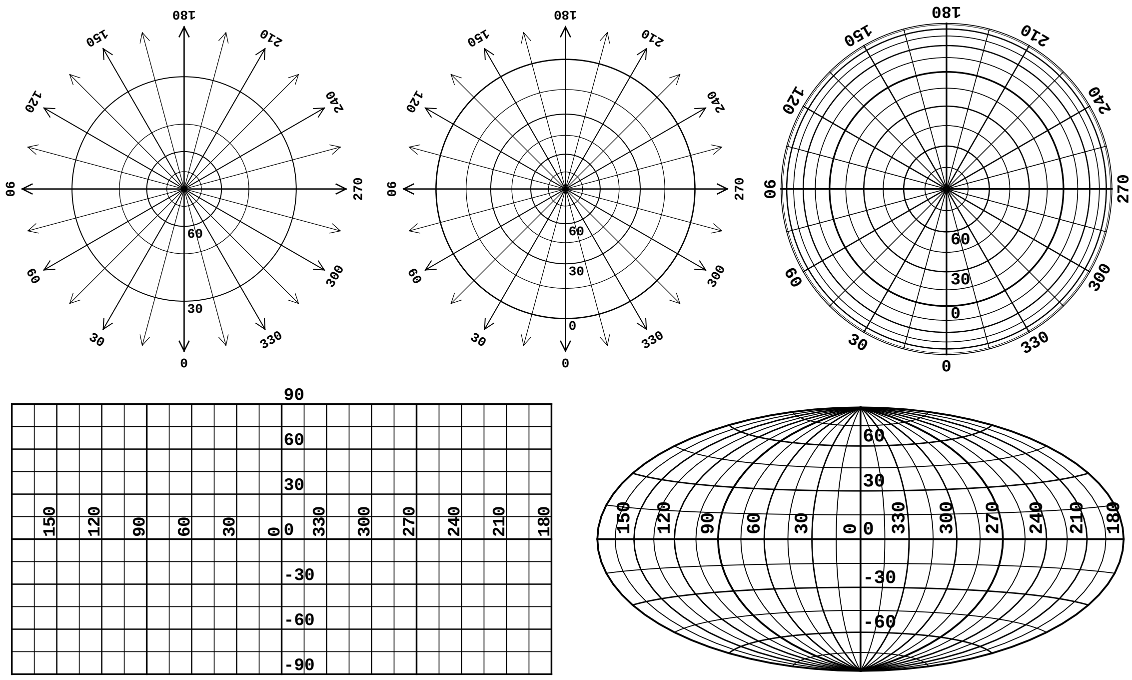


Figure 4.2: A selection of spherical map projections. Top row, from left to right: `TAN` (Gnomonic), `STG` (Stereographic), `ZEA` (Zenithal equal-area). Bottom row, from left to right: `CAR` (Plate carrée), `AIT` (Hammer-Aitoff).

as follows:

$$\mathbf{M}_{[N,N]} = \begin{pmatrix} M_{00} & M_{01} & ... & M_{0N} \\ M_{10} & M_{11} & ... & M_{1N} \\ \vdots & & & \vdots \\ M_{N0} & & & M_{NN} \end{pmatrix} \rightarrow [M_{00}, M_{01}, ..., M_{10}, M_{11}, ..., M_{N0}, ..., M_{NN}]$$

4.4.3 Tabular coordinate

- `GROUPTYPE` is the group type descriptor with the fixed value `'TabularCoord'`.
- `COORDINATE_TYPE` is the is the descriptor for the coordinate type, of value `'Tabular'`.
- `STORAGE_TYPE` is the descriptor for the underlying storage type for this coordinate, of value `'Tabular'`.

```
.
`- COORDINATE_{N}            Group
   |- GROUPTYPE              Attr.        string
   |- COORDINATE_TYPE        Attr.        string
   |- STORAGE_TYPE           Attr.        string
   |- NOF_AXES               Attr.        int
   |- AXIS_NAMES             Attr.        array<string,1>
   |- AXIS_UNITS             Attr.        array<string,1>
   |- REFERENCE_VALUE        Attr.        array<double,1>
   |- REFERENCE_PIXEL        Attr.        array<double,1>
   |- INCREMENT              Attr.        array<double,1>
   `- PC                     Attr.        array<double,1>
```

Listing 4.2: Structure of the linear coordinate group.

FIELD/KEYWORD	TYPE	VALUE	DESCRIPTION
GROUPTYPE	string	'LinearCoord'	Group type descriptor
COORDINATE_TYPE	string	'Linear'	Coordinate Type descriptor
STORAGE_TYPE	string	'Linear'	Descriptor for the underlying storage type for this coordinate
NOF_AXES	int	N	Number of coordinate axes
AXIS_NAMES	array<string,1>	$[name_0,..,name_N]$	World axis names
AXIS_UNITS	array<string,1>	$[unit_0,..,unit_N]$	Physical units along each coordinate axis.
REFERENCE_VALUE	array<double,1>	$[val_0,..,val_N]$	Coordinate value at the reference point
REFERENCE_PIXEL	array<double,1>	$[pix_0,..,pix_N]$	Array location of the reference point in pixels.
INCREMENT	array<double,1>	$[incr_0,..,incr_N]$	Coordinate increment at reference point.
PC	array<double,1>	$[p_{00},pc_{01},..,p_{0N},..,p_{NN}]$	Non-singular square matrix, for the transformation from intermediate pixel coordinates to intermediate world coordinates.

Table 4.9: Keywords decribing a Linear Coordinate.

```
.
`- COORDINATE_{N}
   |- GROUPTYPE             Group      string
   |- COORDINATE_TYPE       Attr.      string
   |- STORAGE_TYPE          Attr.      string
   |- NOF_AXES              Attr.      int
   |- AXIS_NAMES            Attr.      array<string,1>
   |- AXIS_UNITS            Attr.      array<string,1>
   |- AXIS_LENGTH           Attr.      int
   |- AXIS_VALUES_PIXEL     Attr.      array<double,1>
   `- AXIS_VALUES_WORLD     Attr.      array<double,1>
```

Listing 4.3: Structure of the tabular coordinate group.

– NOF_AXES is the number of coordinate axes; keep in mind that a coordinate can consist of multiple axes.

– AXIS_NAMES are the world axis names connected with the coordinate axes, e.g.

```
AXIS_NAME=['Distance']
AXIS_NAME=['Time']
```

– AXIS_UNITS are the physical units along each coordinate axis (corresponding to the FITS keyword CUNITi, see [?]). Restrictions on the nature and range of units, if any, will be determined by agreements applying to the specific axis. If they are not so limited, units should conform to the IAU Style Manual [?].

– AXIS_VALUES_PIXEL are the tabulated values of pixel coordinates.

– AXIS_VALUES_WORLD are the tabulated values of world coordinates.

Field/Keyword	Type	Value	Description
`GROUPTYPE`	`string`	`'TabularCoord'`	Group type descriptor
`COORDINATE_TYPE`	`string`	`'Tabular'`	Coordinate Type descriptor
`STORAGE_TYPE`	`string`	`'Tabular'`	Descriptor for the underlying storage type for this coordinate
`NOF_AXES`	`int`	$N \equiv 1$	Number of coordinate axes
`AXIS_NAMES`	`array<string,1>`	$[name_0]$	World axis names
`AXIS_UNITS`	`array<string,1>`	$[unit_0]$	Physical units along each coordinate axis.
`AXIS_LENGTH`	`int`	$N_{\rm Length}$	Length of the axis, i.e. the number of elements stored in the `AXIS_VALUES_PIXEL` and `AXIS_VALUES_WORLD` arrays.
`AXIS_VALUES_PIXEL`	`array<T,1>`	$[p_0, .., p_{N_{\rm Length}}]$	Tabulated values along the pixel axis; depending on the quantity represented `T={double,int}`.
`AXIS_VALUES_WORLD`	`array<T,1>`	$[w_0, .., w_{N_{\rm Length}}]$	Tabulated values along the world axis; depending on the quantity represented `T={double,string}`.

Table 4.10: Keywords decribing a Tabular Coordinate.

4.4.4 Composite containers

4.4.5 Time coordinate

Given the characteristics of the time axis, a time coordinate internal will either be storing its values as a linear axis (`STORAGE_TYPE='Linear'`) or as a 1-dimensional look-up table (`STORAGE_TYPE='Tabular'`).

- `GROUPTYPE` is the group type descriptor with the fixed value `'TimeCoord'`.
- `COORDINATE_TYPE` is the coordinate type descriptor with the fixed value `'Time'`.
- `STORAGE_TYPE` indicates the underlying storage mechanism: if `STORAGE_TYPE='Linear'` the coordinate axis is expected to be linear and represented by the attributes defined for a Linear Coordinate (see section **??**). If set `STORAGE_TYPE='Tabular'`, the values along the coordinate axis are expected to be tabulated, thereby represented by the attributes defined for a Tabular Coordinate (see section **??**).

  > **Comment:**
  > Add description of structure depending on storage type.

- `REFERENCE_FRAME` records the reference frame within which the time coordinate axis is defined; see Tab. **??** for a list of recognized values. This can be a diferent frame as used for e.g. the direction coordinate or as noted in the coordinates group.
- `NOF_AXES` is the number of coordinate axes.
- `AXIS_NAMES` are the world axis names connected with the coordinate axes, i.e. `AXIS_NAMES=['Time']`.
- `AXIS_UNITS` are the physical units world axis of the coordinate (corresponding to the FITS keyword `CUNITi`, see [**?**]). Restrictions on the nature and range of units, if any, will be determined by agreements applying to the specific axis. If they are not so limited, units should conform to the IAU Style Manual [**?**].

4.4.6 Spectral coordinate

Spectral coordinates are commonly given in units of frequency, wavelength, velocity, and other parameters proportional to these three [**?**]. The coordinate types discussed here are then frequency, wavelength, and

Field/Keyword	Type	Value	Description
GROUPTYPE	string	'TimeCoord'	Group type descriptor
COORDINATE_TYPE	string	'Time'	Coordinate Type descriptor
STORAGE_TYPE	array<string,1>	'Linear' | 'Tabular'	Descriptor for the underlying storage type for this coordinate
REFERENCE_FRAME	string		Reference frame within which the time coordinate axis is defined; see Tab. ?? for a list of recognized values. This can be a diferent frame as used for e.g. the direction coordinate or as noted in the coordinates group.
NOF_AXES	int	$N \equiv 1$	Number of coordinate axes
AXIS_NAMES	array<string,1>	$[name_0] \equiv$ ['Time']	World axis names
AXIS_UNITS	array<string,1>	$[unit_0] \equiv$ ['s']	Physical units along each coordinate axis.
REFERENCE_VALUE	array<double,1>	$[val_0]$	Coordinate value at the reference point
REFERENCE_PIXEL	array<double,1>	$[pix_0]$	Array location of the reference point in pixels.
INCREMENT	array<double,1>	$[incr_0]$	Coordinate increment at reference point.
PC	array<double,1>	$[p_{00}] \equiv 1$	Non-singular square matrix, for the transformation from intermediate pixel coordinates to intermediate world coordinates.
AXIS_LENGTH	int	N_{Pixels}	Length of the axis, i.e. the number of elements stored in the AXIS_VALUES_PIXEL and AXIS_VALUES_WORLD arrays.
AXIS_VALUES_PIXEL	array<double,1>	$[p_0, .., p_{N_{\mathrm{Pixels}}}]$	Tabulated values along the pixel axis.
AXIS_VALUES_WORLD	array<double,1>	$[w_0, .., w_{N_{\mathrm{Pixels}}}]$	Tabulated values along the world axis.

Table 4.11: Keywords decribing a Time Coordinate; attributes within the first segment of the table will be present independent of the specific storage method.

apparent radial velocity denoted by the symbols ν, λ, and v. There are also three conventional velocities frequently used in astronomy. These are the so-called radio velocity, optical velocity, and redshift, denoted here by V, Z, and z and given by

$$V \ = \ c\,\frac{\nu_0 - \nu}{\nu_0}\,, \qquad Z \ = \ c\,\frac{\lambda - \lambda_0}{\lambda_0} \quad \text{and} \qquad z \ = \ Z/c\,.$$

The velocities are defined so that an object receding from the observer has a positive velocity. Table ?? below lists the various spectral quantities and their respective encoding as an attribute; the symbols λ_0 and ν_0 are the rest wavelength and frequency, respectively, of the spectral line used to associate velocity with observed wavelength and frequency.

As it turns out, providing a set of parameters to properly describe a spectral coordinate is not straightforward: given the arrangement of frequency channels or bands the values along the coordinate axis might be linear, but does not necessarily have to be. Therefore in principle a spectral coordinate can be considered a derivative of either a linear or a tabular coordinate, with a number of specific attributes added, as they will be required for the transformation between different spectral quantities.

Attribute	FITS Code	Name	Symbol	Associate variable	Default units
`Frequency`	`FREQ`	Frequency	ν	ν	Hz
`Energy`	`ENER`	Energy	E	ν	J
`Wavenumber`	`WAVN`	Wavenumber	κ	ν	m^{-1}
`VelocityRadio`	`VRAD`	Radio velocity	V	ν	$\mathrm{m\ s}^{-1}$
`VelocityOptical`	`VOPT`	Optical velocity	Z	λ	$\mathrm{m\ s}^{-1}$
`VelocityAppRadial`	`VELO`	Apparent radial velocity	v	v	$\mathrm{m\ s}^{-1}$
`Redshift`	`ZOPT`	Redshift	z	λ	–
`WavelengthVacuum`	`FREQ`	Vacuum wavelength	λ	λ	m
`WavelengthAir`	`AWAV`	Air wavelength	λ_a	λ_a	m
`BetaFactor`	`BETA`	Beta factor v/c	β	v	–

Table 4.12: Attributes values corresponding to the spectral coordinate codes, as defined in [?]. The IAU-standard prefixes for scaling the unit are described in [?] and should be used with al coordinate types, except that the dimensionless ones are not scaled.

Name	Symbol	Transformation equation(s)
Frequency	ν	$\nu \;=\; c/\lambda \;=\; E/h$
Vacuum wavelength	λ	$\lambda \;=\; c/\nu \;=\; \lambda_0 \dfrac{c+v}{\sqrt{c^2-v^2}}$
Apparent radial velocity	v	$v \;=\; c\,\dfrac{\nu_0^2-\nu^2}{\nu_0^2+\nu^2} \;=\; c\,\dfrac{\lambda^2-\lambda_0^2}{\lambda^2+\lambda_0^2}$
Energy	E	$E = h\nu$
Redshift	z	$z \;=\; \dfrac{\lambda-\lambda_0}{\lambda}$

Table 4.13: Spectral transformation equations; for the full set of equations – including first order derivatives – see [?].

- `GROUPTYPE` is the group type descriptor with the fixed value `'SpectralCoord'`.
- `COORDINATE_TYPE` is the coordinate type descriptor with the fixed value 'SPECTRAL'.
- `NOF_AXES` is the number of coordinate axes.
- `AXIS_NAMES` are the world axis names associated with the spectral coordinates, e.g.

  `AXIS_NAMES=['Frequency']`
  `AXIS_NAMES=['WavelengthVacuum']`

  Allowed and supported values are listed in Tab. **??** above.
- `AXIS_UNITS` are the physical units along each coordinate axis (corresponding to the FITS keyword `CUNITi`, see [?]). Restrictions on the nature and range of units, if any, will be determined by agreements applying to the specific axis. If they are not so limited, units should conform to the IAU Style Manual [?].
- `STORAGE_TYPE` indicates the underlying storage mechanism: if `STORAGE_TYPE='Linear'` the coordinate axis is expected to be linear and represented by the attributes defined for a Linear Coordinate:

```
.
`- COORDINATE_{N}             Group
   |- GROUPTYPE               Attr.        string
```

FIELD/KEYWORD	TYPE	VALUE	DESCRIPTION
GROUPTYPE	string	'SpectralCoord'	Group type descriptor
COORDINATE_TYPE	string	'Spectral'	Coordinate Type descriptor
STORAGE_TYPE	array<string,1>	'Linear' | 'Tabular'	Descriptor for the underlying storage type for this coordinate
REFERENCE_FRAME	string		Reference position w.r.t. which the spectral coordinate axis are defined; see Tab. ?? for a list of recognized values. This can be a diferent frame as used for e.g. the direction coordinate or as noted in the coordinates group.
REST_FREQUENCY	double		Rest frequency, ν_0
REST_FREQUENCY_UNIT	string	'Hz'	Physical units within which the rest frequency is given
REST_WAVELENGTH	double		Rest wavelength, λ_0
REST_WAVELENGTH_UNIT	string	'm'	Physical units within which the rest wavelength is given
NOF_AXES	int	$N \equiv 1$	Number of coordinate axes
AXIS_NAMES	array<string,1>	$[name_0]$	World axis names
AXIS_UNITS	array<string,1>	$[unit_0]$	Physical units along each coordinate axis.
REFERENCE_VALUE	array<double,1>	$[val_0]$	Coordinate value at the reference point
REFERENCE_PIXEL	array<double,1>	$[pix_0]$	Array location of the reference point in pixels.
INCREMENT	array<double,1>	$[incr_0]$	Coordinate increment at reference point.
PC	array<double,1>	$[p_{00}] \equiv 1$	Non-singular square matrix, for the transformation from intermediate pixel coordinates to intermediate world coordinates.
AXIS_LENGTH	int	N_{Pixels}	Length of the axis, i.e. the number of elements stored in the AXIS_VALUES_PIXEL and AXIS_VALUES_WORLD arrays.
AXIS_VALUES_PIXEL	array<double,1>	$[p_0, .., p_{N_{\mathrm{Pixels}}}]$	Tabulated values along the pixel axis.
AXIS_VALUES_WORLD	array<double,1>	$[w_0, .., w_{N_{\mathrm{Pixels}}}]$	Tabulated values along the world axis.

Table 4.14: Keywords decribing a Spectral Coordinate; attributes within the first segment of the table will be present independent of the specific storage method.

```
    |- COORDINATE_TYPE        Attr.       string
    |- STORAGE_TYPE           Attr.       string
    |- REFERENCE_FRAME        Attr.       string
    |- REST_FREQUENCY         Attr.       double
    |- REST_FREQUENCY_UNIT    Attr.       string
    |- REST_WAVELENGTH        Attr.       double
    |- REST_WAVELENGTH_UNIT   Attr.       string
    |- NOF_AXES               Attr.       int
    |- AXIS_NAMES             Attr.       array<string,1>
    |- AXIS_UNITS             Attr.       array<string,1>
    |- REFERENCE_VALUE        Attr.       array<double,1>
    |- REFERENCE_PIXEL        Attr.       array<double,1>
    |- INCREMENT              Attr.       array<double,1>
    `- PC                     Attr.       array<double,1>
```

In this:

- REFERENCE_VALUE is the coordinate value at the reference point (corresponding to the FITS keyword CRVALn, see [?]).
- REFERENCE_PIXEL is the array location of the reference point in pixels (corresponding to the FITS keyword CRPIXn, see [?]).
- INCREMENT is the coordinate increment at the reference point (corresponding to the FITS keyword CDELTn, see [?]).
- PC is a non-singular square matrix, for the transformation from intermediate pixel coordinates to intermediate world coordinates. The individual matrix elements are stored as a linear array, ordered as follows:

$$\mathbf{M}_{[N,N]} = \begin{pmatrix} M_{00} & M_{01} & ... & M_{0N} \\ M_{10} & M_{11} & ... & M_{1N} \\ \vdots & & & \vdots \\ M_{N0} & & & M_{NN} \end{pmatrix} \rightarrow [M_{00}, M_{01}, ..., M_{10}, M_{11}, ..., M_{N0}, ..., M_{NN}]$$

If set STORAGE_TYPE='Tabular', the values along the coordinate axis are expected to be tabulated, thereby represented by the attributes defined for a Tabular Coordinate:

```
.
`- COORDINATE_{N}             Group
    |- GROUPTYPE              Attr.       string
    |- COORDINATE_TYPE        Attr.       string
    |- STORAGE_TYPE           Attr.       string
    |- REFERENCE_FRAME        Attr.       string
    |- REST_FREQUENCY         Attr.       double
    |- REST_FREQUENCY_UNIT    Attr.       string
    |- REST_WAVELENGTH        Attr.       double
    |- REST_WAVELENGTH_UNIT   Attr.       string
    |- NOF_AXES               Attr.       int
    |- AXIS_NAMES             Attr.       array<string,1>
    |- AXIS_UNITS             Attr.       array<string,1>
    |- AXIS_LENGTH            Attr.       int
    |- AXIS_VALUES_PIXEL      Attr.       array<double,1>
    `- AXIS_VALUES_WORLD      Attr.       array<double,1>
```

In this:

- AXIS_LENGTH is the length of the tabulated axis, i.e. the number of elements stored in the AXIS_VALUES_PIXEL and AXIS_VALUES_WORLD arrays.
- AXIS_VALUES_PIXEL are the tabulated values along the pixel axis.
- AXIS_VALUES_WORLD are the tabulated values along the world axis.

4.4.7 Polarization coordinate

Definition of physical parameters. The Stokes parameters are a set of values that describe the polarization state of electromagnetic radiation. They were defined as a mathematically convenient alternative to the more common description of incoherent or partially polarized radiation in terms of its total intensity (I), (fractional) degree of polarization (p), and the shape parameters of the polarization ellipse. Early pulsar polarisation observations established a convention that is consistent with the Institute of Electrical an Electronics Engineers (IEEE) definition of left-handed and right-handed circular polarisation (LCP and RCP) and the definition of Stokes V by [**?**]. This convention differs from the one later adopted by the International Astronomical Union (IAU).

Consider a quasi-monochromatic electromagnetic wave with mean frequency ω. represented at the origin by the transverse electric field vector

$$\mathbf{e}(t) \;=\; \begin{pmatrix} e_0 \\ e_1 \end{pmatrix} \;=\; \begin{pmatrix} a_0(t)\exp i\left[\phi_0(t)+\omega t\right] \\ a_1(t)\exp i\left[\phi_1(t)+\omega t\right] \end{pmatrix} \tag{4.4}$$

Note that the complex argument *increases* linearly with time; this sign convention is commonly encountered in engineering texts [**?**, **?**, **?**] and is implicit in the definition of most forward discrete Fourier transform (DFT) implementations. It is also adopted in a seminal series of of papers on radio polarimetric calibration [**?**, **?**, **?**]. Given the above definition, time delays correspond to *negative* values of the phase ϕ.

The polarization of an electromagnetic wave is described by the second-order statistics of $\mathbf{e}$, as represeted by the complex 2×2 coherency matrix

$$\mathbf{p} \;=\; \langle \mathbf{e}\otimes\mathbf{e}^\dagger\rangle \;=\; \begin{pmatrix} \langle e_0 e_0^*\rangle & \langle e_0 e_1^*\rangle \\ \langle e_1 e_0^*\rangle & \langle e_1 e_1^*\rangle \end{pmatrix} \tag{4.5}$$

Here, the angular brackets denote an ensemble average, $\otimes$ is the direct matrix product, and $\mathbf{e}^\dagger$ is the Hermitian transpose of $\mathbf{e}$.

As summarized by [**?**], the IAU/IEEE definitions of the Stokes parameters are based on a right-handed Cartesian coordinate system, in which the plane wave propagates toward the observer in the positive z direction, and $e_0 = e_x$ and $e_1 = e_y$ are the components of the electric field projected onto North and East, respectively.

$$I \;=\; \langle |e_x|^2 + |e_y|^2\rangle \tag{4.6}$$
$$Q \;=\; \langle |e_x|^2 - |e_y|^2\rangle \tag{4.7}$$
$$U \;=\; \langle 2\,\mathrm{Re}\left[e_x e_y^*\right]\rangle \tag{4.8}$$
$$V \;=\; \langle 2\,\mathrm{Im}\left[e_x e_y^*\right]\rangle \tag{4.9}$$

Coordinate representation.

- `GROUPTYPE` is the group type descriptor with the fixed value `‘PolarizationCoord’`.
- `COORDINATE_TYPE` is the is the descriptor for the coordinate type, of value `‘Polarization’`.
- `STORAGE_TYPE` is the descriptor for the underlying storage type for this coordinate, of value `‘Tabular’`.
- `NOF_AXES` is the number of coordinate axes represented by this coordinate; as the Polarization coordinate consists of a single tabulared axis, we have `NOF_AXES = 1`.
- `AXIS_NAMES` are the world axis names connected with the coordinate axes; for a Polarization coordinate `AXIS_NAMES = ‘Polarization’`.
- `AXIS_UNITS` are the physical units along each coordinate axis (corresponding to the FITS keyword `CUNITi`, see [**?**]). Restrictions on the nature and range of units, if any, will be determined by agreements applying to the specific axis. If they are not so limited, units should conform to the IAU Style Manual [**?**].
  The units of the Stokes parameters I, Q, U and V, of total polarization (linear, elliptical or circular) and of separate circular polarizations (L, R) are some form of flux density.

Field/Keyword	Type	Value	Description
GROUPTYPE	string	‘PolarizationCoord’	Group type descriptor
COORDINATE_TYPE	string	‘Polarization’	Coordinate Type descriptor
STORAGE_TYPE	array<string,1>	‘Tabular’	Descriptor for the underlying storage type for this coordinate
NOF_AXES	int	$N \equiv 1$	Number of coordinate axes
AXIS_NAMES	array<string,1>	$[name_0] \equiv$ ‘Polarization’	World axis names
AXIS_UNITS	array<string,1>	$[unit_0]$	Physical units along each coordinate axis.
AXIS_LENGTH	int	$N_{\rm Length}$	Length of the axis, i.e. the number of elements stored in the AXIS_VALUES_PIXEL and AXIS_VALUES_WORLD arrays.
AXIS_VALUES_PIXEL	array<int,1>	$[p_0, .., p_{N_{\rm Length}}]$	Tabulated values along the pixel axis.
AXIS_VALUES_WORLD	array<string,1>	$[w_0, .., w_{N_{\rm Length}}]$	Tabulated values along the world axis, listing the stored Polarization parameters.

Table 4.15: Keywords decribing a Polarization Coordinate.

- AXIS_VALUES_PIXEL holds the tabulated values along the pixel axis
- AXIS_VALUES_WORLD holds the tabulated values along the world axis of the Polarization coordinate, i.e. the names of the Polarization components. Commonly used values are:

AXIS_VALUES_WORLD	Description
[‘I’]	Total flux density only data.
[‘I’,‘Q’,‘U’,‘V’]	Full set of standard Stokes parameters.
[‘X’,‘Y’]	Raw time-series TBB data, originating directly from the individual dipoles.
[‘XX’,‘YY’,‘XY’,‘YX’]	Cross-correlation products from a pair of X-linear and Y-linear receiver feeds.
[‘R’,‘L’,‘X’,‘Y’]	X/Y linear components, as well as R/L circular components.

For a full list of recognized values and their description see Tab. **??** below.

Term	Symbol	Description
Stokes Parameters	I	Standard Stokes total intensity, i.e. total Poynting vector or flux density of the wave.
	Q	Standard Stokes linear; degree of polarization, i.e. the difference in intensities between horizontal and vertical linearly polarized components.
	U	Standard Stokes linear; plane of polarization, i.e. the difference in intensities between linearly polarized components oriented at $\pm\pi/4$ w.r.t. the components of Q
	V	Standard Stokes circular; ellipticity, i.e. the differences in intensities between right and left circular polarized components.
Circular feeds	R	Right circular
	L	Left circular
	RR	Right-right circular
	LL	Left-left circular
	RL	Right-left circular
	LR	Left-right circular
Linear feeds	X	X linear
	Y	Y linear
	XX	X parallel linear
	YY	Y parallel linear
	XY	XY cross linear
	YX	YX cross linear

Table 4.16: Recognized values for the Polarization component parameter.

5 Example coordinate representations

5.1 Spectral coordinates

[**Comment:**
Add example for representing wavelength and/or frequency.]

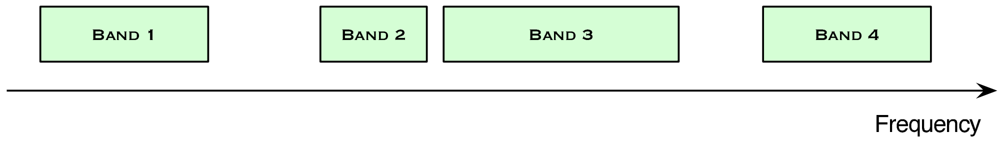
5.2 Combinations of Time and Frequency

1. Total intensity (i.e. Stokes I component only) dynamic spectrum. If a) one was not to make usage of the extra conversion information available as part of a Spectral Coordinate (Sec. **??**) and b) both time and frequency axis were regular linear axes, we could be using the following representation:

```
.
|-- GROUPTYPE                  string              = 'Coordinates'
|-- REF_LOCATION_VALUE         array<double,1>
|-- REF_LOCATION_UNIT          array<string,1>
|-- REF_LOCATION_FRAME         string
|-- NOF_COORDINATES            int                 = 2
|-- NOF_AXES                   int                 = 2
|-- COORDINATE_TYPES           array<string,1>     = ['Time','Spectral']
|-- COORDINATE_0
|   |-- GROUPTYPE              string              = 'TimeCoord'
|   |-- COORDINATE_TYPE        string              = 'Time'
|   |-- STORAGE_TYPE           string              = 'Linear'
|   |-- NOF_AXES               int                 = 1
|   |-- AXIS_NAMES             array<string,1>     = ['Time']
|   |-- AXIS_UNITS             array<string,1>     = ['s']
|   |-- REFERENCE_VALUE        array<double,1>     = [1.0]
|   |-- REFERENCE_PIXEL        array<double,1>     = [0.0]
|   |-- INCREMENT              array<double,1>     = [0.5]
|   `-- PC                     array<double,1>     = [1.0]
`-- COORDINATE_1
    |-- GROUPTYPE              string              = 'SpectralCoord'
    |-- COORDINATE_TYPE        string              = 'Spectral'
    |-- STORAGE_TYPE           string              = 'Linear'
    |-- NOF_AXES               int                 = 1
    |-- AXIS_NAMES             array<string,1>     = ['Frequency']
    |-- AXIS_UNITS             array<string,1>     = ['Hz']
    |-- REFERENCE_VALUE        array<double,1>     = [200.0]
    |-- REFERENCE_PIXEL        array<double,1>     = [0.0]
    |-- INCREMENT              array<double,1>     = [10.0]
    `-- PC                     array<double,1>     = [1.0]
```

   Note: The coordinate system described above as well could be represented using a single linear coordinate consisting of two axes.

2. Given the nature and organization of the data, the sequence of frequency values along the respective data axis cannot be represented properly by a simple linear coordinate. As a consequence of this, the frequency axis spanning multiple frequency-bands is represented using a tabulated coordinate.



The above figure shows gaps in the frequency axis for an observation with 4 bandpasses which are not similar in frequency range. This is typical for a LOFAR Beam-Formed type of observation [?].

A coordinate group to represent such a scenario would look like:

```
.
`-- COORDINATES
    |- NOF_COORDINATES          2
    |- NOF_AXES                 2
    |- COORDINATE_TYPES         ['Time', 'Spectral']
    |- COORDINATE_0
    |  |- COORDINATE_TYPE       'Time'
    |  |- STORAGE_TYPE          'Linear'
    |  |- NOF_AXES              1
    |  |- AXIS_NAMES            ['Time']
    |  |- AXIS_UNITS            ['s']
    |  |- REFERENCE_VALUE       [1.0]
    |  |- REFERENCE_PIXEL       [0.0]
    |  |- INCREMENT             [1.0]
    |  `- PC                    [1.0]
    `- COORDINATE_1
       |- COORDINATE_TYPE       'Spectral'
       |- STORAGE_TYPE          'Tabular'
       |- NOF_AXES              1
       |- AXIS_NAMES            ['Spectral']
       |- AXIS_UNITS            ['MHz']
       |- AXIS_LENGTH           1024
       |- AXIS_VALUES_PIXEL     [0, 1, 2, ..., 512, 513, ...]
       `- AXIS_VALUES_WORLD     [140, 140.1953125, 140.390625, ...,
                                 150, 150.1953125, ...]
```

5.3 Positions in space

Following our data model, positions in space can be divided into two basic groups:

1. Positions on spherical shells, using a spherical map projection onto a plane surface.
2. Positions in 3-dimensional space, as described through a set of cartesian, spherical, cylindrical, etc. coordinates.

Position with spherical map projection

Comment:
Mention in particular usage as part of the Sky Image. Extend the started examples to provide full set of attributes and values.

1. Direction (angular position with spherical map projection) and radial component:

```
.
|-- Direction  [2]
`-- Linear     [1]
```

```
.
|-- Direction  [2]
`-- Tabular    [1]
```

2. Direction (angular position with spherical map projection) with spectral component for Stokes (I, Q, U, V) parameters.

```
.
|-- Direction  [2]
|-- Spectral   [1]
`-- Polarization    [1]
```

Position without spherical map projection

1. Cartesian coordinates, regular (x, y, z) grid:

   a) Single coordinate with three coordinates axes:

   ```
   COORDINATES
   `-- COORDINATE_0            Linear  [3]
   ```

   b) Three coordinates, each representing a single axis:

   ```
   COORDINATES
   |-- COORDINATE_0            Linear  [1]
   |-- COORDINATE_1            Linear  [1]
   `-- COORDINATE_2            Linear  [1]
   ```

2. Cartesian coordinates, regular (x, y) grid, non-regular z-axis

   a) Single coordinate for the two linear axes, tabular coordinate for the non-linear axis:

   ```
   COORDINATES
   |-- COORDINATE_0            Linear   [2]
   `-- COORDINATE_0            Tabular  [1]
   ```

   b) Three coordinates, each representing a single axis:

   ```
   COORDINATES
   |-- COORDINATE_0            Linear   [1]
   |-- COORDINATE_1            Linear   [1]
   `-- COORDINATE_2            Tabular  [1]
   ```

3. Cartesian coordinates, non-regular x-, y-, and z-axes

   ```
   COORDINATES
   |-- COORDINATE_0            Tabular  [1]
   |-- COORDINATE_1            Tabular  [1]
   `-- COORDINATE_2            Tabular  [1]
   ```

6 Discussion

6.1 Open questions/Issues

The following table presents an overview of (some of the) known open questions regarding the format definition:

Item	Description	Status
01	Compare guidelines on physical units from IAU Style Manual [?] with guidelines/definitions in ESO ICDs and IOVA manuals (e.g. [?]).	open
02	Copy Table 1 of [?] to ICD.	open
03	Motivation for reference frame keywords in both spectral coordinate and coordinates group: "The parameters needed to compute geocentric frequencies/velocities from topocentric are the siderial time and the observation location. The observation date is needed to convert from geocentric to barycentric coordinates." [?]	open
04	Provide translation scheme onto representation in FITS [?, ?, ?].	open
05	Add figures with position grids of spatial coordinates	open
06	What exactly is the difference between the separate conventions (e.g see [?])? Is this something simple like e.g. a different sign in front of the ωt? Work out who this actually affects the standard Stokes parameters.	open

6.2 Future enhancements

—/—

Glossary of terms

Az Azimuth.

AIPS++ The AIPS++ project was a project from the nineties supposed to replace the original Astronomical Information Processing System or classical AIPS. The ++ comes from it being mainly developed in C++. It's also known as AIPS 2. It evolved into CASA, casacore and casarest (see those entries).

BBS BlackBoard Selfcal, pipeline used for LOFAR imaging data.

Beam A beam is formed by combining all the SubArrayPointing, one for each station, which are looking in a particular direction. There may be more than one beam for each SubArrayPointing, and different types of beams are available.

BF Beam-Formed data (time series structure).

CASA The Common Astronomy Software Applications package. User software for radioastronomy developed out of the old AIPS++ project. The project is led by NRAO with contributions from ESO, CSIRO/ATNF, NAOJ and ASTRON. [**?**]

casacore The set of C++ libraries that form the basis of CASA and several other astronomical packages. It contains classes for storing and handling visibility and image data, RDBMS-like table system and handling coordinates. Mainly maintained by ASTRON and CSIRO/ATNF. [**?**]

casarest The libraries and tools from the old AIPS++ project that are not part of casacore or CASA but still in use.

CEP Central Processing facility.

Channel The subband data of a LOFAR observation may be passed through a second polyphase filter to obtain a large number of channels (i.e. to increase the spectral resolution).

CLA Common LOFAR attributes. Set of root-level attributes that are used and required as attributes in all LOFAR science data products. If a value is not available for an Attribute, `'NULL'` maybe used.

Co-I Co-investigators on an observation project under the leadership of the PI.

Data Interface Set of definitions that describe the contents and structure of data files.

Data Access Layer (DAL) A C++ library with Python bindings providing read/write functionality for HDF5 format files, as well as access to Measurement Sets.

Dec Declination.

DPPP Default Pre-Processing Pipeline, pipeline used for LOFAR imaging data.

EAS Extensive Air-Shower.

El Elevation.

FITS FITS (Flexible Image Transport System) is a digital file format used to store, transmit, and manipulate scientific and other images. FITS commonly used in astronomy.

HBA High Band Antenna.

HDFView Hierarchical Data Format Viewer; a Java software tool for viewing the HDF5 structure and data. [http://www.hdfgroup.org/hdf-java-html/hdfview/]

HDF5 Hierarchical Data Format, 5 [?]. A file format capable of accommodating large datasets that comprises two (2) primary types of objects: groups and datasets. Implements self-organisation and hierarchical structures within the file format itself, facilitating self-contained data administration. [?, ?]

HDF5 group A grouping structure containing zero or more HDF5 objects, together with supporting meta-data.

HDF5 dataset A multidimensional array of data elements, together with supporting meta-data.

HDU Header-Data Unit Though typically used for FITS data descriptions, the term "HDU" can also be used more generically when discussing any data group that contains both data and a descriptive header.

Hypercube The hypercube is a generalization of a 3-cube to n dimensions, also called an n-cube or measure polytope. In data modelling a hypercube is a cube-like logical model in which all measurements are organized into a multidimensional space.

ICD Interface Control Document.

IVOA International Virtual Observatory Alliance.

KSP Key Science Project. One of several major observational and research projects defined by the LOFAR organization. These Key Science Projects are,

- Cosmic Magnetism in the Nearby Universe
- High Energy Cosmic Rays
- Epoch of Re-ionization
- Extragalactic Sky Surveys
- Transients - Pulsars, Jet Sources, Planets, Flare stars
- Solar Physics and Space Weather

LBA Low Band Antenna.

LOFAR The LOw Frequency ARray. LOFAR is a multipurpose sensor array; its main application is astronomy at low radio frequencies, but it also has geophysical and agricultural applications. [http://www.lofar.org/]

LOFAR Sky Image Standard LOFAR Image Cube. A LOFAR data product encompassing science data, associated meta-data, and associated calibration information, including a Local Sky Model (LSM) , and other ancillary meta groups that are defined in this document.

LSM/GSM The Local Sky Model/Global Sky Model. Sky Models are essentially catalogues of known real radio sources in the sky. A Local Sky Model for an observation is merely a subset of a Global Sky Model catalogue pertaining to that observation's relevant region of the sky.

LTA The Long Term Archive for LOFAR.

MJD Modified Julian Day. Derived from Julian Date (JD) by MJD = JD - 2400000.5. Starts from midnight rather than noon.

MS Measurement Set, a self-described, structured set of casacore tables comprising the data and meta-data of an observation. [?]

PI A Principal Investigator is the lead scientist resopnsible for a particular observation project.

RA Right Ascension.

RFI Radio Frequency Interference.

RM Rotation Measure.

RMSC The Rotation Measure synthesis cube is a data product which contains the output of LOFAR RM synthesis routines, namely the polarized emission as a function of Faraday depth. As with the Sky Image data files, all associated information is stored within an RMSC file.

RSP Remote Station Processing Board.

SIP Standard Imaging Pipeline or Submission Information Package within the context of the LTA.

Station Group of antennae separated from other groups. In it's current cofiguration, LOFAR has 48 stations.

SubArrayPointing This corresponds to the beam formed by the sum of all of the elements of a station. For any given observation there may be more than one SubArrayPointing, and they can be pointed at different locations.

Subband At the station level, LOFAR data are passed through a polyphase filter, producing subbands of either 156.250 kHz or 195.3125 kHz (depending on system settings).

TAI International Atomic Time (Temps Atomique International), atomic coordinate time standard.

TBB Transient Buffer Board.

TRAP Transients Pipeline.

USG LOFAR User Software Group.

UTC Coordinated Universal Time (UTC) is a time standard based on International Atomic Time (TAI) with leap seconds added at irregular intervals to compensate for the Earth's slowing rotation.

UV-Coverage A spatial frequency domain area that must be covered completely by observation in order to assure an optimal target image (Full UV- Coverage). During observation, the radio telescope turns with respect to its target, due to the earth rotation. A certain -instrument geometry dependent-rotation angle has to be covered in order to accomplish full coverage.

VHECR Very high-energy cosmic ray.

WCS World Coordinate Information (WCS). The FITS "World Coordinate System" (WCS) convention defines keywords and usage that provide for the description of astronomical coordinate systems in a FITS image header [?, ?, ?].