

## MeasurementSet description for LOFAR

### Version 2.07.01

	Organisatie / Organization	Datum / Date
<b>Auteur(s) / Author(s):</b>  A.P. Schoenmakers, G.A.Renting	ASTRON	24-03-2011
<b>Controle / Checked:</b>	ASTRON	
<b>Goedkeuring / Approval:</b>	ASTRON	
<b>Autorisatie / Authorisation:</b>  <b>Handtekening / Signature</b>	ASTRON	

© ASTRON 2006-2011  
 All rights are reserved. Reproduction in whole or in part is prohibited without written consent of the copyright owner.

## Distribution list:

---

Group:	Others:

## Document history:

---

Revision	Date	Chapter / Page	Modification / Change
0.1	March 2006	-	Creation
0.2	20 July 2006		Added columns to MAIN; filled in station name descriptions in ANTENNA; transfer to current document template, filling in more details, removal of irrelevant table entries.
0.3	30 November 2006		Added MS schema figure, Added appendix on how to translate parset file to MS fields
0.4	24 January 2007		Small additions due to recent developments
0.5	25 February 2008		Updated to latest field values/formats/keywords used
0.6 <i>Drafts</i>	February-September 2010		Draft updates for better dipole and antenna field support and full LOFAR operations
2.06.07	14 October 2010		New numbering scheme and naming conventions compatible with ICDs Final version of updates for dipole and antenna field support for LOFAR operational phase
2.07.01	19 April 2011		New version with new LOFAR_ANTENNA_FIELD and LOFAR_ELEMENT_FAILURE tables and some smaller updates

## Table of contents:

---

1	Introduction .....	5
1.1	Antenna field clarification .....	5
1.2	Other important conventions .....	5
2	General description of the Measurement Set .....	6
2.1	Structure of the MS .....	6
2.2	MAIN table .....	6
2.3	ANTENNA Table .....	6
2.4	DATA DESCRIPTION Table .....	6
2.5	FEED table .....	6
2.6	FIELD table .....	7
2.7	FLAG_CMD table .....	7
2.8	HISTORY table .....	7
2.9	OBSERVATION table .....	7
2.10	POINTING table .....	7
2.11	POLARIZATION table .....	7
2.12	PROCESSOR table .....	7
2.13	SOURCE table .....	7
2.14	SPECTRAL_WINDOW table .....	7
2.15	STATE table .....	7
2.16	LOFAR_ANTENNA_FIELD table .....	7
2.17	LOFAR_ELEMENT_FAILURE table .....	8
2.18	LOFAR_STATION table .....	8
2.19	EPHEMERIDES table .....	8
3	Detailed description of all MS tables .....	9
3.1	MAIN table .....	9
3.2	ANTENNA table .....	11
3.3	DATA_DESCRIPTION Table .....	12
3.4	FEED table .....	13
3.5	FIELD table .....	14
3.6	OBSERVATION table .....	15
3.7	POINTING table .....	17
3.8	POLARIZATION table .....	18
3.9	PROCESSOR table .....	18
3.10	SOURCE table (Optional) .....	19
3.11	SPECTRAL_WINDOW table .....	20
3.12	STATE table .....	21
3.13	LOFAR_ANTENNA_FIELD table .....	22
3.14	LOFAR_ELEMENT_FAILURE table .....	23
3.15	LOFAR_STATION table .....	23
3.16	HISTORY table .....	24
	Schema of the LOFAR MS .....	25
A	Appendix A: Translating the parset file .....	26
A.1	Pointing information .....	27
A.2	Antenna information .....	27
A.3	Frequency information .....	27
A.4	Time issues .....	28
A.5	Polarization .....	28
A.6	DataSet naming .....	29
	Appendix B: Lofar Storage Manager (LofarStMan) .....	30

B.1	Introduction .....	30
B.2	LOFAR MeasurementSet creation .....	30
B.3	Accessing an existing LOFAR MeasurementSet .....	31
B.4	Lessons learned and future.....	32

## 1 Introduction

In this document I present detailed schemes of the tables in a Measurement Set, and elaborate on their exact definitions. This will grow into the blueprint of a new LOFAR MS.

This is based (and partially copied) from the original AIPS++ MS 2.0 definition document (AIPS++ Note 229) and should be compatible with it. The Measurement Set v2 is described in great detail in an AIPS++ memo, which can be found at these locations:

<http://aips2.nrao.edu/stable/docs/notes/229/229.html> or

<http://www.astron.nl/aips++/docs/notes/229/229.html>

The latest version of this document is available at:

[http://www.lofar.org/wiki/doku.php?id=public:documents:lofar\\_documents](http://www.lofar.org/wiki/doku.php?id=public:documents:lofar_documents)

The fields given in parentheses are not compulsory, and may thus be skipped. Some fields that are not relevant at all have been skipped (see the MS2 definition document for their meaning).

### 1.1 Antenna field clarification

Please observe that LOFAR does not have traditional antennas, but instead uses fields of dipoles. These get beam formed in groups to behave as a more traditional antenna and are called Antenna Fields in this document. A station has several Antenna Fields, some of which can be active at the same time during an observation.

There is one special mode called HBA\_JOINED in which basically joins two different groups of dipoles into one Antenna Field, where normally each group is a separate antenna field. This mode has required the creation of a separate LOFAR\_ANTENNA\_FIELD table and has made it a bit unclear what an Antenna Field exactly means. In this document I make the distinction through capitalisation: An Antenna Field corresponds to a traditional Antenna, and is the end point of a baseline. An antenna field is one of the groups of dipoles that alone or together with another antenna field form the Antenna Field. Each antenna field can be found in the LOFAR\_ANTENNA\_FIELD table.

### 1.2 Other important conventions

Unless otherwise specified, direction coordinates in are specified in reference to the LOFAR Core. This is relevant because of the size of LOFAR a lot of objects like the Moon and planets are in near field.

Also take note that SubArray Pointing was referred to as Station Beam or Beam in older versions of this document. The term Beam has now been reserved for what used to be called Pencil Beam.

## 2 General description of the Measurement Set

This section briefly describes all tables and sub-tables in a Measurement set, and what type of information these contain. I will discuss individual fields in so far as these require additional explanations to understand their exact meaning. These are mostly administrative fields, not the main data fields.

### 2.1 Structure of the MS

In the table below a short summary of the sub-tables in a LOFAR Measurement Set is presented and a brief explanation of their contents. The subsequent sections describe the function of each table.

<i>Table name</i>	<i>Short description of content</i>
MAIN	Data of all samples for individual interferometers
ANTENNA	Antenna information
DATA_DESCRIPTION	Pointers to POLARIZATION and SPECTRAL_WINDOW entries
FEED	Feed (Frontend) related information
FIELD	Information on observed positions
FLAG_CMD	Flag information
HISTORY	History log of MS
OBSERVATION	General observation information
POINTING	Antenna pointing information
POLARIZATION	Polarization description information
PROCESSOR	Correlator information
SOURCE	Information on observed sources
SPECTRAL_WINDOW	Frequency/IF information
STATE	State information (mostly for SD)
LOFAR_ANTENNA_FIELD	LOFAR Antenna Field information
LOFAR_ELEMENT_FAILURE	Time dependent failure information on tiles/dipoles
LOFAR_STATION	Station information
EPHEMERIDES	Ephemerides information

### 2.2 MAIN table

The MAIN table of the MS contains the bulk of all data. For each interferometer and for each sample time The ordering is usually time-baseline, i.e. the MAIN table is divided in subsequent time-blocks and within each time block there is an ordering based on the interferometer antenna pair. Each interferometer pair will appear only once per sample time. Auto- and crosscorrelations are usually mixed. The MAIN table links directly to many other tables through index numbers in several of its columns.

### 2.3 ANTENNA Table

The Antenna table contains information on each Antenna Field used in the observation.

### 2.4 DATA DESCRIPTION Table

This is a simple table that currently only contains identifiers for the SPECTRAL\_WINDOW and POLARIZATION table.

### 2.5 FEED table

This table gives information on the front-ends (Feeds) of the antennas used. It allows for time dependent beam and polarization properties through a TIME and INTERVAL column.

## 2.6 FIELD table

This table contains positional information on the object(s) that is pointed to by the telescope.

## 2.7 FLAG\_CMD table

The FLAG\_CMD table allows for additional flagging information in the MS, to be applied to the data in the MAIN table. For LOFAR this table is not being used, see the LOFAR\_FULL\_RES\_FLAG column in the MAIN table instead.

## 2.8 HISTORY table

This table allows HISTORY information to enter the MS so that one can trace what has happened to the MS since it was created. The on-line systems can add a creation statement here. It should mainly be filled and used by processes that alter the data in the MS after initial creation of the MS. Each step in the processing pipelines will store its parset file settings and other relevant data in this table.

## 2.9 OBSERVATION table

This table contains information on the project(s) and the scheduling. It is pointed to from the MAIN table by the OBSERVATION\_ID column. It contains information related to the SAS/MAC/MoM Observation and Project and the other fields common to the different dataformats for LOFAR.

## 2.10 POINTING table

The POINTING table provides information on the actual pointing of an antenna, which may be time dependent. The structure and the fields are particularly useful for dish-based telescopes and interferometers.

## 2.11 POLARIZATION table

The POLARIZATION table provides information on the used polarization properties of the receivers. It refers to the polarization dependent fields in the FEED table through the CORR\_PRODUCT field.

## 2.12 PROCESSOR table

This table gives some administrative information on the back-end processing. It can be used to store some information on the backend setup (e.g., the mode that was used).

## 2.13 SOURCE table

This table gives some information of the physical properties of the source(s) observed, as well as their use in the observing strategy (calibrator or not). For spectral line observations it contains information on the line rest frequency and the velocity of the object to be observed. This table is the only non-compulsory table we added to the LOFAR Measurement Set as it helps to clarify the goal of the observation.

## 2.14 SPECTRAL\_WINDOW table

This table presents the frequency settings of the observation.

## 2.15 STATE table

This is mostly in use for single dish and may not be useful for LOFAR.

## 2.16 LOFAR\_ANTENNA\_FIELD table

This is a custom table for use by LOFAR only. It contains additional information specific to each antenna field. This table is required to support the mode where OBSERVATION.LOFAR\_ANTENNA\_SET is equal

to HBA\_JOINED because then multiple antenna fields at the station are combined into a single Antenna Field.

### 2.17 LOFAR\_ELEMENT\_FAILURE table

This is a custom table for use by LOFAR only. It contains information about failure in time of the individual dipoles/tiles of each antenna field.

### 2.18 LOFAR\_STATION table

This is a custom table for use by LOFAR only. It contains a list of all the stations that were present in the observation, mainly so the ANTENNA table can reference them with the purpose of identifying which antenna fields are part of which station.

### 2.19 EPHEMERIDES table

This is a table that is referenced in the MS 2.0 definition but never defined. It is currently not needed for LOFAR but added to this document for future extension and definition if it becomes necessary to describe fields or objects being observed which move fast enough to not have a constant position on the sky during an observation. Note that a lot of objects like the Sun, Moon and planets already have ephemerides information available through the standard AIPS++/CASA DATA/EPHIMERIDES. This table is foreseen to provide ephemerides for objects not in that table, like satellites.



### 3 Detailed description of all MS tables

#### 3.1 MAIN table

MAIN Table: Data, coordinates and flags					
Name	Format	Units	Measure	Default	Comments
<i>Keywords</i>					
MS_VERSION	Float			2.0	MS version number
LOFAR_VERSION	Float			07.01	Version of this document
<i>Key</i>					
TIME	Double	s	EPOCH		Integration midpoint
ANTENNA1	Int				Pointers to ANTENNA table
ANTENNA2	Int				
FEED1	Int				Pointers to FEED_ID column in FEED table
FEED2	Int				
DATA_DESC_ID	Int				Pointer to DATA_DESCRIPTOR table
PROCESSOR_ID	Int				Pointer to PROCESSOR table
FIELD_ID	Int				Pointer to FIELD table
<i>Non-key attributes</i>					
UVW	Double (3)	m	UVW J2000		UVW coordinates at TIME_CENTROID in J2000
INTERVAL	Double	s			Sampling interval (this sample)
EXPOSURE	Double	s			Effective integration time
TIME_CENTROID	Double	s	EPOCH		Average time of sample
SCAN_NUMBER	Int			0	Scan number
ARRAY_ID	Int			0	Sub-array number
OBSERVATION_ID	Int			0	Pointer to OBSERVATION table
STATE_ID	Int			0	Pointer to State table
<i>Data</i>					
DATA	Complex (Nc, Nf)	Jy			Raw complex visibilities
CORRECTED_DATA	Complex (Nc, Nf)			0	Corrected complex visibilities
MODEL_DATA	Complex (Nc, Nf)			0	Complex visibilities of data model
(VIDEO_POINT)	Complex (Nc)				Video point
SIGMA	Float (Nc)			[1,1,1,1]	Estimated rms noise for single channel
WEIGHT	Float (Nc)				Weight for whole DATA matrix
WEIGHT_SPECTRUM	Float (Nc, Nf)				Weight for each channel separately
<i>Flag information</i>					
FLAG	Bool(Nc, Nf)				Cumulative data flags
FLAG_CATEGORY	Bool (Nc, Nf, Ncat)			Empty	Flag categories
FLAG_ROW	Bool			F	Row flag
LOFAR_FULL_RES_FLAG	uChar(Nf/8, Ntimeavg)				Full resolution flag before compression

Note that  $N_c$  = number of independent correlation signals (i.e., polarizations),  $N_f$  = number of frequency channels, and  $N_{cat}$  = number of flag categories (not used for LOFAR, so  $N_{cat}=1$ ),  $N_{timeavg}$  = is the number of timeslots averaged to a single timeslot.

**MS VERSION** The MeasurementSet format revision number, expressed as *major revision.minor revision*. This version is 2.0.

**LOFAR VERSION** The LOFAR MeasurementSet format revision number as defined by this document.

**TIME** This is the mid-point (not centroid) of data interval in UTC.

**ANTENNA $n$**  This is the antenna number ( $\geq 0$ ), and a direct index into the ANTENNA sub-table *rownr*.

**FEED $n$**  This is the feed number ( $\geq 0$ ). This points to the FEED\_ID column in the FEED table.

**DATA\_DESC\_ID** Data description identifier ( $\geq 0$ ), and a direct index into the DATA DESCRIPTION sub-table *rownr*.

**PROCESSOR\_ID** Processor identifier ( $\geq 0$ ), and a direct index into the PROCESSOR sub-table *rownr*.

**FIELD\_ID** Field identifier ( $\geq 0$ ), and a direct index into the FIELD sub-table *rownr*.

**INTERVAL** Data sampling interval. This is the nominal data interval and does not include the effects of bad data or partial integration.

**EXPOSURE** This is the effective data interval, including bad data and partial averaging. For now equal to INTERVAL.

**SCAN\_NUMBER** Arbitrary scan number to identify data taken in the same logical scan. Not required to be unique. For LOFAR this is used to identify records in mosaic observations that should be grouped by pointing or time.

**ARRAY\_ID** This is the subarray identifier ( $\geq 0$ ), which identifies data in separate sub arrays. Currently always 0 for LOFAR as only one sub-array can be active at the same time using between one and all available antenna fields.

**OBSERVATION\_ID** This is the observation identifier ( $\geq 0$ ), a direct index into the OBSERVATION sub-table *rownr*. This is not the same as the ObservationId that is used by MoM, SAS and LTA, which can be found in the OBSERVATION table.

**STATE\_ID** This is the state identifier ( $\geq 0$ ), and a direct index into the STATE sub-table *rownr*. Always 0 for LOFAR.

**UVW** *uvw* coordinates for the baseline from ANTENNA2 to ANTENNA1, i.e. the baseline is equal to the difference POSITION2 - POSITION1. The UVW given are for the TIME\_CENTROID, and correspond in general to the reference type for the PHASE\_DIR of the relevant field. I.e. J2000 if the phase reference direction is given in J2000 coordinates. However, any known reference is valid. Note that the choice of baseline direction and UVW definition (*W* towards source direction; *V* in plane through source and system's pole; *U* in direction of increasing longitude coordinate) also determines the sign of the phase of the recorded data.

**DATA** These are the measured visibilities in units of correlation coefficients. This contains the output of the correlator.

**CORRECTED\_DATA** The visibilities corrected for instrumental effects (calibration), or even after subtraction of a local/global sky model. When not used, this contains an array of zero values.

**MODEL\_DATA** The predicted visibilities of a local/global sky model. When not used, this contains an array of zero values.

**VIDEO\_POINT** The video point for the spectrum, to allow for the full reverse FFT transform of the spectrum to the correlation function. Can be ignored for LOFAR.

**SIGMA** The estimated rms noise for a single channel, for each correlator. If we don't have this number, it will be set to 1.

**SIGMA\_SPECTRUM** The estimated rms noise for each channel.

**WEIGHT** The weight for the whole data matrix for each correlator, as assigned by the correlator or processor. Will be set to the average weight in the WEIGHT\_SPECTRUM column (without taking FLAGS into account).

**WEIGHT\_SPECTRUM** The weight for each channel in the data matrix, as assigned by the correlator or processor. The weight spectrum should be used in preference to the WEIGHT, when available. For LOFAR the weight spectrum is determined in the correlator by the ratio of received packets per integration time and then updated adjusted for flagging when compressing in time or frequency.

**FLAG** An array of Boolean values with the same shape as DATA (see the DATA item above) representing the cumulative flags applying to this data matrix, as specified in FLAG CATEGORY. Data are flagged as bad if the FLAG array element is True.

**FLAG\_CATEGORY** An array of flag matrices with the same shape as DATA, but indexed by category. The category identifiers are specified by a keyword CATEGORY, containing an array of string identifiers, attached to the FLAG CATEGORY column and thus shared by all rows in the MeasurementSet. The cumulative effect of these flags is reflected in column FLAG. Data are flagged bad if the FLAG array element is True. This column is not used for LOFAR see LOFAR\_FULL\_RES\_FLAG instead.

**FLAG\_ROW** True if the entire row is flagged. Not used, so always set to False.

**FULL\_RES\_FLAG** These are an array of flags at the full resolution of the original uncompressed data containing what flags were generated before compression. This is needed to correct for bandwidth and time smearing due to compression. It contains bits. uChar is used for it, because Int is sensitive to endianness. It contains a 2-dim array with shape  $[(Nf+7)/8, Ntimeavg]$  where Ntimeavg is the number of time slots averaged to a single time slot. As for LOFAR all polarizations are flagged if one is flagged, no axis with Nc length is needed.

Note that in the last time slot of an averaged MS, FULL\_RES\_FLAG will contain True flags for in case of missing time slots in the original MS.

The column has two keywords defined with it:

NCHAN\_AVG defines the number of channels averaged to one.

NTIME\_AVG defines the number of time slots averaged to one. It is equal to the second axis of the shape.

Finally note that a FLAG value True does not mean that all corresponding FULL\_RES\_FLAG values are True. But when averaging further, FULL\_RES\_FLAG values will be set to True for FLAG values equal to True.

## 3.2 ANTENNA table

ANTENNA Table: Antenna Field Characteristics					
Name	Format	Units	Measure	Default	Comments
<i>Data</i>					
NAME	String			LOFAR	Unique antenna field names
LOFAR_STATION_ID	Int				Pointer to STATION table
TYPE	String			GROUND-BASED	Antenna Type
MOUNT	String			X-Y	Antenna Mounting
POSITION	Double (3)	m	POSITION		Antenna center of light position in ITRF or WGS84
OFFSET	Double (3)	m	POSITION	0	Axes offset of mount to FEED_REFERENCE point
DISH_DIAMETER	Double	m		0	No meaning for LOFAR
LOFAR_PHASE_REFERENCE	Double (3)	m	POSITION		Beam forming phase reference
<i>Flag information</i>					
FLAG_ROW	Bool			False	Row flag

Notes: This sub-table contains the global antenna field properties for each antenna field in the MS. It is indexed directly from MAIN via ANTENNAn.

**NAME** Antenna field name. This is either HBA, LBA, HBA0 or HBA1.

**LOFAR\_STATION\_ID** This is the station identifier ( $\geq 0$ ), a direct index into the LOFAR\_STATION sub-table *rownr*.

**TYPE** Antenna type. Reserved keywords include: ("GROUND-BASED" - conventional antennas; "SPACE-BASED" - orbiting antennas; "TRACKING-STN" - tracking stations). We will use "GROUND-BASED".

**MOUNT** The mount type of the antenna. Reserved keywords include: ("EQUATORIAL" - equatorial mount; "ALTAZ" - azimuth-elevation mount; "X-Y" - x-y mount; "SPACE-HALCA" - specific orientation model.). For LOFAR we will use "X-Y".

**POSITION** In a right-handed frame, X towards the intersection of the equator and the Greenwich meridian, Z towards the pole. The exact frame should be specified in the MEASURE\_REFERENCE keyword (ITRF or WGS84) attached to this column. On traditional telescopes the reference point is the point on the azimuth or horizontal ascension axis closest to the elevation or declination axis. For LOFAR this is the effective centre of the collecting area of the antenna field, usually a weighted average of the positions of the individual antennas. It is needed because one calculates the (u,v,w) coordinates as differences of the center of light positions between antenna fields, not the phase reference positions.

**OFFSET** Axes offset of mount to feed reference point.

**DISH\_DIAMETER** This is the nominal diameter of dish, as opposed to the effective diameter. This is not a constant for LOFAR. For LOFAR this is set at 0 meter to make it obvious that it should not be used.

**FLAG\_ROW** This is the boolean flag to indicate the validity of this entry. Set to True for an invalid row. This does not imply any flagging of the data in MAIN, but is necessary as the ANTENNA index in MAIN points directly into the ANTENNA sub-table. Thus FLAG\_ROW can be used to delete an antenna entry without re-ordering the ANTENNA indices throughout the MS.

**LOFAR\_PHASE\_REFERENCE** This is the phase reference position, the location at which the antenna field tries to adjust the "phases" for all of the dipoles of an antenna field to be equal for the Sub Array Pointing direction.

Time variant LOFAR antenna field and station properties (e.g., switching individual dipoles in a field off during an observation) cannot be handled directly in the antenna table. This is only relevant when such a change leads to different antenna characteristics (e.g., center of light position). For the current version of the LOFAR Measurement Set the choice has been made to handle this through the LOFAR\_ELEMENT\_FAILURE table. This means that the effect such changes have on the center of light position are not reflected in the POSITION in this table.

### 3.3 DATA\_DESCRIPTION Table

DATA_DESCRIPTION Table: Frequency/polarization characteristics					
Name	Format	Units	Measure	Default	Comments
<i>Data</i>					
SPECTRAL_WINDOW_ID	Int				
POLARIZATION_ID	Int				
<i>Flag information</i>					
FLAG_ROW	Bool				Row flag

Notes: This table defines the shape of the associated DATA array in MAIN, and is indexed directly by DATA\_DESC\_ID.

**SPECTRAL\_WINDOW\_ID** Spectral window identifier. It is a direct index into the SPECTRAL\_WINDOW sub-table.

**POLARIZATION\_ID** Polarization identifier ( $\geq 0$ ). It is a direct index into the POLARIZATION sub-table.

**FLAG\_ROW** True if the row does not contain valid data; This does not imply flagging in MAIN.

## 3.4 FEED table

FEED Table: Feed characteristics					
Name	Format	Units	Measure	Default	Comments
<i>Key</i>					
ANTENNA_ID	Int				
FEED_ID	Int			0	
SPECTRAL_WINDOW_ID	Int			-1	
TIME	Double	s	EPOCH		Interval midpoint
<i>Data description</i>					
NUM_RECEPTORS	Int			2	Number of receptors on this feed
<i>Data</i>					
BEAM_ID	Int			-1	Beam model
BEAM_OFFSET	Double (2, NUM_RECEPTORS)	rad	DIRECTION	0	Beam position offset (on sky but in antenna reference frame).
POLARIZATION_TYPE	String (NUM_RECEPTORS)			X, Y	Type of polarization to which a given RECEPTOR responds.
POL_RESPONSE	Complex (NUM RECEPTORS, NUM RECEPTORS)				Feed polarization response
POSITION	Double(3)	m	POSITION	0,0,0	Position of feed relative to feed reference position for this antenna
RECEPTOR_ANGLE	Double (NUM RECEPTORS)	rad			The reference angle for polarisation.

Notes: A feed is a collecting element on an antenna, such as a single horn, that shares joint physical properties and makes sense to calibrate as a single entity. It is an abstraction of a generic antenna feed and is considered to have one or more RECEPTORS that respond to different polarization states. A FEED may have a time variable beam and polarization response. Feeds are numbered from 0 on each separate antenna for each SPECTRAL WINDOW ID. Consequently, FEED\_ID should be non-zero only in the case of feed arrays, i.e. multiple, simultaneous beams on the sky at the same frequency and polarization. For LOFAR this describes the behaviour of the signal after having the signals from the individual dipoles beam formed within an antenna field. It does not describe the individual dipoles. For LOFAR this in essence describes a virtual device.

**ANTENNA\_ID** Antenna field number, as indexed from ANTENNA<sub>n</sub> in MAIN.

**FEED\_ID** Feed identifier, as indexed from FEED<sub>n</sub> in MAIN.

**SPECTRAL\_WINDOW\_ID** Spectral window identifier: A value of -1 indicates the row is valid for all spectral windows.

**TIME** This value is the mid-point of time interval for which the feed parameters in this row are valid. The same Measure reference used for the TIME column in MAIN must be used.

**INTERVAL** Time interval for which the feed parameters in this row are valid.

**NUM\_RECEPTORS** Number of receptors on this feed. See POLARIZATION\_TYPE for further information.

**BEAM\_ID** Beam identifier. Points to an optional BEAM sub-table defining the primary beam and polarization response for this FEED. A value of -1 indicates that no associated beam response is defined.

**BEAM\_OFFSET** Beam position offset, as defined on the sky but in the antenna reference frame.

**POLARIZATION\_TYPE** Polarization type to which each receptor responds (e.g. "R", "L", "X" or "Y"). This is the receptor polarization type as recorded in the final correlated data (e.g. "RR"); i.e. as measured after all polarization combiners.

**POL\_RESPONSE** Polarization response at the centre of the beam for this feed. Expressed in a linearly polarized basis ( $e_x, e_y$ ) using the IEEE convention. For LOFAR this is (1,0),(0,1)

**POSITION** This is the offset of feed relative to the feed reference position for this antenna (see ANTENNA sub-table). For LOFAR this is always (0,0,0)

**RECEPTOR\_ANGLE** Polarization reference angle. Converts into parallactic angle in the sky domain.

### 3.5 FIELD table

FIELD Table: Field positions for each source					
Name	Format	Units	Measure	Default	Comments
<i>Data</i>					
NAME	String				Name of field
CODE	String				Special characteristics
TIME	Double	s	EPOCH		Time origin for the directions and rates
NUM_POLY	Int			0	Polynomial series order
DELAY_DIR	Double (2, NUM_POLY+1)	rad	DIRECTION		Direction of delay centre
PHASE_DIR	Double (2, NUM_POLY+1)	rad	DIRECTION		Direction of Phase centre
REFERENCE_DIR	Double (2, NUM_POLY+1)	rad	DIRECTION		Direction of Reference centre
SOURCE_ID	Int			-1	Index in SOURCE table
<i>Flag information</i>					
FLAG_ROW	Bool			False	Row flag

Notes: The FIELD table defines a field position on the sky. For interferometers, this is the correlated field position.

**NAME** Field name: user specified in proposal or MoM.

**CODE** Field code indicating special characteristics of the fields, user specified.

**TIME** Time reference for the directions and rates. Required to use the same TIME Measure reference as in MAIN. We use the start time of the observation.

**NUM\_POLY** Series order for the \* DIR columns. This can be used to describe time-variant behaviour of these direction columns. Time origin is given by the TIME field. Default value is 0 (no polynomial expansion).

**DELAY\_DIR** Direction of delay centre; can be expressed as a polynomial in time, but we will only give constant values. Final result converted to the defined Direction Measure type.

**PHASE\_DIR** Direction of phase (fringe stopping) centre; can be expressed as a polynomial in time, but we will only give constant values. Final result converted to the defined Direction Measure type.

**REFERENCE\_DIR** Reference centre: This can be expressed as a polynomial in time, but we will only give constant values. Final result converted to the defined Direction Measure type. For interferometric data, this is the original correlated field centre, and may equal DELAY\_DIR or PHASE\_DIR.

**SOURCE\_ID** Points to an entry in the (optional) SOURCE subtable, a value of -1 indicates there is no corresponding source defined.

**FLAG\_ROW** True if data in this row are invalid, else False. Does not imply flagging in MAIN.

When doing position mosaic, the number of entries in this table will equal the number of mosaic positions.



### 3.6 OBSERVATION table

OBSERVATION Table: Observation information					
Name	Format	Units	Measure	Default	Comments
<i>Data</i>					
TELESCOPE_NAME	String			LOFAR	
TIME_RANGE	Double(2)	s	EPOCH		Specified start/end times
OBSERVER	String			Observer	Name of observer
LOG	String (*)				Observing log
SCHEDULE_TYPE	String			LOFAR	
SCHEDULE	String (*)			[corrSchedule]	Project Schedule
PROJECT	String			NONE	Project identification
LOFAR_PROJECT_TITLE	String				Project description
LOFAR_PROJECT_PI	String				Primary Investigator
LOFAR_PROJECT_CO_I	String				Co Investigators
LOFAR_PROJECT_CONTACT	String				Contact Author
LOFAR_OBSERVATION_ID	String				Observation ID
LOFAR_OBSERVATION_START	Double	s	EPOCH		Observation start
LOFAR_OBSERVATION_END	Double	s	EPOCH		Observation end
LOFAR_OBSERVATION_FREQUENCY_MAX	Double	MHz			Maximum frequency
LOFAR_OBSERVATION_FREQUENCY_MIN	Double	MHz			Minimum frequency
LOFAR_OBSERVATION_FREQUENCY_CENTER	Double	MHz			Centre Frequency
LOFAR_SUB_ARRAY_POINTING	Int				SubArrayPointing
LOFAR_ANTENNA_SET	String				SAS AntennaSet
LOFAR_FILTER_SELECTION	String				SAS FilterSelection
LOFAR_CLOCK_FREQUENCY	Double	MHz			SAS Clock setting
LOFAR_NOF_BITS_PER_SAMPLE	Int				NrBitsPerSample
LOFAR_TARGET	String (*)				Single or list
LOFAR_SYSTEM_VERSION	String				As per ICD
LOFAR_PIPELINE_NAME	String				Pipeline identification
LOFAR_PIPELINE_VERSION	String				Pipeline version
LOFAR_FILENAME	String				As per ICD
LOFAR_FILETYPE	String			uv	As per ICD
LOFAR_FILEDATE	Double	s	EPOCH		As per ICD
RELEASE_DATE	Double	s	EPOCH		Target release date
<i>Flag information</i>					
FLAG_ROW	Bool			False	Row flag

Notes: This table contains information specifying the observing instrument or epoch. It is indexed directly from MAIN via OBSERVATION\_ID.

**TELESCOPE\_NAME** Telescope name ("LOFAR").

**TIME\_RANGE** The start and end times of the overall observing period spanned by the actual recorded data in MAIN. Required to use the same TIME Measure reference as in MAIN.

**OBSERVER** The name(s) of the observer(s).

**LOG** The observing log, as supplied by the telescope or instrument.

**SCHEDULE\_TYPE** The schedule type, with current reserved types ("VLBA-CRD", "VEX", "WSRT", "ATNF"). "LOFAR" for LOFAR.

**SCHEDULE** Unmodified schedule file, of the type specified, and as used by the instrument.

**PROJECT** Project code (e.g. "LEA1234")

**LOFAR\_PROJECT\_TITLE** Description of the project (e.g. "Polarization properties of 3C1234")

**LOFAR\_PROJECT\_PI** Project primary investigator

**LOFAR\_PROJECT\_CO\_I** comma separated list of co-investigators

**LOFAR\_PROJECT\_CONTACT** e-mail address of the contact author

**LOFAR\_OBSERVATION\_ID** Observation ID as used by SAS, MAC and MOM based on the SAS VIC tree number.

**LOFAR\_OBSERVATION\_START** Observation start time/date. Doesn't need to be the same as for this specific SubArrayPointing or MeasurementSet.

**LOFAR\_OBSERVATION\_END** Observation end time/date. Doesn't need to be the same as for this specific SubArrayPointing or MeasurementSet.

**LOFAR\_OBSERVATION\_FREQUENCY\_MAX** Observation maximum frequency. Not for a single spectral window, but for the entire observation.

**LOFAR\_OBSERVATION\_FREQUENCY\_MIN** Observation minimum frequency. Not for a single spectral window, but for the entire observation.

**LOFAR\_OBSERVATION\_FREQUENCY\_CENTER** Observation center frequency. Not for a single spectral window, but for the entire observation.

**LOFAR\_SUB\_ARRAY\_POINTING** Number of the SubArrayPointing used for this measurement within the SAS observation

**LOFAR\_ANTENNA\_SET** SAS AntennaSet setting (e.g. "LBA\_SPARSE\_INNER", "HBA\_ONE")

**LOFAR\_FILTER\_SELECTION** SAS FilterSelection setting (e.g. "10-90 MHz")

**LOFAR\_CLOCK\_FREQUENCY** SAS ClockFrequency setting (e.g. "160 Mhz")

**LOFAR\_NOF\_BITS\_PER\_SAMPLE** SAS NrBitsPerSample setting: 4, 8 or 16

**LOFAR\_TARGET** Single or list of targets. This should basically be a condensed version of the SOURCES table. Can be empty.

**LOFAR\_SYSTEM\_VERSION** Data processing system name and version number

**LOFAR\_PIPELINE\_NAME** Pipeline processing name. If there are multiple runs only the first one will be identified here, further processing details can still be found in the HISTORY table

**LOFAR\_PIPELINE\_VERSION** Pipeline version

**LOFAR\_FILENAME** File name

**LOFAR\_FILETYPE** File type "uv" for MeasurementSets

**LOFAR\_FILEDATE** File creation date

**RELEASE\_DATE** Project release date. This is the date on which the data may become public if the project finishes as scheduled.

**FLAG\_ROW** Row flag. True if data in this row is invalid, but does not imply any flagging in MAIN.

Notes: The OBSERVATION\_NOF\_STATIONS and OBSERVATION\_STATIONS\_LIST as used in the ICDs are not used here, as the information is readily available from the LOFAR\_STATION table.



### 3.7 POINTING table

POINTING Table: Antenna Pointing information					
Name	Format	Units	Measure	Default	Comments
<i>Key</i>					
ANTENNA_ID	Int				
TIME	Double	S	EPOCH		Interval midpoint
INTERVAL	Double	S			Interval duration
<i>Data</i>					
NAME	String				Pointing position descriptor
NUM_POLY	Int			0	Polynomial series order
TIME_ORIGIN	Double	S	EPOCH		Origin for polynomial.
DIRECTION	Double (2, NUM POLY+1)	Rad	DIRECTION		Antenna pointing direction (J2000)
TARGET	Double (2, NUM POLY+1)	Rad	DIRECTION		Target direction (J2000)
TRACKING	Bool			True	True if on-position

Notes: This table contains information concerning the primary pointing direction of each antenna as a function of time. Note that the pointing offsets for individual feeds on a given antenna are specified in the FEED sub-table with respect to this pointing direction.

**ANTENNA\_ID** Antenna identifier, as specified by ANTENNA in MAIN.

**TIME** This is the mid-point of the time interval for which the information in this row is valid. Required to use the same TIME Measure reference as in MAIN.

**INTERVAL** Time interval during which the information in this row is valid.

**NAME** Pointing direction name; user specified. Leave empty.

**NUM\_POLY** Series order for the polynomial expressions in DIRECTION and POINTING OFFSET. Default value is 0 (no polynomial expansion)

**TIME\_ORIGIN** Time origin for the polynomial expansions. Equal this to the start time of the observation.

**DIRECTION** Antenna pointing direction, optionally expressed as polynomial coefficients. The final result is interpreted as a Direction Measure using the specified Measure reference. Use RA, Dec in J2000.

**TARGET** Target pointing direction, optionally expressed as polynomial coefficients. The final result is interpreted as a Direction Measure using the specified Measure reference. This is the true expected position of the source, including all coordinate corrections such as precession, nutation etc. For now, equal this to DIRECTION.

**TRACKING** True if tracking the nominal pointing position.

### 3.8 POLARIZATION table

POLARIZATION Table: Polarization setup information					
Name	Format	Units	Measure	Default	Comments
<i>Data description</i>					
NUM_CORR	Int			4	Nr. Of correlation cross-products
<i>Data</i>					
CORR_TYPE	Int (NUM_CORR)				Polarization of correlation
CORR_PRODUCT	Int (2, NUM_CORR)				Receptor cross products
<i>Flag information</i>					
FLAG_ROW	Bool			False	Row flag

Notes: This table defines the polarization labelling of the DATA array in MAIN, and is directly indexed from the DATA DESCRIPTION table via POLARIZATION ID.

**NUM\_CORR** The number of correlation polarization products. For example, for (RR) this value would be 1, for (RR, LL) it would be 2, and for (XX, YY, XY, YX) it would be 4, etc.

**CORR\_TYPE** An integer for each correlation product indicating the Stokes type as defined in the CASA Stokes class enumeration. For (XX, YY, XY, YX) this is (9,10,11,12).

**CORR\_PRODUCT** Pair of integers for each correlation product, specifying the receptors from which the signal originated. The receptor polarization is defined in the POLARIZATION TYPE column in the FEED table. An example would be (0,0), (0,1), (1,0), (1,1) to specify all correlations between two receptors.

**FLAG ROW** Row flag. True is the data in this row are not valid, but does not imply the flagging of any DATA in MAIN.

### 3.9 PROCESSOR table

PROCESSOR Table: Processor (backend) information					
Name	Format	Units	Measure	Default	Comments
<i>Data</i>					
TYPE	String			CORRELATOR	
SUB_TYPE	String			CEP	
TYPE_ID	Int			0	
MODE_ID	Int			0	
<i>Flag information</i>					
FLAG_ROW	Bool			False	Row flag

Notes: This table holds summary information for the back-end processing device used to generate the basic data in the MAIN table. Such devices include correlators, radiometers, spectrometers, pulsar-timers, amongst others.

**TYPE** Processor type; reserved keywords include ("CORRELATOR" - interferometric correlator; "SPECTROMETER" - single-dish correlator; "RADIOMETER" - generic detector/integrator; "PULSAR-TIMER" - pulsar timing device). For LOFAR this will be 'CORRELATOR'

**SUB\_TYPE** Processor sub-type, e.g. "GBT" or "JIVE". For LOFAR 'CEP' will be used.

**TYPE\_ID** Index used in a specialized sub-table named as subtype type, which contains time-independent processor information applicable to the current data record (e.g. a JIVE CORRELATOR sub-table). Time-dependent information for each device family is contained in other tables, dependent on the device type.

**MODE ID** Index used in a specialized sub-table named as subtype type mode, containing information on the processor mode applicable to the current data record. (e.g. a GBT SPECTROMETER MODE table).

**FLAG\_ROW** Row flag. True if data in the row is not valid, but does not imply flagging in MAIN.

### 3.10 SOURCE table (Optional)

SOURCE Table: Source information					
Name	Format	Units	Measure	Default	Comments
<i>Key</i>					
SOURCE_ID	Int				
TIME	Double	s	EPOCH		Midpoint of interval
INTERVAL	Double	s			Duration of interval
SPECTRAL_WINDOW_ID	Int				
<i>Data description</i>					
NUM_LINES	Int				Number of Spectral lines
<i>Data</i>					
NAME	String				Name of source during observation
CALIBRATION_GROUP	Int				Group number for calibration purposes
CODE	String				Special characteristics of source, e.g. Bandpass calibrator
DIRECTION	Double(2)	rad	DIRECTION		Direction of Source
PROPER_MOTION	Double(2)	rad/s		0	

Notes: This table contains time-variable source information, optionally associated with a given FIELD ID.

**SOURCE\_ID** Source identifier ( $\geq 0$ ), as specified in the FIELD sub-table.

**TIME** This is the mid-point of the time interval for which the data in this row is valid. Required to use the same TIME Measure reference as in MAIN.

**INTERVAL** Time interval for which the data in this row is valid.

**SPECTRAL\_WINDOW\_ID** Spectral window identifier: A -1 indicates that the row is valid for all spectral windows.

**NUM\_LINES** Number of spectral line transitions associated with this source and SPECTRAL\_WINDOW\_ID combination.

**NAME** Source name: User specified.

**CALIBRATION\_GROUP** Calibration group number to which this source belongs; user specified.

**CODE** Source code, used to describe any special characteristics of the source, such as the nature of a calibrator. Reserved keyword, including ("BANDPASS CAL").

**DIRECTION** Source direction at this TIME.

**PROPER MOTION** Source proper motion at this TIME.

### 3.11 SPECTRAL\_WINDOW table

SPECTRAL WINDOW Table: Frequency setup information					
Name	Format	Units	Measure	Default	Comments
<i>Data description</i>					
NUM_CHAN	Int			256	Number of Spectral channels
<i>Data</i>					
NAME	String			SB-x	Subband name/identifier
REF_FREQUENCY	Double	Hz	FREQUENCY		Reference Frequency
CHAN_FREQ	Double (NUM_CHAN)	Hz	FREQUENCY		Centre frequencies for each channel in the data matrix.
CHAN_WIDTH	Double (NUM_CHAN)	Hz			Channel width for each channel in the data matrix.
MEAS_FREQ_REF	Int			5	FREQUENCY Measure ref.
EFFECTIVE_BW	Double (NUM_CHAN)	Hz			The effective noise bandwidth of each spectral channel
RESOLUTION	Double (NUM_CHAN)	Hz			The effective spectral resolution of each channel
TOTAL_BANDWIDTH	Double	Hz			Total bandwidth for this Spectral window
NET_SIDE BAND	Int			0	Upper or lower
IF_CONV_CHAIN	Int			0	The IF conversion chain
FREQ_GROUP	Int			-1	Not defined for LOFAR
FREQ_GROUP_NAME	String				Same as FILTER_SELECTION in OBSERVATION table
<i>Flag information</i>					
FLAG_ROW	Bool			False	Row flag

Notes: This table describes properties for each defined spectral window. A spectral window is both a frequency label for the associated DATA array in MAIN, but also represents a generic frequency conversion chain that shares joint physical properties and makes sense to calibrate as a single entity.

**NUM\_CHAN** Number of spectral channels (must be equal throughout the MS). For LOFAR this number always is 256.

**NAME** Spectral window name: For LOFAR we use SBx where x is a number starting from 0 that indicates the subband number used. A LOFAR MS may have multiple subbands, but a subband can only be in a single MS.

**REF\_FREQUENCY** The reference frequency. This is the frequency representative of this spectral window, usually the sky frequency corresponding to the DC edge of the baseband. Used by the calibration system if a fixed scaling frequency is required or in algorithms to identify the observing band. Note: For the WSRT this field gives the *midband* sky frequency of the IF band (NOT the frequency corresponding to the DC edge of the IF band, as stated in the MS2 definition). It is also the frequency of channel number  $(NUM\_CHAN/2) + 1$  in the array given in field CHAN\_FREQ.

**CHAN\_FREQ** Centre frequencies for each channel in the data matrix. Note that the channel frequencies may be in ascending or descending frequency order.

**CHAN\_WIDTH** Nominal channel width of each spectral channel. Although these can be derived from CHAN\_FREQ by differencing, it is more efficient to keep a separate reference to this information.

**MEAS\_FREQ\_REF** Frequency Measure reference for CHAN\_FREQ. This allows a row-based reference for this column in order to optimize the choice of Measure reference when Doppler tracking is used. Modified only by the MS access code. The number is the enum value as given in the AIPS++ class MFrequency (5 is TOPO).

**EFFECTIVE\_BW** The effective bandwidth of each spectral channel. Usually equals CHAN\_WIDTH.

**RESOLUTION** The effective spectral resolution of each channel (may be different from CHAN\_WIDTH if some form of tapering has been applied).

**TOTAL\_BANDWIDTH** The total bandwidth for this spectral window.

**NET\_SIDE BAND** The net sideband (upper or lower) for this spectral window.

**IF\_CONV\_CHAIN** Site specific identification of the electronic signal path for the case of multiple (simultaneous) lfs (e.g. VLA: AC=0, BD=1, ATCA: Freq1=0, Freq2=1). Not relevant for LOFAR and always 0.

**FREQ\_GROUP** A frequency group identifier to which the spectral window belongs. This is used to associate spectral windows for joint calibration purposes. For LOFAR this is not defined.

**FREQ\_GROUP\_NAME** The frequency group name. For LOFAR this contains the same value as the LOFAR\_FILTER\_SELECTION in the OBSERVATION table.

**FLAG\_ROW** True if the row does not contain valid data.

### 3.12 STATE table

STATE Table: State information					
Name	Format	Units	Measure	Default	Comments
<i>Data</i>					
SIG	Bool			True	Signal
REF	Bool			False	Reference
CAL	Double	K		0	Noise calibration
LOAD	Double	K		0	Load temperature
SUB_SCAN	Int			0	Sub-scan number
OBS_MODE	String			Empty	Observing mode
<i>Flag information</i>					
FLAG_ROW	Bool			False	Row flag

Notes: This table defines the state parameters for a particular data record as they refer to external loads, calibration sources or references, and also characterizes the observing mode of the data record, as an aid to defining the scheduling heuristics. It is indexed directly via STATE ID in MAIN.

**SIG** True if the source signal is being observed.

**REF** True for a reference phase.

**CAL** Noise calibration temperature (zero if not added).

**LOAD** Load temperature (zero if no load).

**SUB\_SCAN** Sub-scan number ( $\geq 0$ ), relative to the SCAN\_NUMBER in MAIN. Used to identify observing sequences.

**OBS\_MODE** Observing mode: defined by a set of reserved keywords characterizing the current observing mode (e.g. "OFF-SPECTRUM"). It is used to define the schedule strategy.

**FLAG\_ROW** True if the row does not contain valid data. Does not imply flagging in MAIN.

This table is compulsory, but has no direct relevance to LOFAR. Its entries will not be used in any post-processing tool as far as I am aware. We may use it eventually for simple calibration information (e.g. LOAD temperature), but currently it's not has no use within LOFAR.

### 3.13 LOFAR\_ANTENNA\_FIELD table

LOFAR_DIPOLE Table: Dipole information					
Name	Format	Units	Measure	Default	Comments
<i>Data</i>					
NAME	String				Either HBA, LBA, HBA0 or HBA1
ANTENNA_ID	Int				Pointer to ANTENNA table
POSITION	Double(3)	m	POSITION		Position of antenna field
COORDINATE_SYSTEM	Double(3,3)	m	DIRECTION		P,Q,R
TILE_ROTATION	Double	rad			
TILE_ELEMENT_OFFSET	Double(3,16)	m	POSITION		
ELEMENT_OFFSET	Double(3,nelements)		POSITION		Offset per element
ELEMENT_FLAG	Bool(2,nelements)				Flag per element
<i>Flag information</i>					
FLAG_ROW	Bool			False	Row flag

Notes: This is a custom table for use by LOFAR only. It contains information about the one or more antenna field that form each Antenna Field. It uses the ANTENNA\_ID to specify to which Antenna Field the antenna field belongs.

**NAME** Name of the antenna field, currently either HBA, LBA, HBA0 or HBA1.

**ANTENNA\_ID** Index into the ANTENNA table to show to which antenna field this dipole belongs.

**POSITION** Position of the antenna field in absolute ITRF coordinates.

**COORDINATE\_SYSTEM** (cartesian) direction vectors in ITRF (or measure defined) describing the local field coordinate system (see "LOFAR reference plane and reference direction", Michiel Brentjens).

This is the up direction of the antennas, and is supposed to be the normal direction to the antenna field plane. Note that in general this is not the zenith direction, and can be many degrees away from zenith for some antenna fields. You need to know this direction in order to do the antenna beam pattern directions. This also contains the information needed to determine the +X and +Y polarization directions.

**TILE\_ROTATION** This is the angl at which the antenna pattern is pointing, in the ground plane of the antenna field. This affects where the HBA tile grating lobes will appear on the sky among other things. In principle this is redundant if all tile/dipole positions are known, but that's non-trivial it's and therefore this is included as a useful reference. NB. Only filled in for antenna fields of type HBA.

**TILE\_ELEMENT\_OFFSET** Position offsets in ITRF (or measure defined) of the dual dipole elements inside a tile, with respect to the centre of the tile. NB. Only valid for antenna fields of type HBA.

**ELEMENT\_OFFSET** Relative offsets of each element from POSITION.

**ELEMENT\_FLAG** Two flags for each row in ELEMENT\_OFFSET identifying if a dipole (X or Y) in an element is operational.

**FLAG\_ROW** True if the row does not contain valid data. Does not imply flagging in MAIN.

### 3.14 LOFAR\_ELEMENT\_FAILURE table

LOFAR_ELEMENT_FAILURE Table: Dipole/tile failure information					
Name	Format	Units	Measure	Default	Comments
<i>Data</i>					
ELEMENT_INDEX	Int				Pointer to LOFAR_ELEMENT table
TIME	Double	s	EPOCH		Time of failure
<i>Flag information</i>					
FLAG_ROW	Bool			False	Row flag

Notes: This is a custom table for use by LOFAR only. It contains information about the failure of separate dipoles/tiles of each antenna field. It uses the ELEMENT\_ID to specify to which dipole/tile failed.

**ELEMENT\_INDEX** Index of the row in the ELEMENT\_OFFSET array in the LOFAR\_ANTENNA\_FIELD table to identify which dipole/tile failed.

**TIME** Time when the failure was detected. Currently updated in 15 second intervals.

**FLAG\_ROW** True if the row does not contain valid data. Does not imply flagging in MAIN.

### 3.15 LOFAR\_STATION table

LOFAR_STATION Table: Station information					
Name	Format	Units	Measure	Default	Comments
<i>Data</i>					
NAME	String				Name of the station
CLOCK_ID	Int				Index of shared clock
<i>Flag information</i>					
FLAG_ROW	Bool			False	Row flag

Notes: This is a custom table for use by LOFAR only. It contains a list of all the stations that were present in the observation, mainly so the ANTENNA table can reference with the purpose of identifying which antenna fields are part of which station. It is indexed directly from ANTENNA though STATION\_ID. It currently only contains a few fields, but will probably be extended in the future.

**NAME** Name of the station. (e.g. "CS103", "DE602")

**CLOCK\_ID** Number identifying a clock signal. It should be the same number for stations sharing a clock signal.

**FLAG\_ROW** True if the row does not contain valid data. Does not imply flagging in MAIN.

## 3.16 HISTORY table

HISTORY Table: History information					
Name	Format	Units	Measure	Default	Comments
<i>Key</i>					
Time	Double	s	EPOCH		Time-stamp for message
OBSERVATION_ID	Int				Points to OBSERVATION table
<i>Data</i>					
MESSAGE	String				Log message
PRIORITY	String				Message priority
ORIGIN	String				Code origin
OBJECT_ID	String				Originating ObjectID
APPLICATION	String				Application name
CLI_COMMAND	String(*)				CLI command sequence
APP_PARAMS	String(*)				Application parameters

Notes: This sub-table contains associated history information for the MS.

**TIME** Time-stamp for the history record. Required to have the same TIME Measure reference as used in MAIN.

**OBSERVATION\_ID** Observation identifier. This is an index into the OBSERVATION table.

**MESSAGE** Log message.

**PRIORITY** Message priority with allowed types: DEBUGGING, WARN, NORMAL, SEVERE

**ORIGIN** Source code origin from which the message originated. Contains version/revision number.

**OBJECT\_ID** Origination ObjectID if available, else blank.

**APPLICATION** Application name.

**CLI\_COMMAND** CLI command sequence invoking the application.

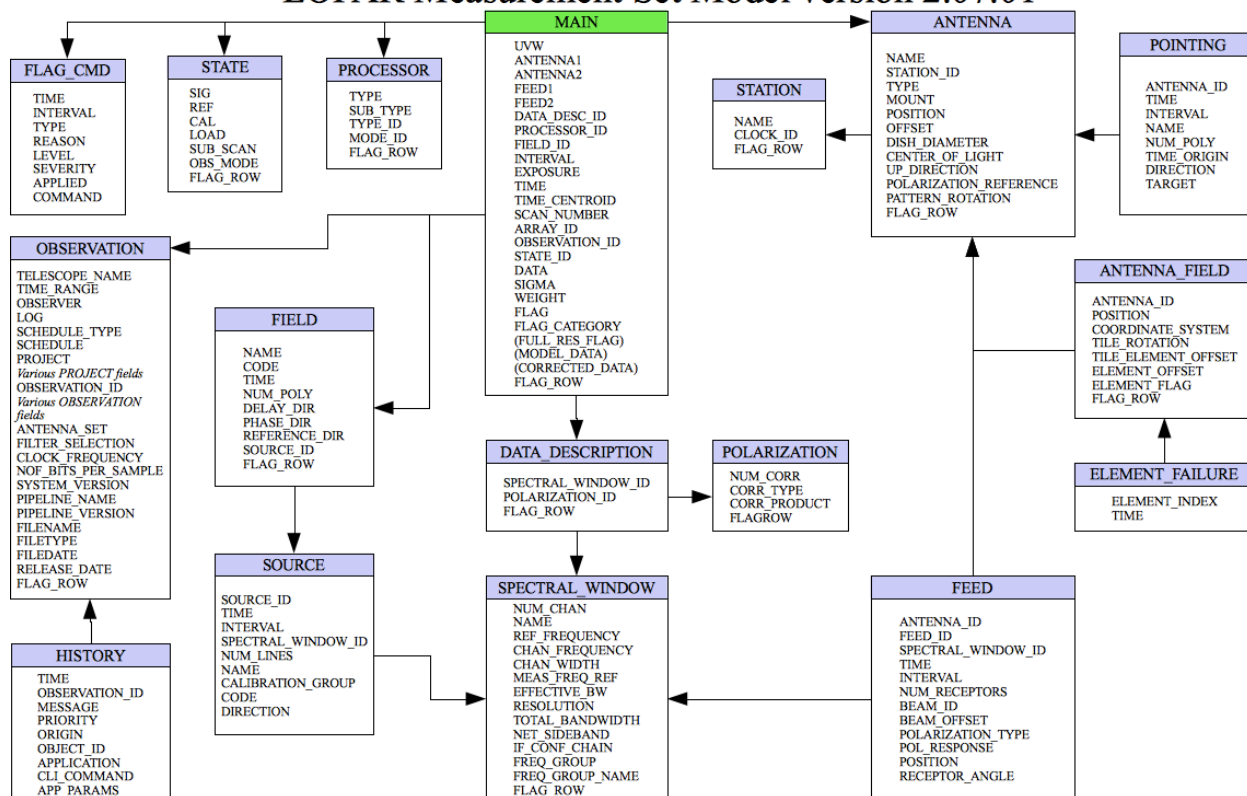
**APP\_PARAMS** Application parameter values. For LOFAR this is an array of strings with the parset key-value pair values.



## Schema of the LOFAR MS

The figure below shows a graphical representation of the LOFAR Measurement Set. All subtables and columns in the MS are presented.

LOFAR Measurement Set Model version 2.07.01



LOFAR\_ prefixes have been omitted for clarity

Figure 1: Schema of the LOFAR Measurement Set

## A Appendix A: Translating the parset file.

The storage process that writes the MS obtains most of its meta-data through parset files. In this Appendix I will present how the translations are being done, and which Measurement Set fields are filled with which parset parameter values.

The Parset files are ASCII files containing lines of the type 'keyword = value(s)'. These are the basic command files for all LOFAR processes. Eventually the parset files are produced for each observation directly from SAS, but currently this is still a manual process.

A typical parset file contains many lines. The majority is used to setup the connections between the various hardware pieces (e.g., through MAC or IP addresses). Some of the parameters are related to the observation to be conducted. An example is given below (from the parset file L2006\_00417.MS.parset):

```

Observation.BeamDirections = [0, 0.92]
Observation.DirectionType = J2000
Observation.NBeams = 1
Observation.NChannels = 256
Observation.NPolarisations = 2
Observation.NStations = 16
Observation.NSubbandSamples = 155648
Observation.NSubbands = 1
Observation.PositionType = ITRF
Observation.RefFreqs = [60000000]
Observation.SampleRate = 156250
Observation.StartTime = 1164387436
Observation.StationPositions = [0.119879729, 0.923509308999999994, -
472.8850689999999999, 0.119887278, 0.923512320000000005, -472.884916999999997,
0.119878255, 0.923508922999999998, -472.885050999999998, 0.119884052,
0.923510786000000003, -472.885062, 0.119851103, 0.923481756000000004, -
472.882482999999998, 0.11985772, 0.923485596000000002, -472.882901,
0.119864071, 0.923481595000000004, -472.882206, 0.119857454,
0.923477756000000004, -472.881766000000003, 0.119868258000000001,
0.923536131999999995, -472.887060000000002, 0.119874876000000001,
0.923539972000000004, -472.886810000000003, 0.11989796, 0.923531999000000005, -
472.887468000000001, 0.119891342, 0.923528160000000004, -472.886754999999999,
0.119958428000000001, 0.923509286000000004, -472.885465000000001,
0.119965046000000001, 0.923513125000000005, -472.885848000000001,
0.119971396999999999, 0.923509125000000004, -472.885186999999997,
0.119964778999999999, 0.923505284999999995, -472.884425000000002]
Observation.StopTime = 1164402434
Storage.IntegrationTime = 60
Storage.MSName = /data/L2006_00417.MS
Storage.StorageStationNames = ['CS10_dipole0', 'CS10_dipole4',
'CS10_dipole8', 'CS10_dipole12', 'CS01_dipole0', 'CS01_dipole4',
'CS01_dipole8', 'CS01_dipole12', 'CS08_dipole0', 'CS08_dipole4',
'CS08_dipole8', 'CS08_dipole12', 'CS16_dipole0', 'CS16_dipole4',
'CS16_dipole8', 'CS16_dipole12']

```

How and where are these fields entered in the MS? Let's pick out the associated lines of the parset file.

## A.1 Pointing information

```
Observation.BeamDirections = [0, 0.92]
Observation.DirectionType = J2000
Observation.NBeams = 1
```

These fields give the RA, Dec coordinates of the beam in rad. They are entered in the following MS tables and fields.

- Table *Pointing*; Columns *Direction*, *Target*; Observation.BeamDirections. The Observation.BeamDirections field goes into the columnheader as keyword.
- Table *Field*; Columns *Delay\_Dir*, *Phase\_Dir*, *Reference\_Dir*. The Observation.DirectionType field goes into the columnheader as keyword.
- Table *Source*; Column *Direction*; Observation.BeamDirections. The Observation.DirectionType field goes into the columnheader

As there is no source/target name supplied in the parset file, these are defaulted to "Beam-0". This is entered in the *Field::Name* field.

## A.2 Antenna information

Details of the Antennae used are observation dependent . The relevant parset lines are:

```
Observation.PositionType = ITRF
Observation.NStations = 16
Observation.StationPositions = [0.119879729, ... etc...]
Storage.StorageStationNames = ['CS10_dipole0', etc...]
```

This information is fed into the ANTENNA table, in the following columns:

*Name*: Storage.StorageStationNames

*Position*: Observation.StationPositions (preferably in X,Y,Z (m) ITRF; this comes from Observation.PositionType and must be fed into the *Position* column header keywords) .

## A.3 Frequency information

The frequency setup is handed over through the parset fields:

```
Observation.SampleRate = 156250
Observation.NChannels = 256
Observation.NSubbands = 1
Observation.RefFreqs = [60000000]
```

This is translated into the *SPECTRAL\_WINDOW* table as follows:

The number of rows in the *SPECTRAL\_WINDOW* table is given by the Observation.NSubbands; this is only for LOFAR-CS1 at this point. Later, the data will be distributed over the 12 storage nodes with each node handling a part of the total number of subbands.

The Observation.RefFreqs field gives the reference frequency, or the midpoint frequency of the observation for each subband.

The total bandwidth of the subband is given by Observation.SampleRate. This can either be 156250, or ????. This is in Hz.

From these numbers all information in the *SPECTRAL\_WINDOW* table is determined:

*Num\_chan*: Observation.NChannels.

*Total\_bandwidth*: Observation.SampleRate (in Hz).

*Ref\_frequency*: `Observation.RefFreqs` (this is in Hz).

*Chan\_width*: Array of `Observation.NChannels` values which contains `Observation.SampleRate` divided by `Observation.NChannels` (in Hz).

*Effective\_bw* and *Resolution* are equal to *Chan\_width*.

*Chan\_freq*: Array of `Observation.NChannels` values. Startvalue is *Ref\_frequency* minus

$Total\_bandwidth / 2 + (Total\_bandwidth / Num\_chan) / 2$ . It is assumed that the *Ref\_frequency* is the value at the bottom side of channel `Observation.NChannels/2` (first channel is channel 0).

*Name* contains the string SB-x, where x is the subband number.

## A.4 Time issues

Samples are correlated at a fundamental rate (determined by the samplerate at the stations). Before storing the data in the MS, a number of samples are integrated and averaged to a given integration time. The integration time (or rather, the number of samples to be integrated before writing them into the MS as a single sample) is user-defined in the parset file. Starttime and endtime of the observation are also user defined and handed over in the parset file, and used at several locations in the MS. The parset lines related to time keeping are:

```
Observation.NSubbandSamples = 155648
Observation.SampleRate = 156250
Observation.StartTime = 1164387436
Observation.StopTime = 1164402434
Storage.IntegrationTime = 60
```

The `Observation.StartTime` and `Observation.StopTime` are given in Unix seconds, i.e. seconds since Jan 1, 1970 in UT.

The fundamental correlator integration time is given by dividing `Observation.NSubbandSamples` and `Observation.SampleRate`, which for this example yields 0.9961472 seconds. The number given by `Observation.NSubbandSamples` can be determined as follows:

```
256 * (int)(Observation.SampleRate/(16*256))
```

This is related to the polyphase filtering used and the need to work with integer numbers.

The number of samples integrated into a single row in the MAIN table of the MS is given by `Storage.IntegrationTime`. The total sample integration time in this example is thus: 59.768832 seconds. This is also the separation time between consecutive samples.

Based on these numbers the following fields are filled:

*Main table: Exposure and Interval* (set equal): total sample integration time

*Time and Time\_centroid* (which are set equal): This is `Starttime + Scan_number * Interval`; this is also used to calculate the UVW coordinates of the baseline.

*Observation table: Timerange*: this is `Observation.StartTime`, `Observation.StopTime`

*Pointing table: Time and Interval*: `Observation midtime and duration from Observation.StartTime, Observation.StopTime.`

*Source table: Time and Interval*: as in *Pointing table*.

*Feed table: Time and Interval*: as in *Pointing table*.

*Field table: Time*: `Observation.StartTime`.

## A.5 Polarization

Polarization information is stored at two locations in the MS: The Polarization subtable and the Feed subtable. Indirectly, the number of polarizations used and correlated also determines the dimension of the fields that stores the data and the data-related information.

The parset file only has one field for polarization:

```
Observation.NPolarisations = 2
```

For CS1, the polarization is always linear, thus the polarizations used are (X,Y). This is put in the *Feed* subtable:

*Polarization\_type*: Fixed

*Polarization\_respons*: Fixed

*Receptor\_angle*: Fixed.

*Num\_receptors*: Fixed; equals `Observation.NPolarisations`

The BGL correlator calculates all cross-products and the data in the MS thus contains all of these: XX, XY, YX, YY.

This information is stored in the *Polarization* subtable:

*Corr\_type*: contains the AIPS++-internal code for the four cross products.

*Corr\_product*: contains the matrix that determines the four cross-products from the vector (X,Y).

*Num\_corr*: Gives the number of cross-products (i.e., 4).

## A.6 DataSet naming

The name with which the MS is written to disk is taken directly from the parset file:

```
Storage.MSName = /data/L2006_00417/SBx.MS
```

Note, that in case more than one storage node is used in writing the data (e.g, when multiple subbands are written), this parameter is an array holding the MS names to be used for each storage node.

## Appendix B: Lofar Storage Manager (LofarStMan)

### B.1 Introduction

For the 50-station LOFAR instrument the data rate for a single subband is about 10 MB/s. Because multiple subbands are kept on the same storage node, it is essential that the data can be dumped to disk as fast as possible, thus with hardly any overhead. On the other hand the data should be written in the standard casacore MeasurementSet format to be able to inspect and process them with the standard toolset.

Writing a MeasurementSet in the native way incurs overhead. One reason is that under the hood buffered IO is used; another reason is that several meta data columns are written that are constant for the LOFAR case.

MeasurementSets are handled by the casacore Table System that provides a way to use a dedicated storage manager. When accessing a table, the storage manager can be loaded dynamically provided that the shared library containing the code can be found in the library path.

This feature is used by the LOFAR data writer. It creates the MeasurementSet such that the Table System knows it has to use the special LofarStMan storage manager to understand the file containing the data that are written directly to disk.

### B.2 LOFAR MeasurementSet creation

The data writer creates the MeasurementSet in the standard way, but binds the columns of the main table to the LofarStMan storage manager. It fills all subtables and creates a small file *table.f0meta* in the main table directory containing information like antenna numbers, time, endianness, etc.. It also contains a version number making it possible to evolve the format. The meta file is written in the casacore AipsIO format.

After this initialisation phase the MeasurementSet is closed and the main data file *table.f0data* is created. Data are written to that file when received from the online system. If possible O\_DIRECT is used to avoid the kernel file cache overhead.

In version 1 and 2 each time slot in the file contains three blocks of data:

1. A 32-bit signed integer sequence number. It is used to derive the time.
2. Per channel the number of samples (16-bit unsigned integer) used by the correlator. It is used to derive the weights and flags.
3. The data array (ncorr,nchan) as single precision complex numbers.

The meta file tells how each block is aligned. Usually this is on 512 bytes because that is required to make use of the O\_DIRECT option.

There can be gaps in the sequence numbers, thus time slots might be missing.

In case of a crash, the data are always fine. I.e., no indices or so have to be updated as would be the case if the native table format or a system like HDF5 is used.

Note that LofarStMan is only used for the raw data. The first NDPPP processing step will create a MeasurementSet using the standard storage managers.

## B.3 Accessing an existing LOFAR MeasurementSet

When an existing LOFAR MeasurementSet is opened, the Table System detects that the LofarStMan storage manager is needed and will load it dynamically from the shared library with that name. First a register function is called to make the storage manager known. Thereafter all accesses to the data are done through LofarStMan.

LofarStMan calculates the number of rows in the data file and reports it back to the Table System.

On 64-bit systems the data file is memory-mapped. In this way the system takes care of caching if random access is done. The memory space of 32-bit systems is too small for mmap, so buffered IO is used instead.

The columns handled by LofarStMan and their values are given in the following table. All columns except DATA are readonly. Note that bytes are swapped if the endianness requires so.

TIME	$\text{starttime} + (\text{seqnr} + 0.5) * \text{interval}$
ANTENNA1	from meta file
ANTENNA2	from meta file
FEED1	0
FEED2	0
DATA_DESC_ID	0
PROCESSOR_ID	0
FIELD_ID	0
INTERVAL	interval from meta file
EXPOSURE	interval from meta file
TIME_CENTROID	same as TIME
SCAN_NUMBER	0
ARRAY_ID	0
OBSERVATION_ID	0
STATE_ID	0
UVW	calculated on the fly using FIELD and ANTENNA subtables (for phase reference direction and antenna positions)
DATA	from data file
SIGMA	1
WEIGHT	1
WEIGHT_SPECTRUM	nr of samples / nominal nr of samples
FLAG	nr of samples == 0
FLAG_CATEGORY	empty
FLAG_ROW	false

A MeasurementSet stored with LofarStMan has some special characteristics:

- All columns are readonly with the exception of the DATA column.

- It is not possible to add or remove rows.
- It is possible to remove a column.
- A column can be added to the MS, but only with a standard storage manager.

For example, it is possible to make the FLAG column writable by creating a FLAGX column, copy FLAG to it, remove FLAG, and finally rename FLAGX to FLAG.

## B.4 Lessons learned and future

Besides the gain in write speed, LofarStMan proved successful for another reason:

Accidentally the data writer wrote the conjugates of the data during the first period. This could be solved elegantly by adapting LofarStMan such that version 1 conjugated the data and by setting the version to 2 for the new data writer.

MeasurementSets are often selected or sorted on TIME. This proved to be slow because the entire data file had to be traversed. Therefore a new version 3 will get an optional extra file *table.f0seqnr* containing a copy of the sequence numbers. Normally this file will be present and correspond to the data file size. If so, LofarStMan will take advantage of it. Because the file is written afterwards, it may not be present or correct in case of failures. In that case LofarStMan will get the sequence numbers from the data file.

The only real problem encountered was that an MS stored with LofarStMan could not be used in the CASA package. The reason is that CASA uses different casacore libraries than LofarStMan, so even after it has registered itself, LofarStMan was still not known to CASA. This will be solved if CASA can be built using the same libraries.