ALMA Export Data Format

ALMA-70.00.00.00-004-A-SPE

Version: A

Status: Released

2004-04-14

Prepared By:	Organization	Date
F. Viallefond, R. Lucas	European Southern Observatory	2004-04-14
IPT Leader Approvals:	**Organization**	**Date**
Brian Glendenning Gianni Raffi	National Radio Astronomy Observatory European Southern Observatory	
System Engineering Approvals:	**Organization**	**Date**
Richard Sramek 2004.12.02 12:16:34 -07'00' Christoph Haupt 2004.12.13 14:21:37 +01'00'	National Radio Astronomy Observatory European Southern Observatory	
Configuration Control Board Approval:	**Organization**	**Date**
	ALMA Configuration Control Board Secretary, signing for the Control Board	
JAO Director Release Authorization:	**Organization**	**Date**
	Joint ALMA Office Project Director	

Change Record

Version	Date	Affected Section(s)	Change Request #	Reason/Initiation/Remarks
Draft A	2003-06-24	All		Initial Draft - RL
Draft A	2003-08-21	Sect 1,3.1.2, 4		3 changes from Farris included – FV
Draft A	2003-09-30	All		Multiple revisions – FV
Draft A	2004-04-01	All		Multiple revisions – FV, RL
Draft A	2004-04-14	All		Several Details – FV

Table of Contents

1 Introduction

1.1 Purpose

This document describes the format in which raw science data will be provided to ALMA users. This was preceded by a discussion document (ref. version 1 of the Proposed ALMA Export Data Format) on which the present document is based. The format is able to contain an arbitrary amount of data (a data set) but it is intended to include data with a scientific meaning, like a whole project, parts of it (ObsUnitSets), one or more Execution Blocks, a subset of an Execution Block, data on a source from several related projects, ...

1.2 Scope

Raw data is defined as data available on-line (including its on-line calibration), as input to the Science Pipeline. In addition the Science pipeline will provide image data and science calibration data. In order to keep track of the relationship between the data and the whole ALMA observing process that led to the science results, the structure of the corresponding ALMA project(s) is included in the raw data format itself. Thus we keep track of ObsUnitSets, execution blocks, scans, observations and integrations.

We consider first the data model, i.e. the way the data, metadata, and auxiliary data will be logically organized. We also include a rough description of the actual implementation of the data format.

While for the actual implementation there is a default choice as FITS is the recommended IAU standard, and an emerging format (VOTable), there is at present no recognized standard for the data model. As it is planned to use AIPS++ as the off-line data reduction package for ALMA, as well to provide the basic reduction engines for the ALMA science pipeline, it made sense to consider exporting data in a form that can be easily made available for this package. The data model has thus many characteristics of the Measurement Set data model (Kemball and Wieringa, 2000); however there are notable differences.

2 ALMA Data models

ALMA science data which may be delivered to users include:

Raw data : This is the data directly pertinent to the science, i.e. recorded during the observations of a given project. It contains correlator data, but also metadata and auxiliary data.

Calibration data : These are results of the data processing, which are to be needed to tune the instrument for observing, or as input to the data reduction of science observations.

Image data : These are the direct science results of the observing. They are built on the basis of raw data and calibration data by the science pipeline.

As stated above the present document only describes the raw data model. The Calibration data will be described elsewhere. The image data will follow closely data models well recognized in the astronomical community (FITS/VOTables).

The raw data contains three kinds of data:

Correlator data : this is produced by the correlator and its associated data-processing computers, and represents the bulk of the data volume.

Auxiliary data : Other data monitored at the telescope during the observing, and that will be needed for data reduction. This includes antenna coordinates, tracking errors, atmospheric measurements, WVR data, receiver total powers, ... etc.

Metadata : Data needed to describe the observing process as planned and the resulting correlator and auxiliary data (array description, spectral setup, source information, ... etc).

3 The model for A L M A Raw Data

3.1 Design considerations

We have adopted the strategy to prepare the data, meta data, and auxiliary data as soon as possible in the observing process, in a format which is very close to the one that will be exported to the users.

- During the observing process the meta data and auxiliary data will be delivered in this format to the on-line data reduction systems (on-line calibration, quick look system), after each atomic observation and scan.
- The meta data and auxiliary data will be stored in the Archive at the end of each Execution Block.
- The Science Pipeline will need data sets relevant to individual science results (ObsUnitSets) which contain one or several Execution Blocks.
- The data will be mainly extracted from the archive at the level of Observing Projects or subsets of these, either selected by time (scans, execution blocks), or frequency (individual basebands or frequency windows), or sky space (calibrator sources, target sources).

The actual quantity of observations contained in the data sets, represented in the AEDF format in the Archive, is beyond the scope of the document. Currently the minimal extent is an Execution Block. It could be extended to a full project, with some intermediate possibilities. This operational scope will be guided by requirements proper to the archiving and retrieval needs (among which archival research requirements); probably more weight will be given to the main product (Image data sets).

The data organization of the raw data format was motivated by:

- efficiency of the data writing process
- having the data as soon as possible in the most suitable form for data reduction (thus minimizing transpositions)
- minimization of data volume
- efficiency of conversion into the internal format used by the main data reduction package.

3.2 Data Model description

3.2.1 Logical structure

In essence the data set contents are a large number of data/auxiliary data items, most varying with time while the observations are made. However the time granularity varies greatly from item to item: The astronomical data changes at all integrations, while some parameters are in fact static; it is also desirable to keep together parameters that are closely interrelated and vary together (like receiver parameters, antenna configuration, ...), with similar time granularity. The parameters are thus grouped in Tables.

These tables are composed of rows and columns. The columns are grouped in different sections: *key*, non-*key*, data-description and data sections. The contents of those sections define how the various columns can be used to associate the tables in a locical way, to build up a relational data base: for instance, specifying the contents the *key* columns uniquely define a given row in a table.

A more extensive description of the logical meaning of the table contents, in terms of UML, is given elsewhere .

Each table contains, in addition to the data elements, **identifiers** that are used to link the table entries together. Each row of a given table is uniquely identified by a set of *key*s. These *key*s are either identifiers and/or the TIME coordinate. There is one identifier associated with each table.

The links between the astronomical data and the associated data (meta and auxiliary) are present in the MAIN table. Each entry has three *key*s CONFIG_DESCRIPTION_ID (leading to a description of the telescope/correlator setup), FIELD_ID (leading to a description of the astronomical object), and TIME.

The main astronomical data in each entry are contained in a single data cell which is referred to by a data object identifier (the data itself is usually in a separate le). The shape of the data cell that is referred to is described in the corresponding entry in the CONFIG_DESCRIPTION table. Alternately the data object identifier can be replaced by the data itself.

The complete list of tables in the data model is given in Tab. 2A.

These tables typically contain:

configuration setups of the instrument: antenna, correlator (spectral_window), polarization

hardware characteristics: feed, processor, receiver, caldevice

information about the targets observed: source, ephemeris, doppler

recipes for tracking: pointing, doppler, frequencyoffset, gaintracking

monitored auxiliary data: pointing, weather

descriptions of state sequences: state, field

project execution information: observation_unit, execute_summary, scan_summary, obs_summary, history

There are static or quasi-static information, such as configuration setups, hardware characteristics, state sequences; by contrast the some monitored data are varying continuously with time (pointing, weather).

3.3 Relation to the Measurement Set format

The present data model is derived from the Measurement Set Model in the sense that most of the structure of the auxiliary Tables is kept; only the parameters that have no meaning for ALMA have been omitted, and parameters that are ALMA-specific have been added (placeholders for some of these were actually present in the MSv2.0 document).

One major difference is the relocation of the actual astronomical data outside of the Main Table. This has some advantages for the data flow inside the ALMA Computing system: the bulk of the data is produced in large binary amounts by the correlator data processing computer, and is stored untouched in the archive. The main user of this data is the Science pipeline which will access this binary data in its original form.

A second difference is the introduction of a new Table (the CONFIG_DESCRIPTION table) to identify the sets of antennas and of frequency windows used. These change rarely during a project and this introduction results in a simplification of the Main Table.

Also several items and tables have been renamed to correspond to ALMA terminology (scans, observations, integrations).

3.4 Identifiers

All data elements in the data model can be referenced through **identifiers**. Each identifier can be considered as one integer coordinate in a hyper-cube. Together these coordinates uniquely identify all the information associated to each piece of data. In the following the identifiers are all suffixed by _ID (e.g. ANTENNA_ID).

Any cell in that hyper-cube is an entity which hosts a piece of data. In addition to the astronomical data the cell contains metadata and auxiliary data: many arrays are multidimensional (dimensions being e.g. the number of polarization cross products, the number of spectral channels, ...).

4 Data Format Contents

ALMA will have a wide variety of observing modes. Atmospheric effects are critical, so calibration methods which are specific to millimeter and sub-millimeter radio astronomy are needed; specific devices (radiometers) will be used. Though there is significant experience in the millimeter with existing arrays, sub-millimeter wave interferometry is still in its infancy. The data model must then be flexible enough to prevent potential limitations, so that unforeseen telescope operations, hardware devices, calibration plans, observing modes, data reduction processing, ...etc., present a minimal risk.

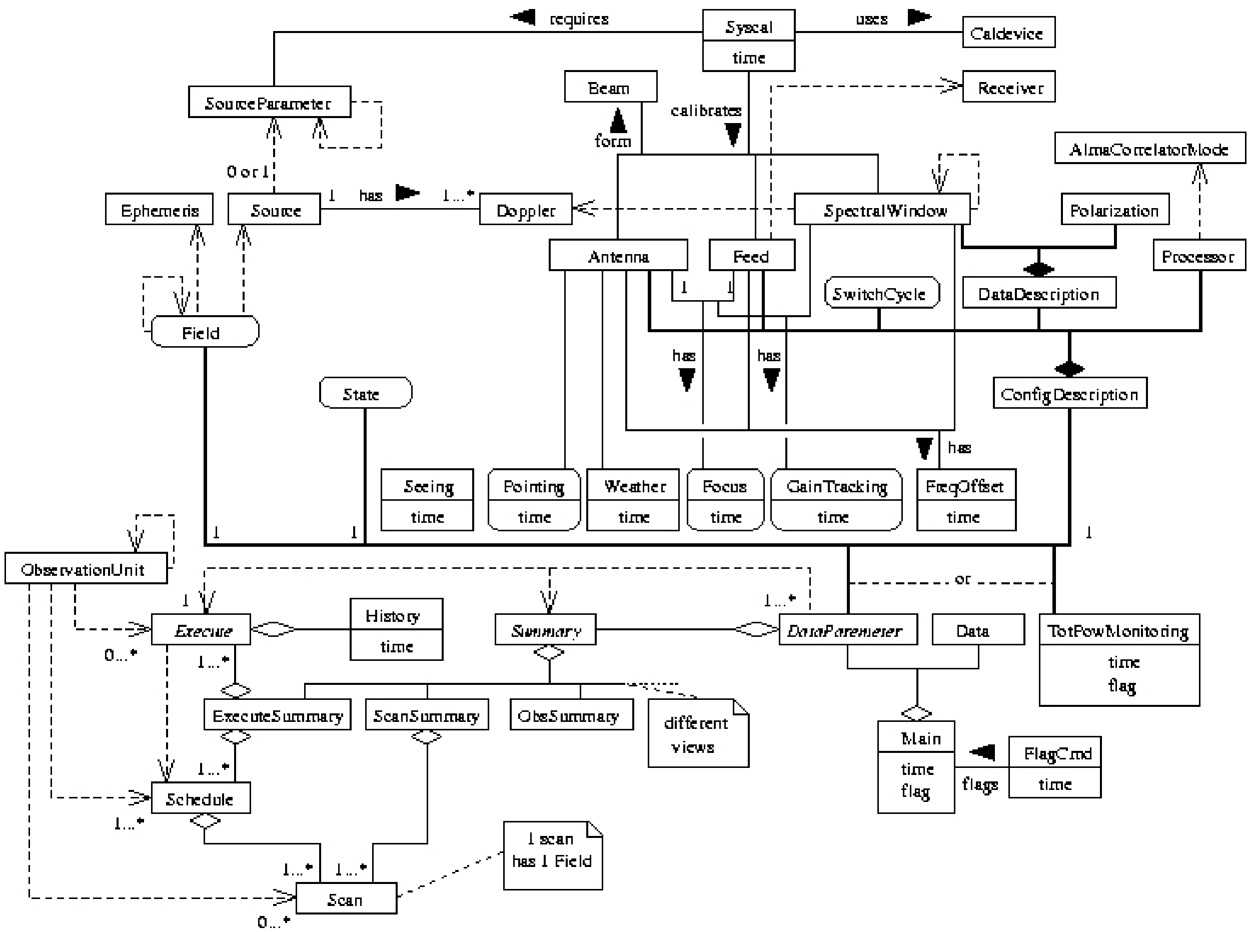

Figure 1: Diagram representation of the AEDF. Dashed arrows indicate dependencies in relationships (i.e. references to identifiers which are present in the Data sections for some MS tables). Filled diamonds highlight compositions. An open diamond marks the component holding the responsibility for a given relation. The filled arrows indicate association with constraints. There are several state components in particular FIELD defining sequences of spatial coordinates, STATE and SWITCH_CYCLE defining temporal sequences associated to some properties. They are represented by rounded rectangles. The Main table is split into two parts, its descriptor columns (*DataParameter*) and the data columns (Data). The "*Key*" section of the MAIN table, a composite, is highlighted by the thick connection in this diagram DataParameter is written in italic to indicate that it is essentially virtual entities.

There are unprecedented aspects with ALMA: a large number of antennas, continuous array reconfiguration, the hardware and methods for the short spacings and the zero spacing, a large instantaneous frequency bandwidth, dynamic scheduling, quasi real time calibration, pipeline processing etc... All these new aspects are demanding for the data model; performance issues are also important.

In this Section we attempt to describe how we extended the original Measurement set for ALMA use.

4.1 ALMA Observing Objects and related structure

4.1.1 Observing Objects

Section 2 of the SSR document defines a hierarchy of observing objects for ALMA data: Project, Scheduling Block, Scan, Observation, Integration, Sub-integration, Switching Cycle. It is required that these observing objects can be easily identified in the exported data as well as in archived data. Here we describe the options we propose to take to achieve this.

Project this is represented in the EXECUTE_SUMMARY [1] table as PROJECT ; the **observer** (the PI) appears also there as OBSERVER .

Programme: An observing project (ObsProject) contains an observing programme, which is, in general, a complex hierarchically structured tree with arbitrarily many levels. This programme contains one or more observing unit sets (ObsUnitSet) or scheduling blocks (SchedBlock). Each of these observing unit sets contains either observing unit sets or scheduling blocks as members. The leaves of the tree are ultimately a series of scheduling blocks. The definition of an observing project, observing unit sets and scheduling blocks are contained in the Observation Preparation Subsystem. It is important to state that this is a tree and not a lattice. Each scheduling block is an immediate member of one and only one observing unit set. A scheduling block is a member of more than one observing unit set only in the sense that its parent may itself be contained as a member in a higher-level observing unit set.

The manner in which a scheduling block is related to an observing unit set is shown in a document discussing the project structure. . Since a scheduling block may have been executed several times for different reasons it is associated to one or several executing blocks. This association is represented in the EXECUTE_SUMMARY table with one EXECUTE per row. The EXECUTE_SUMMARY table contains the full sequence of executing blocks. These EXECUTE define data blocks which are identified in the MAIN table via the EXECUTE_ID identifier. Each executing block must have an intent and this intent may be identical for several executing blocks, those being not necessarily associated to the same scheduling block; this EXECUTE intent is used to define logics in the project. A new table, the OBSERVATION_UNIT table, is required to define associations of data blocks for their logics in the data processing and the scientific output.

Scheduling Block (SB) is indexed in the EXECUTE_SUMMARY table using a SCHEDULE_ID identifier referencing to a row in a SCHEDULE table, one of the tables which will describe the Project Structure. The Execution Blocks (XB) which represent each repeated execution of the same SB can also identified here as EXECUTE. We propose to attach to an EXECUTE an EXECUTEEINTENT to indicate the reason when repeating EXECUTEs; examples would be improve sensitivity and/or UV coverage. With one EXECUTE per row this defines data blocks which will be indexed in the MAIN using an EXECUTE_ID [2] identifier.

Scan Scans are sequences of consecutive observations with a common goal. The goal will be stored in the STATE table in a new data column named OBSINTENT . Examples of attributes would be FOCUS () , POINTING () , SKYDIP () etc...

We propose to store the scan number itself in a column SCAN_NUMBER of the SCAN_SUMMARY table Each scan in the SCAN_SUMMARY table has a time stamp allowing to find, from the project structure tree, on which rows in the main table is that scan. There is no need to reference the SCAN_SUMMARY rows anywhere else in the model. One advantage of the SCAN_SUMMARY table is to provide a quick view and quick data access, for performance reasons. Other kinds of summary tables could be created with different scopes, e.g. CALSUMMARY tables to access efficiently to the rows related to calibration purposes, an ATM SUMMARY table for efficient access to the radiometric data etc...

Observation According to the ALMA definition an observation is a set of time-contiguous integrations, while the antennas complete an elemental pattern. Patterns are described in the FIELD table. However a new observation is not only defined by a change of FIELD_ID as observations may be repeated. If ALMA only identifies observations within the scan (starting from 0 or 1) then there is no real need for a specific column in the Main table (a new observation being identified by INTEG_NUMBER in MAIN restarting from 0 or 1 inside a given scan; alternately one might have added one column in the SCAN_SUMMARY table to keep track of the actual numbers, at the expense of increasing the length of that table).

Integration As integrations are defined as the basic recorded unit, it is natural to have the integration number at the primary level in the main table. Integrations inside the same observation are labeled starting from 1 by the INTEG_NUMBER[3] column in MAIN.

[1]Table originally named OBSERVATION in MS2

[2]Originaly named OBSERVATION _ID in MS2

Sub-integration Sub-Integrations inside a given integration (used for channel averaged output from the correlator, WVR, total-power data, antenna positions, ... are labeled in the STATE table by SUBINTEGNUM[4] . A sub-integration may correspond to a single correlator dump but will be more likely already the sum of several dumps for durations of typically 0.5 to 1 second.

Switching Cycle In some observing modes, there is a switching cycle below the integration/sub-integration level (case for positions of a nutating subreflector and when switching in frequency). A cycle is described in one row of the SWITCH_CYCLE table. This sets the size along a bin axis for each baseband (and eventually antenna for the single-dish mode). The integration/sub-integration will appear as many times as there are states in the switching cycle, each time with the positions in the cycle known from a SWITCH_CYCLE_ID [5] identifier. The corresponding data will have to represent the average results in each state.

For other types of switching modes, e.g. fast switching for phase referencing, position switch etc... which are above the integration/sub-integration level the characteristics of the cycle are in the STATE table. To improve the semantics we propose to add an OBSINTENT in the STATE table which holds the intent of a scan and the attribute for OBSMODE would hold only the observing mode such as the various switching modes.

4.2 Other ALMA specific auxiliary data / meta data

4.2.1 User input

We should have all user (observer) input in the data model, as required by SSR. This includes spectral information (required velocity ranges, and velocity resolution, in each spectral window, as well as spatial information: target areas that are required to be mapped, entered e.g. as polygons.

4.2.2 Data Cell

Each row in the MAIN table has a link (OID) to a data cell, which is the binary data corresponding to an integration or sub-integration.

4.2.2.1 Data Cell for an integration (full channel data):

This is an array with the following seven dimensions, from the fastest varying index to the slowest one:

1. Polarization: in order (XX, YY, XY, YX); size is N_c = 1, 2, or 4.
2. Frequency channel number Size is N_f.
3. Atmospheric pathlength correction: (APC, size N_{apc} is 1 or 2). Coordinate values are 0 for uncorrected phases, 1 for corrected phases.
4. Bin number: Size N_{bin}. This means that the integration led to two different sets of output spectra due to use of e.g. nutator switching or frequency switching. Sideband separation by 90-degree phase switching results in doubling the number of subbands (one set fro each sideband).
5. Window number. Size N_{sw}. Note the we mean here spectral windows after the stitching operations in the correlator software: these do not correspond to the correlator subbands defined in the tunable correlator input lters, but to the spectral windows specified by the astronomical observer.
5. Baseband number: Size is N_{bb}. Different baseband numbers correspond to different second local oscillators and digitizers.
6. Baseline products: Sizes are $N_a(N_a - 1)/2$ and N_a.

In order cross products $A_1.A_2$, $A_1.A_3, A_2.A_3, A_1.A_4, A_2.A_4, A_3.A_4, A_{n-1}.A_n$ followed by the self-products $A_1.A_1, ... A_n.A_n$. $A_1, ... A_n$ refer the the N_a antennas present in the collection (as defined in the CONFIG_DESCRIPTION table). Self-products have only APC=0. Cross products have byte sizes B_{cross}while self products have byte sizes B_{self} stored in BITSIZE, respectively the second and rst element of this vector, in the MAIN table). Self products have at most two polarizations.

[3]Originally named SCAN_NUMBER in MS2

[4]Originally named SUB_SCAN in MS2

[5]Originaly named PHASE_ID in MS2

Altogether the size is:

$$\left(\frac{N_a(N_a-1)}{2}B_{cross}N_{apc}N_c + N_a B_{self}(N_c,2)\right)\sum_{i=1}^{i=N_{bb}} N_{bin}\left(\sum_{j=1}^{j=N_{sw}(i)} N_f(i,j)\right)$$

4.2.2.2 Data Cell for a sub-integration (channel averaged data) :

This is an array with the following six dimensions, from the fastest varying index to the slowest one:

1. **Polarization:** in order (XX, YY, XY, YX); dimension is N_c = 1, 2, or 4.
2. **Atmospheric pathlength correction:** (APC, dimension N_{apc} is 1 or 2). (as in the spectral data)
3. **Bin number:** N_{bin}(as in the spectral data)
4. **Window number.** Dimension N_{sw}(as in the spectral data)
5. **Baseband number:** Dimension N_{bb}(as in the spectral data)
6. **Baseline products:** Dimensions $N_a(N_a-1)/2$ and N_a.
   In order cross products $A_1.A_2, A_1.A_3, A_2.A_3, A_1.A_4, A_2.A_4, A_3.A_4, A_{n-1}.A_n$ followed by the self-products $A_1.A_1, ... A_n.A_n$. $A_1, ... A_n$ refer the the N_a antennas present in the collection (as defined in the CONFIG_DESCRIPTION table). Self-products have only APC=0. Cross products have byte sizes B_{cross}, while self products have byte sizes B_{self} Self products have at most two polarizations.

Altogether the size is:

$$\left(\frac{N_a(N_a-1)}{2}B_{cross}N_{apc}N_c + N_a B_{self}(N_c,2)\right)\sum_{i=1}^{i=N_{bb}} N_{bin}N_{sw}$$

4.2.3 Data normalization and scaling

- The cross-correlation spectra are in fractional cross correlation units as specified in SSR requirements: the cross-correlation spectral data for a baseline $A_i.A_j$ has been divided by the square root of the autocorrelation spectra of both antennas A_i and A_j.
- The auto-correlation spectra are normalized so that the average of spectral channel data is equal to unity. To obtain actual power one has to multiply by the square on the digitizer sampling level for the relevant antenna and baseband. (SAMPLING_LEVEL in the GAIN_TRACKING table).
- Each actual cross correlation spectral data element is stored as a couple of signed integers of size B_{cross}(=2 or 4)

bytes (real part then imaginary part) to be multiplied by the second element of SCALE_FACTOR (in MAIN)

- Each actual auto-correlation spectral data element is stored as one integer of size B_{self} bytes to be multiplied by the rst element of SCALE_FACTOR (in MAIN).
- Cross correlation and autocorrelation channel average data are stored as complex and real numbers respectively
- All binary representations are little endian, following Intel x86 format.

Table 1: Definition of Flags -- taken from SSR(should be generated automatically from data model)

Flag Word	Bit	Condition
FLAG_ANT	0	Last WVR calibration failed
FLAG_ANT	1	Current WVR hardware defect
FLAG_ANT	2	Last pointing calibration failed
FLAG_ANT	3	Last Temperature Scale Calibration failed
FLAG_ANT	4	Temperature Scale Calibration hardware defect
FLAG_ANT	5	Last WVR calibration failed
FLAG_ANT	6	Shadowing
FLAG_ANT	7	LO1 out of lock
FLAG_ANT	8	No correlation detected on last calibrator
FLAG_ANT	9	Too much decorrelation on last calibrator
FLAG_POL	0	Mixer not operational
FLAG_POL	1	Total power aout of range (pol)
FLAG_POL	2	LO2 out of lock
FLAG_POL	3	Integration totally blanked
FLAG_POL	4	Interference detected
FLAG_POL	7	No correlation detected on last calibrator
FLAG_POL	9	Too much decorrelation on last calibrator
FLAG_BASEBAND	0	Last Temperature Scale Calibration failed, Tsys currently estimated
FLAG_BASEBAND	1	Total power out of range
FLAG_BASEBAND	2	Interference detected
FLAG_BASEBAND	3	No correlation detected on last calibrator
FLAG_BASEBAND	9	Too much decorrelation on last calibrator
FLAG_BASEBAND	4	Integration totally blanked

4.2.4 Flags

Flags are present in the MAIN table in several columns. All flags relevant to raw are antenna based.

FLAG_ANT int (4-byte) vector of size N_a. Each bit correspond to a specified flagging condition (e.g. 2^2 = pointing, 2^6 = shadowing, ...). When different from zero all data concerning the antenna is affected.

FLAG_POL int (4-byte) vector of size $N_a N_c$. When different from zero all data concerning the polarization and antenna is invalid (all basebands) (e.g. problem incident before the down converters)

FLAG_BASEBAND int (4-byte) vector of size $N_a N_c N_{bb}$. When different from zero all data concerning the baseband and antenna is invalid (e.g. problem incident after the down converters)

See table 4.2.4 (TBD) for list of flag conditions. **Note: One could duplicate the structure to have e.g. two severity levels (warning/fatal)**

4.2.5 Weights

Weights are stored implicitly as:

- A system temperature spectrum T_{SYS} for each antenna in table SYSCAL
- Channel width for each spectral datum (EFFECTIVE_BW in table SPECTRAL_WINDOW)
- Actual integration time for each integration/sub-integration (EXPOSURE in table MAIN)
- Actual correlator sensitivity as deduced from correlator mode (w =WEIGHT_FACTOR in SPECTRAL_WINDOW)(to **be added, or deleted if already included in EFFECTIVE_BW**)

The weights may then be computed according to the radiometric formula:

$$W_{ij}(\nu) = w\sqrt{\frac{\delta_{\nu\tau}}{T_{SYS_i}(\nu)T_{SYS_j}(\nu)}}$$

4.2.6 Water Vapour Radiometry data

The original radiometric raw data will not be synchronized with the data from the correlator. For that reason they have to be put in a specific table: TOTAL_POWER_MONITORING . This table has a similar structure as the MAIN table. The spectral windows associated to the radiometric data will hold as associates (ASSOC_SPW_ID) the spectral windows used for the astronomical data which are obtained in parallel.

These radiometric data are resampled by the on-line system for synchronization at the same intervals (about 1 second) of the correlator data taking. The Metadata and Auxiliary Data tables refer to these data synchronized at the sub-integration level. A filler option should allow to get the data synchronized with the astronomical data, when downloading into the o-line data reduction and analysis software package. If exported to the MSv2.0 MAIN table, these data would be naturally identified via the PROCESSOR_ID coordinate. The alternative is to keep them in a separate table but with a one to one correspondance with the rows in the MAIN to preserve efficient access to the association when both, the astronomical and the radiometric data, are used in conjunction. The radiometers on the different antennas, together with their spectral analyzing devices, may differ slightly from each other. If these differences reduce to global spectral shifts, the FREQOFFSET table can accomodate the need but if the differences are less simple to describe, there would be the need to define one spectral window for each radiometer (i.e. one per antenna). In that case the -1 option for SPECTRAL_WINDOW_ID in the FEED table cannot be used. For this reason we propose to add a rule when setting SPECTRAL_WINDOW_ID for the FEED table (see that item below the FEED table).

For the cross-correlations the ATMPHASE_CODE in the DATA_DESCRIPTION table indicates if the data-cell contains or not the uncorrected correlator data together with the data corrected by the on-line system. This presence is defined baseband per baseband.

4.2.7 Temporal and frequency multi-resolutions

The ALMA data set will need to host not only the full frequency resolution data for each integration but also associated data with lower spectral resolutions. Typically we anticipate for each spectrum per baseband at full resolution, a 1 0 0 MHz resolution spectrum and/or a single channel, resulting from the averaging of a selected list of channels, both with sub-integration sampling (typically 1 second).

To hold the different temporal granularities we use the STATE table, SUBINTEG_NUMcarrying the subdivision of the integration for the high temporal resolution.

To hold the different frequency resolutions we use the optional (ASSOC_SPW_ID) column in the SPECTRAL_WINDOW table. In the SYSCAL table the *key* SPECTRAL_WINDOW_ID has to refer to the associated spectral window corresponding to the lower resolution.

4.2.8 Sidebands

ALMA will produce data in DSB, SSB (LSB or USB) and 2SB (in case of sideband separation). With the MeasurementSet model, when there are two sidebands there is one SPECTRAL_WINDOW_ID per sideband. The lower and upper sidebands can be linked together via the optional ASSOC_SPW_ID to provide a list of identifiers. For SSB data and auto-correlated DSB data associated spectral windows can be used to carry the information for the calibration in both sidebands separately, while keeping track of their close association. The NET_SIDEBAND item in table SPECTRAL WINDOW is reserved for the following: -1 (no sideband) , 0(DSB) , 1(LSB SSB) , 2(USB SSB) , 3(LSB 2SB (sideband separation by phase switching)) , 4(USB 2SB (sideband separation by phase switching)) , 5(LSB 2SB) and 6(USB 2SB) .

4.2.9 Subarray and array configuration:

Sub-arrays: the information is contained in the ANTENNAARRAY collection which sets the array of ANTENNA_ID identifiers.
There is no more than one sub-array in a data cell.
Sub-arrays with coordinated activities appear in the MAIN table as different rows with the same values for the TIME and INTERVAL *key*s. These sub-arrays may be characterized by having different pointing directions (eg if they have different FIELD_ID identifiers in the MAIN) or different set of spectral windows if the DATA_DESCRIPTIONARRAY collections are different in the CONFIG_DESCRIPTION table.
Indeed the MAIN table can be used with sub-arrays with un-coordinated activities.
There is a ANTENNA_ID_SET collection in the EXECUTE_SUMMARY table which gives the antenna configuration during the execution of the Scheduling Block.

Multi-configuration: There is multi-configuration when at least two of the Ececuting Blocks indexed in the MAIN involve antennas which have been moved during the lifetime of the project. It should be noticed that with ALMA, because of the combined effects of dynamic scheduling and continuous antenna reconfiguration, multiconfiguration data set will not be exceptional.

4.2.10 Source parameters:

In contrast to lower frequencies, in the millimeter and sub-millimeter the sources observed for flux calibrations are variable. In that context we propose to add a new table, SOURCE_PARAMETER , to trace properly the various steps required for this calibration. This table has a recurrence to allow to store bootstrap solutions with the interdependency between different sources for the determination of their fluxes.

4.2.11 Summary:

Fig. 1 illustrates the diagram of the AEDF model. Its ensemble of tables is listed in Tab. 2A. Relative to the MSv2.0 tables a few have been added, some renamed, e.g. the EXECUTE_SUMMARY [6] table, and a few have been associated differently, e.g. the RECEIVER table. Table 3 gathers the list of items defining sizes in the AEDF together with their visibility in the data model and restrictions relevant for ALMA.

5 Data format implementation

External users will be able to receive exported data either as FITS les or VOTables.

5.1 FITS Implementation

FITS is a format widely used in astronomy, with many years of successful experience, and is ocially recommended by the IAU. The data model we have described here, a set of tables, can be handled as FITS binary table extensions. The measurement set can be currently exported in FITS format in AIPS++. We have lacked time to check in detail how it is done. It is important that the FITS les produced conform to the current FITS standards, and in particular do implement the WCS scheme (World Coordinate System) to describe the coordinate axes.

[6] Table originaly named OBSERVATION in MS2

5.2 VOTable Implementation

The VOTable format is the proposed standard for the Virtual Observatory projects. ALMA has proposed to use VOTables, both as an internal archive format, and to transfer large amounts of data between real-time subsystems.

There is an experimental implementation of the AEDF as a set of VOTables, one for each of the AEDF tables. In addition a Master VOTable contains links to each of the individual tables. See figures 5.3 and 5.3 for samples. It must be noted that this implementation does not have all the semantics present in the data model of the AEDF. In particular it can be seen that the AEDF identifiers are not recognized as such in the examples presented here (see e.g. the VOTable <FIELD> for the DOPPLER_ID identifier in 5.3). The VOTable (vers. 1.0) format does not yet allow to specify that a column in one table is a *key* in another table. It is clear that this format needs to be upgraded to support this feature which is widely used in the AEDF. A more general difficulty is that at present there is no way to retrieve with a standard XPath expression the position of a row in a XML table. This problem could be surpassed by adding a column in the AEDF tables which have only implicit identifiers such as the ANTENNA table with being indexed in other tables. The impact of this need has to be investigated for performance reasons.

The binary data is referred to from the MAIN table rows following a VOTable format extension proposed by Andreas Wicenec.

5.3 VOTable-FITS compatibility

It remains to be proven that an implementation of the AEDF into VOTables with the required extensions to completely describe its AEDF data model, it is possible to transform them into FITS le without loosing some information (ref. section 2.2 of the VOTable specification document).

6 A Proposed MS model table structure for ALMA use

Data, coordinates and flags.

Figure 2: A sample SPECTRAL_WINDOW table

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<VOTABLE version="1">
  <RESOURCE name="PARAMS">
    <PARAM datatype="char" name="AedfType" value="SpectralWindow"> <DESCRIPTION />
    </PARAM>
    <PARAM datatype="char" name="ProjectUID"
          value="uid://X0000000000000066/X00000000">
      <DESCRIPTION>This parameter contains the identifier of the project this table belongs
                to.</DESCRIPTION>
    </PARAM>
  </RESOURCE>
  <RESOURCE name="TABLE">
    <TABLE>
      <FIELD datatype="int" name="numChan">
        <DESCRIPTION># spectral channels</DESCRIPTION>
      </FIELD>
      <FIELD arraysize="*" datatype="char" name="name"> <DESCRIPTION>Spectral window
        name</DESCRIPTION>
      </FIELD>
      <FIELD datatype="double" name="refFreq">
        <DESCRIPTION>The reference frequency</DESCRIPTION>
      </FIELD>
      <FIELD arraysize="*" datatype="double" name="chanFreq">
        <DESCRIPTION>Center frequencies for each channel in the data matrix.
                Array of size numChan.</DESCRIPTION>
      </FIELD>
      <FIELD arraysize="*" datatype="double" name="chanWidth">
        <DESCRIPTION>Channel width for each channel in the data matrix.
                Array of size numChan.</DESCRIPTION>
      </FIELD>
      <FIELD datatype="int" name="measFreqRef">
        <DESCRIPTION>FREQUENCY Measure ref.</DESCRIPTION>
      </FIELD>
      <FIELD arraysize="*" datatype="double" name="effectiveBW"> <DESCRIPTION>The effective noise
        bandwidth for each channel in
                the data matrix</DESCRIPTION>
      </FIELD>
      <FIELD arraysize="*" datatype="double" name="resolution"> <DESCRIPTION>The effective spectral
        resolution for each channel
                in the data matrix</DESCRIPTION>
      </FIELD>
      <FIELD datatype="double" name="totBandwidth">
        <DESCRIPTION>Total bandwidth for this window</DESCRIPTION>
      </FIELD>
      <FIELD datatype="int" name="netSideband">
        <DESCRIPTION>Net sideband</DESCRIPTION>
      </FIELD>
      <FIELD datatype="int" name="bbcNo">
        <DESCRIPTION>Baseband converter no.</DESCRIPTION> </FIELD>
      <FIELD datatype="int" name="bbcSideband">
        <DESCRIPTION>BBC sideband</DESCRIPTION>
      </FIELD>
```

```
        <FIELD datatype="int" name="ifConvChain">
          <DESCRIPTION>The IF conversion chain.</DESCRIPTION> </FIELD>
        <FIELD datatype="int" name="freqGroup">
          <DESCRIPTION>Frequency group</DESCRIPTION>
        </FIELD>
        <FIELD arraysize="*" datatype="char" name="freqGroupName"> <DESCRIPTION>Frequency group
          name</DESCRIPTION>
        </FIELD>
        <FIELD datatype="int" name="dopplerId"> <DESCRIPTION>Doppler id.</DESCRIPTION>
        </FIELD>
        <FIELD arraysize="*" datatype="int" name="assocSpwId">
          <DESCRIPTION>Associated spectral window ids</DESCRIPTION> </FIELD>
        <FIELD arraysize="*" datatype="char" name="assocNature">
          <DESCRIPTION>Nature of association</DESCRIPTION>
        </FIELD>
        <FIELD datatype="int" name="bitSize">
          <DESCRIPTION>Bit size</DESCRIPTION>
        </FIELD>
        <FIELD datatype="boolean" name="flagRow">
          <DESCRIPTION>The row flag.</DESCRIPTION>
        </FIELD> <DATA>
          <TABLEDATA>
             <TR> <TD>1</TD>
                <TD>SpectralWindow#0</TD>
                 ...
                <TD>1 2 3</TD> <TD>CHANNEL_ZERO|EQUAL_FREQUENCY|SUBSET</TD> <TD>32</TD>
                <TD>false</TD>
              </TR> ...
          </TABLEDATA>
        </DATA> </TABLE>
  </RESOURCE> </VOTABLE>

  <?xml version="1.0" encoding="UTF-8"?> <VOTABLE>
  <RESOURCE name="PARAMS">
     <PARAM name="Name" value="my project" />
     <PARAM name="AedfType" value="Project" /> <PARAM
     name="DataDescriptionTableUID"
                    value="uid://X000000000000006b/X00000000" /> <PARAM name="PolarizationTableUID"
                                            value="uid://X000000000000006c/X00000000" />
                                               <PARAM name="SpectralWindowTableUID"
                                            value="uid://X000000000000006d/X00000000" />
  </RESOURCE>
</VOTABLE>
```


Figure 3: A sample MASTER table in the case of a data set corresponding to an ObsProject, i.e. eventually several ExecutionBlocks but all common to a single project.

MAIN Data, coordinates and ags.				
Name	Format	Units	Measure	Comments
Keywords				
AEDFVERSION	Float	s		AEDF format version
Columns				

Key				
TIME *(TIME_EXTRA_PREC)* CONFIG_DESCRIPTION_ID FIELD_ID	Double Double Int Int	s s	EPOCH	Integration mid-point extra TIME precision Cong-description identifier Field identifier
Data description				
SCALE_FACTOR BITSIZE	Float(N_{cm}, N_{bb}) Int(N_{cm}, N_{bb})			Scale factor(s) for the cross and/or auto-correlations Bit size(s) for the cross and/or auto-correlations
Non-key attributes				
INTERVAL EXECUTE_ID SCAN_NUMBER (OBS_NUMBER) INTEG_NUMBER STATE_ID UVW EXPOSURE TIME_CENTROID	Double Int Int Int Int Int(N_a) Double(3,N_a) Double(N_{bb}) Double(N_{bb})	s m s s	 EPOCH	Time interval Execute identifier Scan number Observation number Integration number State identifier UVW coordinates Effective integration time Time centroid
Data				
DATA_OID	String			Data object identifier
Flag information				
FLAG_ANT FLAG_POL FLAG_BASEBAND FLAG_ROW	Int(N_a) Int(N_a, N_c) Int(N_a, N_c, N_{bb}) Bool			Flags concerning antennas Flags concerning antennas and polarization Flags concerning antennas, polarization and basebands The row flag

Notes: N_{bb} is the number of basebands (attribute of the CONFIG_DESCRIPTION table), The form of the data object is determined by the attributes of the *Data description* section in the CONFIG_DESCRIPTION table and by N_c, the number of correlation products, N_f, the number of frequency channels and N_{apc} which is 1 or 2. These three sizes N_c, N_f and N_{apc} are known via the identifiers in the DATA_DESCRIPTION_ARRAY collection.

AEDF_VERSION EDF revision number, expressed as *major_revision, minor_revision*.

TIME Mid-point (not centroid) of data interval.

TIME_EXTRA_PREC Extra time precision

CONFIG_DESCRIPTION_ID Configuration descriptor identifier >(0) providing a direct index into the CONFIG_DESCRIPTION sub-table row number. Note that two or more sub-arrays cannot refer to the same CONFIG_DESCRIPTION_ID identifier.

FIELD_ID Field identifier (≥0)

SCALE_FACTOR Scaling factors for the auto-correlated data (case $N_{cm=}$ 1) or for the cross-correlated and auto-correlated data (case $N_{cm=}$ 2) , this for each of the N_{bb} basebands, which are stored in the data cell.

BITSIZE The numbers of bytes (2 or 4) for the auto-correlations (case $N_{cm=}$ 1) or for the autocorrelated and cross-correlated data, this for each of the N_{bb} basebands, which are stored in the data cell indexed by the DATA_OIDobject identifier.

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

EXECUTE_ID This provides access to a row number in the EXECUTE_SUMMARY table. This meta coordinate defines the data-base in term of an implicit collection of data blocks.

SCAN_NUMBER SCAN (ref. ALMA glossary) number. Enumeration relative to the Executing Block.

OBS_NUMBER OBSERVATION (ref. ALMA glossary) number.

INTEG_NUMBER INTEGRATION (ref. ALMA glossary) number. The enumeration is relative to the OBSERVATION (ref. ALMA glossary).

STATE_ID State identifier (≥ 0).

UVW Antenna-based uvw coordinates for the N_a antennas. This is defined at the centroid time, TIME_CENTRIOD.

EXPOSURE Effective duration of an INTEGRATION (or a SUBINTEGNUM in the STATE table is > 0).

TIME_CENTROID Time stamp reflecting the average time the non-blanked data was integrated.

DATA_OID Reference to the data cell object. It is an object identifier and, as such, is of type String.

FLAG_ANT Unsigned long int (4-bytes) vector. Each bit corresponds to a specified flagging condition (e.g. 2^2 = pointing, 2^6 = shadowing, ...). When different from zero all data concerning the antenna is affected.

FLAG_POL Unsigned long int (4-bytes) vector . When different from zero all data concerning the polarization and antenna is invalid (all basebands) (e.g. problem incident before the down converters)

FLAG_BASEBAND Unsigned long int (4-bytes) vector. When different from zero al l data concerning the baseband and antenna is invalid (e.g. problem incident after the down converters)

FLAG_ROW True if the entire row is flagged. Remarks:

1. **Sub-arrays:** For a given TIME and INTERVAL the presence of different CONFIG_DESCRIPTION_ID attributes reflects different sub-arrays. With CONFIG_DESCRIPTION_ID as a *key* this table can have rows for different arrays, even if their are not synchronized in time The EXECUTE_SUMMARY table contains an ANTENNA_ID_SET attribute which must at least include all the ANTENNA_ID referenced in the ANTENNA_ARRAY collections used in the Execution Block.

2. **Multi-configurations:** In the case of data sets which includes several array configurations, note that there is a CONFIG_NAME in the EXECUTE_SUMMARY table; this allows to distinguish easily the case of several sub-arrays and the case of several array configurations.

Processor mode information.

ALMACORRELATORMODE Processor mode information.				
Name	Format	Units	Measure	Comments
Columns				
Data				

BASEBAND_INDEX	Int			Baseband index
ACCUM_MODE	Int	ms		Accumulation mode
BASEBAND_CONFIG	Int			Baseband configuration mode
BIN_MODE	Int			Bin mode
QUANTIZATION	Bool			Quantization correction
WINDOW_FUNCTION	String			Window function

Notes: This sub-table contains information on the processor mode applicable to the current data record.

BASEBAND_INDEX The baseband index (1-4) configured.
ACCUM_MODE Processor accumulation mode. For the ALMA correlator it can be 1 or 16 ms which defines the correlator chip level accumulation duration.

BASEBAND_CONFIG Baseband configuration mode (1-66) which defines the bandwidth, polarizations and over-sampling.

BIN_MODE Single Bin = 0. Multiple bins, subarray-based (e.g., nutating sub-reflector or frequency switch) = 1. Multiple bins, baseline based

QUANTIZATION Boolean selection to perform quantization correction on the lag data

WINDOW_FUNCTION An enumerated value representing the windowing function to use: Uniform, Hanning, Hamming, Bartlett, Blackman, Blackman-Harris, Welch.

Antenna characteristics.

ANTENNA Antenna characteristics.				
Name	Format	Units	Measure	Comments
Columns				
Data				

NAME	String			Antenna name
STATION	String			Station name
TYPE	String			Antenna type
POSITION	Double(3)	m	POSITION	Antenna X,Y,Z phase reference positions
TIME	Double	s	EPOCH	Time when the antenna POSITION has been calibrated
OFFSET	Double(3)	m	POSITION	Axes offset of mount to FEED REFERENCE point
DISH_DIAMETER	Double	m		Diameter of dish
ASSOC_ANTENNA_ID	Int			Associated antenna identifier
Flag information				
FLAG_ROW	Bool			The row flag

Notes: This sub-table contains the global antenna properties for each antenna in the AEDF. It is indexed directly from the ANTENNA ARRAY collection in the CONFIG DESCRIPTION table.

NAME Antenna name

STATION Station name

TYPE Antenna type. The reserved *keywords* include GROUND-BASED[7] for conventional antennas would be used in the case of ALMA, SPACE-BASED[8] , TRACKING-STN[9] .

POSITION In 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 *key*word (IRTF or WGSS4). The reference point is the point on the az or ha axis closest to the el or dec axis.

TIME Time when the POSITION of this antenna has been determined. Measured reference as used in the MAIN .

OFFSET Axes offset of mount to feed reference point.

DISH_DIAMETER Nominal diameter of dish, as opposed to the effective diameter.

ASSOC_ANTENNA_ID Associated antenna identifier. A value of -1 means no associated identifier When an antenna position has been updated post-observation while it remained on the same station, the antenna tuple corresponding to the ANTENNA_ID used during these observations must be updated to link to an associated new ANTENNA_ID as soon as this new tuple with the updated position has been added in this sub-table. The ASSOC_ANTENNA_ID is the identifier of that new appended tuple.

FLAG_ROW Boolean to indicate the validity of this entity. Set to True for an invalid row. This does not imply any flagging of the data in the MAIN or TOTAL POWERMONITORING , but is necessary

[7] Attribute GROUND-BASED reserved for *key*word TYPE in MS2
[8] Attribute SPACE-BASED reserved for *key*word TYPE in MS2
[9] Attribute TRACKING-STN reserved for *key*word TYPE in MS2

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.

Calibration device characteristics.

CALDEVICE Calibration device characteristics.				
Name	Format	Units	Measure	Comments
Columns				
Key				
ANTENNA_ID FEED_ID SPECTRAL_WINDOW _ID TIME INTERVAL	Int Int Int Double Double	 s s	 EPOCH	Antenna identifier Feed identifier Spectral window identifier Time interval mid-point Time interval
Data description				
NUM_CALLOAD	Int			Number N_{cl} of NCAL or TLOAD
Data				
(NCAL) *(TLOAD)* *(CALEFF)*	Double(N_{cl}) Double(N_{cl}) Float(N_r, N_{cl})			Noise calibration Load temperature Calibration device efficiency/coupling coefficient

Notes: This table contains the characteristics of devices used for the calibration. NUM_RECEPTORS is known through the FEED_ID . The load observed at a given time is specified in the STATE table via the item CALLOAD_NUM when REF is TRUE. Note that a single row MAIN table is associated to a given state as indexed from STATE_ID , this state being identical for all the involved antennas i.e. the members of the ANTENNAARRAY collection. This state has parameters which may be different from one antenna, feed, spectral window combination to an other as described here concerning the calibration devices.
Note that N_r, the number of receivers, is specified in the FEED table.
N.B.: Additional calibration device properties will have to be defined if necessary.

ANTENNA_ID Antenna identifier, as referenced in the ANTENNA_ARRAY collection in the CONFIG_DESCRIPTION table.

FEED_ID Feed identifier, as referenced in the FEED_LIST collection in the CONFIG_DESCRIPTION table.

SPECTRAL_WINDOW_ID Spectral window identifier. A value of -1 indicates the row is valid for all spectra windows. If a specific spectral window has its own feed characteristics these have to be define separately using a value 0, to override the global specifications for that specific window.

TIME Mid-point of time interval for which the parameters in the row are valid. The same reference used for the TIME column in the MAIN must be used.

INTERVAL Time interval. It should be the same as in the FEED table.

NUMCAL_LOAD Number of calibration noise or load N_{load} available for this feed.

NCAL Noise calibration temperatures (0 if not added) Either NCAL or TLOAD must be provided but not both simultaneously.

TLOAD Physical temperatures of the load for each calibration device. Either NCAL or TLOAD must be provided but not both simultaneously.

CAL_EFF Calibration device efficiency/coupling efficiencies. These efficiencies may not be strictly identical for the different receivers.

Configuration description

CONFIG_DESCRIPTION Configuration description.				
Name	Format	Units	Measure	Comments
Columns				
Key				
PROCESSOR_ID	Int			Processor identifier
Data description				
NUM_ANTENNA	Int			Number of antennas N_a in the collection
NUM_CORRBIN	Int(N_a)			Number of correlator data bins N_{bin} for each base-band
NUM_BASEBAND	Int			Number of basebands N_{bb} in the collection
NUM_SUBBAND	Int(N_{bb})			Number of SUBBANDs N_{sw} for each baseband
Data				
ANTENNA_ARRAY	Int(N_a)			Antenna collection
FEED_LIST	Int(N_a)			Feed collection
PHASED_ARRAY_LIST	Int(N_a)			Phased array collection
DATA_DESCRIPTION_ARRAY	Int($\sum_{i=1}^{i=N_{bb}} \left(N_{bin} x N_{sw}\right)$))			Data description collection
SWITCH_CYCLE_ ARRAY	Int(N_{bb},N_a)			Switch cycle collection
CORRELATION_MODE	Int			Correlation mode N_{cm}
Flags				
FLAG_ANTENNA	Bool(N_a)			Antenna flag collection

Notes: This table sets a referential constraint in the AEDF. It defines the shape of the associated data cell in the MAIN table where it is indexed directly by CONFIG_DESCRIPTION_ID identifier. The number of identifiers in the DATA_DESCRIPTION_ARRAY collection is $\sum_{i=1}^{i=N_{bb}} \left(N_{bin} x N_{sw}\right)$ The collections in this table which are of type array contain ordered sequences of elements to describe how the data are organized in the data cell (note that any identifier must appear only once in a collection of type array).

PROCESSOR_ID Processor identifier (≥ 0) providing a direct index into the PROCESSOR sub-table row number. This identifier would allow to distinguish data which come from e.g. ACA or dedicated single-dish antennas in case these antennas are connected to a correlator of different design.

NUM_ANTENNA Number of antennas in a collection. This collection may correspond to an entire array or to a sub-array.

NUM_CORRBIN Number of correlator data bins N_{bin} for each baseband

NUM_BASEBAND Number of basebands N_{bb} in the collection

NUM_SUBBAND Number of subbands N_{sw} for each baseband. Note that this number may not correspond to the number of correlator subbands in the baseband as several may be stitched together. NUMSUBBAND must be used to represent the number N_{sw} of spectral windows after the stitching operations.

ANTENNA_ARRAY ANTENNA collection. It contains an ordered sequence of ANTENNA_ID identifiers to define the antenna (sub-)array. This information is required to know the form of the datacell object. Example: consider a sub-array of 4 antennas, $N_a = 4$, with their respective identifier ANTENNA_ID 3 7 70 and 30. In this case the data-cell contains successively the data for the pairs 3-7, 3-70, 7-70, 3-30, 7-30, 70-30, 3-3, 7-7, 70-70 and 30-30, this order[10] being determined by the position of the identifiers in the sequence. As illustrated with this example it is seen that the data-cell contains firstly the cross-correlated data and then the auto-correlated data. The number of atomic data per pair corresponds to $N_{APC} N_c \sum_{i=1}^{i=N_{bb}} N_{bin}(i) \sum_{j=1}^{N_{sw}(i)} N_f(i,j)$ these data being stored 2 or 4 bytes integers (see item BITSIZE in the MAIN table) which can be converted into oat-complex or oat using the item SCALE_FACTOR (also in the MAIN table).
Note: see item PHASED_ARRAY_LIST in case of phased arrays.

FEED_LIST List of FEED_ID identifiers. This list is a collection with the same number of elements as in the ANTENNAARRAY collection. Each element corresponds to a FEED_ID identifier. In almost all cases, for ALMA, FEED_ID = 0 because there is in general only one feed for the pair of *key*s ANTENNA_ID ,SPECTRAL_WINDOW_ID . In a few exceptional cases one could have however two tuples with FEED_ID = 0 and 1 respectively for a given SPECTRAL_WINDOW_ID , this occurring when one would observe in a spectral region where two adjacent receiver bands overlap, e.g. with the bands 2 and 3 of ALMA.

PHASED_ARRAY_LIST List of phased array numbers. This list is a collection with the same number of elements as in the ANTENNAARRAY collection. Each element corresponds to a phased (sub)array number. A value of -1 means that the corresponding antenna is not co-phased with other antennas. Taking the example given for the item ANTENNA_ARRAY , if the list PHASED_ARRAY_LIST is (-1 -1 -1 -1) none of the antennas with their identifiers 3, 7, 70 and 30 are co-phased. This would also be the case if each element is unique in the list as e.g. with a list (0 1 2 3). If several elements have the same value then the corresponding antennas are co-phased and assigned to the subarray number corresponding to that value. For example with (0 0 0 1) the antennas with the identifiers 3, 7 and 70 are co-phased and the co-phased result is cross-correlated with the antenna 30, provided that CORRELATION_MODE is 0 or 2. The order of the elements in the ANTENNA_ARRAY collection must be such that the co-phased antennas appear in adjacent positions in this PHASED_ARRAY_LIST list. The positions of the cross-correlated data in the data cell are ordered according to the order in the PHASED_ARRAY_LIST list. With (0 0 0 1), or for a list as (0 0 0 1 1), there is only one baseline, 0-1. With a list like (0 0 0 1 1 2) the cross-correlated data will appear successively for 0-1, 0-2 and 1-2 in the data cell.

DATA_DESCRIPTION_ARRAY DATA_DESCRIPTION collection of ordered DATA_DESC_ID identifiers which defines the ensemble of DATA_DESC_ID used by the correlator for the antenna (sub-)array. The order of the elements in this collection gives the sequence of data_descriptions to define the organization of the data in the data cell. Together with the ANTENNA_ARRAY collection it contributes to determine the form of the data-cell object. This DATA_DESCRIPTION_ARRAY collection defines a domain of integrity in a tri-dimensional space, the axes being the correlator bin axis, the sub-band axis and the baseband axis. The bin axis size may be greater than one, e.g. in case of observations using the frequency-switch mode. The order of these axes is fixed, the coordinates along the bin axis being scanned the first.

SWITCH_CYCLE_ARRAY Array of SWITCH_CYCLE_ID identifiers, one per baseband. For a given identifier this provides N_{bin} which sets the size of the bin axis, for a certain baseband, in the data-cell object.N_{bin} has to be given for each baseband.

CORRELATION_MODE Correlation mode. 0 means no auto-correlation i.e. only cross-correlations, 1 means no cross-correlation i.e. only auto-correlations and 2 means both cross-correlations and auto-correlations. For ALMA valid values are restricted to 1 and 2 which can be represented by N_{cm}.

FLAG_ANTENNA Flag antenna collection with the same number of elements as for ANTENNAARRAY collection. One element in this collection is TRUE if the data cell in the MAIN table has valid data for the corresponding antenna; does not imply flagging in MAIN .

Spectro-polarization description.

DATA_DESCRIPTION Spectro-polarization description.				
Name	Format	Units	Measure	Comments
Columns				
Data				
SPECTRAL_WINDOW_ID POLARIZATION_ID ATMPHASECODE	Int Int Int			Spectral window identifier Polarization identifier Code for atm phase correction
Flags				
FLAG_ROW	Bool			The row flag

Notes: This table sets a referential constraint in the AEDF. It is indexed directly by DATA_DESC_ID in the collection DATA_DESCRIPTION_ARRAY in the CONFIG_DESCRIPTION table.

SPECTRAL_WINDOW_ID Spectral window identifier (0): direct index into the DATA_DESCRIPTION sub-table.

POLARIZATION_ID Polarization identifier (0): direct index into the POLARIZATION sub-table.

ATMPHASE_CODE Atmospheric phase correction code: 0 for un-corrected, 1 for corrected, 2 for both un-corrected and corrected. This gives the size of the “APC” axis in the data cell; it is given by N_{apc}=1 for ATMPHASE_CODE<2 else $N_{apc=2.}$ DATA_DESC_ID is indexed in the DATA_DESCRIPTION_ARRAY collection of the CONFIG_DESCRIPTION table.

FLAG_ROW True if the row does not contain valid data; does not imply flagging in the MAIN table.

[10]Note that this baseline order is not the same as in the correlator ICD document (vers. Jan 2004)

Doppler tracking information.

DOPPLER Doppler tracking information.				
Name	Format	Units	Measure	Comments
Columns				
Key				
DOPPLER_ID SOURCE_ID	Int Int			Doppler identifier Source identifier
Data				
TRANSITION_ID VELDEF	Int Double	$m\ s^{-1}$	DOPPLER	Transition identifier Velocity definition of Doppler shift

Notes: This sub-table contains frame information for different Doppler tracking modes. It is indexed from the SPECTRAL_WINDOW_ID sub-table (with SOURCE_ID as a secondary index) and thus allows the specification of a source-dependent Doppler tracking reference for each spectral window. This mode allows multiple possible transitions per source per spectral window, but only one reference at any given time.

DOPPLER_ID Doppler identifier, as used in the SPECTRAL_WINDOW sub-table.

SOURCE_ID Source identifier (as used in the SOURCE sub-table).

TRANSITION_ID This index selects the appropriate line from the list of transitions stored for each SOURCE_ID in the SOURCE table. A reference catalog for the line transitions need TBD.

VELDEF Velocity definition of the Doppler shift, e.g., RADIO or OPTICAL velocity in $m\ s^{-1}$.

Schedule and Execute summary.

EXECUTE_SUMMARY Schedule and Execute summary.				
Name	Format	Units	Measure	Comments
Columns				
Data				
TELESCOPE_NAME	String			Telescope name
CONFIG_NAME	String			Name of the antenna configuration
NUM_ANTENNA	Int			Number of antennas N_a in the collection
ANTENNA_ID_SET	Int(N_a)			List of antenna identifiers
(BASE_RANGE)	Float(2)	m		Minimum and maximum baseline lengths
(BASE_RMS)	Float(2)	m		rms along the major and minor axis of the baselines distribution
(BASE_PA)	Float	rad		Position angle of the major axis in the baseline distribution
TIME	Double	s	EPOCH	Time interval mid-point
INTERVAL	Double	s		Time interval
OBSERVER_NAME	String			Name of observers
OBSERVING_LOG	String(*)			Observing log
SCHEDULE_ID	Int			Schedule identifier
EXECUTEE_INTENT	String(*)			Execute intent
EXECUTE_SCRIPT	String			Project execution script
PROJECT	String			Project identification string
RELEASE_DATE	Double	s	EPOCH	Target release date
Flags				
FLAG_ROW	Bool			The row flag

Notes: This table gives the set of Executing Blocks associated to each Scheduling Block; at the same time it gives the list of Scheduling Block actually used. It is indexed via the identifier EXECUTE_ID directly from the HISTORY and as a secondary index, from the TOTAL_POWER_MONITORING and MAIN tables.

TELESCOPE_NAME Name of the telescope, ALMA.

CONFIG_NAME String type to support names with current instruments (TBD).

NUM_ANTENNA Number of antennas N_a (indicative) for the antenna configuration.

N.B.: It must be understood that this number does not necessarily correspond to the number of rows in the ANTENNA table; it may be smaller but never larger. This number does not correspond neither to N_a in the CONFIG_DESCRIPTION table.

ANTENNA_ID_SET Collection: set of antenna identifiers ANTENNA_ID giving the ensemble of antennas which belong to the current array configuration. These identifiers do not need to be given in a specified order but obviously the members in this collection must be unique.
Note that if there are subarrays, the subsets of antennas forming these subarrays are specified in the CONFIG_DESCRIPTION table. The present collection does not imply that all its members are cross-correlated.

BASE_RANGE Baseline range (minimum, maximum).

N.B.: This range gives the extent over which the phase structure function is sampled.

BASE_RMS rms along the minor and major axis in the baseline distribution. This provides one characteristic for the current array configuration. This could also characterize how is sampled the atmospheric phase structure function.

BASE_PA Position angle in the distribution of baselines.

TIME Time mid-point interval. The same reference used for the TIME column in the MAIN must be used.

INTERVAL Time interval. This is the difference between the end and begin TIME of the SB execution

OBSERVER_NAME Name of the observer.

OBSERVING_LOG Name of the observing log.

OBSERVING_MODE Observing mode, e.g. singleField, mosaic
The definition of the observing modes at the level of a SchedulingBlock are TBD.

SCHEDULE_ID Scheduling Block identifier. Not required to be unique.

EXECUTEEINTENT Intent given for the execution of the SB. When the execution of a given SB is repeated the intent would be e.g. increaseUvCoverage, monitorVariability, increaseStoN etc... Not required to be unique.

EXECUTE_SCRIPT The same le name, if the same script is used, may be given here if what has been scheduled is exactly what has been executed. Not required to be unique. Each Executing Block has to be notified with one row in this table.
It is indexed directly from the MAIN table for a straightforward segmentation of whole data base into data blocks.
EXECUTE_ID is used for data selection to define an Observation Unit (see the OBSERVATION_UNIT sub-table).

PROJECT Project reference (TBD)

RELEASE_DATE Date the Executing Block becomes public.

FLAG_ROW True if the row does not contain valid data; does not imply flagging in the MAIN table.

Remark:
CONFIG_NAME provides the state of the antenna configuration during the execution of a Scheduling Block, even if not all antennas are used for that execution. The actual description of a sub-array is given by an ANTENNA_ARRAY collection in the CONFIG_DESCRIPTION table. CONFIG_NAME allows to discriminate observations obtained while the array was in one or several array configurations. If the plan is that ALMA will be in continuous reconfiguration, then it will be quite common to have multi-configuration projects. CONFIG_NAME takes clearly its importance in this context. The set of antenna identifiers in the collection ANTENNA_ID_SET is given here to fully describe the antennas defining the antenna configuration.
Note that this table must contains as many rows as there were executions of Scheduling Blocks in the

project, i.e. $\sum_{i=1}^{i=N_{SB}} N_{XB}^{i}$.

SCHEDULE_ID and EXECUTE_SCRIPT are not required to be unique, as it would be the case when a SB and an EXECUTE_SCRIPT are repeated.
Sub-arrays operating asynchronously must appear with different EXECUTE_SCRIPT.

Feed characteristics.

FEED Feed charactersitics.				
Name	Format	Units	Measure	Comments
Columns				
Key				
ANTENNA_ID FEED_ID SPECTRAL_WINDOW_ID TIME INTERVAL	Int Int Int Double Double	 s s	 EPOCH	Antenna identifier Feed identifier Spectral window identifier Interval mid-point Time interval for validity of the parameters
Data description				
NUM_RECEPTORS	Int			Number of receptors N_r on this feed
Data				

FEED_NUM	Int			Feed number
RECEIVER_ID	Int(N_r)			Receiver properties.
BEAM_ID	Int(N_r)			2D antenna beam response.
BEAM_OFFSET	Double(2,N_r)	rad	DIRECTION	Beam position offset (on sky but in antenna reference frame)
(FOCUS_REFERENCE)	Double(3,N_r)	m		Focus reference position
(ILLUM_OFFSET)	Float	m		Illumination offset
(ILLUM_OFFSET _PA)	Float	rad		Illumination offset position angle
POLARIZATION_TYPE	String(N_r)			Type of polarization to which a given RECEPTOR responds
POL_RESPONSE	Complex(N_r, N_r)			Feed polarization response
POSITION	Double(3)		POSITION	Offset of the feed relative to the feed reference
RECEPTOR_ANGLE	Double(N_r)	rad		The reference angle for polarization

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 and for each SPECTRAL_WINDOW_ID . Consequently, FEEDNUM should be non-zero only in the case of feed arrays, i.e. multiple, simultaneous beam on the sky at the same frequency and polarization.

ANTENNA Antenna identifier, as indexed from an element in the ANTENNA_ARRAY collection in the CONFIG_DESCRIPTION table.

FEED_ID Feed identifier, as indexed from an element in the FEED_LIST collection in the the CONFIG_DESCRIPTION table.
Note that FEED_ID is zero-based relative to each combination of the 5 other *key*s.

SPECTRAL_WINDOW_ID Spectral window identifier. A value of -1 indicates the row is valid for all spectra windows. If a specific spectral window has its own feed characteristics these have to bedefined separately using a value 0, to override the global specifications for that specific window.

TIME Mid-point of time interval for which the feed parameters in the row are valid. The same reference used for the TIME column in the MAIN must be used.

INTERVAL Time interval.

NUMRECEPTORS Number of receptors N_r on this feed. See POLARIZATIONTYPE for further information.

FEEDNUM Feed number, 0 based. For ALMA it is always 0.

RECEIVER_ID Receiver identifier(s) for each receptor.
N.B.: in MSv2.0 RECEIVER_ID is indexed in the SPECTRAL_WINDOW table.

BEAM_ID Beam identifier. Points to a BEAM sub-table defining the far field voltage pattern and th polarization response for this FEED. TBD

BEAMOFFSET Beam position offset, as defined on the sky but in the antenna reference frame.

FOCUSREFERENCE Focus reference position. Mandatory for ALMA. Notice that receptor independent part in the focus position is specified in the FOCUS sub-table.

ILLUM_OFFSET Illumination offset

ILLUM_OFFSET_PA Illumination offset position angle.

POLARIZATIONTYPE Polarization type to which each receptor responds (e.g. “R”, “L”, “X”, “Y”) . This is the receptor polarization type as recorded in the nal correlated data (eg “RR”); i.e. measured after all polarization combiners.

POLRESPONSE Polarization response at the center of the beam for this feed. Expressed in a linearly polarized basis using the IEEE convention.

POSITION Offset of the feed relative to the feed reference position for this antenna (see ANTENNA subtable).

RECEPTOR ANGLE Polarization reference angle. Converts into parallactic angle in the sky domain.

Field position for each source.

FIELD Field position for each source.				
Name	Format	Units	Measure	Comments
Columns				
Data				

FIELD_NAME	String			Name of field
CODE	String			Special characteristics of field
TIME	Double	s	EPOCH	Time origin for directions and rates
NUM_POLY	Int			Series order N_{poly}
DELAY_DIR	Double(2,N_{poly}+1)	rad	DIRECTION	Direction of delay center
PHASE_DIR	Double(2,N_{poly}+1)	rad	DIRECTION	Phase center
REFERENCE_DIR	Double(2,N_{poly}+1)	rad	DIRECTION	Reference center
(SOURCE_ID)	Int			Source identifier
(EPHMERIS_ID)	Int			Ephemeris identifier
(*ASSOC_FIELD_ID*)	Int(*)			Associate field identifier
(ASSOC_NATURE)	String			Nature of the association
Flags				
FLAG_ROW	Bool			The row flag

Notes: The FIELD table defines a field position on the sky. For interferometers, this is the correlate field position. For single dishes, this is the nominal pointing direction.

FIELD_NAME Field name; user specified.

CODE Field code indicating special characteristics of the field; user specified. The notion of pattern contained in the concept of OBSERVATION (ALMA glossary) is associated here to the polynomial forms which describe an elementary pattern. OBSINTENT belongs to the STATE sub-table. ALMA can use here this CODE as needed.

TIME Time reference for the directions and rates. Require to use the same TIME Measure reference as in the MAIN .

NUM_POLY Series order for the *DIR columns.

DELAY_DIR Direction of delay center; can be expressed as a polynomial in time. Final result converted to the defined Direction Measure type.

PHASE_DIR Direction of the phase center; can be expressed as a polynomial in time. Final result converted to the defined Direction Measure type.

REFERENCE_DIR Reference center; can be expressed as a polynomial in time. Final result converted to the defined Direction Measure type. Used in single-dish to record the associated reference direction if position switching has already been applied. For interferometric data, this is the original correlated field center, and may equal the DELAYDIR or PHASEDIR.

SOURCE_ID Points to an entry in the SOURCE sub-table; a value of -1 indicates there is no corresponding source defined (as this will be the case for the radiometric data)

EPHMERIS_ID Points to an entry to the EPHEMERIS sub-table, which defines the ephemeris used to compute the field position. Useful for moving, near-field objects, where the ephemeris may be revised over time

ASSOC_FIELD_ID Associated field identifier(s) for fields(s) which are related in some fashion (e.g. field the phase reference source when using fast switching mode)

ASSOC_NATURE Nature of the association e.g. phase reference source

FLAG_ROW True if in this row are invalid, else False. Does not imply flagging in MAIN

Flag commands.

FLAG_CMD Flag commands.				
Name	Format	Units	Measure	Comments
Columns				
Key				
TIME INTERVAL	Double Double	s s	EPOCH	Time interval mid-point Time interval
Data				
TYPE REASON LEVEL SEVERITY APPLIED COMMAND	String String Int Int Bool String			FLAG or UNFLAG Flag reason Flag level Severity code True if applied in MAIN Flag command

Notes: This sub-table defines global flagging commands which apply to the data in MAIN

TIME Mid-point of the time interval to which this flagging commands apply. Required to use the same TIME Measured reference as used in the MAIN .

INTERVAL Time interval

TYPE Type of flag command, representing either a flagging ("FLAG") or an un-flag ("UNFLAG") operation.

REASON Flag reason; user specified

LEVEL Level (≥ 0); reflects different revisions of flags which have the same REASON

APPLIED True if flag has been applied to MAIN , and update in FLAG_CATEGORY and FLAG. False if this flag has not been applied to MAIN

COMMAND Global flag command, expressed in the standard syntax for the data selection, as adopted within the project as a whole.

Remark:
With radiometric data a in TOTAL_POWER_MONITORING table it will be necessary to extend the tool to be able also to flag the radiometric data.

Focus information.

FOCUS Focus information.				
Name	Format	Units	Measure	Comments
Columns				

Key				
ANTENNA _ID FEED_ID TIME INTERVAL	Int Int Double Double	 s s	 EPOCH	Antenna identifier Feed identifier Interval mid-point Time interval for validity of the parameters
Data				
FOCUSPOSITION FOCUSTRACKING FOCUSMODEL _ID FOCUSOFFSET	Double(3) Bool Int Double(3)	m m		Focus position True if tracking the target position Focus identifier Focus position offset

Notes: This table contains information concerning the focus for each antenna as a function of TIME. Note that the receptor dependent part of the focus is located in the FEED table. N.B.: The number of rows in this table is expected to be significantly smaller than in the POINTING table

ANTENNA_ID Antenna number, as indexed from one element in the ANTENNAARRAY collection in the CONFIG DESCRIPTION table.

FEED_ID Feed identifier, as indexed in the FEED LIST collection in the CONFIG_DESCRIPTION table.

TIME Mid-point of time interval for which the focus parameters in the row are valid. The same reference used for the TIME column in the MAIN must be used.

INTERVAL Time interval.

FOCUS_POSITION Actual focus position.

FOCUS_TRACKING True if tracking the target position.

FOCUS_MODEL_ID Model providing the target position. The FOCUSMODEL table will be defined at a later stage. The receptor dependent part of the model is located in the FEED table. The model would includes e.g. antenna elevation and thermal effects.

FOCUS_OFFSET Offset applied in the focus position relative to the target position.

Frequency offset information

FREQ_OFFSET Frequency offset information.				
Name	Format	Units	Measure	Comments
Columns				
Key				
ANTENNA_ID	Int			Antenna identifier
FEED_ID	Int			Feed identifier
SPECTRAL_WINDOW _ID	Int			SPECTRAL WINDOW_ID
FREQ_OFFSET_ID	Int			Freq-offset identifier
TIME	Double	s	EPOCH	Time interval mid-point
INTERVAL	Double	s		INTERVAL
Data				
OFFSET	Double	Hz		Frequency offset

Notes: The table contains frequency offset information to be added directly to the defined frequency labeling in the SPECTRAL_WINDOW sub-table as a Measure offset. This allows bands with small time-variable, ad hoc frequency offsets to be labeled as the same SPECTRAL_WINDOW_ID, and calibrated together if required.
This table may also be useful for fine tuning in the definition of the spectral windows for the radiometer as small frequency offsets may be present from one such device to an other. However it may be necessary to use a more general solution by defining one spectral window for each radiometer in case there is not only a global offset from one device to another.
This table can also be used, e.g. for the frequency switch observing mode. It is indexed via the FREQ_OFFSET _ID identifier in the SWITCH_CYCLE table.

ANTENNA _ID Antenna identifier, as indexed in the ANTENNA_ARRAY collection in the CONFIG_DESCRIPTION table.
Note that if there is an offset antenna independent, e.g., in case the frequency switch observing mode is used, a value of -1 can be used for ANTENNA_ID . In case there is simultaneously an antenna-based offset and a global antenna independent offset the actual offset will be considered as the sum of both.

FEED_ID Feed identifier, as indexed from FEED in the FEED_LIST collection in the CONFIG_DESCRIPTION table.

SPECTRAL_WINDOW _ID Spectral window identifier

FREQ_OFFSET_ID Frequency offset identifier, as indexed in the FREQ_OFFSET_ARRAY collection in SWITCH_CYCLE table.

TIME Mid-point of the time interval over which the data in the row are valid. Require to use the same TIME Measure reference as in MAIN

INTERVAL Time interval

OFFSET Frequency offset to be labeled to the frequency axis for this spectral window, as defined in the SPECTRAL_WINDOW sub-table. Required to have the same Frequency Measure reference as CHAN_FREQ in that table.

Remark:

In the MeasurementSet (MSv2.0) this table is baseline-based to support various peculiarities of practice in Doppler tracking, especially in VLBI. In that context it has also a FIELD_ID attribute. This level of generality is not required for the AEDF.

Antenna gain tracking information.

GAIN_TRACKING Antenna gain tracking information.				
Name	Format	Units	Measure	Comments
Columns				
Key				
ANTENNA_ID FEED_ID SPECTRAL_WINDOW _ID TIME INTERVAL	Int Int Int Double Double	 s s	 EPOCH	Antenna identifier Feed identifier Spectral window identifier Time interval mid-point Time interval
Data				
ATTENUATOR SAMPLING_LEVEL DELAYOFF1 DELAYOFF2 PHASEOFF1 PHASEOFF2 RATEOFF1 RATEOFF2 *(PHASE_REF_OFFSET)*	Float Float Double Double Double Double Double Double Double	 s s rad rad rad s^{-1} rad s^{-1} rad		Attenuator Sampling level Delay offset of antenna Relative delay offsets Phase offset on LO Phase offset on LO2 Phase rate offset on LO Phase rate offset on LO2 A priory phase offset applied

Notes: Time variable gain trackings. These data are indexed via the TIME *key* in the MAIN table, via the ANTENNA_ARRAY and FEED_LIST collections in the CONFIG_DESCRIPTION table (ANTENNA_ID and FEED _ID) and via the DATA_DESCRIPTION table (SPECTRAL_WINDOW _ID).

ANTENNA_ID Antenna identifier

FEED_ID Feed identifier

SPECTRAL_WINDOW _ID Spectral window identifier

TIME Mid-point of time interval for which the feed parameters in the row are valid. The same reference used for the TIME column in the MAIN must be used.

INTERVAL Time interval. ATTENUATOR Attenuator (TBD)

SAMPLING_LEVEL Sampling level of the digitizer

DELAYOFF1 Delay offset. It changes when the antenna is moved. Note that the offsets are applied in addition to the geometrical values.

DELAYOFF2 Relative delay offsets in basebands; note that this should not change when the antenna is moved.

PHASEOFF1 Phase offset applied in addition to the geometrical values on LO1, in addition of normal phase tracking

PHASEOFF2 Phase offset applied in addition to the geometrical values on LO2, in addition of normal phase tracking

RATEOFF1 Phase rate offset applied in addition to the geometrical values on LO1, in addition of geometrical tracking..

RATEOFF2 Phase rate offset in addition of geometrical tracking. It is physically the same thing as a frequency offset; it adds up to the nominal LO frequencies as specified in the spectral windows for each baseband

PHASE_REF_OFFSET A priori phase offset correction applied when using phase referencing (Fast-Switching observing mode) This phase offset is determined from the last observations on a FIELD_ID which is an associated FIELD of the current FIELD

Remarks:

1. These delay and phase offsets being applied, in hardware, per baseband, SPECTRAL_WINDOW_ID corresponds to CHANNEL-ZERO associated spectral windows. (Would, between antennas, uncorrelated phase and delay offsets with a spectral dependency be present through the baseband, this information would be stored referring to associated spectral windows with the proper spectral resolution. In that case the dimension N_f needs to be added.)
2. Phase tracking due to the geometry, e.g. in the On-The-Fly with cross-correlation observing mode, is set in the FIELD table. Other effects for phase-tracking are considered here in this GAIN_TRACKING table. This includes atmospheric on-line phase referencing (Fast-Switching phase referencing observing mode).

History information.

HISTORY History information.				
Name	Format	Units	Measure	Comments
Columns				
Key				
TIME EXECUTE_ID	Double Int	s	EPOCH	Time-stamp for message Execute identifier
Data				
MESSAGE PRIORITY ORIGIN OBJECT_ID APPLICATION CLI_COMMAND APPPARMS	String String String String String String String			Log message Message priority Code origin Originating objectID Application name CLI command sequence Application parameters

Notes: The table contains contains associated history information for the AEDF

TIME Time-stamp for the history record. Require to use the same TIME Measure reference as in MAIN

EXECUTE_ID EXECUTE identifier

MESSAGE Log message

PRIORITY Message priority, with allowed types ("DEBUGGING", "WARN", "NORMAL", or "SEVERE")

ORIGIN Source code origin from which message originated

OBJECT_ID Originating objectID, if available, else blank

APPLICATION Application name

CLI_COMMAND CLI command sequence invoking the application

APP_PARMS Application parameters values, in the adopted project-wide format.

Observation summary information.

OBS_SUMMARY Observation summary information.				
Name	Format	Units	Measure	Comments
Columns				
Key				
EXECUTE_ID SCAN_NUMBER	Int Int			Execute identifier Scan number
Data description				
NUM_INTEGRATION	Int			Number of integrations N_{integ}
Data				
TIME INTERVAL OBS_NUM OBS_INTENT OBS_MODE NUM_SUBINTEG	Double Double Int String String Int(N_{integ})	s s	EPOCH	Time interval mid-point Time interval Observation number Goal of Observation Observing mode Number of sub-integrations $N_{subinteg}$
Flags				
FLAG_ROW	Bool			The row flag

Notes: This table stores summary information for each observation (as defined in the ALMA glossary). It gives a mean to get some basic information in the context of the project structure, The TIME,INTERVAL pair of attributes allows to know which part of the MAIN table is relevant for a given observation.

EXECUTE_ID Identifier pointing to one row in the EXECUTE_SUMMARY table.

SCAN_NUMBER Scan number, as enumerated during the execution of the Executing Block. The pair EXECUTE_ID ,SCAN_NUMBER used as a *key* corresponds to a single row in the SCAN SUMMARY table.

NUM_INTEGRATION Number of integrations for this observation.

TIME Time mid-point interval. The same reference used for the TIME column in the MAIN must be used.

INTERVAL Time interval. This is the difference between the end and begin TIME of the observation

OBS_NUMBER Observation number. The enumeration is relative to a SCAN as it was enumerated during the execution of the SchedulingBlock.

OBS_INTENT Observation intent. This item is present here to allow querying data by accessing efficiently the MAIN table. This OBS INTENT is a duplicate of the OBSINTENT which is in the STATE table.

OBS_MODE Observing mode. This item is present here to allow querying data by accessing efficiently the MAIN table. This OBSMODE is a duplicate of the OBSMODE which is in the STATE table.

NUM_SUBINTEG Number of sub-integrations for each integration.

FLAG_ROW True if data in this row are invalid, else False. Does not imply flagging in the MAIN.

Observation unit characteristics.

OBSERVATION_UNIT Observation unit characteristics.				
Name	Format	Units	Measure	Comments
Columns				
Key				
TIME	Double	s	EPOCH	Time interval mid-point
INTERVAL	Double	s		Time interval
Data description				
(NUM_LINES)	Int			Number of spectral lines N_{line}
(NUM_STOKES)	Int			Number of stokes parameters N_{sto}
(NUM_EPOCH)	Int			Number of epochs N_e for provided uxes
(NUM_SCHEDBLOCK)	Int			Number scheduling blocks N_{SB} for this observation unit
(NUM_XBINTENT)	Int			Corresponding number of selected EXECUTE intents
(NUM_SCANINTENT)	Int			Number of selected SCAN Intents
Data				

(SOURCE_NAME)	Int			Name of the selected source
(TRANSITION)	Int(N_{line})			The N_{line} selected transition(s)
(TIME_RANGE)	Double(2,N_e)	s	EPOCH	The N_e TIMERANGEs
(DATADESC_ID)	Int			The selected data description identifier
(CORR_TYPE)	Int(N_{sto})			Polarization of correlation
(OBS_PROJECT SET)	String(*)			Set of selected ObsProject(s)
(SCHEDULE_ID_ARRAY)	Int			Collection (type array) of SCHEDULE_ID identifiers
(SB_ASSOC NATURE)	String			Nature of the association of SBs
(SCHEDULE_ID)	Int(N_{SB})			SCHEDULE_ID
(XB_INTENT ARRAY)	String(N_{XBint})			Collection (array) of XB intents
(SCAN_INTENT ARRAY)	String(N_{scint})			Collection (array) of SCAN intents
(SCAN_ASSOC NATURE)	String			Nature of the association of scans
(ASSOC_OBS_UNIT)	Int			List of associated Observation Units
(OBS_UNIT_ASSOC_NATURE)	String			Nature for the ASSOC_OBS_UNIT association
STANDARD_MODE_ID	Int			Standard mode identifier
PROCESSING_METHOD	String			Processing method
PROCESSING_VERSION	Float			Processing method version
PROCESSING_EXECUTE	String			Processing script actually executed
PROCESSING_DATE	String			Date of processing
PROCESSING_HISTORY				Processing log

Notes: This table defines Observation-Units and provides informations to process them. The instance of an Observation-Unit is a tuple which associates all data, or a subset of them, this association having its own logic as defined by the PI or the author who builds the data-set. The association is valid only if it satises a standard mode such that it can be processed.
The selection of data in the data-set to define the association is obtained by a set of lters, in the scienc target space or according to the logic in the Observing Programmes (project structures) or with these two types of lters together. At least one amongst these lters is mandatory.
The atomic constituent in an Observation-Unit instance is a SCAN.
An Observation-Unit includes all the required internal dependencies for the processing. The sequence of ExecuteBlocks in the EXECUTE_SUMMARY table defines the data-set. The order in that sequence is independent of the processing. For example it may be driven by the observations themselves which would depend of experimental conditions such as weather conditions. In contrast the sequence of associated data in this OBSERVATION_UNIT table is driven with a goal to achieve a scientific result from pipeline processing.
Note that the items for the processing are mandatory.

OBS_PROJECT_SET Set of ObsProject identification(s).

TIME Time mid-point. The same reference used for the TIME column in the MAIN table must be used. N.B.: this may concern more than one pair of EXECUTE_SUMMARY MAIN table if this OBSERVATION_UNIT gathers data from different ObsProject (case of a virtual project).

INTERVAL Time interval.
N.B.: TIME with INTERVAL plays the role of a first lter to select a subset of data within a project If the time range includes the start time if the first XB and the end time of the last XB and with NUM_SCHEDBLOCK=0 and or NUM_XBINTENT=0 and or NUMSCANINTENT=0. the full ObsProject is considered for processing.
TIME with INTERVAL may be set to select data from different ObsProjects in case of virtual projects.

NUM_LINES Number of selected spectral line transitionsN_{line}.

NUM_SCHEDBLOCK If NUMSCHEDBLOCK= 0 all the Scheduling Blocks with all their EXECUTE_ID identifiers are considered. These EXECUTEID must have been indexed in the EXECUTE SUMMARY table.

NUM_XBINTENT If NUMXBINTENT=0 then the full list of EXECUTE_ID will be used.

NUM_SCANINTENT If NUMSCANINTENT=0 then all the scans executed will be considered. If in addition NUMXBINTENT>0 then only the ensemble of the scans for this (these) XBs are selected.

SOURCE_NAME Source name, as specified for the observations

TRANSITION Name(s) for the N_{line} transition(s)

TIME_RANGE The N_e time range(s). Used e.g. for science studies for source time variabilities. The output image has in this case its last axis of type TIME and size N_e.

DATADESC_ID Data description identifier.

CORR_TYPE An integer for each correlation product indicating the Stokes type as defined in the Stokes class enumeration. The size of the STOKES axis of the output image(s) corresponds to N_{sto}.

SCHEDULE_ID_ARRAY Collection of scheduling blocks. The order in this array may indicate a dependency the last in the array depending to the previous one.

SB_ASSOC_NATURE String to defining the nature of the association of Scheduling Blocks. Example multi antenna-configuration

EXECUTE_ID_ARRAY Collection of ExecutingBlocks identied by their identifiers i.e. their row number in the EXECUTE_SUMMARY table. The order in this array may indicate a dependency.

XB_INTENT_ARRAY EXECUTE intent(s) selected if not all XBs required. If this item is empty then all the ExecutionBloks are considered.

SCAN_INTENT_ARRAY SCAN intent(s) selected whithin the XBs. The order in the list may provide dependency within a Scheduling Block.

SCAN_ASSOC_NATURE Nature of that association of scans. This can be used e.g. to proccess the data on a restricted region in the target space (sky and or spectral regions).

ASSOC_OBS_UNIT List of associated Observation-Units in case of recursivity. It defines the union of observing units identifiers to be processed. The order in the list oers the possibility to indicat dependencies (same rule as in Makeles). Example: begin by processing a set of scans on a source used for phase-referencing and then process the interleaved scans for the main target, applying interpolated results obtained from the rst set of scans processed.

OBS_UNIT_ASSOC_NATURE Nature of the association of the observation units. With the example given in the description of the ASSOCOBSUNIT item, the nature would be Phase referencing

STANDARD_MODE_ID Standard processing mode to use. Selected by the user with the observation preparation tool (TBD).

PROCESSING_METHOD Processing method (name of the processing script) The method could be a list of successive categorized methods (TBD)

PROCESSING_VERSION Version for that method.

PROCESSING_EXECUTE Link the the script actually used.

PROCESSING_DATE Date of the processing.

PROCESSING_HISTORY History of the processing.

Antenna pointing information.

POINTING Antenna pointing information.				
Name	Format	Units	Measure	Comments
Columns				
Key				
ANTENNA_ID TIME INTERVAL	Int Double Double	 s s	 EPOCH	Antenna identifier Time interval mid-point Time interval
Data				
NAME NUM_POLY TIME_ORIGIN POINTING_DIRECTION	String Int Double Double(2,(N_{poly}+1))	 s rad	 EPOCH DIRECTION	Pointing position desc. Series order N_{poly} Origin for the polynomial Antenna pointing direction
TARGET *(OFFSET)*	Double(2,(N_{poly}+1)) Double(2,(N_{poly}+1))	rad rad	DIRECTION DIRECTION	Target direction A priori pointing corrections
(SOURCEOFFSET) *(ENCODER)* POINTINGMODEL_ID POINTINGTRACKING	Double(2,(N_{poly}+1)) Double(2) Int Bool	rad rad	DIRECTION DIRECTION	Offset from source Encoder values Pointing model identifier True if tracking the target direction
PHASETRACKING	Bool			True if tracking the phase reference position
(OVER THE TOP)	Bool			True if over the top

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

ANTENNA_ID Antenna identifier

TIME Mid-point of the time interval over which the data in the row are valid. Require to use the same TIME Measure reference as in MAIN

INTERVAL Time interval

NAME Pointing direction name; user specified

NUM_POLY Series order for the polynomial expressions in POINTING_DIRECTION[11] and OFFSET
TIME_ORIGIN Time origin of the polynomial expression in POINTING_DIRECTION and OFFSET.
POINTING_DIRECTION Antenna pointing direction, optionally expressed as polynomial coefficients. The final result is interpreted as a Direction Measure using the specified Measure reference
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...
OFFSET The a priori pointing corrections applied by the telescope in pointing to the DIRECTION position, optionally expressed as polynomial coefficients. The final result is interpreted as a Direction Measure using the specified Measure reference.

Note that these are pointing offsets applied on top of the pointing model. They correct anything not accounted by the model (thermal variations, atmospheric refraction, ...)
SOURCE_OFFSET The commanded offset from the source position, if offset pointing is being used.
ENCODER The current encoder values on the primary axes of the mount type for the antenna, expressed as a Direction Measure. These are absolute values.
POINTING_MODEL _ID Pointing model identifier
POINTING_TRACKING True if tracking the nominal pointing position
PHASE_TRACKING True if tracking the nominal phase reference direction
OVER_THE_TOP True if the antenna was driven to this position over the top (az-el mount).

Polarization information.

POLARIZATION Polarization information.				
Name	Format	Units	Measure	Comments
Columns				
Data description				
NUM_CORR	Int			Number of correlation cross-products N_c
Data				
CORR_TYPE CORR_PRODUCT	Int(N_c) Int(2,N_c)			Polarization of correlation Receptor cross-products
Flags				
FLAG_ROW	Bool			The row flag

Notes: This table defines the polarization labeling of the data; it is directly indexed from the DATA_DESCRIPTION table via POLARIZATION_ID, the DATADESC_ID being indexed in the *DATA DESCRIPTION* ARRAY collection in the CONFIG DESCRIPTION table.

NUM_CORR The number of correlation polarization products. For example (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 Stokes class enumeration.

[11]Originaly named DIRECTION in MS2

CORR_PRODUCT Pair of integer for each correlation product, specifying the receptor from which the signal originated. The receptor polarization is defined in the POLARIZATIONTYPE 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 if the data in this row are not valid, but does not imply the flagging of any DATA in MAIN .

Processor characteristics.

PROCESSOR Processor characteristics.				
Name	Format	Units	Measure	Comments
Columns				
Data				
TYPE	String			Processor type
SUB_TYPE	String			Processor sub-type
MODE_ID	Int			Mode identifier
Flags				
FLAG_ROW	Bool			The row ag

Notes: This table holds summary information for the back-end processing device used to generate the basic data in the MAIN and TOTALPOWERMONITORING table. Such devices include correlators, radiometers, spectrometers amongst others. The PROCESSOR_ID identifier is indexed in the CONFIG DESCRIPTION table.

TYPE Processor type; reserved *keywords* include:
CORRELATOR[12] - interferometric correlator,
SPECTROMETER[13] - single-dish correlator,
RADIOMETER[14] - generic detector/integrator,
Note that ALMA uses the reserved type RADIOMETER (generic detector/integrator) for the data originating from the radiometers to monitor the atmosphere and also for the total power detected over the band, directly from the astronomical receivers.

SUB_TYPE Index used in specialized sub-table named as *subtype_type* which contains time independent processor information applicable to the current data record (e.g. a ALMACORRELATOR subtable).
The time-dependent information for each device family is contained in other tables, dependent on the device type.

MODE_ID Identifier used in a specialized sub-table name as *subtype_type_mode*, containing information on the processor mode applicable to the current data record. (e.g. a ALMA_CORRELATOR_MODE sub-table).

FLAG_ROW Row flag. True if data in the row is not valid, but does not imply flagging in the MAIN or TOTAL_POWER_MONITORING .

[12]Attribute CORRELATOR reserved for *key*word TYPE in MS2 [13]Attribute SPECTROMETER reserved for *key*word TYPE in MS2
[14]Attribute RADIOMETER reserved for *key*word TYPE in MS2

Receiver properties.

RECEIVER Receiver properties.				
Name	Format	Units	Measure	Comments
Columns				
Key				
TIME INTERVAL	Double Double	s s	EPOCH	Time interval mid-point Time interval
Data description				
NUM_LO	Int			Number of local oscillators N_{LO}
Data				
NAME RECEIVER_BAND FREQ_LO SIDEBAND_LO *(DEWAR_NAME)* TDEWAR *(STABILITY_DURATION)* *(STABILITY)*	String String Double(N_{LO}) Int(N_{LO}) String Float Double Double	 Hz K s		Receiver names Receiver band Frequency LO Sideband of local oscillators Dewar name Receiver dewar temperature Time scale for the relative stability Flag for stability
Flags				
(STABILITY_FLAG)	Bool			Flag for stability

Notes: This table holds summary information with measured properties of the individual receivers. Indexed in the FEED table.

TIME Mid point of the time interval over which the data in the row are valid. Require to use the same TIME Measure reference as in MAIN

INTERVAL Each time a receiver is tuned there is a new entry (RECEIVER_ID) in this table.

NUM_LO Number of local oscillators

NAME Receiver name

RECEIVER_BAND Receiver band. For the frequency bands they are at present identified by an integer in the range 1 to 9.

FREQ_LO Sideband of local oscillators

SIDEBAND_LO Sideband (+1,-1) of local oscillators

DEWAR_NAME Dewar name

TDEWAR Receiver Dewar temperature

STABILITY_DURATION Time scale for receiver stability

STABILITY Root square of the Allan variance for this time scale

STABILITY_FLAG Flag for STABILITY

Scan summary information.

SCAN_SUMMARY Scan summary information.				
Name	Format	Units	Measure	Comments
Columns				
Key				
EXECUTE_ID	Int			Execute identifier
Data description				
NUM_OBS	Int			Number of OBSERVA-TIONs N_{obs}
Data				
TIME	Double	s	EPOCH	Time interval mid-point
INTERVAL	Double	s		Time interval
SCAN_NUMBER	Int			Scan number
SCAN_INTENT	String			Intent for the scan
FIELD_NAME	String			Name of field
OBS_NUM_SET	Int(N_{obs})			Collection of OBS NUM
Flags				
FLAG_ROW	Int			The row flag

Notes: This table stores summary information for each scan. It may need to be regenerated if data are removed however.

EXECUTE_ID Identifier pointing to one row in the EXECUTE_SUMMARY table.

TIME Time mid-point interval. The same reference used for the TIME column in the MAIN must be used.

INTERVAL Time interval. This is the difference between the end and begin TIME of the scan.

SCAN_INTENT Scan intent. Since this is the unique location in the AEDF where this item is present, this SCAN SUMMARY table is mandatory.

SCAN_NUMBER Scan number as enumerated during the execution of the SchedulingBlock.

FIELD_NAME Field name, as specified during the execution of the Scheduling Block.

OBS_NUM_ SET Collection of OBSNUM numbers for this scan in the data set. These do not need to be given in a specific order but obviously their values must be unique.

FLAG_ROW True if data in this row are invalid, else False. Does not imply flagging in the MAIN.

Schedule characteristics.

SCHEDULE Schedule characteristics.				
Name	Format	Units	Measure	Comments
Columns				
Data				
SCHEDULE_INTENT	String			Schedule intent
SCHEDULE_TYPE	String			Schedule type
SCHEDULE_DURATION	Float	s		Approximate duration if executed
SCHEDULE	String			Scheduling Block
NUM_SCAN	Int			Number of different scans N_{scan} in this Scheduling Block
SCAN_INTENT	String(N_{scan})			Intent of each scan in the Scheduling Block
OBSERVING_MODE	String(*)			Observing mode

Notes: This table contains the list of scheduling blocks generated by observation preparation tools. A summary of informations is attached to each scheduling block. It is indexed directly in the EXECUTE_SUMMARY table. The associations of Scheduling Blocks for processing are defined in the OB-SERVATION UNIT table using lists of SCHEDULE _ID identifiers.

SCHEDULE_INTENT Goal of Scheduling Block (SB). Several SBs may share a common SCHEDULE INTENT.

SCHEDULE_TYPE The schedule type, with the current reserved types VLBACRD[15] VEX[16] WSRT[17] ATNF[18] For ALMA the reserved type(s) are TBD.

SCHEDULE_DURATION Approximate duration which would be required to execute this SCHEDULE.

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

NUMSCAN Number of SCANs defined for the scheduling. **SCANINTENT** List of intents with these different SCANs.

OBSERVING_MODE Observing mode, e.g. singleField, mosaic (origin: OT) One SB cannot have more than one observing mode Example: FS (for FastSwitching), RC (for with radiometric corrections), FSRC (for both FS and RC), mosaicingPointed, mosaicingOTF etc.

Remarks:

1. Note that the same SCAN may be executed in different SCHEDULEs! This suggests to have a SCAN sub-table and a reference here via a list of SCAN _ID.
2. This table could be generated at the stage of the preparation of the projects while the observations have not yet started.

[15]Attribute VLBA-CRD reserved for *key*word , in MS2

[16]Attribute VEX reserved for *key*word , in MS2

[17]Attribute WSRT reserved for *key*word , in MS2

[18]Attribute ATNF reserved for *key*word in MS2

Seeing information.

SEEING Seeing information.				
Name	Format	Units	Measure	Comments
Columns				
Key				
TIME INTERVAL	Double Double	s s	EPOCH	Time interval mid-point Time interval
Data description				
NUM_BASE_LENGTH	Int			Number of baselengths N_{basept} points sampling the function
Data				
(BASELENGTH) *(PHASERMS)* SEEING EXPONENT	Float(N_{basept}) Float(N_{basept}) Float Float	m deg		Base lengths rms phase fluctuation Seeing parameter, deduced for the LO1 freq. Exponent of the power law of the phase structure function

Notes: This table holds information on the phase structure and seeing.

TIME Time midpoint

INTERVAL Time interval over which the seeing information is extracted

NUM_BASE_LENGTH Number of baselength bins N_{basept} determining the rms fluctuations

BASELENGTH Actual baselengths on which the rms phase is determined.

PHASERMS rms phase uctuations for these baselengths.

SEEING Seeing parameter, deduced for the LO1 frequency.

EXPONENT Exponent of power law of the phase structure function

Comment: The phase structure function could be computed on-the-y when accessing the data in the MAIN and or the TOTAL_POWER_MONITORING table(s). However the intent with this table is to get efficiently the information without reading data!

Source information.

SOURCE Source information.				
Name	Format	Units	Measure	Comments
Columns				
Key				
SOURCE_ID TIME INTERVAL SPECTRAL_WINDOW_ID	Int Double Double Int	 s s	 EPOCH	Source identifier Time interval mid-point Time interval Spectral window identifier
Data description				
NUM_LINES NUM_EPOCH	Int Int			Number of spectral lines N_{line} Number of epochs N_e for provided uxes
Data				

SOURCE_NAME	String			Name of source as given during observations
(CATALOG)	String			Catalog name
CALIBRATION_GROUP	Int			number grouping for calibration purpose
CODE	String			Special characteristics of source. eg. Bandpass calibrator
DIRECTION	Double(2)	rad	DIRECTION	Direction (eg. RA DEC)
(POSITION)	Double(3)	m	POSITION	Position (eg for solar system objects)
PROPER_MOTION	Double(2)	rad s^{-1}		Proper motion
(TRANSITION)	String(N_{line})			Transition name
(REST_FREQUENCY)	Double(N_{line})	Hz	FREQUENCY	Line rest frequency
(SYSVEL)	Double(N_{line})	m s^{-1}	RADIAL VELOCITY	Systemic velocity at reference
(SOURCE_MODEL)	String			Default csm
(DELTAVEL)	Double	m s^{-1}		Velocity resolution as required by the observer
(RANGEVEL)	Double(2)	m s^{-1}		Velocity range form observer (min,max)
(SOURCE_PARAMETER_ID)				Source parameter identifier

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

SOURCE_ID Source identifier

TIME Mid-point of the time interval over which the data in the row are valid. Require to use the same TIME Measure reference as in MAIN

INTERVAL Time interval

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 N_{line} associated with the source and spectral window id. combination

NUM_EPOCH Number of epochs N_e for provided fluxes. Use 0 if the source parameters are unknown.

SOURCE_NAME Source name, user specified.

CATALOG Calalog name from which the source was extracted.

CALIBRATION_GROUP Calibration group number to which this source belongs, user specified CODE Source code, used to describe any characteristics for the source, such as the nature of the calibrator.Reserved *key*word, including BANDPASS CAL[19] .

DIRECTION Source direction at this TIME

POSITION Source position (x,y,z) at this TIME (for near-field objects)

PROPER_MOTION Source proper motion at this TIME

TRANSITION Transition names applicable for this spectral window (e.g., v=1, J=1-0, S I O)

REST_FREQUENCY Rest frequencies for the transitions

SYSVEL Systemic velocity for each transition

DELTAVEL Velocity resolution as required by the observer

RANGEVEL Velocity range from observer (min,max)

SOURCE_MODEL Reference to an assigned component source model table

SOURCE_PARAMETER_ID Should be mandatory for sources used as ux calibrators.

Remark:
SOURCE_ID being associated with a SPECTRAL_WINDOW_ID (unless SPECTRAL_WINDOW_ID =-1 for continuum experiments) SOURCEPARAMETER_ID is intended only for parameters which are considered as useful in the frequency range of the spectral window.

Source parameter information.

SOURCE_PARAMETER Source parameter information.				
Name	Format	Units	Measure	Comments
Columns				
Key				
TIME *(TIME_EXTRA_PREC)* INTERVAL	Double Double Double	s s s	EPOCH	Time interval mid-point extra TIME precision Time interval
Data description				
NUM_FREQ NUM_STOKES NUM_DEP	Int Int Int			Number N_{fp} of frequencies to which parameters are provided Number of stokes parameters N_{sto} Number $_{Ndep}$ of dependencies for bootstraped fluxes
Data				
SOURCE_STOKES SOURCE_FLUX *(SOURCE_SIZE)* *(SOURCE_PA)* DEP_SOURCEPAR _ID	Int(N_{sto}) Float(N_{fp},N_{sto}) Float(2,N_{fp}) Float(2,N_{fp}) Int(N_{dep})	 Jy arcsec rad rad		Stokes _IDentication Flux values Sizes along major and minor axis Position angle major axis Dependent source identifier

Notes: This source parameter table provides information for calibration sources. At the time of the observations it is a priori information. Bootstrapped fluxes are stored in this table as well. There is an optional reference to an entry in this table in the SOURCE table.

[19]Attribute BANDPASS CAL reserved for *key*word CODE in MS2

TIME Time when the source parameters have been measured

TIME_EXTRA_PREC Extra precision time

INTERVAL Interval over which the parameters are believed valid. If the degree of variability of for the source is unknown, the attribute to that *key* would be set to 0.

NUM_FREQ Number of frequencies N_{fp} for which the fluxes are provided

NUM_STOKES Number of stokes parameters $_{Nsto}$ with known fluxes

NUM_DEP Number of dependencies N_{dep} for bootstrapped fluxes. NUMDEP=N_{dep}=0 for primary calibrators. The number of dependencies is internal i.e. must refer to results from data in the AEDF data set itself.

SOURCE_STOKES List of stokes parameters (with the standard axis convention).

SOURCE_FLUX Source fluxes

SOURCE_SIZE Source size (would negative values be acceptable to tell they are only upper limits?)

SOURCE_PA Position angle of major axis

DEP_SOURCEPAR _ID List of identifiers for the dependencies. This list is not empty only for bootstrapped fluxes.

Remark:
During actual observations, TIME here in SOURCE_PARAMETER will indeed not belong to the time INTERVAL of the SOURCE sub-table. This does not prevent refining the calibration subsequently by appending new rows in this SOURCE_PARAMETER sub-table and by extending accordingly the list of SOURCE_PARAMETER_ID in the SOURCE table. Results about bootstrapped fluxes during the calibration stage would also be stored in this table. The source size, or its upper limit, provides the information on valid baseline lengths in the bootstrap.

Spectral window description.

SPECTRAL_WINDOW Spectral window description.				
Name	Format	Units	Measure	Comments
Columns				
Data description				
NUM_CHAN	Int			Number of spectral channels N_f
Data				
NAME	String			Spectral window name
REF_FREQ	Double	Hz	FREQUENCY	The reference frequency
CHAN_FREQ	Double(N_f)	Hz	FREQUENCY	Center frequencies for each channel in the data matrix
CHAN_WIDTH	Double(N_f)	Hz		Channel width for each channel in the data matrix
MEAS_FREQ_REF	Int			FREQUENCY Measure ref.
EFFECTIVE_BW	Double(N_f)	Hz		The effective noise bandwidth for each channel in the data matrix
RESOLUTION	Double(N_f)	Hz		The effective spectral resolution for each channel in the data matrix
TOT_BANDWIDTH	Double	Hz		Total bandwidth for this window
NET_SIDEBAND	Int			Net sideband
(BBC_NO)	Int			Baseband converter no
(BBC_SIDEBAND)	Int			BBC sideband
IF_CONV_CHAIN	Int			IF conversion chain
FREQ_GROUP	Int			Frequency group
FREQ_GROUP_NAME	String			Frequency group name
(*DOPPLER_ID*)	Int			Doppler identifier
(*ASSOC_SPW_ID*)	Int(*)			Associate spectral-window identifier
(ASSOC_NATURE)	String(*)			Nature of association
Flags				
FLAG_ROW	Bool			The row flag

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

NUM_CHAN Number of spectral channels

NAME Spectral window name; user specified.

REF_FREQ Reference frequency. A frequency representative of the 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.

CHAN_FREQ Center frequencies for each channel in the data matrix. These can be frequencydependent, to accommodate instruments such as the WVM radiometers. 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 CHANFREQ 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 the Measure reference when Doppler tracking is used. Modified only by the MS access code.

EFFECTIVE_BW Effective noise bandwidth of each spectral channel

RESOLUTION Effective spectral resolution of each channel

NET_SIDEBAND Net sideband for this spectral window.
The reserved value are: -1 NoSB, 0 DSB, 1 LSB, 2 USB, 3 2SB LSB, 4 2SB USB, 5 2SB LSB SBsep, 6 2SB USB SBsep.

BBC_NO Baseband converter number, if applicable.

BBC_SIDEBAND Baseband converter sideband, if applicable

IF_CONV_CHAIN Identification of the electronic signal path for the case of multiple (simultaneous) IFs.

FREQ_GROUP The frequency group to which the spectral window belongs. This is use to associate spectral windows from joint calibration purposes.

FREQ_GROUP_NAME The frequency group name; user specified

DOPPLER_ID Doppler identifier defining frame information for this spectral window.

ASSOC_SPW_ID Associated spectral windows, which are related in some fashion (e.g. channel-zero).

ASSOC_NATURE Nature of the association for ASSOC_SPW_ID ; reserved *keywords* are CHANNELZERO[20] , EQUAL-FREQUENCY[21] , SUBSET[22] .

FLAG_ROW True if the row does not contain valid data.

---

[20]Attribute CHANNEL-ZERO reserved for *key*word ASSOC_NATURE in MS2

[21]Attribute EQUAL-FREQUENCY reserved for *key*word ASSOC_NATURE in MS2

[22]Attribute SUBSET reserved for *key*word ASSOC_NATURE in MS2

State information.

STATE State information.				
Name	Format	Units	Measure	Comments
Columns				
Data				
SIG	Bool			Signal
REF	Bool			Reference
CALLOAD_NUM	Int			Calibration CAL or LOAD number
(WEIGHT)	Float			Relative weight of the switch state
SUBINTEG_NUM	Int			Sub-integration number
OBS_MODE	String			Observing mode
OBS_INTENT	String			Goal of Observation
Flags				
FLAG_ROW	Bool			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 the STATE ID identifier in the MAIN table

SIG If SIG and REF are both together false there is blanking.

REF True for a generic reference phase in the switching scheme. Example: REF is TRUE when on chopper or any load, or when on a phase reference calibrator in the Fast Switching observing mode.

CALLOAD_NUM Calibration load number. The CALDEVICE table contains the temperature of the loads The CALDEVICE table is referenced via the primary *key*s in the MAIN and SYSCAL tables and this item CALLOADNUM is relevant only auto-correlation measurements i.e. when CORRELATION MODE in the CONFIG DESCRIPTION table is 1.

WEIGHT Relative weight of switch state for final processing.

SUBINTEG_NUM Used to enumerate sub-integrations (from 1 for the first sub-integration of an integration); implicitly associated to the number of sub-integration this being enumerated relative the integration.

OBS_MODE Observing mode. .
Note that here OBS is not attached to the concept of OBSERVATION given in the ALMA glossary!
Reserved attributes are TBD.

OBS_INTENT Intent of the sub-integration. Reserved attributes are TBD. Examples: FOCUS, PHASEREF, POINTING. Not required to be unique. .

FLAG_ROW True if the row does not contain valid data. Does not imply flagging in MAIN

Cycle information in switching modes.

SWITCH_CYCLE Cycle information in switching modes.				
Name	Format	Units	Measure	Comments
Columns				
Data description				
NUM_STEP	Int			Number of steps in the cy-
Data				
WEIGHT_ARRAY	Float(N_{bin})			Weight for each step
OFFSET_ARRAY	Float(N_{bin})	rad	DIRECTION	Spatial offsets for each step
FREQ_OFFSET_ARRAY	Int(N_{bin})			Freq. offset identifiers for

Notes: The SWITCH_CYCLE table describes the different steps in a complete cycle for an observation mode with modulation in frequency and/or space. It is referenced in the SWITCH_CYCLE_ARRAY collection in the CONFIG_DESCRIPTION table.
Note that phase switching for sideband separation is not considered here since it does not imply a size greater than 1 along the bin axis in the data cell.

NUM_STEP Number of steps in a full cycle.

WEIGHT_ARRAY Weight for each step in the cycle. TBD

OFFSET_ARRAY Offset in radian for each step of the complete cycle. A value of 0.0 means no offset

FREQ_OFFSET_ARRAY Collection of FREQOFFSET_ID identifiers, one for each step. A value of -1 means no frequency offset.

System calibration.

SYSCAL System calibration.				
Name	Format	Units	Measure	Comments
Columns				
Key				
ANTENNA_ID	Int			Antenna identifier
FEED_ID	Int			Feed identifier
SPECTRAL_WINDOW_ID	Int			Spectral window identifier
TIME	Double	s	EPOCH	Mid-point of time for which the set of parameters is accurate
INTERVAL	Double	s		Time interval
Data				

Name	Format	Units	Measure	Comments
(NUMLOAD)	Int			number N_{cl} of NCAL or TLOAD used
(CALLOAD)	Int(N_{cl})			List of the N_{cl} LOADs used
(FEFF)	Float(N_r)			Forward efficiency
(AEFF)	Float(N_r)			Aperture efficiency
(PHASE_DIFF)	Float	rad		Phase difference between receptor 0 and receptor 1
(SBGAIN)	Float			Relative gain of LO1 side-band
(TAU)	Float(N_r)	K		Atmospheric optical depth
(TCAL)	Float(N_r)	K		Calibration temperature
(TRX)	Float(N_r)	K		Receiver temperature
(TSKY)	Float(N_r)	K		Sky temperature
(TSYS)	Float(N_r)	K		System temperature
(TANT)	Float(N_r)	K		Antenna temperature
(TANT_TSYS)	Float (N_r)			$T_{ant} T_{sys}$
(PWVPATH)	Float(N_r)	m		Water vapor pathlength
(DPWVPATH)	Float(N_r)	m K^{-1}		Pathlength per K of radio-metric emission
(FEFF_SPECTRUM)	Float(N_r, N_f)			Forward efficiency spec-trum
(SBGAIN_SPECTRUM)	Float(N_r, N_f)			Relative gain spectrum
(TAU_SPECTRUM)	Float(N_r, N_f)			Atmosheric optical depth spectrum
(TCAL_SPECTRUM)	Float(N_r, N_f)	K		Calibration temperature spectrum
(TRX_SPECTRUM)	Float(N_f)	K		Receiver temperature spectrum
(TSKY_SPECTRUM)	Float(N_r, N_f)			Sky temperature spec-trum
(TSYS_SPECTRUM)	Float(N_r, N_f)	K		System temperature spec-trum
(TANT_SPECTRUM)	Float(N_r, N_f)	K		Antenna temperature spectrum
(TANT_TSYS_SPECTRUM)	Float(N_r,N_f)			T_{ant}/T_{sys} spectrum
		m		
(PWVPATH_SPECTRUM)	Float(N_f)	m K^{-1}		Water vapor path length spectrum
(DPWVPATH_SPECTRUM)	Float(N_f)			Path length increment spectrum

(NUM_POLY)	Int			Series order N_{poly}
(NUM_POLY_FREQ)	Int			Freq. series order N_{polyf}
(TIME_ORIGIN)	Double	s	EPOCH	Origin for the polynomial
(FREQ_ORIGIN)	Double	Hz	FREQUENCY	Origin of frequency expansion
(PHASE_CURVE)	Float (N_r,N_{poly})	rad		Phase polynomial coeff.
(DELAY_CURVE)	Float (N_r,N_{poly})	s		Delay polynomial coeff.
(AMPLI_CURVE)	Float (N_r,N_{poly})	Jy K 1		Amplitude polynomial coeff.
(BANDPASS_CURVE)	Float(N_r, N_{polyf})			Bandpass polynomial coeff.
Flags				
(PHASEDIFF_FLAG)	Bool			Flag for PHASEDIFF
(SBGAIN_FLAG)	Bool			Flag for SBGAIN
(TA U_FLAG)	Bool			Flag for TAU
(TCAL_FLAG)	Bool			Flag for TCAL
(TRX_FLAG)	Bool			Flag for TRX
(TSKY_FLAG)	Bool			Flag for TSYS
(TSYS_FLAG)	Bool			Flag for TSYS
(TANT_FLAG)	Bool			Flag for TANT
(TANT_TSYS_FLAG)	Bool			Flag for $T_{ant/}T_{sys}$
(PWVPATH_FLAG)	Bool			Flag for PWVPATH

Notes: This table contains time-variable calibration measurements for each antenna, as indexed on feed and spectral window. N_r is the number of receptors and N_f the number of frequency channels. Note that all the items in the data section are optional. However, at least one non-*key* item must be specified for a given row in this table.

ANTENNA_ID Antenna identifier, as indexed from an element in the ANTENNA_ARRAY collection in the CONFIG DESCRIPTION table.

FEED_ID Feed identifier, as indexed from an element in the FEED_LIST collection in the the CONFIG_DESCRIPTION table.

SPECTRAL_WINDOW _ID Spectral window identifier. In general this will be associated to spectral window identifiers corresponding to low spectral resolutions. .

TIME Mid-point of the time interval over which the data in the row are valid. Require to use the same TIME Measure reference as in MAIN

INTERVAL Time interval

NUM_LOAD Number of loads N_{cl} used for this calibration. (e.g. 2 for the determination of TRX using a a cold and a hot load, 1 for determination of TCAL, a hot load when using the chopper wheel method, ...)

CALLOAD List of the N_{cl} loads used for this calibration. (e.g. 2 for the determination of TRX using a a cold and a hot load, 1 for determination of TCAL, a hot load when using the chopper wheel method, ...)

FEFF Fraction of the beam volume looking at the sky.

AEFF Aperture efficiency.

PHASE_DIFF Phase difference between receptor 0 and receptor 1.

SBGAIN Sideband gain spectrum. Actual normalization to be defined.

TAU Average opacity

TCAL Calibration temperature (temperature used to convert the raw signals to T_a^* scale.)
TRX Receiver temperature.
TSKY Sky temperature
TSYS System temperature
TANT Antenna temperature
TANT_TSYS Antenna temperature over system temperature **PWVPATH**
Water vapor pathlength
DPWVPATH Water vapor pathlength increment per K of radiometric emission
FEFF_SPECTRUM Fraction of the beam volume spectrum.
SBGAIN_SPECTRUM Sideband gain spectrum. The other SPECTRAL_WINDOW_ID for the other side-band is know via ASSOC_SPW_ID
TAU_SPECTRUM Opacity spectrum
TCAL_SPECTRUM Temperature used to convert the raw signals to T_a^* scale
TRX_SPECTRUM Receiver temperature spectrum. Frequency dependency mandatory for ALMA
TSKY_SPECTRUM Sky temperature spectrum
TSYS_SPECTRUM System temperature spectrum
TANT_SPECTRUM Antenna temperature spectrum
TANT_TSYS_SPECTRUM Antenna temperature over system temperature spectrum
PWVPATH_SPECTRUM Water vapor pathlength spectrum
DPWVPATH_SPECTRUM Water vapor pathlength increment spectrum per K of radiometric emission
TAU_SPECTRUM Atmospheric optical depth spectrum
NUM_POLY Series order N_{poly} **NUM_POLY_FREQ** Frequencies
series order N_{polyf}
TIME_ORIGIN Origin of time expansion
FREQ_ORIGIN Origin of frequency expansion
PHASE_CURVE Provisional Phase polynomial coefficients
DELAY_CURVE Provisional Delay polynomial coefficients
AMPLI_CURVE Provisional Amplitude polynomial coefficients
BANDPASS_CURVE Provisional bandpass polynomial coefficients
PHASEDIFF_FLAG True if PHASEDIFF agged
PHASEDIFF_FLAG True if PHASEDIFF agged
SBGAIN_FLAG True if SBGAIN agged T**AU_FLAG** True if
TAU flagged
TCAL_FLAG True if TCAL flagged
TRX_FLAG True if TRX flagged
TSKY_FLAG True if TSKY flagged
TSYS_FLAG True if TSYS flagged
TANT_FLAG True if TANT flagged
TANT_TSYS_FLAG True if TANTTSYS flagged
PWVPATH_FLAG True if PWVPATH flagged

Total power data monitoring.

TOTALPOWERMONITORING Total power data monitoring.				
Name	Format	Units	Measure	Comments
Columns				
Keywords				
AEDF_VERSION	Float	s		AEDF format version
Key				
TIME *(TIME EXTRA_PREC)* CONFIG_DESCRIPTION _ID FIELD_ID	Double Double Int Int	s s	EPOCH	Integration mid-point extra TIME precision Cong-description identifier Field identifier
Non-key attributes				
INTERVAL EXECUTE_ID *(SCAN_NUMBER)* *(OBS_NUMBER)* *(INTEG_NUMBER)* STATE_ID *(UVW)* EXPOSURE TIME_CENTROID	Double Int Int Int Int Int(N_a) Double(3,N_a) Double Double	s m s s	 EPOCH	Time interval Execute identifier Scan number Observation number Integration number State identifier UVW coordinates Effective integration time Time centroid
Data				
FLOAT_DATA	Float(N_c,N_f)			Float data matrix (single dish)
Flag information				
FLAG_ANT FLAG_POL FLAG_ROW	Int(N_a) Int(N_a, N_c) Bool			Flags concerning antennas Flags concerning antennas and polarization The row flag

Notes: This table is dedicated to the raw data originating from the radiometers to monitor the amount of precipitable water vapor in the pointing direction of each antenna and from the astronomical receivers to monitor e.g., their stability. This table is very similar to the MAIN table except that the data are directly present. These data may not be synchronized with the correlated astronomical data.

AEDF_VERSION EDF revision number, expressed as *major revision.minor revision*.

TIME Mid-point (not centroid) of data interval.

TIME_EXTRA_PREC Extra time precision

CONFIG_DESCRIPTION_ID Configuration descriptor identifier >(0) providing a direct index into the CONFIG_DESCRIPTION sub-table row number. Note that two or more sub-arrays cannot refer to the same CONFIG_DESCRIPTION_ID identifier. Note also that in this CONFIG DESCRIPTION table PROCESSOR_ID must be indexed for a processor of type RADIOMETER and that CORRELATION_MODE must have a value N_{cm}=1.

FIELD_ID Field identifier ($\geq$0)

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

EXECUTE_ID This provides access to a row number in the EXECUTE_SUMMARY table. This meta coordinate defines the data-base in term of an implicit collection of data blocks.

SCAN_NUMBER SCAN (ref. ALMA glossary) number. Enumeration relative to the Executing Block. This item is not mandatory. It should be used if the data are synchronized with the data from the correlator.

OBS_NUMBER OBSERVATION (ref. ALMA glossary) number. This item is not mandatory. It should be used if the data are synchronized with the data from the correlator.

INTEG_NUMBER INTEGRATION (ref. ALMA glossary) number.This item is not mandatory. It should be used if the data are synchronized with the data from the correlator.

STATE_ID State identifier (≥ 0).

UVW Antenna-based uvw coordinates for the N_a antennas. This is defined at the centroid time TIME CENTROID .

EXPOSURE Effective duration of an INTEGRATION (or a SUBINTEGNUM in the STATE table is > 0).

TIME_CENTROID Time stamp reflecting the average time the non-blanked data was integrated.

FLOATDATA Total power data.

FLAG_ANT Unsigned long int (4-bytes) vector. Each bit corresponds to a specified flagging condition (e.g. 2^2 = pointing, 2^6 = shadowing, ...). When different from zero all data concerning the antenna is affected.

FLAG_POL Unsigned long int (4-bytes) vector . When different from zero all data concerning the polarization and antenna is invalid.

Weather station information.

WEATHER Weather station information.				
Name	Format	Units	Measure	Comments
Columns				
Key				
ANTENNA_ID	Int			Antenna identifier
TIME	Double	s	EPOCH	Time interval mid-point
INTERVAL	Double	s		Time interval
Data				
WVMCAL_ID	Int			WVMCALIB Id.
H2O	Float	m^{-2}		Average column density of water
PRESSURE	Float	hPa		Ambient atmospheric pressure
REL_HUMIDITY	Float	m^{-2}		Ambient relative humidity
TEMPERATURE	Float	K		Ambient air temperature for an antenna
(DEW_POINT)	Float	K		Dew point
WIND_DIRECTION	Float	rad		Average wind direction
WIND_SPEED	Float	$m\ s^{-1}$		Average wind speed
WIND_MAX	Float	$m\ s^{-1}$		Wind max speed

Flag information				
(H2O_FLAG)	Bool			Flag for H20
(PRESSURE_FLAG)	Bool			Flag for PRESSURE
(REL_HUMIDITY_FLAG)	Bool			Flag for RELHUMIDITY
(TEMPERATURE_FLAG)	Bool			Flag for TEMPERA-TURE
(DEW_POINT_FLAG)	Bool			Flag for DEWPOINT
(WIND_DIRECTION_FLAG)	Bool			Flag for WINDDIRECTION
(WIND_SPEED_FLAG)	Bool			Flag for WINDSPEED
(WIND_MAX_FLAG)	Bool			Flag for WIND MAX FLAG

Notes: This table contains mean external atmospheric and weather informations.

ANTENNA_ID Antenna identifier, as indexed by ANTENNA*n* form MAIN

TIME Mid point of the time interval over which the data in the row are valid. Require to use the same TIME Measure reference as in MAIN

INTERVAL Time interval

WVMCAL_ID WVMCAL identifier

H2O Average column density of water

PRESSURE Ambient atmospheric pressure

REL_HUMIDITY Ambient relative humidity

TEMPERATURE Ambient temperature

DEW_POINT Dew point temperature

WIND_DIRECTION Average wind direction

WIND_SPEED Average wind speed

WIND_MAX Maximum wind speed

H2O_FLAG Flag for the average amount of precipitable water vapor **IONOS_ELECTRON_FLAG** Flag for the average column density of electrons

PRESSURE_FLAG Flag for pressure

REL_HUMIDITY_FLAG Flag for the relative humidity **TEMPERATURE_FLAG** Flag for the temperature

DEW _POINT_FLAG Flag for the dew point **WIND_DIRECTION_FLAG** Flag for the wind direction

WIND_SPEED_FLAG Flag for the wind speed

Radiometric calibration method.

WVMCAL Radiometric calibration method.				
Name	Format	Units	Measure	Comments
Columns				
Key				
ANTENNA_ID SPECTRAL_WINDOW _ID TIME INTERVAL	Int Int Double Double	 s s	 EPOCH	Antenna identifier Spectral window identifier Mid-point of time for which the set of parameters is accurate Time interval
Data				
OPERATION_MODE NUM_POLY FREQ_ORIGIN PATH_COEFF CALIBRATION_MODE *(WVREF_MODEL)*	String Int Double Double(N_{poly}+1) String Float	 Hz		WVR operation mode Series order N_{poly} Origin for the polynomial Conversion formula coefficients WVR calibration mode Reference model to calibrate the WVM

Notes: This table contains time-variable calibration methods for the radiometric corrections. Note that the frequency limits for polynomial expansion of coefficient must be compatible with the SPECTRAL_WINDOW _ID of the radiometer.

ANTENNA _ID

SPECTRAL WINDOW_ID Spectral window identifier for the radiometer.

TIME Mid-point of the time interval over which the data in the row are valid. Require to use the same TIME Measure reference as in MAIN .

INTERVAL Time interval

OPERATION_MODE Operation mode (TBD) e.g. keep corrected and uncorrected, keep average correction to zero, ...

NUM_POLY Series order of the polynomial expressions

FREQ_ORIGIN Frequency origin for the polynomial expansion.

PATH_COEFF Conversion formula coefficients of WVR to Pathlength

CALIBRATION_MODE Calibration mode (TBD) e.g. a priori coefficients or empirical correction added

WVREF_MODEL Calibration parameter which describes the reference model for the calibration of a radiometer.

Table 2: List of the AEDF tables, their purpose and their status compared to the MSv2 tables.

AEDF table name	Property	Purpose	Status/MSMv2
MAIN	fine temporal	Data, coordinates and flags.	re-designed
ALMA_CORRELATOR_MODE	static	Processor mode information.	draft
ANTENNA	static	Antenna characteristics.	modified
CALDEVICE	quasi-static	Calibration device characteristics.	new
CONFIG_DESCRIPTION	static	Configuration description.	new
DATA_DESCRIPTION	static	Spectro-polarization description.	modified
DOPPLER	static	Doppler tracking information.	unchanged
EXECUTE_SUMMARY	quasi-static	Schedule and Execute summary.	renamed & modified
FEED	quasi-static	Feed characteristics.	modified
FIELD	quasi-static	Field position for each source.	modified
FLAG_CMD	quasi-static	Flag commands.	unchanged
FOCUS	temporal	Focus information.	new
FREQ_OFFSET	fine-temporal	Frequency offset information.	modified
GAIN_TRACKING	temporal	Antenna gain tracking information.	new
HISTORY	low-temporal	History information.	unchanged
OBS_SUMMARY	quasi-static	Observation summary information.	new
OBSERVATION_UNIT	fine-temporal	Observation unit characteristics.	new
POINTING	static	Antenna pointing information.	modified
POLARIZATION	static	Polarization information.	unchanged
PROCESSOR	low-temporal	Processor characteristics.	unchanged
RECEIVER	low-temporal	Receiver properties.	new
SCAN_SUMMARY	static	Scan summary information.	new
SCHEDULE	med-temporal	Schedule characteristics.	new
SEEING	static	Seeing information.	new
SOURCE	static	Source information.	modified
SOURCE_PARAMETER	static	Source parameter information.	new
SPECTRAL_WINDOW	static	Spectral window description.	modified
STATE	static	State information.	modified
SWITCH_CYCLE	med-temporal	Cycle information in switching modes.	new
SYSCAL	fine-temporal	System calibration.	modified
TOTALPOWER MONITORING	fine-temporal	Total power data monitoring.	new
WEATHER	med-temporal	Weather station information.	modified
WVMCAL		Radiometric calibration method.	new

Notes: List of the AEDF tables. The number of rows in these tables depends on the temporal granularity of the information they hold. The tables with a static content should have a small number of rows. The table with a fine temporal granularity means that they hold information at a level at least as fine as the time interval between two successive integrations. The majority of these tables are a legacy of the MSMv2. Some are said new although they were anticipated in the MSv2 document. Most of the MSMv2 tables had to be modified by adding attributes required for ALMA. Some tables are completely new. A few inter-relations between these tables have been modified (for example the Receiver table formerly related to the Spectral-window table is now associated to the Feed table).

Table 3: List, notation and properties for the items defining sizes in the AEDF

Name	AEDF Table	Purpose	Symbol	Vis.	Rest.
NUM_ANTENNA	ANTENNA	Number of antennas N_a in the collection	N_a	#	[1,64]
ATMPHASE_CODE	DATA_DESCRIPTION	Code for atm phase correction	N	#	[0,2]
NUM_BASE_LENGTH	SEEING	Number of baselengths N_{basept} points sampling the function	N_{basept}	-	≥2
NUM_CORRBIN	CONFIG_DESCRIPTION	Number of correlator data bins N_{bin} for each baseband	N_{bin}	#	[1,4]
NUM_BASEBAND	CONFIG_DESCRIPTION	Number of basebands N_{bb} in the col-	N_{bb}	#	[1,4]
CORRELATION_MODE	CONFIG_DESCRIPTION	Correlation mode N_{cm}	N_{cm}	#	[1,2]
NUM_COEFF		Number of coefficients in the analyt-	N_{coeff}	-	≥1
NUM_DEP	SOURCE_PARAMETER	Number N_{dep} of dependencies for bootstraped uxes	N_{dep}	-	≥0
NUM_EPOCH	SOURCE	Number of epochs N_e for provided uxes	N_e	-	≥0
NUM_CHAN	SPECTRAL_WINDOW	Number of spectral channels N_f	N_f	#	≥1
NUM_FREQ	SOURCE_PARAMETER	Number N_{fp} of frequencies to which parameters are provided	N_{fp}	-	≥1
NUM_INTEGRATION	OBS_SUMMARY	Number of integrations N_{integ}	N_{integ}	-	≥0
NUM_LINES	SOURCE	Number of spectral lines N_{line}	N_{line}	-	≥0
NUM_LO	RECEIVER	Number of local oscillators N_{LO}	N_{LO}	-	≥1
NUM_CALLOAD	CALDEVICE	Number N_{cl} of NCAL or TLOAD	N_{load}	#	≥1
NUM_OBS	SCAN_SUMMARY	Number of OBSERVATIONs N_{obs}	N_{obs}	#	≥1
NUM_CORR	POLARIZATION	Number of correlation cross-products N_c	N_c	#	≥1
NUM_POLY_FREQ	SYSCAL	Freq. series order N_{polyf}	N_{polyf}		≥0
NUM_RECEPTORS	FEED	Number of receptors N_r on this feed	N_r	#	≥0
NUM_SCHEDBLOCK	OBSERVATION_UNIT	Number scheduling blocks N_{SB} for this observation unit	N_{SB}	-	≥0
NUM_SCAN	SCHEDULE	Number of different scans N_{scan} in this Scheduling Block	N_{scan}	#	≥0
NUM_SCANINTENT	OBSERVATION_UNIT	Number of selected SCAN intents	N_{scint}	-	≥0
NUM_STEP	SWITCH_CYCLE	Number of steps in the cycle	N_{bin}	#	≥1
NUM_STOKES	SOURCE_PARAMETER	Number of stokes parameters N_{sto}	N_{sto}	#	[1,4]
NUM_SUBBAND	CONFIG_DESCRIPTION	Number of “SUBBANDs” N_{sw} for each baseband	N_{sw}	#	[1,32]
NUM_SUBINTEG	OBS_SUMMARY	Number of sub-integrations $N_{subinteg}$	$N_{subinteg}$	-	≥1

Notes: List of items defining sizes in the AEDF. By convention their names begin by NUM . Two items have been added to those, ATMPHASE_CODE and CORRELATION_MODE. Note that for ATMPHASE_CODE $N_{apc=1}$ if the value for the code is 0 else N_{apcis} equal to the code value. With the restricted range of allowed values in the case of ALMA, for CORRELATION_MODEN_{cm} corresponds to the value of CORRELATION_MODE. Col. 1 and 2: item name as used in the AEDF tables and location of its definition. Col. 3 and 4: item purpose and its symbol. Col. 5: item visibility, private (-) or protected (#) when the item is or is not exclusively used in the AEDF table where it is defined. Col. 6: resctrictions