MeasurementSet definition version 2.0

A.J. Kemball (NRAO) and M.H. Wieringa (ATNF), eds.

January 21, 2000

Contents

1 Summary

This note describes a revised MeasurementSet definition which extends that previously given in AIPS++Note 191 (Wieringa and Cornwell 1996). The new definition is designed to support changes in the synthesis calibration system, provide support for VLBI processing, enhance single-dish capabilities and also incorporate more general modifications that have been suggested since the first definition was released. The motivation for the modifications is discussed, and a complete form of the revised definition is presented.

2 Introduction

The MeasurementSet (MS) defines the format in which visibility and single-dish data are stored in AIPS++(Wieringa and Cornwell 1996). The format has been chosen to accommodate synthesis and single-dish data from a variety of instruments in as broad a framework as possible. In addition, it has been designed to be compatible with the requirements of the measurement equation formalism (Noordam 1996), which has been adopted to model instrumental errors in the AIPS++calibration system.

The original definition will be referred to as version 1.0 in what follows. The reasons for the modifications proposed here are as follows:

- **VLBI data reduction:** Some changes are necessary in v1.0 to accommodate the requirements of VLBI data processing. An extension of the basic synthesis MeasurementSet format is presented to provide a common VLBI data format.
- **Synthesis calibration:** General synthesis development currently in progress requires some modifications to the existing format, particularly in support of cross-calibration. Some changes are also proposed to meet more recent requirements in areas such as mosaicing.
- **Single-dish processing:** Some changes have been proposed to enhance support for single-dish processing and to ensure greater compatibility between single-dish and synthesis data reduction.
- **Accumulated changes:** Several changes of a general nature have been proposed since v1.0, and this is a good time to evaluate and incorporate these as appropriate.

This is an opportune time to revise the MeasurementSet, as an increased level of synthesis development is underway. Later revisions will have a broader impact on existing code.

The design philosophy underlying this MS definition is summarized in terms of the following objectives:

- **Incremental change:** The changes proposed here are designed to be as incremental as possible and no extensive re-design has been attempted. The scientific benefit of each modification has been weighed against the scope of the proposed change to the MS design.
- **Compatibility:** Compatibility between single-dish and synthesis data has been retained within one basic MS definition. The proposed VLBI extensions are constrained to be compatible with both the basic synthesis format and across existing VLBI networks and correlators.

- **Separation of information:** A fundamental distinction is made between a priori information which is known at the time of observation or shortly thereafter, and calibration information subsequently derived in post-processing. The MS definition is primarily designed to encompass a priori information. The format of calibration tables in given elsewhere.

- **Storage:** A future document will define standard Data Managers for each column, which are recommended but not required. Attention has been paid to the physical file sizes implied by the MS definition, particularly for larger datasets, and storage managers for a standard compressed MS format will be discussed in the same document.

- **Combining measurement sets:** The MS definition has been revised to facilitate the combining of diverse observational datasets within one MeasurementSet if required, in a manner that is compatible with the calibration system. This does not hold the implication that all observations to be processed jointly require that the underlying MeasurementSets be combined. Applications will support the ability to process groups of MeasurementSets. However, the facility to combine data needs to be provided in the MeasurementSet design.

  The combining of different observations within one MeasurementSet is permitted subject to the conventions of Section 3.3, which separates data primarily by OBSERVATION_ID, but also by PROCESSOR_ID and ARRAY_ID. Cross-calibration between observations is however subject to the capabilities of the calibration system. Specialized inter-conversion of calibration information may be necessary if the calibration in separate observations is sufficiently disjoint. For example, transferring calibration between frequencies sufficiently far apart will likely be performed in an intermediate step, using external utilities.

  An application is envisaged to combine data from one or more separate MeasurementSets, creating a new output MS by copying the input data row by row, and renumbering the indices and interleaving data as required. No sort order is prescribed for a MeasurementSet in general, but it is expected that data sorted in time order will be most useful to the broadest range of applications.

Specific principles adopted in the design are given below:

- **Signal path:** The MeasurementSet provides a format to represent data from a generic radio-telescope or interferometer. Along with the basic observed data, for which a limited set of accepted types are specified in the MAIN table section, there are associated data characterizing the state of the instrument as a whole. These include: i) abstracted antenna properties of components in a generic telescope, such as feeds and spectral windows, which serve also to label the output data; ii) external information, such as flagging, history or weather data; and iii) instrument-specific back-end data which may be difficult to represent in a completely generic form. The final category includes intermediate data and state information from devices such as correlators, radiometers, spectrometers or pulsar timers, amongst others, where they do not overlap with abstracted antenna properties defined elsewhere. This state information may be used in computing or initializing calibration corrections. Thus the MS represents the signal path and state of the instrument, using as generic an interface as possible, but allowing specialization where appropriate. This conforms closely with the overall calibration model.

- **Use of Measures:** Some columns in the MeasurementSet require coordinate and unit specification. This is done in a manner compatible with the AIPS++ Measures system. Measure frame information is implicit in the underlying MS data. Row-based measures are avoided wherever possible due to the overhead this would often impose on the data reduction system through frequent coordinate conversion to a common reference frame. Thus, column-based measures are the default unless otherwise noted. The only place in which a row-based Measure reference is currently allowed is the frequency axis in the SPECTRAL_WINDOW sub-table, where it is supplied in order to allow an efficient representation of Doppler tracking. Column-based Measures of a specific type (e.g. EPOCH), should have a common reference across the MS as a whole, unless there is a compelling reason otherwise. This requirement is necessary for efficiency, and in minimizing transformations when combining diverse MS datasets. No standard reference is enforced for any given Measure type; these should be chosen prudently. Units are also required to be column-based, unless a row-based Measure is allowed for a given column. Recommended units are specified for each Measure column. Access to the MS is assumed to take place through the MS access classes. TableMeasures will be used wherever practical in the MS access classes.

- **Relative indexing:** All indices for antennas, feeds, spectral windows, or related quantities are assumed to start at zero. Thus for direct indices into sub-tables, such as SPECTRAL_WINDOW_ID, a value ID=n maps to row=n. MS indexing is zero-based in C++, and one-based in Glish, as before.

- **Sub-table completeness:** Not all data represented in associated sub-tables are assumed to be present in MAIN, although this is encouraged. For example, extra antennas may appear in the ANTENNA sub-table, even if they have no associated data in MAIN.

- **Blanking and defaults:** Measured data are blanked by setting associated Boolean flag data, and magic value blanking is not used. For integer values constrained to be non-negative, or sub-table indices, however, a value of -1 will generally denote an unset value. All required columns should be filled with suitable defaults if not actually used.

- **Times and intervals:** For quantities which may vary with time, an INTERVAL of zero implies a constant value with no time-dependence, while a negative INTERVAL implies that the value is valid until re-defined. The latter case accommodates values that are time-stamped but have an undefined or unknown period of validity.

- **Non-standard columns:** The naming convention for non-standard columns remains the same as in MS definition v1.0, namely: i) general, non-supported columns start with the prefix NS_$SITE_, as in NS_NRAO_WHATEVER; and ii) columns supported by a consortium site start with the prefix $SITE_, as in NRAO_WHATEVER. Standard columns, such as those described in this definition, are not permitted to use either of the above two prefix forms. Non-standard columns may not be defined to store data already stored in standard columns. In addition, applications are not required to support non-standard columns.

- **Archiving:** MeasurementSet data are assumed to be archived in an external format such as FITS or HDF (Hierarchical Data Format) for example, in as lossless a format as possible. This format will be specified separately. Changes to the MS format will not be made without a compelling scientific reason, but the format will, of necessity, evolve over time. The archiving application has the responsibility to restore the data to the latest format. It is deemed too complex for the underlying MS access code, and data reduction code, to be made capable of recognizing multiple MS revisions at the C++ level. A utility is envisaged, however, to convert MS v1.0 data to the MS v2.0 format.

Suggestions regarding this revision of the MS definition have been contributed broadly from within the AIPS++project. No specific attribution is given for each change but the accompanying text reflects the rationale behind the modification and includes relevant points that were raised in public discussion.

3 Summary of changes

This section contains a description of the changes proposed for each table in the MS definition. A full definition of the v2.0 MS format is given in Section 5.

3.1 MAIN table

The fundamental assumption underlying the MAIN table is that it contains interferometer or single-dish data, represented in cross-power or autocorrelation form. The composite key is baseline-based and auxiliary antenna-based data, that are consequently not fully functionally dependent on the baseline-based key, are stored in some of the associated sub-tables. As the latter data are often time-variable, the design admits multi-key lookups in these tables, as is already assumed in v1.0 (eg. SYSCAL table). In keeping with realistic databases, the design of the MAIN table is not fully normalized. The term is used here in the conventional database sense (see for example, Date 1986). Adding indices to MAIN to avoid sub-table lookups involves a trade-off between the anticipated increase in the physical file size (which is storage manager dependent), and the expected size of the sub-table and frequency with which the auxiliary data need to be accessed in time-critical applications. It also depends on whether the index is commonly used in data selection. Note also that auxiliary data may be sampled on different intervals, which is an important consideration in deciding which antenna-based information should be placed in the same sub-table. These factors have been balanced subject to the following guidelines: i) storage manager independence has been retained (i.e. the worst case is assumed regarding extra MAIN indices; ii) single-key lookups in sub-tables are strongly favoured; and iii) it is assumed that small sub-tables will be held in cache during MS access.

The changes proposed for the MAIN table are presented below:

3.1.1 MAIN keywords

It is proposed that the DEFAULT_CALIBRATION keyword be removed, as it is no longer used.

The following additional keywords are proposed:

MS_VERSION	Float	(MS format revision)	(Reqd.)
SORT_COLUMNS	String	(Indices over which MS is sorted)	(Opt.)
SORT_ORDER	String	('ASCENDING' or 'DESCENDING')	(Opt.)

Keywords SORT_COLUMNS and SORT_ORDER are used to provide information on the sort-order of the underlying MS.

3.1.2 Non-standard MAIN data

Three non-standard data types have been suggested for accommodation within the standard MS definition, namely: (i) lag data; (ii) triple-product data, and (iii) FLOAT_DATA with an arbitrary axis other than frequency.

The need to store lag data is motivated by the output format of XF correlators and the requirements for data loading and inspection in this case. Storage of triple-product data is required for optical interferometry processing but is also useful for the implementation of incoherent VLBI fringe-fitting algorithms. The requirement for non-standard FLOAT_DATA axes is concerned with more flexible single-dish data formats.

Lag data can be incorporated by allowing an optional MAIN column:

LAG_DATA	Complex(NUM_CORR,NUM_LAG)	(Lag function)	(Opt.)

A LAG_ID index (via DATA_DESC_ID) points to a LAG table containing the details of the lag spectrum, including range. Lag data are not expected to be calibrated directly and would be supported primarily as an interim format in an anticipated data reformatting sequence. The routine storing of both lag and transformed spectral data in one MS would often lead to prohibitive disk space requirements, and is not anticipated here. Further information regarding the joint storing of lag functions and frequency spectra in the same MS can be found in the MAIN table description in Section 5.

Triple-product data can be incorporated in the MS definition by adding additional antenna labels such as:

ANTENNA3	Int	(Third antenna)	(Opt.)
FEED3	Int	(Feed on ANTENNA3)	(Opt.)

The UVW column is similarly affected. The triple-product data would be expected to be formed for matching time intervals, so other columns such as SOURCE_ID would be held to be common. The triple-product data would be stored in the DATA column and the presence of a valid ANTENNA3 column would label the data to be of this type. It is stressed that triple-product and lag data will not have nearly equal standing in the calibration system, as compared to conventional frequency spectra.

Support for an arbitrary axis in the FLOAT_DATA column, by generalizing the SPECTRAL_WINDOW_ID index, is more difficult to accommodate, and is best approached in the same manner as the lag data, by enumerating special cases with separate indices such as LAG_ID. The lag and triple-product data formats are still fundamentally visibility data and may also share at least some of the broader synthesis infrastructure. Their impact on the sub-table formats is also limited. This is not true for arbitrary data types in the MS such as may result if the axis type in the FLOAT_DATA column was allowed to vary arbitrarily.

Accommodating non-standard columns (such as LAG_DATA) in this way is contingent on the requirement that they do not permit the representation of identical data in multiple equivalent formats. This would represent a significant application overhead and would affect the global structure of the MS as a whole. Most importantly, this would limit joint processing of data from separate instruments (eg. single-dish and interferometry data).

3.1.3 DATA labeling

At present, polarization labeling for each correlator is stored in the SPECTRAL_WINDOW table, although such information is not properly associated with frequency. Consequently, it is proposed that polarization information be stored in a separate POLARIZATION sub-table, indexed by POLARIZATION_ID.

However, rather than add POLARIZATION_ID to MAIN as an additional index, it is proposed that SPECTRAL_WINDOW_ID be replaced by DATA_DESC_ID, which points to a DATA_DESCRIPTION sub-table containing SPECTRAL_WINDOW_ID and POLARIZATION_ID pairs. This structure has the specific advantage that the DATA array shape is completely determined by DATA_DESC_ID, which is an important consideration in tiling and uv-data access. The DATA_DESC sub-table will be small, so that secondary look-ups to determine SPECTRAL_WINDOW_ID from DATA_DESC_ID can take place in memory and will not represent a significant overhead.

Thus, SPECTRAL_WINDOW_ID in MAIN will be replaced by DATA_DESC_ID, as follows:

DATA_DESC_ID Int (Data description index) (Reqd.)

3.1.4 Processor information

As discussed in Section 2, provision needs to be made for the characterization of the state of the specific back-end device used to produce the final data. Such devices include correlators, radiometers, spectrometers and pulsar timers, amongst others. These device data are not permitted to duplicate general antenna properties represented elsewhere, but are invariably highly instrument-specific. It is proposed that the existing CORRELATOR_ID index be replaced and generalized as:

PROCESSOR_ID Int (Processor identifier) (Reqd.)

This acts as an index into a PROCESSOR sub-table, which similarly replaces the CORRELATOR sub-table, as described in Section 4.0.4. This scheme accommodates parameter and status information associated with each processor, generally by look-up into secondary sub-tables containing this information using a subset of the MAIN key. For single-dish data, however, the processor switching phase is sufficiently fundamental to warrant inclusion as an optional MAIN column, both for access efficiency and data labeling:

PHASE_ID Int (Switching phase identifier) (Opt.)

3.1.5 State information

An associated STATE table is proposed to store combined information relating to active reference signals or loads, as used in phase-switching, sub-scan information and a characterization of the current observing mode. This information is not fully represented in the MS at present, and is required for a more complete integration of single-dish processing in a common framework. An index in MAIN would be required as follows:

STATE_ID Int (State identifier) (Reqd.)

3.1.6 Pulsar information

Synthesis correlation of pulsar data may involve the use of a pulsar gate, which is a Boolean matrix in frequency and pulsar phase. The use of the pulsar gate is either selected for a given baseline or not. The gate may change with frequency group or source. No baseline dependence seems likely but the gate could change with time. A general solution is to implement a new MAIN table column as follows:

PULSAR_GATE_ID	Int	(Pulsar gate identifier)	(Opt.)

This is an index into the PULSAR_GATE table containing the gate mask. It is also proposed that the PULSAR_ID column be absorbed in the SOURCE table (via SOURCE_ID), and removed from MAIN. In this revision of the MS definition, the PULSAR_BIN column is made optional.

3.1.7 Baseline reference antenna

For VLBI data from baseline-based correlators, record needs to be kept of which antenna was used as the reference on each baseline during correlation. The antenna sequence may be switched by the data filler to preserve an ascending baseline numbering scheme. The reference antenna needs to be known when shifting spectra in frequency to a common velocity frame (eg. LSR). An optional column is proposed for this purpose:

BASELINE_REF	Bool	(True for ANTENNA1)	(Opt.)

3.1.8 Flag data

The MS definition has been revised to recognize two distinct forms of flagging, both of which need to be supported, namely: i) flagging most easily expressed as global commands (eg. (ANTENNA1=3) AND (TIME > 12h03 UT)); and ii) highly time-variable flagging of a more random nature, such as that resulting from mean-level clipping. In addition, the flagging system recognizes two further requirements, namely that point-by-point flags stored with the MS MAIN data are more efficient than global flag commands when data are accessed, and secondly that flags should be arbitrarily reversible in both representations.

To this end, the multi-level Boolean column FLAG_HISTORY in MAIN is retained, but renamed to FLAG_CATEGORY. It contains a record of point-by-point flags associated with each MS row in separate categories. The cumulative effect is stored in column FLAG, for all categories combined. Categories may be arbitrarily defined by editing applications, but several standard categories are reserved. This includes a special category, FLAG_CMD, which contains the cumulative effect of flag (or un-flag) commands stored in the FLAG_CMD table. These commands are stored in the FLAG_CMD table with associated identifiers, including flag reason and level. The flag reason indicates the type of flagging and the flag level the revision of flagging to which the command belongs. Flag revisions are stored in the point-by-point flags in FLAG_CATEGORY by creating new, related categories to store separate revisions of the same category if required.

Thus point-by-point or global flags can be reversed individually or jointly. Both representations are scientifically necessary, and supported. If the global flags are updated, it is the responsibility of the flagging application to carry out the required one-time update of the point-by-point flags in MAIN.

3.1.9 Auto-correlation and single-dish data

Single-dish data are represented by (ANTENNA1=n, ANTENNA2= n), and autocorrelation data by the secondary condition (FEED1=m, FEED2=m). MeasurementSets are permitted containing a mixture of FLOAT_DATA and DATA columns. For autocorrelation data, however, in any given row the FLOAT_DATA column will be checked first, and, if absent, the DATA column will be used.

3.1.10 Hypercube indices

A standardized naming convention for hypercube indices is proposed as <column>_HYPERCUBE_ID (eg. DATA_HYPERCUBE_ID), where they are specifically required by Tiled Storage Managers, as implemented in the Table system.

3.1.11 Video point

An optional data column is proposed in MAIN to store the video point. This is preserved on some systems, and is required for a complete transform between the frequency and lag domains.

VIDEO_POINT	Complex(NUM_CORR)	(Video point)	(Opt.)

This representation is more transparent than encoding the information in the DATA array, which leads to display and data selection inconsistencies.

3.1.12 Time centroid

The TIME column in MAIN is defined as the mid-point of the nominal sampling interval, as specified in the column, INTERVAL. The EXPOSURE column, in contrast to INTERVAL, defines the effective integration time, including the effects of missing data. The centroid of the effective data interval is not represented in the current MS definition, but does contain distinct information. It is added here as a new non-key attribute:

TIME_CENTROID	Double	(Centroid of exposure)	(Reqd.)

3.2 ANTENNA table

The changes proposed for the ANTENNA table are presented below:

3.2.1 ARRAY identification

It is proposed that the antennas retain the same global numbering scheme across the MS as a whole, and not be allowed to vary independently for each ARRAY_ID. Thus ARRAY_ID is no longer a secondary antenna index, but does retain its primary purpose to separate antennas in the MAIN table for calibration purposes. There are advantages to enforcing a global antenna numbering scheme, particularly when antennas are shared between subarrays and may thus share certain calibration information.

3.2.2 Antenna type

A new column is proposed to distinguish between tracking stations and standard antennas (in which orbiting antennas are included):

TYPE String (Enumerated antenna type (eg. TRACKING-STN)) (Reqd.)

3.2.3 Orbital elements

Mean Keplerian orbital elements need to be stored for orbiting antennas, in the following proposed format:

MEAN_ORBIT Double(6) (Mean Keplerian elements) (Opt.)

3.2.4 Phased array identifier

It is proposed that the PHASED_ARRAY_ID column be made optional.

3.2.5 Row flag

It is proposed that a flag column be added to validate or invalidate the contents of the row, as follows:

FLAG_ROW Bool (Row flag) (Reqd.)

3.3 ARRAY table

It is proposed that multiple datasets be combined within one MS through the use of the following indices in the MAIN table:

- **OBSERVATION_ID:** This separates distinct observations taken with different instruments (eg. MERLIN, VLBA, VLA), or observations with the same instrument at different epochs.

- **PROCESSOR_ID:** Information obtained via this index separates data from separate processors, such as separate correlators, as may occur for combined VLBI datasets (eg. MK4, VLBA).
- **ARRAY_ID:** This index indentifies subarrays, such as those defined by separate groups of antennas observing independently in source or frequency, which cannot necessarily be calibrated together. Subarrays can be transient and highly variable, as in geodetic VLBI observations, or more directly associated with the clear sub-division of an interferometer array (eg. Fixed Q-band and C-band subarrays during a VLA observing run). Subarrays may also be assigned by users in an arbitrary manner to label separate parts of an array. This may bear no direct relation to the observing schedule. This interpretation is supported here with the proviso that antennas which cannot be calibrated together are not assigned to the same subarray. The index ARRAY_ID is not numbered relative to OBSERVATION_ID.

In this context the ARRAY table can be absorbed in the OBSERVATION table and it is proposed here that it be removed from the MS definition. A new, optional SCAN_SUMMARY table is proposed to store overall array, observation and scan summary information.

3.4 FEED table

It is proposed that the ARRAY_ID column be removed from the FEED table, for reasons discussed in Section 3.2.1. The format of the BEAM sub-table (indexed on BEAM_ID) is deferred for later specification. In addition, it is proposed that the PHASED_FEED_ID be made optional.

A new column is proposed to store the focus length, as:

FOCUS_LENGTH	Double	(Focus length)	(Opt.)

3.5 FIELD table

Changes applicable to the FIELD table are discussed in this section.

3.5.1 Direction information

With the definition of a POINTING sub-table (see Section 4.0.1), the POINTING_DIR and POINTING_DIR_RATE columns can be removed from the FIELD table.

It is also proposed that the DELAY_DIR_RATE, PHASE_DIR_RATE and REFERENCE_DIR_RATE columns be replaced by extending the DELAY_DIR, PHASE_DIR and REFERENCE_DIR columns to dimension (2, NUM_POLY+1) to represent higher order terms. This allows the representation of more specialized correlation modes. The polynomial order is added as a new column:

NUM_POLY	Int	(Polynomial order)	(Reqd.)

3.5.2 Ephemeris identification

A separate identifier is proposed to identify the ephemeris data associated with time-variable FIELD position data, as obtained for moving sources. These data are stored in a separate EPHEMERIS table. This has the added advantage of identifying multiple revisions of the ephemeris, which may vary between observation and final reduction. The proposed new column is:

EPHEMERIS_ID	Int	(Ephemeris identifier)	(Opt.)

3.5.3 Indexing

It is further proposed that the FIELD_ID column be removed and that the table be directly indexed using the FIELD_ID value in MAIN. A row flag is proposed to invalidate the entry, as follows:

FLAG_ROW	Bool	(Row flag)	(Reqd.)

3.6 FLAG_CMD

As discussed in Section 3.1.8, this table is intended to hold all global flag commands applied to the MS. Their cumulative effect is reflected in the MAIN column FLAG_CATEGORY, in the category FLAG_CMD, and therefore further in FLAG and FLAG_ROW. In this proposal, the FLAG_CMD records are represented as character strings, using the MS column names in a limited syntax to represent flagging operations, each with associated labeling information:

TIME	Double	(Midpoint of interval)	(Reqd.)
INTERVAL	Double	(Time interval)	(Reqd.)
TYPE	String	(FLAG or UNFLAG)	(Reqd.)
REASON	String	(Reason)	(Reqd.)
LEVEL	Int	(Flag level)	(Reqd.)
SEVERITY	Int	(Flag severity)	(Reqd.)
APPLIED	Bool	(Flag reflected in MAIN ?)	(Reqd.)
COMMAND	String	(Flagging command)	(Reqd.)

3.7 OBSERVATION table

It is proposed that the CORR_SCHEDULE field be removed (see Section 4.0.4), which is accessed via the PROCESSOR sub-table. The telescope name, which was previously stored in the ARRAY sub-table, is proposed to be recorded as:

TELESCOPE_NAME	String	(Telescope name (eg WSRT))	(Reqd.)

The observation date is required as:

TIME_RANGE Double(2) (Start and stop times of observation) (Reqd.)

A new column is proposed to identify the type of the unmodified schedule summary is SCHEDULE, namely:

SCHEDULE_TYPE String (Schedule type (eg. VEX)) (Reqd.)

A field is proposed to store the log file recorded during observations, as:

LOG String(*) (Observing log) (Opt.)

A field is proposed to store the target release date for the project:

RELEASE_DATE Double (Release date) (Opt.)

A row flag is proposed to invalidate the entry as:

FLAG_ROW Bool (Row flag) (Reqd.)

3.8 OBS_LOG table

It is proposed that this table be renamed as HISTORY, and that its primary function be to act as a processing log. New columns are proposed as follows:

PRIORITY	String	(Message priority (eg. DEBUGGING, WARN, NORMAL or SEVERE)	(Reqd.)
ORIGIN	String	(Source code location from which message originates)	(Reqd.)
OBJECT_ID	String	(Originating ObjectID, if available)	(Reqd.)
APPLICATION	String	(Application name)	(Reqd.)

User notes will be identified by a specific ORIGIN string.

This is also a useful table to store a record of the Glish commands and parameters used to invoke the application. This allows parts of the data reduction to be repeated with minimal effort. An automatic recording mechanism needs to be invoked by individual methods or applications for this purpose. This information can be represented in a string format as:

CLI_COMMAND	String(*)	(Glish parameter or command sequence)	(Reqd.)
APP_PARAMS	String(*)	(Application parameters)	(Reqd.)

It is recognized that a mechanism to log all ad hoc CLI commands affecting a MeasurementSet is impractical, but this field should be filled by all standard applications.

3.9 SOURCE table

The changes proposed for the SOURCE table are given below:

3.9.1 Source model

Information on the detailed source model needs to be specified at several points in calibration and imaging, and is properly selected at that time. A field is added here to record the assignment of a component source model for each SOURCE_ID, as this is made during calibration. This assigned source model will never be used in an application without explicit notification or confirmation, and is not intended as a default source model.

SOURCE_MODEL	TableRecord	(Assigned CSM)	(Opt.)

3.9.2 Spectral line information

Information on the systemic velocity and rest frequency of spectral lines observed towards individual sources is required in various stages of line calibration, particularly for Doppler tracking of individual transitions in software. The indices SOURCE_ID and SPECTRAL_WINDOW_ID are a good composite key for this information and it is proposed that they be added to the SOURCE table. A general extension for multiple spectral lines per spectral window is proposed as follows:

NUM_LINES	Int	(No. of transitions for this spectral window)	(Reqd.)
TRANSITION	String(NUM_LINES)	(Transition name)	(Reqd.)
REST_FREQUENCY	Double(NUM_LINES)	(Rest frequency for this transition)	(Reqd.)
SYSVEL	Double(NUM_LINES)	(Systemic velocity)	(Reqd.)

3.9.3 Pulsar information

The PULSAR_ID is moved from the MAIN table to the SOURCE table, as proposed in Section 3.1.6. This points to the PULSAR table, containing pulsar source information.

3.10 SPECTRAL_WINDOW table

As discussed in Section 3.2.1, it is proposed that the ARRAY_ID key be removed. In addition, the polarization labeling data CORR_TYPE, CORR_PRODUCT and NUM_COR are moved to the POLARIZATION sub-table, as discussed in Section 3.1.3.

3.10.1 Name

A spectral window name, is proposed as:

NAME	String	(Spectral window name)	(Reqd.)

3.10.2 Frequency groups

It is important to identify which SPECTRAL_WINDOW_ID's are associated for calibration purposes as part of a broader frequency group. This is necessary in certain calibration applications, including multi-band VLBI fringe-fitting. Two new columns are added for this purpose:

FREQ_GROUP	Int	(Frequency group)	(Reqd.)
FREQ_GROUP_NAME	String	(Frequency group name)	(Reqd.)

3.10.3 Doppler tracking

The REST_FREQUENCY column is replaced by a DOPPLER_ID field reflecting the Doppler tracking parameters used for the spectral window. These data are stored in a separate DOPPLER table which references the line parameters defined in Section 3.9.2.

DOPPLER_ID	Int	(Doppler identifier)	(Opt.)

3.10.4 Sideband and baseband-converter information

Two new spectral window attributes are added to record baseband converter and sideband information:

NET_SIDEBAND	Int	(Upper or lower)	(Reqd.)
BBC_NO	Int	(Baseband converter no.)	(Opt.)
BBC_SIDEBAND	Int	(BBC sideband)	(Opt.)

3.10.5 Receiver information

An additional index is proposed to reference receiver information:

RECEIVER_ID	Int	(Receiver identifier)	(Opt.)

This points to an optional RECEIVER sub-table.

3.10.6 Associated spectral windows

Individual spectral windows may hold a unique relationship to others, such as that between channel-zero data and the associated, unaveraged spectrum. Other may have a subset or equivalence relationship, and it is very useful to record this information. Additional columns for this purpose are proposed as follows:

ASSOC_SPW_ID	Int(*)	(Associated spw_id.)	(Opt.)
ASSOC_NATURE	String(*)	(Nature of association)	(Opt.)

3.10.7 Frequency information

The nominal channel spacing, actual spectral resolution and effective noise bandwidth are distinct, and it is proposed that they be decoupled by adding new columns:

CHAN_WIDTH	Double(*)	(Nominal channel width)	(Reqd.)
EFFECTIVE_BW	Double(*)	(Effective bandwidth)	(Reqd.)

The RESOLUTION column then denotes the actual spectral resolution. A new column is proposed to allow a row-based Frequency Meaure reference, as:

MEAS_FREQ_REF	Int	(Frequency ref.)	(Reqd.)

3.10.8 Row flag

It is proposed that a flag column be added to validate or invalidate the contents of the row, as follows:

FLAG_ROW	Bool	(Row flag)	(Reqd.)

3.11 SYSCAL table

As discussed in Section 3.2.1 it is proposed that the ARRAY_ID key be removed. Further specific changes to the SYSCAL table are listed below:

3.11.1 Number of receptors

This information is redundant as it is already specified in the FEED sub-table, and it is proposed that NUM_RECEPTORS be removed.

3.11.2 Measured temperatures

Optional antenna temperature fields are proposed as follows, where N_r=NUM_RECEPTORS and N_f=NUM_CHAN:

TANT	Float(N_r)	(Antenna temperature T_{ant})	(Opt.)
TSKY	Float(N_r)	(Sky temperature)	(Opt.)
TANT_SPECTRUM	Float(N_r,N_f)	(Antenna temperature spectrum)	(Opt.)
TSKY_SPECTRUM	Float(N_r,N_f)	(Sky temp. spectrum)	(Opt.)
TANT_TSYS	Float(N_r)	($\frac{T_{ant}}{T_{sys}}$)	(Opt.)
TANT_TSYS_SPECTRUM	Float(N_r,N_f)	($\frac{T_{ant}}{T_{sys}}$ spectrum)	(Opt.)
TANT_FLAG	Bool	(Flag for TANT)	(Opt.)
TSKY_FLAG	Bool	(Flag for TSKY)	(Opt.)
TANT_TSYS_FLAG	Bool	(Flag for TANT_TSYS)	(Opt.)

Calibration measurements in bands other than those for which there are recorded data, are accomodated by creating additional spectral window identifiers in the SPECTRAL_WINDOW table, and recording the data in the same format in the SYSCAL table. This allows simultaneous wide- and narrow-band calibration data to be recorded in a uniform manner.

Existing calibration quantities in the SYCAL table, including PHASE_DIFF, TCAL, TRX and TSYS, and their associated flags, are made optional, on an equal footing with the other entries.

3.11.3 Antenna pointing

There is a key requirement to know where the antenna and feed are pointed at the time of the calibration measurement. In synthesis, these data may be taken while interferometer data are not being recorded for example, but are nonetheless useful in a priori amplitude calibration. This information is obtained by lookup in the POINTING sub-table (see Section 4.0.1).

3.12 WEATHER table

As in previous sections it is proposed that the ARRAY_ID qualifier for ANTENNA_ID be removed (see Section 3.2.1). Time- and direction-dependent ionospheric electron column density data (such as that derived from GPS receiver data) is proposed to be stored in a separate TEC sub-table. The existing data columns are made optional, and associated flags are added:

IONOS_ELECTRON_FLAG	Bool	(Flag for IONOS_ELECTRON)	(Opt.)
H2O_FLAG	Bool	(Flag for H2O)	(Opt.)
IONOS_ELECTRON_FLAG	Bool	(Flag for IONOS_ELECTRON)	(Opt.)
PRESSURE_FLAG	Bool	(Flag for PRESSURE)	(Opt.)
REL_HUMIDITY_FLAG	Bool	(Flag for REL_HUMIDITY)	(Opt.)
TEMPERATURE_FLAG	Bool	(Flag for TEMPERATURE)	(Opt.)
WIND_DIRECTION_FLAG	Bool	(Flag for WIND_DIRECTION)	(Opt.)
WIND_SPEED_FLAG	Bool	(Flag for WIND_SPEED)	(Opt.)

A new column set is proposed to record the dew point:

DEW_POINT	Float	(Dew point temperature)	(Opt.)
DEW_POINT_FLAG	Bool	(Flag for DEW_POINT)	(Opt.)

4 New sub-tables

New sub-tables added to MS v2.0 are included in this Section.

4.0.1 Antenna-based pointing (POINTING)

The current MS definition associates a pointing direction with the FIELD_ID. In general, however, the pointing direction is antenna-based and may be time-variable and distinct from the correlated field position. This is best modeled by introducing a POINTING sub-table indexed from the MAIN table by the key (TIME, ANTENNA), and containing absolute pointing information.

The format for the POINTING table is given later but includes support for tracking arcs on the sky, and storing a priori pointing offset information. The POINTING table indicates the direction the optical axis of the telescope is pointing; feed offsets are stored in the FEED table.

4.0.2 State information (STATE)

The STATE table is used to store information concerning reference signals or loads, sub-scan information and a description of the current observing mode, as these fields constitute a set of related information. Sub-scans are used here to denote component sequences of standard observing patterns. Observing modes are envisaged to be represented by standard reserved keywords or phrases. This provides a means of linking the observing strategy and the data; this is a central requirement in guiding automated data reduction procedures, and also provides greater user information.

4.0.3 Digital samplers (SAMPLER)

Data streams which have in common an antenna, feed and receiver/LO chain may be independently associated with digital samplers with time-variable quantization schemes (eg. 1- versus 2-bit sampling) and sampler voltage levels. For calibration purposes this does not constitute a separate feed but does require separate digital correction. This is certainly important in VLBI. A separate SAMPLER sub-table is proposed to address this, indexed via MAIN table columns (TIME, ANTENNA, FEED, SPECTRAL_WINDOW_ID).

4.0.4 Processor information (PROCESSOR)

The back-end processor is identified through the PROCESSOR sub-table, which contains information concerning the processor type (see Section 3.1.4), and indices for time-dependent and time-independent mode information. A pass number is used to distinguish multiple passes for back-end processors for which this is possible (e.g. VLBI correlators). Each back-end or processor type includes a family of associated tables. These are formatted using common column names where appropriate, and accessed using classes that encapsulate the different information unique to each instrument-specific data behind a common interface. This family of tables will be specified in the future for individual instruments and processor types. It is noted, however, that the optional PHASE_ID index in MAIN indexes directly into processor-specific phase information included in this family of tables. An illustration for generic interferomter correlators is given below:

The correlation parameters associated with an individual uv-record are particularly important in VLBI calibration but have more general application in broader synthesis development. This correlator information has four separate components, which are indexed from the PROCESSOR sub-table (which has primary key PROCESSOR_ID):

- **Time-invariant correlation parameters**

  This includes software revision numbers and geometrical constants adopted by the correlator model. We use a TYPE_ID index to point to this information, which is stored in a separate *subtype*_CORRELATOR table. The correlator sub-type (e.g. MK4, JIVE) defines the prefix of the correlator model, geometry and mode tables discussed below.

- **Time-dependent correlator parameters**

  For synthesis data, these include correlator model values and model polynomials as a function of antenna or baseline, time, spectral window, polarization and correlator identifier. Note that there is no one-to-one mapping between a given uv-record and a correlator model record, as a model update may occur within an accumulation interval, and an indexed lookup is always required to locate the associated correlator model record. The time-dependent data also include geometric information common to the instrument as a whole such as polar motion, time information and the solar system ephemerides. These data are indexed on time and correlator identifier. We propose correlator model and correlator geometry tables to hold these two types of time-dependent data respectively. Each table has a composite key including TYPE_ID and no additional MAIN table columns are required.

  Synthesis correlators differ significantly in their architecture and reference frames. For example, the VLBI MK3 correlator is a baseline-based, lag correlator (XF), while the VLBA correlator is an antenna-based FX correlator. Both also use different celestial and interferometer reference frames. This poses difficulties for the representation of correlator mode, geometry and model information in a common format. Translation between different frames may not be exact or reversible in this case, and reference to the original correlator model data is sometimes required. The primary purpose of correlator record-keeping is to provide parameters for visibility corrections required by correlator digital signal processing effects, and to provide total model information for geodetic or astrometric applications.

To accommodate these correlator differences, and other significant differences between synthesis and single-dish correlators, it is proposed that correlator sub-type (eg. MK4, GBT, VLBA; as given in the PROCESSOR sub-table) act as a global key for correlator information, which resides in separate tables (eg. MK3_CORRELATOR_MODEL, MK4_CORRELATOR_GEOMETRY), but is accessed by uniform MS methods to return common global data such as total model or correlator correction information. The format of these sub-tables needs to form part of a global MS definition, with uniform column names used wherever possible. The same subset of MAIN indices are used to access the correlator sub-tables, in conjunction with TYPE_ID. The alternative representation of diverse correlator information in a single, common table will lead to a proliferation of optional columns, which will obscure the underlying physical correlator model in most cases.

- **Correlator mode information**

  Within a given correlator run, separate subarrays may be correlated in different modes which are distinguished by different setup parameters (eg. FFT size, number of lags, spectral averaging) but which share the same global correlator model. This can be varied independently of TYPE_ID and a secondary index MODE_ID is proposed for this purpose. Separate correlator mode sub-tables are required to represent this information (eg. MK3_CORRELATOR_MODE), as discussed in the previous item.

- **Correlator pass information**

  Redundant data are often produced by multiple VLBI correlator passes. Traditionally, no record is kept of the correlation pass for each uv-record and redundant data may be rejected by other criteria, such as comparing weights on a record-by-record basis. Retaining correlator pass information adds flexibility to the rejection of redundant data, allows more versatile data selection and simplifies the filling of VLBI data. A separate table is proposed to store correlator pass information (including correlator job number), which is indexed on the pass number and time. The correlator pass information is numbered relative to each correlator job identifier.

4.0.5 Frequency offsets (FREQ_OFFSET)

Time-variable frequency tracking at each antenna is properly modeled by incorporating an antenna-based frequency offset for each antenna in each uv-record. This retains the same SPECTRAL_WINDOW_ID for calibration purposes, if appropriate, but accurately reflects frequency changes throughout the observations. These data are indexed by the MAIN table columns (TIME, ANTENNAn, FEED, SPECTRAL_WINDOW_ID).

4.0.6 Orbiting VLBI (TRACKING_STN)

In orbiting VLBI observations, each antenna can be thought of as consisting of both an orbiting component and a ground-based tracking station component, with different calibration errors associated with each. Both require entries in the ANTENNA table and can in principle vary independently. A new TRACKING_STN table is proposed to fully support calibration and data selection in this case, indexed by MAIN columns (TIME, ANTENNA).

4.0.7 DATA labeling (DATA_DESCRIPTION)

A new table is added to hold the data description information, as discussed in Section 3.1.3. As described in the same Section, polarization information is stored in a POLARIZATION sub-table.

4.0.8 Doppler tracking (DOPPLER)

A new doppler tracking table is added to store source-based doppler tracking information for each SPECTRAL_WINDOW, as discussed in Section 3.10.2.

In spectral-line observing, both the on-line telescope and back-end processor control systems and the off-line data reduction package require information specifying the Doppler tracking parameters used during observing or required for post-processing. There are many different possibilities for how Doppler tracking is implemented in practice; these depend on the instrument and science goals of the observing. The MSv2 data format attempts to represent this information in a sufficiently general format that is not instrument-dependent, and covers all Doppler tracking strategies.

In general, Doppler tracking parameters can be sub-divided into those that are dependent on the astronomical source or field, and those that capture the frequency tracking strategy used at the telescope and/or back-end processor. The source-based parameters include:

(a) Celestial coordinates of the source or field position used for Doppler tracking.

(b) Properties of the associated transitions ($1..N_{lines}$).

  (i) Transition name (e.g. CO J=2-1, HI etc.).
  (ii) Systemic velocity (V_{sys}).
  (iii) Rest frequency (ν_0).

(c) Options for frequency tracking during observing include:

  (i) No tracking on-line; shift spectra in post-processing.
  (ii) Coarse tracking on-line; refine shift of spectra in post-processing.
  (iii) Fine tracking on-line; no refinement in post-processing.
  (iv) Shift spectra in post-processing to a new line or new frame, not tracked during observing, but lying in the observed frequency band.

In MSv2, the source properties (a,b) are recorded in the SOURCE sub-table, which is accordingly not optional if Doppler tracking is employed. The transition information is SPECTRAL_WINDOW_ID dependent, as reflected in the SOURCE sub-table key. The SPECTRAL_WINDOW_ID is part of the primary key in the MAIN table, via DATA_DESC_ID, and can accordingly be used in any sub-table key without introducing a circular dependency. If only the field center is tracked, then a SOURCE entry for the field center should be added, named accordingly. The transition list in the SOURCE table should include only those transitions of scientific interest to the observer, not all possible transitions in the frequency band. With increasing observing bandwidths, multiple transition entries will become more common however.

The frequency tracking strategies (c) are recorded in the SPECTRAL_WINDOW, FREQ_OFFSET and DOPPLER sub-tables. An underlying principle is that spectral windows which can be calibrated together should share a common SPECTRAL_WINDOW_ID wherever possible. For minimal or no on-line tracking (c-i,ii), the frequency in the SPECTRAL_WINDOW table should be labelled as topocentric, and will be fixed. Option (c-ii) is accommodated by adding supplementary offset in the FREQ_OFFSET sub-table as required. This table can also be used to accomodate antenna-based and other frequency offsets in a flexible manner. Both (c-i,ii) have DOPPLER_ID = -1.

For (c-iii), the frequency axis can be labelling in a frame in which it is constant, to within meaningful scientific precision. In this case, the DOPPLER_ID index points to the DOPPLER entry describing the transition information used in the on-line tracking. This can be source-dependent, as the DOPPLER key includes a SOURCE_ID. For Doppler tracking in software (c-iv), the DOPPLER_ID key should be adjusted to reflect the resultant frequency shift, in the same way as described for on-line Doppler tracking.

4.0.9 Scan summary information

An optional new table is proposed to store summary information for each observation, array and scan, as outlined in Section 3.3. This table may need to be re-generated if data are flagged or removed, however. It will provide useful user information, and may in addition be used by applications if current.

4.0.10 Associated tables (CAL_TABLES and SORTED_TABLES)

Links to associated sorted reference MeasurementSets, and calibration tables are provided in these two MS sub-tables.

5 MS v2.0 layout

There is a MAIN table containing a number of data columns and keys into various subtables. There is at most one of each subtable. The subtables are stored as keywords of the MS, and all defined sub-tables are tabulated below. Optional sub-tables are shown in italics.

Subtables		
Table	Contents	Keys
ANTENNA	Antenna characteristics	ANTENNA_ID
DATA_DESCRIPTION	Data description	DATA_DESC_ID
(*DOPPLER*)	Doppler tracking	DOPPLER_ID, SOURCE_ID
FEED	Feed characteristics	FEED_ID, ANTENNA_ID, TIME, SPECTRAL_WINDOW_ID
FIELD	Field position	FIELD_ID
FLAG_CMD	Flag commands	TIME
(*FREQ_OFFSET*)	Frequency offset information	FEED_ID, ANTENNAn, FEED_ID, TIME, SPECTRAL_WINDOW_ID
HISTORY	History information	OBSERVATION_ID, TIME
OBSERVATION	Observer, Schedule, etc	OBSERVATION_ID
POINTING	Pointing information	ANTENNA_ID, TIME
POLARIZATION	Polarization setup	POLARIZATION_ID
PROCESSOR	Processor information	PROCESSOR_ID
(*SOURCE*)	Source information	SOURCE_ID, SPECTRAL_WINDOW_ID, TIME
SPECTRAL_WINDOW	Spectral window setups	SPECTRAL_WINDOW_ID
STATE	State information	STATE_ID
(*SYSCAL*)	System calibration characteristics	FEED_ID, ANTENNA_ID, TIME, SPECTRAL_WINDOW_ID
(*WEATHER*)	Weather info for each antenna	ANTENNA_ID, TIME

Note that there are two types of subtables. For the first, simpler type, the key (ID) is the row number in the subtable. Examples are FIELD, SPECTRAL_WINDOW, OBSERVATION and PROCESSOR. For the second, the key is a collection of parameters, usually including TIME. Examples are FEED, SOURCE, SYSCAL, WEATHER.

Note that all optional columns are indicated in italics and in parentheses.

5.1 MAIN table: Data, coordinates and flags

MAIN table: Data, coordinates and flags				
Name	Format	Units	Measure	Comments
Columns				
Keywords				
MS_VERSION	Float			MS format version
(SORT_COLUMNS)	String			Sort columns
(SORT_ORDER)	String			Sort order
Key				
TIME	Double	s	EPOCH	Integration midpoint
(TIME_EXTRA_PREC)	Double	s		extraTIME precision
ANTENNA1	Int			First antenna
ANTENNA2	Int			Second antenna
(ANTENNA3)	Int			Third antenna
FEED1	Int			Feed on ANTENNA1
FEED2	Int			Feed on ANTENNA2
(FEED3)	Int			Feed on ANTENNA3
DATA_DESC_ID	Int			Data desc. id.
PROCESSOR_ID	Int			Processor id.
(PHASE_ID)	Int			Phase id.
FIELD_ID	Int			Field id.
Non-key attributes				
INTERVAL	Double	s		Sampling interval
EXPOSURE	Double	s		The effective integration time
TIME_CENTROID	Double	s	EPOCH	Time centroid
(PULSAR_BIN)	Int			Pulsar bin number
(PULSAR_GATE_ID)	Int			Pulsar gate id.
SCAN_NUMBER	Int			Scan number
ARRAY_ID	Int			Subarray number
OBSERVATION_ID	Int			Observation id.
STATE_ID	Int			State id.
(BASELINE_REF)	Bool			Reference antenna
UVW	Double(3)	m	UVW	UVW coordinates
(UVW2)	Double(3)	m	UVW	UVW (baseline 2)

5.2 MAIN table: continued

MAIN table: continued				
Name	Format	Units	Measure	Comments
Data				
(DATA)	Complex(N_c, N_f)			Complex visibility matrix (synthesis arrays)
(FLOAT_DATA)	Float(N_c, N_f)			Float data matrix (single dish)
(VIDEO_POINT)	Complex(N_c)			Video point
(LAG_DATA)	Complex(N_c, N_l)			Correlation function
SIGMA	Float(N_c)			Estimated rms noise for single channel
(SIGMA_SPECTRUM)	Float(N_c, N_f^*)			Estimated rms noise
WEIGHT	Float(N_c)			Weight for whole data matrix
(WEIGHT_SPECTRUM)	Float(N_c, N_f^*)			Weight for each channel
Flag information				
FLAG	Bool(N_c, N_f^*)			Cumulative data flags
FLAG_CATEGORY	Bool(N_c, N_f^*, N_{cat})			Flag categories
FLAG_ROW	Bool			The row flag

Notes:

Note that N_l= number of lags, N_c= number of correlators, N_f= number of frequency channels, and N_{cat}= number of flag categories.

MS_VERSION The MeasurementSet format revision number, expressed as *major_revision.minor_revision*. This version is 2.0.

SORT_COLUMNS Sort indices, in the form "$index_1$, $index_2$ ..., for the underlying MS. A string containing "NONE" reflects no sort order. An example might be SORT_COLUMNS="TIME ANTENNA1 ANTENNA2", to indicate sorting in in time-baseline order.

SORT_ORDER Sort order as either "ASCENDING" or "DESCENDING".

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

TIME_EXTRA_PREC Extra time precision.

ANTENNAn Antenna number (≥ 0), and a direct index into the ANTENNA sub-table *rownr*. For $n > 2$, triple-product data are implied.

FEEDn Feed number (≥ 0). For $n > 2$, triple-product data are implied.

DATA_DESC_ID Data description identifier (≥ 0), and a direct index into the DATA_DESCRIPTION sub-table *rownr*.

PROCESSOR_ID Processor indentifier (≥ 0), and a direct index into the PROCESSOR sub-table *rownr*.

PHASE_ID Switching phase identifier (≥ 0)

FIELD_ID Field identifier (≥ 0).

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

EXPOSURE Effective data interval, including bad data and partial averaging.

PULSAR_BIN Pulsar bin number for the data record. Pulsar data may be measured for a limited number of pulse phase bins. The pulse phase bins are described in the PULSAR sub-table and indexed by this bin number.

PULSAR_GATE_ID Pulsar gate identifier (≥ 0), and a direct index into the PULSAR_GATE sub-table *rownr*.

SCAN_NUMBER Arbitrary scan number to identify data taken in the same logical scan. Not required to be unique.

ARRAY_ID Subarray identifier (≥ 0), which identifies data in separate subarrays, as defined in Section 3.3.

OBSERVATION_ID Observation identifier (≥ 0), which identifies data from separate observations, as defined in Section 3.3.

STATE_ID State identifier (≥ 0), as defined in Section 3.1.5.

BASELINE_REF Flag to indicate the original correlator reference antenna for baseline-based correlators (True for ANTENNA1; False for ANTENNA2).

UVW *uvw* coordinates for the baseline from ANTENNE2 to ANTENNA1, i.e. the baseline is equal to the difference POSITION2 - POSITION1. The UVW given are for the TIME_CENTROID, and correspond in general to the reference type for the PHASE_DIR of the relevant field. I.e. J2000 if the phase reference direction is given in J2000 coordinates. However, any known reference is valid. Note that the choice of baseline direction and UVW definition (W towards source direction; V in plane through source and system's pole; U in direction of increasing longitude coordinate) also determines the sign of the phase of the recorded data.

UVW2 *uvw* coordinates for the baseline from ANTENNE3 to ANTENNA1 (triple-product data only), i.e. the baseline is equal to the difference POSITION3 - POSITION1. The UVW given are for the TIME_CENTROID, and correspond in general to the reference type for the PHASE_DIR of the relevant field. I.e. J2000 if the phase reference direction is given in J2000 coordinates. However, any known reference is valid. Note that the choice of baseline direction and UVW definition (W towards source direction; V in plane through source and system's pole; U in direction of increasing longitude coordinate) also determines the sign of the phase of the recorded data.

DATA, FLOAT_DATA, LAG_DATA At least one of these columns should be present in a given MeasurementSet. In special cases one or more could be present (e.g., single dish data used in synthesis imaging or a mix of auto and crosscorrelations on a multi-feed single dish). If only correlation functions are stored in the MS, then N_f^* is the maximum number of lags (N_l) specified in the LAG table for this LAG_ID. If both correlation functions and frequency spectra are stored in the same MS, then N_f^* is the number of frequency channels, and the weight information refers to the frequency spectra only. The units for these columns (eg. 'Jy') specify whether the data are in flux density units or correlation coefficients.

VIDEO_POINT The video point for the spectrum, to allow the full reverse transform.

SIGMA The estimated rms noise for a single channel, for each correlator.

SIGMA_SPECTRUM The estimated rms noise for each channel.

WEIGHT The weight for the whole data matrix for each correlator, as assigned by the correlator or processor.

WEIGHT_SPECTRUM The weight for each channel in the data matrix, as assigned by the correlator or processor. The weight spectrum should be used in preference to the WEIGHT, when available.

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

FLAG_CATEGORY An array of flag matrices with the same shape as DATA, but indexed by category. The category identifiers are specified by a keyword CATEGORY, containing an array of string identifiers, attached to the FLAG_CATEGORY column and thus shared by all rows in the MeasurementSet. The cumulative effect of these flags is reflected in column FLAG. Data are flagged bad if the FLAG array element is True. See Section 3.1.8 for further details.

FLAG_ROW True if the entire row is flagged.

5.3 ANTENNA: Antenna characteristics

ANTENNA: Antenna characteristics				
Name	Format	Units	Measure	Comments
Columns				
Data				
NAME	String			Antenna name
STATION	String			Station name
TYPE	String			Antenna type
MOUNT	String			Mount type:alt-az, equatorial, X-Y, orbiting, bizarre
POSITION	Double(3)	m	POSITION	Antenna X,Y,Z phase reference positions
OFFSET	Double(3)	m	POSITION	Axes offset of mount to FEED REFERENCE point
DISH_DIAMETER	Double	m		Diameter of dish
(ORBIT_ID)	Int			Orbit id.
(MEAN_ORBIT)	Double(6)			Mean Keplerian elements
(PHASED_ARRAY_ID)	Int			Phased array id.
Flag information				
FLAG_ROW	Bool			Row flag

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

NAME Antenna name (e.g. "NRAO_140")

STATION Station name (e.g. "GREENBANK")

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

MOUNT Mount type of the antenna. Reserved keywords include: ("EQUATORIAL" - equatorial mount; "ALT-AZ" - azimuth-elevation mount; "X-Y" - x-y mount; "SPACE-HALCA" - specific orientation model.)

POSITION In a right-handed frame, X towards the intersection of the equator and the Greenwich meridian, Z towards the pole. The exact frame should be specified in the MEASURE_REFERENCE keyword (ITRF or WGS84). The reference point is the point on the az or ha axis closest to the el or dec axis.

OFFSET Axes offset of mount to feed reference point.

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

ORBIT_ID Orbit identifier. Index used in ORBIT sub-table if ANTENNA_TYPE is "SPACE_BASED".

MEAN_ORBIT Mean Keplerian orbital elements, using the standard convention (Flatters 1998):

- **0:** Semi-major axis of orbit (a) in m.
- **1:** Ellipticity of orbit (e).
- **2:** Inclination of orbit to the celestial equator (i) in deg.
- **3:** Right ascension of the ascending node (Ω) in deg.
- **4:** Argument of perigee (ω) in deg.

- **5:** Mean anomaly (M) in deg.

PHASED_ARRAY_ID Phased array identifier. Points to a PHASED_ARRAY sub-table which points back to multiple entries in the ANTENNA sub-table and contains information on how they are combined.

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

5.4 DATA_DESCRIPTION: Data description table

DATA_DESCRIPTION: Data description table				
Name	Format	Units	Measure	Comments
Columns				
Data				
SPECTRAL_WINDOW_ID	Int			Spectral window id.
POLARIZATION_ID	Int			Polarization id.
(LAG_ID)	Int			Lag fn. id.
Flags				
FLAG_ROW	Bool			Row flag.

Notes: This table define the shape of the associated DATA array in MAIN, and in indexed directly by DATA_DESC_ID.

SPECTRAL_WINDOW_ID Spectral window identifier.

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

LAG_ID Lag function identifier (≥ 0), and a direct index into the LAG sub-table *rownr*.

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

5.5 DOPPLER: Doppler tracking information

DOPPLER: Doppler tracking information				
Name	Format	Units	Measure	Comments
Columns				
Key				
DOPPLER_ID	Int			Doppler tracking id.
SOURCE_ID	Int			Source id.
Data				
TRANSITION_ID	Int			Transition id.
VELDEF	Double	m/s	Doppler	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 model 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.

VELDEF Velocity definition of the Doppler shift, e.g., RADIO or OPTICAL velocity in m/s.

5.6 FEED: Feed characteristics

FEED: Feed characteristics				
Name	Format	Units	Measure	Comments
Columns				
Key				
ANTENNA_ID	Int			Antenna id
FEED_ID	Int			Feed id
SPECTRAL_WINDOW_ID	Int			Spectral window id.
TIME	Double	s	EPOCH	Interval midpoint
INTERVAL	Double	s		Time interval
Data description				
NUM_RECEPTORS	Int			# receptors on this feed
Data				
BEAM_ID	Int			Beam model
BEAM_OFFSET	Double(2, NUM_RECEPTORS)	rad	DIRECTION	Beam position offset (on sky but in antenna reference frame).
(FOCUS_LENGTH)	Double	m		Focus length
(PHASED_FEED_ID)	Int			Phased feed
POLARIZATION_TYPE	String (NUM_RECEPTORS)			Type of polarization to which a given RECEPTOR responds.
POL_RESPONSE	Complex (NUM_RECEPTORS, NUM_RECEPTORS)			Feed polzn. response
POSITION	Double(3)	m	POSITION	Position of feed relative to feed reference position for this antenna
RECEPTOR_ANGLE	Double (NUM_RECEPTORS)	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 for each SPECTRAL_WINDOW_ID. Consequently, FEED_ID should be non-zero only in the case of feed arrays, i.e. multiple, simultaneous beams on the sky at the same frequency and polarization.

ANTENNA_ID Antenna number, as indexed from ANTENNAn in MAIN.

FEED_ID Feed identifier, as indexed from FEEDn in MAIN.

SPECTRAL_WINDOW_ID Spectral window identifier. A value of -1 indicates the row is valid for all spectral windows.

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

INTERVAL Time interval.

NUM_RECEPTORS Number of receptors on this feed. See POLARIZATION_TYPE for further information.

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

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

FOCUS_LENGTH Focus length. As defined along the optical axis of the antenna.

PHASED_FEED_ID Phased feed identifier. Points to a PHASED_FEED sub-table which in turn points back to multiple entries in the FEED table, and specifies the manner in which they are combined.

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

POL_RESPONSE Polarization response at the center of the beam for this feed. Expressed in a linearly polarized basis $(\vec{\mathbf{e}}_{\mathbf{x}}, \vec{\mathbf{e}}_{\mathbf{y}})$ using the IEEE convention.

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

RECEPTOR_ANGLE Polarization reference angle. Converts into parallactic angle in the sky domain.

5.7 FIELD: Field positions for each source

FIELD: Field positions for each source				
Name	Format	Units	Measure	Comments
Columns				
Key				
Data				
NAME	String			Name of field
CODE	String			Special characteristics of field
TIME	Double	s	EPOCH	Time origin for the directions and rates
NUM_POLY	Int			Series order
DELAY_DIR	Double(2, NUM_POLY+1)	rad	DIRECTION	Direction of delay center.
PHASE_DIR	Double(2, NUM_POLY+1)	rad	DIRECTION	Phase center.
REFERENCE_DIR	Double(2, NUM_POLY+1)	rad	DIRECTION	Reference center
SOURCE_ID	Int			Index in Source table
(EPHEMERIS_ID)	Int			Ephemeris id.
Flags				
FLAG_ROW	Bool			Row flag

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

NAME Field name; user specified.

CODE Field code indicating special characteristics of the field; user specified.

TIME Time reference for the directions and rates. Required to use the same TIME Measure reference as in 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 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 DELAY_DIR or PHASE_DIR.

SOURCE_ID Points to an entry in the optional SOURCE subtable, a value of -1 indicates there is no corresponding source defined.

EPHEMERIS_ID Points to an entry in 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.

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

5.8 FLAG_CMD: Flag commands

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

Notes: The FLAG_CMD sub-table defines global flagging commands which apply to the data in MAIN, as described in Section 3.1.8.

TIME Mid-point of the time interval to which this flagging command applies. Required to use the same TIME Measure reference as used in MAIN.

INTERVAL Time interval.

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

REASON Flag reason; user specified.

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

SEVERITY Severity code for the flag, on a scale of 0-10 in order of increasing severity; user specified.

APPLIED True if this 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 data selection, as adopted within the project as a whole.

5.9 FREQ_OFFSET: Frequency offset information

FREQ_OFFSET: Frequency offset information				
Name	Format	Units	Measure	Comments
Columns				
Key				
ANTENNA1	Int			Antenna 1.
ANTENNA2	Int			Antenna 2.
FEED_ID	Int			Feed id.
SPECTRAL_WINDOW_ID	Int			Spectral window id.
TIME	Double	s	EPOCH	Interval midpoint
INTERVAL	Double	s		Time 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.

ANTENNAn Antenna identifier, as indexed from ANTENNAn in MAIN.

FEED_ID Antenna identifier, as indexed from FEEDn in MAIN.

SPECTRAL_WINDOW_ID Spectral window identifier.

TIME Mid-point of the time interval for which this offset is valid. Required to use the same TIME Measure reference as used in MAIN.

INTERVAL Time interval.

OFFSET Frequency offset to be added 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.

5.10 HISTORY: History information

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

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

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

OBSERVATION_ID Observation identifier (see the OBSERVATION table)

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_PARAMS Application parameter values, in the adopted project-wide format.

5.11 OBSERVATION: Observation information

OBSERVATION: Observation information				
Name	Format	Units	Measure	Comments
Columns				
Data				
TELESCOPE_NAME	String			Telescope name
TIME_RANGE	Double(2)	s	EPOCH	Start, end times
OBSERVER	String			Name of observer(s)
LOG	String(*)			Observing log
SCHEDULE_TYPE	String			Schedule type
SCHEDULE	String(*)			Project schedule
PROJECT	String			Project identification string.
RELEASE_DATE	Double	s	EPOCH	Target release date
Flags				
FLAG_ROW	Bool			Row flag.

Notes: This table contains information specifying the observing instrument or epoch. See the discussion in Section 3.3 for details. It is indexed directly from MAIN via OBSERVATION_ID.

TELESCOPE_NAME Telescope name (e.g. "WSRT" or "VLBA").

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

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

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

SCHEDULE_TYPE The schedule type, with current reserved types ("VLBA-CRD", "VEX", "WSRT", "ATNF").

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

PROJECT Project code (e.g. "BD46")

RELEASE_DATE Project release date. This is the date on which the data may become public.

FLAG_ROW Row flag. True if data in this row is invalid, but does not imply any flagging in MAIN.

5.12 POINTING: Antenna pointing information

POINTING: Antenna pointing information				
Name	Format	Units	Measure	Comments
Columns				
Key				
ANTENNA_ID	Int			Antenna id.
TIME	Double	s	EPOCH	Interval midpoint
INTERVAL	Double	s		Time interval
Data				
NAME	String			Pointing position desc.
NUM_POLY	Int			Series order
TIME_ORIGIN	Double	s	EPOCH	Origin for the polynomial
DIRECTION	Double(2, NUM_POLY+1)	rad	DIRECTION	Antenna pointing direction
TARGET	Double(2, NUM_POLY+1)	rad	DIRECTION	Target direction
(POINTING_OFFSET)	Double(2, NUM_POLY+1)	rad	DIRECTION	A priori pointing correction
(SOURCE_OFFSET)	Double(2, NUM_POLY+1)	rad	DIRECTION	Offset from source
(ENCODER)	Double(2)	rad	DIRECTION	Encoder values
(POINTING_MODEL_ID)	Int			Pointing model id.
TRACKING	Bool			True if on-position
(ON_SOURCE)	Bool			True if on-source
(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 offsets for inidividual feeds on a given antenna are specified in the FEED sub-table with respect to this pointing direction.

ANTENNA_ID Antenna identifier, as specified by ANTENNAn in MAIN.

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

INTERVAL Time interval.

NAME Pointing direction name; user specified.

NUM_POLY Series order for the polynomial expressions in DIRECTION and POINTING_OFFSET.

TIME_ORIGIN Time origin for the polynomial expansions.

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.

POINTING_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.

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.

TRACKING True if tracking the nominal pointing position.

ON-SOURCE True if the nominal pointing direction coincides with the source, i.e. offset-pointing is not being used.

OVER-THE-TOP True if the antenna was driven to this position "over the top" (az-el mount).

5.13 POLARIZATION: Polarization setup information

POLARIZATION: Polarization setup information				
Name	Format	Units	Measure	Comments
Columns				
Data description columns				
NUM_CORR	Int			# correlations
Data				
CORR_TYPE	Int(NUM_CORR)			Polarization of correlation
CORR_PRODUCT	Int(2, NUM_CORR)			Receptor cross-products
Flags				
FLAG_ROW	Bool			Row flag

Notes: This table defines the polarization labeling of the DATA array in MAIN, and is directly indexed from the DATA_DESCRIPTION table via POLARIZATION_ID.

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

CORR_TYPE An integer for each correlation product indicating the Stokes type as defined in the Stokes class enumeration.

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

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

5.14 PROCESSOR: Processor information

PROCESSOR: Processor information				
Name	Format	Units	Measure	Comments
Columns				
Data				
TYPE	String			Processor type
SUB_TYPE	String			Processor sub-type
TYPE_ID	Int			Processor type id.
MODE_ID	Int			Processor mode id.
(PASS_ID)	Int			Processor pass number
Flags				
FLAG_ROW	Bool			Row flag

Notes: This table holds summary information for the back-end processing device used to generate the basic data in the MAIN table. Such devices include correlators, radiometers, spectrometers, pulsar-timers, amongst others. See Section 4.0.4 for further details.

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

SUB_TYPE Processor sub-type, e.g. "GBT" or "JIVE".

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

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

PASS_ID Pass identifier; this is used to distinguish data records produced by multiple passes through the same device, where this is possible (e.g. VLBI correlators). Used as an index into the associated table containing pass information.

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

5.15 SOURCE: Source information

SOURCE: Source information				
Name	Format	Units	Measure	Comments
Columns				
Key				
SOURCE_ID	Int			Source id
TIME	Double	s	EPOCH	Midpoint of time for which this set of parameters is accurate
INTERVAL	Double	s		Interval
SPECTRAL_WINDOW_ID	Int			Spectral Window id
Data description				
NUM_LINES	Int			Number of spectral lines
Data				
NAME	String			Name of source as given during observations
CALIBRATION_GROUP	Int			# grouping for calibration purpose
CODE	String			Special characteristics of source, e.g. Bandpass calibrator
DIRECTION	Double(2)	rad	DIRECTION	Direction (e.g. RA, DEC)
(POSITION)	Double(3)	m	POSITION	Position (e.g. for solar system objects)
PROPER_MOTION	Double(2)	rad/s		Proper motion
(TRANSITION)	String(NUM_LINES)			Transition name
(REST_FREQUENCY)	Double(NUM_LINES)	Hz	FREQUENCY	Line rest frequency
(SYSVEL)	Double(NUM_LINES)	m/s	RADIAL VELOCITY	Systemic velocity at reference
(SOURCE_MODEL)	TableRecord			Default csm
(PULSAR_ID)	Int			Pulsar id.

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

SOURCE_ID Source identifier (≥ 0), as specified in the FIELD sub-table.

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

INTERVAL Time interval.

SPECTRAL_WINDOW_ID Spectral window identifier. A -1 indicates that the row is valid for all spectral windows.

NUM_LINES Number of spectral line transitions associated with this source and spectral window id. combination.

NAME Source name; user specified.

CALIBRATION_GROUP Calibration group number to which this source belongs; user specified.

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

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, SiO").

REST_FREQUENCY Rest frequencies for the transitions.

SYSVEL Systemic velocity for each transition.

SOURCE_MODEL Reference to an assigned component source model table.

PULSAR_ID An index used in the PULSAR sub-table to define further pulsar-specific properties if the source is a pulsar.

5.16 SPECTRAL_WINDOW: Spectral window description

SPECTRAL_WINDOW: Spectral window description				
Name	Format	Units	Measure	Comments
Columns				
Data description columns				
NUM_CHAN	Int			# spectral channels
Data				
NAME	String			Spectral window name
REF_FREQUENCY	Double	Hz	FREQUENCY	The reference frequency.
CHAN_FREQ	Double(NUM_CHAN)	Hz	FREQUENCY	Center frequencies for each channel in the data matrix.
CHAN_WIDTH	Double(NUM_CHAN)	Hz		Channel width for each channel in the data matrix.
MEAS_FREQ_REF	Int			FREQUENCY Measure ref.
EFFECTIVE_BW	Double(NUM_CHAN)	Hz		The effective noise bandwidth of each spectral channel
RESOLUTION	Double(NUM_CHAN)	Hz		The effective spectral resolution of each channel
TOTAL_BANDWIDTH	Double	Hz		total bandwidth for this window
NET_SIDEBAND	Int			Net sideband
(BBC_NO)	Int			Baseband converter no.
(BBC_SIDEBAND)	Int			BBC sideband
IF_CONV_CHAIN	Int			The IF conversion chain
(RECEIVER_ID)	Int			Receiver id.
FREQ_GROUP	Int			Frequency group
FREQ_GROUP_NAME	String			Freq. group name
(DOPPLER_ID)	Int			Doppler id.
(ASSOC_SPW_ID)	Int(*)			Associated spw_id.
(ASSOC_NATURE)	String(*)			Nature of association
Flags				
FLAG_ROW	Bool			

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

NUM_CHAN Number of spectral channels.

NAME Spectral window name; user specified.

REF_FREQUENCY The reference frequency. A frequency representative of this spectral window, usually the sky frequency corresponding to the DC edge of the baseband. Used by the calibration system if a fixed scaling frequency is required or in algorithms to identify the observing band.

CHAN_FREQ Center frequencies for each channel in the data matrix. These can be frequency-dependent, to accommodate instruments such as acousto-optical spectrometers. Note that the channel frequencies may be in ascending or descending frequency order.

CHAN_WIDTH Nomical channel width of each spectral channel. Although these can be derived from CHAN_FREQ by differencing, it is more efficient to keep a separate reference to this information.

MEAS_FREQ_REF Frequency Measure reference for CHAN_FREQ. This allows a row-based reference for this column in order to optimize the choice of Measure reference when Doppler tracking is used. Modified only by the MS access code.

EFFECTIVE_BW The effective noise bandwidth of each spectral channel.

RESOLUTION The effective spectral resolution of each channel.

TOTAL_BANDWIDTH The total bandwidth for this spectral window.

NET_SIDEBAND The net sideband for this spectral window.

BBC_NO The baseband converter number, if applicable.

BBC_SIDEBAND The baseband converter sideband, is applicable.

IF_CONV_CHAIN Identification of the electronic signal path for the case of multiple (simultaneous) IFs. (e.g. VLA: AC=0, BD=1, ATCA: Freq1=0, Freq2=1)

RECEIVER_ID Index used to identify the receiver associated with the spectral window. Further state information is planned to be stored in a RECEIVER sub-table.

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

FREQ_GROUP_NAME The frequency group name; user specified.

DOPPLER_ID The 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 ("CHANNEL-ZERO" - channel zero; "EQUAL-FREQUENCY" - same frequency labels; "SUBSET" - narrow-band subset).

FLAG_ROW True if the row does not contain valid data.

5.17 STATE: State information

STATE: State information				
Name	Format	Units	Measure	Comments
Columns				
Data				
SIG	Bool			Signal
REF	Bool			Reference
CAL	Double	K		Noise calibration
LOAD	Double	K		Load temperature
SUB_SCAN	Int			Sub-scan number
OBS_MODE	String			Observing mode
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 STATE_ID in MAIN.

SIG True if the source signal is being observed.

REF True for a reference phase.

CAL Noise calibration temperature (zero if not added).

LOAD Load temperature (zero if no load).

SUB_SCAN Sub-scan number (≥ 0), relative to the SCAN_NUMBER in MAIN. Used to identify observing sequences.

OBS_MODE Observing mode; defined by a set of reserved keywords characterizing the current observing mode (e.g. "OFF-SPECTRUM"). Used to define the schedule strategy.

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

5.18 SYSCAL: System calibration

SYSCAL: System calibration				
Name	Format	Units	Measure	Comments
Columns				
Key				
ANTENNA_ID	Int			Antenna id
FEED_ID	Int			Feed id
SPECTRAL_WINDOW_ID	Int			Spectral window id
TIME	Double	s	EPOCH	Midpoint of time for which this set of parameters is accurate
INTERVAL	Double	s		Interval
Data				
(PHASE_DIFF)	Float	rad		Phase difference between receptor 0 and receptor 1
(TCAL)	Float (N_r)	K		Calibration temp
(TRX)	Float (N_r)	K		Receiver temperature
(TSKY)	Float (N_r)	K		Sky temperature
(TSYS)	Float (N_r)	K		System temp
(TANT)	Float (N_r)	K		Antenna temperature
(TANT_TSYS)	Float(N_r)			$\frac{T_{ant}}{T_{sys}}$
(TCAL_SPECTRUM)	Float (N_r, N_f)	K		Calibration temp
(TRX_SPECTRUM)	Float (N_r, N_f)	K		Receiver temperature
(TSKY_SPECTRUM)	Float (N_r, N_f)	K		Sky temperature spectrum
(TSYS_SPECTRUM)	Float (N_r, N_f)	K		System temp
(TANT_SPECTRUM)	Float (N_r, N_f)	K		Antenna temperature spectrum
(TANT_TSYS_SPECTRUM)	Float (N_r,N_f)			$\frac{T_{ant}}{T_{sys}}$ spectrum
Flags				
(PHASE_DIFF_FLAG)	Bool			Flag for PHASE_DIFF
(TCAL_FLAG)	Bool			Flag for TCAL
(TRX_FLAG)	Bool			Flag for TRX
(TSKY_FLAG)	Bool			Flag for TSKY
(TSYS_FLAG)	Bool			Flag for TSYS
(TANT_FLAG)	Bool			Flag for TANT
(TANT_TSYS_FLAG)	Bool			Flag for $\frac{T_{ant}}{T_{sys}}$

Notes: This table contains time-variable calibration measurements for each antenna, as indexed on feed and spectral window. Note that N_r= number of receptors, and N_f= number of frequency channels.

ANTENNA_ID Antenna identifier, as indexed by ANTENNAn in MAIN.

FEED_ID Feed identifier, as indexed by FEEDn in MAIN.

SPECTRAL_WINDOW_ID Spectral window identifier.

TIME Mid-point of the time interval for which the data in this row are valid. Required to use the same TIME Measure reference as that in MAIN.

INTERVAL Time interval.

PHASE_DIFF Phase difference between receptor 0 and receptor 1.

TCAL Calibration temperature.

TRX Receiver temperature.

TSKY Sky temperature.

TSYS System temperature.

TANT Antenna temperature.

TANT_TSYS Antenna temperature over system temperature.

TCAL_SPECTRUM Calibration temperature spectrum.

TRX_SPECTRUM Receiver temperature spectrum.

TSKY_SPECTRUM Sky temperature spectrum.

TSYS_SPECTRUM System temperature spectrum.

TANT_SPECTRUM Antenna temperature spectrum.

TANT_TSYS_SPECTRUM Antenna temperature over system temperature spectrum.

PHASE_DIFF_FLAG True if PHASE_DIFF 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 TANT_TSYS flagged.

5.19 WEATHER: weather station information

WEATHER: weather station information				
Name	Format	Units	Measure	Comments
Columns				
Key				
ANTENNA_ID	Int			Antenna number
TIME	Double	s	EPOCH	Mid-point of interval
INTERVAL	Double	s		Interval over which data is relevant
Data				
(H2O)	Float	m^{-2}		Average column density of water
(IONOS_ELECTRON)	Float	m^{-2}		Average column density of electrons
(PRESSURE)	Float	hPa		Ambient atmospheric pressure
(REL_HUMIDITY)	Float			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		Average wind speed
Flags				
(H2O_FLAG)	Bool			Flag for H2O
(IONOS_ELECTRON_FLAG)	Bool			Flag for IONOS_ELECTRON
(PRESSURE_FLAG)	Bool			Flag for PRESSURE
(REL_HUMIDITY_FLAG)	Bool			Flag for REL_HUMIDITY
(TEMPERATURE_FLAG)	Bool			Flag for TEMPERATURE
(DEW_POINT_FLAG)	Bool			Flag for DEW_POINT
(WIND_DIRECTION_FLAG)	Bool			Flag for WIND_DIRECTION
(WIND_SPEED_FLAG)	Bool			Flag for WIND_SPEED

Notes: This table contains mean external atmosphere and weather information.

ANTENNA_ID Antenna identifier, as indexed by ANTENNA*n* from MAIN.

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

INTERVAL Time interval.

H2O Average column density of water.

IONOS_ELECTRON Average column density of electrons.

PRESSURE Ambient atmospheric pressure.

REL_HUMIDITY Ambient relative humidity.

TEMPERATURE Ambient air temperature.

DEW_POINT Dew point temperature.

WIND_DIRECTION Average wind direction.

WIND_SPEED Average wind speed.

H2O_FLAG Flag for H2O.

IONOS_ELECTRON_FLAG Flag for IONOS_ELECTRON.

PRESSURE_FLAG Flag for PRESSURE.

REL_HUMIDITY_FLAG Flag for REL_HUMIDITY.

TEMPERATURE_FLAG Flag for TEMPERATURE.

DEW_POINT_FLAG Flag for DEW_POINT.

WIND_DIRECTION_FLAG Flag for DEW_POINT.

WIND_SPEED_FLAG Flag for DEW_POINT.

6 Correction and additions

2000/08/14: Add expanded description of Doppler tracking to Section 4.0.8; correctly identify SOURCE sub-table as optional; expand definition of SPECTRAL_WINDOW_ID=-1 in the description of the SOURCE sub-table; expand the definition of the REF_FREQUENCY in the SPECTRAL_WINDOW sub-table description.

2000/08/28: Make POSITION field in SOURCE table optional.

2000/09/12: Add required VELDEF column in DOPPLER table.

7 References

Date, C.J., 1986, Introduction to database systems, (Addison-Wesley).

Flatters, C., 1998, AIPS Memo. 102.

Noordam, J., 1996, AIPS++Note 185.

Wieringa, M.H., and Cornwell, T., 1996, AIPS++Note 191.