Atacama Pathfinder EXperiment

Interface Control Document

APEX-MPI-ICD-0002
Revision: 1.63
Release: August 5, 2011
Category: 4
Author: Dirk Muders

Multi-Beam FITS Raw Data Format

Dirk Muders, Edward Polehampton and Jennifer Hatchell

The MBFITS working group:
Robert Lucas lucas@iram.fr
Helmut Wiesemeyer wiesemey@iram.fr
Albrecht Sievers sievers@iram.es
Frédéric Gueth gueth@iram.fr
Dirk Muders dmuders@mpifr-bonn.mpg.de

Keywords: Data Format, FITS, APEX

Author Signature: Dirk Muders	Date: August 5, 2011
Approved by: G. Wieching	Signature: G. Wieching
Institute: APEX	Date: August 5, 2011
Released by: G. Wieching	Signature: G. Wieching
Institute: APEX	Date: August 5, 2011

Change Record

Revision	Date	Author	Section/ Page affected	Remarks
0.0	2002	J. Hatchell	All	Initial Version
1.3	2003-07-23	J. Hatchell	All	see 'What's new in v.1.3'
1.4	2003	J. Hatchell	All	see 'What's new in v.1.4'
1.51	2003	J. Hatchell	All	see 'What's new in v.1.51'
1.52	2003-12-18	J. Hatchell	All	see 'What's new in v.1.52'
1.53	2003-11-11	J. Hatchell	All	see 'What's new in v.1.53'
1.54	2004-08-09	J. Hatchell	All	see 'What's new in v.1.54'
1.55	2005-03-22	E. Polehampton	All	see 'What's new in v.1.55'
1.56	2005-09-15	E. Polehampton	All	see 'What's new in v.1.56'
1.57	2005-12-06	D. Muders	21, 40, 43, 56	see 'What's new in v.1.57'
1.6	2006-05-30	D. Muders	38, 39, 46, 50, 60, 61	see 'What's new in v.1.6'
1.61	2007-04-02	D. Muders	Section 15.5.7, Keywords	see 'What's new in v.1.61'
1.62	2007-08-14	D. Muders	DATAPAR table structure, Keywords	see 'What's new in v.1.62'
1.63	2011-08-05	D. Muders	New keywords in several tables	see 'What's new in v.1.63'

Contents

Abstract

This document describes a FITS raw data format for multibeam multireceiver/backend single dish telescopes. It is intended for use at the IRAM 30m, Effelsberg 100m and APEX.

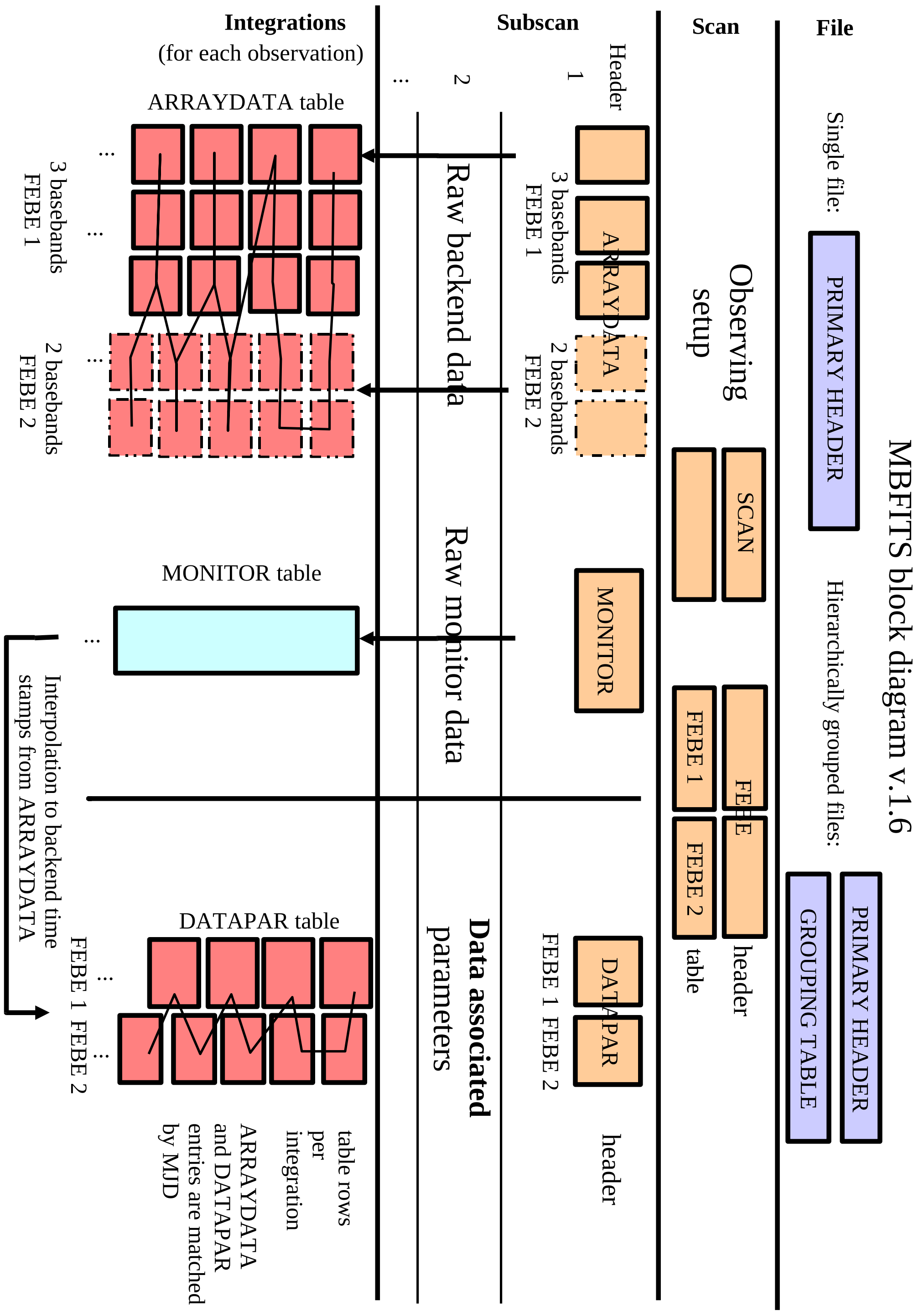


Figure 1: MBFITS file structure

1 Introduction

1.1 MBFITS and ALMA-TI FITS

The MBFITS working group was created at the array receiver meeting at IRAM Grenoble in December 2001. The goal was to define a new raw data format for multibeam receivers based on FITS to be used at the IRAM 30m and APEX telescopes. With a common raw data format it is much easier to share software developments in the areas of data calibration (chopper wheel, atmospheric, etc.) and data reduction. The resulting MBFITS format can be used for all single-dish bolometer and heterodyne observations including multiple frontend/backend combinations and array receivers.

The MBFITS format was derived structurally from the ALMA-TI FITS raw data format, although a number of changes had to be made to accommodate the special needs of the IRAM and APEX telescopes. We would like to thank especially Robert Lucas, who is one of the authors of the ALMA-TI format, for his valuable contributions to our discussions.

The MBFITS format is based on the scan-subscan-integration scheme used by ALMA-TI FITS and retains many of its keywords. However, due to the changes in structure and additional keywords needed to accommodate single-dish configurations, particularly multiple beam observing and multiple frontend/backend combinations, the MBFITS format can now be considered to be an independent format.

In the subsequent sections we outline the updates in all the versions of MBFITS since release 1.3. In Sect. 15, various aspects of MBFITS are described in detail. Then follows the specification of the FITS tables (Sect. 17). Finally, we include references (Sect.16).

1.2 Scans, subscans and integrations

Extracted from the ALMA Software Glossary (via ALMA-TI FITS definition, Lucas & Glendenning 2001):

dump The smallest interval of time for which a set of correlated data can be accumulated and output from the correlator.

integration A set of dumps, all identical in configuration (except for the antenna motion and some others), that is accumulated and forms the basic recorded unit.

subscan A set of integrations while the antennas complete an elemental pattern across the source, possibly while frequency switching, nutator switching, etc. (previous to v.1.54: observation)

scan A set of subscans with a common goal, for example, a pointing scan, a focus scan, or an atmospheric amplitude calibration scan, or a correlation scan on a continuum source or a line source.

For instance in the case of holography measurements a subscan would be a drift across the transmitter or a bore-sight measurement, while a scan could be the whole set of subscans needed to acquire a beam map. Or a scan could be a pointing scan with two subscans (an azimuth drift and an elevation drift across the pointing calibrator) or an atmospheric calibration scan with three subscans (autocorrelations on the sky, and two loads at different temperatures, ...).

A scan can be as simple as a short integration on a celestial source while total power and/or correlator output are recorded; or it could be a set of pointed subscans that are used together to form a map of an extended celestial source.

Here are some examples of how this scheme works for single dish observing.
Examples of a scan:

- An on-the-fly map of an astronomical source, including associated sky off subscans
- A raster map . . .
- A pointing scan (cross-raster or cross-OTF)
- A focus scan
- A skydip
- A flux calibration measurement
- Five on-source measurements forming a cross and a sky off position

- A holography measurement

…and a subscan:

- A line of an OTF map
- One sample in a raster
- A sample on a sky off position
- A heterodyne calibration (HOT/COLD/SKY)
- One step in a focus scan

2 What's new in v.1.63?

MBFITS 1.63 is a maintenance release to define additional spectral axis descriptions in the ARRAYDATA-MBFITS table header to detail source rest and telescope reference system frequency axes. In addition, a couple of optional keywords are introduced in other tables.

2.1 Spectral axis descriptions

To observe a given rest frequency $f_{rest,0}$ in a celestial object, one must correct for the Doppler shift due to earth rotation and orbital motion plus the source radial velocity relative to the solar system. The receiver is tuned to the resulting "sky frequency"

$$f_{sky} = f_{rest,0} \cdot k_{Doppler}$$

This correction is made for one point in the frequency axis usually at the center of the band, i.e. its source rest frequency is known a priori. Any other source rest frequency must be calculated using the Doppler factor since the receiver records the transformed frequency range. A signal at an offset Δf (measured in the telescope system) from $f_{rest,0}$ in the spectrometer actually comes from a source rest frequency at

$$f_{rest,\Delta f} = f_{rest,0} + \Delta f / k_{Doppler}$$

This means that the channel width in the resulting spectrum is compressed or stretched compared to the lab values. Within a band this effect is usually neglected since the bandwidth used to be too small to show noticeable effects. With ever increasing receiver bandwidths this becomes a problem that needs to be addressed.

The Doppler shifted frequency axis also means that the rest frequency at the center of the opposite sideband (OSB) of 2SB receivers must be calculated as

$$f_{rest,OSB,0} = f_{rest,0} \pm \Delta f_{SBsep.} / k_{Doppler}$$

One consequence of the above is the fact that the spectral channel width and the OSB rest frequency will change with time.

A caveat about atmosphere model corrections is that they need to be made using the sky frequencies rather than the rest frequencies since these effects occur in the telescope rest frame.

A caveat about extra-galactic observations is that the usual approach to have the observer calculate a virtual rest frequency and use a source radial velocity of 0 km/s would still lead to wrong frequency axes in the signal band and wrong rest frequencies and axes in the OSB since the real Doppler factor is lost.

Up to v.1.62 the MBFITS format definition of the ARRAYDATA-MBFITS table frequency axis specified only a signal sideband source rest frequency and a radial velocity description. In addition, there were keywords for the central sky frequency, the bandwidth, and the sideband separation in the telescope reference system. While this is sufficient to calculate everything needed for the above use case, it is a bit awkward since the information is scattered in various disconnected keywords. We thus introduce three new axis descriptions to cover the following cases explicitly:

- Source rest frequency in the signal band (WCS version code `F`)
- Source rest frequency in the image band (WCS version code `I`)
- Telescope (sky) frequency in the signal band (WCS version code `S`)
- Telescope (sky) frequency in the image band (WCS version code `J`)
- Source radial velocity in the rest frame (e.g. LSRK) (WCS version code `R`)

See detailed keyword names and description in section 17.5.

Note that a related discussion item about the source velocity reference frame type (RADIO, OPTICAL, RELATIVISTIC, etc.) is already covered by the `1CTYP2R` keyword which can take values `VRAD`, `VOPT`, etc. (see Greisen et al. 2006).

2.2 Keyword changes

2.2.1 SCAN-MBFITS **table**

There was an inconsistency in the coordinate system setup as introduced in MBFITS v.1.62. We had added `CTYPE1B` and `CTYPE2B` to describe the basis coordinate system and projection. The source coordinates would thus have had to be written to keywords `CRVAL1B` and `CRVAL2B` while we kept the default names `CRVAL1` and `CRVAL2`. To remedy this situation, we revert to letting `CTYPE1` and `CTYPE2` describe the basis system as it was in MBFITS v.1.61 and earlier. Instead we introduce two new keywords **CTYPE1N** and **CTYPE2N** to describe the user native frame system and projection. The keywords ending in "B" are removed.

Some optional higher order pointing coefficients were added to the existing pointing constants.

2.2.2 FEBEPAR-MBFITS **table**

The `DEWCABIN` keyword length has been extended to 70A to optionally include more details of the receiver cabin setup. Two optional keywords (**LOCATION** and **OPTPATH**) have been introduced to store details of the location of the receiver within the cabin and about the optical path.

The new optional keyword **NOPTREFL** stores the number of optical reflections for a given receiver. This can be used to calculate the feed derotation.

Complementary higher order receiver pointing coefficients were added.

Corrected some definitions of character keywords that were defined to `80A` which is not allowed by the FITS standard.

2.2.3 DATAPAR-MBFITS **table**

The `OBSTYPE` has been deprecated. It is to be replaced with the newly introduced **SUBSTYPE** keyword. The new optional **CTYPE1N** and **CTYPE2N** keywords are to be written to DATAPAR-MBFITS if the value for the current subscan differs from the SCAN header entry (e.g. for horizontal reference points in an equatorial map).

2.3 Additional Effelsberg keywords

2.3.1 SCAN-MBFITS **table**

The Effelsberg control system allows for harmonic oscillation of the focus position to reduce standing waves. A number of optional keywords have been added to describe such a setup.

3 What's new in v.1.62?

MBFITS 1.62 is a maintenance release to add some useful columns in the DATAPAR-MBFITS table, to clarify the coordinate system and the scan setup keywords and to document some features that were omitted in previous releases but were actually already in use at APEX. In addition, there are a few minor changes for some optional keywords.

3.1 Projection used in APEX MBFITS data

The MBFITS documentation has always suggested to use the Sanson-Flamsteed (SFL) projection as the default for single-dish radio astronomy data. For APEX data we now realized that we have used the Global Sinusoidal (GLS) projection for the `LONGOFF` and `LATOFF` columns in the DATAPAR-MBFITS table. This was done to be more compatible with the Gildas software that is used for subsequent spectral line data reductions. Many bolometer data reduction packages also inherently assume this kind of projection. The GLS projection is very similar though not identical to SFL. Originally, it was described in AIPS memo 46 by E. Greisen. The difference is that the cosine is applied to the latitude rather than the delta in latitude:

GLS projection:

$$\begin{aligned} x &= (\phi - \phi_0)\cos\theta \\ y &= (\theta - \theta_0) \end{aligned} \tag{1}$$

SFL projection:

$$\begin{aligned} x &= (\phi - \phi_0)\cos(\theta - \theta_0) \\ y &= (\theta - \theta_0) \end{aligned} \tag{2}$$

The SFL projection can always be re-constructed from the GLS one using existing MBFITS keywords and columns. For APEX we thus will keep using GLS. The header keywords `CTYPE1` and `CTYPE2` have been corrected accordingly in the APEX MBFITS writer. The MBFITS format description now allows to use both projection types.

3.2 Table changes

The recent addition of drift scans for the Sunyaev-Zel'dovich project at APEX showed some shortcomings of the DATAPAR-MBFITS table structure. In those drift scans, the telescope is moved along lines of constant elevation to reduce the effect of varying atmospheric emission. The patterns need to be set up with horizontal basis and user native frames and thus all references to astronomical coordinates are lost. Since the DATAPAR-MBFITS table is meant to facilitate the subsequent data reduction, the SZ group asked for the addition of the current mean RA and Dec as well as the parallactic angle.

The Laboca group asked for the interpolated wobbler positions to be stored in the DATAPAR-MBFITS table. We added those as optional columns. The APEX MBFITS writer does write them.

In the FEBEPAR-MBFITS table description we corrected the type of the `FEEDTYPE` column to be 1PJ instead of 1PA. The `FDTYPCOD` keyword content changes accordingly, i.e. entries are indexed with numbers starting at "1" (see section 15.5.10). This change had already been introduced at APEX with MBFITS 1.55 but was not documented properly. The change was necessary to avoid problems with long variable length string vectors that caused segmentation faults in the standard cfitsio library and the FITS viewer at the time.

3.3 Keyword changes

The keywords describing the coordinate systems have led to some confusion among MBFITS users. The `CTYPE1` and `CTYPE2` keywords were referring to the basis frame while they actually describe the user native frame. This has been corrected in the documentation. In order to describe the basis frame setup, two new keywords (`CTYPE1B` and `CTYPE2B`) have been added. The naming follows WCS conventions to add a single letter (here "B" for "basis frame") to the base name.

Some MBFITS users have commented on the possible values listed for the `SCANTYPE` keyword. There were, indeed, confusions with the `SCANMODE` keyword that was added later. While the listed values are only recommendations, we felt that we should correct these lists to make them self-consistent. We thus removed a few options (OTF, RASTER, CROSS, etc.) which belong to `SCANMODE` and `SCANGEOM` instead.

We also added a few new scan types for focus scans. Due to the previous lack of such a recommendation, the Effelsberg group began coding their focus scans in the SCANDIR keyword. Ideally, this should not be done since SCANDIR refers to sky coordinate systems while focus scans are being performed in the subreflector coordinate system, i.e. a telescope reference frame.
The Effelsberg group has asked to add some keywords to store the azimuth encoder zero correction (PDELTAIA and FDELTAIA) and the pointing hysteresis (HYSA and HYSE (both optional)). The length of the Effelsberg phase combination description keywords (SIG_ONLN, etc.) needed to be extended to 80A to be able to store the typical formulae.
The OBSNUM has been deprecated and should not be used for new data sets anymore. The subscan number was originally named *observation* number in ALMA-TI-FITS and thus in MBFITS. However, nobody really liked the name and eventually ALMA renamed *observation* to *subscan* and MBFITS did too. The old keyword was kept for backward compatibility but it will be phased out at some point.
Since MBFITS 1.57 the phase descriptions are stored as PHASEn keywords in the SCAN-MBFITS header. The documentation omitted that those keywords may change from subscan to subscan (e.g. for symmetric wobbler modes where the reference and signal beams alternate). Like for the SCANxxxx keywords, the PHASEn keywords are then written to the DATAPAR-MBFITS header if their values differ from the one in the SCAN-MBFITS header.

3.4 MBFITS is a Registered Convention

Since July 2007 the MBFITS format description is an officially registered FITS convention. The Registry of FITS Conventions (http://fits.gsfc.nasa.gov/fits_registry.html) provides a central and authoritative repository for documenting conventions that have been developed by the FITS user community for storing and transmitting various types of information in FITS format data files.

4 What's new in v.1.61?

4.1 Az/El patterns

Rectified the content of the `DATAPAR::LONGOFF` and `DATAPAR::LATOFF` columns for az/el patterns on a native user frame offset position (see section 15.5.7). Originally, the offsets needed to be stored in the native celestial frame (e.g. equatorial) although the scanning was performed in horizontal coordinates. The reasoning was that the offset position would otherwise be lost. However, the subsequent calibration and reduction software usually needs rather the horizontal offsets (e.g. for beam maps). Therefore, we changed the scheme to always store the offsets in the horizontal system for those modes and to store the pattern center as two extra keywords (**PATLONG** and **PATLAT**) in the SCAN table.

4.2 Keywords

The keywords for bolometer frontend and backend gains have been iterated during the first LABOCA run at APEX. The `BOLREFGN` column in FEBEPAR has been renamed **BEGAIN** and a new scalar frontend gain keyword (**FEGAIN**) was added to the FEBEPAR header.
During the CHAMP+ installation, the dewar rotation keywords were used for the first time at APEX. A few changes were necessary to represent all information. First, we renamed the `DEWANG` keyword in the FEBEPAR table to `DEWUSER` since it is meant to represent the user defined angle of the array relative to the frame that is used. A new keyword `DEWZERO` was needed to describe the offset of the array coordinate system relative to the cabin / telescope systems. In the DATAPAR table we added two new keywords. `DEWANG` describes the actual position of the dewar for this subscan and `DEWEXTRA` is an extra angle applied by the control system based on the array symmetry to reduce the necessary dewar motions. All those numbers are needed to calculate the pixel offsets on sky.
The wobbler observing modes required a new SCAN keyword called **WOBPATT** to store the wobbler pattern, i.e. the sequence of wobbler offset directions throughout the scan (e.g. "NEG(ative)", "POS(itive)", "SYM(metric)"). This information is needed by the calibration programs to assign the correct offsets and possibly add data from several subscans in symmetric mode.
The general (APEX) and ESO project ID keyword lenthgs were increased to be able to save the ESO run ID character.

5 What's new in v.1.6?

5.1 MBFITS Hierachical Grouping Structure

The original MBFITS format was intended to be written as one file in order to facilitate transporting the data from the observatory to the users. However, the FITS standard requires that the binary tables be written sequentially in the file. When writing an MBFITS file during the actual observations, the underlying library (e.g. CFITSIO in case of APEX) thus always needs to re-arrange the tables to make room for the new incoming data.
The alternatives of keeping the data in memory or in temporary files before writing is impractiable because of the final sizes of the files for long (>100 seconds) subscans which occur for wobbler or frequency switched scans or for long bolometer maps. Copying at the end would introduce a large overhead and delay the online calibration and display.
To remedy this problem, the FITS hierarchical grouping standard (see Jennings et al. 1997) is employed. In this case, each MBFITS binary table is stored in a separate file in a directory tree. This removes the problems of re-arranging the data since each file can now simply be written sequentially. A new table called GROUPING is needed to store the locations of the member files so that the data set can be accesses as one entity. The CFITSIO library and the FitsView application support this standard and thus allow handling this data structure. In addition to the file locations we have defined a few columns to allow speeding up searching when reading the MBFITS dataset for further processing. The GROUPING table structure is defined in section 17.8. The primary header of the GROUPING file is used as the "Primary header" as defined in section 17.2. The primary headers of the other files contain only minimal data (see section 17.8.3) to allow identification of the MBFITS dataset to which the file is related.

5.1.1 MBFITS Hierarchical Grouping Directory Structure

The MBFITS hierarchical grouping directory structure is defined as follows:

- Main directory name according to the `PROJID` keyword. Inside this main directory there are the files for the scan-level tables:
  - The grouping table file: `GROUPING.fits`
  - The scan table file: `SCAN.fits`
  - The FEBEPAR table files for each FEBE combination: `<FEBE name>-FEBEPAR.fits`
- The actual data is stored in subdirectories per subscan named according to the subscan number. Each subdirectory contains the following types of member files:
  - The MONITOR table file: `MONITOR.fits`
  - The ARRAYDATA table files for each FEBE combination and baseband: `<FEBE name>-ARRAYDATA-<Baseband number>.fits`
  - The DATAPAR table files for each FEBE combination: `<FEBE name>-DATAPAR.fits`

An example is the following calibration scan (3 subscans: SKY/HOT/COLD) performed with the FLASH460 and FLASH810 receivers and the FFTS1 and FFTS2 spectrometers at the APEX telescope:

```
APEX-81261-M-77.C-0014-2006
APEX-81261-M-77.C-0014-2006/GROUPING.fits
APEX-81261-M-77.C-0014-2006/SCAN.fits
APEX-81261-M-77.C-0014-2006/FLASH460-FFTS1-FEBEPAR.fits
APEX-81261-M-77.C-0014-2006/FLASH810-FFTS2-FEBEPAR.fits
APEX-81261-M-77.C-0014-2006/1/MONITOR.fits
APEX-81261-M-77.C-0014-2006/1/FLASH460-FFTS1-ARRAYDATA-1.fits
APEX-81261-M-77.C-0014-2006/1/FLASH460-FFTS1-ARRAYDATA-2.fits
APEX-81261-M-77.C-0014-2006/1/FLASH460-FFTS1-DATAPAR.fits
APEX-81261-M-77.C-0014-2006/1/FLASH810-FFTS2-ARRAYDATA-1.fits
APEX-81261-M-77.C-0014-2006/1/FLASH810-FFTS2-ARRAYDATA-2.fits
APEX-81261-M-77.C-0014-2006/1/FLASH810-FFTS2-DATAPAR.fits
APEX-81261-M-77.C-0014-2006/2/MONITOR.fits
APEX-81261-M-77.C-0014-2006/2/FLASH460-FFTS1-ARRAYDATA-1.fits
```

```
APEX-81261-M-77.C-0014-2006/2/FLASH460-FFTS1-ARRAYDATA-2.fits
APEX-81261-M-77.C-0014-2006/2/FLASH460-FFTS1-DATAPAR.fits
APEX-81261-M-77.C-0014-2006/2/FLASH810-FFTS2-ARRAYDATA-1.fits
APEX-81261-M-77.C-0014-2006/2/FLASH810-FFTS2-ARRAYDATA-2.fits
APEX-81261-M-77.C-0014-2006/2/FLASH810-FFTS2-DATAPAR.fits
APEX-81261-M-77.C-0014-2006/3/MONITOR.fits
APEX-81261-M-77.C-0014-2006/3/FLASH460-FFTS1-ARRAYDATA-1.fits
APEX-81261-M-77.C-0014-2006/3/FLASH460-FFTS1-ARRAYDATA-2.fits
APEX-81261-M-77.C-0014-2006/3/FLASH460-FFTS1-DATAPAR.fits
APEX-81261-M-77.C-0014-2006/3/FLASH810-FFTS2-ARRAYDATA-1.fits
APEX-81261-M-77.C-0014-2006/3/FLASH810-FFTS2-ARRAYDATA-2.fits
APEX-81261-M-77.C-0014-2006/3/FLASH810-FFTS2-DATAPAR.fits
```

The hierarchical grouping setup is to be viewed as an alternative way of storing MBFITS data. The actual content of the MBFITS binary tables has not changed. The single file approach is thus still valid, although it is not recommended for the above reasons.

5.2 Keywords

The throws for frequency switching can be asymmetric. Thus the original single keyword FEBEPAR-MBFITS::FRQTHROW has been replaced with two new keywords called FRTHRWLO and **FRTHRWHI** denoting the throws for the two phases. As before they are given for the radio frequency axis.

To store the image frequency in the CLASS format one needs the sky frequency that includes the total velocity correction (earth, orbit and source) at the observing time. This information was not yet included in the MBFITS format. Thus the new keyword ARRAYDATA::**SKYFREQ** has been introduced to store the line sky frequency for each baseband.

The lengths of the **ORIGFILE**, **OBSID** and **PROJID** keywords have been increased to allow storing the typical contents that occur for the APEX telescope.

6 What's new in v.1.57?

Version 1.57 introduces a few additional keywords and some changes in the DATAPAR-MBFITS binary table structure to enhance the table creation efficiency.

6.1 Switched observations

Until MBFITS 1.56, the phase information was stored as the `ISWITCH` ASCII column in the DATAPAR-MBFITS binary table while all other columns have numeric types. This is very inefficient when creating these tables. Therefore the `ISWITCH` column type was changed to `1J` and it was renamed to `PHASE` indicating the we store the 1-based phase number. The descriptions are now stored as keywords `PHASEn` in the SCAN-MBFITS header to describe the phase number states.

6.2 Array observing

For array observations it is necessary to record the possible dewar coordinate system zero offset angle due to mechanical alignment errors. This is stored in the FEBEPAR-MBFITS header keyword `DEWZERO`.

6.3 Other changes

Split columns `DFOCUS` and `DPHI` into three columns each (`DFOCUS_X`, `DFOCUS_Y`, `DFOCUS_Z` and `DPHI_X`, `DPHI_Y`, `DPHI_Z`) to allow writing DATAPAR-MBFITS from a simple 2D data structure in MBFITS writing software.

7 What's new in v.1.56?

7.1 Primary header

In order to allow full search possibilities in the ESO archive, we have extended the description of the instrument, which was previously only contained in the INSTRUME keyword. In the ESO archive this keyword can be at most 10 characters long and preferably shorter. In order to record details of all used FEBEs we add 4 new keywords per FEBE used: FEBEn, FREQn, LINEn, BWIDn, where n is an integer running from 1 to the number of FEBE combinations used. The INSTRUME keyword now codes whether the FEBEs are heterodyne or bolometer - e.g. using APEXHET or APEXBOL.

Have added the 'ESO Data Interface Control Document' to the list of references - this describes the required format for ESO files (including the HIERARCHICAL header keywords).

Currently the full project ID (including the project origin) is only recorded in the filename and in the SCAN header. As the filenames will be changed in the ESO archive, we have added the full project ID to the primary header.

7.2 Other minor changes

The header keyword describing the system of the user native frame, USRFRAME, has been moved from the SCAN header to the MONITOR header. This is because the description can change with subscan. For example, references made with horizontal offset for an equatorial source have USRFRAME equal to EQEQHO, whereas the corresponding ON measurement has EQEQEQ.

The system of recording pointing model information has been updated to make it easier to distinguish the different contributions to the poining offsets:

- The base pointing model is stored in the SCAN header (e.g. the model based on optical pointing)
- Receiver dependent corrections to this model are stored in the FEBEPAR header
- Corrections to the pointing in IE and CA by changing the focus are stored in the SCAN header in FDELTAIE and FDELTACA
- Accumulated user determined pointing corrections to IE and CA (from the pointing source measurements in the current session) are stored in the SCAN header in PDELTAIE and PDELTACA

The descriptions of monitor points describing focus offsets have been clarified:

- DFOCUS_X_Y_Z: Focus delta offsets for the subscan X, Y, Z (e.g. during a focus scan or after focus change by the observer)
- FOCOBS_X_Y_Z: Receiver dependent X, Y, Z focus offsets
- FOCUS_X_Y_Z: Focus (absolute subreflector position) X, Y, Z (including elevation dependent correction, receiver dependent offsets, and subscan dependent delta offsets)

Monitor point THOTCOLD_<febe> changed to THOTCOLD_<cabin> (the measured load temperatures in that cabin) and TCHOTCOLD_<febe> changed to THOTCOLD_<febe>_<band> (the corrected load temperatures at the frequency of each FEBE/Baseband combination).

8 What's new in v.1.55?

8.1 Feeds and backend 'sections'

In order to make a clear distinction between the different inputs of a backend (sometimes called backend channels) and the spectral channels used for heterodyne observations, we now refer to each physical input of a backend as a 'section'. The keyword `CHANNELS` in the ARRAYDATA header is reserved for spectral channels and so should be set to 1 for continuum observations. Groups of backend sections that share the same frequency setup are called basebands (`BASEBAND`), with the data for each baseband written to a single ARRAYDATA table.
To allow all combinations of setups for feeds and backend sections in the FEBEPAR table, the `NUSEFEED` header keyword is moved into the binary part and now contains the number of used feeds for each used baseband - this value is repeated in each ARRAYDATA header to allow dimensioning the `DATA` array. The overall arrangement of the backend/feed description in FEBEPAR is now as follows (further explanation and an example diagram are given in Section 15.5.10):

- **FEBEFEED** is the total number of physical feeds available from the frontend (N_{FD}) - unchanged
- **FEBEBAND** is the maximum number of basebands for the backend (this is the maximum number of configurable frequency setups for the backend) - unchanged
- **NUSEBAND** is the number of basebands in use (each baseband consists of one or more backend sections, each connected to one feed) - unchanged
- **FDTYPCOD** gives a definition of the 1-character type code used to describe each feed (80A) - added e.g. `'A:AMPLITUDE P:PHASE'`
- **NUSEFEED** is the number of feeds in use for each baseband - moved from header to table. Now has dimension N_{USBD}
- **USEFEED** is a list of the feed numbers associated with each baseband - extra dimension added - now $N_{\mathrm{USFD}} \times N_{\mathrm{USBD}}$
- **BESECTS** is a list of the backend section numbers connected to the feeds for each baseband (in the same order as listed in `USEFEED`) - renamed (was `BECHANNS`) and extra dimension added - now $N_{\mathrm{FD}} \times N_{\mathrm{USBD}}$
- **FEEDTYPE** Single character code to describe the type of each used feed (string length $N_{\mathrm{USFD}} \times N_{\mathrm{USBD}}$)

8.2 Frequency scale axis

A number of updates/corrections to the names of keywords in the ARRAYDATA header so that all conform to the standard FITS definition (see Greisen et al. 2004). The axis description keywords now all follow the following format:
i`CTYP`na
where i is the axis number (for ARRAYDATA the table format is $N_{\mathrm{CH}} \times N_{\mathrm{USFD}}$- therefore $i = 1$ for the channel axis and $i = 2$ for the feed axis), n is the column in the binary table (=2 for ARRAYDATA) and a is a 1-character coordinate version code ('F' for frequency, 'R' for radio velocity).
The number of IF conversions, which is needed to determine the direction of the frequency axis is added as a possible monitor point.

8.3 Observation → Subscan

The change in nomenclature 'Observation' → 'Subscan' from MBFITS v.1.54 was never implemented in the keywords. Therefore, to enable a gradual change in existing software, extra header keywords **SUBSNUM** and **NSUBS** are added alongside `OBSNUM` and `NOBS`. Eventually these keywords referring to observations will be phased out.

8.4 Additional Effelsberg changes

- SIG_ONLN (FEBEPAR header) - optional descriptions of how to recombine different phases
- REF_ONLN (FEBEPAR header)
- SIG_POL (FEBEPAR header)
- REF_POL (FEBEPAR header)
- TBLANK (FEBEPAR header) - blank time of backend
- TSYNC (FEBEPAR header) - sync. time of backend
- TCAL (FEBEPAR binary table) - radio calibration factor
- NINTS - moved from DATAPAR table to DATAPAR header: number integrations in block (must be same number of integrations in each block)
- INTEGNUM (DATAPAR binary table) - no longer needed: can calculate it from NINTS
- ISWITCH (DATAPAR binary table) - new dimension: now for each integration in block

8.5 Minor changes

- TRANDIST (SCAN) type changed from integer to double.
- TRANFREQ (SCAN) type changed from integer to double and units from m to Hz.
- TRANFOCU (SCAN) type changed from integer to double and units from m to deg.
- Extra pointing coefficient keywords added in SCAN describing the ALMA pointing model - see Section 15.5.8.
- BOLDCOFF (FEBEPAR) added to contain the bolometer DC offset for each feed.

9 What's new in v.1.54?

9.1 Array rotation: PC matrix, CRPIX and parallactic angle

The PC matrix and CRPIX in DATAPAR are not useful and are taken out.
They are not useful because the frame transform between array offset coordinates $(p1, p2)$ and (x, y) user projection frame coordinates cannot simply be described by an offset and rotation. The transformation additionally involves stretch and skew.
Instead, the rotation angle ROTANGLE of the array offsets frame with respect to the user native frame (ϕ, θ) is given. Note that this is the angle between two spherical coordinate systems, not in the projection frame. With this angle, it is easy to work out (ϕ, θ) and thus (x, y) for each array offset from FEEDOFFX, FEEDOFFY and the array centre (ϕ, θ). (ϕ, θ) for the array centre can be calculated from (LONGOFF,LATOFF) or (BASLONG,BASLAT) pairs. See sect. 15.5 for documentation on coordinate frames.
Fig. 15.5.3 showing the relationship between the various coordinate systems has been revised.

9.2 Observation → Subscan

We follow ALMA in changing the ambiguous nomenclature "Observation" to the more easily understood "Subscan" officially endorsed by the SSR in their meeting of 2004-05-07. As an example we give the new definitions for a scan and a subscan:

- Scan:

  The scan is the lowest level object normally used by an observer. It is a sequence of one or more *subscans* that share a single goal: for instance pointing and focus scans involve a pattern of *subscans*. Whether OTF mosaicing observations are considered a single scan or a scan per point is rather a matter of how you would like to define it.

- *Subscan*:

  A *subscan* is the minimal amount of data taking that can be commanded at the script language level. It is highly desirable that it should be a simple enough element so that the script language may be used to define the content of scans (at the staff member/expert level), as a means to develop and debug new observing modes.

9.3 Minor changes

- WIND_SPEED_DIR (MONITOR) units for speed changed from km/s to m/s.
- DEWANG (FEBEPAR) is now a real rather than an integer type.
- RADECSYS/RADESYS: RADESYS without a "C" to conform to the WCS standard appears in SCAN. This was mistakenly changed to include a "C" and has been changed back. RADECSYS to conform to the ESO standard appears in the primary header.
- CTYPE option for hour angle/ declination added: HA/DEC. WCSNAME options updated. See Sect. 15.5.5.
- BECHANNS A list of backend channel numbers connected to feeds in same order as listed in USEFEED / data storage in ARRAYDATA.

9.4 Additional Effelsberg parameters

Additional keywords in use at Effelsberg:
SCANROT OBSINP SK0P; rotation of user frame meridian wrt basis meridian, measured positive E of N.
(this list is not exhaustive).

10 What's new in v.1.53?

10.1 Revised documentation

Revised documentation on coordinate systems in 15.5.

10.2 New keywords

Several new keywords added to store previously missing information:

- **BOLREFGN** reinstated in the FEBEPAR header. This was changed into `BOLCALFC` but is now back again. Integer scale factor which works as follows:

  $$\text{ADCs dynamic range} = (\pm 10 \text{ Volts})/g$$

  For example:

g	Dynamic range
1	±10 Volt
2	±5 V
10	±1 V
100	±100 mV

  When the signals exceed the limits given by g the acquired signals will be saturated.
- **WCSNAME** included for a human-readable description of the astronomical basis frame.
- **DIAMETER** dish diameter included with the observatory parameters in SCAN header.
- **WOBCOORD** (DATAPAR header) flags if wobbler offsets have been included in the astronomical coordinates calculation (true) or not (dish positions only; false).
- **DEWCABIN** (FEBEPAR header) describes whether the dewar is in the `NASMYTH_A`, `NASMYTH_B` or `CASS_C` cabin, important for the dewar rotation.
- No new keywords are introduced to describe beam switching at the 30m (moving the beam with a chopping mirror). In the understanding that this will not be used simultaneously with the wobbler, the wobbler parameters (`WOBTHROW`, `WOBCYCLE` and `WOBDIR` can be used for beam switching if needed).

11 What's new in v.1.52?

11.1 Types in ESO header

Several of the entry types for ESO header keywords have been changed from 30A to numerical types J/E/D as appropriate.

11.2 Times for integrations

To bring the MBFITS standard into agreement with what is being used in practice (Effelsberg, APEX receiver development) and thus clear up confusion:

- `MJD` in ARRAYDATA is the time at the **m**idpoint of the integration
- `MJD` in DATAPAR is also the time at the **m**idpoint of the integration
- `MIDTIME` in DATAPAR is removed (use `MJD` in DATAPAR)
- `LST` in DATAPAR is also given at the midpoint of the integration.
- `INTEGTIM` (DATAPAR) as always gives the integration time, from which the start and end of the integration can be calculated.

11.3 Frequency bands

`FEBEBAND` (ARRAYDATA) is being used as a descriptive label of the frequency band relating to the backend setup, rather than a simple counter. In order to know which bands are in use and therefore how to index the arrays of calibration parameters in the FEBEPAR table, the array `USEBAND` in FEBEPAR table has been introduced. This lists the basebands in use. The number of bands in use is given in `NUSEBAND` (FEBEPAR header). The total number (maximum number) of available bands for this FEBE is given in `FEBEBAND` (FEBEPAR header). This has to be greater or equal to the highest number listed in `USEBAND`. This system is similar to that already in place in MBFITS for feed numbering.
An example:

```
FEBEPAR header:
  FEBEBAND = 12 # total number of available bands
  NUSEBAND = 3 # number of bands in use for this scan
FEBEPAR table:
  USEBAND = [1,3,10]   # list of bands in use for this scan
ARRAYDATA 1:
  BASEBAND = 1
ARRAYDATA 2:
  BASEBAND = 3
ARRAYDATA 3:
  BASEBAND = 10
```

Unlike the feed case, the arrays in the FEBEPAR table are then dimensioned by N_{USFD} rather than by N_{FD}.

12 What's new in v.1.51?

12.1 FOCUS keywords

For focus scans only (SCANTYPE = FOCUS), the focus offsets and rotations (X,Y and Z axes) are stored in DATAPAR as DFOCUS, DPHI. These are interpolated from the MONITOR keywords DFOCUS_X_Y_Z, DPHI_X_Y_Z. The absolute focus positions are also stored in MONITOR in FOCUS_X_Y_Z, PHI_X_Y_Z.
The observer's chosen focus (set after a focus scan) is stored at the beginning of each observation in FOCOBS_X_Y_Z (MONITOR). An extra MONITOR point has been added to store the absolute subreflector rotations PHIOBS_X_Y_Z.

12.2 ARRAYDATA to DATAPAR moves.

INTEGTIM and ISWITCH have moved back to DATAPAR. The overheads in ARRAYDATA were too great when writing out several bands from one continuum backend, with one ARRAYDATA for each band. The move of these keywords back to DATAPAR prohibits blocking of DATAPAR entries if ISWITCH or INTEGTIM changes between entries. The blocking parameters are retained for e.g. continuum fast scanning where INTEGTIM and ISWITCH do not change but the data dump rate is high. INTEGNUM is no longer explicitly given in ARRAYDATA, though it remains in DATAPAR. In ARRAYDATA, the integration number can be taken to be the row number (also to reduce overheads in ARRAYDATA). If the case arises that a backend dumps data at different rates in different bands, these can be defined for the purposes of MBFITS as separate backends.

12.3 MJD, DATE-OBS

MJD, DATE-OBS and MJD-OBS are given in time system TIMESYS. This has been made clear in the comments. At APEX, the TIMESYS(SCAN) will be TAI. The choice of TAI is for the following reasons:

- TAI is the original source of time at APEX
- TAI is monotonically increasing. This is necessary for ordering of events, particularly in the MONITOR table, and so that we can sensibly plot other quantities against it. In contrast, UTC has leap seconds and UT1 if calculated from UTC has a varying offset UT1UTC.
- TAI increments at a fixed rate: it counts in SI seconds
- TAI can easily be converted to TT (terrestrial time) or UTC (coordinated universal time) as required

TIMESYS is also added to the Primary Header for ESO archiving.
(v.1.51: we step back from using MJDTAI to allow some flexibility to users other than APEX who may be based on other time systems).

12.4 Minor keyword changes

Some keywords have been renamed to conform with the 8-or-less-chars, no underscores rules:

`ELEVATION`	→ `ELEVATIO` (SCAN)
`NO_SWITCH`	→ `NPHASES` (FEBEPAR)
`N_OBS`	→ `NOBS` (SCAN)
`SCANTYPE`	has an extra option `ON` option `PSW` removed; use `ONOFF` instead.

`SWTCHMOD` can now take the value `NONE`, if no switching is active. If `SWTCHMOD`=`NONE` then `NPHASES`= 1.

13 What's new in v.1.4?

Keywords needed for the ESO archive of APEX data have been added to the primary header. These are optional (ie. only needed for APEX data going to the archive, not for 30m or Effelsberg data).
`HIERARCH ESO DPR CATG` and `HIERARCH ESO DPR TECH` replace `SCANCATG` and `SCANTECH` in the primary header.

14 What's new in v.1.3?

Keywords which are new, moved or altered since v.1.1 are given in bold in the tables.

1. **Minor updates to v.1.2**
   - Order of `DATA` in ARRAYDATA changed to (channels, feeds). First index should be the most rapidly varying (Cotton et al. 1995).
   - `BOLFLAT` → `FLATFIEL`
   - `OBSMODE` can be HOT,COLD,SKY if cals are split into multiple observations
   - `ISWITCH` and `INTEGTIM` moved to ARRAYDATA. This is to enable 4-phase observations from Effelsberg to be blocked in DATAPAR. This may cause difficulties in cases where `ISWITCH` does not come from the backend e.g. the value of `ISWITCH` depends on knowing e.g. the wobbler position – let me know if this is the case (JH).
   - `OBSTATUS`(DATAPAR header) flags aborted scans.
   - `ETUTC` and `GPSTAI` time system conversion constants in SCAN.
   - `SCANCATG` and `SCANTECH` added to primary header for ESO archiving and also for our own reference.
   - Units for MONITOR points are now stored in variable length character arrays. See description of MONITOR for details. If any of the units are unlikely, let me know (JH).
   - `11PC` – `22PC` are optional; can be left out for single feeds. CRPIX1 and CRPIX2 are *not* optional.
   - Apparent coords. recommended for moving bodies.
2. **Telescope-specific parameters**

   Telescope-specific parameters (e.g. the inputs to the telescope control system) can be stored in three ways:

   (a) As MONITOR points: MONITOR is very flexible in what it can store. See its description for details.

   (b) As extra keywords in the existing MBFITS binary table headers. These will simply be skipped by the data reduction software. Please inform the MBFITS working group of extra keywords added, so that they can be documented as (optional) entries in MBFITS.

   (c) Where many extra parameters are needed, write a separate binary table (or tables) with names that easily identifies them as telescope-specific e.g. '<telescope>-MBFITS'. The level (scan/obs/int) and format of these tables is left to the telescope staff. *Parameters stored in these tables are for information only. Information which is necessary for the data reduction should be in the specified MBFITS tables, where it will be considered by the calibration/reduction software.*
3. **The** INTMON **tables have been merged into** DATAPAR**.**

   There is now only one additional table, DATAPAR, associated with each ARRAYDATA data table. What previously were INTMON and DATAPAR have been combined. Originally, the two were separated because the information in DATAPAR can be written directly at the time of an integration, whereas the INTMON values are interpolated/calculated from information in MONITOR which is not available until the end of an observation. By buffering the DATAPAR entries until the end of the observation, when the MONITOR values are available, all the data-associated parameters can be written together, saving on duplication in table and header. This change forces the positions etc. which were previously in INTMON to be written in quasi-real-time rather than filled in later offline, but it has become clear that this is necessary anyway to maintain an uninterrupted data flow at the telescope.

4. **Variable length arrays in MONITOR**

   The MONITOR array, which stores real-time data at its natural rate, was limited by its format to storing single floating-point values. This is hugely restrictive when one wants to store e.g. 5 structural temperatures, a spectral line gain array, or total power measurements for 3 frequency bands. We have changed MONITOR to store variable length arrays instead of single floating point values.

   This would introduce additional overheads if the majority of entries were still single floating point values. This is not the case as most entries group naturally into small arrays e.g. 2 encoder readings, 2 temperatures and 3 powers from a calibration. We store one time, description and units for each array. Thus by changing to variable length arrays in fact we reduce the storage overheads.

   Variable length arrays in FITS are described in [9] Cotton, Tody & Pence 1995 Appendix A. The IAU FITS group has not yet voted to include this as part of the FITS standard though it is likely that it eventually will. The North American regional FITS group has accepted it and it is already in use in many FITS formats. It is supported by many FITS packages including CFITSIO and FitsView.

   The storage works like this: in the fourth column of MONITOR a pointer to the variable length data array is stored. The data are stored at the end of the table. See [9] for details.

   With variable length arrays, almost anything can be stored in MONITOR. This gives MBFITS the flexibility to cope with future unforeseen requirements without making changes to the existing table structure.

5. **Calibration**

   A close look has been taken at what's required in order to calibrate the data and where this should be stored; see 15.4 below.

6. **Scanning/mapping description**

   Keywords which describe scanning/mapping keywords have been updated following a work-through of examples. See 15.2.

7. **Phase coding**

   We have looked description of multiphase (switching) observations and describe schemes for some common cases in 15.3.

8. **Data blocking**

   Data blocking is now possible ie. several rows in the ARRAYDATA table can now be written for one set of DATAPAR. This is controlled by `DPBLOCK` (DATAPAR header), a flag set to true if blocking, and `NINTS`(DATAPAR table) which stores how many ARRAYDATA entries correspond to the DATAPAR entry.

   When blocking, all DATAPAR values refer to the first integration of the block. An integration for which no direct entry in DATAPAR exists should interpolate between times and angles from bracketing entries. If blocking, a DATAPAR entry should always be written for the first and last integration of each observation.

9. **Holography**

   The HOLODATA table has been removed. Holography data is now stored as any other data in ARRAYDATA. If `OBJECT=*HOLO*`, special holography keywords `TRANDIST` (transmitter distance), `TRANFREQ` (transmitter frequency) and `TRANFOCU` (transmitter offset from prime focus) are stored in the SCAN header.

10. **Keyword compatibility: dashes and underscores, and SDFITS**

    Unnecessary dashes and underscores in keywords have been purged. An exception is the unavoidable FITS keyword `DATE-OBS`. Underscores are used for clarity in the MONITOR point descriptions: these are not FITS keywords.

    Keyword clashes with SDFITS (Single Dish FITS) have been resolved and a couple of keywords added:

    - `OBSID` stores the observer's and operator's initials (scan header);
    - `OBS-LONG`, `OBS-LAT` and `OBS-ELEV` are now `SITELONG`, `SITELAT`, `SITEELEV` (scan header);
    - `MOLECULE` and `TRANSITI` (optionally) together contain the molecule and transition for the main spectral line

11. **MJD** (Modified Julian Date) is used more widely than DATE-OBS to mark the integrations. FITS readers such as fitsview need a numerical (rather than text-format) timestamp to display sequential data. See 15.1 below.

12. **Other keyword additions and alterations**

    Keywords which have been altered (new, name changed, or moved) are marked in bold. Includes:

    - `FRQTHROW` in the FEBEPAR header gives the frequency step for frequency switched observations.
    - `DEWRTMOD` (FEBEPAR) is one of `CABIN` = dewar rotation fixed in Nasmyth/Cassegrain system, `EQUA`: RA/DEC; or `HORIZ`: AZ/EL. `DEWANG` (FEBE) is measured counterclockwise (anticlockwise) from vertical in the `DEWRTMOD` system in which the dewar rotation is fixed. Therefore `DEWANG` is also fixed throughout a scan. The feed offsets `FEEDOFFX`, `FEEDOFFY` (FEBEPAR table) are measured in degrees in the system in which the dewar rotation remains fixed.

15 Explanatory notes

In this section we describe how various aspects of data storage are handled in MBFITS. In particular, these explanatory notes show where to find the keywords/table columns associated with one theme (e.g. positions, mapping parameters), as these can be scattered across the tables. Explanation of individual keywords and small keyword groups can be found in the introductions to the individual tables and as comments in the table descriptions. (sections 17.3–17.7).

15.1 MJD, DATE-OBS and time system

`MJD`, `DATE-OBS` and `MJD-OBS` are given in time system `TIMESYS`(SCAN and Primary Header). This has been made clear in the comments.
At APEX, the `TIMESYS`(SCAN) will be TAI. The choice of TAI is for the following reasons:

- TAI is the original source of time at APEX
- TAI is monotonically increasing. This is necessary for ordering of events, particularly in the MONITOR table, and so that we can sensibly plot other quantities against it. In contrast, UTC has leap seconds and UT1 if calculated from UTC has a varying offset `UT1UTC`.
- TAI increments at a fixed rate: it counts in SI seconds
- TAI can easily be converted to TT (terrestrial time) or UTC (coordinated universal time) as required

The date (integer) part of MJD must also be calculated in `TIMESYS`: one cannot mix e.g. a UTC date with a TAI time.
Explanatory notes on MJD from http://tycho.usno.navy.mil/mjd.html:
'NOTE: The MJD is always referred to as a time reckoned in Universal Time (UT) or the closely related Coordinated Universal Time (UTC) , International Atomic Time (TAI), or Terrestrial Dynamic Time (TDT). '
'We frequently make use of the Modified Julian Date (MJD), which is defined as MJD = JD - 2400000.5. An MJD day thus begins at midnight, civil date. Julian dates can be expressed in UT , TAI, TDT, etc. and so for precise applications the timescale should be specified, e.g. MJD 49135.3824 TAI.'
The following description of `DATE-OBS` as used in MBFITS is adapted from Bunclark & Rots 1996. For more details on the `DATE-OBS` format see that reference.

> The new format is a restricted subset of ISO-8601:
>
> `CCYY-MM-DDThh:mm:ss.ssss`[1]
>
> <CCYY> represents a calendar year, <MM> the ordinal number of a calendar month within the calendar year, and <DD> the ordinal number of a day within the calendar month. <hh> represents the hour in the day, <mm> the minutes, <ss[.s...]> the seconds. The value of the integer part of the seconds field is normally in the range [0..59] but may take the value 60, if the time scale is UTC, to indicate a leap second. The literal 'T' is the ISO 8601 time designator.
>
> There must be a 'T' time designator between the date and the time. The decimal point character is an ASCII full-stop (hexadecimal value 0x2E).

[1] APEX `DATE-OBS` requires the time field with a precision of 100μs or four decimal places in the 'seconds' field.

15.2 Scanning/mapping description

The movements which the observer intended the telescope to carry out during a scan are stored in the following scanning parameters, so that the mapping scheme can be envisaged by the recipient of the data/ deduced by the data reduction software.
Note that the scanning parameters are not needed for the data reduction, which requires only the absolute positions of the current optical axis(`LONGOFF`, `LATOFF`) and information on whether an integration is part of the scan or off-scan (stored in `ISWITCH` in the DATAPAR table). Thus the scan parameters are optional. From a quick map of the observed positions, the scan geometry can be reconstructed. **The data reduction software should not calculate positions based on this description but instead rely on the actual observed positions given in** DATAPAR**.**

At the highest level, SCANTYPE shows the type of astronomical observation: POINT, FOCUS, CAL, SKYDIP, ON, ONOFF, etc. Then two parameters define how the telescope moves during the scan: SCANMODE and SCANGEOM. SCANMODE describes the mapping mode (SAMPLE, RASTER, OTF) and SCANGEOM the geometry (LINE, CROSS, CIRC, ARC etc.).
Then follow a number of scan keywords (lengths, directions etc.). Which of these are needed depends on the type of SCANMODE:

SCANMODE	keywords required
SAMPLE	SCANRPTS
RASTER	SCANLINE, SCANRPTS, SCANXSPC, SCANYSPC, SCANLEN, SCANDIR, ZIGZAG, CROCYCLE
OTF	SCANLINE, SCANRPTS, SCANXVEL or SCANTIME, SCANYSPC, SCANLEN, SCANDIR, ZIGZAG, CROCYCLE

The keywords are:

SCANDIR	(optional) scan direction, described as: USER (user native frame) or xLON/xLAT as in CTYPEj (standard basis system), including ALON/ALAT for Az or El scanning.
SCANLINE	(optional) number of lines in a scan. Default 1.
SCANRPTS	(optional) number of repeats of each scan line. Default 1.
SCANLEN	(optional, OTF/RASTER) For OTF, the line length or turn angle (SCANGEOM=CIRCLE) in Deg; for RASTER, the number of samples in a line.
SCANXVEL	(optional, OTF) tracking rate along line (units depend on SCANMODE definition)
SCANTIME	(optional, OTF) time for one line
SCANXSPC	(optional, RASTER) step along line between samples
SCANYSPC	(optional, OTF/RASTER) step between scan/raster lines
SCANSKEW	(optional, OTF/RASTER) offset in scan direction between lines
SCANPAR1	(optional) spare scan parameter (for modes I haven't thought of)
SCANPAR2	(optional) another spare scan parameter
CROCYCLE	CAL/REF/ON loop string showing how often to go to CAL and REF. e.g. CROOCOO is a REF every four ONs and CAL every two ONs. See 30m NCS documentation. CAL/REF/ON are stored in OBSMODE per subscan.
ZIGZAG	(optional, OTF/RASTER) TRUE if alternate lines traced in opposite directions, FALSE if all lines traced in same direction

Most standard types of map can be coded in the SCAN header using these keywords. A few unusual types require one or more parameters to change during the scan, between subscans. In this case, the parameters which change are taken out of the SCAN header and appear in the DATAPAR header.
Jenny Hatchell has examples of how to code standard scans (on-off, fivepoint, rectangular raster map etc.) using these parameters.

15.3 Phase description for switched observations.

Switched observations include wobbler switching, frequency switching, 2-horn beam switch, load switching, and calibration observations. Which of these modes is active for each frontend is determined by SWTCHMOD (FEBEPAR), apart from wobbler switching which applies to all receivers and is flagged by WOBUSED (SCAN).
SWTCHMOD can take the values:

TOTP	total power
FSW	frequency switch
BEAMSW	beam switch (as 30m)
HORNSW	horn switch
LOADSW	switch between source and load
CAL	calibration cycle
NONE	no switching

Beam switching (as with the chopper mirror at the 30m) needs three additional parameters: throw, direction and chop rate/frequency. As this is a relatively uncommon mode and in the understanding that this will not be used simultaneously with the wobbler, the wobbler parameters (WOBTHROW, WOBCYCLE and WOBDIR can be used for beam switching if needed.

Wobbler switching is controlled by 5 parameters in the SCAN header: WOBUSED, WOBTHROW, WOBDIR, WOBCYCLE and WOBMODE.

WOBUSED	True if wobbler in use
WOBTHROW	wobbler throw in deg.
WOBDIR	wobbler throw direction - described as USER (user native frame) or xLAT or xLON, inc. ALON or ALAT for Az or El scanning
WOBCYCLE	wobbler period in seconds
WOBMODE	wobbler mode (SQUARE/TRIANGULAR).

Triangular switching is wobbling in a direction perpendicular to the scan direction, during OTF mapping. The wobbler movements can be deduced from plotting the positions of each integration (e.g. LATOFF and LONGOFF in DATAPAR).

WOBCOORD is a flag in DATAPAR which is true if wobbler offsets are included in the output astronomical coordinates.

Frequency switching: FRQTHROW (FEBEPAR header) gives the frequency step for frequency switched observations. LOFREQ<fe> (MONITOR) gives the LO frequencies from the frontend, one of which switches when frequency switching.

Phase numbers for switched observations are stored with each integration in the PHASE column (DATAPAR). Phase descriptions are stored in the SCAN header as keywords PHASEn with n being the 1-based phase number. Some common examples of phase descriptions are:

Type of switching	**PHASE1**	**PHASE2**	**PHASE3**	**PHASE4**
frequency switching	FLO	FHI		
wobbler (nutating subreflector)				
2 phases	ON or WNEG	OFF or WPOS		
4 phases (symm. switch)	LON	ROFF	LOFF	RON
2-horn beam switch				
2 phases	L	R		
4 phases (symm. switch)	LON	ROFF	LOFF	RON
load switching				
2 phases	SKY	LOAD		
calibration				
3 phases	HOT	COLD	SKY	
no switching				
1 phase	NONE			

Users of more complex switch cycles (e.g. combined wobbler and frequency switching) can invent their own coding.

A hot load / cold load / sky calibration may be stored as one subscan or be split into its component parts and stored as three subscans. When stored as three subscans, OBSMODE rather than PHASEn takes the values HOT/COLD/SKY.

The phase weighting has to be calculated from the total integration times in each phase (INTEGTIM in ARRAYDATA).

15.4 Calibration parameters

A rough calibration of both spectral line and continuum data is carried out at the telescope. The derived calibration parameters are stored in the FEBEPAR table (for parameters which don't usually vary from scan to scan, but are measured occasionally) and in the MONITOR table. Although this at-the-telescope calibration is stored, a careful data reduction should go back to assess the quality of the original calibration data, average and interpolate between calibrations as necessary. Calibration parameters in the FEBEPAR table which are not available at the time of writing should be left blank to be updated later.

15.4.1 Continuum calibration

To calibrate a continuum observation the following are needed: opacity per feed(elevation-dependent); gain-elevation correction; counts-to-Jy calibration factor; feed offsets, HPBW of each feed, and beam shapes; and flatfield. This information is stored in MBFITS as follows.
Ref: APEX-SRS-RUB-0002 (BOA Software Requirements).

– **Opacity correction**

The elevation-dependent opacity per feed per frequency band can be derived from the zenith opacity at each frequency. This comes from skydips, taumeter or bolometer total power measurements. In the case of bolometer total power and taumeter reports, the zenith opacity can be updated on an integration-by-integration basis. `TAUZEN` in MONITOR stores the zenith opacity. Bracketing taumeter readings from at least the beginning and end of each scan should be available so that the optical depth can be interpolated to each integration. If `TAUZEN` is determined from skydips, the value from the last skydip will have to be written at the telescope (but can be updated during further data reduction).

– **Gain-elevation correction**

For homology telescopes (Effelsberg and the 30m) this can be simply parameterised but is empirically measured. We take the parameterisation: intensity = intensity$/\cos^2(\epsilon - p1) + p2\sin^2(\epsilon - p1)$ where $p1$,$p2$ are two wavelength-dependent parameters.

– **Bolometer attenuation factor `BOLREFGN`**

This was changed into `BOLCALFC` but is now back again. Integer scale factor which works as follows:

$$\text{ADCs dynamic range} = (\pm 10 \text{ Volts})/g$$

For example:

g	Dynamic range
1	±10 Volt
2	±5 V
10	±1 V
100	±100 mV

When the signals exceed the limits given by g the acquired signals will be saturated. (G. Siringo)

– **Calibration factor (counts-to-Jy)**

From comparison of observed/theoretical planet fluxes and secondary calibrators. At the telescope a standard value will be taken but during later data reduction, the planet observations should be taken into account. The standard value is stored in `BOLCALFC` (FEBEPAR header) (previously `BOLREFGN`). Measured values from scans on flux calibrators are stored in the MONITOR point `OBSFLUX_CALFLUX_CALFAC`.

– **Array geometry**

Feed offsets and HPBW are determined occasionally from maps of strong sources and stored in `FEEDOFFX`, `FEEDOFFY` and `HPBW` (FEBEPAR table, per observation) per frequency band. More information about the beam shapes is not stored.

The array feed position and other calibration parameters in the FEBEPAR table change with wobbler position. This is not taken into account at the moment. It may not be as simple as storing these parameters twice for two wobbler positions, as the wobbler displacement may be more complex than this (e.g. 2-D, and data taken at intermediate positions). One possibility would be to parameterise the change in the calibration parameters with wobbler position. Calibration information at this level of detail is unlikely to be available for the raw data file and should be handled by the data reduction software.

– **Flatfield**

Array flatfield/ relative gains (measured occasionally) are stored in `FLATFIEL` (FEBEPAR, per scan).

15.4.2 Spectral line calibration

To calibrate a spectrum one requires: the Gain Array, the calibration temperature $T_{\rm cal}$, and the beam efficiency. These quantities are derived from cal colds (comparison of cold load, hot load and sky) plus continual monitoring of ambient and chopper temperatures and occasionally measured telescope efficiencies. Alternatively, T sky and $T_{\rm cal}$ can be calculated more frequently via an atmospheric model which reads the opacity from a water vapour radiometer.
ref: APEX-SRS-MPI-0000 (APEX spectral line software requirements)

– **Gain array**

  The channel gains are determined by comparing hot load and sky during a cal cold. Cal colds are observations which can occur within one scan interspersing astronomical data collection. The gain array is stored per cal cold in the MONITOR table.

– Calibration, receiver and system temperatures

  The calibration temperature `TCAL` for each frequency band is determined from cal colds or the atmospheric model, along with receiver (`TRX`), main band and image system (`TSYS`, `TIMAG`) temperatures. These temperatures are stored in `TRX_TSYS_TIMAG_TCAL_<baseband>` in the MONITOR table at the time of the cal cold. The basic measurements (`THOT`, `TCOLD`, `PHOT`, `PCOLD`, `PSKY`) are also stored in MONITOR.

– Image/signal sideband gain ratio

  `GAINIMAG`, which is also used in the calibration to derive `TCAL`, is measured occasionally by comparison with atmospheric models and stored in the FEBEPAR table every scan.

– Beam efficiency

  The main beam, aperture and forward efficiencies are measured occasionally on calibrators and are stored in the FEBEPAR table for each feed and band.

15.5 WCS coordinates

The representation of spatial coordinates underwent a major revision in v. 1.0. The aims of this are to take into account the latest version of the World Coordinate System (WCS), making clear the relationship between parameters in current use at the 30m and those stored in this format; and to clarify the sources of positional information written in the raw data. Spectral coordinates have also been updated to comply with WCS (see Sect. 15.5.9).

15.5.1 Sources of spatial coordinate information

The spatial coordinate information stored in the raw data file has several origins:

– **Observer's setup** the user's setup of spatial coordinate frame, wobbler and scanning setups.

  The observer's setup is stored in the SCAN header (reference frames, source position, observing, switch and scan modes) and the DATAPAR header (scan direction).

– **Commanded coordinates** User's setup translated into commanded antenna and wobbler position at a given time. For the antenna drive the commanded coordinates must be translated into (Az,El).

  For later comparison with the real-time coordinates, the commanded coordinates *including* the wobbler offsets can also be calculated at this stage, in any appropriate frame (e.g. RA/Dec, or the user's native frame). In v.1.0 we store the commanded longitude and latitude offsets from the source in the user's chosen basis frame (defined by `CTYPE`j in SCAN header, e.g. RA/Dec) in the DATAPAR table as (`CBASLONG`, `CBASLAT`)

– **Raw drive commanded and readout coordinates**

  The commanded (Az,El) antenna position is translated into telescope drive commands via the pointing model: part of this is handled by the telescope control system and part by the antenna pointing computer (see Sect. 15.5.8).

- **Real coordinates**

  The readout from the antenna drive is then back-translated by the pointing model into the real antenna Az/El at the time of observation. The telescope pointing can be affected by wind, tracking errors etc. and thus the readout positions will differ from the commanded coordinates. This pointing correction should be carried out at the telescope and once only (see Sect. 15.5.8).

  The real coordinates at a given time — the pointing-corrected antenna position in Az/El, plus the wobbler offsets – are stored in the MONITOR stream.

- **Derived coordinates** From the real antenna coordinates in Az/El plus the wobbler offsets, various forms of the celestial coordinates associated with an integration can be derived. Example coordinate systems are a standard basis frame such as RA/Dec, or offsets with respect to source position in user native frame according to the current observing setup.

  In v.1.0 we store derived Az/El (`AZIMUTH`, `ELEVATIO`), offsets in the user's native frame (`LONGOFF`, `LATOFF`), and longitude and latitude in the chosen basis frame(`BASLONG`, `BASLAT`) (which can be directly compared with the commanded coordinates `CBASLONG`, `CBASLAT`) in the DATAPAR table.

15.5.2 WCS representation and telescope control system input

The World Coordinate System (WCS) (Greisen & Calabretta 2002, Calabretta & Greisen 2002, Greisen et al. 2004) is considered by the FITS committee as a soon-to-be standard. It is a general, flexible and powerful system for the representation of image coordinates, both spatial and spectral, notably including support for nonlinear coordinates (such as spatial projections and nonlinear spectrometers). The ALMATI-FITS format, and our MBFITS format, based their representation of coordinates on this format (following the ALMATI-FITS data-reduction-oriented storage scheme). However, WCS is an evolving format and has been updated since the ALMATI FITS format was designed (latest update considered here Apr 2002). At some stage in the data reduction process, there will almost certainly be the requirement to write images out as FITS. The current plans for NIFTI and for APEX envisage carrying out all the data reduction within FITS formats, starting with MBFITS, until finally FITS images are written. It makes sense to ensure at the raw data stage that all the necessary coordinate information is stored in suitable format to allow easy construction of WCS-format image headers.

This does not mean storing a complete FITS-format image at the raw data stage. A data reduction program will be required to turn the raw data into a useful image (averaging integrations, subtracting off positions etc.). The same reduction software can compile the headers for the image, provided all the necessary information is made available. At the raw data stage, the only useful images that could be produced are uncalibrated maps of observed positions e.g. for array o-t-f maps (note FITS images do not have to be gridded on rectangular pixels).

Even a full description of the image axes does not comprise all the positional information: a small amount of additional description of how the observations were actually carried out is necessary for the calibration/data reduction process (as well as for the construction of the image headers).

The 30m currently stores extra positional information by storing the inputs to the telescope control system, OBSINP (see OBSINP manual), as well as the fundamental real-time positions in the DAPs (Data Associated Parameters). The OBSINP inputs reflect all the current flexibility of the telescope control systems. The system is tailored to single-dish observing and encompasses the observing modes that are likely to be in use at APEX. However, APEX will have a different telescope control system with different input keywords, and may have different observing modes. There are plans to upgrade the 30m control system to the New Control System (NCS), which will be more flexible than the current system. MBFITS should be general enough to support all the modes available at the 30m (current and future) and APEX. MBFITS keywords need to be general enough to cover what is needed from the OBSINP input, the NCS, and the APEX TICS commands.

The coordinate storage scheme used in MBFITS up until now has consisted of limited WCS-style keywords with added 30m-type parameters. But there has been duplication of information without it being clear which parameters are redundant, and the format has neither fully conformed with WCS-FITS nor provided for the full range of observing modes of the 30m system. Thus we undertake a revision in v.1.0, continuing with a scheme that is based on FITS WCS, but incorporates additional coordinate information where this is necessary for the data reduction or clarifies the observing procedure.

MBFITS is (as it stands) a raw data and not an image format. The information needed to construct an image becomes available at different times during the observing from different sources (e.g. TCS and telescope position monitors) and thus appears in various tables. We have left this information scattered about, but coded in a form such that it could be easily collected together to produce an image table. A next step is to write a definition for a separate image table, as a necessary step in the data reduction, but this is postponed until a later version.

Clearly, it must be possible to derive the WCS image descriptions from the OBSINP parameters, or from the APEX control system, plus the real-time position information. Where the same information can be stored in more than one way, the WCS format is always given; additional parameters are given where the WCS parameters are derived in such a complex way from the observing setup that a reverse translation is non-trivial, and where information is added that is not stored in the WCS header.

The derivations of WCS quantities from the 30m parameters are given below in Sect. 15.5.5, but first in Sect. 15.5.3 we give a brief description of the relevant aspects of WCS and in Sect. 15.5.4 list the 30m OBSINP parameters which currently control positioning at that telescope.

15.5.3 WCS coordinate systems

In this section we describe the coordinate systems in WCS and their meanings for single-dish observing. For the details of representing spatial coordinates using WCS see Calabretta & Greisen 2002. Fig. 2 illustrates the coordinate systems – more explanation of the figure is given at the end of Sect. 15.5.5.

The WCS specifies how to describe the celestial 'world' coordinate system – the coordinates in a standard astronomical basis frame – of an image given in pixel coordinates. It makes this translation using two intermediate coordinate systems: *intermediate world coordinates*, where angles are given in a spherical frame which is offset and rotated with respect to the basis frame, and *intermediate pixel coordinates*, the flat-plane projection of the intermediate world coordinates, which may be rotated with respect to the pixel axes.

The translation between the (planar) intermediate pixel coordinates (x, y) and the (spherical) intermediate world coordinates (ϕ, θ) is a non-linear one which depends on the projection. Calabretta & Greisen 2002 give many projections: the usual one for single-dish radio astronomy is Sanson-Flamsteed (SFL). This is similar to the former FITS global sinusoidal 'projection' (GLS) (though this was necessarily a linear coordinate translation), and gives a translation between intermediate pixel coordinates (x, y) and intermediate world coordinates (ϕ, θ) of

$$\begin{aligned} x &= \phi \cos\theta \\ y &= \theta. \end{aligned} \tag{3}$$

The two coordinate systems (x, y) and (ϕ, θ) therefore have the same origin (see Fig. 2).

In terms of our needs, the intermediate world coordinates (ϕ, θ) correspond to longitude and latitude offsets in the user's native coordinate frame - an arbitrary rotated frame appropriate to the observations. The pixel $(p1, p2)$ to intermediate pixel (x, y) coordinate offset/rotation then allows for translation of pixel offsets into the user's native system, in our case for array rotation and wobbler offsets.

15.5.4 30m parameters

The main OBSINP parameters which currently control positioning at the 30m are given in Table 1. For details see the OBSINP manual.

In addition, the 30m DAPs store the RA and Dec for each integration, and the longitude and latitude offsets LAM(t) and BET(t) if scanning.

The on-the-fly scanning parameters also control position, but only affect the WCS description through the resulting `LONGOFF` and `LATOFF`. The scan parameters are covered by `SCANxxxx` in the SCAN header: see Sect. 15.2.

15.5.5 WCS parameters (including derivation from 30m parameters)

The following list gives the keywords required in the WCS description. We also give the derivation from the 30m OBSINP+DAP, and what is now stored in MBFITS v.1.0.

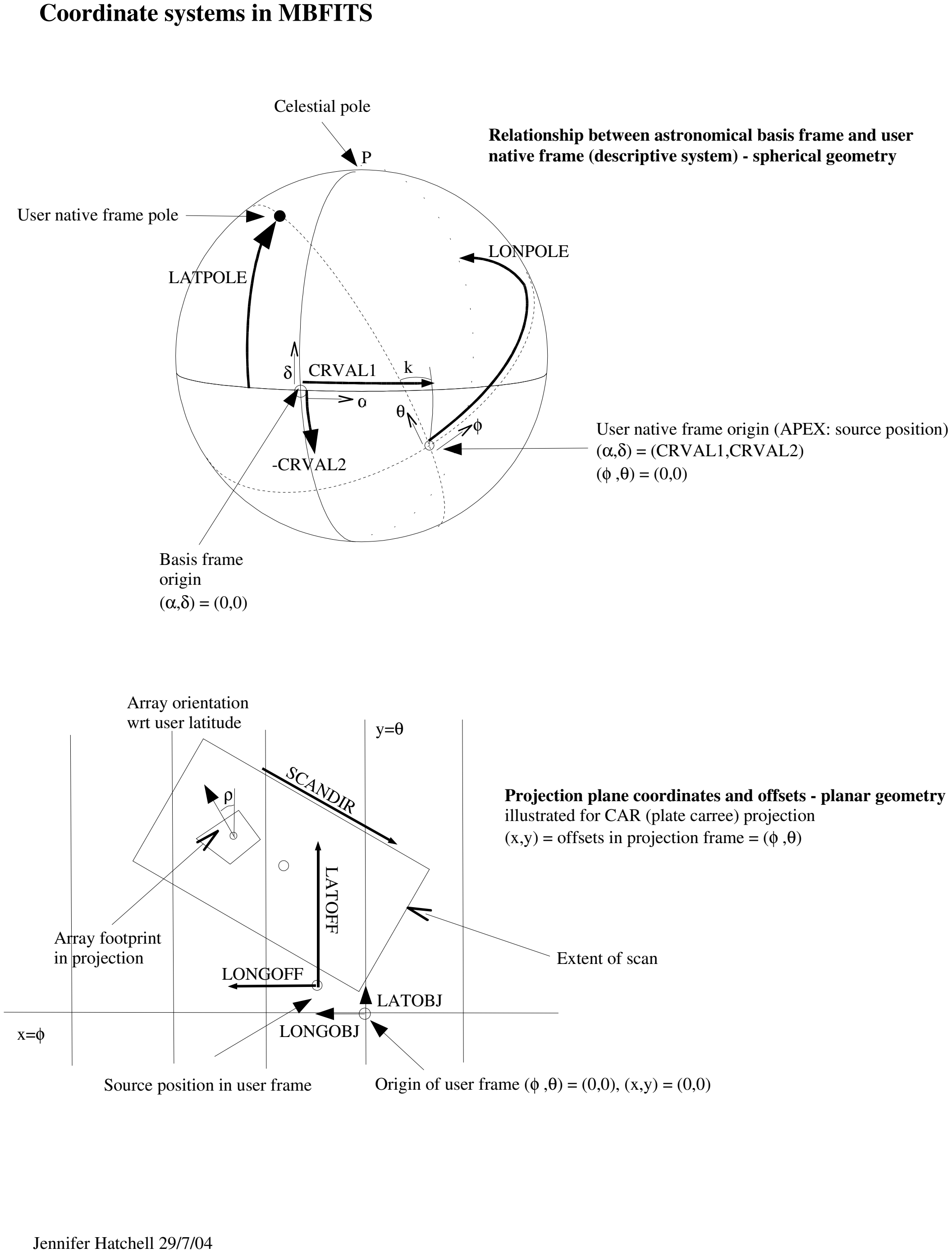

Coordinate systems in MBFITS
Celestial pole
P
Relationship between astronomical basis frame and user native frame (descriptive system) - spherical geometry
User native frame pole
LATPOLE
LONPOLE
δ
CRVAL1
k
o
θ
ϕ
-CRVAL2
User native frame origin (APEX: source position)
(α,δ) = (CRVAL1,CRVAL2)
(ϕ ,θ) = (0,0)
Basis frame origin
(α,δ) = (0,0)
Array orientation wrt user latitude
y=θ
SCANDIR
ρ
Projection plane coordinates and offsets - planar geometry
illustrated for CAR (plate carree) projection
(x,y) = offsets in projection frame = (ϕ ,θ)
LATOFF
Array footprint in projection
Extent of scan
LONGOFF
LATOBJ
x=ϕ
LONGOBJ
Source position in user frame
Origin of user frame (ϕ ,θ) = (0,0), (x,y) = (0,0)
Jennifer Hatchell 29/7/04

Figure 2: Coordinate relationships for single dish array observations in WCS scheme. Coordinate systems are: (α, δ) astronomical basis frame (world coordinates); (ϕ, θ) user native frame (intermediate world coords.); (x, y) intermediate pixel coords.; and (p_1, p_2) pixel coords.

Table 1: OBSINP positioning parameters

Keyword	**Description**
SBAS	Basis system: if followed by D, D * or P then use SL0P, SB0P and SK0P additionally to define a user native frame
SEQN	Equinox (where necessary)
In SBAS D:	
SL0P	origin of user frame in basis system
SB0P	origin of user frame in basis system
SK0P	angle which user frame zero meridian makes with basis system meridian
*In SBAS D * or SBAS P:*	
SL0P	pole of user frame in basis system
SB0P	pole of user frame in basis system
SK0P	angle which user system zero meridian makes with basis meridan through pole
SLAM	Source longitude in basis system
SBET	Source latitude in basis system
OLAM or OLAM*	Long. offset from source in user frame
OBET	Lat. offset from source in user frame

CTYPE*js* (SCAN header) Define basis and user native frame and projection: format is 4-3 with 4 characters for the basis and 3 for the projection, padded with dashes. The first four characters give one of the basis systems available in the FITS standard: RA/DEC, GLON/GLAT, ELON/ELAT, HLON/HLAT, or SLON/SLAT. (G for galactic, E for ecliptic, H for helioecliptic, S for supergalactic: see Calabretta & Greisen 2002).

WCSNAME gives a human-readable description of the coordinate system. For rotated (descriptive) user frames WCSNAME is 'descrip' followed by the basis frame description e.g. 'descrip horizontal'. Where the user frame is not rotated then use 'absolut horizontal' or 'absolut equatorial' (or galactic, etc.).

We need two basis representations which are non-standard, to handle Az/El and moving body coordinates. WCS FITS allows for the possibility of coding your own basis system as xLON/xLAT, with definition of the system given in the additional keyword WCSNAME. We propose to use ALON/ALAT with WCSNAME 'absolut(e) horizontal' for Azimuth/Elevation, and HA/DEC with WCSNAME 'absolut(e) hour angle' for Hour angle/Declination'.

See Sect. 15.5.6 for a description of how MBFITS works in the case of moving bodies.

The projection refers to the conversion between the user native frame (intermediate world coordinates) and its linear version (intermediate pixel coordinates). The usual projection for single dish radio astronomy is Sanson-Flamsteed, coded SFL, for which

$$\begin{aligned} x &= \phi \cos\theta \\ y &= \theta \end{aligned} \tag{4}$$

where ϕ and θ are longitude and latitude in the user's native spherical system, and (x,y) are latitude/longitude offsets. Other projections can be specified if required.

WCSNAME (SCAN header)

A description of the basis and user native coordinate system, especially where it is non-standard. See CTYPE*js* (above).

RADESYS (SCAN header) Additional ecliptic/equatorial basis system definition e.g. FK4/FK5: see Calabretta & Greisen (2002) Table 1.

EQUINOX (SCAN header) Basis system equinox.

A combination of CTYPE, WCSNAME, RADESYS and EQUINOX cover all the astronomical basis systems in the OBSINP scheme.

PVj_m Additional projection parameters – not needed for SFL or GLS, so we leave these out of MBFITS.

$\mathbf{p}_i$ Pixel coordinates if not a regular grid.

Stored in the FEBEPAR table, in the array offsets FEEDOFFX, FEEDOFFY

PCi_j Rotation/skew matrix to translate pixel coordinates to intermediate pixel coordinates (projection of user frame).

For an integration where the array y axis makes an angle ρ with the native (user) frame θ axis, PC is the rotation matrix

$$PC = \begin{pmatrix} \cos\rho & -\sin\rho \\ \sin\rho & \cos\rho \end{pmatrix}.$$

For single feed observations, PC is the identity matrix. The angle ρ is derived from the dewar angle, optical arrangement and elevation.

`PC`i_j are stored in DATAPAR, derived from the dewar rotation mode `DEWRTMOD`, dewar angle `DEWANG` (FEBEPAR), elevation and parallactic angle `PARANGLE` (DATAPAR).

CRPIXi Pixel coordinates of the native frame origin. Note these are not the pixel coordinates of the source, which may not be at (0,0), and that they are measured in the pixel coordinate frame, which we take to be centred at the array centre and oriented along the array axes.

Given the source position in the native frame (`LONGOBJ`, `LATOBJ`), plus lat./long. offsets to the array centre (`LONGOFF`, `LATOFF`)(which may include wobbler offsets), then the `CRPIX` are:

$$\begin{pmatrix} \texttt{CRPIX1} \\ \texttt{CRPIX2} \end{pmatrix} = -PC^{-1} \begin{pmatrix} \texttt{LONGOBJ}\cos(\texttt{LATOBJ}) + \texttt{LONGOFF} \\ \texttt{LATOBJ} + \texttt{LATOFF} \end{pmatrix}$$

where PC is the matrix to rotate the array pixel coordinates into the user native frame, given above, with

$$PC^{-1} = \begin{pmatrix} \cos\rho & \sin\rho \\ -\sin\rho & \cos\rho \end{pmatrix}.$$

The `CRPIX`i condense the source position and offsets in the user frame into one pair of coordinates. We therefore keep the source position in the user frame as `LONGOBJ` and `LATOBJ` (SCAN header), and the offsets from this position in `LATOFF` and `LONGOFF` (INTMON table).

The `CRPIX`i are stored in INTMON.

CDELTj

Scale pixel coords to intermediate pixel coords.

FITS requires all angles to be given in degrees. As long as `FEEDOFFX` and `FEEDOFFY` comply with this, the `CDELT` take their default values of 1 and can be left out.

CRVALj

Origin of native frame in basis coordinates

Stored in SCAN header.

In the SBAS D scheme, these are SL0P and SB0P.

LONPOLE

ϕ_{p}, Native frame longitude of basis system pole. Values between -180 and $+180$ deg.

Stored in SCAN header.

In the SBAS D scheme,

$$\tan(\texttt{LATPOLE}) = \frac{\sin(\mathrm{SK0P})\cos(\mathrm{SB0P})}{\sin(\mathrm{SB0P})}$$

LATPOLE

δ_{P} Basis system latitude of native frame pole. Note the opposite definition to `LONPOLE`, but by symmetry this is equal to the native frame latitude of the basis system pole θ_{P}

Stored in SCAN header.

In the SBAS D scheme,

$$\sin(\texttt{LATPOLE}) = \cos(\mathrm{SB0P})\cos(\mathrm{SK0P}).$$

CRVAL, LONPOLE and LATPOLE can also be calculated in terms of SL0P, SK0P and SB0P for the alternative SBAS P or D * scheme (not yet done). We propose to store the frame definition in the WCS format. Storing CRVALj, LONPOLE and LATPOLE is equivalent to storing SL0P, SK0P and SB0P, and the translation is straightforward.
The keywords required for each step in the coordinate transformation are:

Pixel coordinates (p_1, p_2) to projection plane coordinates (x, y)

- (FEEDOFFX, FEEDOFFY): pixel coordinates
- CRPIXi: pixel coordinates of projection plane origin
- PCi_j: rotation matrix

Projection plane (x, y) to user native spherical coordinates (ϕ, θ)

- CTYPEj: xxxx-SFL projection

User native spherical (ϕ, θ) to celestial basis (α, δ)

- CRVALj : native system origin in basis coordinates
- LONPOLE: longitude of native pole in basis coordinates
- LATPOLE: latitude of basis pole in native coordinates
- CTYPEj: xLON/xLAT-xxx user native system
- CTYPEjB: xLON/xLAT-xxx basis system
- RADESYS: equatorial/ecliptic system additional description
- EQUINOX

Fig. 2 illustrates the general situation which we have just described using WCS keywords. We show the array centred at offsets (LATOFF, LONGOFF) from a source position (LONGOBJ, LATOBJ) in the user's rotated frame. The user's native frame is in turn centred at (CRVAL1, CRVAL2) in the standard astronomical basis frame defined by CTYPEj.
If observations are made with a single feed, then the PC matrix reduces to the identity matrix and the pixel offsets are all zero. In this case, it would be simpler to define the pixel offsets as (LONGOFF, LATOFF) and measure the pixel offset of the user frame origin CRPIX from the source position in the user native frame. This scheme could well be useful later in the data reduction process, but here in the raw data format we give the more general description for rotated array observations so that all observing possibilities are covered.

15.5.6 Moving bodies

Moving bodies are a special case for coordinate storage, because the local reference frame is defined by the ephemeris and rotates with time. To support moving body observations, we have a flag MOVEFRAM in the SCAN header. The orbital elements are then also listed in the SCAN header.
As the native coordinate frame rotates with time, the parameters which define it with respect to the fixed basis frame need to be stored with each integration. These are given as MCRVAL1, MCRVAL2, MLONPOLE and MLATPOLE in the DATAPAR table, per integration. The equivalents in the SCAN header can be ignored for moving bodies.
We then need two WCS descriptions of the image (multiple image axes descriptions are allowed in FITS). Firstly, as in order to carry out the observations we have already defined the body frame in terms of a fixed basis system (e.g. RA/Dec), we have the description to provide output in that fixed frame, using the CTYPE values in the SCAN header. This is only useful on an integration-by-integration basis, as the body moves across the sky.
More usefully, we propose a second coordinate description with CTYPE=xxLN/xxLT (longitude and latitude in the frame of the body, with xx a two-letter abbreviation of the body name) and WCSNAME 'Moving body coordinates' or '<bodyname> coordinates'. As this frame exactly tracks the body's native coordinate system, CRVALj= $(0, 0)$, LONPOLE= 0 and LATPOLE= 90. At present this second axis description is not written explicitly into MBFITS as it does not change and is only useful at the point where an image table is written.
Whether the more complex observing modes such as on-the-fly can be used on a moving body will depend on the cleverness of the telescope control system. The MBFITS format allows for a full description in the case of a moving frame as for a fixed frame.

15.5.7 Observations in Az/El

Another special case for coordinate representation are observations where the user native frame is Az/El but the required basis frame is a fixed celestial one. The Az/El frame shifts and rotates with time with respect to the celestial frame. Again, the moving frame flag `MOVEFRAM` is set and the horizontal frame centre is defined by `MCRVAL1`, `MCRVAL2`, `MLONPOLE` and `MLATPOLE` in the DATAPAR table, per integration, and the equivalents in the SCAN header can be ignored. If output is only required in Az/El frame, then the native frame is the basis frame and the frame centre stays fixed.

For observations in Az/El about an offset position defined in a celestial system, this position becomes the center of the user native (horizontal) frame. The offset relative to the source center is stored in the SCAN keywords `PATLONG` and `PATLAT`. The pattern offsets in the DATAPAR table are given in the user native frame, i.e. Az/El.

In all cases, the derivation of (LONGOFF, LATOFF) from the real-time antenna coordinates and wobbler offset plus frame definitions must take into account the chopping direction, stored in `WOBDIR`.

15.5.8 Pointing

The corrections to the pointing can be divided into antenna, subreflector and receiver-dependent static terms, dynamic pointing and focus corrections from observations of pointing sources during the observations, focus/elevation interplay, and the refraction correction.

The pointing corrections are dealt with in two stages: the telescope control system (TICS) handles the refraction correction, the dynamic antenna pointing correction, and the receiver terms; and the antenna pointing computer deals internally with the static pointing, the dynamic focus correction, and the focus/elevation interplay.

The static pointing coefficients for the APEX antenna follow the 7-coefficient model described by Mangum (2001), which follows the Stumpff (1972) model, plus an extra flexure term which behaves in the same way as receiver offsets at the Nasmyth focus. Following ALMA developments, this has now been extended to include higher order sine and cosine terms and this is reflected in the SCAN header keywords from v.1.55. Note the change in sign for some terms over their equivalents in the original 7-coefficient model. The pointing terms are given in Table 2.

At APEX it is yet to be decided if the raw telescope drive readouts and the focus/elevation correction (?) will be available from the pointing computer and therefore if the full pointing calculation can be repeated offline. We also don't store the full set of coefficients for reconstructing the refraction correction. However, we do store the static antenna and receiver terms and the dynamic antenna and focus corrections, plus the total refraction correction, for reference.

15.5.9 WCS spectral coordinates

The v.0.3 scheme for spectral coordinates followed the ALMA-TI FITS format which basically complied with WCS, but as WCS has been updated since ALMA-TI FITS was written we have made some changes in v. 1.0 to keep in line with the latest version (Greisen et al. 2004). These axes descriptions are in the ARRAYDATA header. (The velocity description is only given for spectral line receivers.)

There are alternative frequency and velocity descriptions of the data axis, both in the chosen rest frame. These are described by WCSNAME `'xxxxFreq'` and `'xxxxVRad'` where xxxx represents the rest frame (e.g. LSRK). The frequency description is labelled 'F' and the radio velocity description 'R'.

In the WCS system, `VELOSYS` (`VSYS`) stores the observer's velocity with respect to the rest frame, which we previously stored in `V_FRAME`. This velocity difference changes with time – here it is stored every subscan, but if needed it could be stored closer to the integration level (in MONITOR?). The old keywords `VELO_SYS` and `V_FRAME` have been removed from the scan header. `VELO_SYS` – a 4-letter description of the velocity frame – clashed with the WCS name.

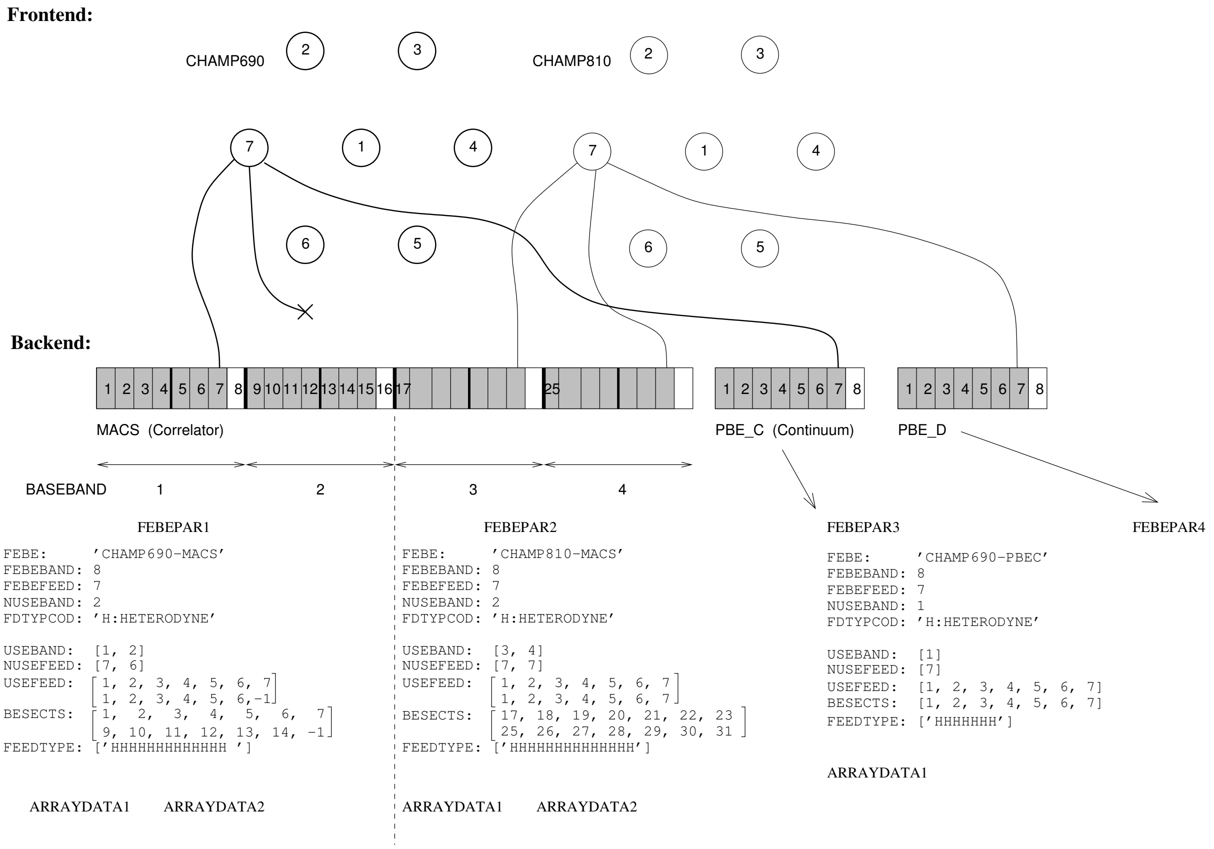
`SPECSYS` (`SPEC`) describes the velocity standard of rest frame. `SSYSOBS` (`SOBS`) is added to take the constant spectral coordinate of each image (pixel) plane - here the observer's frame 'TOPOCENT'. `VSOURCE` (`VSOU`) gives the source velocity with respect to the standard of rest, and `VELOSYS` (`VSYS`) gives the observer's velocity with respect to the standard of rest.

The keywords `V_EARTH`, `V_HEL` and `V_SYS` subdividing `VELOSYS` into components are not included. These values can be stored in the WCS system by adding extra velocity frame descriptions in addition to the

Table 2: Pointing terms

Term	30m equiv.	Description
Antenna static terms, in SCAN *header*		
IA	−P1	Azimuth encoder zero offset
CA	−P2	Collimation error of the electromagnetic axis
NPAE	−P3	Collimation of the axes / non-perpendicularity between mount azimuth and elevation axes
AN	−P5	Azimuth axis offset / misalignment north-south / zenith shift
AW	−P4	Azimuth offset / misalignment east-west / zenith shift
IE	P7	Elevation encoder zero offset
HECE (ECEC)	P8	Gravitational flexure perpendicular to optical axis plus vertical receiver offset at Nasmyth focus
HESE (ZFLX)		Gravitational flexure parallel to optical axis plus horizontal receiver offset at Nasmyth focus
HASA		Azimuth correction, function of sin(Az)
HACA		Azimuth correction, function of cos(Az)
HESA		Elevation correction, function of sin(Az)
HASA2		Azimuth correction, function of sin(2Az)
HACA2		Azimuth correction, function of cos(2Az)
HESA2		Elevation correction, function of sin(2Az)
HECA2		Elevation correction, function of cos(2Az)
HACA3		Azimuth correction, function of cos(3Az)
HECA3		Elevation correction, function of cos(3Az)
HESA3		Elevation correction, function of sin(3Az)
HECE2		Elevation correction, function of cos(2El)
HECE6		Elevation correction, function of cos(6El)
HESA4		Elevation correction, function of sin(4Az)
HESA5		Elevation correction, function of sin(5Az)
HSCA		Horizontal correction, function of cos(Az)
HSCA2		Horizontal correction, function of cos(2Az)
HSSA3		Horizontal correction, function of sin(3Az)
HSCA5		Horizontal correction, function of cos(5Az)
NRX		Horizontal displacement of Nasmyth receiver
NRY		Vertical displacement of Nasmyth receiver
HYSA		Azimuth backlash hysteresis (used in Effelsberg)
HYSE		Elevation backlash hysteresis (used in Effelsberg)
Receiver static terms, in FEBEPAR *table*		
		Add to the corresponding Antenna static terms.
Focus/elevation correction: internal to pointing computer?		
Observer applied pointing corrections, in SCAN		
PDELTAIA	NULA	Azimuth offset determined from pointing observation
PDELTACA	COL*	Collimation offset determined from pointing observation
PDELTAIE	NULE	Elevation offset determined from pointing observation
Correction due to focus change by observer, in SCAN		
FDELTAIA		Correction to IA due to focus change in X or YTILT
FDELTACA		Correction to CA due to focus change in X or YTILT
FDELTAIE		Correction to IE due to focus change in Y or XTILT
Refraction correction, in MONITOR		
REFRACTIO		Refraction correction, from `HUMIDITY`, `TAMBIENT`, and `PRESSURE` as a function of elevation

'LSRK' one. It would also be simple to provide alternative frequency/velocity axis descriptions for the image sideband or for velocities in the source frame, if required.
Actual frequency settings for the receivers/LO chain are stored in the MONITOR stream.



```
FEBE:     'CHAMP690-MACS'
FEBEBAND: 8
FEBEFEED: 7
NUSEBAND: 2
FDTYPCOD: 'H:HETERODYNE'

USEBAND:  [1, 2]
NUSEFEED: [7, 6]
USEFEED:  ⎡1, 2, 3, 4, 5, 6, 7⎤
          ⎣1, 2, 3, 4, 5, 6,-1⎦
BESECTS:  ⎡1,  2,  3,  4,  5,  6,  7⎤
          ⎣9, 10, 11, 12, 13, 14, -1⎦
FEEDTYPE: ['HHHHHHHHHHHHH ']
```

```
FEBE:     'CHAMP810-MACS'
FEBEBAND: 8
FEBEFEED: 7
NUSEBAND: 2
FDTYPCOD: 'H:HETERODYNE'

USEBAND:  [3, 4]
NUSEFEED: [7, 7]
USEFEED:  ⎡1, 2, 3, 4, 5, 6, 7⎤
          ⎣1, 2, 3, 4, 5, 6, 7⎦
BESECTS:  ⎡17, 18, 19, 20, 21, 22, 23 ⎤
          ⎣25, 26, 27, 28, 29, 30, 31 ⎦
FEEDTYPE: ['HHHHHHHHHHHHHH']
```

```
FEBE:     'CHAMP690-PBEC'
FEBEBAND: 8
FEBEFEED: 7
NUSEBAND: 1
FDTYPCOD: 'H:HETERODYNE'

USEBAND:  [1]
NUSEFEED: [7]
USEFEED:  [1, 2, 3, 4, 5, 6, 7]
BESECTS:  [1, 2, 3, 4, 5, 6, 7]
FEEDTYPE: ['HHHHHHH']
```

Figure 3: Example FEBE setup showing the values of keywords in the FEBEPAR header and binary table. In this case there are 2 frontends each with 7 physical pixels. Each pixel is connected to 2 correlator bands and a continuum backend (used backend sections shown in grey). This results in 4 different FEBE combinations (therefore 4 FEBEPAR tables). The two correlator FEBEs each have 2 basebands (therefore 2 ARRAYDATA tables). The correlator can only be configured in groups of 4 sections (hence the maximum number of bands is 8). In the case of `CHAMP690`, pixel 7 is not connected to the second correlator band. Therefore the corresponding entries in `USEFEED` and `BESECTS` are set to -1.

15.5.10 Feeds and backend sections

The term 'Feed' is used to describe a part of the frontend that delivers a certain type of data to the backend. This can either reflect physical 'pixels' or different data types from a single pixel (AC/DC coupled bolometer, amplitude/phase for holography). The total number of available 'physical' feeds is recorded in the FEBEPAR header in the `FEBEFEED` keyword. The characteristics of these are described in the binary table with keywords dimensioned by N_{FD} such as `FEEDOFFX` and a single letter character describing the type of each one is recorded in the FEBEPAR binary table in the `FEBETYPE` keyword.

Physical inputs to the backend that can be connected to the different feeds are called backend 'sections'. Backend sections that share the same frequency setup are grouped into 'basebands' (`BASEBAND`), with the data for each baseband written to a single ARRAYDATA table. The maximum number of configurable frequency setups that are possible for the backend is recorded in the `FEBEBAND` keyword and the number that are actually in use is recorded in the `NUSEBAND` keyword (this reflects the number of ARRAYDATA tables needed per subscan).

Actual lists of the *used* basebands and feeds are given in the FEBEPAR binary table. The list of basebands in use is recorded in `USEBAND` and the feeds connected to each of these are listed in `USEFEED`. As a physical feed can be connected to more than one backend section, the `USEFEED` array may be longer than N_{FD}, and may have varying lengths for different basebands. To accomodate this, it must be dimensioned according to the baseband with greatest number of used feeds with blank spaces in the array given a nominal value (e.g. `-1`) - it is given the FITS "variable length array" type. The backend sections that are connected to

each feed within each baseband are recorded in BESECTS (in the order used in USEFEED - therefore there may be blank entries in BESECTS too).
The header keyword, **FDTYPCOD**, is needed to describe the numerical codes used in FEEDTYPE to allow for more complicated explanations. For example, holography data with one physical pixel but outputting both amplitude and phase can be coded with a two feed types which would be explined in FDTYPCOD with e.g. '1:AMPLITUDE 2:PHASE'.

16 References

References

[1] Lucas, R., and Glendenning, B., 2001, ‘ALMA Test Interferometer Raw Data Format’, ALMA-SW-0015.

[2] Brunswig W., Thum C., Salter C., Ruiz M., Schraml J., ‘OBSINP manual (30m)’, http://www.iram.es/IRAMES/otherDocuments/manuals/

[3] Greisen E. W., Calabretta M.R. 2002, ‘Representation of world coordinates in FITS’, A&A, 395, 1061

[4] Calabretta M.R., Greisen E. W., 2002, ‘Representation of celestial coordinates in FITS’, A&A, 395, 1077

[5] Greisen E.W., Calabretta M.R., Valdes F.G., Allen, S.L., 2006, ‘Representations of spectral coordinates in FITS’, A&A, 446, 747

[6] Greisen E.W., Calabretta M.R., Valdes F.G., Allen, S.L., 2004, ‘Representations of spectral coordinates in FITS’, in prep., available from http://www.aoc.nrao.edu/∼egreisen/

[7] Wells, D.C., Greisen E.W., Harten R.H., 1981, A&AS, 44, 363

[8] Mangum J. G., 2001, ’A Telescope Pointing Algorithm for ALMA’, ALMA Memo 366

[9] Bunclark P., Rots A., 1996, ‘Precise re-definition of the DATE-OBS Keyword encompassing the millenium’, http://www.cv.nrao.edu/fits/wcs/year2000.txt

[10] Cotton W.D., Tody D., Pence W.D., 1995, ‘Binary table extension to FITS’, A&AS 113, 159

[11] Quinn, P., 1997, ’ESO Data Interface Control Document’, GEN-SPE-ESO-19400-794/1.1/0, http://archive.eso.org/dicb/dic-1.1/dic-1-1.html

[12] Jennings, D.G., Pence, W.D., Folk, M., Schlesinger, B.M., 1997, A Hierarchical Grouping Convention for FITS, Rev. 8, http://fits.gsfc.nasa.gov/group.html

17 MBFITS specification

17.1 Data types

The following coding is used in the tables for data types:

L	1-byte logical
A	1-byte ASCII character
I	2-byte integer (±32767)
J	4-byte integer
E	4-byte real
D	8-byte double
P	16-byte pointer

17.2 The Primary header

Basic keywords necessary for FITS, a brief description of the file origin plus the keywords for the ESO archive. For a full description of the ESO keywords, see Quinn 1997.
The ESO keywords are of three types:

– Keywords which only appear in the primary header: `HIERARCH ESO DPR CATG` and `HIERARCH ESO DPR TECH`.

– Keywords which are direct copies of keywords in other parts of MBFITS: `OBJECT`, `MJD-OBS`, `DATE-OBS`, `RADECSYS`, `EQUINOX`, `HIERARCH ESO TEL ALT`, `HIERARCH ESO TEL AZ`.

– Keywords which are derived from keywords elsewhere in MBFITS, but require some calculation or translation of values: `HIERARCH ESO DPR TYPE` (`SCANTYPE`), `RA`, `DEC`, `EXPTIME`, `INSTRUME`, `ORIGFILE`, `AIRM START`, `AIRM END`.

The ESO keywords are only required for APEX data headed for the ESO archive, ie. not for 30m or Effelsberg data.

Keyword	Type	Value	Description
`NAXIS`	1I	0	
`SIMPLE`	1L	T	
`BITPIX`	1I	32	
`EXTEND`	1L	T	
`TELESCOP`	20A	APEX/EFF100/IRAM30 etc.	Telescope Name
`ORIGIN`	20A		Organisation or Institution
`CREATOR`	20A		Software (including version)
`MBFTSVER`	10A		MBFITS version
`COMMENT`	20A		
ESO archive keywords:			The following keywords are for the ESO archive of APEX data.

`HIERARCH ESO DPR CATG`	30A	Scan category: SCIENCE/ TEST/ CALIB/ SIMULATION/ TECHNICAL/ OTHER.	CATG is used primarily to govern access to the files: SCIENCE has proprietary rights, CALIB is open, TECHNICAL is used to describe highly technical data of no use to anybody other than the instrument inner circle to be hidden from the outside world, and TEST is a notch higher than TECHNICAL, i.e. it is not hidden, but no warranties for the usefulness of the data for any purpose are assumed or implied (Adam Dobrzycki, ESO).
`HIERARCH ESO DPR TYPE`	30A	Scan type: OBJECT/ POINT/ FOCUS/ FLUX(fluxcal)/ CAL/ SKYDIP/ HOLO/ OTHER	From `SCANTYPE`(SCAN): OTF, ON, ONOFF, RASTER, CROSS and other scan configurations on the source are mapped onto OBJECT. Up to 3 comma-separated entries allowed.
`HIERARCH ESO DPR TECH`	30A	Scan technique: CONTINUUM/ SPECTRUM(spectroscopy)/ POLARIMETRY/ IMAGE	Up to 3 comma-separated entries allowed, e.g. CONTINUUM, IMAGE.
`HIERARCH ESO OBS PROG ID`	30A		ESO program ID, where available. Format: K PP.C-NNNN R
`HIERARCH ESO OBS ID`	1J		Observation block ID, where available
`HIERARCH ESO TEL AIRM START`	1D		Airmass at start, calc. from `ELEVATIO`(DATAPAR). If a more sophisticated calculation than AM = 1/sin(EL) is available, then expect these values to be in MONITOR.
`HIERARCH ESO TEL AIRM END`	1D		Airmass at end. See above.
`HIERARCH ESO TEL ALT`	1D	`ELEVATIO`(DATAPAR) of first integration.	Altitude at start,
`HIERARCH ESO TEL AZ`	1D	`AZIMUTH`(DATAPAR) of first integration.	Azimuth at start.
`INSTRUME`	30A	e.g. `APEXHET`, `APEXBOL`, `APEXTEST`	Instrument type (heterodyne/bolometer/test)
`FEBEn`	30A	`FEBE`(FEBEPAR)	FE and BE used, where `n` is an integer 1 to N_{FEBE}.
`FREQn`	1D	`RESTFREQ`(ARRAYDATA)	Centre frequency for `FEBEn`
`LINEn`	30A	`TRANSITI`(ARRAYDATA)	Line name corresponding to `FREQn`
`BWIDn`	1D	`BANDWID`(ARRAYDATA)	Bandwidth for `FEBEn`

OBJECT	30A	OBJECT(SCAN)	Object observed
TIMESYS	4A	TIMESYS(SCAN)	Time system (TAI, UTC, UT1, etc ...) for MJD and DATE-OBS. Always TAI for APEX.
MJD-OBS	1D	MJD(SCAN)	MJD start in TIMESYS system.
DATE-OBS	30A	DATE-OBS(SCAN)	Date of observation in TIMESYS system.
RA	1D	Derived from LONGOBJ, LATOBJ and native frame configuration	Apply the same conversion as for BASLONG/BASLAT(DATAPAR) with LATOFF/LONGOFF= $(0, 0)$ then convert from basis frame to equatorial coords. if necessary
DEC	1D	Derived from LONGOBJ, LATOBJ and native frame configuration	see above.
RADECSYS	30A	RADESYS(SCAN)	Coordinate reference frame. RADESYS is the WCS keyword
EQUINOX	1E	EQUINOX(SCAN)	Equinox.
EXPTIME	1D	$\sum\left[\texttt{INTEGTIM}(\text{DATAPAR})\right]$	Total integration time.
ORIGFILE	40A	e.g. 'APEX-<scan>-O- K PP.C-NNNN R -YYYY.fits'	Original file name.
PROJID	18A	PROJID(SCAN)	Full Project ID.

17.3 The SCAN-MBFITS Binary Table

Stored every scan, containing parameters which do not change between subscans, including:

- Telescope and observatory parameters
- Time system
- Coordinate system
- Velocity system
- Project ID
- Source and coordinates
- Observing mode
- Pointing coefficients

17.3.1 SCAN-MBFITS Binary Table Header Keywords

Keyword	Type	Units	FITS Description	Comment
EXTNAME	20A	-	SCAN-MBFITS	
TELESCOP	20A	-	Telescope Name	
SITELONG	1D	deg	Observatory longitude	
SITELAT	1D	deg	Observatory latitude	
SITEELEV	1E	m	Observatory elevation	
DIAMETER	1E		dish diameter	
PROJID	18A	-	Project ID	was 17 characters, now 18 for full APEX projext ID including the ESO run ID
OBSID	24A	-	Observer and operator initials	was 12 characters - now 24
SCANNUM	1J	-	Scan number	
TIMESYS	4A	-	time system (TAI, UTC, UT1, etc ...) for MJD and DATE-OBS. Always TAI for APEX.	
DATE-OBS	23A	-	scan start in TIMESYS system	
MJD	1D	day	Scan date/time (Modified Julian Date) in TIMESYS system	
LST	1D	s	Local apparent sidereal time (scan start)	
NOBS	1J	-	Number of subscans in this scan	was N_OBS pre-v.1.5
NSUBS	1J	-	Number of subscans in this scan	
UTC2UT1	1D	s	UT1-UTC time translation	

TAI2UTC	1D	s	UTC-TAI time translation	See Sect. 15.5 for coordinate system definition
ETUTC	1D	s	Ephemeris Time - UTC time translation	
GPSTAI	1D	s	GPS time - TAI translation	a constant, but here for reference
WCSNAME	20A	-	Human-readable description of frame e.g. 'Horizontal coordinates'	
RADESYS	8A	-	additional system definition for ecliptic/equatorial coords	
EQUINOX	1E	Julian yrs	Equinox	
CTYPE1	8A	-	Basis system type (longitude) – XLON-SFL	where x takes appropriate values – see Sect. 15.5.3
CTYPE2	8A	-	Basis system type (latitude) – XLAT-SFL	
CRVAL1	1D	deg	Native frame zero in basis system (long.)	
CRVAL2	1D	deg	Native frame zero in basis system (lat.)	
CTYPE1N	8A	-	Native system type (longitude) – XLON-SFL / XLON-GLS	where x takes appropriate values – see Sect. 15.5.3
CTYPE2N	8A	-	Native system type (latitude) – XLAT-SFL / XLAT-GLS	Note that APEX uses GLS instead of SFL to make it easier to use the LONGOFF/LATOFF DATAPAR-MBFITS columns for the Gildas software. The SFL projection can be calculated using the BLATOBJ keyword and the new DATAPAR-MBFITS DEC column.
LONPOLE	1D	deg	Native longitude of celestial pole: range -180 to $+180$ deg	usually around zero
LATPOLE	1D	deg	Basis latitude of native pole	
OBJECT	20A	-	Source name	
BLONGOBJ	1D	deg	Source longitude in basis frame	
BLATOBJ	1D	deg	Source latitude in basis frame	
LONGOBJ	1D	deg	Source longitude in user native frame	
LATOBJ	1D	deg	Source latitude in user native frame	
PATLONG	1D	deg	Pattern long. offset in user native frame	Needed to capture EQEQHO setups (see section 15.5.7)
PATLAT	1D	deg	Pattern lat. offset in user native frame	Needed to capture EQEQHO setups (see section 15.5.7)
CALCODE	4A	-	Calibrator Code	
MOVEFRAM	1L	-	True if tracking a moving frame	In the case of moving objects, the orbital elements are also stored. The abbreviations in the description match JPL Horizons.

PERIDATE	1D	Julian days	TP, Full Julian date of perihelion passage	MJD = JD − 240000.5
PERIDIST	1D	AU	QR, perihelion distance	
LONGASC	1D	deg	OM, Longitude of ascending node (in degrees)	
OMEGA	1D	deg	W, Angle from asc. node to perihelion	
INCLINAT	1D	deg	IN, Inclination	
ECCENTR	1D	-	EC, Eccentricity	
ORBEPOCH	1D	Julian days	EPOCH, Epoch of orbital elements	
ORBEQNOX	1D	years	Elements equinox: J2000.0 or B1950.0	
DISTANCE	1D	AU	Geocentric Distance	
SCANTYPE	20A	-	Scan astronomical type	One of POINT, FOCUS, CAL, FLUXCAL, SKYDIP, ON, ONOFF, MAP, UNKNOWN,. . . The focus direction should be coded like FOCUS-X, FOCUS-Y, FOCUS-Z, FOCUS-XTILT, FOCUS-YTILT, FOCUS-ZTILT since the SCANDIR keyword refers to celestial coordinates. NB: Any of the following SCANxxxx parameters can change from subscan to subscan move to the DATAPAR header and should be looked for there. Particularly SCANDIR often changes between subscans. The parameters should be included as needed depending on SCANMODE. See text.
SCANMODE	20A	-	Mapping mode	including SAMPLE, RASTER, OTF, DRIFT, NONE, UNKNOWN.
SCANGEOM	20A	-	Scan geometry	including SINGLE, LINE, CROSS, RECT, QUAD, CIRC, CURVE, UNKNOWN.
SCANDIR	4A	-	(optional) scan direction	described as USER (user native frame) or xLON/xLAT as in CTYPEj (standard basis system), including ALON/ALAT for Az or El scanning. Effelsberg is using XLIN, YLIN, ZLIN, XROT, YROT and ZROT to indicate the focus direction since they perform OTF focus scans. This should be rather coded in SCANTYPE as described above.
SCANLINE	1J	-	(optional) number of lines in a scan. Default 1.	
SCANRPTS	1J	-	(optional) number of repeats of each scan line. Default 1.	
SCANLEN	1D	user-defined	(optional, OTF/RASTER) line length	For OTF, the line length or turn angle (SCANGEOM=CIRCLE) in Deg; for RASTER, the number of samples in a line.

SCANXVEL	1D	user-defined	(optional, OTF) tracking rate along line.	Units depend on SCANMODE definition.
SCANTIME	1D	user-defined	(optional, OTF) time for one line	
SCANXSPC	1D	user-defined	(optional, RASTER) step along line between samples	
SCANYSPC	1D	user-defined	(optional, OTF/RASTER) step between scan/raster lines	
SCANSKEW	1D	user-defined	(optional, OTF/RASTER) offset in scan direction between lines	
SCANPAR1	1D	user-defined	(optional) spare scan parameter	for modes I haven't thought of...
SCANPAR2	1D	user-defined	(optional) another spare scan parameter	SWTCHMOD has moved to the FEBEPAR header, as it is receiver-dependent
CROCYCLE	20A	-	CAL/REF/ON loop string	showing how often to go to CAL and REF, e.g. CROOCOO is a REF every four ONs and CAL every two ONs – see 30m NCS documentation.
ZIGZAG	1L	-	(optional, OTF/RASTER) Scan in zigzag?	TRUE if alternate lines traced in opposite directions, FALSE if all lines traced in same direction.
TRANDIST	1D	m	(optional, HOLO) Holography transmitter distance	If OBJECT=*HOLO*, these three special holography keywords are stored.
TRANFREQ	1D	Hz	(optional, HOLO) Holography transmitter frequency	
TRANFOCU	1D	deg	(optional, HOLO) transmitter offset from prime focus	
WOBUSED	1L	-	Wobbler used?	Wobbler parameters apply to all receivers.
WOBTHROW	1D	deg	wobbler throw	can also be used for beam switching
WOBDIR	4A	-	wobbler throw direction	Described as USER (user native frame) or xLAT or xLON, inc. ALON or ALAT for Az or El scanning.
WOBCYCLE	1E	s	wobbler period	
WOBMODE	20A	-	wobbler mode (SQUARE/TRIANGULAR)	
WOBPATT	10A	-	Wobbler pattern (e.g. NEG,POS,SYM)	
PHASEn	20A	-	Phase n description where n is an integer 1 to NPHASES	
NFEBE	1J	-	$N_{\rm FEBE}$ number of FEBEs	Previously N_FEBE
PDELTAIA	1E	deg	Accumulated user ptg correction IA	
PDELTACA	1E	deg	Accumulated user ptg correction CA	
PDELTAIE	1E	deg	Accumulated user ptg correction IE	
FDELTAIA	1E	deg	Accumulated ptg corr to IA due to focus	

FDELTACA	1E	deg	Accumulated ptg corr to CA due to focus	
FDELTAIE	1E	deg	Accumulated ptg corr to IE due to focus	
IA	1E	deg	Pointing Coefficient (-P1)	(note change in sign from v.1.54) For a full description of the pointing terms see Sect. 15.5.8.
IE	1E	deg	Pointing Coefficient (P7)	
HASA	1E	deg	Pointing Coefficient	
HACA	1E	deg	Pointing Coefficient	
HESE	1E	deg	Pointing Coefficient	(was ZFLX in v.1.54)
HECE	1E	deg	Pointing Coefficient (P8)	(was ECEC in v.1.54)
HESA	1E	deg	Pointing Coefficient	
HASA2	1E	deg	Pointing Coefficient	
HACA2	1E	deg	Pointing Coefficient	
HESA2	1E	deg	Pointing Coefficient	
HECA2	1E	deg	Pointing Coefficient	
HACA3	1E	deg	Pointing Coefficient	
HECA3	1E	deg	Pointing Coefficient	
HESA3	1E	deg	Pointing Coefficient	
NPAE	1E	deg	Pointing Coefficient (-P3)	(note change in sign from v.1.54)
CA	1E	deg	Pointing Coefficient (-P2)	(note change in sign from v.1.54)
AN	1E	deg	Pointing Coefficient (-P5)	(note change in sign from v.1.54)
AW	1E	deg	Pointing Coefficient (-P4)	
HECE2	1E	deg	(Opt.) Pointing Coefficient	
HECE6	1E	deg	(Opt.) Pointing Coefficient	
HESA4	1E	deg	(Opt.) Pointing Coefficient	
HESA5	1E	deg	(Opt.) Pointing Coefficient	
HSCA	1E	deg	(Opt.) Pointing Coefficient	
HSCA2	1E	deg	(Opt.) Pointing Coefficient	
HSSA3	1E	deg	(Opt.) Pointing Coefficient	
HSCA5	1E	deg	(Opt.) Pointing Coefficient	
NRX	1E	deg	(Opt.) Pointing Coefficient	
NRY	1E	deg	(Opt.) Pointing Coefficient	
HYSA	1E	deg	(Opt.) Pointing Coefficient for Azimuth Hysteresis (used in Effelsberg)	
HYSE	1E	deg	(Opt.) Pointing Coefficient for Elevation Hysteresis (used in Effelsberg)	
SETLINX	1D	m	(Opt.) Focus X linear zero position (used in Effelsberg)	

SETLINY	1D	m	(Opt.) Focus Y linear zero position (used in Effelsberg)
SETLINZ	1D	m	(Opt.) Focus Z linear zero position (used in Effelsberg)
SETROTX	1D	deg	(Opt.) Focus X rotational zero position (used in Effelsberg)
SETROTY	1D	deg	(Opt.) Focus Y rotational zero position (used in Effelsberg)
SETROTZ	1D	deg	(Opt.) Focus Z rotational zero position (used in Effelsberg)
MOVEFOC	1L	-	(Opt.) Harmonic oscillation of focus? (used in Effelsberg)
FOCAMP	1D	m	(Opt.) Harmonic oscillation focus amplitude (used in Effelsberg)
FOCFREQ	1D	Hz	(Opt.) Harmonic oscillation focus frequency (used in Effelsberg)
FOCPHASE	1D	deg	(Opt.) Harmonic oscillation focus phase (used in Effelsberg)

17.3.2 SCAN-MBFITS Binary Table Columns

The scan table just contains a list of FEBEs.

Keyword	Type	Units	Description	Comments
FEBE	17A	-	Frontend-backend combination identification	format: <FE>–<BE> where FE and BE are 8-letter identifiers

17.4 The FEBEPAR-MBFITS Binary Table

The FEBEPAR table is stored per FEBE per scan and contains the frontend-backend setup. Parameters common to all FEBEs are in the SCAN table. Includes:

- FEBE setup: number of pixels, polarisations and basebands
- Pointing coefficients specific to this FE
- Calibration parameters specific to this FEBE

A note on feed/pixel counting:
An array has a certain number of pixels/elements/feeds, each of which has associated polarisation, offset and other properties. For any one scan, only a subset of these may be in use (or some may be used twice). In order that the arrays storing the fixed properties (polarisation etc.) remain the same for each use of the array, but to minimise data storage when only a subset is in use, we differentiate between the number of feeds on the array (FEBEFEED = total number of array elements) and the number of feeds in use for each baseband (USEFEED).
A receiver outputting two polarised feeds is equivalent to an 'array' with two 'pixels': the polarisations are then stored in POLTY and the feeds (here polarisations) in use in USEFEED.

17.4.1 FEBEPAR-MBFITS Binary Table Header Keywords

Keyword	Type	Units	Description	Comment
EXTNAME	20A	-	'FEBEPAR-MBFITS'	
FEBE	17A	-	Frontend-backend combination identification.	format: <FE>–<BE> where FE and BE are 8-letter identifiers
SCANNUM	1J	-	Scan number	
DATE-OBS	23A	-	observing date (Y2K format with time) in TIMESYS system (scan start)	
DEWCABIN	70A		Dewar cabin: CASS_C or NASMYTH_A or NASMYTH_B	
DEWRTMOD	5A	-	Dewar tracking system	CABIN = fixed in Nasmyth/Cassegrain system, EQUA: RA/DEC; HORIZ: Az/El
DEWUSER	1E	Deg	Dewar user angle	Fixed in DEWRTMOD system, measured counterclockwise from vertical. Fixed throughout a scan.
DEWZERO	1E	Deg	Dewar coordinate offset angle relative to the cabin system measured counterclockwise from vertical.	
LOCATION	70A	-	(Opt.) Dewar location description (e.g. A-PI2)	

OPTPATH	70A	-	(Opt.) Optical path description (e.g. SABOCA:OPTICS->STOWED OPTICS:C->C OPTICS:C2->PI)	
NOPTREFL	1J	-	(Opt.) Number of optical reflections to calculate array derotation	
FEBEBAND	1J	-	$N_{\rm BD}$ maximum number of configurable basebands for this febe	Frequency bands
FEBEFEED	1J	-	$N_{\rm FD}$ total number of feeds	FEBEFEED stores the total number of feeds for the receiver in use. A receiver outputting two polarisations counts as two feeds. For an array, count the total no. of pixels, even if not all in use.
NUSEBAND	1J	-	$N_{\rm USBD}$ Number of bands in use.	Determines the number of ARRAYDATA tables and the dimensions of the calibration arrays in the FEBEPAR table.
FDTYPCOD	70A	-	Feed type code definition	Defines the numerical codes used in FEEDTYPE - e.g. 1:HETERODYNE, 1:AC 2:DC 3:UNKNOWN
FEGAIN	E	-	Frontend amplifier gain	
SWTCHMOD	20A	-	Switch mode	TOTP, FSW, BEAMSW, HORNSW, LOADSW, CAL... Moved from SCAN header, as receiver-dependent - apart from WOBSW, which is covered by WOBUSED in SCAN header
NPHASES	1J	-	no. of switch phases in a switch cycle	FS/wobbler, was N_SWITCH pre v.1.5
FRTHRWLO	1E	Hz	Phase 1 frequency switching throw	
FRTHRWHI	1E	Hz	Phase 2 frequency switching throw	
TBLANK	1E	s	Blank time of backend	
TSYNC	1E	s	Sync. time of backend	
IARX	1E	deg	Pointing Coefficient (receiver), adds to IA	Receiver-specific pointing coefficients
IERX	1E	deg	Pointing Coefficient (receiver), adds to IE	
HASARX	1E	deg	Pointing Coefficient (receiver), adds to HASA	
HACARX	1E	deg	Pointing Coefficient (receiver), adds to HACA	
HESERX	1E	deg	Pointing Coefficient (receiver), adds to HESE	
HECERX	1E	deg	Pointing Coefficient (receiver), adds to HECE	
HESARX	1E	deg	Pointing Coefficient (receiver), adds to HESA	
HASA2RX	1E	deg	Pointing Coefficient (receiver), adds to HASA2	
HACA2RX	1E	deg	Pointing Coefficient (receiver), adds to HACA2	
HESA2RX	1E	deg	Pointing Coefficient (receiver), adds to HESA2	
HECA2RX	1E	deg	Pointing Coefficient (receiver), adds to HECA2	
HACA3RX	1E	deg	Pointing Coefficient (receiver), adds to HACA3	
HECA3RX	1E	deg	Pointing Coefficient (receiver), adds to HECA3	
HESA3RX	1E	deg	Pointing Coefficient (receiver), adds to HESA3	

NPAERX	1E	deg	Pointing Coefficient (receiver), adds to NPAE	
CARX	1E	deg	Pointing Coefficient (receiver), adds to CA	
ANRX	1E	deg	Pointing Coefficient (receiver), adds to AN	
AWRX	1E	deg	Pointing Coefficient (receiver), adds to AW	
HECE2RX	1E	deg	(Opt.) Pointing Coefficient (receiver), adds to HECE2	
HECE6RX	1E	deg	(Opt.) Pointing Coefficient (receiver), adds to HECE6	
HESA4RX	1E	deg	(Opt.) Pointing Coefficient (receiver), adds to HESA4	
HESA5RX	1E	deg	(Opt.) Pointing Coefficient (receiver), adds to HESA5	
HSCARX	1E	deg	(Opt.) Pointing Coefficient (receiver), adds to HSCA	
HSCA2RX	1E	deg	(Opt.) Pointing Coefficient (receiver), adds to HSCA2	
HSSA3RX	1E	deg	(Opt.) Pointing Coefficient (receiver), adds to HSSA3	
HSCA5RX	1E	deg	(Opt.) Pointing Coefficient (receiver), adds to HSCA5	
NRXRX	1E	deg	(Opt.) Pointing Coefficient (receiver), adds to NRX	
NRYRX	1E	deg	(Opt.) Pointing Coefficient (receiver), adds to NRY	
SIG_ONLN	70A	-	(Opt.) Standard signal definition	Defines the combination of phases that are required to obtain the standard signal (e.g. for (feed,phase) combinations: '(1,1)+(1,2)-2*(1,3)+(1,4)')
REF_ONLN	70A	-	(Opt.) Standard reference definition	
SIG_POL	70A	-	(Opt.) Standard polarisation definition	
REF_POL	70A	-	(Opt.) Standard polarisation reference definition	

17.4.2 FEBEPAR-MBFITS Binary Table Columns

Multidimensional parameters that can't go in the FEBEPAR header. These include parameters for the whole array and for the subset which is in use.

Keyword	Type	Units	Description	Comments
USEBAND	N_{USBD} J	-	List of basebands which are in use	List of BASEBAND (ARRAYDATA header) for which data are stored.
NUSEFEED	N_{USBD} J	-	N_{USFD} Number of feeds in use for each baseband	So that we can minimally dimension the data storage array in each ARRAYDATA table.
USEFEED	1PJ($N_{\mathrm{USFD}} \times N_{\mathrm{USBD}}$)	-	List of feeds which are in use for each baseband	The data in each ARRAYDATA table are stored in the order listed in USEFEED.
BESECTS	1PJ($N_{\mathrm{USFD}} \times N_{\mathrm{USBD}}$)	-	List of backend sections connected to feeds in same order as listed in USEFEED / data storage in ARRAYDATA	
FEEDTYPE	1PJ($N_{\mathrm{USFD}} \times N_{\mathrm{USBD}}$)	-	feed type index	Numerical code to index various feed types in the array. The plain text explanations are given in the FDTYPCOD keyword.
				The following parameters depend on feed and (sometimes) frequency band, and are given for the whole array, not just the feeds in use.
FEEDOFFX	N_{FD} D	deg	feed x offset	x offset of each feed from rotation centre in DEWRTMOD system.
FEEDOFFY	N_{FD} D	deg	feed y offset	y offset of each feed from rotation centre in DEWRTMOD system.
REFFEED	1J	-	feed number of reference feed	we assume that it is always a physical feed- though this is not always the case, e.g. CHAMP - but not necessarily at the rotation centre (0,0)
POLTY	N_{FD} A	-	Feed polarisation type (X, Y, L, R)	Here is the polarisation of each feed for this FE
POLA	N_{FD} E	deg	Feed polarisation angle	depends on feed orientation, measured ccw from vertical in DEWRTMOD system
				The keywords ACCOUPLE and DCCOUPLE are unnecessary given FEEDTYPE and have been removed (v.1.2).
APEREFF	$N_{\mathrm{FD}} \times N_{\mathrm{USBD}}$ E	-	Aperture efficiency	
BEAMEFF	$N_{\mathrm{FD}} \times N_{\mathrm{USBD}}$ E	-	Beam efficiency	
ETAFSS	$N_{\mathrm{FD}} \times N_{\mathrm{USBD}}$ E	-	Forward efficiency	
HPBW	$N_{\mathrm{FD}} \times N_{\mathrm{USBD}}$ E	deg	Half-power beam width	
ANTGAIN	$N_{\mathrm{FD}} \times N_{\mathrm{USBD}}$ E	K/Jy	Antenna Gain	
TCAL	$N_{\mathrm{FD}} \times N_{\mathrm{USBD}}$ E	mK	Calibration temperature	Constant radio calibration factor.

BOLCALFC	N_{USBD} E	Jy/counts	Bolometer calibration factor	Set by planet/secondary calibrator observations. Previously BOLREFGN.
BEGAIN	N_{USBD} E		Backend reference gain/attenuation factor	reinstated. See Sect. 15.4.
FLATFIEL	$N_{\mathrm{FD}} \times N_{\mathrm{USBD}}$ E	-	Bolometer flat field (relative gains)	previously BOLFLAT but useful for heterodyne rx too.
BOLDCOFF	$N_{\mathrm{FD}} \times N_{\mathrm{USBD}}$ E	-	Bolometer DC offset	DC offset applied in amplifier to avoid saturation in the ADC.
GAINIMAG	$N_{\mathrm{FD}} \times N_{\mathrm{USBD}}$ E	-	(spectral line) Gain ratio image/signal sideband	
GAINELE1	N_{USBD} E	deg	Gain-elevation correction parameter 1	Gain-elevation correction parametrised as $\cos^2(\epsilon - p1) + p2\sin^2(\epsilon - p1)$ where $p1$,$p2$ are two wavelength-dependent parameters
GAINELE2	N_{USBD} E	-	Gain-elevation correction parameter 2	

17.5 The ARRAYDATA-MBFITS Binary Table

There is one `ARRAYDATA-MBFITS` table per baseband per FEBE. It stores the data description (header) and the data (table). Includes:

- Frequency band setup: frequency, channels, polarisations, line ID
- Data axes description

17.5.1 ARRAYDATA-MBFITS Binary Table Header Keywords

Keyword	Type	Units	Value	Description	Comments
`EXTNAME`	20A	-		‘ARRAYDATA-MBFITS’	
`FEBE`	17A	-		Frontend-backend combination ID	
`BASEBAND`	1J	–		Baseband number	
`SCANNUM`	1J	-		Scan number	
`SUBSNUM`	1J	-		Subscan number	
`OBSNUM`	1J	-		(Deprecated) Subscan number	The subscan number was originally named *observation* number in ALMA-TI-FITS and thus in MBFITS. However, nobody really liked the name and eventually ALMA renamed *observation* to *subscan* and MBFITS did too. The old keyword is kept for compatibility issues but should not be used in new data.
`DATE-OBS`	23A	-		Subscan start in `TIMESYS` system	Y2K format with time
`CHANNELS`	1J	-		N_{CH} Number of spectral channels for this baseband	This is reserved for spectral channels - in continuum data it is set to 1.
`NUSEFEED`	1J	-	N_{USFD}	Number of feeds in use for this baseband	
`FREQRES`	1D	Hz		Frequency resolution	Actual backend resolution, not channel width, which is stored in the axis description `11CD2F`.
`BANDWID`	1D	Hz		Bandwidth for this band in the sky/telescope frequency frame	
`MOLECULE`	20A	-		main line molecule (optional)	
`TRANSITI`	20A	-		main line transition (optional)	
`RESTFREQ`	1D	Hz		Rest frequency of line (optional)	
`SKYFREQ`	1D	Hz		Sky frequency of line including all velocity corrections	
`SIDEBAND`	3A	-		Main sideband is USB/LSB	

SBSEP	1D	Hz		Sideband separation including band offsets in the sky/telescope frequency frame	
2CTYP2	8A	-	'PIX-INDX'	Feed axis (in USEFEED array)	Axis (2) description for DATA
2CRPX2	1J	-	1	Ref. position = 1	The order of feeds is given by USEFEED
2CRVL2	1J	-	1	Feed index value at this position = 1	
21CD2A	1J	-	1	Feed index separation = 1	

Next we assign to the spectral axis a rest and image frequency, and (radio) velocity description relative to the rest frame, as well as a rest and image frequency description relative to the telescope frame. Other alternative axes descriptions could also be given.

WCSNM2F	8A	-	e.g. 'LsrkFreq'	Axis name	Frequency description in rest frame (e.g. LSR) for main sideband:
1CTYP2F	8A	-	'FREQ '	Frequency axis for col.2	Axis (1) description for DATA
1CRPX2F	1E	-		Ref. channel	
1CRVL2F	1D	Hz		Frequency at ref. channel in rest frame	
11CD2F	1D	Hz		Channel Separation	
1CUNI2F	8A	-	'Hz'	Unit	
1SPEC2F	8A	-	e.g. 'LSRK'	Standard of rest for frequencies	
1SOBS2F	8A	-	'TOPOCENT'	Observing frame	Ref. frame where spectral coord. has no spatial variation.
WCSNM2I	8A	-	e.g. 'LsrkFreq'	Axis name	Frequency description in rest frame (e.g. LSR) for image sideband:
1CTYP2I	8A	-	'FREQ '	Frequency axis for col.2	Axis (1) description for DATA
1CRPX2I	1E	-		Ref. channel	
1CRVL2I	1D	Hz		Frequency at ref. channel in rest frame	
11CD2I	1D	Hz		Channel Separation	
1CUNI2I	8A	-	'Hz'	Unit	
1SPEC2I	8A	-	e.g. 'LSRK'	Standard of rest for frequencies	
1SOBS2I	8A	-	'TOPOCENT'	Observing frame	Ref. frame where spectral coord. has no spatial variation.
WCSNM2S	8A	-	e.g. 'LsrkFreq'	Axis name	Frequency description in sky (telescope) frame for main sideband:
1CTYP2S	8A	-	'FREQ '	Frequency axis for col.2	Axis (1) description for DATA
1CRPX2S	1E	-		Ref. channel	
1CRVL2S	1D	Hz		Frequency at ref. channel in rest frame	
11CD2S	1D	Hz		Channel Separation	
1CUNI2S	8A	-	'Hz'	Unit	
1SPEC2S	8A	-	e.g. 'LSRK'	Standard of rest for frequencies	

1SOBS2S	8A	-	'TOPOCENT'	Observing frame	Ref. frame where spectral coord. has no spatial variation.
WCSNM2J	8A	-	e.g. 'LsrkFreq'	Axis name	Frequency description in sky (telescope) frame for image sideband:
1CTYP2J	8A	-	'FREQ '	Frequency axis for col.2	Axis (1) description for DATA
1CRPX2J	1E	-		Ref. channel	
1CRVL2J	1D	Hz		Frequency at ref. channel in rest frame	
11CD2J	1D	Hz		Channel Separation	
1CUNI2J	8A	-	'Hz'	Unit	
1SPEC2J	8A	-	e.g. 'LSRK'	Standard of rest for frequencies	
1SOBS2J	8A	-	'TOPOCENT'	Observing frame	Ref. frame where spectral coord. has no spatial variation.
WCSNM2R	8A	-	e.g. 'LsrkVRad'	Axis name	Velocity description in rest frame (e.g. LSR) for heterodyne receivers.
1CTYP2R	8A	-	e.g. 'VRAD '	Velocity axis for col.2	Axis (1) description for DATA
1CRPX2R	1E	-		Ref. channel	
1CRVL2R	1D	km/s		Velocity at ref. channel	
11CD2R	1D	km/s		Velocity Channel Separation	
1CUNI2R	8A	-	'km/s'	Unit	
1SPEC2R	8A	-	e.g. 'LSRK'	Standard of rest frame for velocities	
1SOBS2R	8A	-	'TOPOCENT'	Observing frame	Ref. frame where spectral coord. has no spatial variation.
1VSOU2R	1E	km/s		Source velocity in rest frame	
1VSYS2R	1E	km/s		Observer vel. in rest frame	in direction of observation, at subscan start

17.5.2 ARRAYDATA-MBFITS Binary Table Columns

Keyword	Type	Units	Description	Comments
MJD	1D	day	MJD at integration midpoint in TIMESYS system	
DATA	$N_{\mathrm{CH}} \times N_{\mathrm{USFD}}$ E	-	Data	USEFEED gives the number of feeds in use. Data is listed in feed order following the list in the FEBEPAR table.

17.6 The MONITOR-MBFITS Binary Table

This table stores raw monitoring data (real-time updates other than the backend data) at its natural rate, ie. not synchronised to backend dump times. The monitor data are stored as time-keyword-units-values. The update intervals for any monitor stream are thus fully flexible.

We anticipate input from different sources at different rates: a fast stream of telescope positions, meteorological and water vapour monitor readouts at slower rates, intermittent readings from CAL observations, etc. Input to monitor also includes calculated values such as system and receiver temperatures from CALs, opacities from skydips, other calibration factors.

We recommend that the telescope control system should call for updates on monitor points at the beginning and end of scans. As many of these as possible should be measured at these times. For points where a new measurement is not possible - e.g. CAL data - the last measurement should be resaved in the MONITOR table with its original timestamp. In this way, interpolation between points to fill in the DATAPAR table will be possible even without access to previous/later scan data.

MONITOR table updates:

- At the beginning/ends of scans: calibration data, pointing data, radiometer data, environmental data.
- At the beginning of integrations: frequencies, current real positions.
- At the end of observations: current real positions.

To avoid gaps in coverage, filling of the MONITOR table for one subscan should not stop when data taking stops. The MONITOR table should be closed either when a new subscan starts (and a new MONITOR table is opened) or when a scan ends.

As of v.1.2, the monitor point values are stored as a variable length array (Cotton, Tody & Pence 1995 appendix A). Most entries group naturally into small arrays e.g. 2 encoder readings, 2 temperatures and 3 powers from a calibration. We store one time, description and units for each array. Large arrays, such as the spectral line gain array, can also be stored.

With variable length arrays, almost any information stream can be routed to MONITOR. This gives MBFITS the flexibility to cope with future unforeseen requirements without making changes to the existing table structure.

The monitor units are also stored as variable length arrays (character strings). A units string only has to be written for the first occurrence of the monitor point in the table. Later appearances of the same monitor point can give a pointer to the units array first written.

Anticipated monitor points include:

- Real-time telescope position
- Real-time frequencies from frontends
- Focus measurements
- Dewar angle measurements
- Total power in each baseband
- Environmental measurements
- Calibration scans
- Water vapour monitor results
- Current pointing and refraction corrections

Groups of associated values which are available at the same time should be stored as one entry in MONITOR. Guidelines for descriptions of a group where naturally each element would have its own description are to give (optionally) a general description plus identifiers for each element separated by underscores. Examples: `FOCUS_X_Y_Z`, `WIND_SPEED_DIR`.

17.6.1 `MONITOR-MBFITS` Table Header Keywords

Keyword	Type	Units	Description	Comments
`EXTNAME`	20A	-	'MONITOR-MBFITS'	
`SCANNUM`	1J	-	Scan number	
`SUBSNUM`	1J	-	Subscan number	
`OBSNUM`	1J	-	(Deprecated) Subscan number	The subscan number was originally named *observation* number in ALMA-TI-FITS and thus in MBFITS. However, nobody really liked the name and eventually ALMA renamed *observation* to *subscan* and MBFITS did too. The old keyword is kept for compatibility issues but should not be used in new data.
`DATE-OBS`	23A	-	observing date in `TIMESYS` system at monitor point	
`USRFRAME`	20A	-	Human-readable description of user frame	Reflects the 3-tier setup of source, pattern offset and pattern frames which can be different combinations of astronomical and horizontal systems. For equatorial coordinates the possible values are `'EQEQEQ'`, `'EQHOEQ'`, `'EQEQHO'`, `'EQHOHO'` and `'HOHOHO'`. For galactic coordinates one would use `'GA'` instead of `'EQ'`. This setup may change per subscan (e.g. if the system for the REF offset differs from the pattern setup).

17.6.2 `MONITOR-MBFITS` Binary Table Columns

Keyword	Type	Description	Comments
`MJD`	1D	day	Scan date/time (Modified Julian Date) in `TIMESYS` system.

MONPOINT	A30	Monitor point description	The reason for long (A30) monitor points is to accommodate FEBEPAR descriptions (A17) and band numbers (2 digits) where necessary. E.g. TAUZEN_<febe>_<band>
MONVALUE	1PD(maxelem)	Pointer to monitor values (doubles)	which are stored in a heap at the end of the table.
MONUNITS	1PA(maxelem)	Pointer to units for monitor point (ascii)	which are stored in a heap at the end of the table. Where a monitor point stores more than one value, units are separated by semicolons e.g. Pa;K;Deg. Repeating units can be stored e.g. 3*K for three temperatures.

17.6.3 Anticipated monitor points

Keyword	Elements	Units	Description	Comments
Current real positions				
ENCODER_AZ_EL	2D	deg;deg	Encoder azimuth and elevation	The raw drive positions above are not available for the APEX antenna.
TRACKING_AZ_EL	2D	deg;deg	Average tracking azimuth and elevation	
INCLINOMETER_1_2	2D	deg;deg	Inclinometer 1 and 2	
FOCUS_X_Y_Z	3D	mm;mm;mm	Focus (absolute subreflector position) X, Y, Z	including elevation dependent correction, receiver dependent offsets (FOCOBS_X_Y_Z), and subscan dependent delta offsets (DFOCUS_X_Y_Z)
DFOCUS_X_Y_Z	3D	mm;mm;mm	Focus delta offsets for the subscan X, Y, Z	e.g. during a focus scan
PHI_X_Y_Z	3D	deg;deg;deg	Phi (absolute subreflector rotation) X,Y,Z	including elevation dependent correction, receiver dependent offset (PHIOBS_X_Y_Z) and subscan dependent delta offsets (DPHI_X_Y_Z)
DPHI_X_Y_Z	3D	deg;deg;deg	Phi delta offsets for the subscan X, Y, Z	e.g. during a focus scan
ANTENNA_AZ_EL	2D	deg;deg	Pointing-corrected antenna azimuth – real-time position	
WOBDISPL	1D	deg	wobbler displacement	Dewar angle has moved to FEBEPAR header suggested for MAMBO117. Switch beams using the wobbler but read out data at intermediate positions, storing the displacement here.

Frequencies labelled by frontend				
GUNF_<fe>	1D	Hz	Gunn frequency	Frequencies can change rapidly e.g. frequency switching, Doppler tracking
PLLF_<fe>	1D	Hz	Phase locked loop frequency	
LOFREQ_<febe>	$N_{\rm LO}$D	Hz	LO frequencies from frontend/LO chain	
LOSIDEBAND_<febe>	$N_{\rm LO}$D	Hz	LO sidebands from frontend/LO chain	
NUMIFCONV_<febe>	J	-	Number of IF conversions	
Environmental parameters				
TSTRUCT	nD	K	Structural temperature(s)	
WIND_SPEED_DIR	2D	m/s;deg	Wind speed and direction	
TAMB_P_HUMID	3D	C;hPa;%	Ambient conditions triplet: temperature, pressure, humidity	
TCABIN	1D	K	Receiver Cabin temp.	
TDEWAR	1D	K	Receiver Dewar temp.	
Calibration				
TP_<febe>	$N_{\rm USBD}$ D	Counts	Total Power in each baseband (per FEBE)	
				The following update each spectral line calibration scan
THOTCOLD_<cabin>	2D	K;K	Measured chopper temperature, Cold load temperature	
THOTCOLD_<febe>_<band>	2D	K;K	Corrected temperatures from calibration loads at receiver frequency	e.g. corrected for losses in internal calibration unit
PHOT_<febe>_<band>	$N_{\rm USFD}$ D	counts	Total power on chopper	band-averaged, for each feed in use
PCOLD_<febe>_<band>	$N_{\rm USFD}$ D	counts	Total power on cold load	band-averaged, for each feed in use
PSKY_<febe>_<band>	$N_{\rm USFD}$ D	counts	Total power on sky	band-averaged, for each feed in use
				TRX,TSYS,TSYSIMAG and TCAL update more frequently than cal scans if a radiometer and an atmospheric model are in use. Stored for each feed for each band.
TRX_<febe>_<band>	$N_{\rm USFD}$ D	K	receiver temperature	
TSYS_<febe>_<band>	$N_{\rm USFD}$ D	K	system temperature	
TSYSIMAG_<febe>_<band>	$N_{\rm USFD}$ D	K	image band system temperature	
TCAL_<febe>_<band>	$N_{\rm USFD}$ D	K	calibration temperature	
			Optical depths may update between calibrations if using a radiometer	

TAUZEN_<febe>_<band>	1D	-	Zenith opacity at band centre	
TAU_<febe>_<band>	1D	-	Main band opacity at current elevation	
TAUIMAGE_<febe>_<band>	1D	-	Image band opacity at current elevation	
GAINARRAY_<febe>_<band>	$N_{\mathrm{USFD}}\times N_{\mathrm{CH}}$ D	-	Spectral line gain array	For each feed in use. order: all channels for first feed, then second feed, etc. No. of channels, bandwidth etc. is in ARRAYDATA header for this subscan.
OBSFL_<febe>_<band>	3D	counts	Calibrator measured flux	from continuum calibrator obs. (SCANTYPE=FLUXCAL)
CALFL_<febe>_<band>	3D	Jy	Calibrator predicted flux	From calibrator flux model (SCANTYPE=FLUXCAL)
CALFAC_<febe>_<band>	3D	Jy/Counts	Counts-to-Jy calibration factor	From continuum calibrator obs. (SCANTYPE=FLUXCAL)
Water vapour radiometer				
TSYS_TRX_RD<freq>	2D	K;K;K	radiometer system and receiver temperatures	
CORCF_VALID_RD<freq>	2D	-;-	correlation coefficient for fit to radiometer data and validity of radiometric correction (was fit good?)	
TAU_WPATH_RD<freq>	2D	-;mm	opacity and water vapour column	at radiometer frequency
Observer Pointing				
IAOBS_CAOBS_IEOBS	3D	arcsec;arcsec;arcsec	Azimuth pointing offset from pointing obs. to add to IA (-P1), CA (-P2) and IE (P7)	in 30m terms, IAOBS = NULA, CAOBS = COL*, IEOBS = NULE
FOCOBS_X_Y_Z	3D	mm;mm;mm	receiver dependent X, Y, Z focus offsets	(see also FOCUS_X_Y_Z and DFOCUS_X_Y_Z)
PHIOBS_X_Y_Z	3D	deg;deg;deg	receiver dependent X, Y, Z subreflector rotation offsets	(see also PHI_X_Y_Z and DPHI_X_Y_Z)
REFRACTIO	1D	arcsec	Refraction correction (current), calculated from HUMIDITY, TAMBIENT, and PRESSURE as a function of elevation.	

17.7 The DATAPAR-MBFITS Binary Table

The DATAPAR table contains data-associated parameters which change with integration, but not the data itself (in ARRAYDATA). There is one DATAPAR table per FEBE. Parameters common to all subscans are in the SCAN table, and the FEBE setup is in the FEBEPAR table (assumed not to change between subscans).
DATAPAR includes:

- Time and coordinate information specific to this subscan and integration
- interpolated data from the MONITOR table, resampled to the timestamps of the *midpoints* of the integrations, as given by `MIDTIME`.

In most cases, there will be one DATAPAR row per ARRAYDATA entry, that is, per integration. In this case the order of integrations will be the same as in the corresponding ARRAYDATA table and `NINTS`= 1. In the case of fast data rates or data with several simultaneous phases, the `DPBLOCK` flag in the DATAPAR header can be set to `TRUE` and `DATAPAR` table rows can be written at less frequent intervals. In this case, each row in DATAPAR corresponds to `NINTS` rows in ARRAYDATA.
When blocking, all DATAPAR values refer to the first integration of the block. An integration for which no direct entry in DATAPAR exists should interpolate between times and positions from bracketing entries (or extrapolate if for the last blocked entry).
As of v.1.2, DATAPAR is the only additional table associated with each ARRAYDATA data table. What previously were INTMON and DATAPAR have been combined. Originally, the two were separated because the information in DATAPAR can be written directly at the time of an integration, whereas the INTMON values are interpolated/calculated from information in MONITOR which is not available until the end of a subscan. By buffering the DATAPAR entries until the end of the subscan, when the MONITOR values are available) all the data-associated parameters can be written together, saving on duplication in table and header. This change forces the positions etc. which were previously in INTMON to be written in quasi-real-time rather than filled in later offline, but it has become clear that this is necessary anyway to maintain the data flow .
NB All values in DATAPAR **are interpolated to the midpoint of the integration.**

17.7.1 `DATAPAR-MBFITS` Binary Table Header Keywords

Keyword	Type	Units	Description	Comments
`EXTNAME`	20A	-	‘DATAPAR-MBFITS’	
`SCANNUM`	1J	-	Scan number	
`SUBSNUM`	1J	-	Subscan number	
`OBSNUM`	1J	-	(Deprecated) Subscan number	The subscan number was originally named *observation* number in ALMA-TI-FITS and thus in MBFITS. However, nobody really liked the name and eventually ALMA renamed *observation* to *subscan* and MBFITS did too. The old keyword is kept for compatibility issues but should not be used in new data.
`DATE-OBS`	23A	-	Subscan start in `TIMESYS` system	(Y2K format with time)
`FEBE`	17A	-	FEBE descriptor	

LST	1D	s	Local apparent sidereal time (obs. start)	
SUBSTYPE	4A	-	Subscan type	Calibration (CAL), reference position (REF), or on-source (ON), or in the case where a calibration is split into multiple subscans, HOT/COLD/SKY.
OBSTYPE	4A	-	(Deprecated) Subscan type	Calibration (CAL), reference position (REF), or on-source (ON), or in the case where a calibration is split into multiple subscans, HOT/COLD/SKY.
SCANxxxx			(Opt.) Scan description	Any of the SCANxxxx parameters that change from subscan to subscan appear here rather than in the SCAN header. Particularly SCANDIR. These are only written to DATAPAR if their values for the current differ from the SCAN header entries.
CTYPE1N	8A	-	(Opt.) Native system type (longitude) – XLON-SFL / XLON-GLS	This is only written if the value for the current subscan differs from the SCAN header entry (e.g. for horizontal reference points in an equatorial map).
CTYPE2N	8A	-	(Opt.) Native system type (latitude) – XLAT-SFL / XLAT-GLS	This is only written if the value for the current subscan differs from the SCAN header entry (e.g. for horizontal reference points in an equatorial map).
DPBLOCK	1L	-	Data blocking?	TRUE if blocking, ie. DATAPAR parameters hold for multiple integrations, FALSE if there is a DATAPAR row for every integration from this FEBE.
NINTS	1J	-	N_{BLK} Integrations in block	greater than 1 if DPBLOCK true
OBSTATUS	10A	-	Subscan ok? OK/ABORT (if aborted).	The last integration of an aborted scan may be discarded
WOBCOORD	1L	-	Flag: Coordinates include wobbler offsets?	
PHASEn	20A	-	(Opt.) Phase n description where n is an integer 1 to NPHASES	These keywords are only written if the phase assignments for this subscan differ from the global setup in SCAN-MBFITS (e.g. for symmetric wobbling)
DEWANG	1E	deg	Current (measured) dewar angle	
DEWEXTRA	1E	deg	Extra dewar angle for symmetry purposes	

17.7.2 DATAPAR-MBFITS Binary Table Columns

Keyword	Type	Units	Description	Comments
MJD	1D	Julian day	MJD at integration midpoint in TIMESYS system	

LST	1D	s	Local apparent sidereal time at integration midpoint	
INTEGTIM	1D	s	Integration time	
PHASE	1J	-	1-based phase number	v.1.57 Renamed from ISWITCH and changed type to 1J to increase efficiency when writing binary table. Blocked Effelsberg phases can still be indicated as e.g. numeric 1234.
LONGOFF	1D	deg	long. offset from source in user native frame (except for horizontal patterns (see section 15.5.7)) intermediate pixel coords (x)	absolute coords of integration in user native frame are $\phi = \mathtt{LONGOBJ} + \mathtt{LONGOFF}\sec(\mathtt{LATOBJ} + \mathtt{LATOFF})$
LATOFF	1D	deg	lat. offset from source in user native frame (except for horizontal patterns (see section 15.5.7)) intermediate pixel coords (y)	$\theta = \mathtt{LATOBJ} + \mathtt{LATOFF}$
AZIMUTH	1D	deg	Azimuth (derived) inc. wobbler offsets	
ELEVATIO	1D	deg	Elevation (derived) inc. wobbler offsets	Note that this column is usually **not** corrected for refraction when using the horizontal basis system since those scans are usually meant for terrestrial observations (e.g. of a holography transmitter) where the correction would not make sense.
RA	1D	deg	Mean right ascension inc. wobbler offsets	
DEC	1D	deg	Mean declination inc. wobbler offsets	
PARANGLE	1D	deg	Parallactic angle	
CBASLONG	1D	deg	Commanded long. in basis frame frame (defined by CTYPEjB), inc. wobbler offsets	
CBASLAT	1D	deg	Commanded lat. in basis frame frame (defined by CTYPEjB), inc. wobbler offsets	
BASLONG	1D	deg	Actual long. in basis frame frame, inc. wobbler offsets	
BASLAT	1D	deg	Actual lat. in basis frame frame, inc. wobbler offsets	
ROTANGLE	1D	deg	Angle of array, measured E wrt N in user native frame	Needed to calculate array offset positions.

MCRVAL1	1D	deg.	(opt.) Native frame zero in basis system (long.)	Frame tracking for moving bodies and az/el mapping – see Sect. 15.5.6 and 15.5.7 – these columns exist if MOVEFRAM in the SCAN header is TRUE.
MCRVAL2	1D	deg.	(opt.) Native frame zero in basis system (lat.)	
MLONPOLE	1D	deg.	(opt.) longitude of basis celestial pole in body system	
MLATPOLE	1D	deg.	(opt.) basis latitude of body frame pole	
DFOCUS_X	1D	mm	(opt.) Commanded X focus offset (subreflector position)	For SCANTYPE = FOCUS. Commanded offset, from MONITOR focus offset positions DFOCUS_X.
DFOCUS_Y	1D	mm	(opt.) Commanded Y focus offset (subreflector position)	For SCANTYPE = FOCUS. Commanded offset, from MONITOR focus offset positions DFOCUS_Y.
DFOCUS_Z	1D	mm	(opt.) Commanded Z focus offset (subreflector position)	For SCANTYPE = FOCUS. Commanded offset, from MONITOR focus offset positions DFOCUS_Z.
DPHI_X	1D	deg	(opt.) Commanded Phi-X offset (subreflector rotation)	”
DPHI_Y	1D	deg	(opt.) Commanded Phi-Y offset (subreflector rotation)	”
DPHI_Z	1D	deg	(opt.) Commanded Phi-Z offset (subreflector rotation)	”
WOBDISLN	1D	deg	(opt.) Wobbler displacement along wobbler coordinate system longitude axis	
WOBDISLT	1D	deg	(opt.) Wobbler displacement along wobbler coordinate system latitude axis	

17.8 The GROUPING Binary Table

Stored every scan when using the FITS hierarchical grouping standard to save the MBFITS data (see sections 5). The GROUPING table is used store the locations of the member files and some details such as FEBE name and baseband number which can be used to speed up searching when reading the files. Note that the primary header as defined in section 17.2 is written to the GROUPING file if it is being used.

17.8.1 GROUPING Binary Table Header Keywords

Keyword	Type	Units	Description	Comments
EXTNAME	20A	-	‘GROUPING’	
SCANNUM	1J	-	Scan number	
DATE-OBS	23A	-	Scan start in TIMESYS system	(Y2K format with time)

17.8.2 GROUPING Binary Table Columns

Keyword	Type	Units	Description	Comments
MEMBER_POSITION	1J	-	HDU number of MBFITS table in member file	Default column name defined by hierarchical grouping standard
MEMBER_LOCATION	256A	-	Member file name relative to the top-level directory	Default column name defined by hierarchical grouping standard
MEMBER_URI_TYPE	3A	-	Member URI type (e.g. ”URL”)	Default column name defined by hierarchical grouping standard
EXTNAME	30A	-	MBFITS extension name that is stored in the member file	
SUBSNUM	1J	-	Member file’s subscan number if applicable (else -999). Used for faster searching.	
FEBE	30A	-	Member file’s FEBE name if applicable else blank. Used for faster searching.	
BASEBAND	1J	-	Member file’s baseband number if applicable (else -999). Used for faster searching.	

17.8.3 Member File Primary Header Keywords

In order to be able to identify the relationship between member files the hierarchical grouping standard (see Jennings et al. 1997) adds a keyword to point to the respective GROUPING table. In addition to that, we added one keyword in the primary headers of all member files to store the MBFITS dataset name:

Keyword	Type	Units	Description	Comments
`MBFITS`	40A	-	MBFITS dataset name	