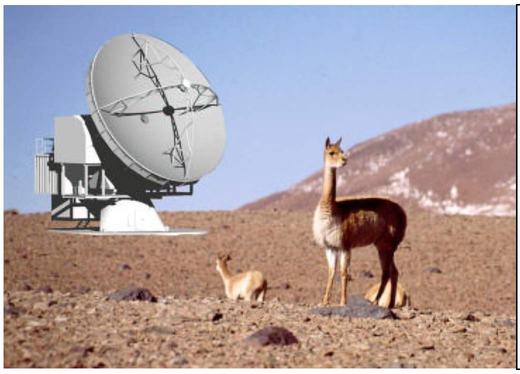

Atacama Pathfinder EXperiment

APEX-IFD-MPI-0002

Revision: 1.2

April 10, 2003

Interface Description

Jennifer Hatchell

Multi-Beam FITS Raw Data Format

Interface Description

Jennifer Hatchell and Dirk Muders
hatchell, dmuders@mpifr-bonn.mpg.de

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

Keywords: Data Format, FITS, APEX

Author Signature: Jennifer Hatchell	Date: April 10, 2003
Approved by: NN Institute: MPIfR	Signature: Date:
Released by: Institute:	Signature: Date:

Contents

1 **Introduction** 5
- 1.1 MBFITS and ALMA-TI FITS 5
- 1.2 Scans, observations and integrations 5

2 **What's new in v.1.2?** 7

3 **Explanatory notes** 9
- 3.1 Scanning/mapping description 9
- 3.2 Phase description for switched observations. 10
- 3.3 Calibration parameters 11
  - 3.3.1 Continuum calibration 11
  - 3.3.2 Spectral line calibration 12
- 3.4 DATE-OBS 13
- 3.5 WCS coordinates 13
  - 3.5.1 Sources of spatial coordinate information 13
  - 3.5.2 WCS representation and telescope control system input 14
  - 3.5.3 WCS coordinate systems 15
  - 3.5.4 30m parameters 16
  - 3.5.5 WCS parameters (including derivation from 30m parameters) 16
  - 3.5.6 Moving bodies 19
  - 3.5.7 Observations in Az/El, On-the-fly mapping in Az/El 19
  - 3.5.8 Pointing 20
  - 3.5.9 WCS spectral coordinates 21

4 **References** 21

5 **MBFITS specification** 22
  - 5.0.10 Data types 22
- 5.1 The Primary header 22
- 5.2 The `SCAN-MBFITS` Binary Table 23
  - 5.2.1 `SCAN-MBFITS` Binary Table Header Keywords 23
  - 5.2.2 `SCAN-MBFITS` Binary Table Columns 26
- 5.3 The `FEBEPAR-MBFITS` Binary Table 27
  - 5.3.1 `FEBEPAR-MBFITS` Binary Table Header Keywords 27
  - 5.3.2 `FEBEPAR-MBFITS` Binary Table Columns 28
- 5.4 The `ARRAYDATA-MBFITS` Binary Table 30
  - 5.4.1 `ARRAYDATA-MBFITS` Binary Table Header Keywords 30
  - 5.4.2 `ARRAYDATA-MBFITS` Binary Table Columns 31
- 5.5 The `MONITOR-MBFITS` Binary Table 32
  - 5.5.1 `MONITOR-MBFITS` Table Header Keywords 33
  - 5.5.2 `MONITOR-MBFITS` Binary Table Columns 33
  - 5.5.3 Anticipated monitor points 33
- 5.6 The `DATAPAR-MBFITS` Binary Table 36
  - 5.6.1 `DATAPAR-MBFITS` Binary Table Header Keywords 36
  - 5.6.2 `DATAPAR-MBFITS` Binary Table Columns 37

Abstract

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

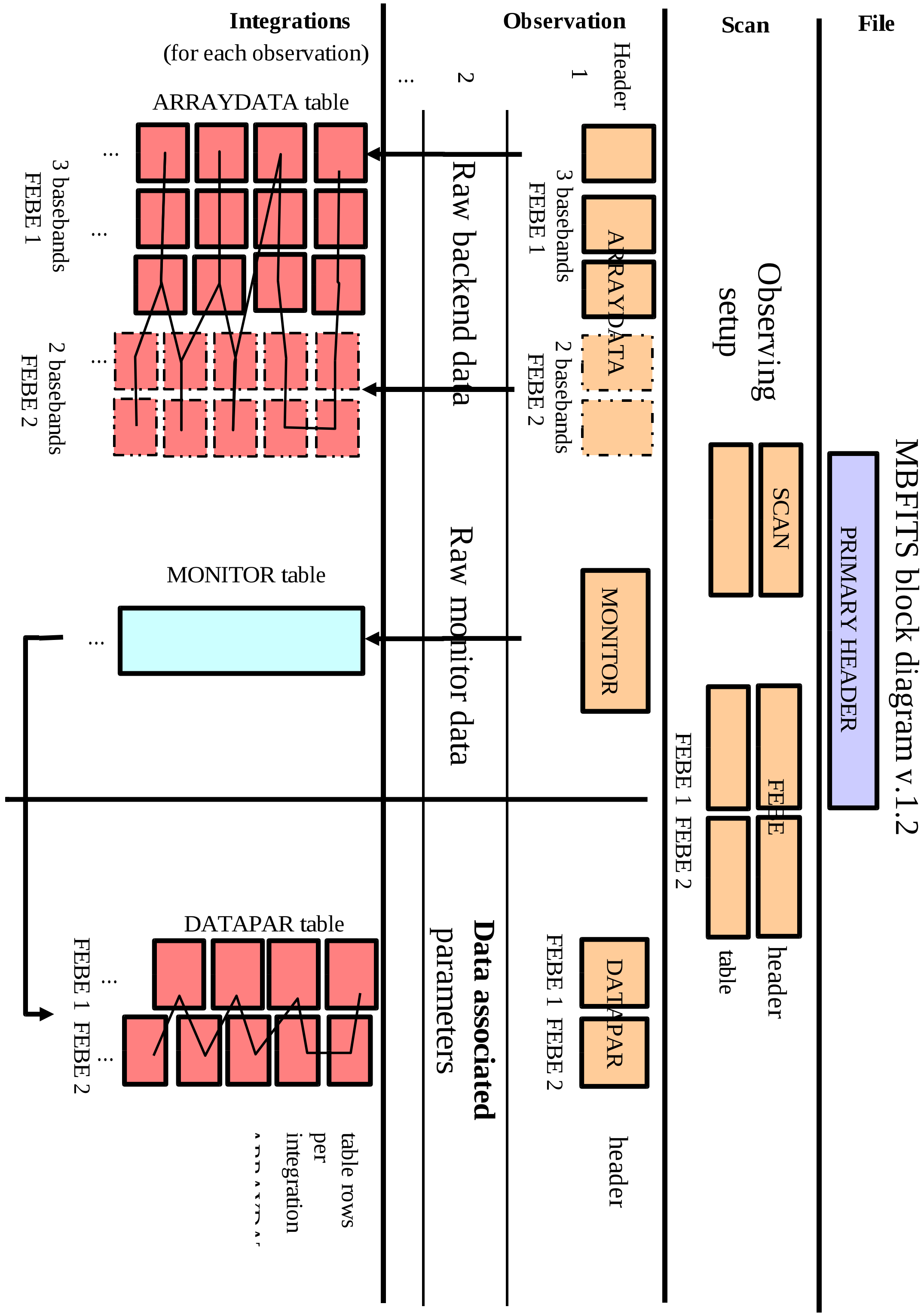

Figure 1: MBFITS file structure

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 accomodate 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-observation-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 Sect. 2 we outline the updates in the latest version of MBFITS. In Sect. 3, various aspects of MBFITS are described in detail. Then follows the specification of the FITS tables (Sect. 5. Finally, we include references (Sect.4).

1.2 Scans, observations 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.
>
> **observation** A set of integrations while the antennas complete an elemental pattern across the source, possibly while frequency switching, nutator switching, etc.
>
> **scan** A set of observations 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 an observation would be a drift across the transmitter or a bore-sight measurement, while a scan could be the whole set of observations needed to acquire a beam map. Or a scan could be a pointing scan with two observations (an azimuth drift and an elevation drift across the pointing calibrator) or an atmospheric calibration scan with three observations (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 observations 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.
More examples of a scan:

- An on-the-fly map of an astronomical source, including associated sky off observations
- 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 an observation:

- 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.1.2?

Keywords which are new, moved or altered in v.1.2 are given in bold in the tables.

1. **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.

2. **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 eg. 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 eg. 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.

3. **Calibration**

   A close look has been taken at what's required in order to calibrate the data and where this should be stored; see 3.3 below.

4. **Scanning/mapping description**

   Keywords which describe scanning/mapping keywords have been updated following a work-through of examples. See 3.1.

5. **Phase coding**

   We have looked description of multiphase (switching) observations and describe schemes for some common cases in 3.2.

6. **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 take ISWITCH from the previous entry and 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.

7. **Holography**

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

8. **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

9. **MJD** 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 3.4 below.

10. **Other keyword additions and alterations**

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

    - `FRQTHROW` in the FEBE header gives the frequency step for frequency switched observations.
    - `DEWRTMOD` (FEBE) 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` (FEBE table) are measured in degrees in the system in which the dewar rotation remains fixed.

3 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 (eg. 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 5.2–5.6).

3.1 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, HOLO, OTF, ONOFF, PSW, RASTER, CROSS, FLUXCAL,` 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, SCAPTS, 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 `CTYPE`j (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`. eg. `CROOCOO` is a `REF` every four `ON`s and `CAL` every two `ON`s. See 30m NCS documentation. `CAL/REF/ON` are stored in `OBSMODE` per observation.
`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 observations. 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.

3.2 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` (FEBE), 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

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 (eg. `LATOFF` and `LONGOFF` in DATAPAR).

Frequency switching: `FRQTHROW` (FEBE 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.

Phases for switched observations are stored with each integration in ISWITCH (DATAPAR). Some common examples are:

Type of switching	**ISWITCH**
frequency switching	`FHI/FLO`
wobbler (nutating subreflector)	
2 phases	`ON/OFF`
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`

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

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

3.3 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 FEBE 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 FEBE table which are not available at the time of writing should be left blank to be updated later.

3.3.1 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? What is the parameterisation for APEX? Should we store these parameters in MBFITS?

- **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` (FEBE 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` (FEBE 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 FEBE 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 (eg. 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 `BOLFLAT` (FEBE, per scan).

3.3.2 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 FEBE table every scan.

- Beam efficiency

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

3.4 DATE-OBS

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.

3.5 WCS coordinates

The representation of spatial coordinates has undergone 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. 3.5.9).

3.5.1 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 (eg. RA/Dec, or the user's native frame). In v.1.0 we store the commanded longitude and latitude offsets in the user's chosen basis frame (defined by `CTYPE`*j*, eg. 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. 3.5.8).

- **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. 3.5.8).

  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 INTMON table.

3.5.2 WCS representation and telescope control system input

The World Coordinate System (WCS) (Greisen & Calabretta, Calabretta & Greisen, Greisen & Valdes 2002) 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 eg. 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 (eg. 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

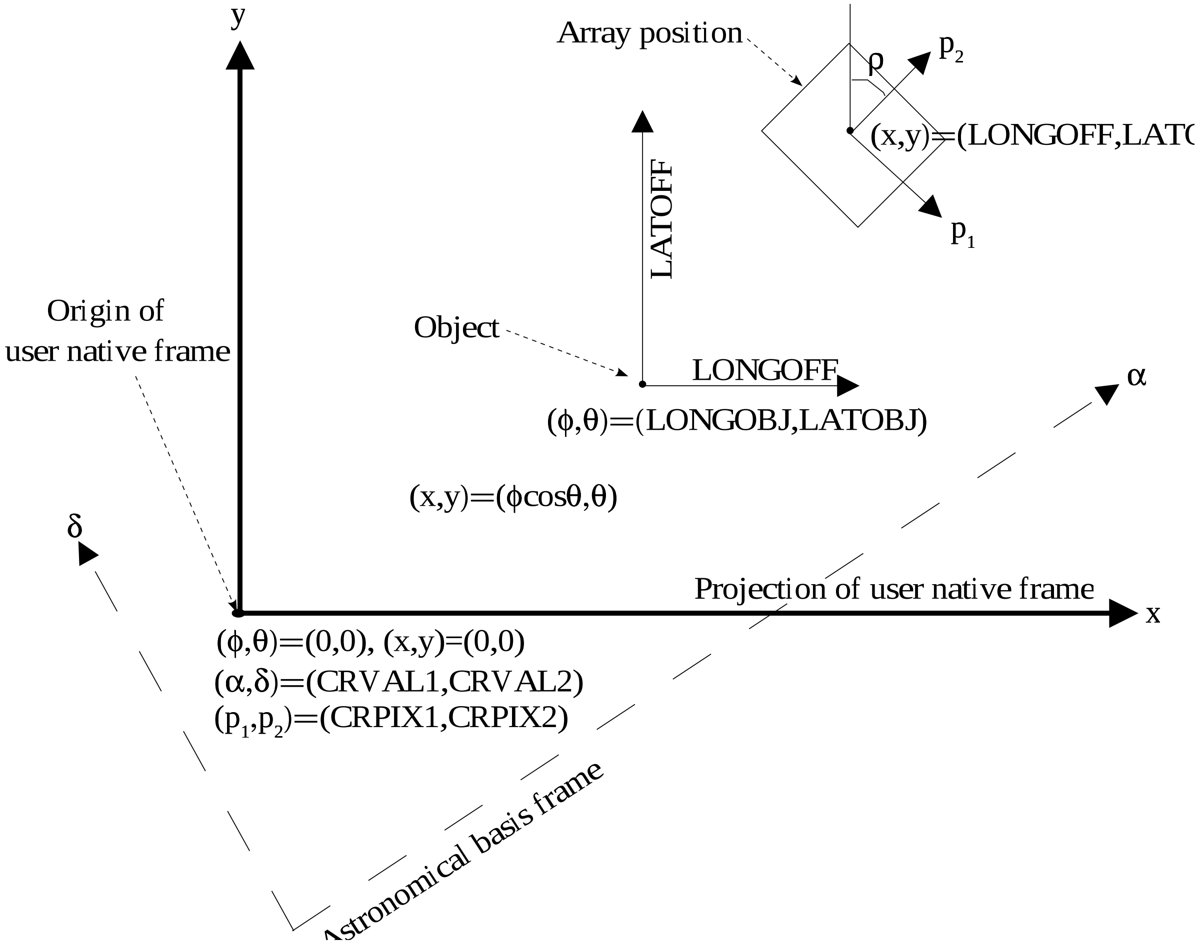

Figure 2: Coordinate relationships for single dish array observations in WCS scheme.

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. 3.5.5, but first in Sect. 3.5.3 we give a brief description of the relevant aspects of WCS and in Sect. 3.5.4 list the 30m OBSINP parameters which currently control positioning at that telescope.

3.5.3 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. 3.5.5.

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

$$\begin{aligned} x &= \phi \cos\theta \\ y &= \theta. \end{aligned} \tag{1}$$

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.

3.5.4 30m parameters

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

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

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. 3.1.

3.5.5 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.

CTYPEj Define basis 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).

The usual projection for single dish radio astronomy is Sanson-Flamsteed, coded SFL, for which

$$\begin{aligned} x &= \phi \cos\theta \\ y &= \theta \end{aligned} \tag{2}$$

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.

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 'Horizontal coordinates' for Azimuth/Elevation.

See Sect. 3.5.6 for a description of how MBFITS works in the case of moving bodies.

`CTYPE`j are stored in the SCAN header.

WCSNAME

A description of the basis coordinate system, especially where it is non-standard. See `CTYPE`j (above).

Stored in SCAN header.

RADESYS Additional ecliptic/equatorial basis system definition eg. FK4/FK5: see Calabretta Greisen (2002) Table 1.

Stored in SCAN header.

EQUINOX Basis system equinox.

Stored in SCAN header.

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, so we leave these out of MBFITS.

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

Stored in the FEBE 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 INTMON, derived from the dewar angle `DEWANG` (MONITOR table).

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 $(\mathrm{LONGOBJ}, \mathrm{LATOBJ})$, plus lat./long. offsets to the array centre (which may include wobbler offsets), then the `CRPIX` are:

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

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

Longitude of native pole in basis coordinates

Stored in SCAN header.

In the SBAS D scheme,

$$\sin(\texttt{LONPOLE}) = \frac{-\sin(\mathrm{SK0P})\cos(\mathrm{SB0P})}{\sin[\arccos(\cos(\mathrm{SB0P})\cos(\mathrm{SK0P}))]}.$$

LATPOLE

Latitude of basis pole in native coordinates (note the opposite definition to `LONPOLE`

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 `CRVAL`j, `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
- `CRPIX`i: pixel coordinates of projection plane origin
- `PC`i`_`j: rotation matrix

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

- `CTYPE`j: xxxx-SFL projection

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

- `CRVAL`j : native system origin in basis coordinates
- `LONPOLE`: longitude of native pole in basis coordinates
- `LATPOLE`: latitude of basis pole in native coordinates
- `CTYPE`j: 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 `CTYPE`j.

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.

3.5.6 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 (eg. 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`=BLON/BLAT (longitude and latitude in the frame of the body) with `WCSNAME` 'Moving body coordinates'. As this frame exactly tracks the body's native coordinate system, `CRVAL`$j= (0, 0)$, `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.

3.5.7 Observations in Az/El, On-the-fly mapping 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 on-the-fly mapping in azimuth (or elevation) about a source/offset position defined in a celestial system, the basis frame and native frame are celestial frames related by the usual parameters (neither is Az/El), and (LATOFF, LONGOFF) is given in the native celestial frame for each integration as usual, although the scanning was specified in azimuth. The derivation of as (LATOFF, LONGOFF) from the real-time antenna coordinates and wobbler offset plus frame definitions must then take into account the chopping direction, stored in `WOBDIR`.

3.5.8 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.

The pointing terms are given in Table 2.

Table 2: Pointing terms

Term	30m equiv.	Description
Antenna static terms, in SCAN *header*		
IA	P1	Azimuth zero offset
CA	P2	Collimation
NPAE	P3	Collimation of the axes /Non-perpendicularity between mount azimuth and elevation axes
AN	P5	Azimuth azis offset north-south / zenith shift
AW	-P4	Azimuth offset east-west / zenith shift
IE	P7	Elevation zero offset
ECEC	P8	Gravitational flexure perpendicular to optical axis plus vertical receiver offset at Nasmyth focus
ZFLX		Gravitational flexure parallel to optical axis plus horizontal receiver offset at Nasmyth focus
Receiver static terms, in FEBE *table*		
IARX		Receiver azimuth zero offset
CARX		Receiver collimation
IERX		Receiver elevation zero offset
ECEC		Receiver flexure term
Focus/elevation correction: internal to pointing computer?		
Observed pointing corrections, in MONITOR		
IAOBS	NULA	azimuth offset determined from pointing obs.
CAOBS	COL*	collimation determined from pointing obs.
IEOBS	NULE	Elevation offset determined from pointing obs.
Dynamic focus correction, in MONITOR		
FOCUSOBS	SFC2	Radial focus correction determined from focus measurement.
Refraction correction, in MONITOR		
REFRACTIO		Refraction correction, from `HUMIDITY`, `TAMBIENT`, and `PRESSURE` as a function of elevation

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.

3.5.9 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 & Valdes 2002). 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 (eg. 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 observation, 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 '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.

4 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', in prep., available from http://www.aoc.nrao.edu/∼egreisen/

[4] Calabretta M.R., Greisen E. W., 2002, 'Representation of celestial coordinates in FITS', in prep., available from http://www.aoc.nrao.edu/∼egreisen/

[5] Greisen E. W., Valdes F. G., 2002, 'Representations of spectral coordinates in FITS', in prep., available from http://www.aoc.nrao.edu/∼egreisen/

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

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

[8] 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

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

5 MBFITS specification

5.0.10 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

5.1 The Primary header

Keyword	Type	Value	Description
`NAXIS`	1I	0	
`SIMPLE`	1L	T	
`BITPIX`	1I	32	
`EXTEND`	1L	T	
`TELESCOP`	20A		Telescope Name
`ORIGIN`	20A		Organisation or Institution
`CREATOR`	20A		Software (including version)
`MBFTSVER`	10A		MBFITS version
`COMMENT`	20A		

5.2 The `SCAN-MBFITS` Binary Table

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

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

5.2.1 `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	
`PROJID`	12A	-	Project ID	was 4 characters, now 12 - less restrictive
OBSID	12A	-	Observer and operator initials	
`SCANNUM`	1J	-	Scan number	
`DATE-OBS`	23A	-	scan start in `TIMESYS` system	
`MJD`	1D	day	Scan date/time (Modified Julian Date)	
`LST`	1D	s	Local apparent sidereal time (scan start)	
`N_OBS`	1J	-	Number of observations in this scan	
`TIMESYS`	4A	-	time system (TT, TAI, UTC ...)	
`UT1UTC`	1D	s	UT1-UTC time translation	
`TAIUTC`	1D	s	TAI-UTC time translation	See Sect. 3.5 for coordinate system definition
`CTYPE1`	8A	-	Basis system (longitude) – XLON-SFL	
`CTYPE2`	8A	-	Basis system (latitude) – xLAT-SFL	
`RADESYS`	8A	-	additional system definition for ecliptic/equatorial coords	
`EQUINOX`	1E	Julian yrs	Equinox	

`CRVAL1`	1D	deg	Native frame zero in basis system (long.)	
`CRVAL2`	1D	deg	Native frame zero in basis system (lat.)	
`LONPOLE`	1D	deg	Native longitude of celestial pole	
`LATPOLE`	1D	deg	Basis latitude of native pole	
`OBJECT`	20A	-	Source name	
`LONGOBJ`	1D	deg	Source longitude in native frame	
`LATOBJ`	1D	deg	Source latitude in native frame	
`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 yrs	TP, Julian date of perihelion passage	
`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	yrs	EPOCH, Epoch of orbital elements	
`ORBEQNOX`	1D	Julian yrs	Elements equinox	
`DISTANCE`	1D	AU	Geocentric Distance	
SCANTYPE	20A	-	Scan astronomical type	including POINT, FOCUS, CAL, SKYDIP, HOLO, OTF, ONOFF, PSW, RASTER, CROSS, UNKNOWN, FLUXCAL... Formerly OBSTYPE. Any of the following `SCANxxxx` parameters can change from observation to observation move to the DATAPAR header and should be looked for there. Particularly `SCANDIR` often changes between observations. The parameters should be included as needed depending on SCANMODE. See text.
SCANMODE	20A	-	Mapping mode	SAMPLE, RASTER, OTF. v.1.2: SCANMODE now holds map type and SCANGEOM the geometry
SCANGEOM	20A	-	Scan geometry	including SINGLE, LINE, CROSS, RECT, QUAD, CIRC, CURVE
SCANDIR	4A	-	(optional) scan direction	described as USER (user native frame) or xLON/xLAT as in `CTYPE`*j* (standard basis system), including ALON/ALAT for Az or El scanning.

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 FEBE header, as it is receiver-dependent
CROCYCLE	20A	-	CAL/REF/ON loop string	showing how often to go to CAL and REF, eg. 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	1J	m	(optional, HOLO) Holography transmitter distance	If `SCANTYPE=HOLO`, these three special holography keywords are stored.
TRANFREQ	1J	m	(optional, HOLO) Holography transmitter frequency	
TRANFOCU	1J	m	(optional, HOLO) transmitter offset from prime focus	
WOBUSED	1L	-	Wobbler used?	Wobbler parameters apply to all receivers.
WOBTHROW	1D	deg	wobbler throw	
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`)	
NFEBE	1J	-	N_{FEBE} number of FEBEs	Previously N_FEBE
IA	1E	deg	Pointing Coefficient (P1)	For a full description of the pointing terms see Sect. 3.5.8.

`CA`	1E	deg	Pointing Coefficient (P2)
`NPAE`	1E	deg	Pointing Coefficient (P3)
`AN`	1E	deg	Pointing Coefficient (P5)
`AW`	1E	deg	Pointing Coefficient (-P4)
`IE`	1E	deg	Pointing Coefficient (P7)
`ECEC`	1E	deg	Pointing Coefficient (P8)
`ZFLX`	1E	deg	Pointing Coefficient (Flexure)

5.2.2 `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

5.3 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. 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 (`USEAC`, `USEDC` = AC/DC coupled array elements in use for bolometers, `USEFEED` for heterodyne receivers).

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`.

5.3.1 `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)	
DEWRTMOD	5A	-	Dewar tracking system	CABIN = fixed in Nasmyth/Cassegrain system, EQUA: RA/DEC; HORIZ: AZ/EL
DEWANG	1J	Deg	Dewar angle	Fixed in DEWRTMOD system, measured counterclockwise from vertical. Fixed throughout a scan.
`FEBEBAND`	1J	-	$N_{\rm BD}$ number of 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.

NUSEFEED	1J	-	$N_{\rm USFD}$ Number of feeds in use.	So that we can minimally dimension the data storage array in ARRAYDATA table. Now for bolometers or heterodyne. A bolometer feed which is AC- and DC-coupled counts twice here
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
`NO_SWITCH`	1J	-	no. of switch phases in a switch cycle	fs/wobbler
FRQTHROW	1E	Hz	Frequency switching throw	
`IARX`	1E	deg	Pointing Coefficient (receiver), adds to IA (P1)	Here are the receiver-specific pointing coefficients:
`CARX`	1E	deg	Pointing Coefficient (receiver), adds to CA (P2)	
`IERX`	1E	deg	Pointing Coefficient (receiver), adds to IE (P7)	
`ECECRX`	1E	deg	Pointing Coefficient (receiver), adds to ECEC (P8)	
`ZFLXRX`	1E	deg	Pointing Coefficient (receiver), adds to ZFLX	

5.3.2 `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
USEFEED	$N_{\rm USFD}$ J	-	List of feeds which are in use	The data in the ARRAYDATA table are stored in the order listed in USEFEED (heterodyne receivers). v.1.2: USEAC, USEDC and corresponding data arrays are removed so that table structure remains constant.
FEEDTYPE	$N_{\rm USFD}$ A	-	feed type	'H' for heterodyne; for bolometers, 'A'=AC-coupled, 'D'=DC-coupled.
				The following parameters depend on feed and (sometimes) frequency band, and are given for the whole array, not just the feeds in use.
`FEEDOFFX`	$N_{\rm FD}$ D	deg	feed x offset	x offset of each feed from rotation centre in DEWRTMOD system.

`FEEDOFFY`	$N_{\rm FD}$ 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, eg. CHAMP - but not necessarily at the rotation centre (0,0)
`POLTY`	$N_{\rm FD}$ A	-	Feed type (X, Y, L, R)	Here is the polarisation of each feed for this FE
`POLA`	$N_{\rm FD}$ E	deg	Feed orientation	ccw from vertical in DEWRTMOD system
				The keywords `ACCOUPLE` and `DCCOUPLE` are unnecessary given `FEEDTYPE` and have been removed (v.1.2).
`APEREFF`	$N_{\rm FD}{\times}N_{\rm BD}$ E	-	Aperture efficiency	
`BEAMEFF`	$N_{\rm FD}{\times}N_{\rm BD}$ E	-	Beam efficiency	
`ETAFSS`	$N_{\rm FD}{\times}N_{\rm BD}$ E	-	Forward efficiency	
`HPBW`	$N_{\rm FD}{\times}N_{\rm BD}$ E	deg	Half-power beam width	
`ANTGAIN`	$N_{\rm FD}{\times}N_{\rm BD}$ E	K/Jy	Antenna Gain	
BOLCALFC	$N_{\rm BD}$ E	Jy/counts	Bolometer calibration factor	Set by planet/secondary calibrator observations. Previously BOLREFGN.
`BOLFLAT`	$N_{\rm FD}{\times}N_{\rm BD}$ E	-	Bolometer flat field (relative gains)	
`GAINIMAG`	$N_{\rm FD}{\times}N_{\rm BD}$ E	-	(spectral line) Gain ratio image/signal sideband	

5.4 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

5.4.1 `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	
`OBSNUM`	1J	-		Observation number	
`DATE-OBS`	23A	-		observation start in `TIMESYS` system	Y2K format with time
`CHANNELS`	1J	-		N_{CH} Number of channels for this baseband	
`FREQRES`	1D	Hz		Frequency resolution	Not channel width, which is stored in the axis description 21CD4F.
BANDWID	1D	Hz		Bandwidth for this band	
MOLECULE	20A	-		main line molecule (optional)	
TRANSITI	20A	-		main line transition (optional)	
`RESTFREQ`	1D	Hz		Rest frequency of line (optional)	
`SIDEBAND`	3A	-		Main sideband is USB/LSB	
`SBSEP`	1D	Hz		Sideband separation	
`1CTYP4`	8A	-	‘PIX-INDX’	Pixel/feed index in USEPIX array	The data axis description
`1CRPX4`	1J	-		Ref. position = 1	
`1CRVL4`	1J	-		Pixel index value at this position = 1	
`11CD3A`	1J	-		Pixel index separation = 1	
Next we assign to the spectral axis a frequency and velocity description relative to the rest frame, but other alternative axes descriptions could also be given.					
`WCSNM4F`	8A	-	eg. ‘LsrkFreq’	Axis name	Frequency description in rest frame (eg. LSR) for main sideband:
`2CTYP4F`	8A	-	‘FREQ-FRQ’	Frequency axis for col.3	
`2CRPX4F`	1E	Hz		Ref. channel	
`2CRVL4F`	1D	Hz		Frequency at ref. channel in rest frame	
`21CD4F`	1D	Hz		Channel Separation	

`2CUNI4F`	8A	-	‘Hz’	Unit	
`2SPEC4F`	8A	-	eg. ‘LSRK’	standard of rest for frequencies	
`2SOBS4F`	8A	-	‘TOPOCENT’	Observing frame	
`WCSNM4R`	8A	-	eg. ‘LsrkVRad’	Axis name	Velocity description in rest frame (eg. LSR) for heterodyne receivers.
`2CTYP4R`	8A	-	eg. ‘VRAD-FRQ’	Velocity axis for col.3	
`2CRPX4R`	1E	-		Ref. channel	
`2CRVL4R`	1D	km/s		Velocity at ref. channel	
`21CD4R`	1D	km/s		Velocity Channel Separation	
`2CUNI4R`	8A	-	‘m/s’	Unit	
`2SPEC4R`	8A	-	eg. ‘LSRK’	standard of rest frame for velocities	
`2SOBS4R`	8A	-	‘TOPOCENT’	Observing frame	
`2VSOU4R`	1E	km/s		Source velocity in rest frame	
`2VSYS4R`	1E	km/s		Observer vel. in rest frame	in direction of observation, at observation start

5.4.2 `ARRAYDATA-MBFITS` Binary Table Columns

Keyword	Type	Units	Description	Comments
`INTEGNUM`	1J	-	Integration point number	Use a separate INTEGNUM sequence for each FEBE.
`MJD`	1D	day	Scan date/time (Modified Julian Date)	
`DATA`	$N_{\mathrm{USFD}} \times N_{\mathrm{CH}}$ E	-	Data	USEFEED gives the number of feeds in use. Data is listed in feed order following the list in the FEBEPAR table.

5.5 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 - eg. 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 observation should not stop when data taking stops. The MONITOR table should be closed either when a new observation 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 eg. 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.

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.

5.5.1 `MONITOR-MBFITS` Table Header Keywords

Keyword	Type	Units	Description	Comments
`EXTNAME`	20A	-	'MONITOR-MBFITS'	
`SCANNUM`	1J	-	Scan number	
`OBSNUM`	1J	-	Observation number	
`MJD`	1D	day	Scan date/time (Modified Julian Date)	

5.5.2 `MONITOR-MBFITS` Binary Table Columns

Keyword	Type	Description	Comments
`DATE-OBS`	23A	observing date in TIMESYS system at monitor point	
MONPOINT	A30	Monitor point description	The reason for long (A30) monitor points is to accommodate FEBE descriptions (A17) and band numbers (2 digits) where necessary. Eg. `TAUZEN_<febe>_<band>`
`MONUNITS`	A8	Units for monitor point	
`MONVALUE`	1PJ(maxelem)	Pointer to monitor values	which are stored in a heap at the end of the table.

5.5.3 Anticipated monitor points

Keyword	Elements	Description	Comments
Current real positions			
ENCODER_AZ_EL	2D	Encoder azimuth and elevation	The raw drive positions above are not available for the APEX antenna.
TRACKING_AZ_EL	2D	Average tracking azimuth and elevation	
INCLINOMETER_1_2	2D	Inclinometer 1 and 2	
FOCUS_X_Y_Z	3D	Focus (subreflector position) X, Y, Z	
PHI_X_Y_Z	3D	Phi (subreflector rotation) X,Y,Z	
ANTENNA_AZ_EL	2D	Pointing-corrected antenna azimuth – real-time position	
position			
			Dewar angle has moved to FEBE header

WOBDISPL	1D	wobbler displacement	suggested for MAMBO117. Switch beams using the wobbler but read out data at intermediate positions, storing the displacement here.
TP_<febe>	N_{BD} D	Total Power in each baseband (per FEBE)	
Frequencies labelled by frontend			
GUNF_<fe>	1D	Gunn frequency	Frequencies can change rapidly eg. frequency switching, Doppler tracking
PLLF_<fe>	1D	Phase locked loop frequency	
LOFREQ_<febe>	N_{LO}D	LO frequencies from frontend/LO chain	
LOSIDEBAND_<febe>	N_{LO}D	LO sidebands from frontend/LO chain	
Environmental parameters			
TSTRUCT	nD	Structural temperature(s)	
WIND_SPEED_DIR	2D	Wind speed and direction	
TAMB_P_HUMID	3D	Ambient conditions triplet: temperature, pressure, humidity	
`TCABIN`	1D	Receiver Cabin temp.	
`TDEWAR`	1D	Receiver Dewar temp.	
Calibration			
			The following update each spectral line calibration scan
THOTCOLD_<febe>	2D	Chopper temperature, Cold load temperature	
PHOT_<febe>_<band>	N_{USFD} D	Total power on chopper	band-averaged, for each feed in use
PCOLD_<febe>_<band>	N_{USFD} D	Total power on cold load	band-averaged, for each feed in use
PSKY_<febe>_<band>	N_{USFD} D	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_{USFD} D	receiver temperature	
TSYS_<febe>_<band>	N_{USFD} D	system temperature	
TSYSIMAG_<febe>_<band>	N_{USFD} D	image band system temperature	
TCAL_<febe>_<band>	N_{USFD} D	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 observation.
OBSFL_<febe>_<band>	3D	Calibrator measured flux	from continuum calibrator obs. (`SCANTYPE`=FLUXCAL)
CALFL_<febe>_<band>	3D	Calibrator predicted flux	From calibrator flux model (`SCANTYPE`=FLUXCAL)
CALFAC_<febe>_<band>	3D	Counts-to-Jy calibration factor	From continuum calibrator obs. (`SCANTYPE`=FLUXCAL)
Water vapour radiometer			
TSYS_TRX_RD<freq>	2D	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	opacity and water vapour column	at radiometer frequency
Observer Pointing			
IAOBS_CAOBS_IEOBS	3D	Azimuth pointing offset from pointing obs. to add to IA (P1), CA (P2) and IE (P7)	
X_Y_Z_FOCOBS	3D	X,Y,radial focus correction	determined from focus measurement and relayed to pointing computer. In 30m terms, IAOBS = NULA, CAOBS = COL*, IEOBS = NULE
`REFRACTIO`	1D	Refraction correction (current), calculated from `HUMIDITY`, `TAMBIENT`, and `PRESSURE` as a function of elevation.	

5.6 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 observations are in the SCAN table, and the FEBE setup is in the FEBE table (assumed not to change between observations).

DATAPAR includes:

– Time and coordinate information specific to this observation 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 `INTEGNUM` will equal the row number and `NINTS`= 1. In the case of fast data rates, 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, `INTEGNUM` will skip values with `NINTS` showing the number of integrations to which each row of data-associated parameters applies. The parameters apply to the `NINTS` integrations starting with the matching `INTEGNUM`.

When blocking, all DATAPAR values refer to the first integration of the block. An integration for which no direct entry in DATAPAR exists should take ISWITCH from the previous entry and interpolate between times and positions from bracketing entries. If blocking, a DATAPAR entry should always be written for the first and last integration of each observation, so that the interpolation can be done.

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 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 the data flow .

NB All values in DATAPAR **are interpolated to the midpoint of the integration.**

5.6.1 `DATAPAR-MBFITS` Binary Table Header Keywords

Keyword	Type	Units	Description	Comments
`EXTNAME`	20A	-	‘DATAPAR-MBFITS’	
`SCANNUM`	1J	-	Scan number	
`OBSNUM`	1J	-	Observation number	
`DATE-OBS`	23A	-	observation start in `TIMESYS` system	(Y2K format with time)
FEBE	17A	-	FEBE descriptor	
`LST`	1D	s	Local apparent sidereal time (obs. start)	
OBSTYPE	4A	-	Observation type	Calibration (CAL), reference position (REF), or on-source (ON)

SCANxxxx			Scan description	Any of the `SCANxxxx` parameters that change from observation to observation appear here rather than in the SCAN header. Particularly `SCANDIR`.
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.

5.6.2 `DATAPAR-MBFITS` Binary Table Columns

Keyword	Type	Units	Description	Comments
`INTEGNUM`	1J	-	Integration point number	Use a separate INTEGNUM sequence for each FEBE.
NINTS	1J	-	Integrations in block	Number of integrations to which these parameters apply, including this one.
`MJD`	1D	Julian day	MJD at integration start `TIMESYS` system (integration start)	
`LST`	1D	s	Local apparent sidereal time (integration start)	
`INTEGTIM`	1D	s	Integration time	
`MIDTIME`	1A	-	Time of midpoint of observations **in MJD format**	
start)				
`ISWITCH`	4A	-	Integration type frequency switch FHI/FLO wobbler 1/2 (/3/4...) horn switch 1/2 load switch SKY/LOAD calibration HOT(AMB)/COLD/CSKY/NULL/SG	
LATOFF	1D	deg	lat. offset from source in user native frame (derived)	
LONGOFF	1D	deg	long. offset from source in user native frame (derived)	
AZIMUTH	1D	deg	Azimuth (derived) inc. wobbler offsets	
ELEVATION	1D	deg	Elevation (derived) inc. wobbler offsets	
`CBASLONG`	1D	deg	Commanded lat. in astronomical basis frame (defined by CTYPE), from TCS	

`CBASLAT`	1D	deg	Commanded long. in astronomical basis frame (defined by CTYPE), from TCS	
BASLONG	1D	deg	Actual lat. in astronomical basis frame	
BASLAT	1D	deg	Actual long. in astronomical basis frame	
PARANGLE	1D	deg	Parallactic angle	needed for derotator if tracking in equatorial system
CRPIX1	1D	-	zero of native frame in pixel units axis 1	
CRPIX2	1D	-	zero of native frame in pixel units axis 2	
11PC	1D	-	array pixel to native coord. rotation matrix cpt	Derived from DEWANG in MONITOR– see Sect. 3.5 for notes on coordinate systems.
12PC	1D	-	array pixel to native coord. rotation matrix cpt	
21PC	1D	-	array pixel to native coord. rotation matrix cpt	
22PC	1D	-	array pixel to native coord. rotation matrix cpt	
`MCRVAL1`	1D	deg.	(opt.) body long. in basis system	Frame tracking for moving bodies – see Sect. 3.5.6 – these columns exist if MOVEFRAM in the SCAN header is TRUE.
`MCRVAL2`	1D	deg.	(opt.) body lat. in basis system	
`MLONPOLE`	1D	deg.	(opt.) longitude of basis celestial pole in body system	
`MLATPOLE`	1D	deg.	(opt.) basis latitude of body frame pole	