FITS and MBFITS output formats in ESCS

	Name	Date	Signature
Authors	Righini S (IRA) Zanichelli A (IRA)	28/04/14	
	Contributions from the DISCOS project group	12/06/15	

DOCUMENT CHANGE RECORD				
Issue No.	**Issue date**	**No. of pages**	**Pages changed, added, deleted**	**Description of change**
01	28/04/14			FITS format aligned to ESCS0.3
02	12/08/15			**FITS format aligned to ESCS 0.5** Addition of summary.fits Spectroscopy-related updates Derotator position is now online

→ **IMPORTANT** ***Notice*** The FITS content will undergo further revisions in a near future. More keywords will be added, others will be renamed in order to converge to the MBFITS keyword containing the same information (both for consistency and to better develop a common archiving system); the **keywords to be renamed** are written in **bold** in the tables provided in this document.

Keywords which still hold **dummy values** (i.e. numbers/strings which are not actually read on-line, but are instead assigned as placeholders) are given in **square beackets [ ]**.

ESCS output files

The ESCS system is provided with data-writing *containers/components* able to gather the data coming from the fully integrated backends and record them into two possible formats:

1) *FitsZilla* → substantially standard FITS;
2) *MBFitsWriter* → hierarchical MBFITS *(not available as it is not fully tested)*

This document, referring to ESCS 0.5*,* aims at describing these formats in order to allow the users to find the data and ancillary information recorded inside the files.
MBFitsWriter is not yet available for observations, as it is still under testing. Nonetheless, we anticipate here the main features of the MBFITS format it will produce. Since this format is very vast and complex, only a general description is provided here; the reader is invited to access the full manual and other literature resources to read the details (see MBFITS section).

1 FITS

This version of the output file is an almost-standard FITS made out of the following extensions:

Index	Extension	Type
0	Primary	Image
1	SECTION TABLE	Binary
2	RF INPUTS	Binary
3	FEED TABLE	Binary
4	DATA TABLE	Binary
5	ANTENNA TEMP TABLE	Binary
6	SERVO TABLE	Binary

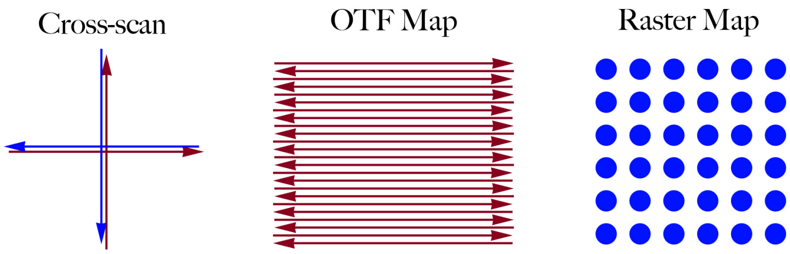
It opens and plots with any software reading regular FITS (FitsViewer, IDL routines, FITS I/O libraries, etc...).

ESCS writes a FITS file for each subscan composing the ongoing observation, according to the following convention:

Scan
It 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 cross-scans and maps involve a pattern of subscans. Whether OTF maps mosaicing observations are considered a single scan or a series of scans is rather a matter of how the user would like to define it. In our implementation each map is considered a scan.

Subscan
it is the minimal amount of data acquisition that can be commanded at the script language level. It is highly desirable that it is a simple enough element. For example, it is the single OTF “line” of a map or of a cross-scan.

The figure below visually represents what cross-scans, OTF maps and raster maps are.



In the case of cross-scan, a subscan is a single arrow (a line across the target), four arrows – i.e. two full crosses – constitute the schema which might be repeated as many times as needed within the scan.
For OTF maps, the subscan is again the single arrow, and the scan coincides with the whole map obtained with lines along one axis only (e.g. along RA or Dec). For raster maps, which are based on discrete acquisitions, each point is a subscan, and the final map constitutes the scan.

When choosing FITS as the data output format, a distinct FITS file is produced for each subscan listed in the schedule. In addition, for each scan a **summary.fits** file is produced at the end of the acquisitions: it sums up the main keywords/parameters in order to simplify archiving and querying procedures.

```
MainFolder
   |-----> YYYYMMDD
   |         |-----> Scan1Folder
   |         |                  |-----> Subscan1.FITS
   |         |                  |-----> Subscan2.FITS
   |         |                  |-----> …
   |         |                  |-----> summary.fits
   |         |
   |         |-----> Scan2Folder
   |                            |-----> Subscan1.FITS
   |                            |-----> Subscan2.FITS
   |         |                  |-----> …
   |         |                  |-----> summary.fits
   |
   |-----> YYYYMMDD
             |-----> Scan1Folder
             |                  |-----> Subscan1.FITS
             |                  |-----> Subscan2.FITS
             |                  |-----> …
             |                  |-----> summary.fits
             |
             |-----> Scan2Folder
                                |-----> Subscan1.FITS
                                |-----> Subscan2.FITS
                                |-----> …
                                |-----> summary.fits
```

Scan folder names are composed as: **YYYYMMDD-HHMMSS-Project-Suffix**
where
HHMMSS is the UT time associated to the first sample of the acquisition
Project is the code/name specified using the "project=" command, or when starting a schedule with "startSchedule=project/schedulename.scd,N"
Suffix is a user-defined string retrieved from the schedule files. Though no control can be applied on the choice/check of this string, the agreement is that it should coincide with the target name.

FITS files, each corresponding to a subscan, are named like:

YYYYMMDD-HHMMSS-Project-Suffix_Scan#_Subscan#.fits

→ ***Notice*** **The example contents provided in the following sub-sections are relative to an ON-OFF scan performed with XARCOS using the XK01 configuration in zoom mode.**

1.0 PRIMARY HEADER

Here the compulsory FITS header keywords are stored. A list of keywords dedicated to observing site info and telescope setup details then follows.

KEYWORD	**EXAMPLE VALUE**	**DESCRIPTION**
SIMPLE	T	file does conform to FITS standard
BITPIX	8	number of bits per data pixel
NAXIS	0	number of data axes
EXTEND	T	FITS data set may contain extensions
DATE	'2015-06-11T07:33:48'	File creation date (YYYY-MM-DDThh:mm:ss UT)
HIERARCH Project_Name	'TestESCSFTrack'	Name of the project
OBSERVER	' '	Name of the observer
ANTENNA	'MED '	Name of the station
HIERARCH SiteLongitude	0.161358481873679	Longitude of the site (radians)
HIERARCH SiteLatitude	0.689283579621821	Latitude of the site (radians)
HIERARCH SiteHeight	650.	Height of the site (meters)
BEAMS	2	Number of beams
SECTIONS	4	Total number of sections
HIERARCH Sample Size	8	Number of bytes
HIERARCH Receiver Code	'KKC '	Keyword that identifies the receiver
SOURCE	'W3OH '	Source identifier
HIERARCH RightAscension	0.641706660521798	Source J2000 Right Ascension (radians)
HIERARCH Declination	1.079883689577	Source J2000 Declination (radians)
VLSR	-48.8	Source radial velocity
HIERARCH Azimuth Offset	0.	Longitude offset in horizontal frame
HIERARCH Elevation Offset	0.	Latitude offset in horizontal frame
HIERARCH RightAscension Offset	0.	Longitude offset in equatorial frame
HIERARCH Declination Offset	0.	Latitude offset in equatorial frame
HIERARCH GalacticLon Offset	0.	Longitude offset in galactic frame
HIERARCH GalacticLat Offset	0.	Latitude offset in galactic frame
SCANID	1	Scan Identifier
HIERARCH SubScanID	2	Subscan Identifier
HIERARCH ScheduleName	'/archive/schedules/Maintenance/spettro_new.scd'	
HIERARCH SubScanType	'TRACKING'	Scan type (or OTF scanning axis)
HIERARCH Scan Tag	1	Scan tag identifier

1.1 SECTION TABLE

It shows basic info about the sections (i.e. “data streams”).

Select	id	type	sampleRate	bins	flux
	J	6A	D	J	D
All			MHz		
Invert	Modify	Modify	Modify	Modify	Modify
1	0	stokes	1.250000000000E+02	2048	0.000000000000E+00
2	1	stokes	7.812500000000E+00	2048	0.000000000000E+00
3	2	stokes	1.250000000000E+02	2048	0.000000000000E+00
4	3	stokes	7.812500000000E+00	2048	0.000000000000E+00

Column meanings and units are also described in its header, as it happens for all the tables.
Each row is dedicated to one section:

id = section number

type = it is “simple” when the section provides total power (scalar) or spectral (array, with *bins* elements) data relative to a single polarization or “stokes” when it is an array listing the following contents [L, R, Q, U], each occuppying ¼ of the array in the given sequence.

sampleRate = data sampling rate (MHz)

bins = number of frequency bins (1 for total power)

Header example:

KEYWORD	EXAMPLE VALUE	DESCRIPTION
XTENSION= 'BINTABLE'	'BINTABLE'	binary table extension
BITPIX	8	8-bit bytes
NAXIS	2	2-dimensional binary table
NAXIS1	30	width of table in bytes
NAXIS2	4	number of rows in table
PCOUNT	0	size of special data area
GCOUNT	1	one data group (required keyword)
TFIELDS	5	number of fields in each row
TTYPE1	'id '	label for field 1
TFORM1	'J '	data format of field: 4-byte INTEGER
TTYPE2	'type '	label for field 2
TFORM2	'6A '	data format of field: ASCII Character
TTYPE3	'sampleRate'	label for field 3
TFORM3	'D '	data format of field: 8-byte DOUBLE
TUNIT3	'MHz '	physical unit of field
TTYPE4	'bins '	label for field 4
TFORM4	'J '	data format of field: 4-byte INTEGER
TTYPE5	'flux '	label for field 5
TFORM5	'D '	data format of field: 8-byte DOUBLE
EXTNAME	'SECTION TABLE'	name of this binary table extension
HIERARCH Integration	10000	Integration time (milliseconds)

1.2 RF INPUTS

Receiver general setup.

Select	feed	ifChain	polarization	frequency	bandWidth	localOscillator	attenuation	calibratonMark	section
All	J	J	8A	D	D	D	D	D	J
Invert				MHz	MHz	MHz	db	K	
	Modify	Modify	Modify	Modify	Modify	Modify	Modify	Modify	Modify
1	1	0	LCP	2.220831800000E+04	6.250000000000E+01	2.206331800000E+04	1.858837851429E+01	5.636472721667E+01	0
2	1	1	RCP	2.220831800000E+04	6.250000000000E+01	2.206331800000E+04	1.858837851429E+01	5.286038054182E+01	0
3	1	0	LCP	2.223761487500E+04	3.906250000000E+00	2.206331800000E+04	1.858837851429E+01	5.636576707213E+01	1
4	1	1	RCP	2.223761487500E+04	3.906250000000E+00	2.206331800000E+04	1.858837851429E+01	5.286134878834E+01	1
5	6	0	LCP	2.220831800000E+04	6.250000000000E+01	2.206331800000E+04	1.858837851429E+01	3.576332554239E+01	2
6	6	1	RCP	2.220831800000E+04	6.250000000000E+01	2.206331800000E+04	1.858837851429E+01	4.077726777809E+01	2
7	6	0	LCP	2.223761487500E+04	3.906250000000E+00	2.206331800000E+04	1.858837851429E+01	3.576389140702E+01	3
8	6	1	RCP	2.223761487500E+04	3.906250000000E+00	2.206331800000E+04	1.858837851429E+01	4.077802993913E+01	3

feed = feed number
ifChain = IF number
polarisation = LCP, RCP, HLP or VLP
frequency = observed frequency at the beginning of the band (MHz)
bandWidth = actual observed bandwidth (MHz)
localOscillator = frontend LO frequency (MHz)
attenuation = attenuation (dB) applied to the section
calibrationMark = temperature of the frontend calibration mark
section = number of section associated to this RF input

Header example:

KEYWORD	EXAMPLE VALUE	DESCRIPTION
XTENSION	'BINTABLE'	binary table extension
BITPIX	8	8-bit bytes
NAXIS	2	2-dimensional binary table
NAXIS1	60	width of table in bytes
NAXIS2	8	number of rows in table
PCOUNT	0	size of special data area
GCOUNT	1	one data group (required keyword)
TFIELDS	9	number of fields in each row
TTYPE1	'feed '	label for field 1
TFORM1	'J '	data format of field: 4-byte INTEGER
TTYPE2	'ifChain '	label for field 2
TFORM2	'J '	data format of field: 4-byte INTEGER
TTYPE3	'polarization'	label for field 3
TFORM3	'8A '	data format of field: ASCII Character
TTYPE4	'frequency'	label for field 4
TFORM4	'D '	data format of field: 8-byte DOUBLE
TUNIT4	'MHz '	physical unit of field
TTYPE5	'bandWidth'	label for field 5
TFORM5	'D '	data format of field: 8-byte DOUBLE
TUNIT5	'MHz '	physical unit of field
TTYPE6	'localOscillator'	label for field 6
TFORM6	'D '	data format of field: 8-byte DOUBLE
TUNIT6	'MHz '	physical unit of field
TTYPE7	'attenuation'	label for field 7
TFORM7	'D '	data format of field: 8-byte DOUBLE

TUNIT7	'db '	physical unit of field
TTYPE8	'calibratonMark'	label for field 8
TFORM8	'D '	data format of field: 8-byte DOUBLE
TUNIT8	'K '	physical unit of field
TTYPE9	'section '	label for field 9
TFORM9	'J '	data format of field: 4-byte INTEGER
EXTNAME	'RF INPUTS'	name of this binary table extension

1.3 FEED TABLE

Information on the feeds position (meaningful for Multi Feed receivers).

Select	id	xOffset	yOffset	relativePower
All	J	D	D	D
Invert	Modify	Modify	Modify	Modify
1	0	0.000000000000E+00	0.000000000000E+00	1.000000000000E+00
2	1	3.335520500000E-04	-5.777285900000E-04	9.700000000000E-01
3	2	-3.335520500000E-04	-5.777285900000E-04	9.900000000000E-01
4	3	-6.671036500000E-04	0.000000000000E+00	9.700000000000E-01
5	4	-3.335520500000E-04	5.777285900000E-04	9.500000000000E-01
6	5	3.335520500000E-04	5.777285900000E-04	9.700000000000E-01
7	6	6.671036500000E-04	0.000000000000E+00	9.700000000000E-01

id = feed number

xOffset = x offset position (radians) w.r.t. the central feed, along the Azimuth axis, considering the receiver in its reference position (no rotation is applied to dewar); x>0 for increasing azimuth

yOffset = y offset position (radians) w.r.t. the central feed, along the Elevation axis, considering the receiver in its reference position (no rotation is applied to dewar); y>0 for increasing elevation

relativePower = nominal ratio between this feed gain and the central feed gain. Do not use these values for calibration purposes.

Header example:

KEYWORD	**EXAMPLE VALUE**	**DESCRIPTION**
XTENSION	'BINTABLE'	binary table extension
BITPIX	8	8-bit bytes
NAXIS	2	2-dimensional binary table
NAXIS1	28	width of table in bytes
NAXIS2	7	number of rows in table
PCOUNT	0	size of special data area
GCOUNT	1	one data group (required keyword)
TFIELDS	4	number of fields in each row
TTYPE1	'id '	label for field 1
TFORM1	'J '	data format of field: 4-byte INTEGER
TTYPE2	'xOffset '	label for field 2
TFORM2	'D '	data format of field: 8-byte DOUBLE
TTYPE3	'yOffset '	label for field 3
TFORM3	'D '	data format of field: 8-byte DOUBLE

TTYPE4	'relativePower'	label for field 4
TFORM4	'D '	data format of field: 8-byte DOUBLE
EXTNAME	'FEED TABLE'	name of this binary table extension
DEWRTMOD	‘FIXED’	Dewar positioner configuration mode
DEWUSER	0.	Dewar static initial angle

1.4 DATA TABLE

Large table containing all the raw data, one row for each sample.

Select	time	raj2000	decj2000	az	el	par_angle
	D	D	D	D	D	D
All	MJD	radians	radians	radians	radians	radians
Invert	Modify	Modify	Modify	Modify	Modify	Modify
1	5.718431500400E+04	6.417066605217E-01	1.079883689577E+00	3.703873471257E+00	8.001563450002E-01	4.239983411729E-01
2	5.718431514491E+04	6.417066605217E-01	1.079883689577E+00	3.705034653391E+00	7.997907647051E-01	4.248299729345E-01
3	5.718431528470E+04	6.417066605217E-01	1.079883689577E+00	3.706185651017E+00	7.994274548734E-01	4.256540173091E-01
4	5.718431542334E+04	6.417066605217E-01	1.079883689577E+00	3.707326484710E+00	7.990664325294E-01	4.264704959239E-01
5	5.718431556176E+04	6.417066605217E-01	1.079883689577E+00	3.708464574694E+00	7.987053639212E-01	4.272847233834E-01
6	5.718431569987E+04	6.417066605217E-01	1.079883689577E+00	3.709599356122E+00	7.983444343684E-01	4.280962970380E-01

derot_angle	flag_cal	flag_track	weather	Ch0	Ch1	Ch2	Ch3
D	J	J	3D	8192D	8192D	8192D	8192D
radians							
Modify	Modify	Modify	Modify	Modify	Modify	Modify	Modify
0.000000000000E+00	0	1	Plot	Plot	Plot	Plot	Plot
0.000000000000E+00	0	1	Plot	Plot	Plot	Plot	Plot
0.000000000000E+00	0	1	Plot	Plot	Plot	Plot	Plot
0.000000000000E+00	0	1	Plot	Plot	Plot	Plot	Plot
0.000000000000E+00	0	1	Plot	Plot	Plot	Plot	Plot
0.000000000000E+00	0	1	Plot	Plot	Plot	Plot	Plot

Columns:

time = MJD (Modified Julian Day)
raJ2000 = J2000.0 Right Ascension (radians)
decJ2000 = J2000.0 Declination (radians)
az = azimuth (radians)
el = elevation (radians)
par_angle = parallactic angle (radians)
derot_angle = rotation angle of the dewar (radians) wrt its reference position
flag_cal = calibration mark flag, 0=off, 1=on
flag_track = tracking flag: 1 = pointing error is < 0.1*HPBW, 0 = pointing error is > 0.1*HPBW
weather = array of three values: temperature (°C), relative humidity (%) and atmospheric pressure (hPa), measured at ground level
Ch0,...,ChN = N columns, one for each active section, containing the signal data in raw counts (scalar values for total power acquisitions, arrays for spectra/full-stokes data)

Header example:

KEYWORD	**EXAMPLE VALUE**	**DESCRIPTION**
XTENSION	'BINTABLE'	binary table extension
BITPIX	8	8-bit bytes
NAXIS	2	2-dimensional binary table
NAXIS1	262232	width of table in bytes
NAXIS2	6	number of rows in table
PCOUNT	0	size of special data area

GCOUNT	1	one data group (required keyword)
TFIELDS	14	number of fields in each row
TTYPE1	'time '	label for field 1
TFORM1	'D '	data format of field: 8-byte DOUBLE
TUNIT1	'MJD '	physical unit of field
TTYPE2	'raj2000 '	label for field 2
TFORM2	'D '	data format of field: 8-byte DOUBLE
TUNIT2	'radians '	physical unit of field
TTYPE3	'decj2000'	label for field 3
TFORM3	'D '	data format of field: 8-byte DOUBLE
TUNIT3	'radians '	physical unit of field
TTYPE4	'az '	label for field 4
TFORM4	'D '	data format of field: 8-byte DOUBLE
TUNIT4	'radians '	physical unit of field
TTYPE5	'el '	label for field 5
TFORM5	'D '	data format of field: 8-byte DOUBLE
TUNIT5	'radians '	physical unit of field
TTYPE6	'par_angle'	label for field 6
TFORM6	'D '	data format of field: 8-byte DOUBLE
TUNIT6	'radians '	physical unit of field
TTYPE7	'derot_angle'	label for field 7
TFORM7	'D '	data format of field: 8-byte DOUBLE
TUNIT7	'radians '	physical unit of field
TTYPE8	'flag_cal'	label for field 8
TFORM8	'J '	data format of field: 4-byte INTEGER
TTYPE9	'flag_track'	label for field 9
TFORM9	'J '	data format of field: 4-byte INTEGER
TTYPE10	'weather '	label for field 10
TFORM10	'3D '	data format of field: 8-byte DOUBLE
TTYPE11	'Ch0 '	label for field 11
TFORM11	'8192D '	data format of field: 8-byte DOUBLE
TTYPE12	'Ch1 '	label for field 12
TFORM12	'8192D '	data format of field: 8-byte DOUBLE
TTYPE13	'Ch2 '	label for field 13
TFORM13	'8192D '	data format of field: 8-byte DOUBLE
TTYPE14	'Ch3 '	label for field 14
TFORM14	'8192D '	data format of field: 4-byte REAL
EXTNAME	'DATA TABLE'	name of this binary table extension

1.5 ANTENNA TEMP TABLE

It contains N columns (Ch0, ..., ChN) with the signal converted in antenna temperature (K). Conversion is performed using a counts-to-K factor retrieved from the last available T_{sys} measurement. This means that the conversion factor, if the T_{sys} value had been achieved in a distant time or position w.r.t. the data stream, could be obsolete and/or not applicable to the data! Pay much attention to the usage of this table, as discussed in the "Observing in Medicina with ESCS" user's guide.

→ Notice: XARCOS is not yet provided with the possibility to perform T_{sys} measurements. The ANTENNA TEMP TABLE data streams are thus filled with 0.0 values.

Header example:

KEYWORD	EXAMPLE VALUE	DESCRIPTION
XTENSION	'BINTABLE'	binary table extension
BITPIX	8	8-bit bytes
NAXIS	2	2-dimensional binary table
NAXIS1	64	width of table in bytes
NAXIS2	6	number of rows in table
PCOUNT	0	size of special data area
GCOUNT	1	one data group (required keyword)
TFIELDS	4	number of fields in each row
TTYPE1	'Ch0 '	label for field 1
TFORM1	'2D '	data format of field: 8-byte DOUBLE
TUNIT1	'K '	physical unit of field
TTYPE2	'Ch1 '	label for field 2
TFORM2	'2D '	data format of field: 8-byte DOUBLE
TUNIT2	'K '	physical unit of field
TTYPE3	'Ch2 '	label for field 3
TFORM3	'2D '	data format of field: 8-byte DOUBLE
TUNIT3	'K '	physical unit of field
TTYPE4	'Ch3 '	label for field 4
TFORM4	'2D '	data format of field: 8-byte DOUBLE
TUNIT4	'K '	physical unit of field
EXTNAME	'ANTENNA TEMP TABLE'	name of this binary table extension

1.6 SERVO TABLE

It contains the readout of each axis for every servo system involved.
Details will be provided in the next release of this document.

1.7 summary.FITS contents

Each scan is provided with an additional FITS file, whose goal is to sum up the most useful information. This file is always named summary.fits and is exploited for archiving/querying procedures.
It contains only a primary header, at the moment listing the following keywords:

KEYWORD	EXAMPLE VALUE	DESCRIPTION
SIMPLE	T	file does conform to FITS standard
BITPIX	8	number of bits per data pixel
NAXIS	0	number of data axes
EXTEND	T	FITS data set may contain extensions
DATE	'2015-06-11T07:33:48'	File creation date (YYYY-MM-DDThh:mm:ss UT)
[HIERARCH BackendName]	'NULL '	Backend name
[CREATOR]	‘NULL ‘	Software (incl. version)
HIERARCH Declination	1.06475750272639	Target J2000 declination (radians)
[EQUINOX]	0.	Equinox of RA, Dec
[EXPTIME]	0.	Total integration time (seconds)
FITSVER	‘V.1.01 ‘	FITS version
[LST]	0.	Local sidereal time
[HIERARCH LogFileName]	‘NULL ‘	Name of the log file
[HIERARCH NUSEBANDS]	0	Number of sections
OBJECT	‘W3OH ‘	Target source name
[OBSID]	‘NULL ‘	Observer or operator initials
[PROJID]	‘NULL ‘	ProjectID
HIERARCH RESTFREQ1	22235.07985	Rest Frequency (MHz)
HIERARCH RESTFREQ2	22235.07985	Rest Frequency (MHz)
HIERARCH RESTFREQ3	22235.07985	Rest Frequency (MHz)
HIERARCH RESTFREQ4	22235.07985	Rest Frequency (MHz)
HIERARCH ReceiverCode	‘KKG ‘	Receiver name
[SCANGEOM]	‘NULL ‘	Scan geometry
[SCANMODE]	‘NULL ‘	Mapping mode
[SCANTYPE]	‘NULL ‘	Scan type
[SCANXVEL]	0.	Scanning rate (optional)
[SWTCHMOD]	‘NULL ‘	Switch mode
[HIERARCH ScheduleName]	‘NULL ‘	Name of schedule
[TELESCOP]	‘NULL ‘	Telescope name
VDEF	‘OP ‘	Radial velocity definition
VFRAME	‘LSRK ‘	Radial velocity reference frame
VRAD	-48.8	Radial velocity (km/s)
WOBUSED	0.	Wobbler used?

2 MBFITS

The Multi-Beam FITS format has been conceived, as its name suggests, to handle multi-beam observations.
Following the official document listing its features (APEX Report APEX-MPI-ICD-0002) the MBFITS produced by ESCS is a hierarchical structure of FITS files, each devoted to the storage of a different set of data and environmental info acquired during the observation. Details on the system component MBFitsWriter can be found inside the IRA Technical Report 461/12.

The MBFITS hierarchical grouping directory structure is defined as follows:

- Main directory name

- 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, one for each subscan, named according to the subscan number.
  Each subdirectory contains the following types of member files:
  - The MONITOR table file: **MONITOR**.fits
  - One ARRAYDATA table file for each FEBE combination and baseband: **<FEBE name>-ARRAYDATA-<Baseband number>**.fits
  - One DATAPAR table file for each FEBE combination: **<FEBE name>-DATAPAR**.fits

2.0 GROUPING Table

This table exists only in the hierarchical implementation of the MBFITS format and it is created once for each scan. It is used to store the locations of the member files, plus other details which can be exploited to speed up searching when reading the files.

2.1 SCAN Table

It is stored for every scan. It contains parameters which do not change among the subscans, including:

- telescope and observatory parameters
- time system
- coordinate system
- velocity system
- project ID
- target of the scan and its coordinates
- observing mode
- pointing model coefficients

2.2 FEBEPAR Table

The FEBEPAR table is stored per FEBE (FrontEnd–BackEnd) combination for each scan and contains the frontend-backend setup. Parameters common to all FEBEs are written in the SCAN table.
It includes:

- FEBE setup: number of pixels, polarisations and basebands
- pointing model coefficients specific to this FE
- calibration parameters specific to this FEBE

2.3 ARRAYDATA Table

A new ARRAYDATA table is created for each subscan, for each FEBE and for each baseband. It stores the data description (header) and the data (table).
It includes:

- frequency band setup: frequency, bins (freq. channels), polarisations, line ID

- data axes description

If some parameters change for the individual subscan with respect to the general value stored in the SCAN table, data analysis applications should get these values from the ARRAYDATA table rather than from the SCAN one.

2.4 DATAPAR Table

A new DATAPAR table is created for each subscan and for each FEBE.
Parameters common to all the subscans are written in the SCAN table, while the FEBE setup is recorded in the FEBEPAR table (also assumed to be constant for all subscans).
The DATAPAR table contains those data-associated parameters which change with the integration, but not the data themselves – as they are stored in the ARRAYDATA table.
The table includes:

- time and coordinates 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 the MJD timestamp.

2.5 MONITOR Table

This table stores raw monitoring data (real-time updates other than the backend data) at their natural rate, i.e. 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.
It is recommended that the telescope control system should call for updates on monitor points at least at the beginning and end of the scans. As many of these as possible should be measured at these times. For points where a new measurement is not possible the last measurement should be saved again 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/end of scans: calibration data, pointing data, radiometer data, weather station data
- at the beginning of integrations: frequencies, current real antenna positions
- at the end of observations: current real antenna positions.