SUN/259.1

Starlink Project
Starlink User Note 259.1

Jen Balfour

9th April 2008

GSD2ACSIS
Version 1.1
Programmer's Guide

Abstract

GSD2ACSIS is a SMURF routine used to convert DAS and AOSC data stored in the GSD file format into the format used for ACSIS data.

Contents

1 Introduction

GSD2ACSIS has the task of mapping the GSD data, stored in headers and arrays, into the ACSIS format output file(s). Due to the differences in the two file formats, there is not a one-to-one mapping between the two. This document provides the mapping used to obtain the various headers and arrays found in the ACSIS format files, from those given in the GSD input file.

2 ACSIS File Format

ACSIS files use the NDF file format and consist of three primary extensions, the MORE extension (with the ACSIS, JCMTSTATE, and FITS extensions), the WCS extension (world coordinates of RA/Dec and frequency) and the data array.

See the ACSIS File Format interface Control Document (OCS/ICD/022) for more details.

3 GSD File Format

The DAS and AOSC backends wrote files in the GSD file format. GSD files contain lists of scalar and array data elements, prefaced by a prolog which describes the data and is used to retrieve it. The Starlink GSD library contains routines for accessing the data within GSD format files.

See GSD - Global Section Datafile System and The Global Section Datafile(GSD) access library for more details.

For a full list of scalar and array data stored in DAS and AOSC GSD files see the Appendix of this document.

4 ACSIS NDF Extension

RECEPTORS

The receptors array contains the names of the frontend receptors. These names can be determined by checking the name of the frontend and the number of receptors used. The name of the frontend is stored in the FRONTEND (C1RCV) header, and the number of receptors is determined from the values found in the DAS_MIXER (C3MIXNUM) array. Some GSD input files do not contain a DAS_MIXER (C3MIXNUM) array. In these cases, a guess at the number of receptors is made from examining the contents of the BES_NUOBS (C12CF) array. If the second half of the BES_NUOBS (C12CF) array contains the same values as those found in the first half, it is likely that two receptors were used.

The receptor naming convention used by GSD2ACSIS is to use a single letter name for single-receptor observations:

- **Receptor RXA:** 'A'
- **Receptor RXB:** 'B'
- **Receptor RXW and BES_NUOBS (C12CF)** $<$ **600 :** 'C'
- **Receptor RXW and BES_NUOBS (C12CF)** $>$ **600 :** 'D'
- **Receptor MPI:** 'E'

For observations with two receptors, the single letter is followed by either 'A' or 'B' to indicate the different receptors (e.g. 'BA' and 'BB').

FOCAL STATION

'DIRECT' (default).

FPLANEX

0.0 (default).

FPLANEY

0.0 (default).

RECEPPOS_SYS

The system for the receptor positions defaults to 'TRACKING' except when the CELL_COORDS (C4LSC) is 'AZEL', in which case the RECEPPOS_SYS is also 'AZEL'.

RECEPPOS

The absolute positions of the receptors are the tracking coordinates, and so the values for RECEPPOS are copied from TCS_TR_AC1 and TCS_TR_AC2.

TSYS

For continuous calibration observations the BES_T_SYS (C12SST) array contains temperatures for each backend section at each time step, for other observations this array contains one temperature per backend section. In continuous calibration observations the TSYS for each receptor & time step is the corresponding value from the BES_T_SYS (C12SST) array, for other observations the values are copied for each time step.

TRX

The receiver temperatures for each backend section are retrieved from the BES_T_REC (C12RT) array.

5 JCMTSTATE NDF Extension

The JCMTSTATE extension contains the timestep-specific data.

RTS_NUM

The RTS sequence number increments at each time step of the observation, beginning with RTS_NUM = 1 for the first time step. The total number of RTS steps is:

$$NSCAN \times NO\_SCAN\_PNTS \quad (1)$$

(NSCAN (C3NSAMPLE), NO_SCAN_PNTS (C3MXP))

RTS_END

The GSD format files recorded one LST time for each time step, as well as a scan time. Assuming that the recorded LSTs represent the middle of the time step, the RTS_END can be calculated by adding half the length of the time step to the TCS_TAI of this step (see below for TCS_TAI calculation).

For rasters, the SCAN_TIME (C3SRT) is the entire time for one raster across the sky, so the RTS_END in this case is:

$$RTS\_END = TCS\_TAI + (SCAN\_TIME/(NO\_SCAN\_PNTS \times 2)) \quad (2)$$

For grids, the RTS_END is simply:

$$RTS_END = TCS_TAI + (SCAN_TIME/2) \quad (3)$$

(SCAN_TIME (C3SRT), NO_SCAN_PNTS (C3MXP))

RTS_TASKS

For each time step this defaults to 'PTCS FE_X DAS' where X is A, B, C, D or E depending on the frontend (see RECEPTORS in the ACSIS extension).

SMU_X, SMU_Y, SMU_Z

0.0 (default).

SMU_CHOP_PHASE

'M' (default).

SMU_JIG_INDEX

Set to VAL__BADI (integer bad value).

SMU_AZ_JIG_X, SMU_AZ_JIG_Y, SMU_AZ_CHOP_X, SMU_AZ_CHOP_Y, SMU_TR_JIG_X, SMU_TR_JIG_Y, SMU_TR_CHOP_X, SMU_TR_CHOP_Y

0.0 (default).

TCS_TAI

The TCS_TAI records the TAI for each time step. The GSD format files record the starting LST time, as well as the LST time for each time step (assumed to be the LST at the middle of the time step). To calculate the TAI for each step, we first convert the starting LST to a TAI, then add the TAI time offset for each time step by converting the 'dLST' (difference between the LST of this step and the starting LST) to a TAI offset.

To obtain the starting TAI, the following headers from the GSD file are used:

- OBS_UT1D (C3DAT): UT1 date of observation
- OBS_UT1H (C3UT): UT1 hour of observation
- OBS_UT1C (C3UT1C): UT1-UTC correction

- TEL_LONGITUDE (C1LONG): telescope longitude
- TEL_LATITUDE (C1LAT): telescope latitude

The LSTs for each time step are stored in the SCAN_TABLE1 (C12SCAN_TABLE_1) array.

For rasters, only the LST of the beginning of the scan is recorded in SCAN_TABLE1 (C12SCAN_TABLE_1). In this case, the 'dLST' can be calculated from the LST of the beginning of the scan, the total time for the scan (SCAN_TIME (C3SRT)) and the number of points in the scan (NO_SCAN_PNTS (C3MXP)). This assumes that the time between each spectrum in the scan is equal (scanning at a constant velocity).

TCS_AIRMAS

The airmass at each time step is determined from the elevation (TCS_AZ_AC1) using Hardie's (1962) polynomial fit to Bemporad's data for the relative air mass in units of thickness at the zenith as tabulated by Schoenberg (1929).

TCS_AZ_ANG

The angle between PA=0 in the focal plane and PA=0 in AZEL defaults to 0.0 because all instruments are in the receiver cabin.

TCS_AZ_AC1/TCS_AZ_AC2

The actual coordinates in AZEL are determined from the CELL setup of the GSD input file and the WCS Frameset which maps CELL offsets to RA/Dec and Az/El offsets (see WCS NDF Extension).

TCS_AZ_DC1/TCS_AZ_DC2

The demand coordinates are assumed to be the same as the actual coordinates.

TCS_AZ_BC1/TCS_AZ_BC2

The base coordinates are calculated as are the TCS_AZ_AC1 and TCS_AZ_AC2 above, but with an XY cell offset corresponding to the centre of the grid of cells (0,0).

TCS_BEAM

'M' (default).

TCS_INDEX

The index into the observing area for grids is the offset into the MAP_TABLE (C14PHIST) of the current observation. For rasters, the index increments at the beginning of each scan, so the value of TCS_INDEX is the offset into the MAP_TABLE (C14PHIST) divided by the number of points per scan. TCS_INDEX begins counting at 1.

TCS_SOURCE

'SCIENCE' (default).

TCS_TR_SYS

The name of the tracking coordinate frame is identified by the GSD CENTRE_CODE (C4CECO)and translated into a string representation of the coordinate system as follows:

- 1: 'AZEL'

- 3: 'HADEC' (Currently unsupported)
- 4: 'APP'
- 6: 'B1950'
- 7: 'J2000'
- 8: 'GAL'

TCS_TR_ANG

The angle between the focal plane and PA=0 and PA=0 in the tracking coordinate frame is calculated as follows:

$$SQSZ = \cos(PHI) \times \sin(HA) \tag{4}$$

$$CQSZ = \sin(PHI) \times \cos(DEC) - \cos(PHI) \times \sin(DEC) \times \cos(HA) \tag{5}$$

If both SQSZ and CQSZ are equal to 0, then CQSZ is set to 1. Then,

$$TCS_T R_A NG = \arctan(SQSZ/CQSZ) \tag{6}$$

TCS_TR_AC1/TCS_TR_AC2

The actual coordinates in RA/Dec are determined from the CELL setup of the GSD input file and the WCS Frameset which maps CELL offsets to RA/Dec and Az/El offsets (see WCS NDF Extension).

TCS_TR_DC1/TCS_TR_DC2

The demand coordinates are assumed to be the same as the actual coordinates.

TCS_TR_BC1/TCS_TR_BC2

The base coordinates are calculated as are the TCS_TR_AC1 and TCS_TR_AC2 above, but with an XY cell offset corresponding to the centre of the grid of cells (0,0).

JOS_DRCONTROL

0 (default).

ENVIRO_REL_HUM

The relative humidity is assumed to be equal to the value stored in HAMB (C5RH).

ENVIRO_PRESSURE

The relative humidity is assumed to be equal to the value stored in PAMB (C5PRS).

ENVIRO_AIR_TEMP

The air temperature is assumed to be equal to the value stored in TAMB (C5AT).

ACS_SOURCE_RO

'SPECTRUM_RESULT' (default).

ACS_NO_PREV_REF, ACS_NO_NEXT_REF, ACS_NO_ONS

Set to VAL__BADI (integer bad value).

ACS_EXPOSURE

The on-source integration times for rasters is given by:

$$ACS\_EXPOSURE = SCAN\_TIME/NO\_SCAN\_PNTS \tag{7}$$

(SCAN_TIME (C3SRT), NO_SCAN_PNTS (C3MXP))

For other observations, the INTGRN_TIME (C3INTT) array stores integration times for each time step.

ACS_OFFEXPOSURE

The off-source integration times for rasters is given by:

$$ACS\_OFFEXPOSURE = ACS\_EXPOSURE \times \sqrt{NO\_SCAN\_PNTS} \tag{8}$$

(NO_SCAN_PNTS (C3MXP))

For other observations, the ACS_OFFEXPOSURE time is equal to the ACS_EXPOSURE time.

POL_ANG

Set to VAL__BADD (float bad value).

FE_LOFREQ

The frontend LO frequencies for each subband are stored in the BES_FE_NULO (C3BEFENULO) array.

FE_DOPPLER

The rest frequencies, LO frequencies and IF for each subband are stored in the BES_NUREST (C12RF), BES_FE_NULO (C3BEFENULO), and BES_TOT_IF (C3BETOTIF) arrays, respectively. From these, the frontend doppler correction can be calculated:

$$FE\_DOPPLER = REST\_FREQ/\left(LOFREQ + TOT\_IF\right) \tag{9}$$

6 FITS headers

The FITS headers give additional information about the observation required for ORAC-DR to reduce the data. In many cases there is an equivalent header in the original GSD data file.

6.1 Telescope Specific

TELESCOP

TEL_NAME (C1TEL).

ORIGIN

'Joint Astronomy Centre' (default).

OBSGEO-X, OBSGEO-Y, OBSGEO-Z

The cartesian coordinates of the telescope are calculated using the values stored in TEL_LATITUDE (C1LAT), TEL_LONGITUDE (C1LONG), and TEL_HEIGHT (C1HGT).

ALT-OBS

Convert TEL_HEIGHT (C1HGT) from km to m.

LAT-OBS

TEL_LATITUDE (C1LAT).

LONG-OBS

TEL_LONGITUDE (C1LONG).

ETAL

Convert ETAL (C8EL) from percentage to decimal.

6.2 OMP and ORAC-DR Specific

PROJECT

PROJECT (C1PID).

RECIPE

'REDUCE_SCIENCE' (default).

DRGROUP

AST__UNDEFS (string bad value).

MSBID

AST__UNDEFS (string bad value).

SURVEY

AST__UNDEFS (string bad value).

RMTAGENT

AST__UNDEFS (string bad value).

AGENTID

AST__UNDEFS (string bad value).

6.3 Observation, Date and Pointing Specific

OBSID

The OBSID is generated using the BACKEND (C1BKE) header, the observation number, and the date of the observation. For observations after February 2nd, 2003, the observation number found in the GSD file's NOBS (C1SNO) header can be used, prior to this date the user provides the observation number as an input to GSD2ACSIS (see OBSNUM). The date of the observation provided by OBS_UT1D (C3DAT) and OBS_UT1H (C3UT) headers are stored in UT1 time and must be converted to UTC time using OBS_UT1C (C3UT1C).

OBSIDSS

The OBSIDSS header simply appends the subsystem number to the OBSID.

OBJECT

GSD files store two strings for the source name: CENTRE_NAME_1 (C1SNA1) and CENTRE_NAME_2 (C1SNA2). These two names are concatenated to produce the OBJECT header.

STANDARD

The STANDARD flag indicates whether or not the observation was of a spectral line standard. The OBJECT name is compared with a lookup table of standards to determine if STANDARD is true.

OBSNUM

Prior to Feb 2nd, 2003, observation numbers were assigned to individual projects which could span several nights, and data files are stored in subdirectories according to project name. After Feb 2nd, 2003, observation numbers changed from night to night, and data files are stored in subdirectories by date. The observation number for data collected prior to Feb 2nd, 2003 should be determined from the project name and date, and must be supplied as a user input to GSD2ACSIS, whereas for later observations the GSD file's NOBS (C1SNO) header can be used.

NSUBSCAN

For longer observations split over several files, the NSUBSCAN would represent this file's place in the sequence of data. For GSD2ACSIS converted files we would expect this number to always be 1 as only one output file per subsystem would be written.

OBSEND

The OBSEND flag indicates whether or not the file is the last in the current observation. For GSD2ACSIS converted files we would expect this number to always be 1 as only one output file per subsystem would be written.

UTDATE

The UTDATE is a 8 digit integer represnting the UT date of the observation in YYYYMMDD format. It is obtained from the OBS_UT1D (C3DAT) header.

DATE-OBS

The DATE-OBS header is a string representation of the UTC datetime of the start of the observation in the format YYYY-MM-DDTHH:MM:SS (eg. 2004-01-01T01:01:01). The GSD OBS_UT1D (C3DAT) and OBS_UT1H (C3UT) headers are used to create the DATE-OBS header, but as these

GSD headers are in UT1 time, the OBS_UT1C (C3UT1C) header is used to convert to a UTC datetime.

DATE-END

The DATE-END is derived by adding the length of the observation to the DATE-OBS. The observation length is estimated by finding the difference between the recorded LSTs of the first and last time steps in the SCAN_TABLE1 (C12SCAN_TABLE_1) array.

DUT1

OBS_UT1C (C3UT1C).

INSTAP

AST__UNDEFS (string bad value).

**NOTE: Possibly this should be the same as the value stored in REFRECEP.

INSTAP_X

0.0 (default).

INSTAP_Y

0.0 (default).

AMSTART

The airmass at observation start corresponds to the JCMTSTATE TCS_AIRMAS of the first time step.

AMEND

The airmass at observation end corresponds to the JCMTSTATE TCS_AIRMAS of the last time step.

AZSTART

The azimuth at observation start corresponds to the JCMTSTATE TCS_AZ_AC1 of the first time step.

AZEND

The azimuth at observation end corresponds to the JCMTSTATE TCS_AZ_AC1 of the last time step.

ELSTART

The elevation at observation start corresponds to the JCMTSTATE TCS_AZ_AC2 of the first time step.

ELEND

The elevation at observation end correspond to the JCMTSTATE TCS_AZ_AC2 of the last time step.

HSTSTART

The HST at observation start is simply the DATE-OBS converted to HST.

HSTEND

The HST at observation end is simply the DATE-END converted to HST.

LSTSTART

The LST of the start of the observation is the first LST stored in the SCAN_TABLE1 (C12SCAN_TABLE_1).

LSTEND

The LST of the end of the observation is the last LST stored in the SCAN_TABLE1 (C12SCAN_TABLE_1).

6.4 Integration Time Specific

INT_TIME

For rasters, the total integration time is the time per scan, SCAN_TIME (C3SRT) multiplied by the number of complete scans, NSCAN (C3NSAMPLE). For other observations, the total integration time is the sum of the integration times found in the INTGRN_TIME (C3INTT) array.

6.5 ACSIS Specific

BACKEND

BACKEND (C1BKE).

MOLCULE, TRANSITI

The MOLECULE and TRANSITI headers are found by comparing the rest frequency against a lookup table of molecular transition lines.

DRRECIPE

AST__UNDEFS (string bad value).

BWMODE

The bandwidth mode takes the format '<BANDWIDTH>MHzx<CHANNELS>' (e.g. '250MHzx2048') where BANDWIDTH is the value for this subsystem in the BES_BANDWIDTH (C12BW) array, and CHANNELS is the value for this subsystem in the NO_BES_O_CH (C3LSPC) array.

SUBSYSNR

The number of the current subsystem.

SUBBANDS

Same as BWMODE.

NSUBBAND

1 (default).

SUBREFP1

Same as REFCHAN (see below).

SUBREFP2

AST__UNDEFF (float bad value).

NCHNSUBS

The number of channels for this subband in the NO_BES_O_CH (C3LSPC) array.

REFCHAN

The reference channel is calculated as follows:

$$REFCHAN = (NUMCHANNELS/2) + ((IF - TOTIF) / (|FREQ\_RESOLUTION|/1000.0)); \quad (10)$$

where NUMCHANNELS, TOTIF, and FREQ_RESOLUTION are the values for this subband in the NO_BES_O_CH (C3LSPC), BES_TOT_IF (C3BETOTIF) and BES_DELTANU (C12FR) arrays, respectively, and IF is as defined in IFFREQ below.

IFCHANSP

Value from the BES_DELTANU (C12FR) array for this subband.

FFT_WIN

'truncate' (string bad value).

BEDEGFAC

For the DAS backend, BEDEGFAC is 1.15, for AOSC it is 1.0.

MSROOT

AST__UNDEFS (string bad value).

6.6 Front End Specific

INSTRUME

FRONTEND (C1RCV).

**NOTE: Receiver RxG is not currently supported by GSD2ACSIS.

SB_MODE

C3SBMODE (C3SBMODE).

IFFREQ

The IF for special observations for each group of subbands to be merged is defined to be the average of the values in the BES_TOT_IF (C3BETOTIF) array for those subbands. **NOTE** currently GSD2ACSIS treats each subband in a special observation separately and so each subband uses its corresponding value in BES_TOT_IF (C3BETOTIF) for its IF. For non-special observations, the IF is 4.0 GHz, with the exception of the MRI frontend which used an IF of 3.5 GHz.

N_MIX

The number of mixers can be determined from the frontend. For RxB and RxW frontends, the N_MIX is 2, for all others N_MIX is 1.

OBS_SB

If the sideband sign found in the BES_FE_SB_SIGN (C3BEFESB) array for this subband is positive, OBS_SB = 'USB', otherwise OBS_SB = 'LSB'.

LOFREQS, LOFREQE

The value found in the BES_FE_NULO (C3BEFENULO) array for this subband.

Because the GSD data does not record both a starting and ending LO frequency, the same value is used for LOFREQE as for LOFREQS.

RECPTORS

Receptor names come from the name of the FRONTEND, and the mixers used in the observation as listed for each subband in the DAS_MIXER (C3MIXNUM) array. All receptors used in the observation are listed in a comma separated list in RECEPTORS (eg. 'BA, BB' would indicate that both receptors of an RXB frontend were used).

**NOTE: Receiver RxG is not currently supported by GSD2ACSIS.

REFRECEP

REFRECEP is the first of the receptors listed in RECPTORS.

MEDTSYS

For sample observations, MEDTSYS is AST__UNDEFF (float bad value), for all other observations it is the value stored in the BES_T_SYS (C12SST) array for this subsystem.

TEMPSCAL

'TA*' (default).

DOPPLER

VEL_DEFN (C12VDEF).

SSYSOBS

'TOPOCENT' (default).

6.7 Environment Specific

ATSTART, ATEND

TAMB (C5AT).

Because the GSD data does not record both a starting and ending air temperature, the same value is used for ATEND as for ATSTART.

HUMSTART, HUMEND

HAMB (C5RH).

Because the GSD data does not record both a starting and ending relative humidity, the same value is used for HUMEND as for HUMSTART.

BPSTART, BPEND

PAMB (C5PRS).

**NOTE: GSD files state that PAMB (C5PRS) is in units of mmHg, although it is actually in mbar, so no conversion is required.

Because the GSD data does not record both a starting and ending air pressure, the same value is used for BPEND as for BPSTART.

WNDSPDST

AST__UNDEFF (float bad value).

WNDSPDEN

AST__UNDEFF (float bad value).

WNDDIRST

AST__UNDEFF (float bad value).

WNDDIREN

AST__UNDEFF (float bad value).

TAU225ST, TAU225EN

CSO_TAU (C7TAU225).

Because the GSD data does not record both a starting and ending TAU225 value, the same value is used for TAU225EN as for TAU225ST.

TAUDATST, TAUDATEN

CSO_YYMMDDHHMM (C7TAUTIME).

Because the GSD data does not record both a starting and ending TAU225 value, the same value is used for TAUDATEN as for TAUDATST.

TAUSRC

’CSO225GHZ’ (default).

WVMTAUST

AST__UNDEFF (float bad value).

WVMTAUEN

AST__UNDEFF (float bad value).

WVMDATST

AST__UNDEFF (float bad value).

WVMDATEN

AST__UNDEFF (float bad value).

SEEINGST, SEEINGEN

SAO_SEEING (C7SEEING).

Because the GSD data does not record both a starting and ending seeing, the same value is used for SEEINGEN as for SEEINGST.

SEEDATST, SEEDATEN

SAO_YYMMDDHHMM (C7SEETIME).

Because the GSD data does not record both a starting and ending seeing, the same value is used for SEEINGEN as for SEEINGST.

FRLEGTST

AST__UNDEFF (float bad value).

FRLEGTEN

AST__UNDEFF (float bad value).

BKLEGTST

AST__UNDEFF (float bad value).

BKLEGTEN

AST__UNDEFF (float bad value).

6.8 Switching and Mapping Details

SAM_MODE

If the GSD header OBS_TYPE (C6ST) is 'RASTER', SAM_MODE is 'raster'. If OBS_TYPE (C6ST) is 'SAMPLE', SAM_MODE is 'sample'. In all other cases, SAM_MODE is 'grid'.

SW_MODE

The Switch Mode header is generated by checking the following GSD headers:

SWITCH_MODE (C6MODE)
CHOPPING (C4SM)
REFERENCE_X (C4RX)
REFERENCE_Y (C4RY)

C6MODE	*C4SM*	*C4RX = C4RY = 0*	*SW_MODE*	*Warnings*
'POSITION_SWITCH'	TRUE	TRUE	'freq'	-
'POSITION_SWITCH'	TRUE	FALSE	'pssw'	-
'POSITION_SWITCH'	FALSE	TRUE	'freq'	'Likely intended to be a frequency switch.'
'POSITION_SWITCH'	FALSE	FALSE	'pssw'	-
'BEAMSWITCH'	TRUE	TRUE/FALSE	'chop'	-
'BEAMSWITCH'	FALSE	TRUE/FALSE	'none'	'May be an error.'
'CHOPPING'	TRUE	TRUE/FALSE	'freq'	'May be a misconfigured frequency switch.'
'CHOPPING'	FALSE	TRUE/FALSE	'freq'	-
'NO_SWITCH'	TRUE	TRUE/FALSE	'none'	'May be an error.'
'NO_SWITCH'	FALSE	TRUE/FALSE	'none'	-

SKYREFX

SKYREFX takes the form '[OFFSET] <REFX> [LOCL_CRD]' (eg. [OFFSET] 180 [J2000]') where REFX is the value found in REFERENCE_X (C4RX) and LOCL_CRD is the value from the FITS LOCL_CRD header.

SKYREFY

SKYREFY takes the form '[OFFSET] <REFY> [LOCL_CRD]' (eg. [OFFSET] 0 [J2000]') where REFY is the value found in REFERENCE_Y (C4RY) and LOCL_CRD is the value from the FITS LOCL_CRD header.

OBS_TYPE

If the GSD OBS_TYPE (C6ST) is 'FIVEPOINT', the ACSIS OBS_TYPE is 'pointing'. If the GSD OBS_TYPE (C6ST)is 'FOCUS', the ACSIS OBS_TYPE is 'focus'. For all other observations, the ACSIS OBS_TYPE is 'science'.

The following 4 headers are only defined for observations where SAM_MODE is 'grid' and SW_MODE is either 'chop' or 'sample'.

CHOP_CRD

The name of the chopping coordinate is translated to the modern name from the value stored in COORDS (C4SMCO).

- 'AZ': 'AZEL'
- 'EQ': 'HADEC'
- 'RB': 'B1950'
- 'RJ': 'J2000'
- 'RD': 'APP'
- 'GA': 'GAL'

CHOP_FRQ

FREQUENCY (C4FRQ).

CHOP_PA

POSANG (C4POSANG).

CHOP_THR

THROW (C4THROW).

ROT_CRD

AST__UNDEFS (string bad value).

ROT_PA

AST__UNDEFF (float bad value).

JIGL_CNT

AST__UNDEFI (integer bad value).

JIGL_NAM

AST__UNDEFS (string bad value).

JIG_PA

AST__UNDEFF (float bad value).

JIG_CRD

AST__UNDEFS (string bad value).

JIG_SCAL

AST__UNDEFF (float bad value).

The following 11 headers are only defined for observations where SAM_MODE is 'raster' and SW_MODE is 'pssw'.

MAP_HGHT

The map height is:

$$MAP\_HGHT = NO\_X\_MAP\_PNTS \times CELL\_X \quad (11)$$

(NO_X_MAP_PNTS (C6XNP), CELL_X (C6DX))

MAP_PA

The MAP_PA is defined to be the SCAN_PA - 90.0°.

MAP_WDTH

The map width is:

$$MAP\_WDTH = NO\_Y\_MAP\_PNTS \times CELL\_Y \quad (12)$$

(NO_Y_MAP_PNTS (C6YNP), CELL_Y (C6DY))

LOCL_CRD

The name of the local offset coordinate frame is identified by the GSD CELL_CODE (C6FC) and translated into the modern name:

- 1: 'AZEL'
- 3: 'HADEC' (Currently unsupported)
- 4: 'APP'
- 6: 'B1950'
- 7: 'J2000'
- 8: 'GAL'

MAP_X

CENTRE_OFFSET_X (C4SX).

MAP_Y

CENTRE_OFFSET_Y (C4SY).

SCAN_CRD

Same as LOCL_CRD.

SCAN_VEL

For observations where the OBS_DIRECTION (C6SD) is 'HORIZONTAL', the SCAN_VEL is:

$$SCAN\_VEL = CELL\_X \times NO\_X\_MAP\_PNTS/SCAN\_TIME \tag{13}$$

For observations where the OBS_DIRECTION (C6SD) is 'VERTICAL', the SCAN_VEL is:

$$SCAN\_VEL = CELL\_Y \times NO\_Y\_MAP\_PNTS/SCAN\_TIME \tag{14}$$

(CELL_X (C6DX), CELL_Y (C6DY), NO_X_MAP_PNTS (C6XNP), NO_Y_MAP_PNTS (C6YNP), SCAN_TIME (C3SRT))

SCAN_DY

For observations where the OBS_DIRECTION (C6SD) is 'HORIZONTAL', the SCAN_DY is CELL_Y (C6DY). For observations where the OBS_DIRECTION (C6SD) is 'VERTICAL', the SCAN_DY is CELL_X (C6DX).

SCAN_PA

CELL_V2X (C6MSA).

SCAN_PAT

If SCAN_REVERSAL (C6REV) is 1 (true), SCAN_PAT is 'BOUSTROPHEDON', otherwise SCAN_PAT is 'RASTER'.

6.9 Secondary Mirror Specific

ALIGN_DX

DX (C2FV).

ALIGN_DY

DY (C2FL).

FOCUS_DZ

DZ (C2FR).

DAZ

OFFS_EW (C4OFFS_EW).

DEL

OFFS_NS (C4OFFS_NS).

UAZ

SDIS(36) (UAZ).

UEL

SDIS(37) (UEL).

6.10 JOS Parameters

STEPTIME

For observations where SAM_MODE is 'raster' and OBS_DIRECTION (C6SD) is 'HORIZONTAL', STEPTIME is:

$$STEPTIME = SCAN\_TIME/NO\_X\_MAP\_PNTS \quad (15)$$

For observations where SAM_MODE is 'raster' and OBS_DIRECTION (C6SD) is 'VERTICAL', STEPTIME is:

$$STEPTIME = SCAN\_TIME/NO\_Y\_MAP\_PNTS \quad (16)$$

For all other observations, STEPTIME is SCAN_TIME (C3SRT).

(SCAN_TIME (C3SRT), NO_X_MAP_PNTS (C6XNP), NO_Y_MAP_PNTS (C6YNP))

NUM_CYC

NCYCLE (C3NCYCLE).

NUM_NODS

1 (default).

JOS_MULT

AST__UNDEFI (integer bad value).

JOS_MIN

For observations where SAM_MODE is 'sample', JOS_MIN is NSCAN (C3NSAMPLE). For all other observations, JOS_MIN is 1.

NCALSTEP

AST__UNDEFI (integer bad value).

NREFSTEP

When OBS_CONTINUOUS (C3FLY) is 1, NREFSTEP is calculated as follows:

$$NREFSTEP = \sqrt{NO\_SCAN\_PNTS} \times SCANTIME/NO\_SCAN\_PNTS \quad (17)$$

For all other observations, NREFSTEP is the value found in SCAN_TIME (C3SRT).

(NO_SCAN_PNTS (C3MXP), SCAN_TIME (C3SRT))

STBETREF

If SW_MODE is 'chop', STBETREF is undefined (AST__UNDEFI).

When OBS_CONTINUOUS (C3FLY) is 1, STBETREF is NO_SCAN_PNTS (C3MXP). For all other observations, STBETREF is 1.

STBETCAL

AST__UNDEFI (integer bad value).

STARTIDX

For observations where SAM_MODE is 'sample' or 'raster', STARTIDX is 1. In 'grid' observations, the STARTIDX is the index into the grid of the first pair of coordinates found in the MAP_TABLE (C14PHIST).

FOCAXIS

AST__UNDEFS (string bad value).

NFOCSTEP

AST__UNDEFI (integer bad value).

FOCSTEP

AST__UNDEFF (float bad value).

6.11 Miscellaneous

OCSCFG

AST__UNDEFS (string bad value).

SIMULATE

0 (default).

SIM_CORR

0 (default).

SIM_SMU

0 (default).

SIM_TCS

0 (default).

SIM_RT

0 (default).

SIM_IF

0 (default).

STATUS

'NORMAL' (default).

6.12 ROVER Polarimeter Specific

POL_CONN

0 (default).

POL_MODE

AST__UNDEFS (string bad value).

ROTAFREQ

AST__UNDEFF (float bad value).

POL_CRD

AST__UNDEFS (string bad value).

POLFAXIS

AST__UNDEFF (float bad value).

7 WCS NDF Extension

The world coordinate frameset used to map the spatial and frequency coordinates of the observation are written to an NDF extension in the output file(s). The frameset consists of a mapping from the GSD CELL coordinates to both tracking RA/Dec and Az/El coordinates. The third axis represents the frequency scale of the observation.

The mapping from GSD CELLs to ACSIS TCS tracking and AZEL coordinates is created using the frameset from the GSD CELL headers and the GSD CENTRE coordinates given for the source's position at the time of the observation, in combination with the LST and CELL offsets of each time step.

In order to create this frameset, several components from the original GSD file are required:

- Source RA/Dec, telescope location, and observation time.
- Cell definition.
- Frequency scale.
- Velocity definition.

GSD's CELL coordinates describe a grid placed upon the sky with a centre cooresponding to the location described in the CENTRE_RA1950 (C4ERA), CENTRE_DEC1950 (C4EDEC), CENTRE_RA (C4RADATE), CENTRE_DEC (C4DECDATE), CENTRE_RA2000 (C4RA2000), CENTRE_DEC2000 (C4EDEC2000), CENTRE_GL (C4GL), CENTRE_GB (C4GB), CENTRE_AZ (C4AZ), and CENTRE_EL (C4EL) coordinates. These pairs of coordinates are assumed to correspond with the time given in the OBS_UT1D (C3DAT) and OBS_UT1H (C3UT) headers.

The CENTRE_CODE (C4CECO) (coordinate system of the tracking frame) along with the corresponding coordinate pair from those listed above is used as the RA/Dec input, along with the TEL_LONGITUDE (C1LONG) and TEL_LATITUDE (C1LAT) to create the frameset.

7.1 Source RA/Dec, telescope location, and observation time

The description of the CELL coordinate frame comes from the GSD CELL headers:

- CELL_COORDS (C4LSC) and CELL_CODE (C6FC): cell coordinate system
- CELL_X (C6DX): size of cell in x dimension (arcsec)
- CELL_Y (C6DY): size of cell in y dimension (arcsec)
- CELL_V2Y (CELL_V2Y): position angle of cell y axis (degrees counterclockwise)

7.2 Frequency scale

The third axis of the WCS frameset is the frequency scale. The GSD values used to generate this frameset are:

- Frequency of the transition line for this subband (for special observations, this is the value for this subband in BES_NUREST (C12RF), for all other observations this is the value for this subband in BES_NUOBS (C12CF).
- BES_NUOBS (C12CF): Centre frequency for this subband.
- BES_DELTANU (C12FR): Frequency resolution for this subband.
- IF for this subband. The IF should correspond with the definition of the REFCHAN.

7.3 Velocity definition

The WCS frameset requires two further inputs in order to be created. These are the velocity and the velocity definition ('LSRFLG'). For total power and cross correlation observations, the velocity is assumed to be 0, while for other observations the value in VELOCITY (C7VR) is used. The velocity definition is a flag which defines both the rest frame of the source velocity (topocentric, kinematic LSR, heliocentric, or geocentric, given by the bottom 4 bits of 'LSRFLG') and the source velocity definition (radio, optical, or relativistic, given by bits 5 and 6 of 'LSRFLG').

7.4 Using the WCS frameset

The WCS frameset is used to generate the TCS tracking and AZEL base and offset coordinates found in the JCMTState NDF extension. At each time step of the observation, the time can be obtained from the SCAN_TABLE1 (C12SCAN_TABLE_1)array (see TCS_TAI

). The current cell offset is found in the list of xy coordinates stored for each scan in the MAP_TABLE (C14PHIST)array. Base coordinates are generated using a cell offset of 0, 0.

8 Suggested example GSD files

These are a selection of .gsd files used to test GSD2ACSIS. These files can be found in the JCMT archive.

- RXA 125 MHz single subband : 24 Jul 2005 scan 17 (m05au25)
- RXA 3x3 grid : 14 Jan 2004 scan 10 (m03bec06)
- RXA pattern : 08 Jul 2005 scan 64 (m05an11)
- RXA Raster in Galactic coordinates : 19 Aug 2002 scan 49 (m02bn13)
- RXA Special Configuration : 11 Jul 2005 scan 22 (m05bn27b)
- RXB raster : 08 Oct 2005 scan 20 (m05bu72)
- RXB wideband observation : 19 Oct 2004 scan 18 (m04bn10)

A Headers and Arrays in GSD DAS files

The following is a list of the headers and arrays found in GSD format (version 5.3) files in the order in which they appear in the file. Listed is the NRAO name, JCMT name, data type, description for each element. In addition, the dimensionality of the arrays for various observation types (Normal, Total Power, Continuous Calibration, Cross Correlation) is given (for scalar headers, the table indicates for which observation types the headers were used).

A.1 GSD Headers

NRAO name	*JCMT name*	*Data type*	*Description*
C1TEL	TEL_NAME	char*	Telescope name
C1PID	PROJECT	char*	Identifies the observing program
C1OBS	PROJECT_OBS_1	char*	Name of the primary observer
C1ONA1	PROJECT_OBS_2	char*	Name of the support scientist
C1ONA2	PROJECT_OBS_3	char*	Name of the telescope operator
C1SNA1	CENTRE_NAME_1	char*	Source name part 1
C1SNA2	CENTRE_NAME_2	char*	Source name part 2 or altern. name
C4CSC	CENTRE_COORDS	char*	Character code of commanded centre or source coordinate system
C4CECO	CENTRE_CODE	int	Centre coords. AZ=1; EQ=3; RD=4; RB=6; RJ=7; GA=8
C4EPT	EPOCH_TYPE	char*	Type of epoch, JULIAN, BESSELIAN or APPARENT
C4MCF	CENTRE_MOVING	char	Centre moving flag (solar system object)
C4EPH	CENTRE_EPOCH	double	Date of the RA
C4ERA	CENTRE_RA1950	double	Right ascension of source for EPOCH
C4EDEC	CENTRE_DEC1950	double	Declination of source for EPOCH
C4RADATE	CENTRE_RA	double	Right Ascension of date
C4DECDATE	CENTRE_DEC	double	Declination of date
C4RA2000	CENTRE_RA2000	double	Right ascension J2000
C4EDEC2000	CENTRE_DEC2000	double	Declination J2000
C4GL	CENTRE_GL	double	Galactic longitude
C4GB	CENTRE_GB	double	Galactic latitude
C4AZ	CENTRE_AZ	double	Azimuth at observation date
C4EL	CENTRE_EL	double	Elevation at observation date
C4LSC	CELL_COORDS	char*	har. code for local x-y coord.system

C6FC	CELL_CODE	int	Local x-y AZ=1; EQ=3; RD=4; RB=6; RJ=7; GA=8
C4ODCO	CELL_UNIT	char*	Units of cell and mapping coordinates;offset definition code
C6DX	CELL_X	double	Cell x dim,; descriptive origin item 1
C6DY	CELL_Y	double	Cell y dimension; descriptive origin item 2
C6MSA	CELL_V2X	double	Scanning angle - angle from local vertical to x axis measured CW
CELL_V2Y	CELL_V2Y	double	Position angle of cell y axis (CCW)
C4AXY	CELL_X2Y	double	Angle between cell y axis and x-axis (CCW)
C4SX	CENTRE_OFFSET_X	double	Commanded x centre position (JCMT cells wrt to centre; NRAO abs. degrees)
C4SY	CENTRE_OFFSET_Y	double	Commanded y centre position (JCMT cells wrt to centre; NRAO abs. degrees)
C4RX	REFERENCE_X	double	Reference x position (JCMT cells wrt to centre; NRAO abs. degrees)
C4RY	REFERENCE_Y	double	Reference y position (JCMT cells wrt to centre; NRAO abs. degrees)
C1HGT	TEL_HEIGHT	double	Height of telescope above sea level
C1LONG	TEL_LONGITUDE	double	Geographical longitude of telescope (West +ve)
C1LAT	TEL_LATITUDE	double	Geodetic latitude of telescope (North +ve)
C1SNO	NOBS	double	Observation number
C6ST	OBS_TYPE	char*	Type of observation
C1RCV	FRONTEND	char*	Name of the frontend
C1FTYP	FE_TYPE	char*	Type of frontend
C1BKE	BACKEND	char*	Name of the backend

C1BTYP	BE_TYPE	char*	Type of backend
C3DAT	OBS_UT1D	double	UT1 date of observation
C3UT	OBS_UT1H	double	UT1 hour of observation
C3UT1C	OBS_UT1C	double	UT1-UTC correction interpolated from time service telex (in days)
C3LST	OBS_LST	double	Local sidereal time at the start of the observation
C3CAL	OBS_CALIBRATION	char	Calibration observation?
C3CEN	OBS_CENTRE	char	Centre moves between scans?
C3FLY	OBS_CONTINUOUS	char	Data taken on the fly or in discrete mode?
C3FOCUS	OBS_FOCUS	char	Focus observation?
C3MAP	OBS_MAP	char	Map observation?
C3NPP	NO_MAP_DIMS	int	Number of dimension in the map table
C3NMAP	NO_MAP_PNTS	int	Number of map points
C6XNP	NO_X_MAP_PNTS	int	X map dimension; number of points in the x-direction
C6YNP	NO_Y_MAP_PNTS	int	Y map dimension; number of points in the y-direction
C6XGC	X_MAP_START	float	X coordinate of the first map point
C6YGC	Y_MAP_START	float	Y coordinate of the first map point
C6REV	SCAN_REVERSAL	char	Map rows scanned in alternate directions?
C6SD	OBS_DIRECTION	char*	Map rows are in X (horizontal) or Y(vertical) direction
C6XPOS	X_MAP_POSITIVE	char	In first row x increases (TRUE) or decreases (FALSE)
C6YPOS	Y_MAP_POSITIVE	char	In first row y increases (TRUE) or decreases (FALSE)
C3NIS	NO_SCANS	int	Number of scans
C3NSAMPLE	NSCAN	int	Number of scans done
C3NO_SCAN_VARS1	NO_SCAN_VARS1	int	Number of scan table 1 variables

C3NO_SCAN_VARS2	NO_SCAN_VARS2	int	Number of scan table 2 variables
C3SRT	SCAN_TIME	int	Total time of scan (=total integration time if OBS_CONTINUOUS = .FALSE.)
C3MXP	NO_SCAN_PNTS	int	Maximum number of map points done in a phase
C3NCI	NO_CYCLES	int	Maximum number of cycles in the scan
C3NCYCLE	NCYCLE	int	Number of cycles done in the scan
C3CL	CYCLE_TIME	int	Duration of each cycle
C3NCP	NO_CYCLE_PNTS	int	Total number of xy positions observed during a cycle
C6NP	NCYCLE_PNTS	int	Number of sky points completed in the observation
C3NSV	NO_PHASE_VARS	int	Number of phase table variables
C3PPC	NO_PHASES	int	Number of phases per cycle
C5AT	TAMB	double	Ambient temperature
C5PRS	PAMB	double	Mean atmospheric pressure
C5RH	HAMB	double	Mean atmospheric relative humidity
C4AZERR	SDIS(7)	double	DAZ:Net Az offset at start (inc.tracker ball setting and user correction)
C4ELERR	SDIS(8)	double	DEL:Net El offset at start (inc.tracker ball setting and user correction)
UAZ	SDIS(36)	double	User az correction
UEL	SDIS(37)	double	User el correction
C7SZVRAD	SZVRAD	int	Number of elements of vradial array
C8AAE	APERTURE_EFF	double	Ratio total power observed
C8ABE	BEAM_EFF	double	Fraction of beam in diffraction limited main beam
C8GN	ANTENNA_GAIN	double	Antenna gain

C8EL	ETAL	double	Rear spillover and scattering efficiency
C8EF	ETAFSS	double	Forward spillover and scattering efficiency
C4SM	CHOPPING	char	Secondary mirror is chopping
C4FUN	WAVEFORM	char*	Secondary mirror chopping waveform
C4FRQ	FREQUENCY	float	Secondary mirror chopping period
C4SMCO	COORDS	char*	Secondary mirror chopping coordinate system
C4THROW	THROW	float	Secondary mirror chop throw
C4POSANG	POSANG	float	Secondary mirror chop position angle
C4OFFS_EW	OFFS_EW	float	Secondary mirror offset parallel to lower axis (East-West Tilt)
C4OFFS_NS	OFFS_NS	float	Secondary mirror offset parallel to upper axis (North-South Tilt)
C4X	X	float	Secondary mirror absolute X position at observation start
C4Y	Y	float	Secondary mirror absolute Y position at observation start
C4Z	Z	float	Secondary mirror absolute Z position at observation start
C4EW_SCALE	EW_AMPL_SCALE	float	Secondary mirror ew chop scale
C4NS_SCALE	NS_AMPL_SCALE	float	Secondary mirror ns chop scale
C4EW_ENCODER	AMPL_E_SET	int	Secondary mirror ew encoder value
C4NS_ENCODER	AMPL_N_SET	int	Secondary mirror ns encoder value
C2FV	DX	float	Secondary mirror x displacement from nominal at observation start
C2FL	DY	float	Secondary mirror y displacement from nominal at observation start

C2FR	DZ	float	Secondary mirror z displacement from nominal at observation start
C4MOCO	TEL_COORDS	char*	Mounting of telescope; defined as LOWER
C3NFOC	NO_FE_O_CH	int	NO_FE_O_CH:No. of frontend output channels
C7VR	VELOCITY	double	Radial velocity of the source
C12TCOLD	T_COLD	float	Cold load temperature
C12TAMB	T_HOT	float	Ambient load temperature
C12VDEF	VEL_DEFN	char*	Velocity definition code - radio, optical, or relativistic
C12VREF	VEL_REF	char*	Velocity frame of reference - LSR, Bary-, Helio-, or Geo- centric
C3NRC	NO_BE_I_CH	int	Number of backend input channels
C3NCH	NO_BE_O_CH	int	Number of backend output channels
C3NRS	NO_BES	int	Number of backend sections
C7BCV	BAD_CHANNEL	float	Bad channel value
C12CAL	DATA_UNITS	char*	Units of spectrum data
C6MODE	SWITCH_MODE	char*	Observation mode
C12CALTASK	BE_CAL_TASK	char*	Calibration instrument used (FE, BE, or USER)
C12CALTYPE	BE_CAL_TYPE	char*	Type of calibration (THREELOADS or TWOLOADS)
C12REDMODE	BE_RED_MODE	char*	Way of calibrating the data (RATIO or DIFFERENCE)
C3NOIFPBES	NO_IF_PER_BES	int	Number of IF inputs to each section (2 for correlator, 1 for AOS)
C3CONFIGNR	DAS_CONF_NR	int	Backend configuration
C3DASOUTPUT	DAS_OUTPUT	char*	Description of output in DAS DATA (SPECTRUM, T_REC, T_SYS, etc.)

C3DASCALSRC	DAS_CAL_SOURCE	char*	DAS calibration source for backend calibration (POWER or DATA)
C3DASSHFTFRAC	DAS_SHIFT_FRAC	float	DAS calibration source for backend calibration (POWER or DATA)
C7TAU225	CSO_TAU	float	CSO tau at 225GHz
C7TAURMS	CSO_TAU_RMS	float	CSO tau rms
C7TAUTIME	CSO_YYMMDDHHMM	char*	CSO tau time (YYMMDDHHMM)
C7SEEING	SAO_SEEING	float	Seeing at JCMT
C7SEETIME	SAO_YYMMDDHHMM	char*	SAO seeing time (YYMMDDHHMM)
C3POLARITY	C3POLARITY	char*	Frontend Polarity
C3SBMODE	C3SBMODE	char*	Sideband mode
C55NPH	DAS_NO_PHASE	int	DAS number of phases for interferometry observing
C55NCYC	DAS_NO_CYCLES	int	DAS number of phases for interferometry observing
C55NINT	DAS_NINTCYCLE	int	DAS number of phases for interferometry observing
C55NCORR	DAS_NCORRCYCLE	int	DAS number of correlation cycles
C55DASPRBIT	DAS_PROC_BITS	int	DAS data processing done
C55DASPRLOC	DAS_PROC_LOC	char*	Description of where processing is done
C55LX	RXJ.LX	double	RXJ X length of projected baseline in metres
C55LY	RXJ.LY	double	RXJ Y length of projected baseline in metres
C55LZ	RXJ.LZ	double	RXJ Z length of projected baseline in metres
C55A	RXJ.A	double	RXJ Coefficient of sin term in expression for fringe rate (metres)
C55B	RXJ.B	double	RXJ Coefficient of cos term in expression for fringe rate (metres)

C55C	RXJ.C	double	RXJ Coefficient of constant term in expression for fringe rate (metres)
C55CSOSW	RXJ.CSO_SWITCH	int	RXJ Delay setting of RXJ micro for CSO side
C55JCMTSW	RXJ.JCMT_SWITCH	int	RXJ Delay setting of RXJ micro for JCMT side
C55SECOND	RXJ.NSECS	int	RXJ Number of the tick on which integration started
C55ABSORB	RXJ.ABSORB	char*	CSO Position of absorber IN or OUT
C55TAU	RXJ.CSO_TAU	float	CSO TAU value
C55DAZ	RXJ.DAZ	float	CSO Position offset in az (arcsec)
C55DEL	RXJ.DEL	float	CSO Position offset in elevation (arcsec)
C55RA	RXJ.RA	double	CSO RA
C55DEC	RXJ.DEC	double	CSO Dec
C55EPOCH	RXJ.EPOCH	double	CSO Epoch of CSO RA and Dec
C55PAZ	RXJ.PAZ	float	CSO Pointing offset in az (arcsec)
C55PEL	RXJ.PEL	float	CSO Pointing offset in el (arcsec)
C55TRACK	RXJ.TRACK	char*	CSO Track mode of telescope Y or N
C55FMODE	RXJ.FMODE	char*	CSO Focus mode of CSO
C55FX	RXJ.FX	float	CSO X position of focus
C55FY	RXJ.FY	float	CSO Y position of focus
C55FZ	RXJ.FZ	float	CSO Z position of focus
C55VLSR	RXJ.VLSR	float	CSO LSR velocity of source (km/s)
C55VOFF	RXJ.COFF	float	CSO velocity offset (km/s)
C55VRAD	RXJ.VRAD	float	CSO radial velocity (km/s)
C55PLOCK	RXJ.PLOCK	char*	CSO Phase lock status L or U
C55RFREQ	RXJ.RFREQ	double	CSO Rest frequency of line (GHz)
C55IFFREQ	RXJ.IFFREQ	double	CSO IF frequency (GHz)
C55LOFREQ	RXJ.LOFREQ	double	CSO LO frequency (GHz)

C55FREQOFF	RXJ.FREQOFF	double	CSO frequency offset (GHz)
C55SIDEBAND	RXJ.SIDEBAND	char*	CSO Sideband U or L
C55MHN	RXJ.MHN	int	CSO Multiplier Harmonic number
C55CSOSTATUS	RXJ.CSO_STATUS	int	CSO overall status 0 = bad 1 = good
C55TELAZ	TEL.SDIS(9)	double	CENTRE_AZ from tel sdis array
C55TELEL	TEL.SDIS(10)	double	CENTRE_EL from tel sdis array

A.2 GSD Arrays

NRAO name	*JCMT name*	*Data type*	*Description*
C55FENUOBS	FE_NUOBS	double	Observing frequencies
C55FESBSIGN	FE_SB_SIGN	int	FE sideband signs
C55FENULO	FE_NULO	double	FE LO frequencies
C7VRADIAL	C7VRADIAL	double	Array for computer radial velocities
C12SCAN_VARS1	SCAN_VARS1	char*	Names of the cols. of scan table1
C12SCAN_VARS2	SCAN_VARS2	char*	Names of the cols. of scan table2
C12SCAN_TABLE_1	SCAN_TABLE1	float	Begin scan table
C12SCAN_TABLE_2	SCAN_TABLE2	float	End scan table
C14PHIST	MAP_TABLE	float	List of xy offsets for each scan
C11VD	PHASE_VARS	char*	Names of the cols. of phase table
C11PHA	PHASE_TABLE	float	Phase table: switching scheme dependent
C12CM	BES_CORR_MODE	int	Correlation function mode
C12BM	BES_BITMODE	int	Correlation bit mode
C3OVERLAP	BES_OVERLAP	float	Subband overlap
C3MIXNUM	DAS_MIXER	int	
C3BESCONN	BES_CONN	int	BE input channels connected to this section
C3BEINCON	BE_IN_CONN	int	IF output channels connected to BE input channels
C3LSPC	NO_BES_O_CH	int	Number of channels per backend section
C3BESSPEC	BES_SPECTRUM	int	Subsystem nr to which each backend section belongs.
C12CF	BES_NUOBS	double	Centre frequencies (rest frame of source) [GHz]
C12RF	BES_NUREST	double	Rest frequency [GHz]
C3BEFENULO	BES_FE_NULO	double	Copy of frontend LO frequency per backend section
C3BETOTIF	BES_TOT_IF	double	Total IF per backend section

C3BEFESB	BES_FE_SB_SIGN	int	Copy of frontend sideband sign per backend section
C12INFREQ	BE_NUIN	double	BE input frequencies [GHz]
C12FR	BES_DELTANU	float	Frequency resolution [MHz]
C12BW	BES_BANDWIDTH	float	Bandwidth [MHz]
C12RT	BES_T_REC	float	Receiver temperature
C12SST	BES_T_SYS	float	System temperature
C12TSKY	BES_T_SKY	float	Sky temperature at last calibration
C12TTEL	BES_T_TEL	float	Telescope temp. from last skydip
C12GAINS	BES_GAIN	float	Gain value (kelvins per volt or equivalent)
C12CT	BES_T_TEL	float	Calibration temperature
C12WO	BES_T_TEL	float	Water opacity
C12ETASKY	BES_ETA_SKY	float	Sky transmission from last calibration
C12ALPHA	BES_ALPHA	float	Ratio of signal sideband to image sideband sky transmission
C12GS	BES_G_S	float	Normalizes signal sideband gain
C12ETATEL	BES_ETA_TEL	float	Telescope transmission
C12TSKYIM	BES_T_SKY_IM	float	Frontend-derived Tsky, image sideband
C12ETASKYIM	BES_ETA_SKY_IM	float	Frontend-derived sky transmission
C12TSYSIM	BES_T_SYS_IM	float	Frontend-derived Tsys, image sideband
C12TASKY	BES_TA_SKY	float	Ratio of signal sideband to image sideband sky transmission
C3INTT	INTGRN_TIME	int	Scan integration time
C13DAT	DATA	float	Spectrum data
C55HOTPOWER	DAS_HOT_POWER	float	Total power measurement on hot load
C55SKYPOWER	DAS_SKY_POWER	float	Total power measurement on hot load (??)

C55SAM	SAMPLES	float	Samples to store for cross_correlation mode
C55POWER	DAS_POWER	float	Total power measurement per subband per integration

A.3 Dimensionality of GSD Arrays

NRAO name	*Dims (normal)*	*Dims (total power)*	*Dims (cont. cal)*	*Dims (cross corr.)*
C55FENUOBS	N/A			C3NFOC
C55FESBSIGN	N/A			C3NFOC
C55FENULO	N/A			C3NFOC
C7VRADIAL	C7SZVRAD	N/A	C7SZVRAD	N/A
C12SCAN_VARS1	C3NO_SCAN_VARS1			
C12SCAN_VARS2	C3NO_SCAN_VARS2			
C12SCAN_TABLE_1	C3NO_SCAN_VARS1 × C3NIS			
C12SCAN_TABLE_2	C3NO_SCAN_VARS2 × C3NIS			
C14PHIST	C3NPP × C3NMAP			
C11VD	C3NSV			
C11PHA	C3NSV × C3PPC			
C12CM	C3NRS			
C12BM	C3NRS			
C3OVERLAP	C3NRS			
C3MIXNUM	C3NRS			
C3BESCONN	C3NRS			
C3BEINCON	C3NRC			
C3LSPC	C3NRS			
C3BESSPEC	C3NRS			
C12CF	C3NRS			
C12RF	C3NRS			
C3BEFENULO	C3NRS			
C3BETOTIF	C3NRS			
C3BEFESB	C3NRS			
C12INFREQ	C3NRC			
C12FR	C3NRS			
C12BW	C3NRS			
C12RT	C3NRS			

C12SST	C3NRS		C3NRS × C3NIS	C3NRS
C12TSKY	C3NRS		C3NRS × C3NIS	C3NRS
C12TTEL	C3NRS			
C12GAINS	C3NRS			
C12CT	C3NRS			
C12WO	C3NRS			
C12ETASKY	C3NRS		C3NRS × C3NIS	C3NRS
C12ALPHA	C3NRS			
C12GS	C3NRS			
C12ETATEL	C3NRS			
C12TSKYIM	C3NRS		C3NRS × C3NIS	C3NRS
C12ETASKYIM	C3NRS		C3NRS × C3NIS	C3NRS
C12TSYSIM	C3NRS		C3NRS × C3NIS	C3NRS
C12TASKY	C3NRS		C3NRS × C3NIS	C3NRS
C3INTT	C3NIS			
C13DAT	C3NCH × C3MXP × C3NIS			
C55HOTPOWER	N/A			C3NRS × C3NOIFPBES
C55SKYPOWER	N/A			C3NRS × C3NOIFPBES
C55SAM	N/A			C3NCH × C55NPH × C3NCI
C55POWER	N/A	C3NRS C3NOIFPBES C55NPH C3PPC C3NCI	N/A	C3NRS × C3NOIFPBES × C55NPH × C3NCI