

SCUBA-2 DR Pipeline Project Office

University of British Columbia
 6224 Agricultural Road
 Vancouver, British Columbia
 CANADA
 V6T 1Z1

Tel: +1-604-822-2211

Fax: +1-604-822-5324

Email: jmolnar@ubc.ca


WWW: <http://scuba2.jach.hawaii.edu>

Document Title: Data Acquisition/Data Reduction Pipeline Interface Control Document

Document Number: SC2/SOF/IC210/01

Issue: 1.56

Date: 2007/08/01

Document Prepared By:	Tim Jenness & Andy Gibb	Signature and Date:	2005-05-24
Document Approved By:	B. D. Kelly	Signature and Date:	2005-05-24
Document Released By:	Janos Molnar	Signature and Date:	 2005-05-24



Contents

1 Introduction	4
2 Raw data	4
2.1 Naming conventions	4
2.2 Directory structure	4
2.3 File naming convention	5
2.4 Completion Notification	6
2.4.1 Online Pipeline	6
2.4.2 Quick Look	7
2.5 File Format and content	8
2.5.1 FITS headers	8
2.5.2 Data values	9
2.5.3 Readout headers	10
2.5.4 OCS config file	11
3 Real Time Data Reduction Output	11
3.1 Storage format	12
3.2 DREAM parameters	12
3.3 DREAM weights	13
4 Engineering Data	14
A Data file structure: example trace	15
B FITS Headers	20
B.1 Notes on the tables	21
B.2 Keywords for IMAGE and TSTREAM	21
B.2.1 Telescope Specific	21
B.2.2 Observation, Date and Pointing Specific	22



B.2.3 Environment Specific	23
B.2.4 Integration Time Specific	24
B.2.5 OMP and ORAC-DR Specific	24
B.2.6 SCUBA-2 (Front end) Specific	24
B.2.7 Switching and Mapping Details	25
B.2.8 Secondary Mirror Specific	26
B.2.9 JOS Parameters	26
B.2.10 Miscellaneous	26
B.2.11 Polarimeter Specific	27
B.2.12 FTS Specific	27
B.2.13 IMAGE specific	27
B.3 Keywords for the DREAM weights file	28



1 Introduction

This document describes the interface between the SCUBA-2 Data Acquisition (DA) computers and the Data Reduction (DR) Pipeline. Since the DR Pipeline is data driven (see Software Requirement **GR13**) this document mainly describes the files generated by the DA system; their location, format and content. This document assumes the observing modes specified in SC2/ANA/S100/028,¹³ the data output discussion in SC2/SOF/S200/007,¹⁴ the field distortion discussion in SC2/ANA/S200/042¹⁸ and the raw file format discussion in SC2/SOF/S200/032.¹⁶

2 Raw data

Raw data are defined as the least processed data product generated by the DA system. This does not necessarily mean that the data have not been processed (20-kHz readouts are co-added in the DA system to at least 200 Hz), simply that these are the data that appear on disk and are archived. Every observing mode (including start-up observations, darks, flat-fields and SQUID set-up) will generate data files using the format described in this section – *with no exceptions*. The only difference will be in the integration time for an individual bolometer readout: some modes (especially the set-up modes) may co-add internally in the DA system and then store at a slower rate than 200 Hz.

The DA computers are configured such that there is one computer per sub-array, each writing a separate file to a local disk.

2.1 Naming conventions

This document follows the standard sub-array naming conventions adopted in SC2/SOF/S200/042¹⁸ where each sub-array is named *s_nx* where *s* is a prefix indicating SCUBA-2, *n* is a number corresponding to the wavelength of the sub-array (4 for 450 μm and 8 for 850 μm) and *x* is the sub-array identifier for that wavelength (letters a to d). This leads to the 850- μm sub-arrays being named s8a, s8b, s8c, and s8d and the 450- μm sub-arrays being named s4a, s4b, s4c, and s4d.

Any data file, directory or data acquisition computer associated with a specific sub-array will follow this naming scheme.

2.2 Directory structure

The JAC standard specifies that instruments use a specific directory structure for raw and Pipeline-reduced data:

```
/jcmtdata/raw/INSTRUMENT/xxx/YYYYMMDD/
```

where INSTRUMENT is the name of the instrument (lower case), xxx is an (optional) indication of which sub-array is writing the data (not required for SCUBA, but *is* required for ACSIS)



and YYYYMMDD is the UT-date associated with the start of a specific observation. For development systems the root file system (/jcmtdata at the JAC) can be replaced with any location and specified using an environment variable (ORAC-DR uses ORAC_DATA_ROOT, but the JCMT has no standard for the name of this variable). For SCUBA-2 the xxx in the above definition is replaced with the sub-array name as described in 2.1. Each DA computer for SCUBA-2 writes data to disk from a single sub-array and for performance reasons this disk has to be local for each acquisition computer. The SCUBA-2 directories at the JCMT will therefore look something like:

```
/jcmtdata/raw/scuba2/s8a/20060510/  
/jcmtdata/raw/scuba2/s8b/20060510/  
/jcmtdata/raw/scuba2/s8c/20060510/  
/jcmtdata/raw/scuba2/s8d/20060510/  
/jcmtdata/raw/scuba2/s4a/20060510/  
/jcmtdata/raw/scuba2/s4b/20060510/  
/jcmtdata/raw/scuba2/s4c/20060510/  
/jcmtdata/raw/scuba2/s4d/20060510/
```

For the rest of this document, the fixed prefix /jcmtdata/raw/scuba2 will be shortened to DATAROOT.

2.3 File naming convention

The JAC naming convention for raw data files is to use xxxYYYYMMDD_NNNNN where xxx is the instrument prefix, YYYYMMDD is the UT date and NNNNN is the 5 digit observation number (zero-padded) which starts from 1 for the first observation of the night and increments each time a file is written to disk by the acquisition system. For SCUBA-2 this scheme is not scalable since the instrument will need to write data to disk at a continuous rate (opening and closing files when the files exceed a maximum size) that is not related to the observer's definition of an observation (a single configuration sent by the queue).

The SCUBA-2 solution is to create a sub-directory rather than a file for each configuration, and to write data for that configuration to files that are tagged with the UT date, observation number and finally a number that increments each time a new data file is written to disk (called the sub-scan number).

The observation number and sub-scan numbers must be synchronized between all of the acquisition computers such that observation N , sub-scan M , written by one computer is part of the same observation N , sub-scan M , saved by another. Each observation will result in a directory called NNNNN, and this directory will contain files named s_{nx}YYYYMMDD_NNNNN_MMMM.sdf, such as s8a20060501_00032_0012.sdf. For example, two observations using the long-wave filter, each consisting of 2 sub-scans will result in the following files in DATAROOT:

```
s8a/20060510/00001/s8a20060510_00001_0001.sdf  
s8a/20060510/00001/s8a20060510_00001_0002.sdf  
s8a/20060510/00002/s8a20060510_00002_0001.sdf  
s8a/20060510/00002/s8a20060510_00002_0002.sdf  
s8b/20060510/00001/s8b20060510_00001_0001.sdf  
s8b/20060510/00001/s8b20060510_00001_0002.sdf  
s8b/20060510/00002/s8b20060510_00002_0001.sdf  
s8b/20060510/00002/s8b20060510_00002_0002.sdf  
s8c/20060510/00001/s8c20060510_00001_0001.sdf
```



```
s8c/20060510/00001/s8c20060510_00001_0002.sdf
s8c/20060510/00002/s8c20060510_00002_0001.sdf
s8c/20060510/00002/s8c20060510_00002_0002.sdf
s8d/20060510/00001/s8d20060510_00001_0001.sdf
s8d/20060510/00001/s8d20060510_00001_0002.sdf
s8d/20060510/00002/s8d20060510_00002_0001.sdf
s8d/20060510/00002/s8d20060510_00002_0002.sdf
```

2.4 Completion Notification

The DA system must notify the DR system when files are ready for processing. The Quick Look (QL) and online pipeline notification systems use different notification methods due to differing requirements.

2.4.1 Online Pipeline

The online pipeline processes observations and coadds them into observation groups. Each DA system notifies the DR that the observation is complete by writing a file to DATAROOT in a sub-directory named `ok/YYYYMMDD`. From ORAC-DR's viewpoint this directory is defined as `ORAC_DATA_IN`. These *flag* files have the same naming convention as the observations except that they are text files and use a `.ok` file suffix and don't require a sub-scan number: `snxYYYYMMDD_NNNNNN.ok`.

To help the pipeline locate the data files these flag files contain the names of all the sub-scan files written by the DA computer during the associated observation, including a full path to these files relative to `ORAC_DATA_IN`. The pipeline will wait for all four flag files and open them to determine which files are to be processed for that observation. Using the example from §2.3 the flag file `s8a20060501_00001.ok` would contain the following:

```
../../../../s8a/20060510/00001/s8a20060510_00001_0001.sdf
../../../../s8a/20060510/00001/s8a20060510_00001_0002.sdf
```

It is vital that these flag files are written to the file system as an atomic operation to prevent the pipeline from doing a partial read on the file. (Usually this is achieved using a temporary file and the `rename` system call.)

Since it is possible that a single observation may last for a significant amount of time it is not desirable for the DR to have to wait until the end of an observation before it can begin processing that observation. A 2 hour observation that takes 2 hours to reduce is useless if it takes 2 hours before the pipeline begins processing. For this reason as new data are stored during an observation the location of these files can be appended to the flag file as opposed to writing a complete flag file at the end of the observation. The only requirement on the DA is that this append operation is atomic so that the pipeline does not do partial reads. This means that the pipeline will never be behind by more than the time taken between file dumps (but note that there is no explicit requirement on the DA to write files at a specific rate for each observing mode; an upper limit of 5 minutes would be desirable).

Req. PR5



2.4.2 Quick Look

The QL system is only interested in processing the most recent sub-scan and can, in principle, miss sub-scans if it is having trouble keeping up with observing. The DA computers provide an alternative transient notification interface for the QL pipelines using DRAMA¹ parameters.

Req. DR8

Each QL pipeline will monitor the 4 DA DRAMA tasks. The names of these tasks are obtained from a text file on disk which contains the translation between DRAMA task names and subarray names. If the DRAMA networking task is running (which has to be the case in the live system) DRAMA will automatically locate these tasks on the network and so it is not necessary to include the hostname in the task designation.

When the DA computer has new data available for QL display, it updates a DRAMA structure parameter named QL. This parameter contains the following information for DREAM/STARE processed images (see also §3):

```
QL
SEQUENCE      <_INTEGER>      Monotonically increasing number
TIMESTAMP     <_DOUBLE>        Epoch seconds
IMAGE         <_STRUCTURE>
  FITS         <_CHAR*80>    Array of FITS keywords
  DATA_ARRAY  <_REAL>       2D-image
```

where the `TIMESTAMP` is there as a convenient easy to access parameter indicating when the data were made available (fractional seconds since reference epoch); `SEQUENCE` is a monotonically increasing number synchronized across DA computer that should increment each time a new data array is published (this allows the pipeline to report missed frames); `IMAGE` is a structure containing the DREAM/STARE processed image where the array of FITS headers should contain all the same header information as stored in the processed images written to the raw data files and additionally include a **BUNIT** header and a FITS Native AST representation of the distorted astrometric coordinate frames^{18,24} (different to the processed images which will use the standard NDF approach for units and astrometry).

For spinning polarimetry observations the `DATA_ARRAY` in the `IMAGE` parameter will be 3-D rather than 2-D and will include Q and U information.¹⁰

For scan maps and FTS observations, it is impractical to register a parameter that potentially can be greater than a Gigabyte and so an alternative parameter scheme is used. Since scan map scans should be processed when each data file is closed (as opposed to DREAM/STARE where multiple QL images can be created per raw data file) the parameter contains solely the `TIMESTAMP`, `SEQUENCE` number and the path of the sub-scan file relative to the flag directory (ORAC_DATA_IN from ORAC-DR's viewpoint) (i.e. the same path as written to the flag file):

```
QL
SEQUENCE      <_INTEGER>
TIMESTAMP     <_DOUBLE>
FILENAME      <_CHAR>      "../../../s8a/20060510/20060510_00001/s8a20060510_00001_0001.sdf"
```



2.5 File Format and content

The file format for raw data used by all UKIRT instruments, ACSIS and SCUBA is the Starlink N-Dimensional Data Format²³ (which is based on the Hierarchical Data System, HDS)²⁵ and SCUBA-2 will use the same format.

The raw data files must contain the following information:

1. General FITS headers describing the observation, but that are “slowly” varying (usually items are fixed for the duration of the observation, but some need to be stored once for the start of the observation and once for the end).
2. The actual data read from the bolometers.
3. Fast varying meta-data that changes for each readout frame.
4. A copy of the OCS config XML file.

These items will be discussed in turn below.

History recording should be enabled through the use of the HISTORY NDF extension.

2.5.1 FITS headers

Both raw data files and DREAM/STARE processed images generated by the DA system will share these headers. The headers are organized in related groups for ease of locating information and to make the headers more readable (e.g. environmental data are grouped together) and are described in the appendix, §B. Headers directly related to pipeline handling are discussed in the following list:

OBSNUM

The observation number. Matches the number in the filename.

SCANNUM

The sub-scan number. Matches the number in the filename.

OBSEND

A boolean, indicating whether this frame is the last frame in the current observation. This is a convenience parameter, solely for the on-line Pipeline to provide the option for Recipes to defer some steps until a full set of data are available (**OBSEND** is not needed by the Quick-Look display⁷ since the observation image is built up as data appear; the off-line version of the Pipeline can read all the FITS headers before beginning processing).

DRGROUP

A string indicating which observations can be combined. If defined, all observations that share an identical **DRGROUP** value will be combined. This value can be specified during observation preparation via the Observing Tool and/or the translator. If the value is not set the Pipeline will attempt to calculate its own groups.



DRRECIPE

The name of the Data Reduction Recipe the Pipeline should use to process the frame. If this is not defined the Pipeline will attempt to guess.

Req. GR5

The headers in the raw data segment do not duplicate the information found in the SCUBA-2 RTS state structure (§2.5.3) but do in some cases provide averaged values of the fast-varying data.

2.5.2 Data values

Each SCUBA-2 subarray is a 40 rows by 32 columns array of bolometers.⁵ Data are stored compressed in a 3-dimensional short (16-bit) integer array with the first axis matching the number of rows, the second the number of columns and the third corresponding to time. A 2-D slice for fixed time will therefore show the pixels as they are present on the array (without correction for field distortion). The units information will be stored in a UNITS NDF extension.³

Actual instrumental raw values are actually 32-bit integers, and so a compression algorithm is used to allow the 16-bit storage to occur. The compression scheme¹⁵ consists of:

1. Identifying a suitable `STACKZERO` single 2-D frame in the time series (the first frame is a suitable choice);
2. Subtracting the `STACKZERO` frame off all the frames in the time series, leaving a set of frames with much reduced dynamic range;
3. The reduced frames are each analyzed to find a suitable frame-specific `BZERO`;
4. Each reduced frame has its `BZERO` subtracted and all its values are put into 16-bit integers.

If a value in a frame will not fit into 16-bits, then the location of the offending value (counting sequentially through the vectorized array starting at 0) is stored separately along with its 32-bit value (`STACKZERO` subtracted, but not `BZERO`), and `VAL_BADUW` is written into the corresponding location in the compressed frame. The `BZERO` information is written to the `.MORE.SCUBA2` structure as a 1-D NDF.

If the data acquisition system determines that there is a significant number of outliers, it can revert to using an uncompressed 32-bit floating point data array. For this reason it is recommended that the standard data access library¹⁶ is used when manipulating these data files.

Therefore, for each pixel in each frame, the uncompressed value is retrieved by taking the compressed data value and adding on the `STACKZERO` value for that pixel and the `BZERO` for that frame. For incompressible pixels, the full uncompressed value is given by adding the `STACKZERO` for that pixel to the incompressible pixel value. The list of incompressible pixels is stored in separate structure (`INCOMP`) as a $(2, n_{\text{bad}})$ array, where n_{bad} is the number of incompressible pixels (flagged as bad). Each pair of numbers gives the `DATA_ARRAY` offset in the frame (counting from zero) and the 32-bit value left after the `STACKZERO` subtraction.



2.5.3 Readout headers

For every time-slice (frame) present in the data array, there will be corresponding header data that provides information on this frame.

The content of these structures can vary depending on observing mode, but for modes that include the Real Time Sequencer^{6,19} all the information provided by components participating in the sequence should be included in the `.MORE.JCMTSTATE` structure.

The main exception to this is the Water Vapour Monitor which will not participate in the RTS sequence. The DA system should monitor the current temperatures from the WVM (new values are generated every 1.2 seconds) and include them in the time-varying data structure, since some observing modes will result in frames that are written to disk much slower than one per second (SCAN map data are written at the end of every scan).²¹ There is no requirement for other environmental data to be present at this time resolution.

For efficiency, the data obtained from the RTS sequences, will be stored in individual data arrays per time slice rather than RTS-based structures.

The data sources for the `JCMTSTATE` structure will therefore be:

RTS

The RTS provides the sequence number, the duration of the sequence step (the *exposure time*), the TAI time of the end of the sequence step (as a modified Julian day) and the tasks involved in the sequence (along with any that did not participate during the step). Only the end time is stored since that is the accurately timed part of the sequence (the time when “data valid” falls).

TCS

The TCS RTS state structure²⁰ includes the current airmass (of the tracking centre), the position in the observing area, whether the telescope is on a `SCIENCE` target or not (for example, during a SCAN map the telescope could report that it is on source `SKY` when it is turning around at the end of a scan), and the position of the telescope (as a demand position, actual position, and base position in both the current tracking coordinate frame and in `AZEL`). These are all determined for the `TAI_END` value stored in the RTS extension.

SMU

The SMU RTS state structure includes the focus position of the SMU (in millimetres), and the jiggle offset (arcsec in tangent plane offset) in both `AZEL` coordinates and the current tracking coordinates.²⁰ The `SMU_JIG_INDEX` entry will be incremented by 1 at the start of each repetition of the `DREAM` pattern.

FTS

The position of the FTS mirror is all that is required from the FTS. Units are in millimetres.

POL

The SCUBA-2 polarimeter will need to provide the waveplate rotation angle. This angle will be in focal plane coordinates, since the polarimeter will not know the TCS position. It will be a position angle, East of North, in radians.



Additionally, the DA system may want to write information into these structures, although this has not yet been specified.

Some set-up observations can be done without the participation of the RTS (e.g. array setup¹³), but for these observations the sample times (as determined by the DA system) should still be written into the same RTS structure for consistency.

2.5.4 OCS config file

The information in the OCS configuration XML file will be stored in the `.MORE.JCMTOCS.CONFIG` NDF extension. For convenient printing to a terminal screen, the file will be stored as an array of strings, each 72 characters in length.

3 Real Time Data Reduction Output

In addition to writing raw data, the DA computers will generate processed images in STARE, DREAM and SKYDIP observing modes and polynomial fit coefficients per bolometer (for removing $1/f$ drift). This is performed by the DA system in order to help the Quick Look⁹ system to provide images as quickly as possible (since the DA system will be calculating this image on one computer per sub-array it can do it faster than the single-CPU QL would be able to).

Processed science data will be 2-D images tagged with full world-coordinate systems (including field distortion and sky rotation). All files will have identical FITS headers to the raw data (see §B for details of additional keywords), and astrometry will be defined using AST.^{24,18} These data will *not* have RTS state structures, although for FTS⁴ and polarimeter observations^{10,17} using step-and-integrate mode the waveplate and/or mirror position will need to be present in the FITS header.

Polynomial coefficients will be written out as a 3-D array similar to the raw data structure, except that the 3rd axis will be the order rather than time.

Polarimeter observing modes using fast-spinning mode will result in DA-generated cubes similar to STARE observations in the first plane (including AST astrometry and full FITS headers) except that there will be Q and U planes in addition to the I plane.¹⁰

For modes where the FTS mirror is used in fast-scanning mode the DA system will not be able to co-add STARE images. The pipeline will therefore calculate the images from the raw time series data.

The DA system will calculate DREAM/STARE images faster than files are written to disk since images have to be calculated before there is significant sky rotation (no more than a few seconds per image) but, for efficiency reasons, data files will be written more slowly (every minute or so).



3.1 Storage format

These DA-generated reduced data will be stored as NDFs in the `SCU2RED` NDF extension of the raw data file.

Reduced DARK, DREAM and STARE images will be stored in the `.MORE.SCU2RED.Ixx` where `xx` is a monotonically increasing number starting at 1 (not zero-padded). See Appendix A for an example file trace containing 2 DREAM/STARE reconstructions.

The polynomial coefficients will be stored as an NDF in `.MORE.SCU2RED.SCANFIT`.

Additionally, the number of pixels in the processed images will not be required to have the same number of pixels as there are bolometers, since there may be motion of the secondary mirror during a jiggle pattern.

The format of DA-generated reduced data from other observing modes (e.g. flat-fielding or array setup) have not yet been specified.

3.2 DREAM parameters

All DREAM information is stored in the `.MORE.DREAM` extension in the raw data. To reconstruct an image from the data, the vertices of the DREAM jiggle pattern and path of the SMU across the subarray must be known. The former is an extension named `PATTERN` and stores an `_DOUBLE` array of the `X,Y` grid coordinates for each vertex of the jiggle pattern. These grid offsets are converted to arcsec by the FITS keyword `JIG_SCAL`.

The path of the SMU is stored in an extension named `PATH` and contains an array of `_DOUBLE` representing the `X,Y` offsets in arcsec from the centre of any given bolometer for each time step during one DREAM jiggle pattern.

This example trace shows the array of jiggle vertices for an 8-point DREAM pattern. The SMU path contains 8 times the number of samples recorded between each vertex of the pattern (in this case, also 8).

```
DREAM          <DREAM_PAR>      {structure}
  PATTERN      <NDF>            {structure}
    DATA_ARRAY <ARRAY>         {structure}
      DATA(8,2) <_DOUBLE>      0.0,-1.0,1.0,-1.0,
      ... 0.0,-1.0
      ORIGIN(2)  <_INTEGER>     1,1

  PATH         <NDF>            {structure}
    DATA_ARRAY <ARRAY>         {structure}
      DATA(64,2) <_DOUBLE>     0,-0.47803653582912,
      ... 11.603926928342
      ORIGIN(2)  <_INTEGER>     1,1
```

The weights and inverse matrix for reconstructing images from DREAM data²² are stored in a separate file. This is done for reasons of space: this file can be very large and there is no point duplicating the information in every data file, especially if the information does not



change over the course of a night. The format of this file is described in the following section, §3.3.

3.3 DREAM weights

To reconstruct a DREAM image, the DA needs access to the file containing the grid weights and inverse matrix. The name of this file is given in the FITS header `DRMWGHTS`. The naming convention for this file is `dreamweights_snx.sdf`, where `snx` is the subarray name (e.g. `s8a`). The file is a Starlink NDF and all DREAM information is stored in the `.MORE.DREAM` extension. The main data array within this file is a 1×1 NDF of type `_INTEGER` but it is not used to store any pertinent data. The grid weights and inverse matrix are stored as NDFs of type `_DOUBLE` in two extensions `.MORE.DREAM.GRIDWTS` and `.MORE.DREAM.INVMATX` respectively. Given the length of time necessary to create this file (for each subarray), the pipeline (and the data acquisition system) assumes that it exists already. The pipeline will not be responsible for creating this file, though functionality exists within SMURF to do so if desired at a later date.

In addition to the weights themselves, the weights file will also contain the path of the SMU over one cycle and the parameters describing the DREAM pattern used (see § 3.2 above). These will be stored as the NDF extensions `.MORE.DREAM.PATH` and `.MORE.DREAM.PATTERN` respectively. The type is `_DOUBLE` for each.

The DREAM weights file also contains relevant information used to calculate the weights, stored in a FITS extension. This includes the current subarray, the jiggle pattern scaling factor (`JIG.SCAL`), the name and speed of the pattern used, upper and lower X and Y bounds for the reconstruction grid, X_{\min} , X_{\max} , Y_{\min} , Y_{\max} . Also necessary is the grid spacing (or scale factor) in arcsec. See §B.3 in the Appendix.

An example trace from a DREAM weights file for a DREAM pattern with 8 jiggle positions, regridding onto a 9×9 grid is shown below.

```
DREAMWEIGHTS_S8 <NDF>

  DATA_ARRAY <ARRAY> {structure}
    DATA (1,1) <_INTEGER> 0
    ORIGIN (2) <_INTEGER> 0,1

  MORE <EXT> {structure}
    DREAM <DREAM_WEIGHTS> {structure}
      GRIDWTS <NDF> {structure}
        DATA_ARRAY <ARRAY> {structure}
          DATA (81,64) <_DOUBLE> 0,-3.3391327230583D-37,
          ... 2.43025780478164D-6
          ORIGIN (2) <_INTEGER> 1,1

        INVMATX <NDF> {structure}
          DATA_ARRAY <ARRAY> {structure}
            DATA (5121600) <_DOUBLE> 0.036809470042934,
            ... 9120239513.5379
            ORIGIN (1) <_INTEGER> 1

      FITS (8) <_CHAR*80> 'GRIDXMIN= -4 / Mi...'
        ... 'GRIDSIZE= 6.280 / ...'
```



It is conceivable that this file might become part of the ORAC-DR calibration object, in which case access to a date stamp is desirable. It is likely that the HISTORY extension can be used for this purpose since it lists the file creation date.

4 Engineering Data

The array set-up and flat-fielding modes generate data that need to be reduced by the Pipeline⁸ and then subsequently used by the DA system for array configuration and for generating its real-time reduced images (see §3). It is not yet known how often these observations need to be performed or how to reduce these data; the current assumption is that they are not performed every day.

Flat-field and array set-up observations will be performed in an engineering mode with the data appearing in engineering data directories. The engineering directory structure is similar to the science data directory structure, with the addition of “eng” to the path:

```
/jcmtdata/raw/eng/scuba2/snx/YYYYMMDD/
```

The flat-field information is written by the DA system to each raw data file in a FLATCAL extension within the NDF file (see the example trace in § A) as a 3-D data array.

References

- [1] J. A. Bailey, T. Farrell, and K. Shortridge. DRAMA: an environment for distributed instrumentation software. In *Proc. SPIE Vol. 2479, p. 62-68, Telescope Control Systems, Patrick T. Wallace; Ed.*, pages 62–68, June 1995.
- [2] Malcolm J. Currie. HDSTRACE – listing HDS data files. Starlink User Note 102, Starlink Project, CCLRC, 1998.
- [3] Malcolm J. Currie and David S. Berry. KAPPA – Kernel Application Package. Starlink User Note 95, Starlink Project, CCLRC, 2002.
- [4] Brad Gom. SCUBA-2 FTS Operational Concept Definition. SCUBA-2 Project SC2/FTS/SYS/004.
- [5] Wayne S. Holland. Functional and performance requirements for SCUBA-2. SCUBA-2 Project SC2/SRE/SC200/002.
- [6] G. J. Hovey, P. E. Dewdney, R. O. Redman, A. G. Willis, B. D. Kelly, X. Gao, J. F. Lightfoot, W. R. F. Dent, N. P. Rees, P. Friberg, and R. M. Prestage. Co-ordinating and Synchronizing Instruments for a Real-time Distributed Reduction System. In *ASP Conf. Ser. 281: Astronomical Data Analysis Software and Systems XI*, pages 355–+, 2002.
- [7] Tim Jenness. SCUBA-2 Data Display System. SCUBA-2 Project SC2/SOF/S210/002, 2003.
- [8] Tim Jenness. SCUBA-2 Data Reduction Recipes and Primitives. SCUBA-2 Project SC2/SOF/S210/004, 2003.



- [9] Tim Jenness. SCUBA-2 Pipeline Architecture. SCUBA-2 Project SC2/SOF/S210/001, 2003.
- [10] Tim Jenness. SCUBA-2 Polarimetry Data Processing. SCUBA-2 Project SC2/SOF/S210/005, 2004.
- [11] Tim Jenness. Unique Observation Identifiers for JCMT and UKIRT data. JCMT JSA/ANA/001, 2006.
- [12] Tim Jenness, Jamie Leech, Shaun de Witt, and Frossie Economou. ACSIS File Format Interface Control Document. JCMT OCS/ICD/022, 2006.
- [13] B. D. Kelly. SCUBA-2 modes and data processing. SCUBA-2 Project SC2/ANA/S100/028, 2001.
- [14] B. D. Kelly. SCUBA-2 Data Output. SCUBA-2 Project SC2/SOF/S200/007, 2003.
- [15] B. D. Kelly. SCUBA-2 Data Compression. SCUBA-2 Project SC2/SOF/S200/029, 2004.
- [16] B. D. Kelly. SCUBA-2 Data File Format. SCUBA-2 Project SC2/SOF/S200/032, 2004.
- [17] B. D. Kelly. SCUBA-2 Polarimetry. SCUBA-2 Project SC2/SOF/S200/027, 2004.
- [18] B. D. Kelly. SCUBA-2 Pixel Naming and Coordinate Transformations. SCUBA-2 Project SC2/SOF/S200/042, 2005.
- [19] N. Rees, F. Economou, T. Jenness, R. Kackley, C. Walther, B. Dent, M. Folger, X. Gao, D. Kelly, J. Lightfoot, I. Pain, G. Hovey, T. Willis, and R. Redman. The JCMT Observatory Control System. In *ASP Conf. Ser. 281: Astronomical Data Analysis Software and Systems XI*, pages 500–+, 2002.
- [20] Nick Rees. JAC Instrumentation Task (JIT) Library. JCMT OCS/ICD/009, 2002.
- [21] Matt Rippa. The JCMT Environment Task. JCMT OCS/ICD/014, 2002.
- [22] H. W. van Someren Greve. DREAM Algorithms for SCUBA-2. SCUBA-2 Project SC2/ANA/S100/046, 2002.
- [23] Rodney F. Warren-Smith. NDF – Routines for accessing the extensible N-Dimensional Data Format. Starlink User Note 33, Starlink Project, CCLRC, 1995.
- [24] Rodney F. Warren-Smith and David S. Berry. AST – a library for handling world coordinate systems in astronomy. Starlink User Note 211, Starlink Project, CCLRC, 2002.
- [25] Rodney F. Warren-Smith and M. D. Lawden. HDS – Hierarchical Data System. Starlink User Note 92, Starlink Project, CCLRC, 1995.

A Data file structure: example trace

The following is an example data structure generated using HDSTRACE² from a test NDF with a simulated DREAM observation. Note there are two reconstructed images (I1 and I2 components) each of which has its own associated FITS headers and WCS information (see § 3 for further details).




```

DATASET      <_CHAR*91>      '/aspersa/scuba2/jcmt/scu...'
TEXT (4)     <_CHAR*72>      'Parameters: OBSFILE=@^s...'
                                           'Software: <unknown>'

STACKZERO    <NDF>           {structure}
  DATA_ARRAY <ARRAY>        {structure}
    DATA (40,32) <_INTEGER> 5702477,5899862,5756458,
    ... 6536392,5853800,7153128
    ORIGIN (2)  <_INTEGER> 1,1

  HISTORY     <HISTORY>      {structure}
    CREATED   <_CHAR*24>      '2007-MAR-29 09:43:00.092'
    CURRENT_RECORD <_INTEGER> 1
    RECORDS (10) <HIST_REC>   {array of structures}

  Contents of RECORDS (1)
    DATE      <_CHAR*24>      '2007-MAR-29 09:43:00.122'
    COMMAND   <_CHAR*30>      'SC2SIM           (SMURF V...'
    USER      <_CHAR*3>       'agg'
    HOST      <_CHAR*24>      'aspersa.priv.phas.ubc.ca'
    DATASET   <_CHAR*95>      '/aspersa/scuba2/jcmt/scu...'
    TEXT (4)  <_CHAR*72>      'Parameters: OBSFILE=@^s...'
    'Software: <unknown>'

FLATCAL      <NDF>           {structure}
  DATA_ARRAY <ARRAY>        {structure}
    DATA (40,32,150) <_DOUBLE> 7583252,8931506,8421496,
    ... 8805179,7492823,9694916
    ORIGIN (3) <_INTEGER> 1,1,1

  HISTORY     <HISTORY>      {structure}
    CREATED   <_CHAR*24>      '2007-MAR-29 09:43:00.092'
    CURRENT_RECORD <_INTEGER> 1
    RECORDS (10) <HIST_REC>   {array of structures}

  Contents of RECORDS (1)
    DATE      <_CHAR*24>      '2007-MAR-29 09:43:00.123'
    COMMAND   <_CHAR*30>      'SC2SIM           (SMURF V...'
    USER      <_CHAR*3>       'agg'
    HOST      <_CHAR*24>      'aspersa.priv.phas.ubc.ca'
    DATASET   <_CHAR*93>      '/aspersa/scuba2/jcmt/scu...'
    TEXT (4)  <_CHAR*72>      'Parameters: OBSFILE=@^s...'
    'Software: <unknown>'

MORE         <EXT>           {structure}
  FLATDATA    <SCUBA2_FD_PAR> {structure}
    FLATNAME   <_CHAR*16>     'TABLE'
    FLATPAR (150) <_DOUBLE> 24.2,24.21,24.22,24.23,
    ... 25.66,25.67,25.68,25.69

INCOMPS (256) <SCUBA2_RTS_ARR> {array of structures}

Contents of INCOMPS (1)
{structure is empty}
:
:
Contents of INCOMPS (43)
  INCOMP      <NDF>           {structure}
  DATA_ARRAY <ARRAY>        {structure}

```




```

ZERO                                <NDF>                                {structure}
  DATA_ARRAY                        <ARRAY>                                {structure}
    DATA (40, 32)                    <_DOUBLE>                            -76955.586406867,
    ... 93749.925370507
    ORIGIN (2)                        <_INTEGER>                            1,1

FITS (11)                            <_CHAR*80>                            'DATE-OBS= '2007-03-29T09:4...'
    ... 'SUBARRAY= '8a          '      ...'

UNITS                                <_CHAR*2>                                'pW'
WCS                                  <WCS>                                    {structure}
  DATA (368)                          <_CHAR*32>                            ' Begin FrameSet', ' Nframe ...'
    ... ' End Cm...', ' End FrameSet'

I2                                    <NDF>                                {structure}
  DATA_ARRAY                          <ARRAY>                                {structure}
    DATA (44, 36)                      <_DOUBLE>                            8962193.7008181,
    ... 8962193.3209695
    ORIGIN (2)                          <_INTEGER>                            1,1

UNITS                                <_CHAR*2>                                'pW'
WCS                                  <WCS>                                    {structure}
  DATA (370)                          <_CHAR*32>                            ' Begin FrameSet', ' Nframe ...'
    ... ' End Cm...', ' End FrameSet'

MORE                                  <EXT>                                    {structure}
  DRMBOLZERO                          <SCUBA2_ZER_ARR>                        {structure}
  ZERO                                <NDF>                                {structure}
    DATA_ARRAY                          <ARRAY>                                {structure}
      DATA (40, 32)                      <_DOUBLE>                            -76955.531492003,
      ... 93749.65443968
      ORIGIN (2)                          <_INTEGER>                            1,1

FITS (11)                            <_CHAR*80>                            'DATE-OBS= '2007-03-29T09:4...'
    ... 'SUBARRAY= '8a          '      ...'

WCS                                  <WCS>                                    {structure}
  DATA (539)                          <_CHAR*32>                            ' Begin FrameSet', ' Nframe = 4', ' Cu...'
    ... ' E...', ' End CmpMap', ' End FrameSet'

```

End of Trace.

B FITS Headers

This appendix defines and lists the FITS header keywords relevant to SCUBA-2 which will be stored in the data files written to disk by the data acquisition (DA) computers. Observing modes are specified in SC2/ANA/S100/028,¹³ with updates to the polarimetry modes as specified in SC2/SOF/S210/005.¹⁰ Raw data from the DA system will be stored in files using the convention outlined earlier in this document.

There are two different types of data format written by the DA system. There is the 200 Hz time stream (labelled TSTREAM), and the reconstructed images produced in DREAM/Stare mode (IMAGE). Most of the FITS headers will be written for the TSTREAM data. Those that are specific to IMAGE data are noted.



Since a single observation may span multiple SEQUENCES and may be written across multiple files, some of the headers will change within a single observation. These are denoted in **bold**.

B.1 Notes on the tables

The keywords are listed in the following sections. For consistency, names are kept the same as defined for ACSIS¹² where possible.

Positional information for the time stream data (200 Hz) is derived from the telescope position and observation time as stored in the TCS extension. Only the map centre is made available in the FITS headers (and not using the FITS WCS standard in order to avoid confusion) when the data are archived. WCS for processed images is defined separately and is not part of the FITS header definition.

Start and end times referred to in these headers differ depending on context. In the primary TSTREAM header “start” and “end” refer to the time associated with the first and last sample written to the data file. For image reconstructions the “start” and “end” times refer to the samples associated with the specific reconstruction itself.

Finally, note that the NDF standard provides its own scheme for units handling so a **BUNIT** header is not defined.

POL and FTS specific keywords will only be written for observations involving those sub-instruments.

B.2 Keywords for IMAGE and TSTREAM

B.2.1 Telescope Specific

Keyword	Data Type	Default Value	Comment
COMMENT			– Telescope specific –
TELESCOP	STRING	'JCMT'	Name of telescope
ORIGIN	STRING	'Joint Astronomy Centre, Hilo'	Origin of file
COMMENT			x, y, z triplet for JCMT relative to centre of the earth
OBSGEO-X	FLOAT	–5.46454504E+06	[m]
OBSGEO-Y	FLOAT	–2.49298633E+06	[m]
OBSGEO-Z	FLOAT	2.15063534E+06	[m]
ALT-OBS	FLOAT	4111	[m] Height of observatory above sea level
LAT-OBS	FLOAT	19.8258323669	[deg] Latitude of observatory
LONG-OBS	FLOAT	204.520278931	[deg] East longitude of observatory
ETAL	FLOAT		Telescope efficiency



B.2.2 Observation, Date and Pointing Specific

Keyword	Data Type	Default Value	Comment
COMMENT	STRING		– Observation, date and telescope position parameters –
OBSID	STRING		Unique observation ID ¹
OBSIDSS	STRING		Unique observation/wavelength ID ²
OBJECT	STRING		Object Name
STANDARD	LOGICAL	F	True if source is a calibrator
OBSNUM	INTEGER		Observation Number
NSUBSCAN	INTEGER		Sub-scan number
OBSEND	BOOLEAN	F	True if frame is last in current observation
UTDATE	STRING		UT date as a string in yyyyymmdd format
DATE-OBS	STRING		Date and time (UTC) of start of sub-scan ²
DATE-END	STRING		Date and time (UTC) of end of sub-scan ²
DUT1	FLOAT		[days] UT1-UTC correction
INSTAP	STRING		TCS instrument aperture (if any)
INSTAP_X	FLOAT		[arcsec] Aperture X offset relative to instrument centre
INSTAP_Y	FLOAT		[arcsec] Aperture Y offset relative to instrument centre
AMSTART	FLOAT		Airmass at start of sub-scan
AMEND	FLOAT		Airmass at end of sub-scan
AZSTART	FLOAT		[deg] Azimuth at sub-scan start
AZEND	FLOAT		[deg] Azimuth at sub-scan end
ELSTART	FLOAT		[deg] Elevation at sub-scan start
ELEND	FLOAT		[deg] Elevation at sub-scan end
HSTSTART	STRING		HST at start of sub-scan ³
HSTEND	STRING		HST at end of sub-scan ³
LSTSTART	STRING		Local sidereal time at start of sub-scan ⁴
LSTEND	STRING		Local sidereal time at end of sub-scan

¹OBSID is formed from the instrument name, OBSNUM and DATE-OBS (see document JSA/ANA/001¹¹)

²OBSIDSS is created from OBSID and the WAVELEN keywords (i.e. OBSID_WAVELEN)

³DATE-OBS, DATE-END, HSTSTART and HSTEND in FITS standard format; i.e. YYYY-MM-DDTHH:MM:SS

⁴LSTSTART and LSTEND in format HH:MM:SS.xxxx, where xxxx is fraction of second



B.2.3 Environment Specific

Keyword	Data Type	Default Value	Comment
COMMENT			– Environment specific –
ATSTART	FLOAT		[deg C] Air temp at start of sub-scan
ATEND	FLOAT		[deg C] Air temp at end of sub-scan
HUMSTART	FLOAT		Rel humidity - Start obs
HUMEND	FLOAT		Rel humidity - End obs
BPSTART	FLOAT		[mbar] Pressure inside dome - Start obs
BPEND	FLOAT		[mbar] Pressure inside dome - End obs
WNDSPDST	FLOAT		[km/h] Wind speed - Start obs
WNDSPDEN	FLOAT		[km/h] Wind speed - End obs
WNDDIRST	FLOAT		[deg] Wind direction, azimuth - Start obs
WNDDIREN	FLOAT		[deg] Wind direction, azimuth - End obs
TAU225ST	FLOAT		Most recent tau at 225 GHz - Start obs
TAU225EN	FLOAT		Most recent tau at 225 GHz - End obs
TAUOBSST	FLOAT		Tau at observing wavelength, scaled from TAU225ST
TAUOBSEN	FLOAT		Tau at observing wavelength, scaled from TAU225EN
TAUDATST	STRING		UTC Date/time of TAU225ST value
TAUDATEN	STRING		UTC Date/tim of TAU225EN value
TAUSRC	STRING		TAU225 Source
WVMTAUST	FLOAT		Most recent 225-GHz tau derived from JCMT WVM - Start obs
WVMTAUEN	FLOAT		Most recent 225-GHz tau derived from JCMT WVM - End obs
WVMDATST	STRING		UTC Date/time of WVMTAUST value
WVMDATEN	STRING		UTC Date/time of WVMTAUEN value
SEEINGST	FLOAT		[arcsec] SAO atmospheric seeing - Start obs
SEEINGEN	FLOAT		[arcsec] SAO atmospheric seeing - End obs
SEEDATST	STRING		UTC Date/time of SEEINGST value
SEEDATEN	STRING		UTC Date/time of SEEINGEN value
FRLEGTST	FLOAT		[deg C] Mean front Leg Temperature - Start obs
FRLEGTEN	FLOAT		[deg C] Mean front Leg Temperature - End obs
BKLEGTST	FLOAT		[deg C] Mean back Leg Temperature - Start obs
BKLEGTEN	FLOAT		[deg C] Mean back Leg Temperature - End obs

All of these headers may change during the course of an observation. UTC date/time stamps will be in FITS standard format (YYYY-MM-DDTHH:MM:SS).



B.2.4 Integration Time Specific

Keyword	Data Type	Default Value	Comment
COMMENT			– Integration time related –
INT_TIME	FLOAT		[s] Time spent integrating on source
EXP_TIME	FLOAT		[s] Mean integration time per output pixel (IMAGE)

Note the EXP_TIME header only exists for 2-D images.

B.2.5 OMP and ORAC-DR Specific

Keyword	Data Type	Default Value	Comment
COMMENT			– OMP and ORAC-DR specific –
PROJECT	STRING		The proposal ID for the PROJECT
RECIPE	STRING		The ORAC-DR recipe
DRGROUP	STRING		Name of group to combine current observation with
MSBID	STRING		ID of min schedulable block
MSBTID	STRING		Translation ID of this MSB
SURVEY	STRING		Survey Name

The values of these items will come from the translator.

B.2.6 SCUBA-2 (Front end) Specific

Keyword	Data Type	Default Value	Comment
COMMENT			– SCUBA-2 specific –
INSTRUME	STRING	'SCUBA-2'	Instrument name – SCUBA-2
SUBARRAY	STRING		[s8a-d] for 850 μm , [s4a-d] for 450 μm arrays
ARRAYID	STRING		Manufacturer's serial number for this subarray
SHUTTER	STRING		Shutter position for dark frames
FILTER	STRING		Filter name
WAVELEN	FLOAT		[m] Filter centre wavelength
BOLODIST	FLOAT		[arcsec] Bolometer spacing
BBIN	LOGICAL	F	True if the blackbody source is in the beam
BBHEAT	FLOAT		[deg C] Blackbody temperature
PIXHEAT	INTEGER		Pixel heater D/A setting

Remaining content TBD.



B.2.7 Switching and Mapping Details

Keyword	Data Type	Default Value	Comment
COMMENT			– Switching and mapping details –
SAM_MODE	STRING		Sample mode: STARE, DREAM, JIGGLE or SCAN.
SW_MODE	STRING	'NONE'	Switch mode: CHOP, PSSW, FREQ or NONE
OBS_TYPE	STRING		Observation type - can be SCIENCE, POINTING or FOCUS
CHOP_CRD	STRING		Chop coordinate frame
CHOP_FRQ	FLOAT		[Hz] Chop frequency
CHOP_PA	FLOAT		[deg] Chop PA: 0 = in lat, 90 = in long
CHOP_THR	FLOAT		[arcsec] Chop throw
JIGL_CNT	INTEGER		DREAM 'leg length' parameter
JIGL_NAM	STRING		Name of DREAM jiggle pattern
JIGL_PA	FLOAT	0.0	[deg] PA of jiggle pattern (0.0 for DREAM)
JIGL_CRD	STRING		Coordinate frame of jiggle pattern (FPLANE for DREAM)
JIG_SCAL	FLOAT		[bolometers] Scale factor for DREAM pattern
DRMWGHTS	STRING		Name of file containing DREAM weights
MAP_HGHT	FLOAT		[arcsec] Requested Height of rectangle to be mapped
MAP_PA	FLOAT		[deg] Requested P.A. of map vertical, +ve towards +ve long
MAP_WDTH	FLOAT		[arcsec] Requested Width of rectangle to be mapped
LOCL_CRD	STRING		Local offset coordinate system
MAP_X	FLOAT		[arcsec] Requested Map X offset from telescope centre
MAP_Y	FLOAT		[arcsec] Requested Map Y offset from telescope centre
SCAN_CRD	STRING		Coordinate system of scan
SCAN_VEL	FLOAT		[arcsec/s] Requested scanning rate
SCAN_DY	FLOAT		Sample spacing perpendicular to scan
SCAN_PA	FLOAT		Scan PA relative to latitude (north) in SCAN_CRD system
SCAN_PAT	STRING		Scanning pattern (Lissajous, Pong, Boustrophedon, Raster)

Note that the CHOP_* entries are present for consistency only. It is unlikely that they will be used for SCUBA-2. If necessary, new keywords will be added to describe the SCUBA-2 scanning pattern(s).



B.2.8 Secondary Mirror Specific

Keyword	Data Type	Default Value	Comment
COMMENT			– Secondary Mirror Specific –
ALIGN_DX	FLOAT	0.0	SMU tables X axis alignment offset
ALIGN_DY	FLOAT	0.0	SMU tables Y axis alignment offset
FOCUS_DZ	FLOAT	0.0	SMU tables Z axis focus offset
DAZ	FLOAT	0.0	SMU azimuth pointing offset
DEL	FLOAT	0.0	SMU elevation pointing offset
UAZ	FLOAT	0.0	User azimuth pointing offset
UEL	FLOAT	0.0	User elevation pointing offset

B.2.9 JOS Parameters

The JOS recipe parameters control the overall flow of the observation. This section contains parameters common to all instruments.

Keyword	Data Type	Default Value	Comment
COMMENT			– JOS Parameters –
STEPTIME	FLOAT	5.0e-3	[s] RTS step time
NUM_CYC	INTEGER		Number of times to repeat recipe
STARTIDX	INTEGER		Index in pattern at start of observation
FOCAXIS	STRING		Focus axis to move (X, Y, Z)
NFOCSTEP	INTEGER		Number of focal position steps
FOCSTEP	FLOAT		[mm] Distance between focus steps
FOCPOSN	FLOAT		[mm] Absolute SMU position

Remaining content TBD.

B.2.10 Miscellaneous

Keyword	Data Type	Default Value	Comment
COMMENT			– Miscellaneous –
OCSCFG	STRING		Name of OCS Configuration XML file defining the observation
SIMULATE	LOGICAL	F	True if data are produced by simulator
SIM_SMU	LOGICAL	F	True if SMU data simulated
SIM_RTS	LOGICAL	F	True if RTS data simulated
SIM_TCS	LOGICAL	F	True if TCS data simulated
STATUS	STRING	'NORMAL'	Status at obs. end - should be with either NORMAL or ABORT
FLAT	STRING		Name of flat-field file

Remaining content TBD. There may be more SIM values (such as SIM_TEL and SIM_INST).



B.2.11 Polarimeter Specific

Keyword	Data Type	Default Value	Comment
COMMENT			– Polarimeter specific –
POL_CONN	LOGICAL		True if polarimeter is in the beam
POL_MODE	STRING		Step-and-integrate (STEPINT) or fast-spinning (FSPIN)
SKYANG	FLOAT		[rad] Angle of waveplate on the sky if STEPINT
ROTAFREQ	FLOAT		[Hz] Rotation rate in FSPIN
POLCALIN	LOGICAL		True if polarimeter calibrator is in the beam
POLWAVEN	LOGICAL		True if polarimeter waveplate is in the beam
POLANLIN	LOGICAL		True if polarimeter analyzer is in the beam

Note: SKYANG should only be written to the FITS header for the .In images.

B.2.12 FTS Specific

Keyword	Data Type	Default Value	Comment
COMMENT			– FTS specific –
FTS_CONN	LOGICAL		True if FTS is used
FTS_MODE	STRING		Step-and-integrate (STEPINT) or fast-scanning (FSCAN)
MIRPOS	FLOAT		[mm] Mirror position when in STEPINT mode
MIRVEL	FLOAT		[mm/s] Mirror scan rate in FSCAN

Note: MIRPOS should only be written to the FITS header for the .In images.

B.2.13 IMAGE specific

Keyword	Data Type	Default Value	Comment
COMMENT			– IMAGE specific –
SEQSTART	INTEGER		RTS index number of first frame contributing to image
SEQEND	INTEGER		RTS index number of last frame

These entries will only be added to reconstructed DREAM/STARE images.



B.3 Keywords for the DREAM weights file

Keyword	Data Type	Default Value	Comment
COMMENT			– DREAM weights file ONLY –
GRIDXMIN	INTEGER		Minimum X value in reconstruction grid
GRIDXMAX	INTEGER		Maximum X value in reconstruction grid
GRIDYMIN	INTEGER		Minimum Y value in reconstruction grid
GRIDYMAX	INTEGER		Maximum Y value in reconstruction grid
GRIDSIZE	FLOAT		Scale factor for grid spacing

These keywords are only applicable to the DREAM weights file and will not be written into the raw data.

