OCS/ICD/022

Version 3.2-1

Tim Jenness; Jamie Leech; Shaun de Witt;

Frossie Economou

31st May 2007

ACSIS File Format Interface Control Document

1 Introduction

ORAC-DR is a general purpose data driven reduction pipeline currently in use at UKIRT (for all instruments) and JCMT (for SCUBA and for all new instruments). The aim of ORAC-DR is to produce near-publication quality results for observers to aid in the implementation of flexible scheduling.

The ACSIS system generates spectra at a peak rate of 10MB/s (320 spectra per second). These spectra are forwarded from the "reducers" to the real-time gridders for cube creation (mainly used for quick look at the telescope) and are additionally written to disk for detailed data processing and archiving.[1]

ORAC-DR's role is two fold. Firstly, to generate calibration data and stacked and mosaicked cubes to provide near-real-time feedback to the observer, and secondly, to generate data products for the science archive hosted by CADC. ORAC-DR can be configured either to start from the processed cubes or to start from the individual spectra. This document describes the directory layout of these files, the data detection system and the contents of the files.

2 Directory Structure

At JCMT and UKIRT, data are written into standard, known locations for each instrument. Since there are multiple data acquisition computers in the ACSIS system and each computer needs to write to a local disk, the directory structure is more complex than previous instruments. An additional complication is that ACSIS uses the concept of physical and virtual machines and each physical computer can take on the identity of more than one virtual machine. For this reason the layut adopted in Figure 1 is more convoluted than initially thought. The advantage of this scheme is that if a single computer, say acsis07, dies, all that is required to run the system is to edit a single configuration file indicating that gridder07 (the task) should now run on the physical machine acsis02 (say) along with the virtual task gridder02. Since each virtual task writes data to its own subdirectory of the local data disk (which has been exported to the observer computers) there is no need to modify mount points on the observer systems. The control computer (acsis00) is responsible for writing 'flag' files (see the next section) that contain the information necessary to find the files that were written for a particular observation.

1 Additionally, ACSIS can write raw data to AIPS++ measurement sets. Since these are not archived and will be deleted within a few days they are not part of the public interface.

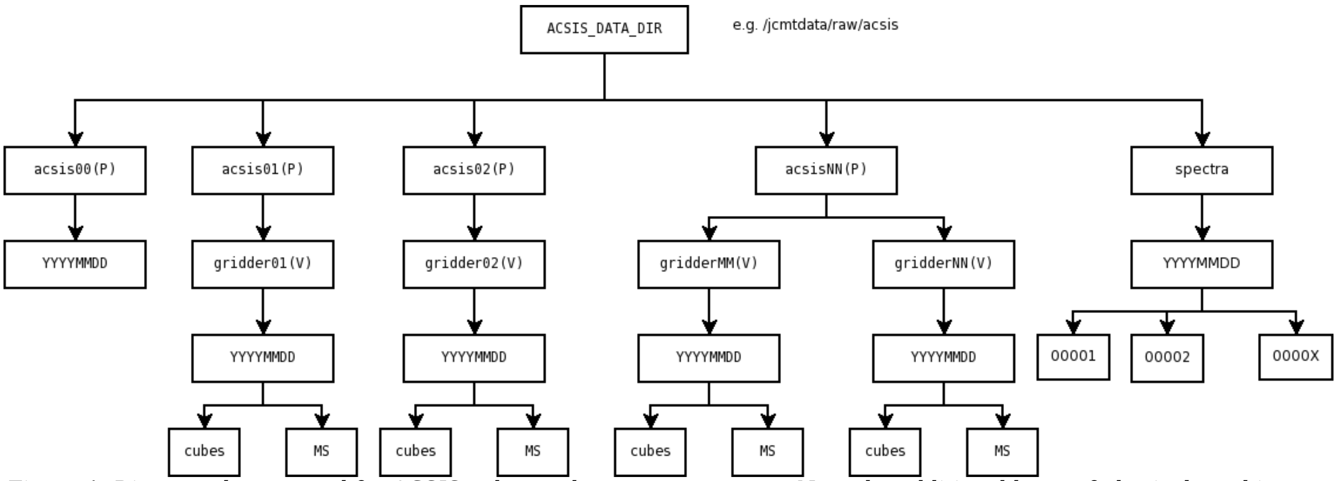

Figure 1: Directory layout used for ACSIS cubes and measurement sets. Note the additional layer of physical machine mount points and virtual machine sub directories. The 'P' indicates a physical machine name that is exporting disks to the rest of the system. The 'V' indicates a data directory for a virtual task running on that physical machine. In this example, the machines acsis01 and acsis02 are only running single gridders, whereas acsisNN has two virtual tasks running on it. The spectrum writer writes files into directories named after the observation number.

The computer that writes the archive data to disk (commonly called the 'specwriter') does not conform to this virtual/physical distinction since it is a specialist machine that has plenty of fast disk necessary for writing the 10MB/s (peak) to disk from all of the reducer tasks. Additionally, whereas cubes are written directly to the cubes directory (one per gridder task), the specwriter writes multiple files per observation to a directory named from the observation number. This is done to keep the files manageable in size (they can be closed whenever the writer task feels is appropriate) and enables a future upgrade where ORAC-DR could pick up the files before the observation itself as finished. These files are known as sub-scans (after the SCUBA-2 nomenclature).

2 Data Detection

The data reduction pipeline needs to know when new data are available to be processed. The standard approach adopted by all ORAC-DR instruments[2] is to write a hidden "flag" file (also called a ".ok" file) to a specific directory when the corresponding data files are ready for the specific observation. For ACSIS, this file then contains the path (relative to the `ORAC_DATA_IN` environment variable) to the actual data file or files associated with that observation. The definition of ORAC_DATA_IN depends on the pipeline mode as two different flag files are written.

The primary (archival) flag files are written to `ACSIS_DATA_DIR/spectra/YYYYMMDD` (where YYYYMMDD changes each night) and contains files written for the associated observation number. These flag files are named `.aYYYYMMDD_NNNNN.ok` (the 'a' being the instrument prefix).

The quick look flag files are written to `ACSIS_DATA_DIR/acsis00/YYYYMMDD`. Similarly to the archive flag files these contain the relative paths to all the files involved in the observation (which may be across multiple mount points). These flag files are named `.acYYYYMMDD_NNNNN.ok` (where the '`c`' indicates cubes).

3 File Naming Conventions

Each file written by the acquisition system must be uniquely named (since you can't guarantee

2 SCUBA-2 instituted an additional scheme based on DRAMA parameter monitoring, thus allowing the quick look pipeline to respond to data faster than files appear on disk. This is not necessary for ACSIS since the real-time system provides instantaneous quick look facilities.

they will always be in the separate directory structure defined above). The JAC convention is to use an instrument prefix, the UT date when the observation started and a (zero-padded) observation number. Additionally, with both SCUBA-2 and ACSIS a sub-scan is required and for ACSIS a single observation can include multiple frequency settings (up to 4 subsytems) and these must be stored in separate files. Archive files are therefore of the form `aYYYYMMDD_NNNNN_SS_MMMM.sdf`, where '`a`' is the instrument prefix for ACSIS, `YYYYMMDD` is the UT date (e.g. 20060420), `NNNNN` is the zero-padded observation number, SS is the zero-padded subsystem number (0 to 3) and MMMM is the zero-padded sub-scan number.

Gridded cubes for the quick look system follow a similar scheme except that there is no sub-scan number and a two digit gridder number is used instead, i.e. `acYYYYMMDD_NNNNN_SS_GG.sdf`.[3] Note that the instrument prefix is here 'ac' (following on from the flag file convention).

4 Archive Data Format

The archive data format was designed with the following requirements:

- Provide easy access to individual calibrated spectra.
- Allow calibrations applied to the spectra during real-time processing to be analysed. They do not have to be completely reversible because of the complication of sub-band merging.
- The data file must be compatible with the archive data reduction pipeline.
- Where possible, make use of the pre-existing SCUBA-2 file layout.

Neither the Quick Look cubes, nor the raw AIPS++ Measurement Set lags (or, indeed, any Measurement Set format) meet these requirements. The following file structure is used:

- Store all spectral data in a 3 dimensional array with dimensions corresponding to frequency (channels), number of receptors involved in the observations and incrementing sequence number.
- Store corresponding sequence information in a number of arrays in an extension.
- Store receptor positions in an extension with dimensions corresponding to “x” and “y” position, number of receptors, and sequence.
- Headers that are fixed (or correspond to the beginning and end) for the observation are stored in a FITS header extension.
- Frequency scales for the spectral axis specified by an AST frameset.
- Calibration information (e.g. REFSKY, LOAD2, AMBIENT, TSYS) will be stored unmerged in an extension array. They are not stored in the primary array since there can be different numbers of channels depending on sub-band merging.[4]
- Data from different subsystems will end up in different files.

Each component will be discussed in turn, with reference to an actual trace (some of the specific details of this trace depend on the internal layout of NDF itself and so should not be treated as part of the ICD as such. These include the definition of an ARRAY structure, the location of UNITS and LABEL components, and the HISTORY structure):

```
A20060607_01_0005  <NDF>

   DATA_ARRAY     <ARRAY>         {structure}
```

3 A configuration option controls whether FITS or NDF data are written (or both). ORAC-DR for quick look processing will always read the NDF versions and these are the files referenced in the flag files.

4 We do not store unmerged or uncalibrated data since we do not wish to reimplement the online reduction system (including, but not limited to, sub-band merging, nod subtraction, and frequency switch processing). The goal here is to provide the spectra as easily as possible to the user but allowing easy analysis and visualization of calibration information.

```
      DATA(4096,16,2000)  <_REAL>    0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,
                                     ... 4104,4105,4106,4107,4108,4109,4110
      ORIGIN(3)      <_INTEGER>      1,1,1
   UNITS <_CHAR*1>            'K'
   LABEL <_CHAR*>             'T*A corrected antenna temperature'
```

In this example, there are 2000 spectra from 16 receptors, each of which contains 4096 channels. The spectral data are stored as 4 byte floating point numbers without compression (compression to 16 bit integers would make the data less accessible but could save almost 50% in disk space. Additionally, given that many of the spectra are already co-added before they are written to disk or are the result of long integrations, data compression to 16-bit may not always be possible even if a system was implemented). For uncalibrated data the units will be “uncalibrated” and the label will be modified to “Power”.

```
   HISTORY        <HISTORY>       {structure}
      CREATED        <_CHAR*24>      '2006-MAR-22 11:23:37.000'
      CURRENT_RECORD  <_INTEGER>     1
      RECORDS(10)    <HIST_REC>      {array of structures}

      Contents of RECORDS(1)
         DATE           <_CHAR*24>      '2006-MAR-22 11:23:37.000'
         COMMAND        <_CHAR*17>      'ACSIS-DA (V0.1-1)'
         USER           <_CHAR*4>       'operator'
         HOST           <_CHAR*21>      'apele.jach.hawaii.edu'
         DATASET        <_CHAR*83>      '/home/operator/acsis/spectra/sc...'
         TEXT(1)        <_CHAR*21>      'ACSIS Data Acquistion'
```

A HISTORY structure is included and will be appended to by each DR operation as output files are created. The DA will create two HISTORY entries, one when the file is first written to disk and another entry when the header information is finally complete.

```
   MORE           <EXT>           {structure}
      JCMTOCS        <OCSINFO>       {structure}
         CONFIG(1865)   <_CHAR*72>      '<?xml version="1.0" encoding="US...'
                                        ... '/interface>...','  </OCS_CONFIG>'
```

The JCMTOCS extension contains the OCS configuration XML that was used to specify the observation itself. It is stored in a _CHAR array but is really a single character string (including embedded newlines). SMURF can be used to retrieve the XML in a readable form.

```
      ACSIS          <ACSIS_COMP>    {structure}
         RECEPTORS(16)  <_CHAR*3>       'H01','H02','H03','H04','H05','H06',
                                        'H12','H13','H14','H15','H16'
         FOCAL_STATION <_CHAR*6>        'DIRECT'
         FPLANEX(16) <_FLOAT>            0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15
         FPLANEY(16) <_FLOAT>            1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16
         RECEPPOS_SYS <_CHAR*8>          'TRACKING'
         RECEPPOS(2,16,2000)  <_DOUBLE>   1,2,1,2,1,2,1,2,1,2,1,2,1,2,1,2,
                                          ... 1,2,1,2,1,2,1,2,1,2,1,2,1,2,1,2
         TSYS(16,2000)  <_FLOAT>        230.0,230.0,230.0,230.0,230.0,
                                          ... 230.0,230.0
         TRX(16,2000)  <_FLOAT>        230.0,230.0,230.0,230.0,230.0,
                                          ... 230.0,230.0
```

The ACSIS extension contains the names of the receptors, the focal station of the instrument, the positions of the receptors in the focal plane (arcsec offsets), the absolute positions of each receptor in either TRACKING or AZEL coordinate (stored in radians with the frame indicated by

RECEPPOS_SYS), and the median system temperature (Tsys) and receiver temperatures (Trx) for each receptor and step. Note that the receptor positions in the tracking frame will have been calculated using the original focal plane receptor positions. If those positions are updated, the telescope positions will be invalidated and should be recalculated (as is done by SMURF).

```
CALDATA    <NDF>
   DATA_ARRAY    <ARRAY>       {structure}
      DATA(4096,16,1)   <_REAL>  *,*,*,*,*,*,*,*,*,*,*,*,*,*,*,*
      ORIGIN   <_INTEGER>        1,1,1
   MORE
      ACSIS
      JCMTSTATE
```

The (optional) CALDATA extension is an NDF within the main NDF containing unmerged calibration spectra. It's layout is identical to the parent NDF (including the same extensions (except for CALDATA itself).

```
JCMTSTATE      <RTS_ARR>       {structure}
```

The JCMTSTATE extension contains all the data that varies within sequences[5] and matches the structure used by SCUBA-2 data files[6]. The ACS_ fields are for ACSIS-specific data. Some fields will be set to bad values in some sequences (for example during cals there will be no need for telescope information).

```
RTS_NUM(2000)  <_INTEGER>      1,2,3,4,5,6,7,8,9,10,11,12,13,14,
                               ... 1994,1995,1996,1997,1998,1999,2000
```

RTS sequence number to uniquely identify this information. [RTS.STATE.NUMBER]

```
RTS_END(2000)  <_DOUBLE>       53797.000000579,53797.000001157,
                               ... 53797.001156822,53797.001157401
```

Modified Julian date (TAI) when the step ended. [RTS.STATE.TAI_END]

```
RTS_TASKS(2000)  <_CHAR*80>    'SIMULATOR','SIMULATOR','SIMULATOR',
                               ... 'SIMUL...','SIMULATOR','SIMULATOR'
```

Tasks involved in the sequence. [RTS.STATE.TASKS]

```
SMU_X(2000)    <_DOUBLE>       0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                               ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
SMU_Y(2000)    <_DOUBLE>       0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                               ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
SMU_Z(2000)    <_DOUBLE>       0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                               ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

SMU Focus position from nominal. [SMU.STATE.SMU_X, SMU.STATE.SMU_Y and SMU.STATE.SMU_Z]

```
SMU_CHOP_PHASE(2000) <_CHAR*1> '','','','','','','','','','',''
```

Chop phase (A, B or M). Note that whilst a particular spectrum may well have been nod and chop subtracted (and so the actual chop phase is not really meaningful) this value is important for consistency in calculations based on the TCS and SMU coordinates themselves.

---

5 Technically the ENVIRO information does not arrive during a sequence and is updated more slowly. From the file format viewpoint, each sequence has associated weather data.

6 SCUBA-2 data files have WVM information instead of ENVIRO and SC2_* information instead of ACS_* fields.

[SMU.STATE.CHOP_PHASE]

```
SMU_JIG_INDEX(2000)  <_INTEGER> 1,2,3,4,5,6,7,8,9,10,11,12,1,2,3
```

Current jiggle position. (Bad value if not jiggling). [SMU.STATE.JIG_INDEX]

```
SMU_AZ_JIG_X(2000)  <_DOUBLE>   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                                ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
SMU_AZ_JIG_Y(2000)  <_DOUBLE>   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                                ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

SMU Jiggle position in AZEL coordinates (arcsec). [SMU.STATE.AZEL.JIGGLE.C1 and SMU.STATE.AZEL.JIGGLE.C2]

```
SMU_AZ_CHOP_X(2000)  <_DOUBLE>   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                                ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
SMU_AZ_CHOP_Y(2000)  <_DOUBLE>   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                                ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

SMU Chop position in AZEL coordinates (arcsec). As for SMU_CHOP_PHASE this field is used when calculating consistent telescope positions. [SMU.STATE.AZEL.CHOP.C1 and SMU.STATE.AZEL.CHOP.C2]

```
SMU_TR_JIG_X(2000)  <_DOUBLE>   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                                ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
SMU_TR_JIG_Y(2000)  <_DOUBLE>   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                                ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

SMU jiggle position in tracking coordinates (arcsec). [SMU.STATE.TRACKING.JIGGLE.C1 and SMU.STATE.TRACKING.JIGGLE.C2]

```
SMU_TR_CHOP_X(2000)  <_DOUBLE>   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                                ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
SMU_TR_CHOP_Y(2000)  <_DOUBLE>   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                                ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

SMU Chop position in tracking coordinates (arcsec). Similarly for SMU_AZ_CHOP_*. [SMU.STATE.TRACKING.CHOP.C1 and SMU.STATE.TRACKING.CHOP.C2]

```
TCS_TAI(2000) <_DOUBLE>        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                              ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

TAI time used to determine TCS values in this extension (may differ from RTS_END by half a step time since the TCS uses the average time of the step when reporting positions). [PTCS.STATE.TAI]

```
TCS_AIRMASS(2000)  <_DOUBLE>   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                               ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

Airmass of the telescope tracking centre. [PTCS.STATE.AIRMASS].

```
TCS_AZ_ANG(2000)  <_DOUBLE>    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                               ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

Angle between PA=0 in the focal plane and PA=0 in AZEL (i.e. Elevation) (radians).

[PTCS.STATE.AZEL.FP_ANGLE]

```
TCS_AZ_AC1(2000)  <_DOUBLE>    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                               ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
TCS_AZ_AC2(2000)  <_DOUBLE>    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                               ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

Actual coordinates of the telescope (radians) in AZEL. [PTCS.STATE.AZEL..ACTUAL.C1 and PTCS.STATE.AZEL.ACTUAL.C2]

```
TCS_AZ_DC1(2000)  <_DOUBLE>    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                               ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
TCS_AZ_DC2(2000)  <_DOUBLE>    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                               ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

Demand coordinates of the telescope (radians) in AZEL. [PTCS.STATE.AZEL.DEMAND.C1 and PTCS.STATE.AZEL.DEMAND.C2]

```
TCS_AZ_BC1(2000)  <_DOUBLE>    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                               ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
TCS_AZ_BC2(2000)  <_DOUBLE>    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                               ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

Coordinates of the base position (radians) in AZEL. [PTCS.STATE.AZEL.BASE.C1 and PTCS.STATE.AZEL.BASE.C2]

```
TCS_BEAM(2000) <_CHAR*1> '','','','','','','','','','','',''
```

Telescope nod position (A, B or M). Similar comments to those appropriate to SMU_CHOP_PHASE are also valid for TCS_BEAM. [PTCS.STATE.BEAM]

```
TCS_INDEX(2000)  <_INTEGER>    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                               ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

Index into the observing area. Usually either row number or grid offset). [PTCS.STATE.INDEX]

```
TCS_SOURCE(2000)  <_CHAR*32>   ' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',
                               '','','','','','','','','',''
```

Label of the base position. Usually SCIENCE or REFERENCE. [PTCS.STATE.SOURCE]

```
TCS_TR_SYS(2000)  <_CHAR*16>   ' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',
                               '','','','','','','','','',''
```

Name of the Tracking coordinate frame. (e.g. AZEL, J200, B1950). [PTCS.STATE.TRACKING.SYSTEM]

```
TCS_TR_ANG(2000)  <_DOUBLE>    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                               ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

Angle between focal plane PA=0 and PA=0 in the tracking coordinate frame (radians) [PTCS.STATE.TRACKING.FP_ANGLE]

```
TCS_TR_AC1(2000)  <_DOUBLE>    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                               ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
TCS_TR_AC2(2000)  <_DOUBLE>    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                               ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

Actual telescope coordinates in the TRACKING frame (radians) [PTCS.STATE.TRACKING.ACTUAL.C1 and PTCS.STATE.TRACKING.ACTUAL.C2]

```
TCS_TR_DC1(2000)  <_DOUBLE>    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                               ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
TCS_TR_DC2(2000)  <_DOUBLE>    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                               ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

Demand telescope coordinates in the TRACKING frame (radians) [PTCS.STATE.TRACKING.DEMAND.C1 and PTCS.STATE.TRACKING.DEMAND.C2]

```
TCS_TR_BC1(2000)  <_DOUBLE>    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                               ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
TCS_TR_BC2(2000)  <_DOUBLE>    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                               ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

Base telescope coordinate in the TRACKING frame (radians). Can be varying if a planet or asteroid is being tracked. [PTCS.STATE.TRACKING.BASE.C1 and PTCS.STATE.TRACKING.BASE.C2]

```
JOS_DRCONTROL(2000)  <_INTEGER>   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                                  ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

This is the DR control flag sent from the JOS (possibly augmented by ACSIS DR itself). It is a bit mask.

```
ENVIRO_REL_HUM(2000)  <_REAL>   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                                ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

Relative humidity (%).

```
ENVIRO_PRESSURE(2000)  <_REAL>   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                                 ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

Atmospheric pressure.

```
ENVIRO_AIR_TEMP(2000)  <_REAL>   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                                 ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

Air temperature (K).

```
ACS_SOURCE_RO(2000)  <_CHAR*16>   ' ',' ',' ',' ',' ',' ',' ',' ',
                                  '','','','','','','','','',''
```

This is the core description of the spectrum data and where it came from in the real time reduction system. In most cases this will contain “SPECTRUM_RESULT”. For FOCUS observations it will contain “SOURCE”. In the calibration extension this can be any string coming out of the reducers (e.g. SOURCE_CAL, LOAD2, TSYS, AMBIENT).

```
ACS_NO_PREV_REF(2000)  <_INTEGER>   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
```

```
                                        ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
ACS_NO_NEXT_REF(2000)  <_INTEGER>   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                                        ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
ACS_NO_ONS(2000)       <_INTEGER>   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                                        ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

These parameters describe the number of references that were taken before and after this spectrum along with the number of steps that were observed on source between those references (used in the calculation of variance for each spectrum). These can be bad for observations such as frequency switch that have no reference spectra. These three items are now deprecated and have been replaced by the ACS_EXPOSURE and ACS_OFFEXPOSURE. In particular, the ACS_NO_PREV_REF and ACS_NO_NEXT_REF were never reliable and should not be believed.

```
ACS_EXPOSURE(2000)  <_REAL>   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                                        ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

This is the total on source integration time for this spectrum (the same for all receptors) in seconds. For some observing modes this will simply be the step time.

```
ACS_OFFEXPOSURE(2000) <_REAL> 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
                                  ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

This is the total off integration time for this spectrum (same for all receptors) in seconds.

```
POL_ANG(2000)  <_DOUBLE>       3.14159,3.14159,3.14159,3.14159,
                                  ... 3.14159,3.14159,3.14159,3.14159
```

Angle of the polarimeter waveplate (in polarimeter coordinates) during this sequence. Will be a bad value if the polarimeter is not involved in the sequence [ROVER.STATE.ANGLE.WPLATE]

```
FE_LOFREQ(2000)  <_DOUBLE>   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                                        ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

The LO frequency used by the frontend.

```
FE_DOPPLER(2000)  <_DOUBLE>   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                                        ... 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

The Doppler correction used by the frontend.

Given the asynchronous nature of data arrival, the first and/or last entries in a given file may not be present and may be found in adjacent files. In that case sequence information will be duplicated between files. Also, since for some observing modes spectra are coadded in the reducers, the state information stored in the file will be that associated with the last step in the sequence rather than an average from all the steps. This is particularly important for modes such as jiggle/chop where nods are subtracted before forming the spectrum and many seconds may have elapsed.

To simplify compatibility with the data reduction pipeline and SCUBA-2, these files will use the Starlink N-Dimensional Format (NDF) which is based on the Starlink Hierarchical Data Format. If required, these files can be converted to multi-extension FITS by end users.

5 Quick Look Data Format

The quick-look system writes gridded cubes to disk to locations described earlier. These cubes have dimensions of RA/Dec and frequency, and for Focus observations there is an additional dimension corresponding to SMU focus position. The headers are described in the next section. An ACSIS extension contains an image of the associated spatial variance (which can be expanded over all frequency channels in ORAC-DR).

6 FITS Headers

Since ORAC-DR receives no external inputs other than the data the data must contain all the information required for ORAC-DR to reduce it. The FITS headers are defined below. Where possible, these header definitions are shared with the SCUBA-2 data files. Shared headers are noted where applicable. Headers only present in quick look cubes are indicated in **bold**.

6.1 General FITS

The FITS headers stored in the NDF FITS "airlock" do not need to contain the basic data array designations for data type and dimensionality (e.g. BITPIX, NAXIS, BUNIT etc) since these are contained in the NDF specification itself. If FITS writing is enabled, then those mandatory keywords must be added.

6.2 Telescope Specific

Keyword	***Data Type***	***Default Value***	***Comment***
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	-5464545.04	[m]
OBSGEO-Y	FLOAT	-2492986.33	[m]
OBSGEO-Z	FLOAT	2150635.34	[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

These items are shared between ACSIS and SCUBA-2. The telescope position is obtained directly from the telescope control system rather than being a hard-coded value.

6.3 WCS Info and Axis Descriptions

The world coordinate information is specified in two ways depending on the format of the underlying file. Cubes are written in 3 dimensions of RA/Dec (or whatever spatial coordinate frame is in use) and frequency. For FITS cubes the world coordinate information must match that defined in the first 3 FITS papers. For NDF cubes, an AST frameset must be created (usually from FITS Paper I,II,III) of a SkyFrame and a DSBSpecFrame. Focus observations are written as 4-dimensional hypercubes with the fourth axis corresponding to secondary mirror position (the first 3 dimensions are defined identically to the normal cubes).

The world coordinates for the archive format are contained in an AST 3-dimensional frameset of a DSBSpecFrame, LutMap (for receptor ID) and TimeFrame (the modified Julian Date).

Data units (and associated label) are part of the NDF standard and must be included in the body of the file itself. For FITS cubes, the standard BUNIT field must be filled in. Additionally, for FITS cubes the IMAGFREQ keyword must be filled in (for NDF this is handled by a DSBSpecFrame).

In all cases the frequency axis will correspond to a single sub-system of merged data. Calibration data will by necessity be unmerged and a frequency scale will therefore be more complex (and will be implemented using AST switchMaps).

One additional comment is required regarding the use of the MJD-AVG header in cubes. This header is used to specify the reference epoch used for constructing the spatial and spectral component of the cube (especially important for AZEL cubes or apparent RA/Dec) and is not simply the epoch of the middle of the observation. It differs from DATE-OBS in that DATE-OBS corresponds to the start of the observation but is not necessarily the reference epoch.

6.4 OMP and ORAC-DR Specific

Keyword	Data Type	Default Value	Comment
COMMENT			-- OMP and ORAC-DR specific ---
PROJECT	STRING		The PATT no. for the PROJECT
RECIPE	STRING		The ORAC-DR recipe
DRGROUP	INTEGER		ORAC-DR Group ID
MSBID	STRING		ID of min schedulable block
MSBTID	STRING		MSB Transaction ID
SURVEY	STRING		Survey Name
RMTAGENT	STRING		Name of remote agent
AGENTID	STRING		Unique identifier for remote agent.

These items are shared between SCUBA-2 and ACSIS. The values of these items will come from the translator.

6.5 Observation, Date and Pointing Specific

These headers are common to ACSIS and SCUBA-2.

Keyword	Data Type	Default Value	Comment
OBSID	STRING		Unique observation identifier[7]
OBSIDSS	STRING		Unique observation + subsystem ID (see SUBSYSNR)
OBJECT	STRING		Object Name
STANDARD	LOGICAL	F	True if this is a spectral line standard
OBSNUM	INTEGER		Observation Number
NSUBSCAN	INTEGER		Sub-scan number
OBSEND	LOGICAL	F	True if file is last in current observation
UTDATE	INTEGER		UT date as an integer in yyyymmdd format
DATE-OBS	STRING		Date and time (UTC) of start of observation[8]
DATE-END	STRING		Date and time (UTC) of end of observation
DUT1	FLOAT		[d] UT1-UTC correction
INSTAP	STRING		TCS Instrument Aperture in use (if any)
INSTAP_X	FLOAT		[arcsec] Aperture X off. rel. to instr centre
INSTAP_Y	FLOAT		[arcsec] Aperture Y off. rel. to instr centre
AMSTART	FLOAT		Airmass at start of observation
AMEND	FLOAT		Airmass at end of observation

7 The OBSID is formed from the backend name, OBSNUM and DATE-OBS. (see document JSA/ANA/001)
8 DATE-OBS, DATE-END, HSTSTART and HSTEND In ISO8601 format; i.e. YYYY-MM-DDTHH:MM:SS

Keyword	Data Type	Default Value	Comment
AZSTART	FLOAT		[deg] Azimuth at observation start
AZEND	FLOAT		[deg] Azimuth at observation end
ELSTART	FLOAT		[deg] Elevation at observation start
ELEND	FLOAT		[deg] Elevation at observation end
HSTSTART	STRING		HST at start of observation
HSTEND	STRING		HST at end of observation
LSTSTART	STRING		Local sidereal time at start of observation[9]
LSTEND	STRING		Local sidereal time at end of observation

6.6 Integration Time Specific

Keyword	Default Value	Comment
COMMENT		-- Integration time related ---
INT_TIME		[s] Time spent integrating
EXP_TIME		[s] Mean integration time per map pixel

The EXP_TIME header only exists for quick look cubes (not ACSIS spectra or SCUBA-2 files) and contains a simple summary of the integration time in the gridded cube.

6.7 ACSIS Specific

Keyword	Data Type	Default Value	Comment
COMMENT			-- ACSIS specific --
BACKEND	STRING	ACSIS	Name of this backend (ACSIS)
MOLECULE	STRING		Target molecule species
TRANSITI	STRING		Target transition for MOLECULE
TEMPSCAL	STRING	'TA*'	Temperature scale in use
DRRECIPE	STRING		The ACSIS-DR recipe name
BWMODE	STRING		ACSIS total bandwidth setup
SUBSYSNR	INTEGER	0	Sub-system number
SUBBANDS	STRING		ACSIS sub-band set up
NSUBBAND	INTEGER		Number of subbands
SUBREFP1	INTEGER		Reference channel for first subband
SUBREFP2	INTEGER		Reference channel for second subband
NCHNSUBS	INTEGER		Number of channels in each subband
REFCHAN	INTEGER		Reference IF channel number (first subband)
IFCHANSP	FLOAT		[Hz] TOPO IF channel spacing (signed)
FFT_WIN	STRING		Type of window used before FFT
BEDEGFAC	FLOAT	1.23	Backend degradation factor
GRIDFUNC	STRING		Gridding convolution function (tophat or Gaussian)
CONVWID	FLOAT		[arcsec] Width of gridding conv function (Gauss)
TRUNCRAD	FLOAT		[arcsec] Trunc. radius of gridding function
MSROOT	STRING		Root name of raw measurement sets

Parameters concerning grid parameters (GRIDFUNC, CONVWID and TRUNCRAD) are only present in output cubes. SUBREFP1 and SUBREFP2 indicate the relative position of the overlap region in non-merged spectra. If 4 sub-band spectra are ever taken additional keywords should be included.

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

6.8 Front End Specific

Keyword	Data Type	Default Value	Comment
COMMENT			-- FE specific ----
INSTRUME	STRING		Name of Front End Receiver
SB_MODE	STRING		Sideband mode
IFFREQ	FLOAT		[GHz] IF Frequency
N_MIX	INTEGER		No. of Mixers.
OBS_SB	STRING		The observed sideband “USB” or “SSB”
LOFREQS	FLOAT		[GHz] LO frequency at start of observation.
LOFREQE	FLOAT		[GHz] LO frequency at end of observation.
RECPTORS	STRING		Active receptors in this observation.
REFRECEP	STRING		Receptor with unit sensitivity
MEDTSYS	FLOAT		[K] Median of Tsys across all receptors
MEDRMS	FLOAT		[K] Median of RMS noise in cube
DOPPLER	STRING		Doppler velocity definition (radio, optical, redshift).
SSYSOBS			Spectral reference frame fixed during observation.

Only INSTRUME is shared with SCUBA-2. MEDRMS is only present in output cubes.

6.9 Environment Specific

Keyword	Data Type	Default Value	Comment
COMMENT			-- Environment specific --
ATSTART	FLOAT		[degC] Air_Temp at start of observation
ATEND	FLOAT		[degC] Air_Temp at end of observation
HUMSTART	FLOAT		Rel humidity at start of observation
HUMEND	FLOAT		Rel humidity at end of observation
BPSTART	FLOAT		[mbar] Pressure inside dome at start of observation
BPEND	FLOAT		[mbar] Pressure inside dome at end of observation
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		Tau at 225 GHz from CSO at start of observation
TAU225EN	FLOAT		Tau at 225 GHz from CSO at end of observation
TAUDATST	STRING		Date of TAU225ST value
TAUDATEN	STRING		Date of TAU225EN value
TAUSRC	STRING		TAU225 Source (CSO or NRAO350MICRON)
WVMTAUST	FLOAT		Most recently calculated median 186GHz tau from JCMT WVM Start obs
WVMTAUEN	FLOAT		Most recently calculated median 186GHz tau from JCMT WVM End obs
WVMDATST	STRING		Date of WVMTAUST value
WVMDATEN	STRING		Date of WVMTAUEN value
SEEINGST	FLOAT		[arcsec] SAO atmospheric seeing - Start obs
SEEINGEN	FLOAT		[arcsec] SAO atmospheric seeing - End obs
SEEDATST	STRING		Date of SEEINGST value
SEEDATEN	STRING		Date of SEEINGEN value

Keyword	Data Type	Default Value	Comment
FRLEGTST	FLOAT		[degC] Mean front Leg Temperature - Start obs
FRLEGTEN	FLOAT		[degC] Mean front Leg Temperature - End obs
BKLEGTST	FLOAT		[degC] Mean back Leg Temperature - Start obs
BKLEGTEN	FLOAT		[degC] Mean back Leg Temperature - End obs

These headers are shared with SCUBA-2.

6.10 Switching and Mapping Details

Keyword	Data Type	Default Value	Comment
COMMENT			-- Switching and mapping details --
SAM_MODE	STRING		Sample Mode - should be jiggle, raster or grid
SW_MODE	STRING	NONE	Switch Mode – chop, pssw, freq or NONE
SKYREFX	STRING		X co-ord of reference position
SKYREFY	STRING		Y co-ord of reference position
OBS_TYPE	STRING		Obs type should be science, pointing or focus
CHOP_CRD	STRING		Chopper coordinate system
CHOP_FRQ	FLOAT		[Hz] Chopper frequency
CHOP_PA	FLOAT		[deg] Chopper P.A., 0 = in lat, 90 = in long
CHOP_THR	FLOAT		[arcsec] Chopper throw
JIGL_CNT	INTEGER		Number of offsets in jiggle pattern
JIGL_NAM	STRING		File containing jiggle offsets
JIG_PA	FLOAT		[deg] Position angle of jiggle pattern
JIG_CRD	STRING		Coordinate frame of jiggle pattern
JIG_SCAL	FLOAT		Scale size of jiggle pattern
ROT_CRD	STRING		Coordinate frame of image rotator
ROT_PA	FLOAT		[deg] Angle of image rotator
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 for MAP_X/MAP_Y
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/sec] Scan velocity
SCAN_DY	FLOAT		[arcsec] Sample spacing perp. to scan
SCAN_PA	FLOAT		[deg] Scan P.A. rel. to lat. line; 0=lat, 90=long in SCAN_CRD system.
SCAN_PAT	STRING		Name of scanning scheme (raster, boustrophedon[10], pong etc)

All except SKYREFX and SKYREFY are present in both ACSIS and SCUBA-2[11]. Note that scanning parameters are currently only defined for “raster” and “boustrophedon” scanning strategies. As SCUBA-2 is developed additional scan patterns will be developed and it is expected that parameters used to define those scan patterns will be added to all JCMT FITS headers.

10 boustrophedon is a raster where scan reversal is true.
11 SCUBA-2 is not expected to chop but the keywords will be present even so.

6.11 Secondary Mirror Specific

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

These headers are shared with SCUBA-2.

6.12 JOS Parameters

The JOS recipe parameters control the overall flow of the observation. This section contains all possible JOS recipe settings. Only certain parameters can be shared with all instruments. Currently those parameters are STEPTIME, NUM_CYC, STARTIDX, FOCAXIS, NFOCSTEP and FOCSTEP.

Keyword	Data Type	Default Value	Comment
COMMENT			--JOS parameters--
STEPTIME	FLOAT		[s] RTS step time
NUM_CYC	INTEGER		Number of times to repeat entire recipe
NUM_NODS	INTEGER		Number of nod sets repeated
JOS_MULT	INTEGER		???
JOS_MIN	INTEGER		???
NCALSTEP	INTEGER		Number of RTS Steps for each CAL
NREFSTEP	INTEGER		Number of RTS Steps for each REF
STBETREF	INTEGER		Max number of steps between refs
STBETCAL	INTEGER		Max Number of steps between cals
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

6.13 Miscellaneous

Keyword	Data Type	Default Value	Comment
COMMENT			--Miscellaneous--
OCSCFG	STRING		OCS configuration file
SIMULATE	LOGICAL	F	True if any data are simulated
SIM_CORR	LOGICAL	F	True if Correlator data are simulated
SIM_SMU	LOGICAL	F	True if SMU data simulated
SIM_TCS	LOGICAL	F	True if TCS data is simulated
SIM_RTS	LOGICAL	F	True if RTS data is simulated
SIM_IF	LOGICAL	F	True if IF data is simulated
STATUS	STRING	‘NORMAL’	status at obs. end - should be with either NORMAL or ABORT

All except the SIM_IF and SIM_CORR items are shared with SCUBA-2.

6.14 ROVER Polarimeter Specific

Keyword	***Data Type***	***Default Value***	***Comment***
COMMENT			--ROVER polarimeter specific --
POL_CONN	LOGICAL	F	True if ROVER polarimeter is fitted
POL_MODE	STRING		Step-and-integrate (STEPINT) or spinning (SPIN)
SKYANG	FLOAT		[deg] Angle of ROVER waveplate on the sky
ROTAFREQ	FLOAT		[Hz] Spin frequency (if spinning)
POL_CRD	STRING		Coordinate frame of polarimeter angles
POLFAXIS	FLOAT		[deg] Angle of pol fast axis

The remainder of the content is TBD, awaiting ROVER commissioning. These items are shared between ROVER (ACSIS) and POL-2 (SCUBA-2) data files. SKYANG is only defined in cubes (else it is present in the file extension).