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: SCUBA-2 Data Display System

Document Number: SC2/SOF/S210/002

Issue: 1.32

Date: 2005/05/25

Document Prepared By:	Tim Jenness	Signature and Date:	2005-05-25
Document Approved By:	Douglas Scott	Signature and Date:	2005-05-26
Document Released By:	Janos Molnar	Signature and Date:	2005-05-26

Contents

1 Introduction

This document describes the data display systems proposed for SCUBA-2, and is based partly on the display requirements described in SC2/SRE/S210/001.[20]

The display system is entirely data driven via the Data Acquisition system. Data are either picked up via files appearing on disk or via DRAMA parameters, where DREAM/STARE images are generated at a much faster rate than files are written.[11] The Quick Look and Pipeline displays are both driven by an ORAC-DR pipeline but do not interfere with the pipeline. ***Req. DR1***

In addition to strip-chart diagnostics,[8] there are three different views of science data that should be displayed: Movie, Quick Look and Science pipelines. The Movie system displays the raw 200 Hz frames to the observer or commissioning scientist, with minimal sky-subtraction and flat-fielding, independently of the underlying observing mode. Quick-look (QL) displays minimally processed images (the current frame and the current observation co-add) to the observer as fast as possible (without any analysis or calibration). And the Pipeline display is responsible for displaying to the observer properly reduced data, including full group (multi-observation) co-adds.

Each of these systems and details of implementation will be discussed in the subsequent sections.

2 Definitions

The following definitions are important for this document:

sample
: Data corresponding to a single time-slice in the raw data array (usually sampled at 200 Hz). Implicitly includes all the (40×32) bolometers read out at that time.

subscan
: A subscan corresponds to a single data file written by a single DA computer from a single sub-array.

frame
: The smallest data unit manipulated by the Pipeline. What this corresponds to depends on the display mode. For the QL pipeline a frame will consist of the 4 processed images or scan files supplied by the 4 sub-arrays. For the science pipeline a frame consists of the filenames written to the flag file which may themselves contain multiple DREAM/STARE reconstructions.

observation
: The collection of frames corresponding to a single CONFIGURE in the observation sequencer. For a pointing observation this might be a single frame, for a SCAN map this may correspond to multiple scans covering a single observing area, and for DREAM this may be multiple repeats of a single region.

group

A collection of related observations, that can be combined by the Pipeline. For example, a group can correspond to repeats of identical observations to improve signal-to-noise, or mosaics, or large areas. For the QL pipeline, a group and an observation are synonymous since the QL pipeline does not coadd across observations.

3 "Movie" display

Data are acquired at a rate of 200 Hz from each of the 40×32 bolometer sub-arrays. The Movie display will display these raw images (with basic data processing: flat-fielding, dark subtraction and simple sky subtraction) as an animated sequence ("Movie"). There will be one Movie display per wavelength.

There will be two modes of operation:

On-line On-line mode will be running all the time during observing and monitoring the arrival of new data files on disk. When a new file appears, the frames will be displayed at full speed (albeit one frame behind real-time) to give the impression of a real-time display. Since the maximum useful display rate is assumed to be about 25 Hz (TV rate, less if that is too challenging) this will be achieved either by throwing away 7 samples out of every 8 (DREAM or SCAN) or by co-adding samples (STARE). For DREAM mode the pixel array will be shifted to correct for the DREAM pattern such that a bright source will not move around on the display (except for sky rotation which may be significant over a long exposure). ***Req. DR5***

The movie mode processing will be implemented using a specialist ORAC-DR recipe. The movie pipeline will monitor the flag files in a similar manner to the science Pipeline since it relies on files appearing on disk to obtain the 200 Hz data (the DRAMA parameters do not make 200 Hz data available).

The standard ORAC-DR display system can not be used for a high speed display, so a specialist primitive will be written to control GAIA,[7,4,5] which is based on ESO's RTD widget.[2] As of v2.7 GAIA has the ability to display cubes in a movie mode and this technique will be used if it is fast enough. The fallback solution is to read the cube into memory in the pipeline and use a shared memory solution.

Off-line Off-line mode can be used to step through the images at a user-controlled rate, ranging from real-time (similar to on-line mode), where it takes as long to display the data as it took to acquire the data, to slow-motion rates where, say, each sample is displayed once a second. There will be controls for adjusting the display rate and stepping forward and back through the sequence. It will be possible for the Movie system to combine multiple data files into a single Movie. ***Req. DR6***

As for on-line mode, a specialist ORAC-DR recipe will be written to do the initial data processing from a set of raw files with optional binning parameters to keep the files within a reasonable size.

Once these cubes are created standard software such as GAIA and DS9 can be used to animate the cube.

The software for processing the raw data samples and combining the four sub-arrays for each wavelength into a single image will be the same in both the on-line and off-line modes.

4 "Quick Look" display

The key requirements for the Quick Look system (repeated here for clarity) are:

- The display should not fall behind observing. [***DR8***, ***DR9***]
- Each data frame should be displayed as quickly as possible to the observer (in a meaningful coordinate system). [***DR3***]
- A co-added image for the current observation should be displayed as quickly as possible. [***DR4***]
- Images of particular significance, e.g. those generated when an observation is complete and possible DARK frames, will be displayed on a separate window so that they remain visible for longer and so they can be examined interactively by the observer without fear of the image being rapidly replaced. [***DR20***]

There is an additional assumption that data displayed by the QL system does *not* need to be flux calibrated. This assumes that a single observation does not last for hours and cover a large range of opacity. (This is, effectively, an anti-requirement.)

Although the main Pipeline could be used for displaying QL images it is likely that the Pipeline would not be able to keep up with the observing rate if it is attempting to display frames as quickly as possibly whilst also combining groups (stacking and/or mosaicking) and performing simple analysis (calibration and source extraction). This also presupposes that QL data are made available to the pipeline at a sufficient rate, which is not the case for all modes. ***Req. DR1***

The simplest solution is therefore to run a separate instance of the Data Reduction Pipeline running _QUICK_LOOK variants of the normal Recipes; each Recipe supported by the Pipeline will also have a version optimized for QL. e.g. DREAM_DEEP_FIELD would have a corresponding DREAM_DEEP_FIELD_QUICK_LOOK. *It is expected that optimized variants of recipes will be treated as synonyms by the Quick Look system so that DREAM_DEEP_FIELD_QUICK_LOOK and DREAM_BRIGHT_POINT_SOURCE_QUICK_LOOK will actually be the same Recipe.*

Whilst for scan map mode the QL and science pipelines receive data at the same rate (when the file appears on disk) this is not the case for DREAM/STARE mode where image reconstructions are made available via DRAMA parameter at approximately 1 Hz whereas files will be written out at a rate of 1 every minute or so. This clearly demonstrates the need for a specialist QL pipeline.

The QL Pipeline initialisation script (`oracdr_scuba2_long_ql`) will initialise ORAC-DR such that the Recipe suffix will be set to the correct value. Additionally, the detection scheme used by ORAC-DR in QL mode will be the `task` looping option[13] to enable the data to be obtained directly from the DRAMA parameters. The DRAMA detection loop has the key advantage that if the pipeline gets behind the data acquisition for whatever reason, it never even notices the missing frames (beyond reporting a jump in the sequence number). ***Req. DR8***

GAIA[7,4,5] is the proposed display tool for Quick Look-generated images. This tool is well known, can be remotely controlled from ORAC-DR, allows interactivity (zoom/pan and also analysis), provides the ability to overlay catalogue data and is already used for a similar role at UKIRT. ***Req. DR22*** ***Req. DR21***

4.1 Quick Look Recipes

This section describes the data processing steps required to generate the data for the Quick Look system.

All the QL Recipes will calculate the noise of the detector elements during the observation and plot the results. ***Req. DR11***

4.1.1 Dark

Dark frames from the sub-arrays will be combined into a single mosaic and then will be displayed "as is" in focal plane coordinates. There will be no co-adding with previous darks.

4.1.2 DREAM and STARE

DREAM/STARE frames are made available by the acquisition system once every few seconds (maximum of 1 per second, the minimum rate is governed by sky rotation but could be every 3 or 4 seconds). The QL Recipe will strive to process the frames and co-add into the existing observation in that time period. ***Req. DR7***

DREAM/STARE frames are processed in two distinct steps. Firstly the frame is reduced on its own, and secondly it is combined into the current observation. When the observation is complete the final image will be displayed on a separate display.

The following steps are required to process the single frame:

1. If this is the first frame in the observation a reference RA/Dec image (1×1 pixels) must be created centred on the tangent point (the telescope tracking centre) aligned with the RA/Dec axes. This image acts as an alignment reference to force the images to be aligned with RA/Dec rather than focal plane coordinates.
2. Align the sub-array images to the reference frame (either the initial reference or the current coadd).
3. We can now create a single mosaiced image from the four sub-array.

Since DREAM/STARE images are tagged with World Coordinate Information by the DA system, this transformation can be done using existing tools (i.e. wcsalign from KAPPA[3] and makemos from CCDPACK). The single frame will then be combined with the current observation using makemos (care must be taken that the frames are wcsalign-ed to the same reference coordinate frame, else an additional resampling will be required). Throughput tests of this pipeline[9] indicate that this approach is acceptable even with for DREAM/STARE Quick Look. ***Req. DR3*** ***Req. DR4*** ***Req. DR2***

Variance information must be calculated from the data itself by looking at the variation between processed images. makemos can calculate variance but requires the mosaic to be built up from a number of frames rather than from a single frame (lacking variance) and the running coadd. This will require more processing power and so will not be implemented for the QL pipeline. It may be possible to extend makemos to store extensions in the intermediate

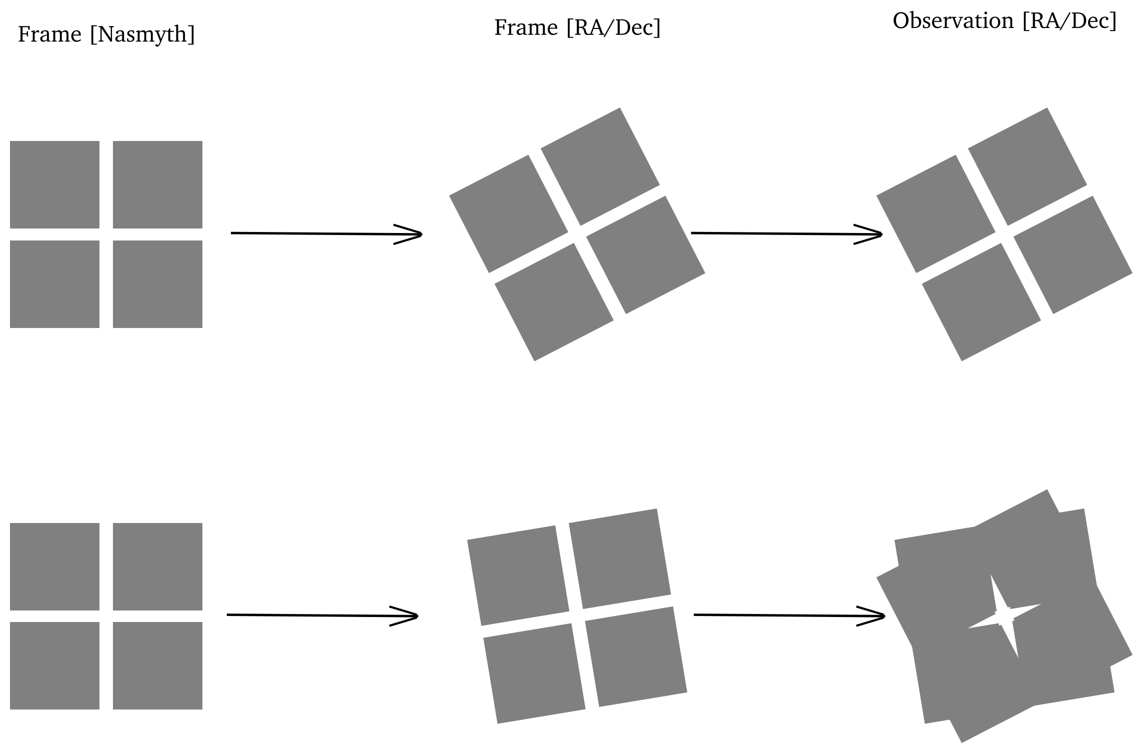

Figure 1: Diagram representing the QL progress for a DREAM/STARE observation. The first column indicates the single frames in focal plane coordinates. The middle column indicates each frame resampled to RA/Dec coordinates and the final column indicates the progress of the observation as data are taken. Only the Observation RA/Dec images are displayed.

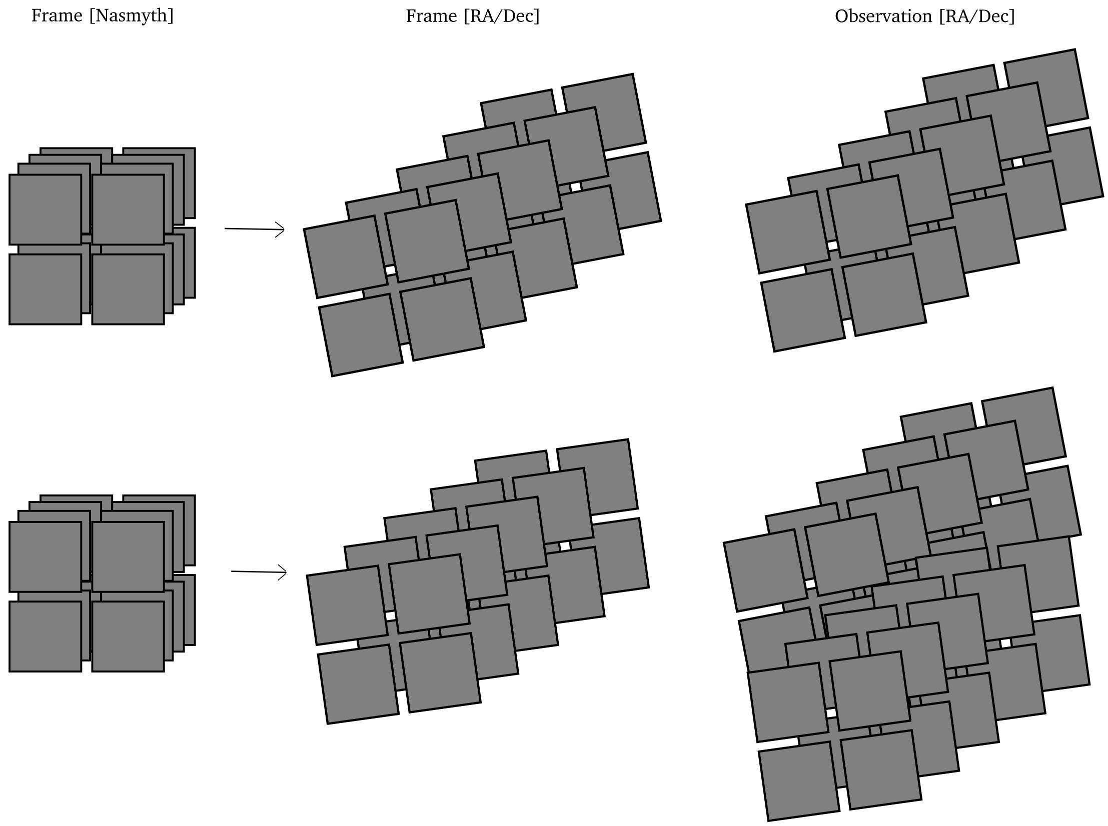

Figure 2: Diagram representing the QL progress for a SCAN observation. The first column indicates the single frames in focal plane coordinates. The middle column indicates each frame (scan) resampled to RA/Dec coordinates and the final column indicates the progress of the observation as data are taken. Only the Observation RA/Dec images are displayed.

image to allow the noise to be recalculated each time a frame is added. It may also be necessary to extend makemos to keep track of the number of data points that have contributed to a specific pixel (although the DA system does not include this information in the data files, so it would only contain the number of DREAM/STARE image pixels contributing to the final map). ***Req. PR11***

4.1.3 SCAN

Req. DR9
Req. MR3

Scan data must be processed from the raw 200 Hz data samples. The DA-system provides hints in the form of coefficients for a polynomial fit to the sky signal for each bolometer but this correction must be applied by the QL Recipe.

The basic steps required to prepare a single frame for display are:[17]

1. Apply the flat-field correction using the technique recommended from the array characterisation.
2. Apply the sky correction using the polynomial coefficients.[18]
3. Regrid into an image.

Step 3 will need to be written. This task will need to be SCUBA-2-specific since it will need to know how to read WCS information from the SCUBA-2 TCS extension for each sample. It will use a rebinning algorithm based on that used for SCUBA and implemented in the SURF rebin task.[16,19] The SURF implementation is not suitable directly because the SCUBA-2 data format is completely different from that used for SCUBA. The new task will have the following properties: ***Req. GR6***

- Mapping a pixel on the input image frame into a pixel in the output grid. The AST library[23,21,22,1] will be used for this transformation. A standard function will be available to request a 2-D AST FrameSet for a specific position in the time series.
- Writing the weights array and pixel count as extensions in the final output image (without applying the ring of virtual bolometers that force the edges of the map to zero flux; the so-called "guard ring"[19]). ***Req. PR11***
- The ability to read weights and pixel count data from input images in order to co-add new data to existing images without having to worry about edge effects (or beginning the combination from the original images as is currently the case with SURF). This will require that the routine can access the data as it looked prior to applying the "guard" ring correction.
- The ability to dynamically increase the size of the output image such that each pixel of the input image is represented in the output.

4.1.4 Set-up observations

It has not yet been decided how the set up observations will be displayed using Quick Look.

4.1.5 Polarimeter and FTS

Polarimeter and FTS Quick Look processing are discussed in separate documents.[14,10]

5 Pipeline Display

The SCUBA-2 Pipeline will use the standard ORAC-DR display interface. The ORAC-DR display system has the following properties:

- The Recipes only include an instruction to display some data, not how they should be displayed.
- The Pipeline user can control which data products they are interested in and how they should be displayed. This can be changed whilst the Pipeline is running.
- A single data product can be displayed in multiple ways without affecting the DR Recipe.
- It is not tied to any particular display engine.
- The display system can be turned off (e.g. during batch processing).

The display system works in the following manner: when the reduction primitive[12] creates some data product that it feels *could* be useful for display it sends the frame information (this can be a frame or a group object) to the display system. The display system then uses the last component of the file name (filenames use underscores as delimiters) as a key and searches for that key in the display configuration system. The configuration system includes information on how to process each file suffix. This includes the backend display engine to use, the type of display required (spectrum, image, contours etc.), and any additional parameters such as pixel bounds for a sub-region of the image or whether the display should autoscale. Furthermore, if the dimensionality of the file is too high for the requested format (e.g. displaying an image as a spectrum) the file is automatically converted to the correct dimensionality by averaging over the redundant axes. For each entry that matches the supplied key (e.g. `_mos` or `_reb`), the relevant display engine is contacted and the file is displayed in the required manner. The database can be edited at any time whilst ORAC-DR is executing, as it is re-read each time a file is displayed.

This approach allows the Pipeline to be effectively decoupled from the display system and relieves the Recipe writer from worrying how their products will be displayed. Consequently, for SCUBA-2 the display problem can be broken into two independent problems:

1. What data generated by the Pipeline should be displayed?
2. Is the display system as currently implemented able to display that data?

There is also no limit on the number of "panels" that can be controlled by the display system. For example, a final mosaic can be displayed in a 2nd image display window whilst the next observation intermediate images are being displayed in the first. ***Req. DR20***

The data generated by the Pipeline are discussed in the Recipes document.[12] The main data products are:

images

The primary data products of SCUBA-2 are images. Includes calibrated observations as well as co-added observations and mosaics. ***Req. DR4***

scatter plots

For skydips and plotting noise statistics.

polarization information ***Req. XR2***

Line segment plots and histograms of polarization angle and magnitude. ***Req. XR3***

spectral cubes ***Req. XR5***

From FTS observations.

spectra ***Req. XR5***

Slices from FTS spectral cubes.

All of these except the FTS products are already supported by the SCUBA ORAC-DR pipeline.[15]

Currently, ORAC-DR supports two display backends: GAIA and KAPPA. GAIA is used for interactive image displays (especially for important data products such as combined groups), allowing panning, zooming, catalogue overlay and simple analysis whilst the Pipeline is running, and KAPPA can support static image displays, contour plots, polarization plots, his- ***Req. DR21*** ***Req. DR22*** tograms, scatter plots and "data-model" (scatter plots overlaid on a model; used for skydip displays) plots. In the near future it is likely that support for SPLAT (the Starlink SPectral Line Analysis Tool)[6] will be added. SPLAT can be thought of as the spectral line equivalent of GAIA and can be used for interactive display of spectra (such as from the FTS).

These systems should be able to handle all the display requirements except the display of FTS cubes. Currently the only way to display "channel maps" from a cube is to do it within a Recipe by extracting slices and sending them to the display system one at a time. Also, in order to display spectra from selected sources in an FTS cube, they must be extracted in the Recipe prior to display.

References

[1] D. S. Berry. Providing Improved WCS Facilities Through the Starlink AST and NDF Libraries. In *ASP Conf. Ser. 238: Astronomical Data Analysis Software and Systems X*, pages 129–+, 2001.

[2] A. Brighton. *Real Time Display User Manual*. Very Large Telescope Software, European Southern Observatory, 1999. VLT-MAN-ESO-17240-0866, http://archive.eso.org/skycat/.

[3] Malcolm J. Currie and David S. Berry. KAPPA – Kernel Application Package. Starlink User Note 95, Starlink Project, CCLRC, 2002.

[4] A. C. Davenhall and P. W. Draper. The GAIA cookbook. Starlink Cookbook 17, Starlink Project, CCLRC, 2001.

[5] P. W. Draper. GAIA: Recent Developments. In *ASP Conf. Ser. 216: Astronomical Data Analysis Software and Systems IX*, pages 615–+, 2000.

[6] Peter W. Draper. SPLAT – A Spectral Analysis Tool. Starlink User Note 243, Starlink Project, CCLRC, 2003.

[7] Peter W. Draper, Norman Gray, and David S. Berry. GAIA - Graphical Astronomy and Image Analysis Tool. Starlink User Note 214, Starlink Project, CCLRC, 2003.

[8] Andy Gibb. SCUBA-2 StripChart Utility. SCUBA-2 Project SC2/SOF/S210/006, 2005.

[9] Andy Gibb. Throughput testing for DREAM or stare images. SCUBA-2 Project SC2/ANA/S210/003, 2005.

[10] Brad Gom. SCUBA-2 FTS data reduction engine. SCUBA-2 Project SC2/FTS/SOF/001.

[11] Tim Jenness. SCUBA-2 DA/DR interface control document. SCUBA-2 Project SC2/SOF/IC210/01, 2003.

[12] Tim Jenness. SCUBA-2 data reduction recipes and primitives. SCUBA-2 Project SC2/SOF/S210/004, 2003.

[13] Tim Jenness. SCUBA-2 pipeline architecture. SCUBA-2 Project SC2/SOF/S210/001, 2003.

[14] Tim Jenness. SCUBA-2 polarimetry data processing. SCUBA-2 Project SC2/SOF/S210/005, 2004.

[15] Tim Jenness and Frossie Economou. ORAC-DR – SCUBA pipeline data reduction. Starlink User Note 233, Starlink Project, CCLRC, 2001.

[16] Tim Jenness and John F. Lightfoot. SURF – SCUBA user reduction facility. Starlink User Note 216, Starlink Project, CCLRC, 2003.

[17] B. D. Kelly. SCUBA-2 modes and data processing. SCUBA-2 Project SC2/ANA/S100/028, 2001.

[18] B. D. Kelly. SCUBA-2 scanmap simulation. SCUBA-2 Project SC2/ANA/S100/038, 2002.

[19] Yan Lucas. A study of the SCUBA resampling software. Master's thesis, Ecole pour les Etudes et la Recherche en Informatique et Electronique and Royal Observatory, Edinburgh, 1993.

[20] Douglas Scott. SCUBA-2 data reduction software requirements. SCUBA-2 Project SC2/SRE/S210/001, 2004.

[21] R. F. Warren-Smith and D. S. Berry. World Coordinate Systems as Objects. In *ASP Conf. Ser. 145: Astronomical Data Analysis Software and Systems VII*, pages 41–+, 1998.

[22] R. F. Warren-Smith and D. S. Berry. Recent Developments to the AST Astrometry Library. In *ASP Conf. Ser. 216: Astronomical Data Analysis Software and Systems IX*, pages 506–+, 2000.

[23] 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.