Starlink Project
Starlink User Note 231.4

Tim Jenness, Frossie Economou
Joint Astronomy Centre, Hilo, Hawaii

June 2004

ORAC-DR – SCUBA Pipeline Data Reduction 4.1-0 User's Manual

Abstract

ORAC-DR is a flexible data reduction pipeline designed to reduce data from many different instruments. This document describes how to use the ORAC-DR pipeline to reduce data taken with the Submillimetre Common-User Bolometer Array (SCUBA) obtained from the James Clerk Maxwell Telescope.

Contents

1 Introduction

The ORAC Data Reduction pipeline (ORAC-DR) is a general purpose pipeline for reducing data from any instrument. A set of data reduction recipes are available for use with reducing data from the Submillimetre Common-User Bolometer Array (SCUBA) [1] at the James Clerk Maxwell Telescope[1], Mauna Kea, Hawaii . This document explains how to run the ORAC-DR data reduction pipeline system on data taken with SCUBA.

Information on the general aspects of ORAC-DR (with more information on loop control and the display system) can be found in document SUN/230.

2 Pipeline Setup

In order to configure the ORAC-DR SCUBA data reduction pipeline a setup command, `oracdr_scuba`, is provided. Assuming you are running the command from the Unix `TC-shell`[2] this command sets all the environment variables and command aliases required to run the pipeline. The UT date of the observations can be specified by supplying an argument to the command, if not specified the current UT date is assumed:

```
% oracdr_scuba 19990703
```

In order for the startup script to be configured to read the input data, an environment variable must be set which points to the directory containing the UT date directory. For example, if your data is in directory `/mydata/19991012` the following is required:

```
% setenv ORAC_DATA_ROOT /mydata
% oracdr_scuba 19991012
```

If `ORAC_DATA_ROOT` is not set then the current directory will be assumed unless the script is run at the Joint Astronomy Centre. With this variable unset when run at the Joint Astronomy Centre the input data directory will be configured to point to the actual SCUBA archive data directory.

When run, this command should list information detailing the current setup. Hopefully, it should be obvious at this point whether something has gone wrong. Here is an example for the JAC:

```
% oracdr_scuba 20000127

     ORAC Data Reduction Pipeline -- (ORAC-DR Version 1.0-0)
     Configured for instrument SCUBA
```

[1] http://www.eaobservatory.org/jcmt/

[2] ORAC-DR requires `tcsh` for full functionality from the system setup. If `tcsh` is not available the startup scripts (`oracdr_ufti` etc) will not set the specified UT date (the argument is ignored). In this case the `ORAC_DATA_IN` and `ORAC_DATA_OUT` environment variables must be set up explicitly. Additionally, the `oracdr` command must include the '**-ut**' option as specified in the `oracdr` documentation.

```
          Type "oracdr -h" for usage
          Type 'showme sun231' to browse the hypertext documentation


      Raw data will be read from /scuba/m99b/20000127/
      Reduced data will appear in /users/timj/oracdr/docs/sun231

     +++++++++ For online SCUBA reduction use oracdr -loop flag +++++++++

     For comments specific to SCUBA data reduction mail timj@jach.hawaii.edu
     For problems with the ORAC-DR system mail helpme@jach.hawaii.edu
              http://www.jach.hawaii.edu/UKIRT/software/oracdr/
```

A warning is printed if the input directory can not be found on the system.

Once this is done, the next step is to set the location of the output directory. By default, the directory from which the `oracdr_scuba` command was issued is chosen. In order to override this value, the `ORAC_DATA_OUT` environment variable must be set:

```
% setenv ORAC_DATA_OUT /largedisk/myoracata/
```

All ORAC-DR output files will appear in `ORAC_DATA_OUT`. For more information on the ORAC-DR environment variables please see SUN/230.

3 Running ORAC-DR

Once the environment has been configured the `oracdr` command will be available. Help on the available command line options can be listed by invoking `oracdr -h`. In its simplest form with no options, the pipeline will launch a logging window and start processing data from observation number 1 until no more data are available. For example[3]:

```
% oracdr
Orac says: No UT date supplied, using 19981001
ORAC says: Starting up monoliths...Done
Setting up display infrastructure (display tools will not
be started until necessary)...Done
No observation numbers supplied - starting from obs 1
Checking for next data file: 19981001_dem_0001.sdf....
```

Note that the default setting is that ORAC-DR will use the current UT date and start looking for observation number 1 in the `ORAC_DATA_IN` directory. It will wait for a flag file to appear until the timeout period expires (1 hour) or until the pipeline is aborted with CTRL-C. This behaviour is equivalent to running ORAC-DR with the following options:

[3] On an xterm that supports ANSI colour (e.g. `dtterm`) the output from ORAC-DR is colour coded depending on the source of the message

```
% oracdr -from 1 -loop flag
```

In many cases this is the correct behaviour at the telescope. In order to modify the behaviour of ORAC-DR command-line options can be used.

3.1 Selecting a UT date

The UT date is required so that the names of the raw data files can be derived via observation numbers. The **-ut** option can be used to specify the UT date of interest. For example,

```
% oracdr -ut 19980715
```

Currently, the pipeline can only process data from a single UT date in any single invocation. Data from multiple nights can not be coadded (even if they are in the same directory since the filename is derived from the UT)[4].

In general, the `oracdr_scuba` can be used to configure the UT date so that the **-ut** flag will not be required. Rerun `oracdr_scuba` when data from a different UT date are to be reduced.

3.2 Choosing the observations

In many cases only a subset of the data in `ORAC_DATA_IN` are to be processed. ORAC-DR provides a number of ways of specifying observations either as a range of observation numbers or as a list.

The options are:

-from

Specify the number of the first observation to be processed. This option defaults to '1' if this option is omitted but the **-to** is present.

-to

Specify the number of the last observation to be processed. If the **-from** option is present but no **-to** option, then all the data will be processed starting from **-from**.

-list

Specify a list of observations. This list should be comma-separated. Colons can be used to indicate a range. For example, **-list 1,2,3,5:10,15** would process observations 1,2,3,5,6,7,8,9,10 and 15.

-file

Specify a file containing names of files to be processed. This is useful for procesing data taken on different nights.

Here are some examples of selecting observations using ORAC-DR:

[4] This can be overcome by using soft links to rename the input files – see TJ for more information

```
% oracdr -from 5
```

Start at observation 5 and continue incrementing the observation number until no more files are available.

```
% oracdr -from 5 -to 20
% oracdr -list 5:20
```

Start at observation 5 and finish at observation 20.

```
% oracdr -to 20
```

Start at observation 1 and finish at observation 20.

```
% oracdr -list 1,2,3,4,5,20:25,30:32
```

Process observations 1,2,3,4,5,20,21,22,23,24,25,30,31 and 32.

```
% oracdr -file myfile.dat
```

Process observations lists in texttmyfile.dat.

3.3 Looping schemes

There are a number of different ways of dealing with the data detection loop in ORAC-DR. If the system is being used 'off-line', the data are all present in the input directory and the pipeline assumes that no new data will appear. In this case the *list* and *inf* detection loops are supplied which stop processing when data files can no longer be found. These are the default loops whenever observation numbers are specified with *list* being used in conjunction with the **-list** and **-from/-to** options and *inf* being used in conjunction with the **-from** option.

At the telescope new data are continually arriving so a different detection loop is required. The *wait* loop is used

Occasionally observation numbers are skipped (e.g. when an observation is aborted and not copied to the Sun). In this case the **-skip** option should be used. Without this option the data detection loop aborts when an observation can not be found (or it continues to wait for a file even though an observation with a higher number now exists). It is probably a good idea to always use the **-skip** option when processing SCUBA data.

4 Calibration

The calibration system can be configured using the **-cal** option. For SCUBA this option can be used to decide how to obtain the sky opacity, which gains to use and which bolometers should be turned off.

Jiggle map and photometry observations are automatically calibrated by the pipline (maps can be calibrated in Jy/beam or Jy/arcsec^2 by configuring the recipe).

The arguments to **-cal** should be comma-separated keyword=value pairs. The recognised keywords are:

- **gains**

  This keyword controls the way that gain values are determined. The options are:

  default: Use generic values for the gain (e.g. use 240 Jy/V at 850 microns). This is the default option.

  index: Derive gains by using the gains index file. The index file is updated every time a calibration observation is reduced (e.g. photometry on a planet). The nearest calibration (in time) will be used for the gain determination. An error will occur if no calibration observation has been taken (or reduced) before an observation is reduced.

- **tausys**

  This keyword controls the behaviour of the tau correction. The options are:

  CSO: Derive taus for all wavelengths by using the CSO tau stored in the header of each frame. This only works if the CSO tau is being updated.

  skydip: Derive taus by using the skydip index file. The index file is updated every time a skydip observation is reduced. The nearest skydip (in time) will be used for the extinction correction. If no skydip observation has been taken (or reduced) before an observation is reduced the CSO tau value will be used. Warnings are issued if the selected skydip was taken more than 3 hours from the current observation. **index** is an allowed synonym for **skydip**.

  a number: If a number is given it is assumed to be a CSO tau value. A value of 0.0 will turn off extinction correction.

  850skydip: Derive taus by using the 850 skydip values. Tau values from other filters are all derived from the 850 value using the standard tau ratios. If no suitable skydip can be found the CSO tau value will be used. Warnings are issued if the selected skydip was taken more than 3 hours from the current observation.

  dipinterp & 850dipinterp: Same as **skydip** and **850skydip** except values either side of the current observation are interpolated to find the current tau. This is *not* the same as specifying two tau values to the SURF extinction task since that will calculate the interpolated tau throughout the time of the observation rather than just calculating the value for the start. If only one value can be found then that value is used; if no values are found then CSO will be used. Warnings are issued if skydips were taken more than 3 hours from the current observation.

  csofit: Derive the CSO tau (and hence the tau for the specified filter) by using a polynomial fit to the CSO data for each night. This is the most accurate method of determining the opacity but is only available for nights between April 1997 and February 2001 (more fits will be provided as they become available). This method has the added advantage that photometrically unstable nights will not have a fit and therefore will not be processed (useful when processing large numbers of observations automatically).

- **badbols**

  This keyword controls the selection of bad bolometers (i.e. bolometers turned off by the pipeline). The options are:

index: Uses an index file containing bad bolometers. The index file is written by the SCUBA_NOISE recipe and contains a list of all bolometers that had noise (from REFLECTOR observations) greater than the specified threshold limit (currently the default threshold is 100 nV).

file: Uses a file badbol.lis found in ORAC_DATA_OUT. The file should contain a single line containing a space-separated list of bolometer names[5]. The file may contain a line such as:

```
A1 E2 H7 I5
```

This is the default system.

a list of bolometers: Finally, it is possible to specify a list of bolometer names. This list should be colon-separated, e.g.: 'H7:A2:b3'

Here are some examples:

```
% oracdr -cal tausys=skydip
```

Derive opacities from the index file but use the default gains.

```
% oracdr -cal gains=index,tausys=0.08
```

Use a constant value for the opacity and use the derived gains from the index file.

```
% oracdr -cal tausys=850dipinterp
```

Use the 850 micron skydips either side of an observation to derive all taus. Use the standard gain values.

```
% oracdr -cal badbols=a3:c14
```

Turn off bolometers a3 and c14.

5 Recipes

Data reduction recipes exist for processing data from the standard SCUBA observing modes. This does limit the flexibility of any given recipe since they are designed to work for any data from that mode. Occasionally it is necessary to modify recipes (e.g. change the sky bolometers or size of pixels to be used for the rebinning) and this can be achieved in a number of ways:

(1) Specify a new recipe name on the command line. This is fine for reducing observations taken in the same way but should not be used in the general case since this command line argument overrides all recipe choices regardless of observation mode.

(2) Create a recipe with a different name and store this name in the header before the observation is taken. This is done by using the DRRECIPE ODF parameter.

[5] Bolometer numbers can not be used since that depends on the sub-instrument in use

(3) Create a recipe with a different name and modify the DRRECIPE FITS header value using the KAPPA fitsmod command.

(4) Use the ORAC_RECIPE_DIR environment variable. This variable should be set before running up ORAC-DR and provides a search path that is used to locate recipes. When ORAC-DR attempts to read a recipe it first looks in ORAC_RECIPE_DIR, then in ORAC_DIR/recipes/SCUBA (the default location) and finally in the current directory (which will be ORAC_DATA_OUT).

In order to modify a recipe, it should be copied from the default location (ORAC_DIR/recipes/SCUBA) to ORAC_RECIPE_DIR and edited there. The next time ORAC-DR tries to read the recipe the modified version will be used in preference to the standard version.

The standard SCUBA recipes are:

SCUBA_NOISE for noise observations

SCUBA_STD_PHOTOM for photometry observations

SCUBA_JIGMAP for jiggle map reduction

SCUBA_POINTING for array pointing observations

SCUBA_EM2SCAN for SCAN/MAP data reduction using the Emerson II technique [2].

SCUBA_SKYDIP for skydip observations.

SCUBA_JIGPOLMAP for array polarimetry (jiggle maps)

SCUBA_EKHSCAN for scan/map data reduction using the EKH [3] technique

Null recipes are provided for ALIGN, and FOCUS observations.

6 Bad bolometers

In some cases data are affected by the presence of excessively noisy bolometers. To overcome this problem a facility is provided for turning off specific bolometers so that they are ignored by the data reduction.

See the section on calibration (§4) for more information on how to set bolometers to bad using the badbols calibration option.

7 Bad observations

In some cases, it is desirable to remove some frames (observations) from a group. For example, you make 5 observations of a source but you see that when the pipeline combines observation 3 into the group the noise is dominated by this observation. In order not to contaminate the group when observations 4 and 5 are coadded the pipeline must be instructed to remove observation 3 from any further group operations.

Observations can be turned off by using a special type of ORAC-DR index file (cf. index files generated by skydip and calibration observations).

This index file is called `index.badobs` and should be present in the `ORAC_DATA_OUT` directory. It should contain a line for each observation to be removed. Each line should contain the observation number and the 8 digit UT date (YYYYMMDD), e.g.:

```
55   19990128
62   19990129
```

8 Bad integrations

Currently it is not possible to turn off specific integrations using the pipeline.

9 Processing specific sub-instruments

By default all sub-instruments are processed by the pipeline. In some cases, e.g. where there is no hope of detecting anything at 450 microns, this is undesirable since this doubles the time required by the pipeline to process each observation. To overcome this the recipe can be edited to select specific sub-instruments as follows:

(1) Copy the recipe you are interested in to `ORAC_RECIPE_DIR` (create this directory and set the environment variable if necessary). For example, for scan maps copy SCUBA_EM2SCAN from `ORAC_DIR/recipes/SCUBA` to `ORAC_RECIPE_DIR`

(2) Change the `_EXTINCTION_CORRECT_` line to '`_EXTINCTION_CORRECT_ SUBS=long`' to only process the LONG sub-instrument. The SUBS argument can take a comma-separated list (e.g. P2000,LONG to select 2mm and LONG).

A modified recipe may look something like:

```
_PRE_PROCESS_
_FLAT_FIELD_
_SET_BAD_PIXELS_
_DESPIKE_SCAN_
_EXTINCTION_CORRECT_ SUBS=long
_REMOVE_SCAN_BASELINE_
_REMOVE_SKY_NOISE_SCAN_
_REBIN_FRAME_ PIXEL_SIZE=3.0 REBIN_METHOD=GAUSSIAN
_REBIN_EM2_GROUP_ PIXEL_SIZE=1.0 REBIN_METHOD=GAUSSIAN
```

10 The ORAC-DR display system

ORAC-DR uses a fully configurable display system. By default the data display is turned on but can be turned off by using the **-nodisplay** option when starting ORAC-DR. For a more general introduction to the display system see SUN/230.

The default configuration is to use KAPPA graphics commands via the KAPVIEW monolith, and uses a single GWM/GKS window split into sections. For mapping observations the individual reduced frames are displayed in the top two quadrants and reduced groups displayed in the lower quadrants (only one quadrant is used per sub-instrument). For skydips and photometry observations the display is split into two horizontal regions.

10.1 Display systems

The ORAC-DR display interface currently can use KAPPA and GAIA[6]. The GAIA interface can only support image display whereas the KAPPA (KAPVIEW) interface can support imaging, graphs, scatter plots and vector plots.

10.2 Display types

ORAC-DR can be configured to use the following display types:

IMAGE: Display an 2-D image file. The X,Y and Z limits can be specified or autoscaling can be used. Supported by GAIA, KAPVIEW and P4.

GRAPH: Display a 1-D data set as a line graph. Supported by KAPVIEW and P4

SIGMA: Display a data set as a scatter plot with a Y-range specified in sigmas and horizontal dashed lines at a specific sigma range (useful for photometry data - equivalent to the SURF routine qdraw) (KAPVIEW only)

DATAMODEL: Display a 1-D data set (as points) with a model (as a solid line). Designed for displaying skydip results. (KAPVIEW only)

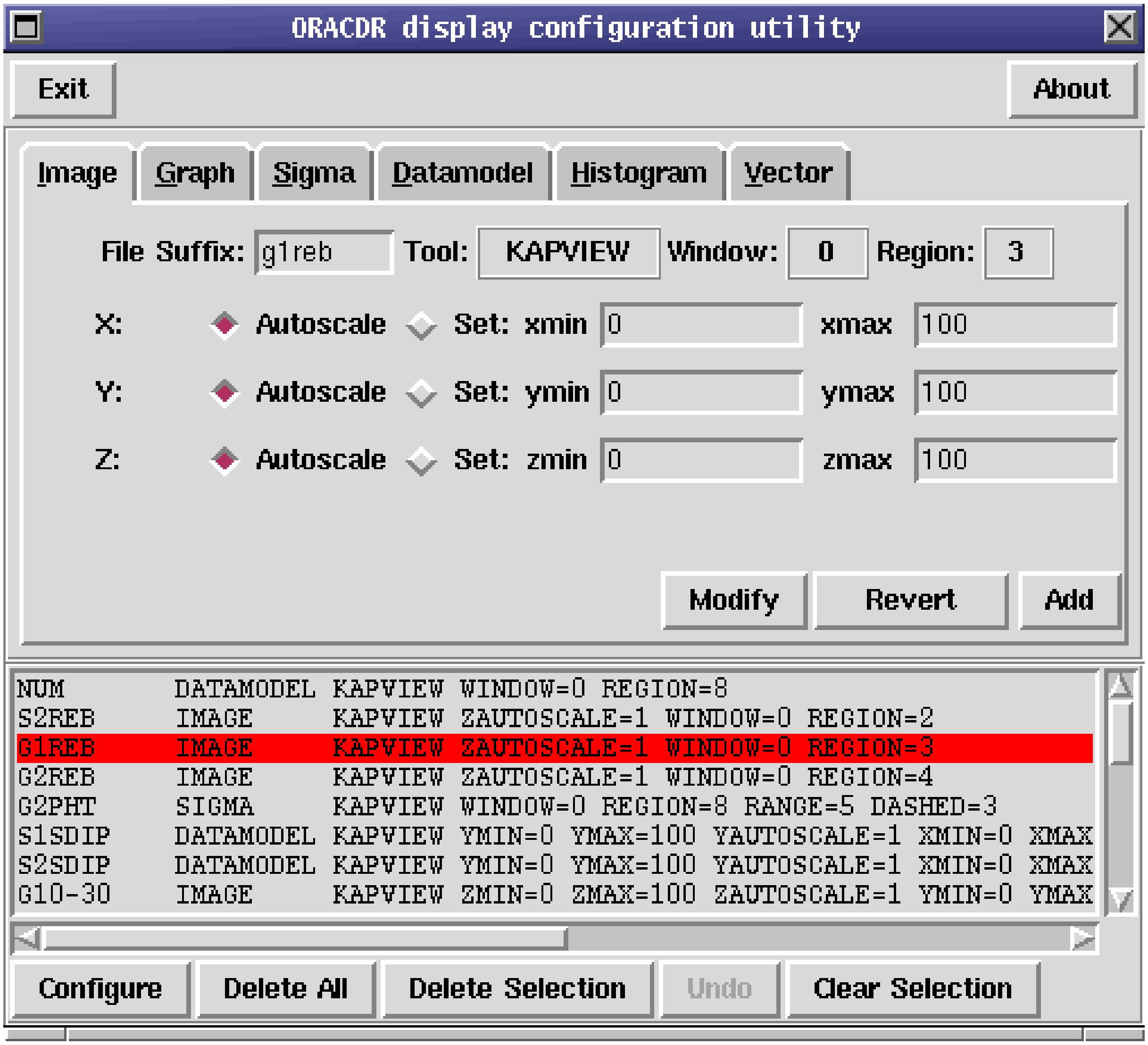
HISTOGRAM: Show a histogram of all data (KAPVIEW only)

VECTOR: Show vectors on top of an image (KAPVIEW only)

10.3 Configuring the ORAC-DR display system

The display is configured via the `oracdisp` tool and the `disp.dat` file found in `ORAC_DATA_OUT`. The `oracdisp` tool provides a graphical front-end to the display system and can be used to control where images are displayed, what type of display is used and how the data should be displayed. `oracdisp` runs independently of the pipeline, the only interaction between the pipeline and `oracdisp` is via a configuration file. Each time a primitive requests that a data file should be displayed the pipeline compares the graphics ID generated by the frame itself

[6] The P4 display engine is also supported, but its use is deprecated



File Suffix: g1reb Tool: KAPVIEW Window: 0 Region: 3

X: ● Autoscale ○ Set: xmin 0 xmax 100

Y: ● Autoscale ○ Set: ymin 0 ymax 100

Z: ● Autoscale ○ Set: zmin 0 zmax 100

```
NUM       DATAMODEL KAPVIEW WINDOW=0 REGION=8
S2REB     IMAGE     KAPVIEW ZAUTOSCALE=1 WINDOW=0 REGION=2
G1REB     IMAGE     KAPVIEW ZAUTOSCALE=1 WINDOW=0 REGION=3
G2REB     IMAGE     KAPVIEW ZAUTOSCALE=1 WINDOW=0 REGION=4
G2PHT     SIGMA     KAPVIEW WINDOW=0 REGION=8 RANGE=5 DASHED=3
S1SDIP    DATAMODEL KAPVIEW YMIN=0 YMAX=100 YAUTOSCALE=1 XMIN=0 XMAX
S2SDIP    DATAMODEL KAPVIEW YMIN=0 YMAX=100 YAUTOSCALE=1 XMIN=0 XMAX
G10-30    IMAGE     KAPVIEW ZMIN=0 ZMAX=100 ZAUTOSCALE=1 YMIN=0 YMAX
```

Figure 1: The ORACDISP display configuration tool.

(usually the last suffix) with the list of suffixes stored in the display configuration file. If they match, the configuration file is read (including information such as where to display it, the type of device and the bounds) and the corresponding display engine is instructed to display the data file using the supplied option. If multiple matches are made, then multiple display requests are processed. In this way a single display request from a primitive can be used to display multiple images (e.g. load an image into GAIA and display a slice using KAPPA).

The `oracdisp` tool is shown in figure 1. The tool is split into 3 major sections:

- A set of 'pages' corresponding to each display type. This is where the display details can be modified or details.
- A lower window containing the current display definition. Double-clicking on any line will select the entry and load the information into the top frame. This can be used to copy

Suffix	Type of image	Primitives
noise	Noise	_REDUCE_NOISE_
sdip	Skydip	_DISPLAY_SKYDIP_
pht	Photometrt data	_DISPLAY_PHOTOM_GROUP_
reb	Rebinned image	_REBIN_FRAME_
pol	Polarisation (I,P,THETA) image	_CALCULATE_POLARISATION_FRAME_
p	Polarisation P image	_CALCULATE_POLARISATION_FRAME_
theta	Polarisation THETA image	_CALCULATE_POLARISATION_FRAME_

Table 1: Suffix values used to display individual frames

an entry (e.g. by modifying the page details and pressing the 'Add' button) or modify the existing entry (pressing the 'Modify' button will change the highlighted entry - note that choosing a new page will reset the selection). A single click on an entry will select it for deletion (see next item).

- The lower frame contains buttons for saving and modifying the definition visible in the lower selection frame. The buttons do the following:

  Configure: This writes the current definition to disk (in `ORAC_DATA_OUT/disp.dat`). No backup is made of the original file.

  DeleteAll: Delete all entries in the selection frame

  DeleteSelection: Delete all selected entries (entries are selected by a single click - they are then highlighted in blue).

  Undo: Undo a deletion

  ClearSelection: Clear all selections.

The string that should be placed in the 'File Suffix' entry widget is discussed in the next section.

10.4 Displaying frame output

For SCUBA data, the products of early stages of data reduction (e.g. flatfielding or despiking) are not really suitable for display so many of the early primitives do not contain display directives.

Table 1 lists the suffices along with the primitives that generate the display request (and therefore must be called in the recipe).

Each of these suffixes can be prefixed by an 'sN' prefix where 's' stands for sub-instrument and N is a number indicating the position of the sub-instrument in the raw data (for SCUBA N is usefully between 1 and 3). For example, a display definition to display both the long and short rebinned images should contain definitions identified by 's1reb' and 's2reb' whereas 'reb' would be relevant if only 1 sub-instrument is being processed.

Suffix	Type of image	Primitives
gNreb	Rebinned image	_REBIN_GROUP_ & _REBIN_EM2_GROUP_
gNpa-thr	Rebinned dual-beam image	_REBIN_EM2_GROUP_
gNpht	Photometry results	_DISPLAY_PHOTOM_GROUP_

Table 2: Suffix values used to display individual groups.

10.5 Displaying group output

Table 2 lists the group suffixes recognised by the display system. In a similar way to frame definitions, all group id's are prefixed by 'gN' where 'g' indicates that a group is being displayed and 'N' is the sub-instrument number. Unlike for frames, the gN prefix is always attached regardless of the number of sub-instruments in the group.

The `_REBIN_EM2_GROUP_` primitive uses a gNpa-thr where 'pa' refers to the position angle of the chopped data and 'thr' to the chop throw. For example, to view the rebinned image of the data taken with 65 arcsec chop at 90 degrees position angle for the first sub-instrument a display id of 'g190-65' would be required.

11 Release Notes

11.1 V1.0

First release of ORAC-DR for SCUBA.

11.2 V1.1

- Add jiggle map calibration
- Add support for wideband and narrow band filters

References

[1] Holland W. S., Robson E.I., Gear W.K., Lightfoot J. F., Jenness T., Ivison R. J., Stevens J. A., Cunningham C. R., Ade P. A. R., Griffin M. J., Duncan W. D., Murphy J. A., Naylor D. A., 1999, *MNRAS*, **303**, 659 1

[2] Emerson D. T., Klein U., Haslam C. G. T., 1979, *ApJ*, **76**, 92 5

[3] Emerson D. T., *A.S.P. Conf. Ser* **75**, 309 5

A Prerequisites

The SCUBA ORAC-DR system requires the following Starlink packages in addition to those packages required to simply run the pipeline itself for a generic instrument:

- SURF
- KAPPA
- FLUXES

For polarimetry data reduction these additional packages are required:

- POLPACK
- CCDPACK
- CURSA

B Configuring the ORAC-DR environment

ORAC-DR uses environment variables to configure the global behaviour of the data reduction pipeline. Environment variables specify the input and output directories, the instrument type and the recipe/primitive locations. In general these variables are set by the start up script.

The following variables are used by the pipeline:

`ORAC_DIR`

This is the location of the ORAC-DR software. At the JAC this will be `/jcmt_sw/oracdr`. This variable is set as part of the Starlink login.

`ORAC_PERL5LIB`

This is the location of the ORAC perl libraries. It should be set to `${ORAC_DIR}/lib/perl5` This variable is set as part of the Starlink login.

`ORAC_INSTRUMENT`

This environment variable is used to inform ORAC-DR of the instrument that is to be used by the pipeline. This configures the primitive and recipe directories, file-naming conventions and other instrument specific options. In order to process SCUBA data this environment variable should be set to 'SCUBA'. This is set by the `oracdr_scuba` command.

`ORAC_DATA_CAL`

This variable should point to the directory containing the calibration information. In SCUBA's case these are the files that determine the rules that should be used for choosing skydip taus or gains. This is set by the `oracdr_scuba` command. At JAC this should be set to `/jcmt_sw/oracdr_cal/`.

`ORAC_DATA_IN`

This is the directory containing the raw data files. It is equivalent to the `DATADIR` environment variable used by the SURF software. Currently ORAC-DR must use raw data – partially processed data can not be pipelined. At the telescope this is usually set to `/jcmtarchive/YYYYMMDD/`. The SURF command scusetenv can be used to set this variable automatically (with no argument the current UT is assumed; an argument is assumed to be the requested UT date). This is set by the `oracdr_scuba` command.

`ORAC_DATA_OUT`

This is the directory used for writing output data. Data files are written to this directory rather than to the current directory. Reduced groups are stored in files of the form `YYYYMMDD_grp_NNNN_sub.sdf` where *NNNN* is the observation number and *sub* is the sub-instrument name (e.g. long, short…) This directory will contain the display definition file (`disp.dat`) and the skydip and gain index files. The `oracdr_scuba` command sets this value to the current directory. This directory should be on a local disk if at all possible.

C Running the pipeline at the JCMT

Running the pipeline at the JCMT is slightly different from running the pipeline off-line in Hilo or at another institution. The `oracdr_scuba` command should be used with no arguments. In this case the data are arriving now file by file and flag files are written when the data files are completely written to disk so that the `-loop flag` option should be used so pipeline will wait for new observation files to appear. This is the default if the pipeline is run without any observation numbers or loop specifiers. In order to restart the pipeline from a later observation, the combination of `-from NN -loop flag` is required.

Additionally, the fastest machine should be used for the reduction with a local disk. Ask your support scientist for more information on this – it is planned that the `oracdr_scuba` command will automatically locate the correct data disk and inform you of the optimal data reduction computer.