INTRODUCTION TO STARLINK

BASIC HETERODYNE AND SCUBA-2 WORKSHOP

OVERVIEW OF STARLINK

- Starlink has data analysis, data reduction, pipelines and visualisation software.
- Starlink software suite includes Starlink, Starjava and ORAC-DR/PICARD.
- Most useful packages for JCMT users: KAPPA, SMURF, CUPID, CONVERT, GAIA, TOPCAT, SPLAT, ORAC-DR and PICARD.
- See Starlink Cheat Sheet for very terse overview.

STARTING UP STARLINK: BASH/SH

```
export STARLINK_DIR=/path/to/your/starlink

source $STARLINK_DIR/etc/profile
```

TCSH/CSH

```
setenv STARLINK_DIR /path/to/your/starlink

source $STARLINK_DIR/etc/login

source $STARLINK_DIR/etc/cshrc
```

- Initialise packages by typing their name.
- Terminal will tell you how to access help on that package.

```
phlebas:~ sarah$ export STARLINK_DIR=~/star-2015B
phlebas:~ sarah$ source $STARLINK_DIR/etc/profile
phlebas:~ sarah$ kappa


      KAPPA commands are now available -- (Version 2.3-2)

      Type kaphelp for help on KAPPA commands.
      Type 'showme sun95' to browse the hypertext documentation.

      See the 'Release Notes' section of SUN/95 for details of the
      changes made for this release.


phlebas:~ sarah$
```

USEFUL TIPS AND TRICK FOR STARLINK

- CONVERT: initalise this package and you can then pass and output FITS files from Starlink commands. They will be automatically converted to/from NDF.
- run KAPPA's `fitslist` on NDF files to see telescope meta data.
- Type ? when prompted for a parameter and you can see the documentation of that parameter.
- GAIA is an extremely powerful GUI – try exploring its options!
- See the Starlink Cheat Sheet for many more!

FINDING HELP

- SCUBA-2 cookbook SC/21:
  http://www.starlink.ac.uk/docs/sc21.htx/sc21.html
- Heterodyne cookbook SC/20:
  http://www.starlink.ac.uk/docs/sc20.htx/sc20.html
- SUNs for each package: can use `findme` and `showme`
- http://www.starlink.ac.uk/docs/starlinksummary.html
- interactive help in command line:
  `kaphelp cupidhelp smurfhelp conhelp` etc.

SCRIPTING

Once you've mastered running Starlink commands manually, you may want to start writing scripts. Some help can be found:

- Shell scripts: C Shell cookbook SC/4, (adapt for bash) http://www.starlink.ac.uk/docs/sc4.htx/sc4.html
- Python wrapper: (early version) being developed http://www.eaobservatory.org/jcmt/science/reductionanalysis-tutorials
  (contact s.graves@eaobservatory.org for more information)

ORAC-DR/PICARD: DATA REDUCTION AND ANALYSIS PIPELINES

- Uses Starlink packages/commands as well as native Perl.
- Based around idea of specific recipes to do different tasks.
- ORAC-DR: Data reduction pipelines, takes raw data files.
- PICARD: Analysis pipeline, takes reduced data files
- Used at telescope for reductions at telescope and off-line reductions put into the JCMT Science Archive.
- Customisable by users via ‘recipe parameters’.
- SCUBA-2 - SUN/264 ; ACSIS - SUN/260; PICARD - SUN/265

HOW TO RUN ORAC-DR

1. Setup by typing: oracdr_<instrumentname>: e.g.

   `oracdr_scuba2_850` or `oracdr_acsis`

2. Set the input/output data directory with:

   ```
   export ORAC_DATA_IN=/path/to/data/dir
   export ORAC_DATA_OUT=/path/to/data/dir
   ```

3. Then run the pipeline:

   ```
   oracdr -loop file -file listofrawfiles.lis RECIPE_NAME
   ```

DATA ENVIRONMENTAL VARIABLES

- If you give the file names with their absolute path, it doesn't matter where the $ORAC_DATA_IN value is set.
- Get a list of files with absolute paths via e.g.:

```
ls `pwd`/myrawdatadir/s*.sdf > listoffiles.lis
```

- If you set set the ORAC_DATA_OUT to . it will always use the directory you are in when you run the command; e.g.:

```
export ORAC_DATA_OUT=.
```

ORAC-DR COMMAND LINE OPTIONS

- Customise ORAC-DR recipes via recipe-parameters.
- Write into a text file with ini format:

```
[REDUCE_SASSY]
MAKEMAP_CONFIG = dimmconfig_MJLSY01.lis
MAKEMAP_PIXSIZE = 4.0
```

(This passes a custom dimmconfig file to makemap, and sets the pixel size to 4”. It will only be applied when running the recipe REDUCE_SASSY)

- Pass to ORAC-DR on the command line via: `-recpars=myrecparfile.ini`
- See cookbooks and ORAC-DR SUNs to see available recpars.
- type ‘oracdr -man’ to see full documentation on options.

ORAC-DR LOGGING & DISPLAY OPTIONS

- Control logging with -log option:
  -log=x : show logging info in specia**l xwindow.**
  -log=f: write a log **file** to disk, named .oracdr<PID>
  -log=s: show logging in terminal **screen**
- Combine any log options, e.g.:
  -log=sf
- Display more logging information by including:
  -verbose
- DR recipes can bring up XWindows and GAIA displays as they go. To turn this off add:
  -nodisplay

PICARD

- Pipeline recipes for analysis of reduced data.
- See SUN/265 for full list of recipes: http://www.starlink.ac.uk/docs/sun265.htx/sun265.html
- Run similar to ORAC-DR, but give files on command line as final argument.
- Like ORAC-DR, customised via recipe parameter files.
- `picard <-log sf -verbose> RECIPE map1.sdf map2.sdf`
- If $ORAC_DATA_OUT is already set, PICARD will put output data in that location.