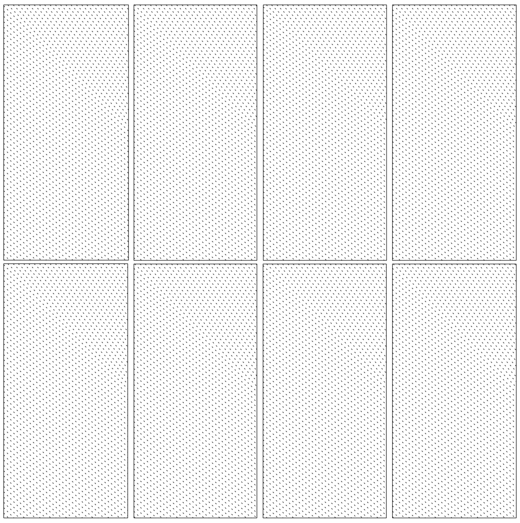
Wide-Field Imager at the MPG/ESO 2.2-m

Handling ESO WFI Data With IRAF



Heath Jones (ESO)
Frank Valdes (NOAO)

June 28, 2000

ESO Document Number: 2p2–MAN–ESO–22200–00002

Contents

1 INTRODUCTION

This guide is intended as a brief introduction to the reduction of European Southern Observatory (ESO) 2.2 m Wide-Field Imager (WFI) data within IRAF. The `mscred` package in IRAF is the main environment for reducing wide-field imager data and as such, contains many tasks for this purpose. However, it can not be used with ESO WFI data without some pre-processing of the images. Using `esowfi`, this pre-processing is possible such that all subsequent processing through `mscred` occurs in the normal way.

The `mscred` and `esowfi` packages were created and are supported by the IRAF Group at NOAO. In particular, the `esowfi` is provided as a service by NOAO to IRAF users with ESO WFI data . ESO is not responsible for the content of these packages nor for any adverse effects they may have on your data, whether used correctly or not.

The contents concentrate on how users can:

- Use `mscred/esowfi` on their data at the data reduction workstations at either the 2.2 m telescope or at ESO Santiago, **or**,
- install and use the `mscred/esowfi` packages on another machine where IRAF already resides.
- Use `esowfi` to modify raw ESO WFI images into a format useful to IRAF.
- Get started on further image processing of their images within IRAF.

More detailed information on fully-fledged WFI reduction is available in the *Guide to the NOAO Mosaic Handling Software* by F. Valdes and available at

`http://iraf.noao.edu/scripts/irafhelp?mscguide` .

This is the most comprehensive guide currently available on the subject of reducing wide-field mosaic data within IRAF. Our guide here is merely a supplement aimed at users with ESO WFI data. Comprehensive descriptions of the NOAO Mosaic Data Handling System are given in papers available at

`http://ecf.hq.eso.org/iraf/web/projects/ccdmosaic/` .

Alternatively, the ESO Imaging Survey Project (`http://www.eso.org/science/eis/`) which has made extensive use of the ESO WFI have provided some WFI mosaic reduction tips at

`http://www.eso.org/science/eis/eis_proj/pilot/wfi/` .

However, some of the scripts provided at this site are superseded by the more recent `esowfi` package described here. We recommend that you spend some time reading the information at all of these web sites.

Acknowledgment: Parts of this guide include information from the IRAF online help pages.

2 OVERALL SUMMARY

We begin with a short summary of the steps needed to use ESO WFI data in IRAF. Detailed descriptions of what is happening at each step are deferred to later sections.

```
> ximtool &
```

Starts the `ximtool` image display.

```
> cl
```

Starts IRAF running.

```
cl> cd /data
```

Changes to the working directory, (`/data` in this case).

```
cl> esowfi
      esohdr
```

Loads the `esowfi` package (and the `mscred` package automatically.

```
es> esohdr *
      wfi51377.fits:  bias
      wfi51378.fits:  bias
      wfi51381.fits:  Skyflat-B
      wfi51382.fits:  Skyflat-B
      wfi51390.fits:  Skyflat-V
      wfi51428.fits:  NGC4594-B
      wfi51429.fits:  NGC4594-B
```

Runs `esohdr` on all the FITS images in the working directory

```
es> set stdimg=imt2048
```

Sets the `ximtool` display buffer to 2048 ×2048 pixels. Doing so lets you preview the full mosaic quickly, albeit at a lesser resolution than the actual image.

```
es> mscdisplay wfi51431 1
 Amp:  Individual  Display (zcombine=none)
 1:   8215.7  12677.0     8215.7  12677.0
 2:  11039.0  16158.8    11039.0  16158.8
 3:   8341.0  10749.0     8341.0  10749.0
 4:   7173.9  13132.0     7173.9  13132.0
 8:   9187.5  11551.0     9187.5  11551.0
 7:  11177.0  13055.0    11177.0  13055.0
 6:  10493.0  13228.7    10493.0  13228.7
 5:   9869.8  12167.5     9869.8  12167.5
```

Displays an image in `ximtool`. If `esohdr` has worked it should show all CCDs in the mosaic in the correct position relative to each other. (An `mscdisplay` on a raw ESO WFI image would see the individual CCD frames display one on top of the other.)

```
es> ccdlist * extname=''im1''
```

Lists the parameters of all images.

```
 wfi51377.fits[im1][2142,4128][ushort][zero][1][U/98]:bias
 wfi51378.fits[im1][2142,4128][ushort][zero][1][U/98]:bias
 wfi51381.fits[im1][2142,4128][ushort][flat][1][B/99]:Skyflat-B
 wfi51382.fits[im1][2142,4128][ushort][flat][1][B/99]:Skyflat-B
 wfi51390.fits[im1][2142,4128][ushort][flat][1][V/89]:Skyflat-V
 wfi51428.fits[im1][2142,4128][ushort][object][1][B/99]:NGC4594-B
 wfi51429.fits[im1][2142,4128][ushort][object][1][B/99]:NGC4594-B
```

The last two steps (`mscdisplay` and `ccdlist`) are checks that `esohdr` has modified the headers of the ESO WFI images correctly. They are not really necessary unless you are trying `esowfi/esohdr` for the first time. With the completion processing by `esohdr`, ESO WFI images are ready for the general mosaic reduction procedures as outline in the *Guide to the NOAO Mosaic Handling Software*. Note also that it is only necessary to load the `esowfi` package. In doing so, `mscred` is loaded automatically. If `mscred` is loaded separately, you run the risk of running it with the wrong default parameters.

If you can not operate `esowfi/esohdr` in the way shown above, (or would like to know more about what it is doing to your images), read on.

3 GETTING STARTED

You may have some preliminaries to sort out before reducing your ESO WFI data, depending on where you are working.

3.1 On The 2.2-m Data Reduction Machine: `w2p2drs`

The data reduction machine at the ESO 2.2 m telescope is called `w2p2drs`. It has v2.11.3 of IRAF installed, including `mscred` and `esowfi`. Both packages are kept up-to-date.

The account name and password for `w2p2drs` are posted on the noticeboard in the 2.2 m telescope control room. IRAF can be started by typing `cl` in the home directory. **Please do not alter the `login.cl` or `loginuser.cl` files in this directory, or run the `mkiraf` command, without first consulting the 2p2 Telescope Team.**

Once IRAF has been started, change to the data directory by typing `cd /data`. From here, `esowfi` and `mscred` tasks can be run on images as they come off the telescope.

Now proceed to Section 4.

3.2 On Some Other Machine

Both `mscred` and `esowfi` are **external** packages which do not accompany the standard IRAF distribution. They need to be obtained and installed separately once IRAF is up and working. We do not deal with the subject of IRAF installation here, but refer the reader to

`http://ecf.hq.eso.org/iraf/web/` .

The installation of `mscred` and `esowfi` is as follows.

3.2.1 Installation of MSCRED

Installation of this external package consists of obtaining the files, creating a directory containing the package, compiling the executables or installing precompiled executables, and defining the environment to load and run the package.

1. In the following steps you will need to know the IRAF architecture identifier for your IRAF installation. This identifier is similar to the host operating system type. The identifiers are things like `ssun` for Solaris, `alpha` for Dec Alpha, and `linux` for most Linux systems. The IRAF architecture identifier is defined when you run IRAF. Start the `cl` and then type

```
cl> show arch
.ssun
```

   This is the value you need to know is without the leading '.'; i.e. the IRAF architecture is `ssun` in the above example.

2. **If you are installing the package for site use** login as IRAF and edit the IRAF file defining the packages.

```
$ set def irafhlib  # VMS example
% cd $hlib          # UNIX example
```

   Define the environment variables `mscred` and `mscdb` to be the pathnames to the mscred package root directory and the instrument database. The `$` character must be escaped in the VMS pathname and UNIX pathnames must be terminated with a tt/. Edit `extern.pkg` to include the following.

```
reset mscred = usr\$1:[mscred]      # VMS example
reset mscred = usr\$1:[mscdb]       # VMS example
reset mscred = /local/mscred/       # UNIX example
reset mscdb = /local/mscdb/         # UNIX example
task  mscred.pkg = mscred$mscred.cl
```

   Near the end of the `hlib$extern.pkg` file, update the definition of `helpdb` so it includes the mscred help database, copying the syntax already used in the string. Add this line before the line containing a closing quote:

```
,mscred$lib/helpdb.mip\
```

3. **If you are installing the package for personal use** define a host environment variable with the pathname of the directory where the package will be located (needed in order to build the package from the source code). Note that Unix pathnames must end with `/`. For example:

```
% setenv mscred /local/mscred/
```

   In your `login.cl` or `loginuser.cl` file make the following definitions somewhere before the "keep" statement.

```
reset mscred = /local/mscred/
reset mscdb = /local/mscdb/
task  mscred.pkg = mscred$mscred.cl
printf ("reset helpdb=%s,mscred$lib/helpdb.mip\nkeep\n",
    envget("helpdb")) | cl
flpr
```

   If you will be compiling the package, as opposed to installing a binary distribution, then you need to define various environment variables. The following is for Unix/`csh` which is the main supported environment.

```
# Example
% setenv iraf /iraf/iraf/             # Path to IRAF root (example)
% source $iraf/unix/hlib/irafuser.csh # Define rest of environment
% setenv IRAFARCH ssun                # IRAF architecture
```

where you need to supply the appropriate path to the IRAF installation root in the first step and the IRAF architecture identifier for your machine in the last step.

4. Login into IRAF. Create a directory to contain the package files and the instrument database files. These directory should be outside the standard IRAF directory tree.

```
cl> mkdir mscred$
cl> mkdir mscdb$
cl> cd mscred
```

5. The package and instrument database are distributed as tar archives for the sources and, as an optional convenience, a `tar` archive of the executables for select host computers. The instrument database distribution includes calibration files for the NOAO Mosaic Imager. For other instruments this file is not necessary.

6. For a source installation you now have to build the package executable(s). First you configure the package for the particular architecture.

```
cl> cd mscred
cl> mkpkg [arch]            # Substitute sparc, ssun, alpha, etc.
```

   This will change the bin link from `bin.generic` to `bin.[arch]`. The binary directory will be created if not present. If an error occurs in setting the architecture then you may need to add an entry to the file `mkpkg`. Just follow the examples in the file.

   To create the executables and move them to the binary directory

```
cl> mkpkg -p mscred -p tables # build executables
cl> mkpkg generic             # optionally restore generic setting
```

   Check for errors. If the executables are not moved to the binary directory then step [1] to define the path for the package was not done correctly. The last step restores the package to a generic configuration. This is not necessary if you will only have one architecture for the package.

   This should complete the installation. You can now load the package and begin testing and use.

3.2.2 Installation of ESOWFI

Installation of this external package consists of obtaining the files, creating a directory containing the package, and defining the environment to load and run the package. The package may be installed for a site or as a personal installation. If you need help with the installation contact `iraf@noao.edu` .

1. **If you are installing the package for site use** login as IRAF and edit the IRAF file defining the packages.

```
% cd $hlib
% vi extern.pkg
```

Add the following to the file.

```
reset esowfi = <path>/esowfi/
task  esowfi.pkg = esowfi$esowfi.cl
```

Near the end of the `hlib$extern.pkg` file, update the definition of `helpdb` so it includes the esowfi help database, copying the syntax already used in the string. Add this line before the line containing a closing quote:

```
,esowfi$lib/helpdb.mip\
```

2. **For personal use:** In your `login.cl` or `loginuser.cl` file make the following definitions somewhere before the `keep` statement.

```
reset esowfi = /mydir/esowfi/
task  esowfi.pkg = esowfi$esowfi.cl
printf ("reset helpdb=%s,esowfi$lib/helpdb.mip\nkeep\n",
    envget("helpdb")) | cl
flpr
```

3. Login into IRAF. Create a directory to contain the package files as defined above. This directory should be outside the standard IRAF directory tree.

```
cl> mkdir esowfi$
cl> cd esowfi
```

4. The package is distributed as a `tar` archive:

```
cl> ftp iraf.noao.edu (140.252.1.1)
login: anonymous
password: [your email address]
ftp> cd iraf/extern
ftp> get esowfi.readme
ftp> binary
ftp> get esowfi.tar.Z
ftp> quit
cl> !uncompress esowfi.tar
```

The `readme` file contains these instructions.

5. Extract the source files from the `tar` archive using `rtar`.

```
cl> softools
so> rtar -xrf esowfi.tar
so> bye
```

The `tar` file can be deleted once it has been successfully installed.

This should complete the installation. You can now load the package and begin testing and use.

4 FIRST PROCESSING – THE ESOWFI PACKAGE

We assume by now that you have both the mscred and esowfi packages installed within IRAF and have some data to process.

The esowfi package in IRAF contains one task: esohdr. This task is run on all ESO WFI images to translate the raw ESO header information into a format workable within IRAF. More specifically this includes compacting all the ESO HIERARCH keywords into single words, renaming the CCD extension names, setting the observation type, setting all the geometry keywords, and appending a first astrometric solution. All of this information is necessary (in the correct format) for successful use of all mscred features.

4.1 Running ESOHDR

To run, start IRAF running and load the esowfi package.

```
cl> esowfi
```

Now examine the parameters in the esohdr task:

```
es> epar esohdr
```

The parameter listing for this task is something like that shown below.

```
                              I R A F
                  Image Reduction and Analysis Facility
PACKAGE =  esowfi
   TASK =  esohdr

input   =                     @jnk5  List of raw ESO MEF files
(queryty=                       no)  Query for observation type?
(namps  =                        1)  Number of amps per CCD
(ntransi=                        2)  Number of bad transient columns
(wcsdb  =        esodb$wcs_feb00.db)  WCS database
(redo   =                       no)  Redo?
obstype =                    object  Observation type
(fd1    =                         )
(fd2    =                         )
(mode   =                       ql)
```

The parameters are as follows:

- The input parameter is a list of raw ESO multiextension (MEF) files to be modified. The task knows if a file has been previously modified so it is okay to include such files in a wildcard template.

- The `querytype` parameter controls whether the image types are input manually (at run time) or read from the header. This is because old ESO WFI headers did not contain an indication as to the type of observation (zero, dark, flat, or object), and this information is vital for efficient processing in IRAF. If this parameter is `yes` then a query will be made for the observation type. For more recent images from the telescope (post November 1999), `querytype` can be set to `no`.

- The `namps` and `ntransient` parameters give the number of amps per CCD and number of bad transient columns. For the latter, this number of columns following the prescan will be excluded from the `TRIMSEC` IRAF keyword. Note that `DATASEC` will still include the all CCD columns.

- The `wcsdb` parameter points to the database file containing the world coordinate system (WCS) information. This is used to provide the relative positions between each of the CCDs in the mosaic, and has been determined specifically for the ESO WFI mosaic. This information is inserted into the headers by `esohdr` and is necessary for the successful use of some `MSCRED` tasks. The default solution `esowfi$lib/wcs.db` is good for most images, taken both before and after the WFI dewar was taken apart and re-assembled in November 1999. However, you may choose to derive your own solution (see `http://iraf.noao.edu/projects/ccdmosaic/astrometry/astrom.html`).

- The `redo` parameter allows you to re-run a previously modifed image. If it is `no` then only files without the `ESOHDR` keyword will be operated upon. If it is `yes` then those files previously modified will be modified again. This skips the step which transforms the `HIERARCH` keywords since these were removed the first time. It also skips the `RA` keyword update.

- The `obstype` parameter used to query for the observation type if `querytype` is `yes`. This parameter only allows a selection from a set of choices which may be abbreviated.

When you execute `esohdr` (by typing `:g` from within the parameter listing), you will see execution output like the following:

```
wfi51377: bias
wfi51378: bias
wfi51381: Skyflat-B
wfi51382: Skyflat-B
wfi51390: Skyflat-V
wfi51391: Skyflat-V
wfi51428: NGC4594-B
wfi51429: NGC4594-B
wfi51431: NGC4594-B
```

Alternatively, you can run `esohdr` directly from the command-line, on all fits images in the current directory:

4.2 What ESOHDR Does

The task first checks for the keyword `esohdr` in the primary header to determine if the task has been run previously. Depending on the `redo` parameter, files containing the keyword will be ignored or the translation will be done again, (excluding the steps of renaming the HIERARCH keywords and converting the RA keyword from degrees to hours).

The HIERARCH convention keywords are converted to simple FITS keywords for access by IRAF and other software not supporting the HIERARCH format. The renaming consists of removing the first three words (HIERARCH, ESO, and the next word), concatenating the rest of the words, and truncating (if necessary) to 8 characters. For example,

"HIERARCH ESO INS FILT NAME" $\rightarrow$ "FILTNAME" .

IRAF supports an inheritance convention for merging the primary header keywords with the extension keywords. This means IRAF tasks that operate on extensions will see a complete set of keywords automatically. The esohdr task sets the INHERIT keyword to provide this feature.

Next the observation type is queried if desired. This is done because the current headers don't include information allowing the automatic identification of the observation type. The observation type is set in the primary header keyword FRAMTYPE.

Each extension is modified to provide the keywords used by MSCRED. The geometry of the readout — the size, direction, location of overscan and prescan, *etc.* — is determined from other information in the header. Also the extension names are modified to a shorter and more convenient form. The names are im$n$ where n is the CCD number (1-8).

In the last step the RA keyword, which is given in degrees in the raw ESO primary header, is converted to hours, which is what is expected by MSCRED. This is only done once. The RA and DEC keywords are then used to define the tangent point of the world coordinate system for the mosaic exposure. The other terms of the linear WCS are set from an astrometric solution specified in a database file. A default database file is provided.

For specific details on what esohdr does to the ESO header information, consult the online help available for the task:

```
cl> help esohdr
```

5 SUBSEQUENT PROCESSING – MSCRED

Wide field imaging data differ from those of single-chips in a few but important ways. These include

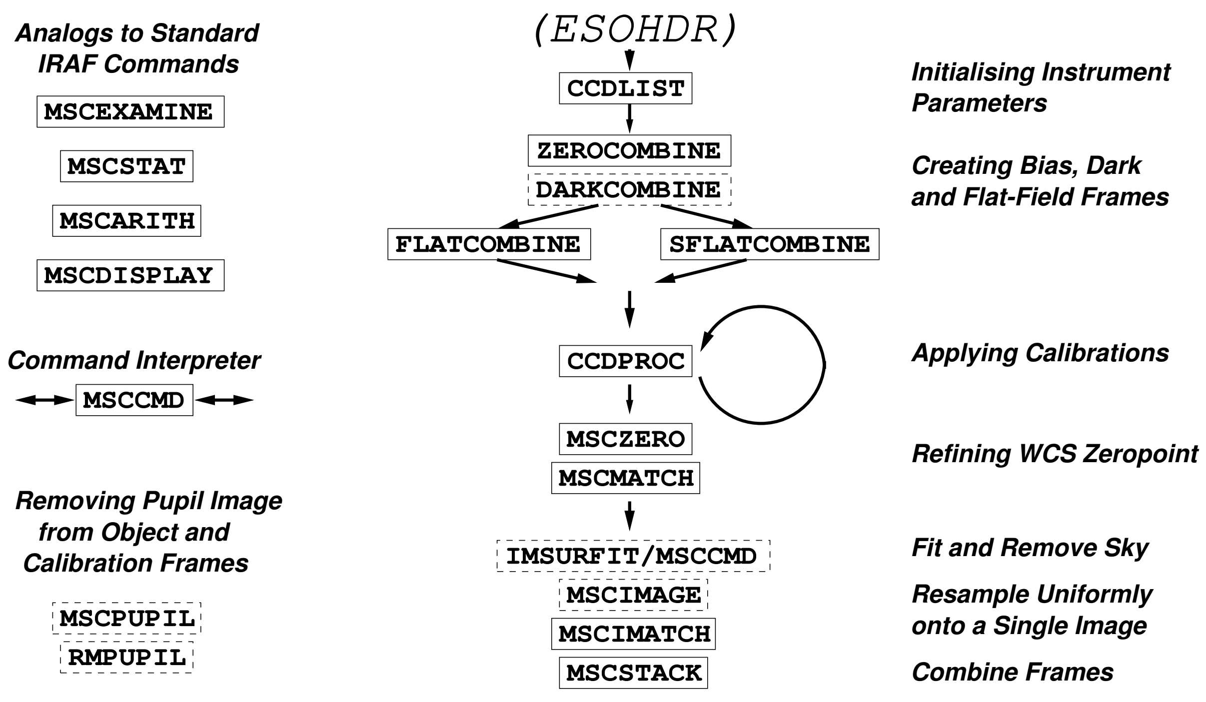
- the storage of data as multi-extension fits (MEF) files, with separate header information for each of the eight CCDs in the WFI mosaic,
- a variable pixel scale,
- the important positional relationship between mosaic CCDs, and
- the critical nature of background corrections for final images, free of gaps and other artefacts,
- the scaling of all CCDs in the same way for flat-fielding.

The `mscred` external package is used to reduce CCD mosaic data in which the data is in the mosaic MEF data format. The package was initially developed to support the NOAO Mosaic Imager but is generally useful for any mosaic data in the MEF image format. More specifically, when used in conjunction with `esowfi`, `mscred` works well with data from the ESO WFI. A current guide to `mscred` can be obtained at the IRAF command-line, by typing

```
es> help mscguide
```

5.1 An Overview of MSCRED

The `mscred` package contains a number of tasks that cover the operations on WFI data throughout the entire reduction chain, from raw to final image. The figure below lists some of the more commonly-used tasks and the relationships between them.



Notice that some `mscred` tasks such as `mscexamine`, `mscstat`, `mscarith` and `mscdisplay` are analogs to the common IRAF tasks `imexamine`, `imstat`, `imarith` and `display`. They function the same; the only difference being that the `msc-` commands can handle the MEF format of WFI images. Existing IRAF tasks can be used with the **individual** CCD frames of a full WFI image, although the extension of the particular frame must be specified. For example, if you have an `XIMTOOL` or `SAOIMAGE` window running, then

```
es> mscdisplay wfi51431 1
Amp:          Individual       Display (zcombine=none)
1:       8215.7  12677.0         8215.7  12677.0
2:      11039.0  16158.8        11039.0  16158.8
3:       8341.0  10749.0         8341.0  10749.0
4:       7173.9  13132.0         7173.9  13132.0
8:       9187.5  11551.0         9187.5  11551.0
7:      11177.0  13055.0        11177.0  13055.0
6:      10493.0  13228.7        10493.0  13228.7
5:       9869.8  12167.5         9869.8  12167.5
```

will display[1] all eight CCD frames side-by-side, in the same way as they image the sky. (This assumes they have already had `esohdr` run on them.) Using

```
es> display wfi51431[im2] 1
z1=7923.229 z2=16268.13
```

will display only **one** of the CCD frames (*i.e.* frame 2), while

```
es> display wfi51431 1
ERROR: FXF: must specify which FITS extension (wfi51431)
```

fails because the specific CCD frame needs to be specified. Other non-`mscred` IRAF tasks will fail in a similar way if forced to operate on a full WFI image.

To use a standard IRAF task on all the frames in a mosaic image, the special `mscred` task `msccmd` can be used as a command interpreter. For example, the `imhead` task can be used in conjunction with `msccmd` to list the headers of all eight CCDs,

```
    ms> msccmd "imhead $input long+" input=wfi51431
```

by issuing the entire `imhead` command as input to `msccmd`. To copy complete MEF files within IRAF, use the `copy` task (and not `imcopy`, unless you specify the image extension). However,

[1]**Image Displays:** You should reserve 8096 × 8096 pixels in the image display buffer if you want to view an entire mosaic image at full resolution. To do this, type `set stdimage=imt8192` before display. Such a large buffer size can make image display painfully slow, so usually sizes of `imt4096`, `imt2048` or `imt1024` are more convenient for quick checks not requiring full resolution. Typing `show stdimage` will display the current buffer size.

both `imrename` and `imdelete` can be used properly on MEF files, while `rename` and `delete` will work on individual extensions. The special `mscred` tasks `mscwfits` and `mscrfits` should be used in place of the usual `wfits` and `rfits`, which will fail on MEF data.

As shown in the figure, `mscred` carries tasks applicable to all stages of reduction of WFI data. The table below summarises the capabilities of all tasks in the package. While most are general-purpose, a couple are more specific, such as the `mscpupil` and `rmpupil` tasks for dealing with pupil image reflections from the NOAO 4 m telescope. These are not relevant to data from the ESO WFI. For detailed individual descriptions of all tasks we refer the reader to the *Guide to the NOAO Mosaic Handling Software*.

Summary of the Tasks Available in MSCRED

`mimpars`	Mosaic image parameters for display
`mscarith`	Image arithmetic mosaic exposures
`mscblkavg`	Block average mosaic exposures with header keyword updating
`msccmatch`	Match coordinates from list by adjusting WCS
`msccmd`	Execute general command with image extension expansion
`mscctran`	Celestial coordinate transformation using plate solution
`mscdisplay`	Display mosaic exposures in single frames
`mscexamine`	Examine mosaic exposures displayed as single frames
`mscfinder`	Package to do astrometry on mosaic data
`mscfocus`	Measure focus from mosaic focus exposures
`mscimage`	Reconstruct single images from a mosaic exposures
`mscimatch`	Match intensity scales in reconstructed mosaic images
`mscjoin`	Join separate images into MEF files
`mscotfflat`	Build on-the-fly flat field calibrations
`mscpixarea`	Compute and apply pixel area correction using WCS
`mscpupil`	Fit and remove pupil ghost from mosaic images
`mscrfits`	Read mosaic data from a FITS tape
`mscskysub`	Fit a sky surface and subtract all but the mean
`mscsplit`	Split MEF files into separate images
`mscstack`	Combine multiple reconstructed mosaic images
`mscstat`	Image statistics on mosaic image extensions
`msctvmark`	Mark coordinates from file on previous MSCDISPLAY
`mscwcs`	Set and adjust mosaic WCS
`mscwfits`	Write mosaic data to a FITS tape
`msczero`	Display, measure coordinates, set WCS zeropoint offsets
`rmpupil`	Remove pupil image, possibly interactively, by scaling
`ccdhedit`	Mosaic header editor
`ccdlist`	List mosaic processing information
`ccdproc`	Process mosaic exposures
`combine`	Combine mosaic exposures
`darkcombine`	Combine and process mosaic dark count exposures
`flatcombine`	Combine and process mosaic flat field exposures
`setinstrument`	Set instrument parameters
`sflatcombine`	Combine and process mosaic sky flat field exposures
`zerocombine`	Combine and process mosaic zero level exposures

5.2 Testing if ESOHDR Has Worked

There are a couple of simple checks you can make to see whether `esowfi` and the `esohdr` task have operated properly on your image headers. First, you can use `mscdisplay` to test whether the full mosaic data are displayed properly. For example, files on which `esohdr` has **not** been run will display all eight CCD frames one on top of the other.

A second test is whether a `ccdlist` of the images recognises all of the filter and image types. This step is a crucial one to pass if subsequent processing with `mscred` is to work. An example listing is shown below.

```
es> ccdlist images=@jnk5 extname="im1"

r40332[im1][2142,4128][ushort][zero][1][U/38]:bias
r40333[im1][2142,4128][ushort][object][1][BtLght-WFI(1)]:beta-light
r40334[im1][2142,4128][ushort][object][1][BtLght-WFI(1)]:beta-light
r40335[im1][2142,4128][ushort][zero][1][U/38]:bias
r40336[im1][2142,4128][ushort][zero][1][U/38]:bias
r40337[im1][2142,4128][ushort][zero][1][U/38]:bias
r40338[im1][2142,4128][ushort][zero][1][U/38]:bias
r40339[im1][2142,4128][ushort][flat][1][U/38]:dome flat
r40340[im1][2142,4128][ushort][flat][1][U/38]:dome flat
r40341[im1][2142,4128][ushort][flat][1][U/38]:dome flat
r40342[im1][2142,4128][ushort][flat][1][U/38]:dome flat
r40343[im1][2142,4128][ushort][flat][1][U/38]:dome flat
r40344[im1][2142,4128][ushort][flat][1][B/99]:dome flat
r40345[im1][2142,4128][ushort][flat][1][B/99]:dome flat
r40346[im1][2142,4128][ushort][flat][1][B/99]:dome flat
r40347[im1][2142,4128][ushort][flat][1][B/99]:dome flat
r40348[im1][2142,4128][ushort][flat][1][V/89]:dome flat
```

Here, the first column shows the **image extension** (`im1` in this case), the second the **pixel dimensions** of that CCD, the third the **data type**. Also listed are the **image** (`zero`, `object`, `flat`, *etc.*) and **filter** types (`U/38`, `B/99`, `V/89`, *etc.*). Part of what `esohdr` does is automatically provide the translations between the ESO keywords and the standard names that IRAF expects. In this way, the usual `setinstrument` task can be avoided altogether. Users can proceed directly to `ccdlist` after running `esohdr`.

If all the image types are listed correctly, then you are ready to proceed with the reduction proper (Sect. 6). If any columns are empty (*i.e.* showing `[ ]`), then something is amiss with either the translation file or the header contents, post-`esohdr`. Sometimes it happens that the `hlib$login.cl` file, (which is the default `login.cl` that is created when you run `mkiraf`), includes the following at the end:

```
noao            # optical astronomy packages

reset mscred = /iraf/extern/mscred/
#reset mscdb = /iraf/extern/mscdb/
```

```
        task  mscred.pkg = mscred$mscred.cl
        printf ("reset helpdb=%s,mscred$lib/helpdb.mip\nkeep\n",
            envget("helpdb")) | cl
        flpr
keep
```

This can cause the **image** and **filter** types to be missing from a `ccdlist`, since the part about `mscred` should not be there. The package gets defined in `hlib$extern.pkg`. These lines would only be used if the package is not installed in the system but just for personal use. But since `hlib$login.cl` is already a system file it does not make sense to have it here. What is happening in this case is that there are two versions of `mscred` and the `mscred.instrument` file is being changed in one but not the other. Note above that the last command before the mscred stuff is `noao`. That means that when the `task mscred.pkg` statement is executed, the `mscred` package becomes a subpackage of `noao`; *i.e.* `noao.mscred`. But the version defined in `hlib$extern.pkg` defines one at the root; *i.e.* `cl.mscred`.

There are two solutions. Someone with system permissions can remove the stuff between `noao` and `keep` in the `hlib$login.cl` file. You then need to do a new `mkiraf` to verify things will work for any user. Then do your `ccdlist` test. Alternatively, to just fix the problem yourself immediately you can just edit your own `login.cl` file to remove the lines.

The translation file (`esodb$esowfi.dat`) can be inspected by doing a `type esodb$esowfi.dat`. Otherwise, a full header listing (CCDs 1 through 8) can be obtained via

```
ms> msccmd "imhead $input long+" input=wfi51431 > header.dump
```

where the file `header.dump` contains the header information.

6 FINALLY

Once you have run `esohdr` on your images and can successfully perform simple `mscred` operations such as image display and `ccdlisting`, you are ready to reduce your data. For this we refer the reader to the *Guide to the NOAO Mosaic Handling Software* by F. Valdes. This describes most of the tasks in `mscred`, and the steps one might use in complete processing of wide-field imager data. The guide is available from

`http://iraf.noao.edu/scripts/irafhelp?mscguide` .

Users are encouraged to contact the IRAF Support Group at NOAO (`iraf@noao.edu`) for questions regarding the installation of `mscred` or `esowfi`. Regarding the use of `esowfi` and `mscred` with ESO WFI data, observers can contact Heath Jones (`hjones@eso.org`) of the 2p2 Team at La Silla.

7 FURTHER READING

The IRAF Support Group at NOAO maintain an extensive set of documentation on many aspects of IRAF. The guides below are some of the more useful ones for wide-field mosaic data and the reduction of imaging data generally.

Guide to the NOAO Mosaic Handling Software
by F. Valdes, (1998)
available at `http://iraf.noao.edu/scripts/irafhelp?mscguide`

The Astrometric Properties of the NOAO Mosaic Imager
by L. Davis, (1998)
available at `http://www.stsci.edu/stsci/meetings/adassVII/davisl.html`

A User's Guide to CCD Reductions with IRAF
by P. Massey, (1997)
available at `http://ecf.hq.eso.org/iraf/ftp/iraf/docs/ccduser3.ps.Z`

A User's Guide to Stellar CCD Photometry with IRAF
by P. Massey and L. Davis, (1992)
available at `http://ecf.hq.eso.org/iraf/ftp/iraf/docs/daophot2.ps.Z`

Photometry Using IRAF
by L. Wells, (1994)
available at `http://ecf.hq.eso.org/iraf/ftp/iraf/docs/photom.ps.Z`

Cleaning Images of Bad Pixels and Cosmic Rays Using IRAF
by L. Wells and D. Bell, (1994)
available at `http://ecf.hq.eso.org/iraf/ftp/iraf/docs/clean.ps.Z`

A Reference Guide to the IRAF/DAOPHOT Package
by L. Davis, (1994)
available at `http://ecf.hq.eso.org/iraf/ftp/iraf/docs/daorefman.ps.Z`

An Introductory User's Guide to IRAF Scripts
revised by R. Seaman, (1992)
available at `http://ecf.hq.eso.org/iraf/ftp/iraf/docs/sppguide.ps.Z`

A Beginner's Guide to Using IRAF
by J. Barnes, (1993)
available at `http://ecf.hq.eso.org/iraf/ftp/pub/beguide.ps.Z`