Contents

1 Introduction 1

2 Installation and Usage of MPIAPHOT 3
   2.1 Installation .................................................. 3
   2.2 Use ........................................................... 3
   2.3 The data file INSTRUMENT_new.PAR .................... 4
   2.4 The data file DETECTOR_new.PAR ....................... 4
   2.5 MIDAS login file ............................................ 6
   2.6 Installing MPIAPHOT on another machine ............. 7

3 General Rules and Conventions 9
   3.1 File names ................................................... 9

4 Data Reduction Procedures 11
   4.1 Data input INPUT/FITS ..................................... 11
   4.2 Checking the descriptors CHECK/DESCR ................ 11
   4.3 Standard pipeline processing PREP/CCD ................ 12
      4.3.1 Defaults for PREP/CCD ................................. 12
      4.3.2 Bias or DARK_0 subtraction ........................ 12
      4.3.3 Dark subtraction ..................................... 13
      4.3.4 Column mask .......................................... 13
      4.3.5 Fixed pattern noise ................................ 14
      4.3.6 Saturated pixels and data limit .................... 14
      4.3.7 What PREP/CCD does ................................ 15
      4.3.8 Error processionog .................................... 15
   4.4 Flat fielding ................................................ 15
      4.4.1 The procedure FLAT/AVERAGE ....................... 16
      4.4.2 Twilight flats ......................................... 17
      4.4.3 Fringe pattern ........................................ 18
      4.4.4 Rings due to nightsky lines in narrow-band imaging 18
      4.4.5 Scattered light ........................................ 18
      4.4.6 Applying the flatfields and clean-up ............ 19
   4.5 Removal of cosmic ray events ............................. 19
   4.6 Creating the sum frames .................................. 20
      4.6.1 Determining relative image orientation and shifts 20
      4.6.2 Summing the images .................................. 20
   4.7 Working with mosaics ...................................... 20
      4.7.1 Rebinning images into the gnomonic projection 20
      4.7.2 Correcting cosmic ray events ....................... 21
      4.7.3 Sum mosaic ............................................ 22
   4.8 Finding objects and creating the master list of objects 22
      4.8.1 Object detection ..................................... 22
      4.8.2 Master list ............................................ 23
   4.9 Photometry .................................................. 25
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.9.1</td>
<td>Separation stars / galaxies</td>
<td>25</td>
</tr>
<tr>
<td>4.9.2</td>
<td>Transformation of master coordinates to individual frames</td>
<td>25</td>
</tr>
<tr>
<td>4.9.3</td>
<td>Determination of the count rates for each object</td>
<td>26</td>
</tr>
<tr>
<td>4.9.4</td>
<td>Flux tables</td>
<td>27</td>
</tr>
<tr>
<td>4.10</td>
<td>Astrometry</td>
<td>28</td>
</tr>
<tr>
<td>5</td>
<td>Description of Commands</td>
<td>31</td>
</tr>
<tr>
<td>5.1</td>
<td>Context: main</td>
<td>32</td>
</tr>
<tr>
<td>5.2</td>
<td>Context: flux_cal</td>
<td>100</td>
</tr>
<tr>
<td>5.3</td>
<td>Context: classy</td>
<td>131</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Generating galaxy colour libraries for multi-colour classification</td>
<td>132</td>
</tr>
<tr>
<td>5.4</td>
<td>Context: synphot</td>
<td>156</td>
</tr>
<tr>
<td>5.5</td>
<td>Context: ametry</td>
<td>159</td>
</tr>
<tr>
<td>5.5.1</td>
<td>Astrometrie mittels DSS</td>
<td>160</td>
</tr>
<tr>
<td>6</td>
<td>WFI-specific commands</td>
<td>165</td>
</tr>
<tr>
<td>6.1</td>
<td>Outline of WFI data reduction</td>
<td>165</td>
</tr>
<tr>
<td>6.2</td>
<td>How to retrieve the LAMP, BETA images</td>
<td>166</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Elapse times for WFI data reduction</td>
<td>183</td>
</tr>
<tr>
<td>7</td>
<td>O2k-specific commands</td>
<td>185</td>
</tr>
<tr>
<td>7.1</td>
<td>Suggestion how to reduce OMEGA2000 data</td>
<td>185</td>
</tr>
<tr>
<td>8</td>
<td>LBC-specific commands</td>
<td>187</td>
</tr>
<tr>
<td>9</td>
<td>Glossary</td>
<td>191</td>
</tr>
<tr>
<td>A</td>
<td>Utilities</td>
<td>195</td>
</tr>
<tr>
<td>A.1</td>
<td>Frame list</td>
<td>195</td>
</tr>
<tr>
<td>A.2</td>
<td>File name conventions</td>
<td>196</td>
</tr>
<tr>
<td>A.2.1</td>
<td>HIROCS file name convention</td>
<td>196</td>
</tr>
<tr>
<td>A.3</td>
<td>Error treatment</td>
<td>197</td>
</tr>
<tr>
<td>B</td>
<td>Descriptors</td>
<td>199</td>
</tr>
<tr>
<td>C</td>
<td>Keywords</td>
<td>205</td>
</tr>
<tr>
<td>D</td>
<td>Astrometric transformation</td>
<td>207</td>
</tr>
<tr>
<td>E</td>
<td>Photometric data tables</td>
<td>209</td>
</tr>
<tr>
<td>E.1</td>
<td>Standard star tables</td>
<td>209</td>
</tr>
<tr>
<td>E.2</td>
<td>Epsilon tables</td>
<td>210</td>
</tr>
<tr>
<td>F</td>
<td>Internals</td>
<td>211</td>
</tr>
<tr>
<td>F.1</td>
<td>Kappa-sigma clipping</td>
<td>211</td>
</tr>
<tr>
<td>G</td>
<td>Cosmic ray removal on single frames</td>
<td>213</td>
</tr>
</tbody>
</table>
# List of Figures

1.1 Organization of the photo-area. Please note the area in red to which all paths are related.  

2.1 Calling `MPIAPHOT` ................................................................. 8

4.1 Principle of the flat/aver procedure. Pixel A has clear sky on all 4 images, whereas pixel B is affected by objects in 2 frames and therefore only 2 frames are entering the final image.  

4.2 Example of plots displayed by `CLEAN/SEX` to clean and prepare an object table for creation of master list .......................................................... 24

4.3 star/galaxy separation .................................................................. 25

4.4 datareduction flowchart .................................................................. 29

5.1 selection of normalization stars .................................................. 104

5.2 Astrometry flow chart ................................................................. 164

6.1 ESO archive LAMP,BETA .............................................................. 167
List of Tables

2.1 Parameter file INSTRUMENT new.PAR ............................................. 4
2.2 Parameter file DETECTOR new.PAR ............................................... 5

5.1 set/evaluate .................................................................................. 94
5.2 evaluate ......................................................................................... 96
5.3 Error messages from EVALUATE .................................................... 98

C.1 MPIAPHOT keywords ..................................................................... 205
Chapter 1

Introduction

As a start some references are given: The MIDAS manuals are found on the web [7]. A very good introduction to the basics of image processing is provided in "The Handbook of Astronomical Image Processing" [1]. The principles of "Imaging and Photometry" are described in the presentation given at the NEON summer school (www.mpia.de/homes/roeser, see also [11] pages 187ff). A brief introduction to the photometry in MPIAPHOT is given in our contribution to the Landoldt-Börnstein-serie [9]. The multi-colour classification is described in the PhD thesis of Christian Wolf and in [13, 14].

References to the FITS format are [1, 12, 5, 6, 8]. To view the header of a FITS-File on disk it is most convenient to create an xterm window with the command "xterm -132 &". Then one can use the UNIX command more to list the FITS-header nicely formatted with 80 characters per line.

The organization of the photo area is shown in Fig. 1.1, where the user has access to the tree entitled user. All paths are relative to user/exe, to which the environment variable PM has to point at (red box in organigram).
CHAPTER 1. INTRODUCTION

Figure 1.1: Organization of the photo-area. Please note the area in red to which all paths are related.
Chapter 2

Installation and Usage of \textit{MPIAPHOT}

2.1 Installation

We assume the reader is familiar with MIDAS terminology, disk organization and usage of MIDAS-command-files ("prg-files"). \textit{MPIAPHOT} uses several files in MID\_WORK. These are context files supported by \textit{MPIAPHOT}, and parameter files describing the instrument and the detector used in obtaining the data to be analyzed. The context files, a login-file and templates for the latter are copied into the \textit{current} directory, which should be the user’s MID\_WORK path using

\begin{verbatim}
csh -v $PM/.copyphot
\end{verbatim}

This command has to issued only once at the very beginning. It should be launched from the user’s MID\_WORK directory. See next paragraph for the definition of PM.

2.2 Use

Before \textit{MPIAPHOT} could be installed and used it is necessary to define the variable PM outside of MIDAS via

\begin{verbatim}
setenv PM /photo/user/exe
\end{verbatim}

most conveniently in the \texttt{.cshrc} file of the user. All other environment variable potentially used by \textit{MPIAPHOT} are to be defined relative to PM, like

\begin{verbatim}
setenv MID\_WORK ...(your own MIDAS directory)
setenv PHOTDATA $PM/.../tbl/PHOTDATA
setenv PHOTSTDS $PM/.../tbl/PHOTSTDS
setenv WFI $PM/WFI
setenv MCC $PM/MCC
setenv MCCTAB $MID\_WORK/MCC
setenv O2K\_UTIL $PM/../../o2k\_util
\end{verbatim}

Before starting \textit{MPIAPHOT} you have to rename the files INSTRUMENT\_new\_par and DETECTOR\_new\_par into INSTRUMENT\_new\_PAR and DETECTOR\_new\_PAR, respectively. This is a security measure so that \texttt{.copyphot} does not overwrite an existing configuration file. Also \texttt{user\_login.prg} has to be edited to adjust to the new user and renamed into \texttt{login.prg}.

\textit{MPIAPHOT} is activated within MIDAS by

\begin{verbatim}
set/cont mpiaphot
\end{verbatim}

The user is prompted for the detector and the instrument used (see \texttt{login.prg} on how to set defaults). These names refer to entries in the files DETECTOR\_new\_PAR (see \ref{sec:detector}) and INSTRUMENT\_new\_PAR (see \ref{sec:instrument}). \textit{MPIAPHOT} will use the data provided in the lines starting with these names to set keywords to be used in the data analysis. In Fig.\ref{fig:example_output} we give an example of what the output at the terminal should look like.
CHAPTER 2. INSTALLATION AND USAGE OF \textit{MPIAPHOT}

Table 2.1: Parameter file \texttt{INSTRUMENT\_new.PAR}

<table>
<thead>
<tr>
<th>Field</th>
<th>format</th>
<th>units</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instrument</td>
<td>A8</td>
<td></td>
<td>Code for the instrument, to be supplied, when \textit{MPIAPHOT} is called</td>
</tr>
<tr>
<td>Telescope ID</td>
<td>A8</td>
<td></td>
<td>Code for the instrument, to be supplied, when \textit{MPIAPHOT} is called</td>
</tr>
<tr>
<td>Longitude</td>
<td>F8.4</td>
<td>decimal degrees</td>
<td>geogr. longitude of telescope</td>
</tr>
<tr>
<td>Latitude</td>
<td>F8.4</td>
<td>decimal degrees</td>
<td>geogr. latitude of telescope</td>
</tr>
<tr>
<td>Altitude</td>
<td>F6.4</td>
<td>decimal meters</td>
<td>altitude of telescope above sea level</td>
</tr>
<tr>
<td>Diameter</td>
<td>F5.3</td>
<td>decimal meters</td>
<td>effective diameter of main mirror</td>
</tr>
<tr>
<td>P.A.</td>
<td>F7.3</td>
<td>decimal degrees</td>
<td>position angle of image Y-axis, if rotatable mount is at 0°</td>
</tr>
<tr>
<td>Scale X</td>
<td>F7.3</td>
<td></td>
<td>as given in next entry</td>
</tr>
<tr>
<td>Scale units Y</td>
<td>A32</td>
<td>mm(^{-1})</td>
<td>units for scale X and Y</td>
</tr>
<tr>
<td>Aperture</td>
<td>mm</td>
<td></td>
<td>dimensions of aperture stop in mm on the detector relative to its center</td>
</tr>
<tr>
<td>Xc</td>
<td>F6.2</td>
<td></td>
<td>center in X</td>
</tr>
<tr>
<td>Yc</td>
<td>F6.2</td>
<td></td>
<td>center in Y</td>
</tr>
<tr>
<td>Rx</td>
<td>F5.2</td>
<td></td>
<td>halfwidth in X or radius</td>
</tr>
<tr>
<td>Ry</td>
<td>F5.2</td>
<td></td>
<td>halfwidth in Y or 0 for circle</td>
</tr>
<tr>
<td>optical axis X</td>
<td>F8.2</td>
<td>pixel</td>
<td>position of optical axis</td>
</tr>
<tr>
<td>optical axis Y</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>distortion</td>
<td>1PE10.3</td>
<td>arcsec etc.</td>
<td>distortion coefficients (a_0 \ldots a_6)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>distortion = (a_0 + a_1 r + a_2 r^2 \ldots ) ((r) = distance to optical axis in arcsec)</td>
</tr>
</tbody>
</table>

Commands now activated may be listed by \texttt{$more \$PM/maincmd.prg.}

2.3 The data file \texttt{INSTRUMENT\_new.PAR}

The user has to supply the file \texttt{INSTRUMENT\_new.PAR} in MID\_WORK, a template of which is copied with the command \texttt{.copyphot} into \texttt{INSTRUMENT\_new.par}. It holds information about the instrument used to acquire the current data set. The template specifies also the format of the different columns. The table is in pure ASCII and is case and format sensitive; TABs are not allowed. An exclamation mark in the first column signals a comment and the line will be ignored upon input. Table[2.1] lists all the entries that are to be supplied before the \textit{MPIAPHOT} session is started.

2.4 The data file \texttt{DETECTOR\_new.PAR}

The user has also to supply the file \texttt{DETECTOR\_new.PAR} in MID\_WORK, a template of which is copied with the command \texttt{.copyphot} into \texttt{DETECTOR\_new.par}. It holds information about the detector used to acquire the current data set. The template specifies also the format of the different columns. Otherwise the same general remarks hold as for \texttt{INSTRUMENT\_new.PAR}. Table[2.2] again lists the relevant entries.
Table 2.2: Parameter file DETECTOR.new.par

<table>
<thead>
<tr>
<th>Field</th>
<th>format</th>
<th>units</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detector</td>
<td>A8</td>
<td></td>
<td>Code for the detector, to be supplied, when <em>MPIAPHOT</em> is called</td>
</tr>
<tr>
<td>Binning X</td>
<td>I1</td>
<td></td>
<td>binning factor for CCD-rows</td>
</tr>
<tr>
<td>Binning Y</td>
<td>I1</td>
<td></td>
<td>binning factor for CCD-columns</td>
</tr>
<tr>
<td>Gain</td>
<td>I3</td>
<td></td>
<td>CCD gain</td>
</tr>
<tr>
<td>n(X)</td>
<td>I4</td>
<td></td>
<td>number of pixels in X (columns)</td>
</tr>
<tr>
<td>n(Y)</td>
<td>I4</td>
<td></td>
<td>number of pixels in Y (rows)</td>
</tr>
<tr>
<td>EPC</td>
<td>F3.1</td>
<td>electrons/count</td>
<td>conversion factor <em>detected photons per digital unit</em></td>
</tr>
<tr>
<td>RON</td>
<td>F4.1</td>
<td>electrons</td>
<td>readout noise</td>
</tr>
<tr>
<td>DC-offset</td>
<td>F5.1</td>
<td>counts</td>
<td>level of the <em>DARK</em> 0 exposure (overscan)</td>
</tr>
<tr>
<td>Saturation</td>
<td>F6.0</td>
<td>counts</td>
<td>maximum useful level of pixel value</td>
</tr>
<tr>
<td>PIX-size X</td>
<td>F4.1</td>
<td>µm</td>
<td>pixelsize X</td>
</tr>
<tr>
<td>PIX-size Y</td>
<td>F4.1</td>
<td>µm</td>
<td>pixelsize Y</td>
</tr>
<tr>
<td>GOOD_PIX Xl</td>
<td>I4</td>
<td>pixels</td>
<td>pixel number of first useful pixel in a row</td>
</tr>
<tr>
<td>GOOD_PIX Yl</td>
<td>I4</td>
<td>pixels</td>
<td>pixel number of first useful row</td>
</tr>
<tr>
<td>GOOD_PIX Xu</td>
<td>I4</td>
<td>pixels</td>
<td>pixel number of last useful pixel in a row</td>
</tr>
<tr>
<td>GOOD_PIX Yu</td>
<td>I4</td>
<td>pixels</td>
<td>pixel number of last useful row</td>
</tr>
<tr>
<td>OVERSCAN Xl</td>
<td>I4</td>
<td>pixels</td>
<td>first pixel in row for overscan</td>
</tr>
<tr>
<td>OVERSCAN Yl</td>
<td>I4</td>
<td>pixels</td>
<td>first row for overscan</td>
</tr>
<tr>
<td>OVERSCAN Xu</td>
<td>I4</td>
<td>pixels</td>
<td>last pixel in row for overscan</td>
</tr>
<tr>
<td>OVERSCAN Yu</td>
<td>I4</td>
<td>pixels</td>
<td>last row for overscan</td>
</tr>
<tr>
<td>Bad Column 1</td>
<td>I4</td>
<td>pixels</td>
<td>serial number of 1. totally corrupted column</td>
</tr>
<tr>
<td>Bad Column 2</td>
<td>I4</td>
<td>pixels</td>
<td>serial number of 2. totally corrupted column</td>
</tr>
<tr>
<td>Bad Column 3</td>
<td>I4</td>
<td>pixels</td>
<td>serial number of 3. totally corrupted column</td>
</tr>
<tr>
<td>Bad Column 4</td>
<td>I4</td>
<td>pixels</td>
<td>serial number of 4. totally corrupted column</td>
</tr>
<tr>
<td>Bad Column 5</td>
<td>I4</td>
<td>pixels</td>
<td>serial number of 5. totally corrupted column</td>
</tr>
<tr>
<td>FR_AREA Xl</td>
<td>F6.1</td>
<td>world coordinates</td>
<td>Statistics area lower left corner X</td>
</tr>
<tr>
<td>FR_AREA Yl</td>
<td>F6.1</td>
<td>world coordinates</td>
<td>Statistics area lower left corner Y</td>
</tr>
<tr>
<td>FR_AREA Xu</td>
<td>F6.1</td>
<td>world coordinates</td>
<td>Statistics area upper left corner X</td>
</tr>
<tr>
<td>FR_AREA Yu</td>
<td>F6.1</td>
<td>world coordinates</td>
<td>Statistics area upper left corner Y</td>
</tr>
<tr>
<td>DET_INFO</td>
<td>A</td>
<td></td>
<td>optional detector information file</td>
</tr>
<tr>
<td>Reference</td>
<td>A</td>
<td></td>
<td>Reference for above entries (separated from previous entry by a !-sign)</td>
</tr>
</tbody>
</table>
2.5 MIDAS login file

You customize MIDAS at login with the login.prg file located in your MID_WORK area. The following gives an example of this file, which was used to develop MPIAPHOT. The obvious lines e.g. like the user name have to be edited. Please note especially the setup for the display and the graphics screens and also the debugger settings.

! LOGIN.PRG

define/par p1 512,512 N/A "Size of display 0 in pixels :"
define/par p2 1024,1024 N/A "Size of display 1 in pixels :"
define/par p3 1200,200 N/A "Size of display 2 in pixels :"

@@ PM:login
assign/print laps_w
monitpar(20) = 2000
mid$debug(1) = "/usr/bin/ddd -gdb -trace "
mid$debug(2) = " "
mid$debug(3) = " "
write/key user/c/1/20 "H.-J. Roeser"
write/key INSTR_ID/C/1/8 OMEGA2k
write/key DETECTOR/C/1/8 02k_77
set/midas commands=2000,1000 path=+$MID_WORK/HIROCS keyw=10000,10000

inquire/key inputc "Create GUI/HELP ? "
inutc = M$UPPER(inputc)
if inputc(1:1) .eq. "Y" THEN
    create/gui help
endif

inquire/key inputc "Create GRAPHIC window ? "
inutc = M$UPPER(inputc)
if inputc(1:1) .eq. "Y" THEN
    create/gra 0 1000,707,5,470 900000
endif

inquire/key inputc "Create DISPLAY window [STD or RGB] ? "
inutc = M$UPPER(inputc)
if inputc(1:3) .eq. "STD" THEN
    init/display p5=RGBQ
    create/disp 0 {p2},565,100 5 ? 9e5
    assign/displ D,0
    load/itt neg 0
    load/itt neg 1
    load/itt neg 2
    load/itt neg 3
    load/itt neg 4
    load/lut ramp
endif
if inputc(1:3) .eq. "RGB" THEN

2.6 Installing \textsc{mpiaphot} on another machine

Having established the directory structure as laid out in Fig. 1.1 and having copied all files into the respective directories the following changes have to be made before re-compiling everything:

1. Change the files \texttt{.make}, \texttt{.inslib}, \texttt{.replib} in \texttt{MAIN}, \texttt{LIB} and \texttt{SUB} to use the appropriate compilers on your machine.

2. Check the definitions in \texttt{pointers.inc}, \texttt{I*4} vs. \texttt{I*8}.

3. Check the parameters that govern the size of the flux tables in \texttt{flux.para.inc} in \texttt{MAIN}.

4. Check the buffer size in \texttt{mosaic.median} (parameter variable \texttt{maxpix}) and adjust it to the size of the memory (current value is okay for 8 Gbyte of memory).

Now re-compile everything with

- \texttt{.create_modules} in \texttt{MAIN}
- \texttt{.create_lib} in \texttt{LIB}
- \texttt{.create_sub} in \texttt{SUB}

That should be it.
Midas 001> set/cont mpiaphot

+MPIAPHOT package +
+ Version: Feb-07 +
+
++++++++++++++++++++++++++

... basic commands for CCD reduction activated

further contexts available: SPECIAL
FPI
POLAR
SYNPHOT

explain ? and explain CMD/qual give some help

Instrument used [WFI], give ? for list > WFI
Detector used [WFI], give ? for list > WFI
Binning [1,1] >
Gain [0=default] >
... DETECTOR found

-----------------------------------------------

Standard keywords for PhotoMetry package are set:

DETECTOR: WFI
Binn,Size: 0001,0001 8472,8259
CCD_parameter: 2.1, 4.7, 180.0, 50000.0, 20.0
(from K.M. / WFI manual)
Pixel size: 15.0, 15.0
Good area: 0001,0001:8472,8259 Overscan: 0000,0000:0000,0000
Bad columns: 0000 0000 0000 0000 0000
Frame Area for statistics: 3236.0,3130.0:5235.0,5129.0

Instrument: WFI at telescope: ESO2.2 (D = 1.9m)
P.A. 0.0 (degrees)
Scale in x,y: 15.834, 15.834 arcsec/mm arcsec/mm
Distortion: 0.000E+00*r**0, 6.153E-05*r**1, 1.621E-06*r**2, -4.447E-09*r**3
2.106E-12*r**4, 0.000E+00*r**5, 0.000E+00*r**6
Field [pix]: X,Y_center = 0000,0000 R_x,R_y = 0000,0000
Axis [pix]: X,Y_axis = 0.0,0.0 (nominal)
Observatory: 70.73, -29.25, 2300.0

-----------------------------------------------

... context level MPIAPHOT activated

Photo 002>

Figure 2.1: Example of the terminal output if *MPIAPHOT* is initialized by *set/cont mpiaphot*. 
Chapter 3

General Rules and Conventions

3.1 File names

File names must follow the rule in that the first 4 characters (yes, always 4) describe the data reduction process and are called the root of the file name. If want to you use less than 4 characters for the root, append underscores to fulfil this requirement.

Please note that MPIAPHOT preserves the characters 5ff of the file names.

Further file name conventions are specified in separate documents for each project (see page 196).
Chapter 4

Data Reduction Procedures

Reference to either MIDAS or MPIAPHOT-commands are set in typewriter font. Please always refer to the detailed description of these commands either in the MIDAS manuals (e.g. on the HIROCS internal web page) or in the detailed command description in this manual (page 32ff).

4.1 Data input INPUT/FITS

Please make yourself familiar with the FITS standard [12, 5, 6, 8]. This is currently the only way to read in files. Information on coordinate systems may be obtained from AIPS memos # 27, 46, as well as the documents ....

Usage of INPUT/FITS is described in detail on page 33. As it automatically does create the proper file names its usage is mandatory for CADIS data!

For HIROCS it is assumed that the data are on disk (e.g. /disk-h/HIROCSraw/...) as FITS files with arbitrary names to begin with. From this data depository files are read with command NAME/HIROCS (page 34), converted to MIDAS format and copied to your local directory with the proper HIROCS file name (see page ??).

4.2 Checking the descriptors CHECK/DESCR

Further processing within MPIAPHOT relies on the presence of certain descriptors (see Appendix). To help creating these in a simple way, the command CHECK/DESCR is provided. This procedure first reads the standard keywords, which describe the data set as defined in INSTRUMENT.PAR and DETECTOR.PAR. It then loops over all frames from the input list and searches, if the relevant descriptors are present. If not, it prompts the user for input. For some descriptors a certain clean-up is also performed (e.g. if ESO and DSAZ use slightly different keywords in their FITS-files to describe the same item).

As for calibration files not all descriptors are really required there is a quick mode installed, which is automatically entered, if the descriptor OBJECT starts with one of the following four letters: flat, dome, dusk, dawn, dark, ff. This mode can also be used for all frames in the input list by CHECK/QUICK, which should only be used for tests but never for the final reduction of science frames.

To avoid prompting, it is a good idea to write descriptors known to be missing and known to hold standard values before CHECK/DESCR is called using e.g. the procedure PROC/IMA (see page 195). This procedure is also strongly advised for the rare case, where it is known that certain descriptors are not correct (e.g. during early changes in the FITS keywords). It is also recommended to fill descriptor EPSILIST. For the latter the alternative is to fill the keyword EPSILIST with meaningful content (the empty keyword itself is created during initialization). Otherwise CHECK/DESCR will abort!

Some keywords/descriptors deserve a few comments.
CHAPTER 4. DATA REDUCTION PROCEDURES

ROT_POS  E.g. at the 3.5m-telescope a ROT_POS value of 999 signals that it was the prime focus. In this case, the descriptor is deleted.

FR_AREA  Many tasks use a clean window in the center of the detector to calculate the exposure level and noise (descriptor FR_STAT). This calculation is performed within the window specified by FR_AREA.

TEL_DIAM  The effective diameter (i.e. diameter of primary mirror corrected for vignetting due to secondary mirror) is used mainly in synthetic photometry.

All MPIAPHOT descriptors are listed on page 199. The entries for time and position on the sky are used to calculate the following descriptors: SUN, MOON, JUL_DATE, O_TIME, overwriting any previous entries in descriptor O_TIME!, AIRMASS (always re-calculated due to problem with airmass at 3.5m-telescope). Due to problems with MIDAS handling of descriptors for positions and time OBS_DATE may sometimes be off by one day!

Note: If you export your BDF file to FITS format and then re-import them again, descriptors for position and time may get lost and stored under the O_POS, O_TIME, O_AIRM!

Along the same lines, if you read e.g. descriptor AIRMASS in a FITS-file, you will get the error message, that this descriptor does not exist, although you will see it if you display the FITS header with UNIX more. The reason is that MIDAS internally converts the FITS-file to BDF, converting AIRMASS to O_AIRM before displaying the content of the descriptor to be read.

Previously WFI images were reduced with the WFI-pipeline made available by set/context WFI. The associated prg-files are located in path WFI. The Feb07 version of MPIAPHOT only needs the first two reduction steps from this: the bias subtraction together with the conversion to electrons (@@ WFI:wfi_debias) and the mosaicing (@@ WFI:wfi_mosaic_new). The latter is a simplified version which does no longer shear the images. All other commands from the WFI-pipeline for pure data reduction have become obsolete in version Feb07. For WFI images CHECK/DESCR should be called after the mosaicing.

Several utility commands specific to WFI are available and are described in a separate chapter on page 168.

4.3 Standard pipeline processing PREP/CCD

Raw CCD images suffer from a number of well understood and easy to correct “deficiencies” such as bias, dark current and pixel-to-pixel sensitivity variations. To correct these in one go the pipeline procedure PREP/CCD is provided, which in many cases already gives satisfactory results. For the syntax see the detailed command description. In what follows, the essential functions of PREP/CCD are described.

4.3.1 Defaults for PREP/CCD

Following the input parameters (see 5.1 page 41) the following keywords are read to provide default values: TOLWORLD, CCD_PARA, FR_AREA, INA_SIZE, GOOD_PIX, and INSTR_AP. The latter two are combined and set GOOD_PIX to their common area on the chip (in case of rectangular aperture) or INSTR_AP is used to blank pixels out side aperture (in case of circular aperture). A missing GOOD_PIX or if it is identical to 0,0,0,0 is equivalent to it being equal to 1,1,NPIX_X,NPIX_Y.

4.3.2 Bias or DARK_0 subtraction

Several different ways are provided to subtract the bias from a CCD frame. The relevant points are discussed in the following paragraphs.

Constant bias level

Specifying just a real constant will result in the subtraction of this value from every pixel. Default is 0.
4.3. STANDARD PIPELINE PROCESSING PREP/CCD

Bias from Overscan

Most CCD cameras read out more pixels in the X-direction than there are physical pixels present. This results in an area on the frame containing pure bias (or DARK0). As the bias level may show some small time variations, the use of the overscan area to calculate the actual bias value is probably the best way. Keyword OVERSCAN is used to locate the overscan area and the median in that area is used as the constant bias value subtracted from all pixels. To make this work, the descriptor CCD_PARA, especially the third entry specifying the bias level, must be roughly correct (within say 100 counts). This is because the routine, which evaluates the median must somehow know the window in counts where to calculate the median, which is done via a histogram.

**Not yet implemented:** The SITe detector at CAFOS does show an offset between a DARK0 exposure and the overscan of about 10 electrons (overscan is lower). To take this offset into account, one could specify it together with the OVERSCAN command by calling PREP/CCD as PREP/CCD .... over,offset ... . The offset will be added to the bias value as determined from the overscan.

Bias from a frame

A frame with zero exposure time (or average of many of these) has to be used if there is structure in the bias. The bias frame has to have descriptor STEP set to 1,1 for security reasons.

Bias from a frame with level adjusted via overscan

The worst case is a structured bias whose level is time dependent. Nevertheless good results can be obtained if the shape of the bias is taken from a dedicated (averaged) bias frame, whose level is adjusted in an additive way using the overscan of the frame to be corrected. This mode is activated by setting the descriptor MODE for the bias frame to ADJUST. This will result in the following action: For every frame to be corrected and for the bias frame, the bias level is taken from the overscan as specified by keyword OVERSCAN. The bias frame will then be adjusted by the difference of the overscan values before bias subtraction. The same remark concerning CCD_PARA as in Section 4.3.2 holds also here.

4.3.3 Dark subtraction

As in the case of the bias subtraction, the dark can be subtracted as a constant or as a file. Both cases refer to a dark exposure of 1 sec, to which the values (constant or frame) have to be normalized. As with the bias subtraction, the dark frame has to have descriptor STEP set to 1,1 for security reasons.

4.3.4 Column mask

Some CCDs show columns defect, which can be fully recovered, as the columns only show an offset, which is constant along the column but may vary from column to column. A one-dimensional column mask, which holds the offsets for each column can thus be subtracted from every row to correct for this effect. A complication arises for low illumination levels, as the then the mask depends on the specific exposure level in the pixel, i.e. the one-dimensional mask may be viewed as the saturation case. Tests have shown, that also in the low-illumination case a satisfactory result can be achieved, if the mask value $C_{offset}$ is calculated for each pixel according to

$$C_{offset} = C_{\infty} \times (1 - \exp[-C/C_{crit}]).$$

Obviously $C_{\infty}$ is the column-specific “saturation” value applicable for high illumination levels. The other constant $C_{crit}$ has to be determined for each column from a series of exposures with increasing illumination levels, which have to cover the range near zero to a level, where all column offsets have reached their saturation value $C_{\infty}$. The offset thus calculated has to be subtracted from the signal $C$ for every pixel.

---

2Hopelessly corrupted columns can be specified in keyword/descriptor BAD_COL. These columns will be replaced by interpolation along the rows.
The column mask for the "unsaturated" (general) case holds in the first row the values of \(C_\infty\) and in the second row the values for \(C_{\text{crit}}\) for each column. Again the descriptor \(\text{STEP}\) has to be 1 for the mask file. The default case is \text{NO\_MASK}.

### 4.3.5 Fixed pattern noise

All CCD detectors show pixel-to-pixel sensitivity variations (sometimes also referred to as fixed-pattern noise FPN), which are determined from flat field exposures. True sensitivity variations are a multiplicative correction and as such cannot be disentangled from vignetting/illumination effects. Distinguishing between the two is nevertheless meaningful and helpful in considering various ways to successful flat field construction.

Obviously both — pixel-to-pixel sensitivity variations and global sensitivity variations — are most easily corrected for by properly exposed twilight flat field exposures. For many cases this kind of flat field correction is already sufficient resulting in variations of the sky background across the detector of about 0.5%.

In the more general case, a true fixed-pattern-noise image is created \textit{e.g.} from internal or dome flats, which in general lack the proper homogenous illumination of the whole detector area. These flats are averaged and then a smooth model of the flat field image is created via \textit{e.g.} filtering with (\text{FILTER/MEDIAN} or \text{FILTER/GAUSS}). Using the model the fixed-pattern-noise image is normalized to 1 to preserve the image shape of the input image to be corrected. Please note: If all input images (twilight flats and science frames) are treated with this fixed-pattern-noise flat, the whole fixed-pattern-noise correction, especially the model, cancels out:

\[
\text{corr} = \text{raw}/(\text{fix/model})/(\text{skyflat}/(\text{fix/model}))
\]

It thus is not worth it to fine tune the normalization to its limits!

Such a separation of pixel-to-pixel sensitivity variations and the global flat field shape is of importance in cases, where the twilight flats do show stellar images. It may prove helpful to first correct for fixed-pattern-noise before \texttt{FLAT/AVERAGE} (see below) is used to build the twilight flats. If the fixed-pattern-noise is removed in a first step, faint stellar halos are much easier recognised by the averaging procedure (see \ref{S3}) and thus do not disturb the final flat field.

In \texttt{PREP/CCD} the fixed pattern noise is usually specified as a frame, but division by a constant is also supported. Again the calibration frame has to have descriptor \text{STEP} set to 1.1. Normalization of the fixed pattern frame is done via the descriptor \text{FR\_STAT} — if present —, in which case also the descriptors \text{FR\_AREA} and \text{LHCUTS(3),(4)} have to be present and valid.

### 4.3.6 Saturated pixels and data limit

The raw input image has a certain saturation level, either due to the response of the detector or due to the digital limit of (usually) 2 bytes of integer representation (65535). The keyword \texttt{CCD\_PARA(4)} holds this saturation level (in counts). Keyword \texttt{CCD\_PARA} is copied by \texttt{CHECK/DESCR} for each raw image into the frame-specific descriptor \texttt{CCD\_PARA}. Once arithmetic operations are performed on the image, the saturation value in the descriptor has to be changed accordingly (\texttt{MPIAPHOT} commands take care of this automatically). Furthermore, \textit{e.g.} flatfielding changes the saturation value across the field. Therefore we flag saturated pixels in the raw input image with a "magic" saturation value, which is the same during all reduction steps. The "magic" saturation value is stored in keyword \texttt{CCD\_PARA(5)} and is applied to the raw data in \texttt{PREP/CCD}, when also the conversion from counts to electrons is performed. At the same instance the descriptor \texttt{CCD\_PARA(5)} is adjusted. Thus even if no arithmetic operation is to be performed with \texttt{PREP/CCD} it should be called anyhow, just to perform the flagging and conversion to electrons.

To keep the information of the dynamic range of the data, we store the "magic" saturation for each image in descriptor \texttt{CCD\_PARA(5)} and keep the value of the upper usable range in descriptor \texttt{CCD\_PARA(4)}, the latter being adjusted to all arithmetic operations performed by \texttt{MPIAPHOT}.

Pixels with no meaningful value (\textit{e.g.} in the gaps of a mosaic or outside a circular aperture) are set the the data-limit value, as specified in keyword \texttt{DATA\_LIMIT}. This is done \textit{e.g.} in \texttt{PREP/CCD} for CAFOS images (circular aperture) or with @\texttt{WFI:mosaic} for the gaps in WFI. The descriptor \texttt{DATA\_LIMIT} is not written by \texttt{CHECK/DESCR} as it requires the user to first mask the meaningless pixels by \textit{e.g.} a bad-pixel-mask. Following this the descriptor has to be written by hand!
4.4 FLAT FIELDING

4.3.7 What PREP/CCD does

The first check is for the descriptor STEP to see if it is really 1,1, which is taken as an indication, that we are really dealing with a raw frame. Only for this is the pipeline processing really meaningful. Then the descriptor BAD_COL is searched. If it is not found, obviously no bad columns are corrected. Next the descriptor EXP_TIME is looked for, which is the only non-standard descriptor desperately needed by PREP/CCD. If CCD_PARA is not attached to the frame, the keyword is taken instead. The same is true for the descriptor FR_AREA.

The procedure will convert counts to electrons and adjust descriptor CCD_PARA accordingly to its actions. The cuts will be calculated based on the statistics calculation in the FR_AREA window and set to $\text{median} + 10\sigma$.

All corrections as specified by the various inputs (bias and dark subtraction, bad column restoration and fixed-pattern-noise correction) are performed in one go and keeping the correction frames in memory for processing of all input image. The routine is thus optimized for CPU and I/O usage. All calculations are only performed for the image area as specified by keyword GOOD_PIX and/or INSTR_AP. The resulting frame will only cover this area. Pixels in the image but outside INSTR_AP are set to zero.

4.3.8 Error processiong

Error conditions during the pipeline loop do in most cases not terminate the procedure but only skip the current file. If automatic deleting of input files is active, the input file is not deleted if any error is detected during its processing. See Appendix page 197 for details on error treatment.

Attention: There is currently a feature in PREP/CCD as well as in CORRECT in the sense that the flat field frames are not allowed to be smaller in any dimension than the desired output image. If this happens you should create an artificial frame with the dimension of the largest frame to be dealt with, whose pixels are set to the value of FR_STAT(1) of the flat field frame. Using INSERT/IMAGE the flatfield frame is then inserted into this artificial image and all non-standard descriptors are copied from the flat field to it with

COPY/DD in *,3 out

Then all should be fine.

4.4 Flat fielding

There is no unequivocally accepted pre-defined way of flat field correction. We will therefore discuss here the most elaborate procedure and from there steps for simplification can be undertaken as allowed by the data set under consideration. We assume that all science images taken in a given filter have been processed by PREP/CCD, i.e. they are bias/dark subtracted and corrected for pixel-to-pixel sensitivity variations. These images still suffer from a variety of flat field defects:

- global sensitivity variations
  These are due to e.g. variable thickness of the CCD structure. As such they affect both object and background signal and they have to be corrected in a multiplicative way using ideal twilight sky flat fields.

- interference fringe pattern
  This pattern results from interference in the thin CCD structure and due to this nature it is highly wavelength and bandwidth dependent. As a consequence, the fringe pattern seen in the images is mainly due to the many narrow nightsky emission lines and does not affect the object signal itself. It is an additive flat field feature and the fringe pattern has to be scaled to each individual image and subtracted from the science frame (see p. 18).

- rings due to nightsky lines
  In the case of imaging through a narrow interference filter or a Fabry-Pérot etalon the wavelength of transmission is a function of the distance to the optical axis. Nightsky lines included in the bandpass therefore produce rings of emission in the background, which have to be modeled (to suppress noise) and then subtracted (see p. 18).
• scattered light
At least in the case of Fabry-Pérot imaging with CAFOB and imaging with MOSCA we know that scattered light is introduced into the instrument. Using the twilight flat straightforward will thus ruin the photometric accuracy of the corrected image.
In the case of the etalon special frames taken through a mask with many holes are used to measure the positional variation of the etalon transmission. This pattern is modeled and used to adjust the flat fields in the pre-filter to be used as etalon (multiplicative) flat fields. Thus the scattered light contribution is isolated and can be subtracted (see p. [15]).
The treatment of the scattered light in MOSCA is currently under investigation and its description will be added later.

4.4.1 The procedure FLAT/AVERAGE
A general problem encountered during flat fielding is that one has to create a clean flat field image out of images disturbed by stars (twilight), objects (fringe pattern) or cosmic ray events (any longer exposure). A convenient way to remove these “artefacts” is provided by FLAT/AVERAGE. Here a batch of images is median filtered pixelwise, i.e. perpendicular to the X-Y-plane. If the images are aligned by pixel coordinates and the telescope did not always point to the same position in the sky (i.e. the objects do not always fall onto the same pixel coordinates, image dithering), the clear sky background can be extracted from this filtering, as stars, cosmics etc. will be removed in the median process.

Image normalization  For the median filtering to work we need, of course, a correct normalization of the input images. FLAT/AVERAGE uses the descriptor FR\_NORM for this. If it is not present, it is calculated for the area as specified by descriptor FR\_AREA. FF\_NORM is similar to FR\_STAT but whereas the latter is of general use, FF\_NORM is only used by FLAT/AVERAGE. Normalization is defined in such a way that

\[
\text{FF\_NORM}(1)_{\text{result}} = \sum_k \text{FF\_NORM}(1)_k
\]

This way the result is representative of the sum of all input images.
A most essential aspect of the functioning of FLAT/AVERAGE is the noise analysis, which in the end determines, which pixels (i.e. which frames) are included in the resulting output flat field image. The scatter \(\sigma_{ij}\) in the normalized pixel values, which is stored in the \(\text{rms}\)-image created together with the flatfield by FLAT/AVERAGE has a two-fold application. It
4.4. FLAT FIELDING

- determines, which pixels are excluded via a $\kappa\sigma$-clipping algorithm and it
- provides the weight $w_{ij} = 1/\sigma^2_{ij}$ (via the $\text{rms}$-image) with which each frame is included in the sum for pixel $(i, j)$, if mode `P' is chosen (see page 14) to combine subsets of the images into flatfields, which are combined by a subsequent call of $\text{FLAT/AVERAGE}$ into a final flat field.

You have a choice from several different methods to determine the way how the uncertainty $\sigma_{ijk}$ is determined for each individual frame $k$ (see command description on page pagerefc:FLAT/AVER for more details):

$\mathbf{T}$ theoretical value: $\sigma^2_{ijk}[\text{cts}] = RON^2_{\text{counts}} + P_{ijk}[\text{cts}]/EPC$, where the values of $RON$ and $EPC$ are taken from descriptor or key $\text{CCD}$_{$\text{PARA}$}.

$\mathbf{X}$ mean value: $\sigma^2_{ijk} = A + B \times P_{ijk}$, where the constants $A$ and $B$ are determined automatically (via an internal call to $\text{STATISTIC/BACKGROUND}$) according to counting statistics (which requires some dynamical range in the image stack!) and stored in descriptor $\text{FR}$_{$\text{NORM}(2)$}, $(3)$. If the dynamic range is insufficient, the read-out-noise $RON$ is taken from $\text{CCD}$_{$\text{PARA}$} and the rest is determined from these.

direct input of $A$ and $B$ as $A, B$

$\mathbf{P}$ the values $\sigma_{ij}$ are taken from the corresponding $\text{rms}$-frame (same name, extension .RMS instead of .BDF) for each input image.

Therefore, averages of $\text{FLAT/AVERAGE}$ results should always be created via `P'.

The smaller the amplitude of the artefact in a given image the harder it is to detect against the noise. To improve the detection e.g. of faint halos around stars the selection of valid pixels can be performed twice internally by the $\text{FLAT/AVERAGE}$ procedure, depending on the value of the input parameter $N_{\text{smooth}}$. The $\kappa\sigma$-check is performed first on the original frame and then on a smoothed frame. Only pixels which pass the acceptance test for both cases do enter into the final averaged image. The smoothing parameter $N_{\text{smooth}}$, which governs this option, specifies the diameter of the box around each pixel in which a modified median is calculated to replace that pixel value. The $N$ entries to the histogram calculated in this box are truncated at the upper and lower end by $N/3$ values and the average is calculated for this truncated histogram. This average has the advantage over a conventional median, that it avoids the uncertainties arising in the median for small $N$.

In detail $\text{FLAT/AVERAGE}$ does the following: For every image $k$ the normalized pixel value $p_{ijk}$ is calculated and the median value $m_{ij}$ determined. Please note that only pixels with values in the interval $[\text{LHCUTS}(5), \text{LHCUTS}(6)]$ are taken into account! These have to be set by the user. If $\text{LHCUTS}(5), (6)$ are not present $\text{lcuts}(3), (4)$ (=default) are used. Next the $\kappa\sigma$-clipping is done by calculating $\delta_{ij} = |(p_{ijk} - m_{ij})/\sigma_{ijk}|$ and comparing $\delta$ with the value of $\kappa$ specified as input. For pixels $i, j, k$ with $\delta \geq \kappa$ the weighted average $M_{ij} = \langle p_{ijk} \rangle_k$ is calculated, with the weights $w_{ijk} = 1/\sigma^2_{ijk}$. If $N_{\text{smooth}} \geq 3$ is specified a second iteration is performed on the smoothed image with pixel values $s_{ijk}$ before the median value $M_{ij}$ is calculated. For the second iteration the clipping boundary is reduced to $\pi = \kappa/2.5$ for $N_{\text{smooth}} = 3$ or $\pi = \kappa/4.0$ for $N_{\text{smooth}} > 4$. So in the end only pixel values enter the final image, which satisfy both criteria $p_{ijk} - M_{ij} < \kappa \times \sigma_{ijk}$ and $s_{ijk} - M_{ij} < \pi \times \sigma_{ijk}$.

4.4.2 Twilight flats

The first step in flat fielding is the creation of filter specific twilight flats using $\text{FLAT/AVERAGE}$. As the command $\text{CORRECT/IMAGE}$ cannot be called twice for the same frame, one has to use $\text{PREP/CCD}$ a second time to correct for the multiplicative flatfield with the twilight flat. In this case, of course, no correction for bias, dark and mask is specified, as this was already corrected for in the first call.

If the dithering amplitude was not large enough to have brighter stars clearly separated on the detector in several images, it may be best to take different subsets of the twilight flats and run $\text{FLAT/AVERAGE}$ on them to create flat fields, which are afterwards combine with $\text{FLAT/AVERAGE}$ in the noise-mode $P$. 

4.4.3 Fringe pattern

After the multiplicative correction with the twilight flats the images may show a pronounced fringe pattern (especially with the old RCA and LORAL detectors at red and near-infrared wavelengths). As discussed above, the fringes only affect the sky background and thus are to be subtracted from the image. For this purpose, a fringe pattern frame is created with FLAT/AVERAGE from the (longer exposed) science frame in each filter, if possible from images of different fields and/or dithered images, removing all objects and cosmic ray events. The output image is the fringe pattern, whose level corresponds to the sum of the input images used in the call to FLAT/AVERAGE. The actual subtraction of the fringe pattern is done in CORRECT, which scales the fringe pattern to the background level of the individual science frame for the actual subtraction using the descriptors FR_STAT.

It is important to pay attention to the gain in S/N obtained by subtraction of the fringe pattern. It should not be applied by default! In many cases where fringing is weak and/or only few images are available in a given filter it is not advisable to perform the fringe correction as the result will have an inferior S/N as compared to the uncorrected image. Local background determination during the actual photometry with EVALUATE will later on then have to take care of the uncorrected additive flat field contribution.

There may be a severe complication to the above procedure, if the continuum background night sky level does change appreciably (e.g. due to the presence of moon light) or the night sky lines do vary in intensity, what they sometimes do. In this case the normalization via FR_STAT will fail and the removal of the fringes will not be satisfactorily. To deal with this problem each frame may be equipped with the optional descriptor FLAT_BKG, which gives the amount of additional continuum background level, which is subtracted from all pixels of a frame by FLAT/AVERAGE before it takes its actions. This correction is also applied in the normalization process by CORRECT. Of course, this procedure should also work for variable night sky emission lines. We admit, however, that there exists little experience with FLAT_BKG and that in the extreme, each frame has to be treated individually to determine the necessary value of FLAT_BKG.

4.4.4 Rings due to nightsky lines in narrow-band imaging

The wavelength of the peak of the transmission curve $\lambda_{\text{max}}$ of a narrowband interference filter or a Fabry-Pérot etalon is a function of the radial distance to the optical axis:

$$\lambda_{\text{max}} = \lambda_{\text{center}} \sqrt{1 - (r/f)^2/n_{\text{eff}}^2}$$

In the case of the etalon $n_{\text{eff}} = 1.15$, as it is flushed with dry nitrogen. The camera focal length is $f$ and the radial distance on the detector from the optical axis is $r$, both in same units. The wavelength of the peak of the transmission curve on the optical axis is $\lambda_{\text{center}}$. Therefore a nightsky line will show up as an enhancement in the sky background in an annulus, whose radius corresponds via the above equation to the wavelength of the nightsky line. Its spread in the radial direction corresponds to the width of the etalon’s transmission curve.

Such rings are modeled via FIND/RING and removed with the command REMOVE/RING. The latter scales the ring amplitude with the ratio of the FR_STAT(1) values.

4.4.5 Scattered light

Etalon in CAFOS

We had to realize that the etalon in CAFOS introduces scattered light in the instrument. This is seen by a comparison of internal flats taken with the pre-filter with those taken with the etalon. As the wavelength range is roughly speaking the same so the flatfields should be the same, but they are not. There is a marked difference of the global flat field shape on the 20% level for the CAFOS etalon.

We have developed a procedure to measure the pure transmission of the etalon and use this knowledge to convert the flatfield in the pre-filter to a etalon flat, preserving the fine-scale (fringe) pattern present due to the narrower etalon bandpass. We use a mask with regularly spaced holes (diameter about 2″, separation 20″) through which we expose internal or dome flat with both the etalon and the pre-filter.
4.5. **REMOVAL OF COSMIC RAY EVENTS**

alone. The flux ratio in the holes provide the means to relate the etalon transmission to the transmission of the pure pre-filter, as the scattered light does not fall onto the holes.

To create a proper etalon flat from scratch we proceed as follows:

- construct a transmission model for the etalon using the mask frame in the sense
  \[
  \text{ratio} = \text{mask flux in etalon} / \text{mask flux in pre-filter}
  \]

- smooth the (twilight) flat for the pre-filter and normalize it to 1 using its `FR_STAT(1)`

- smooth the (internal or dome) flat for the etalon wavelength and normalize it to 1 using its `FR_STAT(1)`

- create the final etalon flat from the internal or dome etalon flat according to
  \[
  \text{flat(final)} = \text{flat} \times \text{flat(pre smooth)} / \text{flat(eta smooth)} \times \text{ratio}.
  \]

Please note that one has to use either dome or internal flats consistently and not a mixture of both. The whole procedure has to be performed for all wavelengths the etalon was used with and the flats derived this way are the multiplicative flatfields. As outlined above, there will remain a contribution due to the scattered light in the frames corrected along these lines. To end up with flat images the background has to be modeled and the surplus over `FR_STAT` be subtracted. This is most easily achieved if the multiplicative correction is done with `PREP/CCD` and all frames for a given etalon setting are averaged with `FLAT/AVERAGE`. The output frame is then used in `CORRECT` as the additive flat field. This at the same time corrects for possible remanent fringes, which are left after the multiplicative correction of long-wavelength etalon images. We currently think this is due to the monochromatic nature of the nightsky emission lines, which are not correctly modeled by the continuum light used for the internal or dome flats. If this is so, these remanent fringes are artefacts of the sky background only and it is indeed correct to subtract them.

Should any additional additive structure remain (e.g. due to variations in the intensity of night sky lines) the individual frame’s background should be modeled with `FIND/RING` and the model removed from the image with `REMOVE/RING`.

### 4.4.6 Applying the flatfields and clean-up

At the beginning we had anticipated that `PREP/CCD` is used to correct only for the pixel-to-pixel sensitivity variations. Therefore we created the commands `CORRECT/IMAGE` and `CORRECT/FLIP` to apply the global multiplicative flat field (derived from the twilight flats) as well as the additive flat fields together in one go. The different paths discussed above necessary to correct all flat field effects made this concept more or less obsolete. As we did not have the time to adjust the software to this situation, we have to live with the two commands for the time being. Please keep in mind, that `CORRECT` can be called only once for a given image. Its main purpose therefore these days is the subtraction of the additive flat field and the clean-up. If you have to iterate your flatfield, you should go back to the prep-stage, once the flatfield has settled to a stable version, and call `CORRECT` just once with this final flatfield.

Clean-up are several trivial operations. First we set the descriptor `STEP` to its value in arcsec. The `START` descriptor is chosen such that the optical axis is at \([0, 0]\). The necessary information comes from the call to `MPIAPHOT` and the proper choice of instrument and detector. The second major thing happening is the change of image orientation. Depending upon the telescope focus and the location of the read-out port on the chip it may be necessary to flip the image in order to have the astronomical standard orientation. In addition the descriptor `PA_IMAGE` is used to rotate the image such that north is up as close as possible. Please note that due to inconsistent FITS keywords it may be necessary to check `PA_INSTR` and `PA_CASS` before you call `CHECK/DESCRIPTOR`. Even if the multiplicative flat field correction is done with `PREP/CCD` and no other correction is applied it is still mandatory to call `CORRECT` in order to perform these clean-up operations.

### 4.5 Removal of cosmic ray events

Cosmic ray events ("cosmics") affect one or more neighbouring pixels, creating a spurious signal in them. These events could be due to real cosmic ray events or radiation from radio-active material like filter
CHAPTER 4. DATA REDUCTION PROCEDURES

glass. These false signals have to be removed before further processing. However, care has to be taken not to destroy the photometric information.

Removal on individual frames is possible (see context badpixel), relying on the gradient to neighbouring pixels. Optimum removal, however, is using dithered images and comparing each pixel in each image with the aligned median image. Pixels deviating more than a given threshold from the median are replaced by the scaled median value. Special care has to be taken not to correct the centres of stellar images. The median is also only meaningful, if the set of input images have about the same seeing. Therefore it is mandatory to check the quality of the cosmic removal by blinking at least some representative images with their cosmic mask and check, if all cosmics in the background had been removed and if the cores of stellar images remain untouched.

Bad pixels can in principle be treated like cosmics, if dithered images are used. For this its best if they are set to 85% of the saturation level using the MIDAS command replace/image.

Related commands are cosmic/median (page 50) and cosmic/refer (page 51).

4.6 Creating the sum frames

4.6.1 Determining relative image orientation and shifts

To determine the relative orientation (shift, scale, rotation and shear) of the dithered images FIND/OBJECT (see page 66) and FIND/MOVE (see 69) are used. The information on the relative orientation of the frames is stored in descriptor XY_move.

4.6.2 Summing the images

Object search is performed on summed images for each pointing and filter. Therefore the first step after pure data reduction is finished will be to produce these summed images from the reduced single frames. The summation is performed with sum/image (see page 72 for details). Do not use the MIDAS command compute/ima, as this does not use the information from xy_move and does not take care of the adjustment of descriptors like CCD_PARA.

4.7 Working with mosaics

For large surveys it is economic to perform the data reduction in frame samples as large as possible, and it is reasonable to work square-degree-wise. A field of 1 × 1 degree size generally contains several hundreds of individual frames in each filter, which thus will be treated simultaneously.

To facilitate handling these large data sets it is convenient to use a table of all images containing the relevant information for each image. Based on such a ”frame-table” catalogues of sub-sets can be easily created to be used as frame-lists (see page 195). Creating and filling a frame-table and extracting a catalogue from it is done with commands frametab/create, frametab/fill, frametab/catalogue (see page 52 ff).

4.7.1 Rebinning images into the gnomonic projection

The strategy of the following reduction is to project all catalogue frames directly on the sky, using large reference catalogues such as USNO, SDSS, or 2MASS. First, object tables are produced for each frame, with about 400 locally well defined objects. This is done with the command find/object (see page 66).

Right after this routine the derived seeing parameter (PSF) should be copied into the frame-table (frametab/fill).

Optical distortion correction

Next step is the correction of the optical distortion of the telescope camera, which may be unmeasurably small (for OMEGA2000) or up to tens of arcsec (for LAICA). For this, the position of the optical axis (descriptor OA_PIX) has to be supplied. The distortion radially symmetric to this optical axis is specified
4.7. WORKING WITH MOSAICS

as a polynomial of up to 6th degree in descriptor DISTORT (both usually written by check/descr, based on information selection by set/cont mpiaphot with the correct specification of instrument and detector.). It is thus essential to have images available which allow a good determination of the distortion.

Coordinates in the table created by find/obj are transformed to the undistorted values by undistort/tab (p.68), which assumes the table to be in the so-called MARK-format (see page 96).

Projection to the sky

For the next step a reference table has to be built containing gnomonic coordinates $\xi$ and $\eta$ relative to the projection centre, and a magnitude in a colour similar to the observed filter. This table needs to have a descriptor for the reference position ref_pos/d/1/3, defining the coordinate centre in RA and DEC and its epoch. It is set by the command SET/REFPOS (see page 65). Usually such an object table is downloaded for e.g. SDSS from the web containing RA and DEC columns. Then :xsi and :eta are calculated using @@PM:radec2xy_tab table_name. Now, the projection parameters relative to this celestial coordinate grid can be derived with the command find/move (p. 69).

In order to allow a small search radius the telescope position as specified by descriptor O_POS should be as good as possible (better than 0.01 degree). When the find/move command results in "Frame does not match ...", this means that the frame coordinate is not accurate enough to match the reference table. In this case, a first trial should be made using an larger search radius, which will finally lead to a matching, and yields in the last line the error of O_POS in degrees. It has, however, always to be made sure that the number of matching objects is of the order of 100. If it is less than 10, or if the RMS is larger than 0.5′′, something is wrong: the frame may belong to another field, or most of the brighter stars are saturated (and thus are not included in the object table), or the frame was not distortion corrected. If USNO is used as a reference catalogue, the RMS never becomes smaller than 0.2″, which is the intrinsic accuracy of the USNO coordinates (after exclusion of objects with high peculiar motions, large position errors, or/and large extension). A correction of O_POS (e.g. for a constant offset) can be done with @@PM:correct_OPOS frame-catalog dRA dDEC (equinox), offsets in degrees. The equinox is useful, since the coordinate mismatch may be partly due to precession.

Once find/move was successful, it yields a transformation matrix with 6 independent parameters, expressing shifts in x and y, scalings in x and y, rotation and shear (which is necessary because of the large fields and is of the order of 0.02%). These parameters will be attached as descriptor XY_MOVE to the frame. The routine also derives a mean flux scale by comparing the magnitudes estimated from the count rates in each frame with the reference catalogue magnitudes, which is attached as descriptor rel_scale (large rel_scale means less deep image!). This descriptor, needed to judge the quality of the frame, should also be copied into the frame table with frametab/fill. One can now determine a depth parameter (combination of seeing, background level and transmission) for each individual frame by means of frametab/fill frame-table = list depth.

For the realization of the cosmic correction and of a summation of all observations, we now need projected images of all available frames in the observed area and with the same filter. In order not to loose too much of angular resolution, this is done with the drizzling technique, projecting the frames onto a coordinate grid with a stepsize somewhat smaller than the stepsize of the input frames. At the moment, this grid step size is taken to be 0.2″ for HIROCS. This is done with the command mosaic/gnomima. These rebinned frames show the objects at their loci on the celestial sphere relative to the reference position. With @@PM:wcs2radec frame the frame descriptors are set up so that e.g. get/curs delivers directly RA and DEC. To reverse this use @@PM:radec2wcs frame. The errors of the absolute object positions in this gnomonic frame are usually considerably smaller than 0.2″. The difference of a gnomonic frame to a drizzled frame is, that there are no holes in the frame. These are, other than in the original drizzle algorithm, filled with the value of the nearest pixel in the original frame.

4.7.2 Correcting cosmic ray events

From these gnomonic frames a large median frame will be produced covering up to $1 \times 1$ degree in extension. Command mosaic/median produces a median and an RMS frame (default size: maximum area covered by the images) and with a pixel size of 0.2″ (defaults). Due to the huge frame size the
median routine can only be performed in dozens of chunks (slices), depending on the number and size of frames incorporated, and needs several hours computing time.

Note: $18 \times 18$ k is about the largest possible frame, which can be handled on sun2.

After production of this median frame, the cosmic correction is performed first on the gnomonic frames (command `mosaic/rmcosmics`). In the case of strongly differing seeing values (differing by more than about 40%), the cosmic correction will produce "rings" around the bright objects. To avoid this, the whole procedure has to be performed in subsets of frames of approximately the same seeing.

Together with the cosmic corrected frames (cosmics replaced by the median value), mask frames are produced. From these, the cosmic corrected frames referring to the original images can easily produced by re-projecting the cosmic corrected pixels from the gnomonic frames to the original frames (command `mosaic/copycos`).

Short exposures should not be corrected for cosmic ray events. For consistency the "corr"-frames have to be moved or copied to "cosm"-frames. This is most conveniently accomplished via FRAME/RENAME (see page 63).

### 4.7.3 Sum mosaic

From the cosmic corrected gnomonic frames one can now build up a sum image, from which the MASTER object list will be derived by means of SourceExtractor. The sum frame has reasonably an extension of $1 \times 1$ degree (although any extension can be chosen), and is produced with the command `mosaic/sumup`. Summation is done in a weighted average. However, a selection could be done e.g. by means of the depth column in the frame-table:

```
sele/tab frame-table :texp.gt.50 and :depth.gt.0.5
frametab/cat cat = frame-table gnom
```

Before running the sumup routine, the background of each individual gnomonic frame should be made as flat as possible and the level determined as accurate as possible; otherwise the sum image shows edges and discontinuities, which may cause problems in the object detection routine.

In addition to the sum (which is actually a weight-averaged frame), the routine produces a so-called weight frame, which monitors the depth of the sum image at each positions within the field and which is necessary for SourceExtractor as a weight frame, a psf-frame with the mean seeing at each position, a background frame, and a frame of the rel scale distribution.

Please note: Due to the way it is constructed, the sum image is not photometrically correct (each original pixel may end up several times in the gnomonic image!). This is different to the original drizzle algorithm. Thus photometry should not be done on the sum image. It is currently not known how large the photometric deviation really is.

### 4.8 Finding objects and creating the master list of objects

#### 4.8.1 Object detection

Objects should be searched on summed frames which should contain as much total integration time in one band as possible. We use SourceExtractor (http://terapix.iap.fr/soft/sextractor/) searching for objects with $N > MIN\_PIX$ connected pixels above a certain threshold. For the search the input image is smoothed with the seeing PSF (taken from descriptor `PSF\_MEAN` or (of not present) from `PSF\_FIND`). We have incorporated the call to SourceExtractor into `MPIAPHOT`, converting the input bdf-file to FITS and at the end converting the output ASCII-file to a MIDAS table.

For mosaic images of LAICA, WFI and OMEGA2k (identified as such via the descriptor mosaic/i/1/1 set to 1), a weight image should be used as the S/N may be variable for the individual pointings entering the mosaic.

See command `FIND/SEXTRACT` on page 80.

However, this process is not quite so straightforward. SourceExtractor tends to find spurious objects around very bright stars and moderately bright stars tend to get extreme FWHM. "Strange" objects are also found along diffraction spikes or satellite trails. All these artifacts have to be eliminated before `MASTER/IMAGE` is called. The basic procedure is as follows (see also Fig.4.2):
4.8. FINDING OBJECTS AND CREATING THE MASTER LIST OF OBJECTS

1. Set cleaning radius around brightest stars, i.e. objects within this radius are flagged by "X" in column ID.

2. Flag largest spurious objects by "L".

3. Plot stellarity index (column :dx_line) against SE magnitude (column :mag_0) and determine location of real stars (upper left corner of plot). Their half width is set to the PSF and the objects are flagged by "Y".

4. Determine location of moderately bright stars who might have artifacts near them (upper centre of plot) and

5. Plot histogram of these moderately bright stars and determine their typical half-width. Objects with too large major axes are flagged by "Z".

6. Patch areas of spurious objects (e.g. along blooming) by using PATCH/SEXTRACTOR before calling CLEAN/SEXTRACTOR. Patched objects are flagged with :ID = "W".

These steps are performed by CLEAN/SEX which allows interactive cleaning of SourceExtractor tables. Any remaining spurious objects are only a few and have to be isolated and flagged manually. Please note that objects which should not enter into the master table are not removed from the table but flagged in column ID. Whereas the W, X, Z and L flags can be reset, once the half width is set to the PSF values from descriptor PSF_find, the original values are lost. For details see the description of the command CLEAN/SEXTRACTOR on page 84.

4.8.2 Master list

From the sum images an object master list has to be created. This is done on the basis of the FIND/SEXTRACT results. For each object found in any of these (e.g. for the different filters) an artificial (noise free) image is produced with strength proportional to the S/N and shape according to the SourceExtractor results (see command MASTER/IMAGE on page 88). An object search is then performed on this artificial image using command MASTER/TABLE (see page 89). The output is the master list and should be named Master_field_date. It need not be in MARK-format.
Investigate the stellarity distribution:

\[
\text{sel/tab test :mag_0.le.18.and.:dx_line.ge.800.and.:mag_0.gt.16}
\]

Almost all remaining 29 objects are spurious and could be checked manually.

real galaxies
real stars

Figure 4.2: Example of plots displayed by CLEAN/SEX to clean and prepare an object table for creation of master list.
4.9 Photometry

4.9.1 Separation stars / galaxies

SourceExtractor does not produce a reliable stellarity index, especially at the fainter end (see manual on page 60 at http://www-int.stsci.edu/holwerda/se.html). Therefore an independent classification based on evaluate is used.

The command CLASSIFY/EVALUATE described on page 91 produces the following classification plot:

![Classification Plot](image)

Figure 4.3: Plotting the effective area of an object as a function of its intensity separates stars (red horizontal distribution) from galaxies (and cosmic ray events).

The optimum classification for each object is finally obtained from the image, on which a given object has the largest S/N. Command CLASSIFY/COMBINE uses as input all CLASS/EVAL outputs and extracts the optimum classification for entry into the master table.

4.9.2 Transformation of master coordinates to individual frames

The master frame / table contains all the objects to be used in the data analysis. The coordinates of these are in the projected frame of the tangential plane to the celestial sphere. The individual frames contain the information in descriptors, on how to transform the coordinates from the master table to each individual frame. This includes also the image distortion introduced by any optics.

Command MASTER/SECTION (see page 90) does this transformation. It extracts all objects lying in the individual frame and produces a MIDAS table in MARK-format for each image ready for use with EVALUATE.
4.9.3 Determination of the count rates for each object

The photometry attempts to analyse all frames with a common effective point-spread-function (PSF) in order to avoid spurious colours when combining images taken under different seeing conditions. To this end every frame is analysed with a different (Gaussian) weighting function, such that

\[ s_{\text{eff}}^2 = s_{\text{seeing}}^2 + s_{\text{weight}}^2 = \text{const.} \]

with \( s \) being the e-folding widths.

Thus frames with good seeing are analysed with a broad weighting function and vice versa. In other words, the image with the worst seeing determines the final resolution. Thus an upper seeing limit is imposed on all data to enter the final data analysis.

Theoretical considerations [11] of the photometric S/N ratio show that an optimum S/N ration is obtained when

\[ s_{\text{seeing}} \approx s_{\text{weight}} \quad (\Rightarrow s_{\text{eff}} \approx \sqrt{2} \times s_{\text{seeing}}). \]

Note: In MPIAPHOT all e-folding widths are expressed in pixel units:

\[ \text{FWHM}_{\text{seeing}} [\arcsec] = 1.67 \times s_{\text{seeing}} \times \text{scale} [\arcsec/\text{pixel}]. \]

scale corresponds to the MIDAS descriptor \text{STEP}.

Setting up EVALUATE

For the standard set of observations (FWHM\text{seeing} \leq \text{seeinglimit}) it is recommended to set

\[ \text{FWHM}_\text{eff} \approx \sqrt{2} \times (\text{seeinglimit} - 0.2\arcsec) \]

However, in case of observations dominated by frames close to the seeing limit one should choose the maximum FWHM, i.e.

\[ \text{FWHM}_\text{eff} \approx \sqrt{2} \times \text{seeinglimit}. \]

The parameter file used by EVALUATE is generated with the command SET/EVAL with mode set to MC. The questions for the effective PSF have to be answered by the appropriate value as discussed above in arcsec. The background should always be determined by the histogram method. Local background corrections must not be used.

Naming convention for the setup-files (≡ to the root of the EVALUATE output tables!) should follow the rule:

\text{exxi}, where \( xx \) gives the effective PSF width in 1/10 arcsec. \( i \) indicates the detector pixel size used: \( i = W \) for WFI, \( L \) for LAICA, and \( O \) for OMEGA2k.

Important: SET/EVAL uses information about the instrument and the detector as set by MPIAPHOT during initialization. Be careful that both are set correctly during set/cont mpiaphot!

Running EVALUATE

a) Long exposures: Once everything is set up correctly, \textit{i.e.} the cosm- and corresponding cmsk-frames are all present and equipped with the necessary descriptors, the count rate is evaluated with

\text{EVALUATE/MC = vvvv in_list tbl_root}

\text{in_list} provides the list of cosm-frames (corresponding cmsk-frames must also be present in the same path) and \text{tbl_root} specifies the leading 4 characters of the MARK-tables created by MASTER/SECTION. The output tables are named according to the input frame name with the root being replaced by vvvv.

\text{EVALUATE/MC} first calls the standard EVALUATE command, which creates extensive output tables, both for the image and its mask. These tables are then condensed to the columns needed for the multi-colour analysis. As the first step is performed for all images in sequence and only afterwards the final MC output tables are created, disk space requirement is dictated by the full EVALUATE table size. Be sure enough space is available on the disk! b) Short exposures: For short exposures no cosmic corrections (usually) done, thus there are no cmsk-files on which EVALUATE can be run. Thus the call has to be \text{EVALUATE/MC = vvvv in_list tbl_root NOSAVART}

\text{i.e. parameter P5 is not used in its default mode COSM.}
4.9. PHOTOMETRY

4.9.4 Flux tables

Flux tables are the final product of MPIAPHOT and are created in context flux_cal. They contain the relevant information for astrometry and photometry. Flux tables are the input for the object classification procedure [13, 14] in context CLASSY (see page 131).

Flux tables are created in several steps, once all the evaluate runs have been done:

- **FLUX/VERIFY** checks if all reduced frames and associated mask images are present and have the necessary descriptors (page 106).
- **FLUX/REFER** defines the master table to be used (page 101).
- **FLUX/STD** identifies objects to be used as standard stars in the master table (page 102).
- **FLUX/SETNORM** defines the best image for each pointing and filter to be used as the normalization reference (page 105).
- **FLUX/SELMNORM** selects for each evaluate table the appropriate stars for normalization (page 103).
- **FLUX/UNITE** creates for each filter a large table holding a subset of columns from each image relevant for photometry (page 107).
- **FLUX/NORM** Does the normalization to the first image entered into the unite table (block 1, should be a short exposure) and corrects saturated images (page 111).
- **FLUX/SURVEY** copies magnitudes from the survey file(s) into the unite table (page 112).
- **FLUX/MOSAIC** scales count rates in the unite table so all exposures look as taken simultaneously (page 114).
- **FLUX/INTEG** does the photometric calibration using standard stars in the fields, for which we know the photon flux as a function of wavelength (page 115).
- **FLUX/COMBINE** finally combines the calibrated unite-tables into one table to be used as input for the multi-colour classification (page 117).
- **FLUX/AVERAGE** takes the average of 2 columns (e.g. in a similar or same filter) (page 120).
- **FLUX/ANALYSE** Copies all CR/flux information for a given object into a table for analysis (plot, compare values etc..) (page 116).
- **FLUX/BLOCK** returns the block number of a list of tables (page 122).
- **FLUX/STORED** returns the table name stored in given block (page 122).
- **FLUX/CLEANUP** deletes unnecessary columns after FLUX/AVERAGE (page 121).
- **FLUX/PRINTCAL** prints the various calibration descriptors created by FLUX/INTEG into an ASCII-file for analysis (page 123).

- Due to the difficulty, disentangling scattered light from flatfield features the photometric properties of an image might have been distorted by the flatfielding. The only way to check this is by a careful comparison of the final flux values with e.g. 2MASS or SDSS magnitudes. Three commands have been created to facilitate this analysis. They assume that the un-detected scattered light is radially symmetric around the optical axis. The distance to the optical axis for every object is calculated by FLUX/UNITE and stored in the column :scrccccii for each block. As this is the scratch column be sure not to over-write this if you plan to use the ANALYSIS feature!
  
  - **ANALYSIS/FILL** collects the information needed for the analysis from the input unite table calibrated with FLUX/INTEG (page 124).
– **ANALYSIS/FIT** smooths the individual data in radial slices and fits polynomials (page 126).
– **ANALYSIS/STORE** does the analysis for each observing run contained in the unite table and stores fitting information for later use in **FLUX/CORRECT** (page 128).
– **FLUX/CORRECT** corrects the count-rate for each object based on the analysis above in a unite-table (page 129).
Following this the whole procedure has to be repeated starting from **FLUX/NORM**.
– An important implication therefore is that if ultimate photometric accuracy is needed, during every observing run images of e.g. an SDSS field should be taken in the filters used for the science observations. Only this will allow to use the **ANALYSIS** feature.

Please refer to the description of the individual commands for details.

### 4.10 Astrometry

A set of commands is supplied which allows easy use of context **astromet** in MIDAS by Richard West (see page 160). For mosaics the plate solution is generated during creation of the master image and table. The unite tables, if created using the mosaic commands, will contain columns :RA and :DEC, which are accurate to the level of the rms in **FIND/MOVE** with a survey table as reference (typically 0′05 for WFI).

For mosaics the gnomonic images as well as the sum frames directly give RA and DEC if descriptors are set correctly via @@ PM:wcs2radec frame. To go back to the usual pixel / wcs coordinates use @@ PM:radec2wcs frame.
4.10. ASTROMETRY

**Fixed-pattern noise**
- INPUT: internal / dome flats
- CHECK: PREP (without FPN)
- FLAT/AVER for FPN
- model FPN
- normalize FPNflat = FPN / model

**Twilight flats**
- INPUT: twilight flats
- CHECK: PREP (with FPN)
- FLAT/AVER
- twilight flat = twilight flat * FPN

**Etalon flats**
- INPUT: etalon flats pre-filter flats
- SMOOTH pre-filter (twilight) flat etalon (internal flat)
- COMPUTE/IMA etalflat = etal(internal) / eta (internal smooth) * pre(twilight smooth) * ratio frame

**Etalon mask image**
- INPUT: mask images
- Flux through holes in prefilter
- Fluxes through holes with etalon
- ratio = flux(eta) / flux(pre)
- ratio image

**Science frames**
- INPUT: science frames
- CHECK: PREP (with twilight flat)
- FLAT/AVERAGE science frames fringe pattern
- CORRECT additive flat with fringe pattern
- FIND/RING
- remove/ring to remove remanent scattered light or night sky rings
- find/object to detect well defined objects
- find/move to establish shift and relative exposure level
- COSMIC/MEDIAN removes cosmic ray events
- sumup/ima sum images
- check/quality Quality control
- store data as FITS files on tape

**CADIS Data Reduction**
- Flatfields
- Science Frames
- CADIS Data Reduction
Chapter 5

Description of Commands

Presently, there are 6 context levels available in \textit{MPIAPHOT}:

(1) \textbf{main} \hspace{1cm} \texttt{set/cont mpiaphot}
Here the whole package is initialized. The context \texttt{mpiaphot} contains those commands which are required for basic reduction and analysis of (direct) CCD images and is listed on page 32.

(2) \textbf{flux\_cal} \hspace{1cm} \texttt{set/cont flux\_cal}
It contains commands for flux calibration and is listed on page 100.

(3) \textbf{fpi} \hspace{1cm} \texttt{set/cont fpi}
It contains commands necessary for reduction and analysis of Fabry-Perot images. The commands are listed on page ??.

(4) \textbf{polar} \hspace{1cm} \texttt{set/cont polar}
The commands for reduction of polarimetric data obtained with a Savart plate. The corresponding commands are listed on page ??.

(5) \textbf{synphot} \hspace{1cm} \texttt{set/cont synphot}
Commands used for photometric analysis, following the idea of synthetic photometry are listed on page 156.

(6) \textbf{backfit} \hspace{1cm} \texttt{set/cont backfit}
Commands used for background modeling are listed on page ??.

In the following sections we will first list the commands available within each context. Subsequently all available commands in each context will be documented in full detail alphabetically.
5.1 Context: main

▷ input of FITS data INPUT/FITS
▷ check/update basic descriptors CHECK/qual
▷ frame-table handling
▷ create flatfields
▷ remove cosmic ray events
▷ create mosaics
▷ photometry
▷ create flux tables
▷
▷ image and table utilities
INPUT/FITS frame_numbers observing_period filter_ID frame_ID offset tape FITS-header

Purpose:
Read in FITS files, convert them to MIDAS and name it according to CADIS conventions:
The filename is 16 characters long and made up of different groups of letters and numbers with the following meaning:

- **xxxx**: Identifier of the reduction stage of the file, e.g. prep.
- **Ffff**: Unique filter specification. F should be set to f for filter, i for interferometer, and g for grism exposures. The fff give the central wavelength in [nm]. Use FOCU, SLIT, BIAS, DARK for focus, slit, dark0, and darkX exposures for the four digits Ffff.
- **Pppp**: Unique code of the observing period (X000 ... X999). For CADIS the first letter may be used to identify the field.
- **iiii**: Frame number within period Pppp. If more than one tape is needed, use offset-parameter.

The resulting filename is then xxxxFfffPpppiiii with the extension .bdf, about which the user doesn’t have to care.

Once different frames are stacked individual frames lose their identity and the above scheme breaks down. In this case the descriptor HISTORY should provide all the necessary information. Appropriate names up to 16 digits long (plus extension) may be used in this case.

Parameters:
- **p1 IN_list**: List of file numbers as in the MIDAS command INTAPE/FITS.
- **p2 Obs_period**: Four digit identifier for observing period during which these data have been taken.
- **p3 Filter_name**: One digit key and three digits wavelength designation for filter used.
- **p4 Name**: Four digit temporary file identifier, same as P2 for MIDAS INTAPE/FITS.
- **p5 Offset**: Offset of file numbers, e.g. if data do not fit onto one tape, this is the number of images on tapes with lower sequence number.
- **p6 TAPE**: Tape unit.
- **p7 HEADER**: Length of header output as in INTAPE/FITS.

Examples:

```
INPUT/FITS 1-3,7,15,34-50 65 F655
```

Read files 1 to 3, 7, 15, 34 to 50 for observing period 65 from default tape unit. These are broad R-band images. The frames on disk will be named `raw_F65565001.bdf`, `raw_F65565002.bdf` ... `raw_F65565050.bdf`.

Actions taken:

1. Old file names will be restored according to the content of keyword FILENAME or FILE_NAME.

Related commands:
INTAPE/FITS
NAME/HIROCS instrument_ID obs_run = frame_list cross-reference data_path

Purpose:
Read raw data FITS files from disk, convert them to MIDAS and name it according to HIROCS conventions using a cross-reference file. All raw data are on disk with a MIDAS catalogue called raw. The filename convention for the output BDF-file is given on page \[196\].

Cross-reference file:
First line is comment. Each subsequent line holds first the file name on disk (with extension .fits, no path). There is no naming convention for the raw data FITS-files. This part is not restricted in the number of characters. The following information — separated by semicolon — is image type, filter, and sequence number (unique within observing campaign).

Example for first 4 lines:
O2k campaign in February 2006
A0001.fits;dome_\_;H_\_;0001
A0002.fits;dome_\_;H_\_;0002
A0003.fits;dome_\_;H_\_;0003

Parameters:
\[p_1\] Instrument Three character ID for instrument (LAI, WFI, O2k)
\[p_2\] Obs_period Four digit identifier for observing period YYMM during which these data have been taken.
\[p_3\] =
\[p_4\] IN_list List of files to be read (see page \[195\]). If a catalogue is used it must exist in the raw data directory, usually it is named raw. The file names in the catalogue must not contain the path!
\[p_5\] cross-reference file Cross-reference file (extension csv) must exist in path $PM/../archive$.
\[p_6\] path of raw data Path where raw data (FITS files) are located. Default: ./
\[p_7\] mode Mode of data input. Normally one wants to convert FITS-files to BDF-files [default]. However, in case of OMEGA2000 the pipeline uses FITS-files as input. In this case the files just have to be copied with the new file names. Any input other than indisk for this parameter will copy the FITS-file without conversion to BDF.
\[p_8\] root Root of the input frames. Default: raw_.

Examples:
NAME/HIROCS O2k 0510 = raw:1-3,7,15,34-50 O2k_0510 /disk-h/HIROCSSraw/OMEGA2000/Sep03

Read files 1 to 3, 7, 15, 34 to 50 for observing period October 2005 (OMEGA2000) from the HIROCS raw data depository. Use cross-reference file O2k_0510.csv.

Actions taken:
1. MIDAS command INDISK/FITS is used to create the BDF frame.
2. Information from cross-reference file is used to build HIROCS file name.

Related commands:
fits-to-disk
ARTIFICIAL/IMAGE OUT_name = object_parameters image_size start/step noise

Purpose:
Create an artificial image with objects as Gaussians for testing purposes. One object is added per call. If image exists, object is added into this without background.

Parameters:

\( p_1 \) **out_name**
  first four characters (root) for name(s) of output images

\( p_2 \) =

\( p_3 \) **object parameters**
  – Peak,
  – center in X,
  – center in Y,
  – major axis (e-folding width),
  – minor axis (e-folding width),
  – position angle [mathematical sense, \textit{i.e.} angle between major axis and the positive X-axis],
  – background (only relevant for new image). No default!

\( p_4 \) **image size**
  Pixel dimensions in \( X,Y \) [default = 500,500]

\( p_5 \) **start, step**
  Start-values in \( X,Y \), step-values in \( X,Y \) [default: 1,1,1,1]

\( p_6 \) **noise**
  Noise level relative to Poisson noise [default: 1.]

Examples:

\texttt{art/ima test = 100,200,250,10,15,20,2000}

Creates an artificial image of size \( 500 \times 500 \) pixels. The object has a peak of 100 at position 200,250. The e-folding axis are 10 and 15 at PA 20. Background is set to 2000 counts. Start and step are all 1.

Actions taken:

1. Create artificial image with one Gaussian object
2. If image exists, Gaussian object is added into this without increasing the background.

Related commands:

\texttt{profile/gauss}
profile/gauss FWHM

**Purpose:**
Create an table with a (normalized) Gaussian profile of width FWHM [arcsec]. This may be used to compare visually the results for the PSF provided by FIND/OBJ (see page 66).

**Parameters:**
- \( p_1 \) FWHM  
  Width of Gaussian [arcsec]

**Examples:**
- `prof/gauss 1.5`
  Creates table `gaussprofile` with values for a Gaussian profile

**Actions taken:**
1. Table with 2 columns is created: \( :x \), \( :gauss \). Number of rows is 61.
2. Column \( :x = -3 + (\text{seq}-1)*0.1 \)
   Column \( :gauss = \exp(-(:x/w)**2) \), \( w = \text{FWHM} / 1.66511 \)
3. Copy the table also into an image (same name: `gaussprofile`)

**Related commands:**
- profile/gauss
5.1. CONTEXT: MAIN

MARK/DISPL

Purpose:
Mark interactively with cursor objects on the display to include them in a MARK table. This command is mainly for testing purposes.

Parameters:
— none —

Actions to be taken by the user:
1. Load image into the display.
2. Mark approximate object centres with cursor (left mouse button).
3. End object selection with click on right mouse button.

Actions taken:
1. Create table with root MARK and characters 5ff of display image.
2. Columns of the table are as for a MARK table.
3. Minimum set of descriptors is written to allow direct application of EVALUATE. Some of them may be meaningless, e.g. FR_STAT(2) at this stage.

Related commands:
BIAS/EXTR OUT_name = IN_list window mode kappa,loop degree

Purpose:
Input frames are fitted pixel-wise with a polynomial of up to degree 2. Output frames give the polynomial coefficients for each pixel: const, slope, quadratic term, and rms.

Parameters:

\( p_1 \text{ out_root} \)
First for characters for name(s) of output images. The full names will be \text{root_const}, \text{root_slope}, \text{root_quad}, \text{root_rms}.

\( p_2 = \) 

\( p_3 \text{ IN_list:} \)
List of input frames (see page 195)

\( p_4 \text{ window:} \)
Window in the frame according to MIDAS conventions. For each pixel within this window a column in the table \text{TEST_TBL} is created holding the actual pixel values for each frame. First column is the column corresponding to the \text{MODE} (see parameter \( p_5 \)). Default: none, \text{i.e.} no output table is created.

\( p_5 \text{ mode} \)
The independent variable of the fit. Could be either \text{TIME} (default) or \text{LEVEL}.

\( p_6 \text{ kappa,loop} \)
Parameters for the \text{kappa-sigma} clipping. Default: 0,1, \text{i.e.} no clipping.

\( p_7 \text{ degree} \)
Degree of the polynomial to be fitted (0, 1 (default) or 2).

Examples:

\text{bias/extr test = cat:2-6 [100,100:103,103]}
Fit each pixel over images 2 to 6 from catalogue cat with a straight line. Output images are \text{test_const}, \text{test_slope}, \text{test_rms}. The table \text{TEST_TBL} will contain columns :ex_time, :pix_100,100 to :pix_103,103.

Actions taken:

1. Create output frames for each polynomial coefficient.

2. Optional produce test table \text{TEST_TBL}.

Related commands:
@@ PM:movie frame_list channel scale centre cuts prompt

Purpose:
Display images from a catalogue one after the other. Cross-references in the call are the same as in LOAD/IMA. Images can be flagged in an output file e.g. to identify bad frames in a catalogue.

Parameters:

\( p_1 \) frame_list List of input frames (see page [195])
\( p_2 \) channel Display channel where to load the image
\( p_3 \) scale Scale (in pixels) for loading
\( p_4 \) centre Position in image of centre of display
\( p_5 \) cuts Cuts for image to be loaded
\( p_6 \) prompt If set to y, a file for flagged images is created. Otherwise it gives the time in seconds until next image is loaded.

Examples:

@@ PM:movie flats: prompt=y
Displays images in catalogue flats and prompts after each image for input. The file movie.lis is created to hold flagged image names.

Actions taken:

1. Displays images in frame_list
2. Waits number of seconds as specified in parameter P6 before next image is displayed or
3. if parameter P6 is set to y, the user is prompted after each image has been displayed. A file movie.lis is created in this case (if it already exists output is appended to this!) into which those filenames are entered which are prompted with a letter.
4. Actions for different prompts:
   - An empty prompt switches to the next image without entering the filename to movie.lis.
   - Any letter indicates the displayed frame to be flagged, i.e. the prompt letter and the filename are appended to the file movie.lis.
   - < goes back to previous image
   - bd determines the cuts interactively via back/det and displays image again.
   - > exits the loop

Related commands:
CHECK/DESCRIPTION IN_list

Purpose:
Check and update the standard descriptors of frames which are to be reduced and analysed under MPIAPHOT.

Parameters:
- $p_1$ IN_list: list of input frames
  The string IN_list may contain one of the following:
  - catalogue name followed by a colon (:) (→ frame list will contain all frames in catalogue) (if all frames from the active catalogue are to be used, only a colon should be given)
  - cat.name:list (e.g. list=1,3–7,9,15–19) to select from cat. If only a list preceded by a colon is given, frames are taken from the active catalogue.
  - frame list, names separated by commas. Here frames from active catalogue may also be referenced via #no!

Blanks in the string are not allowed!

- $p_2$ print flag
  Default is NO_PRINT. Specify NON_IACT to suppress prompt on terminal

- $p_3$ report flag
  Default is REPORT, to suppress terminal output give NO_REPORT.

Actions taken:
Table in the Appendix gives a list of descriptors that are checked for their existence and prompted for contents and created if not present. Already existing descriptors will not be changed!

A special case is descriptor EPSILIST, specifying the instrument transmission. CHECK/DESCR usually copies keyword EPSILIST to the frame descriptor. This keyword is created upon initialization as a blank keyword and has to be filled by the user with a meaningful content before calling CHECK/DESCR. Otherwise the check will be abort. Another option is to write descriptor EPSILIST directly to the frames using PROC/IMA.

Descriptor DATA_LIMIT is not written by CHECK/DESCR but has to be written manually once meaningless pixels have been flagged!

Please note: If check/desc is aborted with CTL-C, the file currently treated will be broken!

Examples:
- CHECK/DESC CATAL:
  use all of CATAL.CAT
- CHECK/DESC CATAL:1,5-7,10
  same as 2. example, but for CATAL.CAT
- CHECK/DESC FRAME1,FRAME5,FRAME6
  use frames given

Related commands:
CHECK/QUICK, CHECK/FPI
PREP/CCD OUT_name = IN_list DARK_0 DARK_X MASK FPN delete

**Purpose:**

Perform the basic CCD-correction task as DARK_0 and DARK subtraction, correction of defect columns, and division by fixed-pattern-noise. Set saturated values to the global saturation value as specified by keyword CCD_PARA(5). Due to the latter it is essential to run PREP/CCD at the beginning of the data reduction stream!

Please note that the calibration frames must have proper descriptors DATA_LIMIT and CCD_PARA.

**Parameters:**

- $p_1$ out_name: first for characters for name(s) of output images
- $p_2$ =
- $p_3$ IN_list: list of input frames (see page [195])
- $p_4$ DARK_0: Amplifier offset subtraction:
  - constant to be subtracted from every pixel
  - OVER calculate constant, to be subtracted from overscan area (as given by keyword OVERSCAN).
  - file to be subtracted as DARK_0
  - file, whose level is to be shifted to level of overscan. To activate this mode, this file has to have descriptor MODE, which holds adjust.
- $p_5$ DARK_X: Dark charge subtraction:
  - constant to be subtracted from every pixel (normalized to exposure time of 1 sec.
  - file to be subtracted as DARK_X. File has to be normalized to exposure time of 1 sec.
- $p_6$ MASK: Column mask, a file of one of the following formats:
  - One line with additive offsets for each column (positive value for missing counts!)
  - Two lines with first holding ... In this case descriptor CRIT_VAL has to be present and hold ...
- $p_7$ FPN: Correction for fixed-pattern-noise:
  - Constant, by which each pixel is to be divided.
  - File holding fixed-pattern-noise
    If descriptor FR_STAT is present, it is used for normalization. Otherwise the FPN-frame has to be normalized to 1.
- $p_8$ delete flag: If set to delete each input frame will be deleted after successful completion.

**Actions taken:**

1. If keyword GOOD_PIX or INSTR_AP are set, only the area described by them is used for the result frame.
2. Keyword **INSTR_AP** is used to define unvignetted area on chip:
   
   $\text{INSTR_AP}(1), \text{INSTR_AP}(2) = \text{center}, \text{INSTR_AP}(3) = \text{halfwidth in X or radius}, \text{INSTR_AP}(4) = \text{halfwidth in Y for rectangle [pixel]}. $

3. Descriptor **CCD_PARA** is updated (elements 3 and 4) if **DARK0** is subtracted in the following way:
   
   $\text{DARK}_0 = \text{const} \rightarrow \text{CCD_PARA(3 and 4)} = \text{CCD_PARA(3 and 4)} - \text{DARK}_0$
   
   $\text{DARK}_0 = \text{file} \rightarrow \text{CCD_PARA(3)} = 0, \text{CCD_PARA(4)} = \text{CCD_PARA(4)} - \text{CCD_PARA(3)}$.

   If descriptor **MODE** of **DARK0_file** is set to **adjust**, then this file will be adjusted to match the **OVERSCAN** values of itself and the input file by adding a constant to the file.

4. If

5. Pixel values are converted from counts to electrons using descriptor **CCD_PARA(1)**.

6. Descriptor **LHCUTS** is set for output frame

7. Bad columns as given in descriptor/keyword **BAD_COL** are interpolated

8. Saturated values are set to the global saturation value as specified in keyword **CCD_PARA(5)**.

9. Descriptor **FR_STAT** is updated

**Examples:**

```
PREP/CCD prep = raw:  over          subtracts DARK_0-value as determined from over-scan
PREP/CCD prep = raw:  over ? ? fpn     as above but in addition divides by fixed-pattern-noise
```
5.1. **CONTEXT: MAIN**

STAT/BACK IN_list window offset flags smooth cuts

**Purpose:**

Calculate frame statistics and write descriptors accordingly.

**Parameters:**

- $p_1$ frame_list: List of input frames (see page 195)
- $p_2$ window: Limits of sub-window, in which statistics is calculated. WINDOW = use FR_AREA (=default)
  
  Another way to specify area is as fraction of image like 0.1,0.1,0.9,0.9.
- $p_3$ offset: Value of bias, defaulted to 0 (letter O !), treated as zero.
- $p_4$ descr. flag: Determines, which descriptors are to be written: Yes = FF_NORM, FR_STAT (only if $p_2$ = WINDOW), Cuts = write LHCUTS, All = Y+C (=default), No = none
- $p_5$ n_smooth: Smoothing area is $n_{\text{smooth}} \times n_{\text{smooth}}$, default = 3
- $p_6$ low, high cuts: Lower and upper cuts value for first histogram (default = 0.,0., meaning that the dynamic range between the limits given by DATA_LIMIT and CCD_PARA(5) is determined internally.)

**Examples:**

stat/back cat:

Determine frame statistics in WINDOW and write all descriptors for all frames contained in catalogue cat.

**Actions taken:**

1.

**Related commands:**
FLAT/AVER OUT_name = IN_list κ,N_{smooth} noise-mode window offset

**Purpose:**

Extract a flat field from a stack of images, *i.e.* remove objects, cosmics *etc.* by sort of median filtering across the images for each pixel. Pixels with values outside the range given by descriptor LHCUTS(5,6) are ignored in the process. If LHCUTS(5,6) are not present LHCUTS(3,4) are used instead. Individual images are normalized by their average level (as measured within FR_AREA). Thus it needs STAT/BACK be run over all input files. For option X, which is the default, this is done internally when calling FLAT/AVER.

In this case, the cuts for calculating the statistics are governed by the default option of STAT/BACK. If the cuts are to be set by the user STAT/back has to be called by the user with the cuts specified and then FLAT/AVER has to be called with the R option.

Averaging dark frames does not work that way, thus the corresponding command DARK/AVER should be used, which normalizes images by means of the exposure time.

The output of both commands is the sum over all input frames!

**Parameters:**

- **p1 out_name** name of output images. An rms-frame with extension .rms holding the scatter in the normalized input images as used for each pixel will also be created.
- **p2**
- **p3 IN_list:** list of input frames (see page[195]). Currently the number of input images is limited to 100.
- **p4 κ,N_{smooth}** κ = limit for κ × σ cut
  N_{smooth} diameter of median-filter box in pixels. Valid values are 1, 3 or 5.
  Default: 5,3.
- **p5 noise mode**
  T theoretical noise (based on RON and EPC from keyword CCD_PARA)
  X local background noise calculated in FR_AREA
  F as X but noise is taken from descriptor FR_STAT(3)
  M as X but use mean value from all frames
  R repeat mode, *i.e.* level and background noise has already been calculated and is taken from FF_NORM(3)
  P use pixel-to-pixel noise from rms-frame
  epc,r numerical values for EPC and RON (in counts!)
  A minus sign in front of the letter specifying the noise mode gives low weight for first image in input.
  Default: X
- **p6 window for noise determination**
  Specifies the window on the frame in which the statistics for the noise determination is calculated. The coordinates are given as X_l, Y_l:X_u,Y_u (in world coordinates).
  The window may also be specified as CENTER, which takes the inner 40% of the image sides, *i.e.* 16% of the image area as the window or as WINDOW, which uses the area as specified by the descriptor FR_AREA for each frame individually. If the descriptor FR_AREA is not present the keyword is taken.
  Default: WINDOW
- **p7 bias level to be subtracted**
  The specified bias-level is subtracted from all images before the filtering is done. If letter O is specified as input, the level of each frame is adjusted by the value as given in descriptor FLAT_BKG (taken to be zero if descriptor is not present).
  Default: 0
Examples:

\texttt{FLAT/AVER fringeI = prep:}  This will create a fringe pattern for the additive flat field correction using all images in catalogue prep with the standard settings (which should be okay for most applications!). It is thus assumed that the images are already bias-subtracted. If the images are sufficiently dithered the resulting frame should be free of objects and cosmic events.

Actions taken:

1. Only pixels in the range specified by descriptor \texttt{lhcuts(5),(6)} are taken into account. These have to be set by the user. If they are not present, \texttt{lhcuts(3),(4)} are used.

2. Corresponding \texttt{rms}-image is created

3. Descriptor FR,STAT is filled

4. Cuts are set

Related commands:

\texttt{CORRECT/IMA CORRECT/FLIP}
CHAPTER 5. DESCRIPTION OF COMMANDS

CORRECT/IMAGE OUT4 = IN_list FFmult FF_add del/no OUT_cat
CORRECT/FLIP OUT4 = IN_list FFmult FF_add del/no OUT_cat

Purpose:

Does multiplicative and additive flatfield correction according to

$$\text{out} = \frac{1}{\text{flat_mult/mult_norm}} \times [\text{in} - (\text{flat_add} - \text{add_norm}) \times (\text{in_norm} - \text{flat_bkg})/\text{add_norm}]$$

The value for $\text{flat_bkg}$ is taken from descriptor FLAT_BKG (set to 0, if not present). If descriptor PA_IMAGE specifies a non-zero angle between the image’s Y-axis and the direction to North, the images are rotated in steps of 90 deg to bring each image as close as possible to the astronomical standard orientation (North up, East to the left) without rebinning.

The FLIP version does a flip around the image’s Y-axis, before the optional rotation.

Parameters:

- $p_1$ OUT4: 1st four characters for output frame, e.g., CCCC = AAAA001,AAAA002,... will produce CCCC001,CCCC002,...
- $p_3$ IN_list: list of INPUT frames (see page 195).
- $p_4$ FF_mult: multiplicative flatfield frame
- $p_5$ FF_add: additive flatfield frame
- $p_6$ del/no: delete, don’t delete input frames after correction [default = nodelete]
- $p_7$ OUT_cat: OUTPUT catalogue. If given the output images are appended to this catalogue.

Examples:

CORRECT/IMA corr = prep:1-5 dusk fringe

Actions taken:

1. Normalization is done via descriptor FR_STAT(1) which needs to be present for the input and calibration frames.
2. The calibration frames must have proper descriptors DATA_LIMIT and CCD_PARA! Saturated values or values at or below DATA_LIMIT in input image and/or calibration frame(s) are flagged in the output image accordingly.
3. Descriptor OLDCOORD holds old START and provides the angle by which the image was rotated.
4. PA_IMAGE is updated.
5. Transformation of START and STEP to world-coordinates [arcsec] such that optical axis is at $[0,0]$.
6. CORRECT can be un only once on a given image!

Related commands:

PREP/CCD   FLAT/AVER
FIND/RING OUT_root = IN_list centre parameter flag

**Purpose:**
Model background in polar coordinates, appropriate for ring-like structures. Applications are *e.g.* rings due to night-sky lines in FPI images.

**Parameters:**

- $p_1$ out_root: first four characters (root) for name(s) of output images
- $p_2$ =
- $p_3$ IN_list: list of input frames (see page[195](#))
- $p_4$ centre: coordinates of symmetry centre in world-coordinates [default: 0,0].
- $p_5$ kappa, win_Y, win_Y: clipping parameter, window width in pixels [default: 2,10,10]
- $p_6$ subtraction flag: If set to *subtract*, the model will be subtracted from the input image and the background set to FR_STAT(1) [default: no_sub].

**Examples:**

```
find/ring ring = corr: cen=-100,300
```
Model background in all images of catalog *corr*. Centre of symmetry is set to [-100,300]. Output names will be ring...

**Actions taken:**

1. Transform image to polar coordinates
2. Model background
3. Transform back to cartesian coordinates

**Related commands:**
remove/ring subtract/ring
REMOVE/RING OUT_root = IN_list ring_model

Purpose:
Subtract the ring model from all input images, scaled by FR_STAT(1).

Parameters:
- $p_1$ out_root: first four characters (root) for name(s) of output images
- $p_2 = \phantom{\text{IN_list}}$
- $p_3$ IN_list: list of input frames (see page 195)
- $p_4$ ring image: Name of ring model

Examples:
- remove/ring flat = corr: my_ring
  Subtract my_ring from all images of catalog corr. Output names will be flat...

Actions taken:
1. Subtract scaled model from input images

Related commands:
find/ring subtract/ring
SUBTRACT/RING OUT_root = IN_list ring_model

Purpose:
Subtract the corresponding ring model from each input images. Add FRSTAT(1) of input image to output image.

Parameters:
- \( p_1 \) out_root: first four characters (root) for name(s) of output images
- \( p_2 = \)
- \( p_3 \) IN_list: list of input frames (see page 195)
- \( p_4 \) root: root for ring models

Examples:
subtract/ring flat = corr: ring
  Subtract ring... from each images of catalog corr. Output names will be flat...

Actions taken:
1. Subtract model from input image
2. Add F_STAT(1) to keep background level

Related commands:
find/ring subtract/ring
CHAPTER 5. DESCRIPTION OF COMMANDS

cosmic/median OUT_root = IN_list thresholds median level window slope

**Purpose:**
Remove cosmic ray events in a series of dithered images by calculating a median image, with which each pixel is compared. Deviating pixels are replaced by the scaled median value.

**Parameters:**
- \( p_1 \) **out_name** first for characters for name(s) of output images
- \( p_2 = \)
- \( p_3 \) **IN_list:** list of input frames (see page 195)
- \( p_4 \) **thresholds** In a first pass cosmics are removed with threshold 1. Then in a second pass pixels are corrected with threshold 2, but only those which are adjacent to pixels already corrected during the first pass. Default values: 4., 2.5
- \( p_5 \) **median output** Name of the median output image, from which the map of pixels to be corrected is derived. Default: MEDIAN
- \( p_6 \) **action level** \( A \) (= default) will correct all images, \( X \) will calculate only the median image, \( list \) will correct a subset of images. The subset is to be given with respect to the input list!
- \( p_7 \) **window** The cosmic removal is done only within the specified window (world coordinates). Default is the whole frame.
- \( p_8 \) **slope** Slope of the RMS. Default: 1.

**Examples:**

```plaintext
cosmic/median cosm = corr:1-15 median
```

Use images 1-15 in catalogue **corr** and correct them for cosmics. The reference median image created will be named **median.bdf**

**Actions taken:**
1. Create a median image using **median/clean**
2. Remove cosmics with **cosmic/refer** using the median image as a reference image.
3. Create a mask (root **cmsk**) of all corrected pixels.

**Related commands:**

**cosmic/refer**
cosmic/refer OUT_root = IN_list reference_frame thresholds slope badlimit

**Purpose:**
Remove cosmic ray events with respect to a reference image, with which each pixel is compared. Deviating pixels are replaced by the scaled reference value.

**Parameters:**

- $p_1$ **out_name**: first for characters for name(s) of output images
- $p_2$: 
- $p_3$ **IN_list**: list of input frames (see page 195)
- $p_4$ **reference**: Name of the reference image, from which the map of pixels to be corrected is derived.
- $p_5$ **thresholds**: In a first pass cosmics are removed with threshold 1. Then in a second pass pixels are corrected with threshold 2, but only those which are adjacent to pixels already corrected during the first pass.
  Default values: 4., 2.5
- $p_6$ **slope**: Slope of the assumed RMS relation. Default = 1.
- $p_7$ **bad_limit**: data_limit, i.e. pixels with values less than this value are ignored. Default = -99999.

**Examples:**

**Actions taken:**

1.

**Related commands:**

cosmic/median
CHAPTER 5. DESCRIPTION OF COMMANDS

FRAMETAB/CREATE table_name = IN_list format

Purpose:
Create a frame table. This table is a useful tool to handle large data sets like in mosaics for surveys. It contains one entry per image with the following columns:

- :frame C*64 name of the frame (only first 32 characters displayed!)
- :filt C*4 filter designation
- :camp C*4 campaign YYMM
- :pos C*3 A3 sub field (in case of HIROCS e.g. 01a or A_a for WFI)
- :texp R*4 F6.1 exposure time [sec]
- :RA R*8 F8.4 right ascension [degree]
- :DEC R*8 F8.4 declination [degree]
- :seeing R*4 F6.2 seeing FWHM [arcsec]
- :rscale R*4 F6.2 relative signal strength (large for shallower images)
- :frstat I*4 I6 background level (descriptor fr_stat)
- :depth R*4 F5.2 relative depth of the image (in terms of S/N)
- :F C*1 A1 flag for excluding frame (s = seeing, t = transparency, i = instrument)

This set of columns is the basic set and should not be changed. It is, however, up to the user to add additional columns as needed.

Parameters:

- $p_1$ out_name Name of output table
- $p_2$ =
- $p_3$ IN_list: List of input frames (see page 195). It may contain one level of a subdirectory.
- $p_4$ format Specifies the format for the RA and DEC columns. deg gives the display always in decimal degrees, hms (=default) lists RA in hh mm ss and DEC in dd mm ss. Note that the internal values stored are always in decimal degrees!

Examples:

```
frametab/cre frames_03hA = cat: Create the table frames_03hA with entries for all images in catalogue cat.
```

Actions taken:
1. Create table
2. Create columns
3. Fill columns 1...7, partly information is taken from frame name.

Related commands:

frametab/fill frametab/cat
FRAMETAB/FILL table_name = IN_list column_specifier

**Purpose:**
Fill the columns of a frame table. As the necessary information may be available only after various reduction steps, the command may have to be issued several times during the reduction process.

**Parameters:**
- \( p_1 \) `out_name` Name of output table
- \( p_2 = \)
- \( p_3 \) `IN_list:` List of input frames (see page 195)
- \( p_4 \) `column specifier` The following are possible: `seeing rscale frstat depth`, one at a time, or `all` to fill all columns in one call. If `rms` is specified a non-standard column `:rms` is created and filled with the rms [arcsec] from find/move. In case of multiple detectors this is the maximum rms found.

**Examples:**
```
frametab/fill frames_03hA = cat: see
```
Fills the column `:seeing` in table `frames_03hA` with values for all frames listed in `cat`.

**Actions taken:**
1. Copy descriptors to table.

**Related commands:**
`frametab/create`  `frametab/cat`
FRAME/TAB/CATLOGUE  cat_name = frame_table root flag short

**Purpose:**
Create a catalogue for those entries in the frame table, which are selected and for which no flag is set. The root of the frames in the table is set to *new root*. An image catalogue is created for FRAME/ICAT, and a table catalogue for FRAME/TCAT.

**Parameters:**

- $p_1$ **catalogue_name** Name of catalogue to be created.
- $p_2$ =
- $p_3$ **frame_table** Name of frame table to be used.
- $p_4$ **root** Root of the frames to be contained in the new catalogue.
- $p_5$ **flag** Flag to indicate if path in frame name should be eliminated for catalogue. This is necessary, if only a single frame table containing the path in the frame name is to be used locally, *i.e.* the frame table is in the directory above the current (local) one and only images in the current directory are to be entered into the catalogue. Default: *local*.
- $p_6$ **short** If set to *no_short* the "s" in the file name for short exposures is replaced by " .". Default: *short*

**Examples:**

- `sel/tab frames_03hA :seeing.lt.1.5`

- `frametab/cat testcat = frames_03hA Cmsk path`
  Creates catalogue *testcat* for those frames with seeing < 1.5 contained in *frames_03hA* for the mask frames *Cmsk*... including the path name in the catalogue.

- `frametab/cat testcat = ../frames_03hA gnom`
  Use frame table in the directory above the current one and create a catalogue of the selected entries .AND. existing in the local path. No path is entered into the catalogue.

**Actions taken:**

1. Create a catalogue of a subset of entries from a frame table.
2. Number of entries in catalogue is stored in keyword OUTPUTI(1).

**Related commands:**

- `frametab/create`
- `frametab/fill`
FRAMETAB/F2B  cat_name

**Purpose:**
Transform a frame table, whose entries are FITS files, to BDF. Similarly FRAMETAB/B2F does the reverse change.

**Parameters:**
- $p_1$ catalogue_name Name of catalogue to be created.

**Examples:**
- frametab/f2b testcat
  Transform names in catalogue testcat from FITS to BDF.

**Actions taken:**
1. Reads all frame names from a frame table.
2. Extensions .FITS are changed to .BDF.
3. Names are copied back to table
4. Flagged frames are included in the transformation.

**Related commands:**
frametab/create frametab/fill
FRAMETAB/FLAG frame_table = flag

Purpose:
Set the flag for the selected entries. Selection has to be done before calling this command.

Parameters:
\[ p_1 \text{ frame_table} \quad \text{Name of frame table.} \]
\[ p_2 = \]
\[ p_3 \text{ flag} \quad \text{One character flag indicating, why this frame is to be excluded from further analysis. Recommended flags are:} \]
\[ s \quad \text{seeing beyond limit} \]
\[ t \quad \text{transparency too bad} \]
\[ b \quad \text{background too high} \]
\[ p \quad \text{wrong pointing} \]
\[ i \quad \text{instrumental problem} \]

Examples:
\[ \text{sel/tab frames_03hA :seeing.gt.1.5} \]
\[ \text{frametab/flag frames_03hA = s} \]
Sets the flag in column :F of table frames_03hA to s, to indicate that the selected frames are above the seeing limit.

Actions taken:
1. Set flag in frame table for selected rows.
2. Select all entries in frame-table

Related commands:
frametab/create  frametab/cat  frametab/badflag
FRAMETAB/BADFLAG frame_table = bad_frames_cat flag I/T

Purpose:
Set the flag for the entries contained in the input catalogue (usually a bad_frames catalogue).

Parameters:

\[ p_1 \text{ frame_table} \] Name of frame table.
\[ p_2 = \]
\[ p_3 \text{ catalogue} \] Name of catalogue. Default: bad_frames:
\[ p_4 \text{ flag} \] One character flag indicating, why this frame is to be excluded from further analysis. Recommended flags are:

- s seeing beyond limit
- t transparency too bad
- b background too high
- p wrong pointing
- i instrumental problem

\[ p_5 \text{ ima tbl} \] Image or table catalogue.

Examples:

\[ \text{frametab/badflag frames}_03hA = ? X \]
Sets the flag in column :F of table frames_03hA to X for all images contained in catalogue bad_frames.

\[ \text{frametab/badflag frames}_03hA = \text{my_bad_tab:}1-5 \ Y \ T \]
Sets the flag in column :F of table frames_03hA to Y for tables of entries 1 to 5 in catalogue my_bad_tab.

Actions taken:

1. Set flag in frame table for each entry in catalogue.
2. Select all entries in frame-table

Related commands:

frametab/create frametab/cat frame/flag
FRAMETAB/COPYFLAGS new frame_table = old frame_frame

**Purpose:**
Copy the flags from the old frame table to the new one. Table selection is ignored!

**Parameters:**

- $p_1$ new frame_table Name of new frame table.
- $p_2$ =
- $p_3$ old frame_table Name of old frame table

**Examples:**

```
frametab/copyf frames.03hA = frames.B.B03hA
```
Copies all flags from the table `frames.B.B03hA` to the (larger) table `frames.03hA`.

**Related commands:**

frametab/flag frametab/badflag frame/reset
5.1. CONTEXT: MAIN

FRAMETAB/RESET frame_table

Purpose:
Clear the flag for the selected entries. Selection has to be done before calling this command!

Parameters:

\[ p_1 \text{ frame_table} \quad \text{Name of frame table.} \]

Examples:

\[
\begin{align*}
\text{sel/tab frames_03hA :f.eq."s"} \\
\text{frametab/reset frames_03hA}
\end{align*}
\]

Clears the flag in column :F of table frames_03hA. The selected frames will from now on be included again in catalogues created with frame/cat.

Actions taken:

1. Clear flag in frame table for selected rows.
2. Select all entries in frame-table

Related commands:

frametab/create frametab/cat frametab/flag
FRAMETAB/SHORT frame_table time_limit root

Purpose:
Rename short exposures (:texp<time_limit) according to HIROCS conventions, *i.e.* replace underscore before file counter by ”s”. Only frames which are selected in the table will be renamed. The root in table column :frame will not be changed by this command.

Parameters:

- $p_1$ frame_table Name of frame table.
- $p_2$ time_limit Upper limit for exposure time [default = 100 sec].
- $p_3$ root Root of frames to be renamed, if different from column :frame.

Example:

frametab/short frames_03hA
- Renames all frames with :texp .le. 100.

frametab/short frames_03hA 50.
- Renames all frames with :texp .le. 50.

Actions taken:

1. Change the name of the frame in the table if entry is selected. Root is not changed, however.
2. Rename the file on disk optionally with different root than given in table column :frame.

Related commands:

frametab/create frametab/cat
FRAMETAB/CPDT frame_table = root column descriptor flag

Purpose:
Copy a descriptor to a specified column in the frame table. If the column does not exist it is created.

Parameters:
- \( p_1 \) frame_table: Name of frame table.
- \( p_2 \) =
- \( p_3 \) root: Root for frames from which the descriptor is taken.
- \( p_4 \) table column: Name of the table column to which the values are to be copied. Only numerical column types are supported.
- \( p_5 \) descriptor: Full specification of descriptor. Only one element can be copied at a time, i.e. vectors are not supported.
- \( p_6 \) flag: Flag to indicate if path in frame name should be eliminated for catalogue. This is necessary if only a single frame table containing the path in the frame name is to be used locally, i.e. the frame table is in the directory above the current (local) one and only images in the current directory are to be entered into the catalogue. Default: local.

Example:
\[
\text{frametab/cpdt frames_03hA = corr :dec o_pos/d/2/1}
\]
Copies \( \text{O_POS(2)} \) for all frames into column :DEC.

Actions taken:
1. Copy descriptor to keyword and then into table column.
2. Table selection is checked and copy is done only for selected rows. Copy is also done only for frames not flagged!

Related commands:
frametab/create  frametab/cat  frametab/cpkt
FRAMETAB/CPKT frame_table = frame column keyword type flag

Purpose:
Copy a keyword to a specified column and specified file into the frame table. If the column does not exist it is created.

Parameters:

- $p_1$ frame_table: Name of frame table.
- $p_2 =$
- $p_3$ root: Frame name
- $p_4$ table column: Name of the table column to which the values are to be copied. Only numerical column types are supported.
- $p_5$ keyword: Keyword name, including index if applicable. Only one element can be copied at a time, i.e. vectors are not supported.
- $p_6$ keyword: Keyword type, being one of R, I, D [default: R]
- $p_7$ flag: Flag to indicate if path in frame name should be eliminated for catalogue. This is necessary, if only a single frame table containing the path in the frame name is to be used locally, i.e. the frame table is in the directory above the current (local) one and only images in the current directory are to be entered into the catalogue. Default: local.

Example:

```plaintext
frametab/cpkt frames_03hA = flatH___22h01a_02k0309_0567 outputr(3)
```

Copies OUTPUTR(3) into column :DEC for row, which holds flatH___22h01a_02k0309_0567 in column :frame.

Actions taken:

1. Copy keyword into table column and row.
2. Table selection is checked and copy is done only for selected rows. Copy is also done only for frames not flagged!

Related commands:

frametab/create  frametab/cat  frametab/cpkt
FRAMETAB/RENAME new_root = frame_table old_root flag

**Purpose:**
Copy or move (depending on the flag) selected frames. This is useful for short exposures, for which no cosmic correction is done, but which have to have the correct root.

**Parameters:**
- $p_1$ new root Root for frames to be created.
- $p_2$ =
- $p_3$ frame_table Name of frame table.
- $p_4$ root Root for source frames.
- $p_5$ copy flag Flag to indicate copy [cp] or move [mv]. Default is cp

**Example:**
- `frametab/rename cosm = frames_03hA corr`
  Copies for all frames selected in frames_03hA the corr-frames to cosm-frames.
- `frametab/rename cosm = frames_03hA corr mv`
  Same as above, but frames are moved.

**Actions taken:**
1. Copy or move the selected frames.
2. Append date/time and action to descriptor HISTORY of the output frame.

**Related commands:**
- `frametab/create`
- `frametab/cat`
FRAMETAB/STATISTICS frame_table short_limit

Purpose:
Print statistical information of the selected entries in a frame table which are not flagged.

Parameters:
\[ p_1 \text{frame_table} \quad \text{Name of frame table.} \]
\[ p_2 \text{short_lim} \quad \text{Limiting exposure time for short exposures in seconds [default: 100].} \]

Example:

frametab/stat frames_03hA
Statistics for frames_03hA

Actions taken:
1. Only selected entries that are not flagged are included in the statistics.
2. The following information is printed (stored in keywords):
   Number of short images \( \text{OUTPUTI}(1) \)
   Number of long images \( \text{OUTPUTI}(2) \)
   Sum of exposure time in long exposure \( \text{OUTPUTR}(1) \)
   Minimum and maximum of seeing \( \text{OUTPUTR}(2), \text{OUTPUTR}(3) \)
   Minimum and maximum of depth \( \text{OUTPUTR}(4), \text{OUTPUTR}(5) \)
   Minimum and maximum of RA \( \text{OUTPUTD}(1), \text{OUTPUTD}(2) \)
   Minimum and maximum of DEC \( \text{OUTPUTD}(3), \text{OUTPUTD}(4) \)

Related commands:
frametab/create    frametab/cat
SET/REFPOS TAB_name = HIROCS_field [RA DEC]

Purpose:
Defines the reference position, which is used to calculate the gnomonic positions. The $\xi, \eta$-coordinates in the sky-survey tables from 2MASS or SDSS are either already calculated for this reference position. Alternatively they can be calculated by `PM:radec2xy tab table name`.

To ensure that the same reference position is used for all HIROCS fields only the name of the field is specified. For other fields RA and DEC can be provided in degrees. Valid field names for HIROCS are 03hA, etc., for COSMOS it is 10h. For other applications give `other` and specify RA and DEC.

Parameters:

$p_1$ tab_name Full name of table, usually a survey table from SDSS or 2MASS.

$p_2 =$

$p_3$ HIROCS field Designation of HIROCS field, e.g. 03hA or other

$p_4$ RA RA of field in degrees, if $P_3$ is other.

$p_5$ DEC DEC of field in degrees, if $P_3$ is other.

Examples:

```
set/refpos 2MASS_03hA = 03hA
```
Write descriptor REF_POS to table 2MASS_03hA with coordinates defined in the PRG.

```
set/refpos 2MASS_03hA = other 150.123 -0.56
```
Write descriptor REF_POS to table 2MASS_03hA with RA = 150.123 and DEC = -0.56.

Actions taken:

1. Write descriptor REF_POS to input table
2. Equinox is always set to 2000.

Related commands:

find/move    mosaic/gnomima
FIND/OBJECTS IN_list Nmax keep-flag PSF-flag

Purpose:
Find stars (objects) in frames to determine the relative orientation of dithered images with FIND/MOVE. Determine PSF from average stellar image. The background in the frame has to be flat (use flat/back first).

Parameters:

\[ p_1 \text{ in\_list} \] List of images to be searched (see page 195)

\[ p_2 \text{ Nmax} \] Maximum number of objects required [200]

\[ p_3 \text{ flag} \] Keep or overwrite the image with the average profile [over]. In the default case the name of the average image profile is profile. If the flag is set to keep, the image root is prof and the profile frames are kept for later inspection.

\[ p_4 \text{ PSF-flag} \] If set to PSF the descriptor PSF\_MEAN will be written in addition to PSF\_FIND to the image. An existing PSF\_MEAN will thus be overwritten! Default is no-psf

Examples:

\[
\text{find/obj cat: 500 keep psf}
\]
Search all frames listed in catalogue cat and set threshold such that about 500 objects are detected. The average profile images are kept for each individual frame with name prof... Descriptor PSF\_MEAN is written/updated.

Actions taken:

1. Requires descriptors FR\_STAT (calculated for whole frame, if not present), CCD\_PARA, and DATA\_LIMIT (set to 0 if not present).

2. Create a MIDAS table in MARK format with object parameters.

3. An average image is created from all unsaturated stellar images found in the table. The PSF is found via an intensity-weighted Gaussian fit to this profile.

4. Descriptors N\_OBJ, PSF\_FIND [e-folding pixels], PSF\_MEAN (if desired) and SEEING [FWHM arcsec] are written (uses fit results, not moments).

5. If less objects are found than desired a warning is issued. If less than 1/3 of the objects desired are actually detected, a warning is issue and the object is entered into that catalogue bad\_frames with the warning "Not enough objects".

6. If more than 50% of the objects detected are removed due to bad PSF, a warning is issue and the object is entered into that catalogue bad\_frames with the warning "Too many eliminated".

The columns of the output MIDAS table have the following meaning (column name and units are still the same as in a MARK table (see page 96)!):
5.1. CONTEXT: MAIN

```
:NR    #
:LABEL s(tar) n(on-stellar) c(osmic)
:ID   1
:MG_O arcsec relative magnitude
:DX_LINE steps position angle from +Y-axis [°]
:DY_LINE steps empty
:X_MARK arcsec position [units given by step]
:Y_MARK arcsec position
:X arcsec empty
:Y arcsec empty
:MAG mag empty
:FI_1 any FWHM_a [units given by step assumed arcsec]
:FI_2 any FWHM_b (elongation = a/b)

Related commands:
find/move undistort/tab find/sex
```
UNDISTORT/TABLE image_list

**Purpose:**
Use the distortion parameters in descriptor DISTORT to correct the coordinates in the MARK table associated with the image (as created by find/obj) for distortion (optical aberrations, differential refraction etc.). The resulting coordinates are the gnomonic projection coordinates. This command has to be executed before the call to FIND/MOVE with a survey table as reference.

**Parameters:**

- \( p_1 \) IN_list: List of input images (see page 195). The associated tables have the same name as the images and have to be in MARK format.

**Examples:**

undist/tab cat: Correct coordinates for all images contained in catalogue cat.

**Actions taken:**

1. Copy columns :X_MARK,:Y_MARK into columns :X,:Y
2. Use descriptors DISTORT, OA_PIX, STEP to calculate the correction to X_MARK,Y_MARK. OA_PIX specifies the pixel position of the optical axis for each frame.
3. Store new values for X_MARK,Y_MARK (≡ ξ,η).

**Related commands:**

find/obj
5.1. CONTEXT: MAIN

FIND/MOVE reference_frame IN_list select max_offset

projection_parameters elimination_parameters

Purpose:

Determine relative position of dithered images. For multi-detector images like WFI this is done separately for each detector.

Parameters:

\( p_1 \) ref_image  
Reference image / table to which all images are to be aligned.

In the default case, this parameter specifies the reference image. The corresponding tables, created by find/obj (in MARK format) have to exist and are searched for via the image name. However, to allow treatment of mosaics, also a table, holding the gnomonic positions of reference objects, can be specified. The existence of descriptor TAB\_FORM in the reference table signals that the table is not in MARK-format but comes from a sky survey like 2MASS or SDSS. In this case no corresponding frame is needed. The content of TAB\_FORM specifies the survey band to be used as magnitude information. The column with label as specified in TAB\_FORM must exist. The reference table in this case must contain descriptor REF\_POS as well as columns :eta and :xsi. REF\_POS specifies the coordinates of the pointing centre, for which \( \eta \) and \( \xi \) are calculated with \texttt{PM:radec2xy}\_tabtable. The latter assumes the existence of the columns :RA and :DEC in the reference table.

\( p_2 \) IN_list:  
List of input frames (see page 195), may also contain reference image

\( p_3 \) select  
To select only stars this is set to ”s”. Default is all

\( p_4 \) offset  
Maximum offset [arcsec] [default: 0]. In the default case the old functionality is preserved, i.e. XY\_MOVE is calculated strictly from the coordinates X\_MARK, Y\_MARK. O\_POS is not used.

If this parameter is > 0 it gives the maximum deviation between O\_POS and the true pointing centre (i.e. search radius for alignment). This speeds up the search for positional match in case of large fields of view. If coordinates in the header are quite accurate a value of 60 is appropriate. If coordinates are bad, use a larger value, e.g. 300.

Optionally, a second value may be added to this parameter (separated by a comma), specifying the maximum allowed change to descriptor O\_POS (see below). O\_POS is adjusted only, if the change is less than this second value. To switch the adjustment of O\_POS off completely, this value should be set to 0.

\( p_5 \) projection parameters  
magnification factor, its tolerance, rotation angle, its tolerance, [defaults: 1.0,0.02,0.0,0.4].

If the magnification and rotation angle are known, they can be entered here with the tolerance set to 0. This speeds up the processing significantly. Please note, that the angle from descriptor PA\_IMAGE is added internally to the rotation angle entered in this parameter. This allows to operate this command on sets of images with varying position angle.
CHAPTER 5. DESCRIPTION OF COMMANDS

Elimination parameters

Minimum rms change, minimal rms, maximal rms, remove fraction [defaults: 0.04, 0.06, 0.1, 0.5].

The number of matches between reference and object table could be more than the number of object in either table as multiple matches can occur. Spurious matches are eliminated via their influence on the rms. In a loop the match is isolated which diminishes the rms most. Removal of matches is governed by the following three parameters, which should be changed only in emergencies:

- Remove entry if fractional rms change is larger than \texttt{min\_rms\_change}.
- If initial rms is larger than \texttt{min\_rms}.
- If rms is larger than \texttt{max\_rms}.

\texttt{rm\_frac}:

Fraction of initially found matches, which may be removed by above process. If this number is exceeded, the removal loop is stopped and the remaining matches are used to calculate the transformation.

Examples:

\begin{verbatim}
find/move refima cat: Use image refima as a reference and find orientation of images in cat with respect to it.
find/move 2MASS_03hA cat:1-3 Use table 2MASS_03hA as reference and find orientation of first three images in cat with respect to it. The tables corresponding to the images in catalogue cat have to be undistorted (see page 68) before this call! This is the preparatory step for mosaic/gnomima (see page 74).
\end{verbatim}

Actions taken:

1. Only objects selected in the reference table and in the input table are taken into account for the analysis.

2. From the reference table only objects with world coordinates in the area of the image are selected for the match, \textit{i.e.} there must be an overlap in area between the image considered and the reference image.

3. Write descriptors \texttt{XY\_move, ref\_frame, ref\_tab, rel\_scale, lin\_scale, angle, n\_match, rms\_arcs, refpos}. For multiple detectors, these descriptors are named \texttt{XY\_move\_ii etc.} with \texttt{ii} = detector number.

4. Update \texttt{O\_POS} in case of large discrepancy (\(\approx 30\)\arcsec). Iterate in this case with \texttt{OFF\_MAX} = 60.

5. Write correction to \texttt{O\_POS} (dRA and dDEC) to descriptor \texttt{OUTPUT\_R(1,2)}. This enables the following procedure to be used in a PRG for a large set of images: Run \texttt{find/move} for the first file in a set with a large value for P4 (maximum offset), \textit{e.g.} 300\arcsec, which is rather slow. This gives the offset for \texttt{O\_POS} for this set. Then use @\texttt{PM:correct\_OPOS} together with keyword \texttt{OUTPUT\_R(1,2)} to correct \texttt{O\_POS} for the rest of the images in the set. Then run \texttt{find/move} with a smaller value for P4 (\textit{e.g.} 60\arcsec).

6. Column :ID is set to 2 for matching objects and to 0 for objects removed from the match process.

7. The following columns are appended to the input table:

- \texttt{mag} magnitude difference \(\text{mag(reference frame)} - \text{mag(current frame)}\)
- \texttt{x\_proj, y\_proj} input coordinates projected onto reference frame
- \texttt{dx, dy} difference input coordinates – transformed reference position
- \texttt{detector} for multiple detectors, the detector number on which the object is located.
8. If keyword TEST_KEY/I/1/1 \( n \) with \( n \) being the chip no. is found and \( n \) is not 0 a table TEST_TBL is created with 6 columns, which hold the \( X \), \( Y \) and \( \text{mag} \) for the reference frame and \( X \), \( Y \), and \( \text{mag} \) for the current frame for detector \( n \). An existing TEST_TBL is overwritten.

Descriptor written to TEST_TBL:

\( \text{NMATCH0} \): number of matches found
\( \text{RMS0} \): rms at start, before objects were eliminated.
\( \text{RMS\_RATIO} \): relative change in rms
\( \text{RMS\_CHANGE} \): Input, determines object elimination
\( \text{RMS\_MIN} \): reduced rms
\( \text{MAX\_RMS} \): Input determines object elimination
\( \text{NMATCH} \): number of remaining matching objects
\( \text{RMS} \): final rms
\( \text{NREMOV} \): number of removed objects
\( \text{RAoff} \): offset in RA
\( \text{DEoff} \): Offset in DEC
\( \text{O\_POS1} \): position of frame
\( \text{tr} \): transformation matrix: frame to reference frame
\( \text{tri} \): inverse transformation matrix
\( \text{dRA} \): correction to RA
\( \text{dDE} \): correction to DEC

Related commands:

find/obj  undistort/table  mosaic/gnomima
SUM/IMA OUT_name = IN_list fill

Purpose:
Calculate sum of images contained in list. The summation is on an integer pixel basis. The output dimension is the common area of all frames.

Parameters:

\[ p_1 \text{ out\_name} \quad \text{Full name of output image} \]
\[ p_2 = \]
\[ p_3 \text{ IN\_list:} \quad \text{list of input frames (see page 195)} \]
\[ p_4 \text{ FILL} / \text{NO\_FILL} \quad \text{For round images only: Fill in or not the missing parts.} \]

Examples:

\[ \text{sum/ima sum}_03hA\_B = \text{cosm}_03hA\_B: \]

Summation of all images contained in catalogue cosm_03hA_B. Name of the output image will be sum_03hA_B.

Actions taken:

1. Use information about relative orientation of the frames from descriptor \textit{XY\_MOVE} to set the descriptor \textit{START}.
2. Summation of frames with \textit{MIDAS} command \textit{compute/image}
3. Takes care of \textit{MPIAPHOT} descriptors like \textit{CCD\_PARA}.
4. Undefined pixels (below frame specific \textit{DATA\_LIMIT} or above frame specific \textit{CCD\_PARA(5)} = global saturation) are set to \textit{DATA\_LIMIT} and \textit{CCD\_PARA(5)} respectively from the current keyword values. The corresponding sum descriptors are updated.
5. Total exposure time (\textit{EXP\_TIME}) and new dynamic range (\textit{CCD\_PARA(4)}) are calculated and stored in the descriptors of the sum frame.

Related commands:

mosaic/sumup
flat/back OUT_name = IN_list smoothing_length window flag statistics range

**Purpose:**
Smooth the background of an image and subtract it. Original count level is preserved.

**Parameters:**
- \( p_1 \) **out_name**: first for characters for name(s) of output images
- \( p_2 = \)
- \( p_3 \) **IN_list**: list of input frames (see page 195)
- \( p_4 \) **length**: Smoothing length in pixels. Default: 500.
- \( p_5 \) **window**: Window in MIDAS convention. Default: [\(<,><,>\)]
- \( p_6 \) **flag**: If set to delete, the input image is deleted after flattening. Default: keep.
- \( p_7 \) **statistics**: If set to no_stat, the statistics (FR_STAT, LHCUTS) is not re-calculated for the output image after flattening. Default: stat.
- \( p_8 \) **range**: Dynamic range factor for background statistics. Histogram is determined within FR_STAT(1) ± range × FR_STAT(2). Default: 20.

**Examples:**
- flat/back test = input_cat:

**Actions taken:**
1. Needs descriptors **CCD_PARA** (for image saturation) and **FR_STAT** for input images. If descriptor **DATA_LIMIT** is not present in the input image, the keyword is taken instead.
2. Calculates median in areas of the width of the smoothing length, interpolates in these values and rebins them to the original image size.
3. Subtracts the smooth background model and adds **FR_STAT** to preserve the background level.

**Related commands:**
- **PM:O2k_flatten_quadrants** operates on the 4 quadrants of OMEGA200 independently and adjusts their levels to FR_STAT(1).
- **WFI:WFI_flatten_chips** operates on the 8 detectors of WFI independently and adjusts their levels to FR_STAT(1).

For these two special commands please check the PRG for the parameters and their meaning.
MOSAIC/GNOMIMA  OUT_root = IN_list output_step_size

Purpose:
Rebin image into gnomonic projection.

Parameters:

- \( p_1 \) out_root
  First for characters for name(s) of output images

- \( p_2 = \)

- \( p_3 \) IN_list:
  List of input frames (see page 195)

- \( p_4 \)
  New step size of output image, usually smaller than the input step size. [default: 0.2]

Examples:

Actions taken:

1. Use descriptors DISTORT and XY_MOVE to calculate projection. Thus find/move had to be operated on the images before this call!

2. Descriptor FR_AREA is deleted as it cannot be transformed in the general case (rotated image!). As the count levels are the same as in the input image, the descriptor FR_STAT is still valid and should be kept. STAT/BACK will thus not write FR_STAT on the gnomonic output image should it be destroyed somehow!

Related commands:

find/move, mosaic/median, mosaic/rmcosmics, mosaic/sumup
MOSAIC/MEDIAN median_ID = IN_list edges fr_area

Purpose:
Median of images in gnomonic projection. Images of different pointings are inserted into large output image at correct places.

Parameters:

\begin{itemize}
\item \textbf{p}_1 \text{ID} \quad \text{ID for median output frame. File name will be med\_ID for the median image and rms\_ID for the associated rms-image.}
\item \textbf{p}_2 =  
\item \textbf{p}_3 \text{IN\_list:} \quad \text{List of input frames (see page 195)}
\item \textbf{p}_4 \text{edges} \quad \text{Outer corners of the median image. Default:0,0,0,0. In the default case the edges are calculated from the minimum and maximum of the start and end coordinates of all input frames.}
\item \textbf{p}_5 \text{fr\_area} \quad \text{Statistics area of the median image in world coordinates. Default:default, which is central area of 300 × 300′′. In case of extended objects near the centre one has to specify a blank area in the median image.}
\end{itemize}

Examples:

\texttt{mosaic/median test = cat:} \quad \text{Will use all images in catalogue cat and calculate the median frame with name med\_test and the associated rms-frame named rms\_test.}

Actions taken:

1. The median is only calculated for pixels for which 3 or more images are available. Otherwise the pixel is the resulting image is set to the data limit.

Related commands:

mosaic/rmcosmics \quad \text{cosmic/median}
MOSAIC/RMCOSMICS OUT_root = IN_list reference_frame_ID mode kappa1,...,kappa4 percentage

Purpose:
Finds and corrects cosmics in the individual frames (gnomonic projection!). It does usually not affect the areas of objects. The individual frame is compared with the MEDIAN frame. Pixels which deviate more than K1*SIGMA(pix) from MEDIAN(pix) are replaced by the median value. In a second iteration, the neighbourhood of each detected cosmic is analysed with a lower kappa parameter KAPPA2.

Parameters:

- $p_1$ out_name: First four characters for name(s) of output images. Output masks will be named as the output images with characters 2-4 replaced by msk.
- $p_2$: 
- $p_3$ IN_list: List of input frames (see page 193). These are the images rebinned to gnomonic projection by mosaic/gnomima.
- $p_4$ reference frame ID: ID for input reference frame and its rms, i.e. second part of images names for reference frame and its associate rms. Median frame name is med_ID and rms-frame name is rms_ID.
- $p_5$ mode: Execution mode:
  1. [= default] do everything
  2. mask creation only
  3. use existing mask to remove cosmics.
- $p_6$ kappa values: Kappa values for central pixel, surrounding pixels, central pixel in case of a hole, surroundings of hole. Defaults: 4.0,2.5,8.0,5.0
- $p_7$ path for median/rms: If the median/rms images are not in the current directory the relative path could be specified here. Default: . i.e. local directory.
- $p_8$ percentage: Percentage of all pixels tolerated as cosmics. Default: 1.0

Examples:

mosaic/rmcosm cosm = cat: test
Remove cosmis from all images in catalogue cat. Use frame med_test and rms_test as reference images / rms. Output frames will be named as input frames but first four characters replaced by cosm. If the input images started with gnom, the output masks will be named gmsk...

mosaic/rmcosm Cosm = gnom_03h01a:1 high_03hA 2 ? ../
Use the median image ../med_high_03hA (i.e. the median and rms frames are located in the directory above the current path!) and create the cosmic mask from the first image in catalogue gnom_03h01a. Output frame will be named Cmask...
No replacement of cosmics will be done.

Actions taken:

1. If an input frame holds a descriptor BPM_ID, which is a bad-pixel-mask, the bad pixels are added to the cosmic mask. Bad pixels in the bad-pixel-mask have a value of 1, good pixels a value of 0.
   The frame then needs to also hold a descriptor BPM_ROOT (character*4) specifying the root for the frames where descriptor XY_MOVE can be found (the input images are gnomonic images and thus
do no longer have `XY_MOVE`). The bad-pixel-mask must be on the same pixel grid as the original images. Internally for each image the gnomonic projection of the bad-pixel-mask is calculated and added to the cosmic mask (cosmic mask pixel value set to current value + 100).

2. The number of cosmics found is checked against the specified percentage. If for a frame more cosmics are detected, the name of the frame is added to catalogue `bad_frames`. Cosmic correction is nevertheless done. This is a warning only. For details see the Appendix on error treatment on page 197.

Related commands:

`mosaic/median` `mosaic/copycos`
MOSAIC/COPYCOS OUTimage_root = IN_list INPUToriginal_root

**Purpose:**
Remove cosmics in original frames using the information from the image in gnomonic projection.

**Parameters:**
- $p_1$ outima_root: First four characters for name(s) of output images. Output masks are named accordingly with characters 2-4 replaced by msk. Output mask name has to be different from input mask name!
- $p_2$ =
- $p_3$ IN_list: List of cosmic corrected gnomonic input frames (see page 195)
- $p_4$ root for original images: First four characters for name(s) of original images, *i.e.* not rebinned to gnomonic images (usually the flattened, corrected image). This image provides the transformation matrix needed.

**Examples:**
mosaic/copy cosm = cat: corr
For all (gnomonic rebinned, cosmics removed) images in catalogue cat (names start with Cosm...) use information from the associated cosmic masks Cmsk... and remove the cosmic from the original (corrected but un-rebinned) images corr... The output images will be named cosm... and their masks cmsk....

**Actions taken:**
1.

**Related commands:**
mosaic/rmcosmics
MOSAIC/SUMUP OUT_name = IN_list edges weighting

**Purpose:**
Weighted sum of gnomonic images (weight according to depth of image). Also create weight-, PSF- (e-folding pixels!), background- (including weight factor) and rel_scale images.

**Parameters:**
- \( p_1 \) **out_name**
  Full name of output sum image. Should start with \texttt{sum} followed by an identifier. The psf-image will then be named \texttt{psf}..., the weight-image \texttt{wght}..., the background-image \texttt{back}..., and the rel_scale-image \texttt{rsc}....

- \( p_2 = \)

- \( p_3 \) **IN_list:**
  List of input frames (see page 195)

- \( p_4 \) **edges**
  Outer corners of the median image. Default: 0,0,0,0. In the default case the edges are calculated from the minimum and maximum of the start and end coordinates of all input frames.

- \( p_5 \) **weighting**
  Weighting of the images is done according to \((1/rel\_scale)^{W1}/background\_noise^{(2*W2)}/PSF^{W3}\) where the default weighting exponents are 1,1,2 (optimum S/N for point sources, see PhD thesis by Armin Gabasch [4]).
  To turn of weighting completely, set these all to 0.

**Examples:**
mosaic/sumup sum_03hA = cat:10,15-45
Create sum image \texttt{sum\_03hA} of 1 \degree from images 10 and 15 to 45 in catalogue \texttt{cat}.

**Actions taken:**
1. Saturation in sum image is set to \(1 \times 10^9\). Descriptor \texttt{CCD\_PARA} is filled accordingly.
2. Descriptor \texttt{FR\_STAT} is calculated in a 500' central window. If image area is less, the whole frame is taken. Descriptor \texttt{FR\_AREA} is set accordingly.
3. Descriptor \texttt{EXP\_TIME} is the total integration time contained in the sum image.

**Related commands:**
FIND/SEXTRACT OUT_name = IN_list threshold MIN_pix cut/nocut check_image weight

Purpose:
Find objects in the input frame using SourceExtractor. The command converts the input BDF frame to FITS, establishes the search parameters (mainly using descriptor PSF_MEAN or PSF_FIND if not present) and runs SourceExtractor. The ASCII output is then transformed to a MIDAS table with the layout of MARK/DISPL.

Parameters:

- $p_1$ out_name: Root (first 4 characters) of output tables. No default. Full names will be the root and the leading 5 to end characters of the input file appended.

- $p_2$ =

- $p_3$ IN_list: list of input frames (see page 195).

- $p_4$ threshold: Threshold in units of the total S/N-ratio. Default value is set empirically to 4.7.

- $p_5$ min_pix: SourceExtractor parameter. If set to the default = 0, the value of min_pix is adjusted to the seeing of the input frame using PSF_MEAN or PSF_FIND

- $p_6$ cut / nocut: Cutout the edges of the image according to the optical axis and an outer radius to be marked interactively on the graphics screen. Default: notcut

- $p_7$ check image: SourceExtractor produces control images. The default is APERTURE. BACKGROUND may also be used.

- $p_8$ weight/no_weight: SourceExtractor allows usage of a weight image (e.g. a flat field) to weight the object detection according to the sensitivity in the image. The associated image (output of mosaic/sum has to have as root wght. Default: no_weight

Examples:

FIND/SEXTRACT src_ = sum_R: Use all images in catalogue sum_R and find objects in each of them. Output tables are named src_xxxxxx.tbl, where xxxxxx are the trailing characters of the input image names. All other parameters are default.

Actions taken:

1. convert input BDF to FITS
2. setup of SourceExtractor input configuration file (name = image_name.config)
3. run SourceExtractor
4. convert output file to MIDAS table (format of MARK table using CREATE/SEXTRACT)
5. copy descriptor SEEING from the input image to the output table.
5.1. CONTEXT: MAIN

Requirements:
The following descriptors have to be present in each of the input images:

- CCD_PARA
- PSF_MEAN or PSF_FIND
- TEL_DIAM
- EXP_TIME
- LAMBDA
- REF_POS

The error message for missing REF_POS might be ignored if the output is not used for the master table. For details on these see Section 13.

Related commands:
CREATE/SEXTRACT
CREATE/SEXTRACT OUT_name = IN_list threshold cut/nocut mode

Purpose:
Use the output from SourceExtractor to create a MIDAS table with the layout of MARK/DISPL. This is convenient to set a new threshold in the object selection without re-running SourceExtractor. However, the object selection will not be exactly identical to using the higher threshold in FIND/SEX directly! Why?

Parameters:

- $p_1$ out_name: Root (first 4 characters) of output tables. No default. Full names will be root + trailing 5 to end characters of the input file.
- $p_2 = \ldots$
- $p_3$ IN_list: list of input frames (see page 195).
- $p_4$ threshold: Threshold in units of the total S/N-ratio. Default value is set empirically to 4.7.
- $p_5$ cut / nocut: Cutout the edges of the image according to the optical axis and an outer radius to be marked interactively on the graphics screen. Default: notcut.
- $p_6$ mode: Normal images are treated differently from the (noise free) master images. Mode is set to STD (=default) for normal images and to MASTER for master images.

Examples:
CREATE/SEXTRACT src. = sum_R:

Use SourceExtractor output tables for all images in catalogue sum_R and create output tables named src_xxxxxx.tbl, where xxxxxx are the trailing characters of the input image names.

Actions taken:

1. Use result from previous SourceExtractor run
2. Convert output file to MIDAS table (format of MARK table)

The columns of the output MIDAS table have the following meaning (column names are still the same as in a MARK table, units specify table content!):

- :NR #
- :LABEL s(tar) g(galaxy) (from CLASS_STAR limit 0.8)
- :ID 1
- :MAG_0 rel.mag relative magnitude (from MAG_BEST)
- :DX_LINE stellarity stellarity index (0 ... 1000) (from CLASS_STAR)
- :DY_LINE degrees position angle of image shape [°] (from THETAWIN_WORLD)
- :X_MARK arcsec position [units given by step] (from XWIN_WORLD)
- :Y_MARK arcsec position
- :X arcsec empty
- :Y arcsec empty
- :MAG S/N S/N (from MAGERR_BEST)
- :FI_1 arcsec FWHM_a [units given by step assumed arcsec] (from FWHM_WORLD and ELONGATION)
- :FI_2 arcsec FWHM_b (elongation = a/b)

Related commands:
FIND/SEXTRACT
5.1. CONTEXT: MAIN

CHKLRG/SEXTRACTOR IN_list major axis limit, positional tolerance

Purpose:
For each input table objects with major axes larger than the specified limit are checked in all other tables of the input list for objects within the specified positional tolerance limit. Objects without counterpart in one or more of the other tables are flagged as spurious with :ID = "l" (letter l, not number 1!).

Parameters:
- $p_1$ in_list: List of SourceExtractor tables, usually from the summed frames.
- $p_2$ lower limit for major axis, positional tolerance: All objects with FWHM larger than the major axis limit are checked. Both limits are in arcsec. Default: 10.,2.

Examples:
CHKLARG/SEX find: Check all tables in catalogue find for spurious large objects.

Actions taken:
1. All objects are read in. Those with FWHM above the major axis limit are checked for corresponding images in all other tables of the catalogue within the tolerance limit. If they are not present in all tables the :ID is set to "l".

Related commands:
CLEAN/SEXTRACTOR MASTER/IMAGE
CLEAN/SEXTRACTOR SE-table_root = frame step select.1 select.2

Purpose:
Clean a SourceExtractor table from spurious objects around bright stars, along blooming and set FWHM of brightest stars to the PSF of the image.

Parameters:

- $p_1$ root of SE-table first four characters of table name
- $p_2$ =
- $p_3$ frame Name of frame on which the table is based, usually a summed image
- $p_4$ step The cleaning procedure is divided into 3 steps.
  - Step 1 = flagging objects in halos around bright stars
  - Step 2 = flagging largest, spurious objects
  - Step 3 = analyse SE-stellarity as a function of SE-magnitude
  - Step 4 = flag only patches
  - Step 10 = flagging was done separately, do only the table clean-up
  - Step 0 = all together [default]
- $p_5$ select 1 Optional selection for step 1 in addition to :ID.eq."1*", usually a magnitude cut. Default [none]
- $p_6$ select 2 Optional selection for step 2 in addition to :ID.eq."1*", usually a FWHM cut. Default [none]

Examples:
CLEAN/SEX find = sumbR..03hA Flags spurious entries in table findR..03hA.
CLEAN/SEX find = sumbR..03hA 2 ? .and.;FI1.gt.10 Re-do the flagging of largest, spurious objects. Only objects larger than 10″ are displayed. Keep all other flags.

Actions taken:
Column :ID is used as the flag for spurious images.

1. The input table is sorted with increasing magnitudes (column :mag.0) and each object selected (according to :ID.eq."1*"{$p_5$}) is displayed with all objects found in its vicinity marked by red circles. A green cross indicates the centre of the field of view, i.e. should coincide with the object in question. Previously flagged objects are marked by a cyan diamond. The user is prompted if the currently displayed star should be used (i.e. cleaned), skipped, the correct object be marked interactively or the procedure ended (as the brightness level reached does no longer require cleaning). If the star is to be used the cursor is displayed and the user marks the outermost position to which objects should be flagged. The central object (bright star) is kept. The same is true if the object is marked with the cursor. At most the brightest 500 stars can be cleaned. The flag for this step (1) is X. This step can be re-done with $P4 = 1$. Previously cleaned stars are kept. To reset the table see below.

2. The input table is sorted in decreasing major axis (column :FI1). Then successively each selected object (according to :ID.eq."1*"{$p_6$}) is displayed and marked. The user is asked to flag, skip or end. At most 500 objects are displayed. The flag for this step (2) is L. It can be re-done with $P4 = 2$. Previously cleaned stars are kept. To reset the table see below.
3. A plot of SE-stellarity (column :dx_line) as a function of SE-magnitude (column :mag_0) is displayed (see Fig. 4.2). The brightest stars (upper left corner of plot) are selected with the cursor put to the lower right corner of the box occupied by them. The major and minor axes of these objects are set to the PSF values given in descriptor PSF_MEAN. Note: This step cannot be reversed as the table entries are over-written. If necessary, the table has to be re-created via CREATE/SEX. The objects affected are flagged by Y.

4. The stars in the upper central area of this plot are often contaminated by spurious objects along diffraction spikes and blooming. Again the lower right corner of this region is to be marked with the cursor. A histogram of all objects in the range between the magnitude limit marked in the previous item and the mark given now is displayed. It show a strong peak at the position of the PSF and spurious objects with larger major axis. The upper limit of the PSF is to be marked on the graphics screen. Then the lower limit to the PSF is marked to eliminate spurious objects like remaining cosmics etc.. The objects above the upper limit are flagged by Z, those below the lower limit by C. Both will not enter the master table.
The second part of this step can be re-done, as no data are overwritten.

5. The table may contain descriptor rm_patches, specifying how many rectangular areas are to be excluded from the master table. The coordinates [xmin, ymin, xmax, ymax] of these patches are to be provided in descriptor rm_corners. Command PATCH/SEXTRACTOR may be used to write these descriptors interactively. Objects are flagged with W.

6. Due to a bug in SourceExtractor it may happen that well defined stellar images (stellarity index very high) come out with FWHM = 0. Their PSF is set to the values provided by PSF_MEAN of the image and the ID is set to V. This step cannot be reversed!
   If these objects are indeed spurious, do not include those flagged with V into the master list.

Please note the following:
Results for all steps are written to the following descriptors in the table:

- **Step 1:** m_clean (number of objects cleaned),
  - nr_clean (object numbers of cleaned stars),
  - X_clean, Y_clean,
  - rad_clean (cleaning radii).

- **Step 2:** nr_large (number of spurious large objects flagged),
  - m_large (number of objects flagged)

- **Step 3:** bright_psf, stellar_psf (first position marked on graphics screen),
  - bright_spur, stellar_spur (second position marked on graphics screen).

A global reset can be done by `compute/tab table :ID = "1 "` and then deleting the above mentioned descriptors. However, this does not reset the PSFs overwritten for bright stars!

As very bright stars end up with multiple entries along the diffraction spikes it is helpful to perform step 1 only first on a selection of the brightest stars. Then the table is selected for objects with ID .eq. 1 only. Following this one can either iterate this or call CLEAN/SEX with the defaults.

**Related commands:**

FIND/SEXTRACTOR CREATE/SEXTRACTOR
CHAPTER 5. DESCRIPTION OF COMMANDS

PATCH/SEXTRACTOR table_root

PATCH/SEXTRACTOR table_root = frame_list table scale

Purpose:
Interactively define areas on the display in which objects should be flagged in the corresponding Source-Extractor table.

In a second mode, a list of input tables could be patched around the positions specified in a table (created with GET/CURS).

Parameters:

\[ p_1 \text{ table_root} \quad \text{First for characters for name of table. The name of the displayed image is determined automatically and the name of the table to be patched is obtained by substituting the first 4 characters with the root (filled with underscores if less than 4 characters are supplied).} \]

\[ p_2 = \]

\[ p_3 \text{ frame_list} \quad \text{List of images in the same coordinate system, e.g. sum frames. Default: display, no default if a table is specified in } P_4! \]

\[ p_4 \text{ table} \quad \text{Table with positions of e.g. bright stars, whose diffraction spikes should be flagged. The table has the format of the output from GET/CURS table name. Default: no_tab} \]

\[ p_5 \text{ scale} \quad \text{Scale for the image display. Default: 2} \]

Examples:

PATCH/SEX find Assuming that the image sumbR___03hA is displayed the table findR___03hA is patched.

PATCH/SEX find = sums: bright_stars 1
   Loop through all images in catalogue sums and display subsequently all positions from table bright_stars with scale 1 for patching.

Actions taken:

1. The cursor box is displayed and the areas are marked with the left mouse button after adjusting the box width with the cursor keys. To terminate the loop hit the right mouse button twice.

2. Only objects with ID = 1, Y, or V are displayed

3. The number of patches and their corners are stored in table descriptors rm_patches and rm_corners. Each call updates resp. expands these descriptors.

4. To remove all patches delete descriptor rm_patches (and also rm_corners if no patches are to be used in the end).

5. The objects are finally to be flagged with :ID = "W" by a call to CLEAN/SEXTRACTOR (step 4 does only the patching).

Related commands:
CLEAN/SECTRACTOR FIND/SEXTRACTOR
MASTER/SET IN_list root S/N_min scale center def_parameters

**Purpose:**
Set the selection parameters in preparation of the master list creation. The selection parameters control which objects from the SourceExtractor output tables (created with FIND/SEXTRACTOR) are to enter the artificial master image created with MASTER/IMAGE.

**Parameters:**
- \( p_1 \) in_list: List of sum frames to be used for master image
- \( p_2 \) root: Root name (first 4 characters to specify the SourceExtractor tables from the name of the input frames.
- \( p_3 \) S/N_min: Default (starting) value for the S/N of the objects as determined from SourceExtractor [default 4.7].
- \( p_4 \) scale: Scale for loading the image [default: 1]
- \( p_5 \) center: Center of the image as loaded into the display [default: c].
- \( p_6 \) default parameters: Selection parameters for min_FWHM and max_elong [default: 0.8,10].

**Examples:**

**Actions taken:**
1. Iterate S/N and set descriptor SELECTION accordingly

**Related commands:**
MASTER/IMAGE MASTER/TABLE
MASTER/IMAGE field_ID = IN_list edges step selection depression

Purpose:
Create an artificial (noise-free) image based on the results of FIND/SEXTRACTOR. If not present, the empty image is created and objects are additively placed into this at the positions of objects found in the individual summed frames. The amplitude of the objects is according to their S/N and their widths correspond to the values found by SourceExtractor.

Parameters:

\( p_1 \) out_name  
Designation of field (e.g. 03hA). The name of the master image will be Master_field_date. If the master image already exists, it will be used. If you want to change image dimensions the old image has to be deleted first!

\( p_2 \) =

\( p_3 \) IN_list:  
list of output tables from SourceExtractor (see page 195).

\( p_4 \) edges  
Lower left X,Y upper right X,Y [default: -1800,-1800,1800,1800].

\( p_5 \) step size  
Step size of the new image in arcseconds [default: 0.2].

\( p_6 \) selection  
Selection parameters: S/N min, min_FWHM, max_ELONG or descriptor. Only objects with S/N larger S/N min, FWHM [arcsec] larger FWHM min and elongation smaller than max_ELONG are used. Recommended values are 4.7 (def. for SourceExtractor), 75% of the seeing, and 10. If the value for min_FWHM is > 10 it is interpreted as the percentage of the seeing, taken from descriptor SEEING of the table. The selection parameters may be stored for each SourceExtractor table separately in the descriptor SELECTION (see command MASTER/SET on page 87). In this case this parameter has to be set to descriptor. default: 4.7,75,10.

\( p_7 \) depression  
Depression parameters: strength, width. To help disentangle neighbouring objects each object can be put into a broad depression. Depth and width of the depression are governed by these two parameters [0.5,1.225]. Turn this feature off by setting this parameter to 0.

Examples:

Actions taken:
1. Create empty image of size 18000 x 18000 pixels
2. Insert objects from each table, which are selected and whose ID is 1, Y or V, into this image. The selection specified via parameter \( p_6 \) is operating in addition to the selection specified for the table.

Related commands:
MASTER/IMAGE MASTER/TABLE
MASTER/TABLE MASTER_{field}_date

Purpose:
Search the artificial master image for objects and create a master table in MARK-format.

Parameters:
\( p_1 \) master image Full name of master image. Has to be Master_{field}_date

Examples:

Actions taken:
1. Convert input master image to FITS
2. Run SourceExtractor on this image with special config file taken from PM/../tbl
3. Rename output file from Master.dat to \{P1\}.dat
4. Create MIDAS table in MARK-format from this with same name as master image
5. Sort table by :Y_mark and number objects consecutively.

Important:
Selection of subsets must only be done via the select/table command and never by deleting rows!

Related commands:
MASTER/SECTION OUT_name = IN.list master_table

Purpose:
The masterlist contains objects in a coordinate system representing the projection of the positions to the celestial sphere. These have to be transformed into the system of each individual frame, including distortion. At the same instance only objects within a given frame are retained in the output table. Also only objects selected in the master table are transferred to the output table.

Parameters:

\[ p_1 \text{ out}_\text{name} \quad \text{first for characters for name(s) of output tables. Default: MARK} \]
\[ p_2 = \]
\[ p_3 \text{ IN}_\text{list:} \quad \text{list of input frames (see page 195).} \]
\[ p_4 \text{ master}_\text{table} \quad \text{name of the master table} \]

Example:

\[
\text{master/sect e12p = cosmH__03h12c_02k0309_1234}
\]

Takes image \text{cosmH__03h12c_02k0309_1234} and creates output table named \text{e12pH__03h12c_02k0309_1234}

Actions taken:

1. read master table
2. read frame descriptors \text{XY}_\text{MOVE, OA}_\text{PIX} \text{ and } \text{BORDER}
3. transform coordinates and skip objects outside destination frame.
4. determine location of object within frame
5. check selection flag
6. write descriptors REF_IMA and REF_TAB

Descriptors used from input files:

- IDENT
- NPIX
- OA_PIX 2 R*8
- XY_MOVE 6 R*4 or XY_MOVE\text{nn}
- DISTORT 7 R*4
- START 2 R*8
- STEP 2 R*8
- BORDER 4 R*4 or BORDER\text{nn}

Related commands:

master/image   master/set
5.1. CONTEXT: MAIN

CLASS/EVAL sum_list SourceExtractor_table_root thresholds

Purpose:

Run evaluate with a predefined parameter file optimized for morphologically separate stars and galaxies by means of an area-vs.-mag plot. The setup file (clss.eval) needs a guess of the rough seeing in the input sum frames. Therefore, the command requires the proper instrumental setup to be defined in set/cont mpiaphot to provide the correct pixel scale.

However: If the input images are mosaics from WFI, LAICA or OMEGA2k the images must have a descriptor mosaic/1/1/1 set to 1. For these the individual pointings are treated separately as the background (i.e. S/N) and PSF may be different for the sub-images. This means that the classification plot is shown 16 times for OMEGA2k data, and 4 times for WFI and LAICA data. The pixel scale is taken from the descriptors of the (re-binned) summed images. Thus different instruments can be used in one call. However, it is not possible to mix mosaics and single images in one call.

Parameters:

\( p_1 \) input list  The input frames are usually sum frames in a given filter (see page [193]). In case of mosaics, only one image at a time can be processed at the moment. Each has to have descriptor mosaic set to 1.

\( p_2 \) table root  These are usually the output tables from FIND/SEXTRAT. First 4 characters are to be supplied. In case of mosaics, this is the full name of the master table starting with Master... .

\( p_3 \) thresholds  Thresholds for star / galaxy separation.

Th-1: bright end: upper limit for stars is \((1 + Th - 1) \times \text{PSF MEAN}.\)
Default = 0.1

Th-2: faint end: allows for broader PSF of faint stars due to their lower S/N.
Default = 1.0

Th-3: level of stellarity limit.
Default = 80.

Examples:

CLASS/EVAL sum_R: tsex
Use sum images in catalogue sum_R and corresponding SourceExtrator tables, whose names start with tsex for classification.

CLASS/EVAL sum_all: Master_03hA
Use sum images in catalogue sum_all, which are all assumed to be re-sampled on the same grid. As table with object positions use Master_03hA for all sum images for classification.

Actions taken:

1. Run setup/evaluate. The user is prompted for
   - Seeing FWHM in [arcsec]
   - Background from histogram? Y/N. Always use "Y" for normal summed frames, "N" for artificial master frame.
   - Fixed position? Y/N. Always use "Y"

2. A plot of all objects of area vs. mag is shown on the graphic screen. Objects to be classified as stars according to the threshold parameters are marked in red. The user has the option to accept the classification or change the thresholds and re-classify.
3. The output table (named clss...) has the standard evaluate table format. However the following columns are replaced by the classification results:

- column 43: star1000 " " I6
- column 44: star_limit "sq-pixel" F10.2
- column 45: star_area "sq-pixel" F10.2
- column 49: star_lbl " " A8
- column 50: stellar " " F10.3
- column 52: dcon_sx "pixel" F10.2
- column 53: dcon_area "sq-pixel" F10.2
- column 54: dcon_ell " " F10.3
- column 56: S_N_R " " F10.1

Column :RDUMMY_2 is removed and column :classima " " A64 C*64 is added.

**Related commands:**

- evaluate
- CLASS/COMBINE
CLASS/COMBINE Master_list = class_list

Purpose:
Use the results from CLASS/EVAL for all sum images to find the optimum classification for each object. Classification is taken from the image with the largest S/N ratio for a given object. This information is transferred to the master table.

THIS IS NOT IMPLEMENTED YET!

Parameters:

$p_1$ master table  Name of master table.

$p_2$ =

$p_3$ input list  The input tables are the CLASS/EVAL output files on the sum frames.
SET/EVALUATE version mode

Purpose:
Create the parameter file, that determines the way EVALUATE will do its photometry.

Parameters:

- \( p_1 \) Version id
  - Four characters which characterize the version of this mode. Will create output file named `vvvv.eval`

- \( p_2 \) mode
  - The mode, which determines the standard parameters entered into the setup file may be set to PHOTO, POLAR, ASTRO, or MC.

Example of the parameter file created:

```
0 0 0 > (TEST,CLEAN,POL), FIX\_PSF, EFF\_PSF
1 0 14.00 0.00 0.00 0. > SUM: (\\SUM, SUM), -, APERT, PSF: SX, SY, AL
0 20 1.00 124.1650 > FIT: (yes/no), ITER, CHI\_LIM, SATUR/1000.
1 0 3.00 0.0000 > BACK: (FIT,L,HST), HW, S\_LOC, B\_OFF/1000.
1 2 2.87 4.00 0.00 > PROF: (DIRT, EDGE), R\_MIN, X, S\_DIRT, -
1 20 3 0.00 0.00 > MAXI: (FIX,FILT), ITER, RAD, X0, Y0, -
```

```
phot.eval created 4-Mar-96 10:57
```

The parameters have the meaning as described in Table 5.1. Please note that in the setup file all values are in pixels, widths are in e-folding pixels (FWHM = 1.66511 s):

- **TEST,CLEAN,POL**: 001 for Savart images, 100 for test runs
- **FIX_PSF**: 1 for fixed PSF’s, 2 for fixed PSF’s and Savart images
- **EFF_PSF**: 1, if PSF is given below as sx, sy, \( \alpha \)
- **W_SUM,SUM**: 01 sum, 11 weighted sum (circular weight), 22 weighted sum with elliptical weighting function
- **APERT**: radius of fixed aperture [pixels]. If zero variable aperture according to profile analysis and value of X (see below) is used.
- **PSF**: sx [pix], sy [pix], \( \alpha \) [°], for eff. PSF. Only sx = sy, \( \alpha = 0 \) is currently possible.
- **FIT**: 0 or 1
- **ITER**: number of iterations for fit
- **CHI_LIM**: [\%] in general saturation level (> 0) in units of 1000, but:
  - If descriptor `CCD_PARA` is present, the minimum of the latter and this value are used.
  - If descriptor `CCD_PARA` is not present, the minimum of the saturation value preset by `set/cont mpiaphot (= keyword CCD_PARA(5))` and this value is taken.
  - If here saturation level is set to 0, the use of the saturation value from descriptor `CCD_PARA(5)` is forced. If this is missing, the image is skipped! Important for IR-images!
  - If set to \(-1\), saturation is turned off.
- **BACK**: background determination: 100 for fit, 010 for local determination, 001 for determination from histogram, 000 for background taken from descriptor `FR_STAT` (see [10])
- **HW**: halfwidth [pix] of window for background determination, if 0, it will be taken from descriptor in table.

Table 5.1: Parameters used in the `*.eval` table.
5.1. CONTEXT: MAIN

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S_LOC</td>
<td>significance level for local background determination</td>
</tr>
<tr>
<td>B_OFF/1000</td>
<td>offset for background, if 0, it will be taken from descriptor FR_STAT in image</td>
</tr>
<tr>
<td>PROF</td>
<td>10 = removal of dirty pixels, 01 = ?</td>
</tr>
<tr>
<td>R_MIN</td>
<td>minimum extend of image</td>
</tr>
<tr>
<td>X</td>
<td>ratio aperture radius / extend of image</td>
</tr>
<tr>
<td>S_DIRT</td>
<td>significance limit for detection of dirty pixels</td>
</tr>
<tr>
<td>- MAXI</td>
<td>10 = no search of image maximum, 01 = smooth image before image pre-analysis, 20 = allow only small deviation in position.</td>
</tr>
<tr>
<td>ITER</td>
<td>iterations for maximum search</td>
</tr>
<tr>
<td>RAD</td>
<td>radius of aperture [pix] for maximum search</td>
</tr>
<tr>
<td>X0</td>
<td>offset to marked position, i.e. a means to shift the coordinates originally marked (not the result of EVALUATE !)</td>
</tr>
<tr>
<td>Y0</td>
<td>dito. X0 = 1 and Y0 = 1 shift position to upper right in image. Fractional pixels are allowed.</td>
</tr>
</tbody>
</table>

Additional information may be found in references [10] and [?]

Examples:

```plaintext
set/eval phot PHOTO
```

Use PHOTO mode to create file phot.eval

Actions taken:

1. PHOTO Radius of aperture = 1.3× seeing value
   Window for background: half width = 3× aperture radius

2. In case of saturation: aperture = max(pre-set, aperture found)

3. Consistency check for aperture width (non-saturated objects): profile → FWHM → (X) aperture requirement: 0.5 < aperture/presetaperture < 1.5
   ignored if X > 2.0 and/or effective PSF

Related commands:

```plaintext
evaluate/image
```
EVALUATE/IMAGE OUT_root = IN_list table_root Savart Statistic Profile stard,end

**Purpose:**
Determine total counts above background for objects with positions drawn from input table (P4).

**Parameters:**
- $p_1$ out_name: Root (first for characters for name(s)) of output tables
- $p_2$ =
- $p_3$ IN_list: list of input frames (see page 195)
- $p_4$ table_root: Root for names of input tables in MARK format (default: MARK). The trailing part of the table names have to be identical to the name of the corresponding image. MARK format corresponds to a table with columns 1 to 13 as described in table 5.2.
- $p_5$ Savart key: Key to indicate if images taken with a Savart plate are analyzed. Has to be one of SAVART, NOSAVART (default) or COSMICS.
- $p_6$ Statistic key: Key to determine extent of image statistics performed:
  - STAT: some overall statistics (default)
  - PSF: same as STAT but will also fill descriptors with PSF-information
  - ITER: mean values of the PSF of all objects are used for a second iteration.
- $p_7$ profile key: Key to specify if additional table with image profiles should be generated.
  - PROFILE or NOPROFILE (default)
- $p_8$ start,end: range of star numbers to be analyzed. Default = 0,0 i.e. all stars.

**Examples:**

**Actions taken:**
1. The column content of the output table is described in Table 5.2.
2. Error messages are written to file evaluate.log for each output table and are described in Table 5.3. A log-file with same name and extension "log" is created. It contains detailed error messages for objects with problems.

**Related commands:**
SETUP/EVALUATE

<table>
<thead>
<tr>
<th>column</th>
<th>label</th>
<th>unit</th>
<th>type</th>
<th>format</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>:NR</td>
<td>I8</td>
<td>I*4</td>
<td></td>
<td>unique object number</td>
</tr>
<tr>
<td>2</td>
<td>:LABEL</td>
<td>A8</td>
<td>C*8</td>
<td></td>
<td>name of object, e.g. standard star</td>
</tr>
<tr>
<td>3</td>
<td>:ID</td>
<td>A8</td>
<td>C*8</td>
<td></td>
<td>identifier for object class (1 character only!)</td>
</tr>
<tr>
<td>4</td>
<td>:MAG,0</td>
<td>arcsec</td>
<td>F10.3</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>:DX_LINE</td>
<td>steps</td>
<td>I8</td>
<td>I*4</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>:DY_LINE</td>
<td>steps</td>
<td>I8</td>
<td>I*4</td>
<td></td>
</tr>
</tbody>
</table>
## 5.1. CONTEXT: MAIN

<table>
<thead>
<tr>
<th>column</th>
<th>label</th>
<th>unit</th>
<th>type</th>
<th>format</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>X_MARK</td>
<td>arcsec</td>
<td>F10.2</td>
<td>R*4</td>
<td>(approximate) X-position</td>
</tr>
<tr>
<td>8</td>
<td>Y_MARK</td>
<td>arcsec</td>
<td>F10.2</td>
<td>R*4</td>
<td>dito for Y</td>
</tr>
<tr>
<td>9</td>
<td>X</td>
<td>arcsec</td>
<td>F10.2</td>
<td>R*4</td>
<td>X-position found</td>
</tr>
<tr>
<td>10</td>
<td>Y</td>
<td>arcsec</td>
<td>F10.2</td>
<td>R*4</td>
<td>dito for Y</td>
</tr>
<tr>
<td>11</td>
<td>MAG</td>
<td>mag</td>
<td>F10.3</td>
<td>R*4</td>
<td>relative magnitude</td>
</tr>
<tr>
<td>12</td>
<td>FL1</td>
<td>any</td>
<td>F10.3</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>FL2</td>
<td>any</td>
<td>F10.3</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>IX</td>
<td>pixel</td>
<td>I6</td>
<td>I*4</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>IY</td>
<td>pixel</td>
<td>I6</td>
<td>I*4</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Z_0</td>
<td>count</td>
<td>F9.1</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>X_FIT</td>
<td>pixel</td>
<td>F8.2</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Y_FIT</td>
<td>pixel</td>
<td>F8.2</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>SX</td>
<td>pixel</td>
<td>F8.2</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>SY</td>
<td>pixel</td>
<td>F8.2</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>BACK</td>
<td>count</td>
<td>F9.2</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>ALPHA</td>
<td>degree</td>
<td>F8.1</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>ERROR</td>
<td>A8</td>
<td>C*8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>AP_RAD</td>
<td>I8</td>
<td>I*4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>SAT_RAD</td>
<td>I8</td>
<td>I*4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>MODE</td>
<td>I5</td>
<td>I*4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>INTENSITY</td>
<td>F12.0</td>
<td>R*4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>SIG_INT</td>
<td>F10.0</td>
<td>R*4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>SIG_MAG</td>
<td>F10.3</td>
<td>R*4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>AREA</td>
<td>F7.2</td>
<td>R*4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>ELLIP</td>
<td>F8.3</td>
<td>R*4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>H_BACK</td>
<td>count</td>
<td>F8.2</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>L_BACK</td>
<td>count</td>
<td>F8.2</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>BITS</td>
<td>I8</td>
<td>I*4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>CHISQR</td>
<td>F8.2</td>
<td>R*4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>Z0_BACK</td>
<td>F9.2</td>
<td>R*4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>GOOD_PIX</td>
<td>F8.3</td>
<td>R*4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>DEL_DIRT</td>
<td>count</td>
<td>F8.1</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>SX_FLAT</td>
<td>F8.2</td>
<td>R*4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>SY_FLAT</td>
<td>F8.2</td>
<td>R*4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>INT_BACK</td>
<td>F12.0</td>
<td>R*4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>OLD_ALPHA</td>
<td>degree</td>
<td>F9.1</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>FIRST_STAR</td>
<td>I10</td>
<td>I*4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>MAG_1</td>
<td>F9.3</td>
<td>R*4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>INT_1</td>
<td>F12.0</td>
<td>R*4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>DIST</td>
<td>arcsec</td>
<td>F8.2</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>ANGLE</td>
<td>degree</td>
<td>F8.1</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>FL1_FL2</td>
<td>F8.3</td>
<td>R*4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>OK</td>
<td>A8</td>
<td>C*8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>MEAN_MAG</td>
<td>F9.3</td>
<td>R*4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>SUM_INT</td>
<td>count</td>
<td>F12.0</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>SIG_SUM</td>
<td>count</td>
<td>F10.0</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>53</td>
<td>DEL_INT</td>
<td>F8.4</td>
<td>R*4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>54</td>
<td>P_FLAG</td>
<td>A8</td>
<td>C*8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>SIG_DEL</td>
<td>F8.4</td>
<td>R*4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>DEL_X</td>
<td>arcsec</td>
<td>F12.3</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>57</td>
<td>DEL_Y</td>
<td>arcsec</td>
<td>F12.3</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>58</td>
<td>RDUMMY_1</td>
<td>G12.4</td>
<td>R*4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

... continued next page
<table>
<thead>
<tr>
<th>column</th>
<th>label</th>
<th>unit</th>
<th>type</th>
<th>format</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>.RDUMMY_2</td>
<td>G12.4</td>
<td></td>
<td>R*4</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.3: Error messages from EVALUATE

<table>
<thead>
<tr>
<th>aperture flag</th>
<th>1</th>
<th>maximum possible profile radius [all pixel]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>aperture found</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>saturation radius</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>number of pixels in aperture</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>error flag:</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>profile hits border of aperture</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>desired aperture inconsistent with aperture found:</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>≤ ratio aperture found/aperture specified ≤ 1.5</td>
</tr>
<tr>
<td></td>
<td>or</td>
<td>$R &lt; R_{\text{min}}$</td>
</tr>
<tr>
<td>center profile back</td>
<td>6</td>
<td>profile</td>
</tr>
</tbody>
</table>

| sumup flag | 1 | diameter of aperture in pixels |
|           | 2 | IX of center as found in MAXIT    |
|           | 3 | IY                              |
|           | 4 | —                                |
|           | 5 | status: -1 no significant signal |
|           | status: -2 one of second moments < 0: |
|           | $\langle I*X^2 \rangle - \langle I*X \rangle^2 > \Sigma I$ |
| vector    | 1 | $\Sigma/100$ in aperture          |
|           | 2 | $X_0$                            |
|           | 3 | variance $X$                      |
|           | 4 | $Y_0$                            |
|           | 5 | variance $Y$                      |
|           | 6 | $\Sigma^2/100$: $\Sigma^2 = \Sigma$(values > 0) |
|           | 7 | background from histogram         |
|           | 8 | variance for equal signal         |
|           | 9 | significance in $X$: $\sqrt{\#10/\#5} \times \sqrt{\#8/\#3}$ |
|           | 10| significance in $Y$:             |
| fit       |   |                                 |
EVALUATE/MC OUT_root = IN_list table_root Savart Statistic Profile start,end

Purpose:
This is the same as evaluate/ima with the additional feature that — in the default mode ($PS = \text{COSM}$) — it is also run over the cosmic mask tables, which have to have the root name cmsk. The content of both output table is then "boiled" own into a single output table, with the following columns:

<table>
<thead>
<tr>
<th>Col.#</th>
<th>1:NR</th>
<th>Unit:</th>
<th>Format:</th>
<th>I*4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col.#</td>
<td>2:LABEL</td>
<td>Unit:</td>
<td>Format:</td>
<td>A8 C*8</td>
</tr>
<tr>
<td>Col.#</td>
<td>3:ID</td>
<td>Unit:</td>
<td>Format:</td>
<td>A8 C*8</td>
</tr>
<tr>
<td>Col.#</td>
<td>4:MAG_0</td>
<td>Unit:arcsec</td>
<td>Format:</td>
<td>F10.3 R*4</td>
</tr>
<tr>
<td>Col.#</td>
<td>5:DX_LINE</td>
<td>Unit:steps</td>
<td>Format:</td>
<td>I8 I*4</td>
</tr>
<tr>
<td>Col.#</td>
<td>6:DY_LINE</td>
<td>Unit:steps</td>
<td>Format:</td>
<td>I8 I*4</td>
</tr>
<tr>
<td>Col.#</td>
<td>7:X_MARK</td>
<td>Unit:arcsec</td>
<td>Format:</td>
<td>F10.3 R*4</td>
</tr>
<tr>
<td>Col.#</td>
<td>8:Y_MARK</td>
<td>Unit:arcsec</td>
<td>Format:</td>
<td>F10.3 R*4</td>
</tr>
<tr>
<td>Col.#</td>
<td>9:X</td>
<td>Unit:arcsec</td>
<td>Format:</td>
<td>F10.3 R*4</td>
</tr>
<tr>
<td>Col.#</td>
<td>10:Y</td>
<td>Unit:arcsec</td>
<td>Format:</td>
<td>F10.3 R*4</td>
</tr>
<tr>
<td>Col.#</td>
<td>11:MAG</td>
<td>Unit:mag</td>
<td>Format:</td>
<td>F10.3 R*4</td>
</tr>
<tr>
<td>Col.#</td>
<td>12:FI_1</td>
<td>Unit:mag</td>
<td>Format:</td>
<td>F10.3 R*4</td>
</tr>
<tr>
<td>Col.#</td>
<td>13:FI_2</td>
<td>Unit:mag</td>
<td>Format:</td>
<td>F10.3 R*4</td>
</tr>
<tr>
<td>Col.#</td>
<td>14:INTENSITY</td>
<td>Unit:counts</td>
<td>Format:</td>
<td>F12.0 R*4</td>
</tr>
<tr>
<td>Col.#</td>
<td>15:SIG_INT</td>
<td>Unit:counts</td>
<td>Format:</td>
<td>F12.0 R*4</td>
</tr>
<tr>
<td>Col.#</td>
<td>16:Z_0</td>
<td>Unit:counts</td>
<td>Format:</td>
<td>F12.3 R*4</td>
</tr>
<tr>
<td>Col.#</td>
<td>17:SX</td>
<td>Unit:pixel</td>
<td>Format:</td>
<td>F8.2 R*4</td>
</tr>
<tr>
<td>Col.#</td>
<td>18:SY</td>
<td>Unit:pixel</td>
<td>Format:</td>
<td>F8.2 R*4</td>
</tr>
<tr>
<td>Col.#</td>
<td>19:ALPHA</td>
<td>Unit:degree</td>
<td>Format:</td>
<td>F8.2 R*4</td>
</tr>
<tr>
<td>Col.#</td>
<td>20:AREA</td>
<td>Unit:pixel</td>
<td>Format:</td>
<td>F8.2 R*4</td>
</tr>
<tr>
<td>Col.#</td>
<td>21:BACK</td>
<td>Unit:counts</td>
<td>Format:</td>
<td>F12.3 R*4</td>
</tr>
<tr>
<td>Col.#</td>
<td>22:AP_RAD</td>
<td>Unit:pixel</td>
<td>Format:</td>
<td>I8 R*4</td>
</tr>
<tr>
<td>Col.#</td>
<td>23:SAT_RAD</td>
<td>Unit:pixel</td>
<td>Format:</td>
<td>I8 R*4</td>
</tr>
<tr>
<td>Col.#</td>
<td>24:ERROR</td>
<td>Unit:</td>
<td>Format:</td>
<td>A8 C*8</td>
</tr>
<tr>
<td>Col.#</td>
<td>25:COSM_INT</td>
<td>Unit:counts</td>
<td>Format:</td>
<td>F10.1 R*4</td>
</tr>
</tbody>
</table>

The original full evaluate table and the corresponding log file are deleted.

As the "boiling" is done only after all evaluate runs have been completed, one has to make sure that enough disk space is available!

This is the evaluate command to be used, if the tables are later on to be combined with the FLUX/UNITE command.

Related commands:

FLUX/UNITE
5.2 Context: flux_cal

- Flux calibration of evaluate tables
- and associated utilities
5.2. CONTEXT: FLUX_CAL

FLUX/REFER name of master table

Purpose:
Defines which master table is to be used.

Parameters:
   $p_1$ name of master table

Examples:
   flux/refer Master_03hA_24Jan06  Sets $Master_03hA_24Jan06$ as master table.

Actions taken:
Writes (global) keyword FLUX_REF, holding the name of the master table to be used throughout this MIDAS session.

Related commands:
FLUX/NORM, FLUX/UNITE, FLUX/INTEG
FLUX/STD  master_table  object_number  designation_of_standard_star

Purpose:
Identifies the objects to be used as standard stars in the master table.

Parameters:
  \( p_1 \) name of master table  Name of the master table to be used.
  \( p_2 \) object number  Number (column :Nr) of the object in the master table to be declared a standard star.
  \( p_3 \) standard star name  Name of the standard star (< 9 characters long). This name is used to find the table holding the star’s spectrum. It has to be identical to the name in the first column of table standards, which is used in flux/integ for photometric calibration.

Examples:

  flux/std Master_03hA_24Jan06 12345 03h_01 Defines object with :Nr = 12345 in table Master_03hA_24Jan06 as standard star with name 03h_01.

Actions taken:

  1. Write name to column :LABEL in master table
  2. Set the :ID to "M" in master table

Related commands:

FLUX/INTEG
5.2. CONTEXT: FLUX_CAL

FLUX/SELNORM IN_list = norm_ID general_ID

selection_parameters check/nocheck

Purpose:
Selects for each evaluate table the appropriate stars for normalization and saturation correction. Selection is done in the Z,0-area-plane.

Parameters:

- \( p_1 \) out name
  - list of input evaluate tables (see page 195)

- \( p_2 \) =

- \( p_3 \) norm_ID
  - ID (column :ID in evaluate table) to be used for normalization stars. Default: N

- \( p_4 \) general_ID
  - ID for most of the objects. Default: 1

- \( p_5 \) Selection parameters
  - upper limit to \( Z_0 \) = fraction of dynamic range (descriptor CCD_PARA(4))
  - lower limit to \( Z_0 \) = fraction of \( Z_0 \),max - background, background is taken from descriptor FR_STAT(1)
  - significance factor
  - maximum value for area
  - number of segments in \( \log(Z_0) \)
  Default values are 0.9,0.03,15.,20.,4.
  For OMEGA2000 data the lower limit to \( Z_0 \) should be lowered to 0.005.

- \( p_6 \) check flag
  - check or no_check (default) or look.
  If check is selected, a plot of \( \log(Z_0) \) versus :area is shown for each table. Objects selected for normalization are marked in red. If look is given, it also produces the plot, but also asks if the plot is acceptable (default). If not, the name will be added to catalogue NORM_PROBLEMS and the plot may be saved as a PostScript file.

Examples:

- flux/selnorm eval_H:1 P6=check
  Selects the stars to be usable for normalization and sets their ID for the first evaluate table in catalogue eval_H and displays the results on the graphic screen.

Actions taken:

1. Read descriptors CCD_PARA, FR_STAT, and PSF_MEAN for each input table

2. Loop over objects and select those having :ID equal to general object (P4) or to normalization objects (P3). An already existing selection is taken into account!

3. Calculate clipped average and sigma of area for those objects satisfying selection criterion in \( Z_0 \).
   Keyword KS_MODE and KS_CLIP (see page 211 for details) may be used to influence the calculation of average. If character keyword TEST_KEY is present, more output to diagnose problems are provided.

4. Set area selection to \(|(area - \langle area \rangle)| < significance \times \sigma\)

5. Check if object within selection box in \( Z_0 \)-area-plane
6. If in box, set :ID to norm_ID (P3), if not, set :ID to general_ID (P4)

7. To change the input parameters for those tables with problems, rename the catalogue NORM_PROBLEMS before calling FLUX/SELNORM again with this new catalogue in parameter P1!

Please note: If an object already has the ID set to the norm_ID but is outside the selection box, its ID is reset to the general_ID!

![Plot Output](image)

Figure 5.1: Example of the plot output, if P6 is set to check. Plotting the effective area of an object as a function of its intensity separates stars from galaxies. Stars to be used for normalization (red) have to be well defined and with high S/N. The black streak at $15 < \text{area} < 16$ is due to objects not detectable on this image.

**Related commands:**

- FLUX/SETNORM
- FLUX/NORM
5.2. CONTEXT: FLUX.CAL

FLUX/SETNORM  IN_list  action

Purpose:
Sets or resets a normalization flag for each evaluate table in the input list. One table (corresponding to
best image) should be selected per pointing and filter to be used as the normalization reference image.
These images have to be entered into the unite-table (see FLUX/UNITE page 107) as the first tables.
Usually the best images are selected as normalization frames.
Please note that for mosaics one normalization image is required for each pointing!

Parameters:

\[ p_1 \text{ out.name} \]  list of input evaluate tables (see page 195)
\[ p_2 \text{ action} \]  Define the operation to be performed: Default is set, i.e. for the input tables the
flag is set. To clear the flag use reset or clear.

Examples:

\[
\text{flux/setnorm eval.H:1,23,47}
\]
Selects the tables in entries 1, 23 and 47 as normalization reference.

\[
\text{flux/setnorm eval.H:23 clear}
\]
Clears the normalization flag for table entry 23 set in the previous example.

Actions taken:

1. The flag is integer descriptor norm ima. Its existence (and currently with a value of 1) signals
that the table is to be used for normalization. In the default form this descriptor is written for
all tables from the input list. Tables holding this descriptors are entered into the unite-table as
the first images and all count rates in subsequent images of each pointing are normalized to the
corresponding normalization image.

2. If the second parameter is not equal to set, this descriptor is deleted from all tables in the input
list.

Related commands:

FLUX/UNITE  FLUX/NORM
FLUX/VERIFY \hspace{1em} \texttt{IN\_list} \hspace{1em} \texttt{mask\_check}

**Purpose:**
Check input frames for descriptors needed in flux calibration process. Optionally also the cosmic masks are checked.

**Parameters:**
- $p_1$ \texttt{IN\_list}: list of input frames (see page \textbf{106})
- $p_2$ mask check: If this parameter is set to \texttt{cmsk}, which is the default, the cosmic mask files with root \texttt{cmsk} are searched and checked, too.

**Examples:**
- \texttt{flux/verify O2k\_H\_ima}: Checks all frames in catalogue \texttt{O2k\_H\_ima} for the existence of descriptors. Cosmic mask files with root \texttt{cmsk} are also checked.

**Actions taken:**
1. Descriptors checked are \texttt{CCD\_PARA}, \texttt{PSF\_MEAN}, \texttt{FR\_STAT}, \texttt{LAMBDA}, \texttt{FILTER}, \texttt{EPSILIST}, \texttt{JUL\_DATE} and \texttt{AIRMASS}.

2. If cosmic masks are to be checked and a mask frame is missing, it is created as a dummy from the primary image with a copy of the descriptors.

**Related commands:**
- FLUX/UNITE
- FLUX/NORM
- FLUX/INTEG
5.2. CONTEXT: FLUX.CAL

FLUX/UNITE OUT_name = IN_list norm_IDs wavelength_id cosmic_version max_tab

Purpose:
Create a table holding a subset of the columns from each image relevant for photometry. The first tables to be entered are the normalization tables, usually corresponding to the best image for each tile and filter. One unite table is created for each filter. Filters must not be mixed within a given unite table!

The selection flag in the input tables and in the master table are recognized.

Parameters:

\[ p1 \] out_name full name of output table

\[ p2 = \]

\[ p3 IN_list: \] list of input evaluate tables (see page 195). These tables must not contain different filters for one unite table.

Note:
If a table is specified as input, which was already entered or is only another version of a table already entered (i.e. the same in characters 5 to 8), the corresponding block of columns will be overwritten by the new call (i.e. this is the way to update the output table, if one or several input tables have been recalculated with other parameters).

\[ p4 ID(s) for normalization stars \]
Up to eight different characters can be specified, which designate stars to be used for normalization in FLUX/NORM. Default: N

\[ p5 cosmic root \]
Root of tables from which the cosmic ray flag is taken (if taken from a separate table). Default = NOC, in which case the cosmic information is imported from the boiled evaluate table created by EVAL/MC. If no cosmic information is available, e.g. for short exposures, the default is to be used and the cosmic signal is set to zero.

\[ p6 max. tables \]
Maximum number of tables in total to be entered in the output table. For speed reasons this parameter has to give the final number of images to be entered into this table upon the first call. Addition of further columns is possible in principle but extremely slow. Default: 200

\[ p7 short limit \]
Limit for short exposures [default: 0.]
Shorter exposures are excluded from the calculation of Rcenter.

The new columns entered for each table follow the following naming scheme: If the descriptor filter of a given input table holds cccc and already data for two other tables of the same wavelength are stored in the output table, the following 8 columns will be appended to the output table (iii is the sequence number of this file with the given filter):

\_Cccciii counts/sec as measured by EVALUATE \( (R^4) \)
sc_cccciii error for above \( (R^4) \)
_Fccciii will hold F_lambda [photons/m^2/sec/um] \( (R^4) \)
sF_cccciii error of above \( (R^4) \)
bk_cccciii background under image \( (R^4) \)
Cr_cccciii cosmic ray hit flag \( (R^4) \)
er_cccci error flag \( (I^4) \)
scrcccccciii scratch (holding Rcenter) \( (R^4) \)
If the filter name is less than four digits long, the rest is automatically filled with underscores (_).

The error flags are currently

- $10000000 = \text{use object for normalization}$
- $01000000 = \text{saturated image}$
- $00100000 = \text{corrected saturated image}$.

**Examples:**

```plaintext
flux/unite flux_H_03hA = eval_H: P4=MNOPQRST
```

Create flux table `flux_H_03hA` from evaluate tables in catalogue `eval_H`. Flag stars with IDs M, N, O, P, Q, R, S, T for normalization.

**Actions taken:**

1. The program is called twice. In the first round only the normalization blocks are filled. Then in the second loop, the other tables are copied.

2. Copy general information from columns in master table, defined via `FLUX/REFER`, to the first columns in the output table. These first 29 columns are

   1. No          : unique object number
   2. ID          : identification, defining object classes (i.e. normalization stars)
   3. CLASS       : classification taken from `CLASS/EVAL`
   4. X           : position on master image [arcsec]
   5. Y
   6. CLASS_VAL   : classification parameter [0 (galaxy) ... 1000 (star)]
   7. MORPH1
   8. MORPH2
   9. MORPH3
   10. MAG0       : relative magnitude from SourceExtractor
   11. BEST_FRM   : sum frame on which this object has best S/N
   12. RA         : Right Ascension of object (J2000)
   13. DEC        : Declination
   14. u          : SDSS u (AB mag)
   15. sig_u      : error of u
   16. g          : SDSS g (AB mag)
   17. sig_g      : error of g
   18. r          : SDSS r (AB mag)
   19. sig_r      : error of r
   20. i          : SDSS i (AB mag)
   21. sig_i      : error of i
   22. z          : SDSS z (AB mag)
   23. sig_z      : error of z
   24. J          : 2MASS J (Vega mag)
   25. sig_J      : error of J
   26. H          : 2MASS H (Vega mag)
   27. sig_H      : error of H
   28. K          : 2MASS K (Vega mag)
   29. sig_K      : error of K
   30. Rcenter    : distance to optical axis
   31. scratch1
   32. scratch2

The columns :u to :sig_k will be filled in by `FLUX/SURVEY`, see page 112.
Following this leading block with general information, the average block for this filter \texttt{ffffff} is inserted, which will hold averages over all images in a given filter with the following twelve columns (to be filled by \texttt{FLUX/INTEG}, see page 115):

- \texttt{F_{ffff}} flux (weighted average over all, no clipping) (R*4) \texttt{ffff} is the 4-digit filter designation
- \texttt{sF_{ffff}} error of \texttt{F_{}} (R*4)
- \texttt{sFcffff} unclipped, corrected error of flux \texttt{F_{}} (R*4)
- \texttt{Fmffff} flux (unweighted, no clipping) (R*4)
- \texttt{sFmffff} error of \texttt{Fm} (R*4)
- \texttt{Fkffff} flux (clipped, weighted) (R*4)
- \texttt{sFkffff} corrected error of \texttt{Fk} (R*4)
- \texttt{chi2ffff} chi\textsuperscript{2} (all data) (R*4)
- \texttt{vldffff} fraction of valid blocks (R*4)
- \texttt{cosiffff} sum of cosmic ray hits (R*4)
- \texttt{cosNffff} number of images hit by a cosmic (I*4)
- \texttt{trnsffff} chi\textsuperscript{2} (after clipping), scratch and transfer column (R*4)

3. Copy columns from input table as described above for P4.

4. Calculate RA and DEC from the reference position and the X/Y coordinates.

5. The keywords \texttt{KS_CLIP/R/1/4 kappa,loop,sig_min,min_num} and \texttt{KS_MODE/C/1/80 "... "} can be used to influence the kappa-sigma clipping in the determination of normalization constants and — to some extend — the saturation correction. If \texttt{KS_CLIP} and \texttt{KS_MODE} are not present, the values 2.5, 5, −0.1, 2 and \texttt{MIN/NO_WEIGHT} are currently used.

6. Descriptors read from each input table:

- \texttt{IDENT} C*72 if present
- \texttt{FILTER} C*32 4 characters used only
- \texttt{EXP\_TIME} R*4
- \texttt{JUL\_DATE} D*8
- \texttt{AIRMASS} R*4
- \texttt{PSF\_MEAN} 4 x R*4
- \texttt{EPSILIST} C*80 list of efficiencies (i.e. table names!), e.g. mirror,mirror,cafos22,BR2,tek7
- \texttt{LAMBDA} 4 x R*4 central wavelength, FWHM, start\_lambda, end\_lambda [nm] for integration

For long exposures only:

- \texttt{N\_DETECT} I*4 number of detectors per image
- \texttt{XYMOVE\_IMA} 6 x R*4 if \texttt{N\_DETECT}= 1
- \texttt{OA\_PIX} 2 x R*8
- \texttt{BORDERS} 4 x R*4
- \texttt{XY\_MOVEii} 6n x R*4 for \texttt{N\_DETECT}= n
- \texttt{OA\_Pixii} 2n x R*8
- \texttt{BORDERii} 4n x R*4
- \texttt{CHIP\_EDGESii} 4n x I*4

7. General descriptors created:

- \texttt{REF\_TAB} C*16 name of master table used
- \texttt{N\_ENTRY} number of files entered (I*4)
- \texttt{NO\_NORM} number of normalization blocks entered (I*4)
- \texttt{MAG\_OFF} mosaic correction offset set to 0. \texttt{no\_norm(R*4)}
- \texttt{MAG\_OFF\_S} error of mosaic correction offset set to 0. \texttt{no\_norm(R*4)}
8. Descriptors created for each block:

- **FILTER**: names of the filters
- **LAMBDA_C**: central wavelength [nm] (filled by FLUX/INTEG) (R*4)
- **FWHM**: FWHM [nm] (R*4)
- **LAM_LOW**: rising 1/2 of peak overall efficiency [nm] (R*4)
- **LAM_HIGH**: falling 1/2 of peak overall efficiency [nm] (R*4)
- **STORED_TAB**: tables stored (C*n)
- **JD_DIFF**: Julian date - 2450000 (D*8)
- **AIRMASS**: (R*4)
- **EXP_TIME**: [sec] (R*4)
- **PSF_X**: [pixels] (R*4)
- **PSF_Y**: [pixels] (R*4)
- **PSF_A**: position angle of major axis of PSF [degrees] (R*4)
- **EPSILIST**: list of epsilons used for this block n(C*80)

(Note: can be different within a given set of colours due to different detector etc.)!!

- **LAMB_MIN**: lower integration limit in FLUX/INTEG (R*4)
- **LAMB_MAX**: upper integration limit in FLUX/INTEG (R*4)
- **CALIB_F**: average conversion factor:
  \[ \text{flux} = \text{count rate} \times \text{NORM_FAC} / \text{NORM_MOSAIC} \times \text{CALIB_F1} \]
- **CALIB_S**: error for CALIB_F1 (R*)
- **INT_EPS**: \( \int \epsilon d\lambda \) [cm] (R*4)
- **INT_EPSL**: \( \int \epsilon \times \lambda d\lambda \) [cm] (R*4)
- **LAMIII**: lambda vector of overall efficiency in block iii
- **EPSIIV**: corresponding overall efficiency (from FLUX/INTEG)

9. For each block the distance to the optical axis (Rcenter) is calculated, if the exposure time is above the short exposure limit (p7). The value is stored in the scratch column for each block for later use.

**Related commands:**

FLUX/NORM, FLUX/INTEG
5.2. CONTEXT: FLUX_CAL

FLUX/NORM UNITE-table process min_num tolerance ratio_limit

Purpose:
Calculates normalization factor to the first image entered into the unite table for this tile and corrects
saturated images. All normalization tables are assumed to have been entered first. The normalization
factor is taken as the clipped unweighted average over the count-rate-ratios of the normalization stars.
Its error is the clipped rms.

Parameters:

\( p_1 \) table
Name of the unite table

\( p_2 \) process
all [default] does everything
norm only calculates the normalization constants
corr only does the saturation correction

\( p_3 \) min_norm
Minimum number of normalization objects for each block. [default: 10] If less
objects are detected, a saturation correction is skipped. The name of the table is
entered into catalogue bad_frames.

\( p_4 \) tolerance
Distance tolerance in arcsec to check for coinciding tile [default: 150]

\( p_5 \) ratio_limit
Maximum ratio of count rate to normalization frame [default: 10]. This eliminates
frames with very bad transmission.

Examples:
FLUX/NORM flux_03hA Does normalization and saturation correction.

Actions taken:
1. Normalization constants are stored in descriptor norm_fac and norm_err. The normalization factor
gives the ratio of the count rate level in the normalization block to the current level, i.e. a low value
indicates a good image.
2. Number of stars used for normalization is stored in descriptor n_used.
3. If integer keyword TEST_KEY exists and holds the block number, details about the normalization
stars and average results are displayed.

Related commands:
evaluate/mc flux/unite flux/mosaic flux/integ
FLUX/SURVEY  unite_table = survey_table  tolerance  sort-flag  

**Purpose:**
Read positional and magnitude information from a sky survey table and fill information into the leading blocks of the unite table.

The standard survey tables are found under 

$PM/...tbl/SDSS/ and $PM/..tbl/2MASS/ with file names like

SDSS_03h_C.tbl, SDSS_10h.tbl or 2MASS_16h_A.tbl etc.. 

**Parameters:**

- $p_1$  **table name**  Unite table, for which the leading blocks are to be fill with the columns listed below.

- $p_2$  

- $p_3$  **survey_table**  Name of the table with the sky survey data. The table must have a descriptor SURVEY (C•8) holding the name of the survey. Currently implemented are data from SDSS, 2MASS. Another option foreseen but not yet implemented is OTHER. The columns are listed below.

- $p_4$  **tolerance**  The objects are related via their celestial coordinates :RA and :DEC as provided from the master table. Objects are treated as identical, if their position differs by less than this tolerance [default = 1.0′′].

- $p_5$  **sort flag**  If set to no_sort, the input UNITE table is assumed to be already sorted by increasing :DEC. [default = sort].

**Examples:**

FLUX/SURVEY  flux_03hA_H = 2MASS_03h  

Uses the data from 2MASS as contained in table 2MASS_03h and fills their magnitudes into the leading blocks of flux_03hA_H.

**Actions taken:**

1. Sorts both tables in declination
2. Reads in the survey data
3. Loop through the unite-table and identifies objects
4. Sorts unite-table by :Nr

**Related commands:**

FLUX/MOSAIC  FLUX/INTEG
The survey tables are stored under path $PM/../tbl/SDSS and $PM/../tbl/2MASS and have the following columns:

### SDSS

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
<th>Units/Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. #</td>
<td>RA</td>
<td>degree, F10.6 R*8</td>
</tr>
<tr>
<td>Col. #</td>
<td>DEC</td>
<td>degree, F10.6 R*8</td>
</tr>
<tr>
<td>Col. #</td>
<td>u</td>
<td>ABmag, F6.4 R*4</td>
</tr>
<tr>
<td>Col. #</td>
<td>g</td>
<td>ABmag, F6.4 R*4</td>
</tr>
<tr>
<td>Col. #</td>
<td>r</td>
<td>ABmag, F6.4 R*4</td>
</tr>
<tr>
<td>Col. #</td>
<td>i</td>
<td>ABmag, F6.4 R*4</td>
</tr>
<tr>
<td>Col. #</td>
<td>z</td>
<td>ABmag, F6.4 R*4</td>
</tr>
<tr>
<td>Col. #</td>
<td>sig_u</td>
<td>ABmag, F6.4 R*4</td>
</tr>
<tr>
<td>Col. #</td>
<td>sig_g</td>
<td>ABmag, F6.4 R*4</td>
</tr>
<tr>
<td>Col. #</td>
<td>sig_r</td>
<td>ABmag, F6.4 R*4</td>
</tr>
<tr>
<td>Col. #</td>
<td>sig_i</td>
<td>ABmag, F6.4 R*4</td>
</tr>
<tr>
<td>Col. #</td>
<td>sig_z</td>
<td>ABmag, F6.4 R*4</td>
</tr>
<tr>
<td>Col. #</td>
<td>SDSS_name</td>
<td>, Format:A24 C*24</td>
</tr>
<tr>
<td>Col. #</td>
<td>type</td>
<td>, Format:I5 I*4</td>
</tr>
<tr>
<td>Col. #</td>
<td>flag</td>
<td>, Format:A24 C*24</td>
</tr>
</tbody>
</table>

**Selection:** ALL

### 2MASS

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
<th>Units/Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Col. #</td>
<td>RA</td>
<td>degree, F10.6 R*8</td>
</tr>
<tr>
<td>Col. #</td>
<td>DEC</td>
<td>degree, F10.6 R*8</td>
</tr>
<tr>
<td>Col. #</td>
<td>err_maj</td>
<td>arcsec, F5.2 R*4</td>
</tr>
<tr>
<td>Col. #</td>
<td>err_min</td>
<td>arcsec, F5.2 R*4</td>
</tr>
<tr>
<td>Col. #</td>
<td>err_ang</td>
<td>degree, I3 I*4</td>
</tr>
<tr>
<td>Col. #</td>
<td>name_2MASS</td>
<td>, Format:A17 C*17</td>
</tr>
<tr>
<td>Col. #</td>
<td>J</td>
<td>Vega_mag, F6.3 R*4</td>
</tr>
<tr>
<td>Col. #</td>
<td>J_csig</td>
<td>Vega_mag, F5.3 R*4</td>
</tr>
<tr>
<td>Col. #</td>
<td>J_sigcom</td>
<td>Vega_mag, F5.3 R*4</td>
</tr>
<tr>
<td>Col. #</td>
<td>J_SNR</td>
<td>, Format:F5.1 R*4</td>
</tr>
<tr>
<td>Col. #</td>
<td>H</td>
<td>Vega_mag, F6.3 R*4</td>
</tr>
<tr>
<td>Col. #</td>
<td>H_csig</td>
<td>Vega_mag, F5.3 R*4</td>
</tr>
<tr>
<td>Col. #</td>
<td>H_sigcom</td>
<td>Vega_mag, F5.3 R*4</td>
</tr>
<tr>
<td>Col. #</td>
<td>H_SNR</td>
<td>, Format:F5.1 R*4</td>
</tr>
<tr>
<td>Col. #</td>
<td>K</td>
<td>Vega_mag, F6.3 R*4</td>
</tr>
<tr>
<td>Col. #</td>
<td>K_csig</td>
<td>Vega_mag, F5.3 R*4</td>
</tr>
<tr>
<td>Col. #</td>
<td>K_sigcom</td>
<td>Vega_mag, F5.3 R*4</td>
</tr>
<tr>
<td>Col. #</td>
<td>K_SNR</td>
<td>, Format:F5.1 R*4</td>
</tr>
<tr>
<td>Col. #</td>
<td>ph_qual</td>
<td>, Format:A7 C*7</td>
</tr>
<tr>
<td>Col. #</td>
<td>rd_flg</td>
<td>, Format:A6 C*6</td>
</tr>
<tr>
<td>Col. #</td>
<td>bl_flg</td>
<td>, Format:A6 C*6</td>
</tr>
<tr>
<td>Col. #</td>
<td>cc_flg</td>
<td>, Format:A6 C*6</td>
</tr>
<tr>
<td>Col. #</td>
<td>ndet</td>
<td>, Format:A6 C*6</td>
</tr>
<tr>
<td>Col. #</td>
<td>gal_contam</td>
<td>, Format:I10 I*4</td>
</tr>
<tr>
<td>Col. #</td>
<td>mp_flg</td>
<td>, Format:I6 I*4</td>
</tr>
</tbody>
</table>

**Selection:** ALL
FLUX/MOSAIC unite_table = survey_band 2. colour coeffs

Purpose:
Relates the instrumental magnitudes with the survey magnitude. From the average magnitude difference
the offset for each tile is calculated and stored as a descriptor. By definition the offset for tile #1 is zero.
This step can be skipped, if only a single pointing is analysed. The descriptors to be created here have
already been written by FLUX/UNITE with the default values.

Parameters:
\[ p_1 \text{ table name} \quad \text{Name of the unite table.} \]
\[ p_2 = \]
\[ p_3 \text{ survey band} \quad \text{The magnitude band is used to locate the appropriate column with the survey} \]
\[ \quad \text{magnitudes in the leading blocks. Currently only the SDSS and 2MASS bands J,} \]
\[ \quad \text{H, K and u, g, r, i, z are supported.} \]
\[ p_4 \text{ 2. survey band} \quad \text{A second survey band could be used to correct colour terms. Default: none} \]
\[ p_5 \text{ coefficients} \quad \text{In case a second survey band is used these are the colour coefficients in the sense} \]
\[ \quad \text{[t.b.d.]. Default: 0.,0.} \]

Examples:
FLUX/SURVEY flux_03hA_H = H
Adjust the tiles in table flux_03hA_H by using the 2MASS magnitudes in H.

Actions taken:
1. Only selected table entries are taken into account. This enables to use of a magnitude limit or
colour cut in calculating the magnitude offset for each tile.

2. Descriptors written MAG_ZERO, MAGD_i, MAGE_i, with i = tile number.
   MAGD_i, MAGE_i are the magnitude offsets for the individual stars and its error.

3. Average tile offsets are written into descriptor vectors
   \[ \text{mag_off(j) = magnitude offset for tile j (clipped), a positive offset indicates a less deep exposure} \]
   \[ \quad \text{in comparison to the first block} \]
   \[ \text{mag_off.s = error of clipped magnitude offset,} \]
   \[ \text{mag_off.r = scatter in magnitude offset,} \]
   \[ \text{mag_off.a = average offset unclipped,} \]
   \[ \text{mag_off.n = number of objects used in clipped average,} \]
   \[ \text{mag_off.i = total number of objects available.} \]

The normalisation factor NORM_MOSAIC for each block is calculated as \[ \text{NORM_MOSAIC = } 10^{(-0.4 \times \text{MAG.OFF})}. \]

Related commands:
FLUX/SURVEY FLUX/INTEG
FLUX/INTEG unite_table = cross-reference_table standard_ID flag

Purpose:
Use the information from FLUX/NORM and FLUX/MOSAIC to adjust all count rates so that the whole mosaic can be treated as a single "image" with respect to the normalisation blocks. Then integrate over the standard stars, identified via the cross-reference table and calculate the optimum calibration factor to convert observed photons/sec into a photon flux. See page 109 for a description of the table columns.

If descriptor CAL_FORCE/R/1/2 is present the integration over the standard stars is skipped and these values specify the content of descriptors CALIBm_all and CALIBs_all. This way the calibration of different tiles can be forced to be identical (e.g. for the COSMOS field).

Parameters:

- \( p_1 \) table name Name of unite table to be calibrated.
- \( p_2 = \)
- \( p_4 \) ID ID for standard stars [default: M]
- \( p_5 \) flag Flag for early termination [default: full]. If set to descr_only, only the integration over the response function will be performed. No integration over standard star spectra is done and no conversion to flux is performed. This is for data which are already calibrated, e.g. COSMOS data.

Examples:

\[
\text{flux/integ flux_03hA_B} \quad \text{Do the final flux calibration for table flux_03hA_B}
\]

Actions taken:

1. Get standard star entries, normalization factors and mosaic offsets
2. Checks the normalization blocks. Abort if not all found.
3. Integrate over effective DQE for all blocks (descriptors LAMBDA_C, FWHM, LAM_LOW, LAM_HIGH, LAMbiii, EPSIIiii)
4. Integrate over standard star spectra and calculate individual calibration factors (all with respect to corresponding normalization block!)
5. Average calibration factors for each star over all exposures (descriptors CAL_ffff_AV(1\ldots n_std), CAL_ffff_AV_S(1\ldots n_std))
6. Average calibration factors over all standard stars to get final calibration (descriptors CALIBs_all and CALIBs_all).
7. Apply calibration to each block using normalization factors and mosaic offsets. Resulting calibration factors are CALIB_F and its error CALIB_S.
8. Average fluxes over all exposures for final result.
9. Turn off updates of descriptor HISTORY for the output table.

Related commands:

flux/unite flux/norm flux/mosaic
CHAPTER 5. DESCRIPTION OF COMMANDS

**FLUX/ANALYSE**  
**OUT_table_root = UNITE_table row**

**Purpose:**  
Create an output table for one object facilitating analysis.

**Parameters:**

- $p_1$ **root for out_table**: First 4 characters for output table, replacing the first 4 characters of the input table name.
- $p_2 = $  
- $p_4$ **UNITE table**: Name of input unite table to be analysed.
- $p_5$ **row number**: Data for this row will be entered into the output table. Column labels contain the row number so different output tables can later be merged.

**Examples:**

```plaintext
flux/analyse test = unite_03hA 12345
```

create table testunite_03hA holding the count rates and fluxes for object in row # 12345.

**Actions taken:**

1.

**Related commands:**

- flux/unite
- flux/integ
FLUX/COMBINE  OUT_name = IN_list  max_filters  IDENT  flag

**Purpose:**
Finally combines the calibrated unite-tables into one table to be used as input for the multi-colour classification.

**Parameters:**

- \( p_1 \) \texttt{out_name}  
  Name of final flux table

- \( p_2 \)

- \( p_3 \) \texttt{IN_list}:  
  List of input unite tables (see page 195)

- \( p_4 \) \texttt{no. tables}  
  Total number of tables to be inserted. Default: 30

- \( p_5 \) \texttt{ident}  
  Identifier for output table

- \( p_6 \) \texttt{flag}  
  Flag to enable/disable updating of descriptor \texttt{HISTORY} in future commands. Default: disable

**Structure of the flux table**

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nr</td>
<td>I*4</td>
<td>I8 unique object number</td>
</tr>
<tr>
<td>ID</td>
<td>C*10</td>
<td>A10 object ID</td>
</tr>
<tr>
<td>CLASS</td>
<td>C*8</td>
<td>A12 class (s/g/u)</td>
</tr>
<tr>
<td>X</td>
<td>R*4</td>
<td>F10.2 arcsec gnomonic position X</td>
</tr>
<tr>
<td>Y</td>
<td>R*4</td>
<td>F10.2 arcsec</td>
</tr>
<tr>
<td>CLASS_VAL</td>
<td>I*4</td>
<td>I9 stellarity index 0..1000</td>
</tr>
<tr>
<td>MORPH1</td>
<td>R*4</td>
<td>F7.2 major axis</td>
</tr>
<tr>
<td>MORPH2</td>
<td>R*4</td>
<td>F7.2 minor axis</td>
</tr>
<tr>
<td>MORPH3</td>
<td>R*4</td>
<td>F7.1 position angle</td>
</tr>
<tr>
<td>MAG0</td>
<td>R*4</td>
<td>F8.3 mag representative magnitude</td>
</tr>
<tr>
<td>BEST_FRM</td>
<td>C*28</td>
<td>A28 best frame containing object</td>
</tr>
<tr>
<td>RA</td>
<td>R*8</td>
<td>F8.2 degrees (J2000) Right Ascension</td>
</tr>
<tr>
<td>DEC</td>
<td>R*8</td>
<td>F8.2 degrees (J2000) Declination</td>
</tr>
<tr>
<td>u</td>
<td>R*4</td>
<td>F6.3 ABmag</td>
</tr>
<tr>
<td>sig_u</td>
<td>R*4</td>
<td>F6.3 ABmag</td>
</tr>
<tr>
<td>g</td>
<td>R*4</td>
<td>F6.3 ABmag</td>
</tr>
<tr>
<td>sig_g</td>
<td>R*4</td>
<td>F6.3 ABmag</td>
</tr>
<tr>
<td>r</td>
<td>R*4</td>
<td>F6.3 ABmag</td>
</tr>
<tr>
<td>sig_r</td>
<td>R*4</td>
<td>F6.3 ABmag</td>
</tr>
<tr>
<td>i</td>
<td>R*4</td>
<td>F6.3 ABmag</td>
</tr>
<tr>
<td>sig_i</td>
<td>R*4</td>
<td>F6.3 ABmag</td>
</tr>
<tr>
<td>z</td>
<td>R*4</td>
<td>F6.3 ABmag</td>
</tr>
<tr>
<td>sig_z</td>
<td>R*4</td>
<td>F6.3 ABmag</td>
</tr>
<tr>
<td>J</td>
<td>R*4</td>
<td>F6.3 VEGAmag</td>
</tr>
<tr>
<td>sig_J</td>
<td>R*4</td>
<td>F6.3 VEGAmag</td>
</tr>
<tr>
<td>H</td>
<td>R*4</td>
<td>F6.3 VEGAmag</td>
</tr>
<tr>
<td>sig_H</td>
<td>R*4</td>
<td>F6.3 VEGAmag</td>
</tr>
<tr>
<td>K</td>
<td>R*4</td>
<td>F6.3 VEGAmag</td>
</tr>
<tr>
<td>sig_K</td>
<td>R*4</td>
<td>F6.3 VEGAmag</td>
</tr>
<tr>
<td>Rcenter</td>
<td>R*4</td>
<td>F8.2 arcsec distance to optical axis</td>
</tr>
</tbody>
</table>
The following columns are allocated for object classification:

[See also the COMBO-17 web page: http://www.mpia.de/COMBO/cat_legend.html]
The following 16 columns are allocated for each input table:

- \( _F \) is the 4-digit filter designation.
- \( s_F \) is the unclipped, corrected error of flux \( _F \).
- \( s_Fc \) is the unclipped, corrected error of flux \( _F \).
- \( Fm \) is the unweighted, no clipping flux.
- \( sFm \) is the unweighted, no clipping flux.
- \( Fk \) is the clipped, weighted flux.
- \( sFk \) is the corrected error of flux \( _F \).
- \( \chi^2 \) is the chi-squared value of all data.
- \( vld \) is the fraction of valid blocks.
- \( \cosI \) is the number of images hit by a cosmic.
- \( \cosN \) is the number of images hit by a cosmic.
- \( \text{trans} \) is the corrected error of flux \( _F \).
- \( \text{M} \) is the Vega magnitude.
- \( sM \) is the error of Vega magnitude.
- \( \text{sc1} \) is scratch 1.
- \( \text{sc2} \) is scratch 2.

Examples:

```
fux/combine flux_03hA = unite_03hA:1-3 7 "MANOS wide 03hA"
```

Use tables 1 to 3 in catalogue `unite_03hA` and create flux table `flux_03hA`. A final number of 7 tables is envisaged to be ultimately entered into this output table by later calls. The identifier for the output table is set to "MANOS wide 03hA".

Actions taken:

1. Copy leading block from first table
2. Create additional columns to be used by multi-colour classification
3. Copy flux columns from each input table
4. Copy descriptors from each input table: \( \text{LAMBDA}_C, \text{FWHM}, \text{LAM}_\text{LOW}, \text{LAM}_\text{HIGH}, \text{INT}_\text{EPS}, \text{INT}_\text{EPSL}, \text{FILTER}, \text{lamb}_\text{ffff}, \text{epsi}_\text{ffff} \).
5. Write descriptor `FLX_VERS` with date and time
6. Update of descriptor `HISTORY` is turned on/off depending on flag.

Related commands:

```
flux/unite    flux/average
```
FLUX/AVERAGE OUT_table OUT_filter = IN_list filter_A filter_B flag1 flag2

Purpose:
Creates a new block from 2 existing input blocks, e.g. averaging 2 blocks for same filter with different coverage of the field.

Parameters:
- \( p_1 \) flux_table: Name of flux table
- \( p_2 \) new filter: Name of new filter, for which block is to be created
- \( p_3 = \)
- \( p_4 \) filter_A: name of first filter (=block) to be averaged.
- \( p_5 \) filter_B: name of second filter (=block) to be averaged.
- \( p_6 \) rename flag1: flag indicates if new block is to be renamed to first input block. Default: new
- \( p_7 \) flag2: Flag to enable/disable updating of descriptor HISTORY in future commands. Default: disable

Examples:

Actions taken:
1. Update of descriptor HISTORY is turned on/off depending on flag2.

Related commands:
flux/combine
5.2. CONTEXT: FLUX.CAL

FLUX/CLEANUP  OUT_table = IN_table  filters

Purpose:
Copy the input table to the output table and omit blocks with filter names as specified. The cleanup is also done for the descriptors!
The selection flag of the input table is ignored. If you also want to have a subset of objects you have to use sel/tab and copy/tab. The latter also copies the descriptors automatically.

Parameters:
- $p_1$ flux_table out  Name of new, cleaned flux table
- $p_2$
- $p_3$ flux_table in  Name of input flux table.
- $p_4$ filters  Name filters to be excluded from copy. No blanks between names!

Examples:
- flux/clean new = old B0_B1_i0_i1_  Copy table columns and corresponding descriptors from table old to table new together with the corresponding descriptors.

Actions taken:
1. Copy the leading columns
2. Copy those blocks whose filter name is not included in the list of P4.
3. Copy also the corresponding descriptors.

Related commands:
flux/average
FLUX/BLOCK \( \text{UNITE\_table} = \text{IN\_list} \)
FLUX/STORED \( \text{UNITE\_table} = \text{block\_no} \)

**Purpose:**
Finds the associated block number / name of stored table for each table from the input list and displays them on the screen.

**Parameters:**
- \( p_1 \) \text{UNITE\_table} \quad \text{Name of unite table to be searched}
- \( p_2 = \)
- \( p_3 \) \text{list / no.} \quad \text{List of tables to be found (see 195) or single block number.}

**Examples:**
- \text{FLUX/BLOCK flux\_B\_03hA = bad\_frames:1-3} Use the table catalogue \text{bad\_frames} and display the block numbers associated with each table in the table \text{flux\_B\_03hA} on the screen.
- \text{FLUX/STORE flux\_B\_03hA = 43} Display the name of the table associated with block number 43 in table \text{flux\_B\_03hA} on the screen.

**Related commands:**
- flux/analyse
5.2. CONTEXT: FLUX.CAL

FLUX/PRINTCAL UNITE_table

Purpose:
Print all calibration constants from descriptors of a unite table run through FLUX/INTEG into an ASCII-file, one line per block. This facilitates check of the calibration either within MIDAS r with EXCEL. The name of the output file is INPUT_calib.lis.

Parameters:

\[ p_1 \text{ UNITE_table} \] Name of unite table, whose calibration descriptors are to be printed.

Examples:

```
flux/printcal flux_B_03hA   Prints calibration descriptors into file flux_B_03hA_calib.lis
```

Actions taken:

1. Write output to scratch file temp_cal.lis for descriptors
   
   - CAL_ffff_IND_.ii
   - CAL_ffff_IND_.ii
   - CAL_ffff_AV
   - CAL_ffff_AV_S
   - CALIB_F
   - CALIB_S
   - CALIBM_ALL
   - CALIBS_ALL

2. Transform scratch file to final output file

Related commands:

```
flux/integ
```
ANALYSE/FILL root = unite_table short_limit

Purpose:
Analyse the photometric quality of the flatfield correction (influence of straylight, distortion etc.). It uses the distance to the optical axis as filled into column :scrrrrrrrr in FLUX/UNITE for each block. This command creates a new output table in which the information for the flat analysis is collected.

Parameters:
- \( p_1 \) out_root: First four characters (root) for name(s) of output images
- \( p_2 \) =
- \( p_3 \) input: Input table: flux calibrated table (output from FLUX/INTEG)
- \( p_4 \) short limit: Limit on the exposure time [default: 100.]. Only exposures with exposure times longer than this limit are taken into account.

Examples:
```
analyse/fill flat = flux_B0_03hA 50.
```
Create table flat_B0_03hA and fill it with the information from table flux_B0_03hA needed for the analysis of the photometric properties of the flatfield correction. Only exposures with \( \Delta t > 50 \) sec are used.

Actions taken:
- Create output table with the following columns:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nr</td>
<td>I6</td>
<td>object number</td>
</tr>
<tr>
<td>camp</td>
<td>A4 C*4</td>
<td>observing campaign (YYMM)</td>
</tr>
<tr>
<td>JD_diff</td>
<td>JD-2450000</td>
<td>modified Julian date (JD - 2450000.5)</td>
</tr>
<tr>
<td>tile</td>
<td>A6 C*6</td>
<td>HIROCS tile from which measurement is taken (\eg\ 03h12c)</td>
</tr>
<tr>
<td>X</td>
<td>arcsec</td>
<td>gnomonic position in X</td>
</tr>
<tr>
<td>Y</td>
<td>arcsec</td>
<td>gnomonic position in Y</td>
</tr>
<tr>
<td>u</td>
<td>ABmag</td>
<td>SDSS magnitude</td>
</tr>
<tr>
<td>sig_u</td>
<td>ABmag</td>
<td></td>
</tr>
<tr>
<td>g</td>
<td>ABmag</td>
<td></td>
</tr>
<tr>
<td>sig_g</td>
<td>ABmag</td>
<td></td>
</tr>
<tr>
<td>r</td>
<td>ABmag</td>
<td></td>
</tr>
<tr>
<td>sig_r</td>
<td>ABmag</td>
<td></td>
</tr>
<tr>
<td>i</td>
<td>ABmag</td>
<td></td>
</tr>
<tr>
<td>sig_i</td>
<td>ABmag</td>
<td></td>
</tr>
<tr>
<td>z</td>
<td>ABmag</td>
<td></td>
</tr>
<tr>
<td>sig_z</td>
<td>ABmag</td>
<td></td>
</tr>
<tr>
<td>J</td>
<td>VEGAmag</td>
<td>2MASS magnitude</td>
</tr>
<tr>
<td>sig_J</td>
<td>VEGAmag</td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>VEGAmag</td>
<td></td>
</tr>
<tr>
<td>sig_H</td>
<td>VEGAmag</td>
<td></td>
</tr>
<tr>
<td>K</td>
<td>VEGAmag</td>
<td></td>
</tr>
<tr>
<td>sig_K</td>
<td>VEGAmag</td>
<td></td>
</tr>
<tr>
<td>rad_1</td>
<td>arcsec</td>
<td>distance to optical axis in normalization block for this tile</td>
</tr>
<tr>
<td>Flux_1</td>
<td>phot/m²/s/nm</td>
<td>flux</td>
</tr>
<tr>
<td>sFlux_1</td>
<td>phot/m²/s/nm</td>
<td>flux error</td>
</tr>
<tr>
<td>CD_mag1</td>
<td>CDmag</td>
<td>CD magnitude = (-2.5\log_{10}(\text{flux1})+20.01)</td>
</tr>
</tbody>
</table>
5.2. CONTEXT: FLUX_CAL

- All blocks are referred to the normalization block
- One entry is created for each block and object
- The descriptor N_ENTRY gives the number of measurements in the output table (roughly = number of blocks x number of objects).
- Descriptor FILTER gives the filter used for this table.

Related commands:

FLUX/UNITE  analyse/fit
ANALYSE/FIT root = in_table column degree slices

**Purpose:**
Calculate median over specified column in r-slices over the table created with ANALYSE/FILL. Fit a polynomial as a function of distance to optical axis. Selection flag of the input table is recognized.

**Parameters:**
- \( p_1 \) out_root : First four characters (root) for name of output table
- \( p_2 \) =
- \( p_3 \) input : Input table (output from ANALYSE/FILL)
- \( p_4 \) column : Name of input column over which median is calculated [default: dF_dr].
- \( p_5 \) degree : Degree of the polynomial to be fitted [default: 2]
- \( p_6 \) slices : Number of slices to be used in radial direction [default: 20]

**Examples:**
```
ANALYSE/FIT dCD_ = flat_B0_03hA ? 3 15
```
Use table `flat_B0_03hA` and analyse the radial dependence of \( dF_dr \). A polynomial of degree 3 is fitted to 15 radial slices.

**Actions taken:**
1. Create output table
2. Calculate median in radial slices and bin borders.
3. Plot median (error bars come from the rms!)
4. Interactively select radial range to be fitted
5. Fit polynomial (using MIDAS command REGRESSION/POLY)
6. Regression result is saved with name `flatanalysis`.
7. Display result on screen.

**Structure of output table:**
<table>
<thead>
<tr>
<th>Col.#</th>
<th>Column</th>
<th>Unit</th>
<th>Format</th>
<th>R*4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>r_cent</td>
<td>arcsec</td>
<td>F8.2</td>
<td>bin center</td>
</tr>
<tr>
<td>2</td>
<td>r_min</td>
<td>arcsec</td>
<td>F8.2</td>
<td>lower bin boundary</td>
</tr>
<tr>
<td>3</td>
<td>r_max</td>
<td>arcsec</td>
<td>F8.2</td>
<td>upper bin boundary</td>
</tr>
<tr>
<td>4</td>
<td>val_min</td>
<td></td>
<td>E10.3</td>
<td>minimum data value in bin</td>
</tr>
<tr>
<td>5</td>
<td>val_max</td>
<td></td>
<td>E10.3</td>
<td>maximum data value in bin</td>
</tr>
<tr>
<td>6</td>
<td>median</td>
<td></td>
<td>E10.3</td>
<td>median in bin</td>
</tr>
<tr>
<td>7</td>
<td>average</td>
<td></td>
<td>E10.3</td>
<td>average in bin</td>
</tr>
<tr>
<td>8</td>
<td>sigma</td>
<td></td>
<td>E10.3</td>
<td>sigma in bin</td>
</tr>
<tr>
<td>9</td>
<td>n_data</td>
<td></td>
<td>I6</td>
<td>number of data entries in bin</td>
</tr>
<tr>
<td>10</td>
<td>scratch</td>
<td></td>
<td>E10.3</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>error</td>
<td></td>
<td>E12.6</td>
<td>sigma/sqrt(n_data)</td>
</tr>
<tr>
<td>12</td>
<td>fit_r</td>
<td></td>
<td>E12.6</td>
<td>fit result independent var.</td>
</tr>
<tr>
<td>13</td>
<td>fit_median</td>
<td></td>
<td>G14.7</td>
<td>dependent var.</td>
</tr>
</tbody>
</table>
5.2. CONTEXT: FLUX.CAL

Descriptors written by the fitting routine:

FLATANALYSIS:

FLATANALYSISI:
(1) number of data points
(2) number of independent variables
(3) column number of dependant variable
(4) column number of independent variable
(5)
(6) degree of polynomial ndeg

FLATANALYSISD:
(1) ndeg+1 coefficients

FLATANALYSISR:
(1) minimum value in X
(2) maximum value in X
(3) minimum value in Y (optional)
(4) maximum value in Y (optional)
(5) standard error of the estimate

Related commands:
CHAPTER 5. DESCRIPTION OF COMMANDS

ANALYSE/STORE out_root = input column limits

Purpose:
Select the input table for the different observing campaigns. Call ANALYSIS/FIT sequentially for all campaigns to fit the specified column with polynomials. The fit can be performed in segments selected interactively. Please note: The segments have to be overlapping or adjacent — no gaps allowed!

Store the coefficients and the segment boundaries in descriptors for later use with FLUX/CORRECT.

Parameters:
\[ p_1 \text{ out_root}\] First four characters (root) for name of output table
\[ p_2 = \]
\[ p_3 \text{ input}\] Input table (output from ANALYSIS/FILL)
\[ p_4 \text{ column}\] Name of input column over which median is calculated [default: \( \text{dF} \_\text{dr} \)].
\[ p_5 \text{ limits}\] Radial limits for table selection [default: 0,0, i.e. all selected entries].

Examples:
ANALYSE/STORE dCD_ = flat_z0.03hA :dCD
Use column :dCD, which was calculated by the user as the difference of :CD_mag2 and a SDSS magnitude, to perform the fits. Results are stored as descriptors in flat_z0.03hA.

Actions taken:
1. Establish a list of all observing campaigns entered in the input table
2. Select the input table sequentially for each campaign and the radial limits
3. Display a histogram of the specified column and mark with the graphic cursor the relevant range. This range is then de-displayed in a subsequent histogram with better resolution to finally select the range to be included in the fit.
4. Call ANALYSE/FIT if necessary for several radial segments
5. Store fitting results in descriptors of the input table.

Descriptors written (all vectors!):
camp = observing campaigns
n_ranges = number of segments in radial direction for fits
start_r = start of segment
end_r = end of segment
ndeg = degree of fit polynomial
coeffs = polynomial coefficients

Related commands:
analyse/fit flux/correct
FLUX/CORRECT root = unite_table   analysis_table

Purpose:
Correct the count rates in a unite table based on the results from ANALYSIS/STORE.

Parameters:
- \( p_1 \) out_root  
  First four characters (root) for name of output table

- \( p_2 \) =

- \( p_3 \) in_table  
  Input is unite-table

- \( p_4 \) analysis table  
  Name of table holding the results from ANALYSIS/STORE.

- \( p_5 \) filter  
  Name of filter in analysis table (=\( p_4 \)) [default: same]. This is a security measure. Usually the correction is done with the same filter.

Examples:

\[ \text{flux/correct FLUX} = \text{flux}_B0_03hA \text{ FLAT}_B0_03hA \]

Actions taken:
1. Get fit coefficients from descriptors of the ANALYSIS table (see ANALYSIS/STORE).
2. Correct the count rate and its error for each object in each block based on the distance to the optical axis as contained in column :scrcccciii.
3. For radial distances below the first segment or above the last segment the polynomial from the first resp. last segment will be extrapolated.
4. For overlapping regions the changeover will be in the middle between the two boundaries.

If the keyword TEST_KEY/C/1/80 exists, it should hold the campaign, a desired object number (6 digits!) and the radial minimum and maximum (with the decimal point!), \( e.g. \) 0812,000045,200.,850.. Then a test table TEST_TBL with 2 columns is created. The first column holds the radial distance and the second the computed correction factor. The block numbers and pointers for the polynomial coefficients are printed on the screen.

No correction is performed in a test run!

Related commands:

\[ \text{flux/unite} \]
CHAPTER 5. DESCRIPTION OF COMMANDS

FLUX/APCORR output = flux_table mag_column SourceExtractor-table tolerance mode

Purpose:
Align the input flux table in coordinates with the SourceExtractor table. Use SourceExtractors MAG_BEST and MAGERR_BEST (contained in the table created via FIND/SEX on a sum frame) to calculate the aperture correction column :AD_M_BEST. Normalization is done via unsaturated stellar images (selection flag set for these in the input flux table), for which EVALUATE delivers the correct total flux.

Parameters:
\( p_1 \) output Full name of output flux table.
\( p_2 = \)
\( p_3 \) in_table Input is a flux table, selected for unsaturated stellar images
\( p_4 \) mag Name of column holding the Vega magnitude
\( p_5 \) SE-table Name of of the SourceExtractor table
\( p_6 \) tolerance Positional tolerance for the alignment of the flux table with the SE table in arcsec [default: 0.5]
\( p_7 \) mode Mode for filling column :AD_M_BEST. Default is SINGLE, which fills column for matching objects and resets the element for non-matching elements. If set to CUMULATIVE then only matches for still empty elements are filled.

Examples:

```
flux/apcorr FLUX_corr = flux_B0_03hA :M_B__ Find_B__03hA 1.,1.
```

Use positional tolerance of 1" in both axis.

Actions taken:
1. Align images
2. Calculate average magnitude offset between the tables for stars (= objects selected in the input flux table).
3. Calculate offset for each object and subtract the stellar offset value.
4. Fill columns :AD_M_BEST, :MAG_BEST_ERR, :MAG_BEST_FILT. The last item gives the name of the filter used in the SE-table.
5. Filter name is taken from descriptor FILTER of the SE-table.
6. Results of stellar averages are written to descriptors AP_CORR_I_ffff and AP_CORR_R_ffff where ffff is the filter name.

Related commands:
flux/integ
5.3 Context: classy

The CADIS survey showed that classification and redshift estimation in surveys with medium-band filter sets are a powerful and accurate tool for studies on the evolution of galaxies and quasars over most of the time the universe existed. The survey strategy is capable of replacing time-consuming spectroscopy as the main tool for classification and redshift measurements (see paper by Wolf, Meisenheimer & Röser, on the method A&A 2001, 365, 660 and paper on the public CDFS catalogue A&A 2004, 421, 913).

Within CADIS we developed a MIDAS tool for classification and redshift estimation. The following MIDAS commands are available in the context CADIS:

Starting out:
Environment variables
MIDAS context classy
Making & testing libraries:
Several commands... Calculation of a new spectral galaxy library starting with the PEGASE code
@@ MC_LIBRARY Calculation of a new colour library
@@ prepare_MC_COMBO Creation of a simulated object catalogue
MCCLASS/CLASSIFY Classification of an object table
Making & testing flux tables:
MCCLASS/COLORS Calculation of colours on an object table
MCCLASS/ERRORS Calculation of depth by band and flux error plots
MCCLASS/MAINEQ Calibration check of an object table
MCCLASS/CLASSIFY Classification of an object table
MCCLASS/PLOT Plot an individual object SED with template from a catalogue
Restframe quantities Calculate various restframe quantities for galaxies and QSOs (several prgs, link not active yet!!)

Before activating this context via set/cont classy the following environment variables have to be set, if not yet done:

setenv MCC $PM/MCC
setenv MCCTAB $MID_WORK/MCC, e.g.

Please make sure the directory to which $MCCTAB is pointing exists and contains the plot routine mc_plot_f04.prg, tailored to your needs.

Except for MCCTAB all this should have been set up correctly already with .copyphot when MPIAPHOT was installed.
5.3.1 Generating galaxy colour libraries for multi-colour classification

The multi-colour classification performed in the COMBO-17 and MANOS surveys relies on a comparison of measured colour data with a-priori known object libraries of colour templates. A common way of generating such colour libraries involves the preparation of spectral libraries and the subsequent conversion of spectra into colours. Here, we explain how libraries for galaxy colours are prepared. If we work with spectra from synthetic population synthesis codes, this procedure involves three phases:

A.) The preparation of a base of synthetic spectra, e.g. from the PEGASE code
B.) The assembly of a suitably structured library from these base spectra
C.) The calculation of colour templates from the spectral templates.

In the following, we explain the individual steps taken with our current software setup. You need the PEGASE package compiled on your machine, in particular the executables SSPs.exe, scenarios.exe, spectra.exe, and the compiled C program sortieren.exe, as well as the MIDAS scripts crea_PEGlib.prg, inter_550_3250.prg and apply_ex_SMC.prg for section A. For section B, you need to write your own scripts following your specific intentions and needs. Section C relies on the MIDAS context classy.ctx and specifically a script defining the filter set such as farben.all.prg available on the /photo/user/exe/MCC/-area, just like the other relevant files including the PEGASE package.

A.) The preparation of a base of synthetic spectra:

- Run **SSPs.exe** only once to generate relevant input files for PEGASE. This step needs to be done only once and forever if you don’t erase the files which are produced now and subsequently required. See the PEGASE manual for details.

- Run **scenarios.exe** to setup a star formation history (SFH) and further parameters for your first set of spectra. You can define several different SFH setups within one call of this program, until you exit the query loop. The setups will all be stored in the same scenario file. Alternatively, you can rerun scenarios.exe any time to generate further scenario files with further specifications of alternative SFHs. In COMBO-17 we do not use PEGASE-internal dust extinction with chemically consistent curves but we apply extinction laws like external screens following an independent recipe.

- Run **spectra.exe** to produce continuum spectra and emission-line fluxes for all SFHs defined in one scenario file. PEGASE outputs the spectral data for a given grid of time steps running from 0 Myr to 20000 Myr after the onset of the first SF.

- Start a MIDAS session, and run `@@ crea_PEGlibs {filename.dat}` to turn the spectral data from one SFH, i.e. one PEGASE spectra output file, into a MIDAS table with spectral templates. The first column of the resulting table holds the wavelength axis $\lambda$ (in nm) and the subsequent columns hold the flux spectra of all time steps calculated by PEGASE. This step also generates artificial emission lines following their calculated fluxes which are inserted and added on top of the continuum spectra.

- Run `@@ inter_550_3250 {Spec-table}` to add further time steps in the grid via interpolation. Certain areas with too low grid density are thus filled to ensure that the colour library obtained later will not contain any holes that reduce classification performance. You have now a spectral template library on the desired grid of time steps for the desired SFH assuming no dust extinction.

- If you want a library with a certain amount of dust extinction applied, you need to run `@@ apply_ex_SMC {Intable} {Outtable} {exA_B}` and specify the amount of extinction in terms of B-band absorption $A_B$ in magnitudes. For the SMC law defined in this script, you can assume $A_B = 4.2 * E_{(B-V)}$ such that a grid in $E_{(B-V)} = [0.0,0.1,...,0.5]$ is realized with a set of absorption
values $A_B = [0.00, 0.42, ..., 2.10]$. You can easily modify the extinction law by adjusting the parameters describing the three optical-UV components of the extinction curve. Pei (1992) tabulates appropriate parameters for the SMC, LMC and Milky Way, but a 3-parameter family of extinction laws can be defined by varying the parameters as you wish.

B.) The assembly of a suitably structured library from these base spectra:

You now want to define a spectral library with precisely the SFHs, extinctions, composite populations and grid structure desired for the multi-colour classification. If you want to allow for the estimation of library parameters besides redshift, then you need to give your library a regular grid structure in these parameters (see MCCLASS/LIBRARY). You can also generate composite populations from your base spectra by just adding spectra of different age in some weighted form, although a consistent treatment of metallicity would require to define a complex star formation history and rerun PEGASE. However, the first option may be simpler.

C.) The calculation of colour templates from the spectral templates:

You can now turn your spectral template library into a colour template library using MCCLASS/LIBRARY.

Last update July 20, 2005, CW
MCCLASS/LIBRARY or @@ MCC:mc_library_lyfo

Purpose:

Calculation of a colour library on the basis of a spectral template library and a chosen filter set as defined in a MIDAS prg. Suitable colour libraries are required for calibration checks with MCCLASS/MAINSEQ and for the classification with MCCLASS/CLASSIFY. The variety MC_LIBRARY_LYFO takes the Lyman forest into account, but is computationally much slower and is currently only suitable for the QSO library.

Syntax: MCCLASS/LIBRARY [spec-lib] [color-lib] [system]

spec-lib = Name of input table with library spectra
(no default)
color-lib = Name of output table, i.e. colour library
(default: outlibrary)
system = Name of MIDAS prg with the colour system
(default: farben_all)

Examples: MCCLASS/LIBRARY wega
@@ MCC:mc_library wega fbib_wega
@@ MCC:mc_library_lyfo lib_QSO_sdss fbib_QSO farben_ESO_HJR

Remarks:

This command invokes the context synphot in order to use the PHOT/INTEG routine of the MPIAPHOT package. The desired colour system is to be defined in a prg-file. The /photo-area already contains a number of pre-defined systems (CADIS, WFI@2.2, SDSS, HDF-4-Farben, Johnson, Cousins, Bessell, ODT) including their interlinking cross colour terms. If you have further filter curves in hand, you can define additional colour systems.

The columns of the input table must contain the individual object spectra in arbitrary units of \( \text{f}_\lambda \). Any particular normalisation of the flux will not propagate into the colour library where only colour indices will be found. The first column \( \lambda \) must hold the wavelength in units of nm. Flux values of 0.0 are totally ignored, and could lead to a reduced spectral coverage see by PHOT/INTEG. As soon as an object spectrum does not cover the entire defined wavelength range of a given filter curve, no flux will be determined for the corresponding filter and hence no colour indices which include this filter. The colour library will contain a value of 99.00 at these places, which is interpreted by the classification program as undefined.

The output colour library is organised as a regular sequential MIDAS table, but is interpreted by the classification program as a logical 4D array, whose array size is defined by the descriptor NELEMENT. This allows the definition of colour libraries which have up to four free parameters. The first parameter whose size is NELEMENT(1) is reserved for the redshift, since the classification program will only try to search for bimodalities along this parameter axis. The following parameters NELEMENT(2) to (4) can be used arbitrarily for any other parameters structuring your library. The spectral library needs to contain spectra for all grid points on the defined parameter grid. The order of these spectra in the library columns needs to run through parameter (4) first and most quickly, then loop over parameter (3) in an intermediate loop and finally go through (2) in the outermost loop. The redshift can be ignored in the spectral library, since the procedure mc_library.prg loops over the relevant redshift values itself. For this reason, the template spectra should be given in restframe. The starting value and grid step size for the redshift are to be defined in descriptors attached to the spectral library. The redshift grid is equidistant
on a log(1+z) axis. Due to the linearity of the logarithm around the argument 1, the step size on the logarithmic axis is equal to the step size on a linear grid at z=0.

Altogether, the input table of template spectra needs three particular descriptors:

- ZSTART/R/1/1 with the smallest (starting) redshift value ($0.0$ for stars)
- ZSTEP/R/1/1 with the grid step size in redshift
- NELEMENT/I/1/4 with the element numbers (axis sizes) of the 4D array

E.g., the Pickles catalogue of 131 unsorted stars needs the following descriptors:

- ZSTART 0.0
- ZSTEP 0.001 (always choose greater than zero)
- NELEMENT 1,1,1,131

Or if the catalogue by Kinney et al. with ten template SEDs needs to be calculated for $z=0.00,0.002,0.004...1.502 = (1.002^{459})^{-1}$, then say:

- ZSTART 0.0
- ZSTEP 0.0002 (always choose greater than zero)
- NELEMENT 460,10,1,1

The unit of the resulting colour indices is CD magnitudes. More about magnitudes and column names, see the command MCCLASS/COLORS.

*Last update July 19, 2005, CW*
@@ MCC:prepare_MC_COMBO

Purpose:

Preparation of a mock catalogue for Monte-carlo simulations of the classification. Here, a colour library of one object class is used to simulate the photometry of test objects with a desired magnitude. Such simulations allow to spot inherent weaknesses in the classification.

Syntax:   @@ MCC:prepare_MC_COMBO [color-lib] [R-mag] [out-file]

        color-lib  = Name of input table, i.e. colour library
               (default: test_lib)
        R-mag     = R magnitude for test objects
               (default: 21)
        out-file  = Name of output table with test objects
               (default: test_out)

Examples:  @@ MCC:prepare_MC_COMBO fMC_UK96
          @@ MCC:prepare_MC_COMBO fMC_QSO 22.5 testqso_225
          @@ MCC:prepare_MC_COMBO fMC_KICA ? test1

Remarks:

This command creates a table of test objects, which is the same set of objects as in the colour library from which it is made. The objects have the same colour indices plus added noise which is determined from the object brightness and the typical COMBO magnitude limits. All columns required by the classification program are created. Hence, the mock table created here can be processed with MCCLASS/CLASSIFY like any observed flux table. The parameter file can be used to select which colour indices should be actively used and which should be ignored to test the dependence of the classification performance on the availability of certain data.

Last update July 19, 2005, CW
5.3. CONTEXT: CLASSY

MCCLASS / CLASSIFY

Purpose:

Classification of a CADIS-format object table: On the basis of colour related information, the probability of being a star, a white dwarf or blue HB star, a galaxy or a quasar is calculated for every object. Together with the morphology, and potentially the variability, a final interpretation is reached.

Syntax:

MCCLASS/CLASSIFY [flux-table] [me] [par] [list] [min] [limit] [var]

- flux-table = Name of the CADIS-format object table to be classified
  (default: flux_SURVEY.tbl)
- me = Minimum error assumed for colour indices
  (default: 0.05)
- par = Name of a parameter file determining the colors
  (default = feature_combo.par)
- list = Flag for MIDAS-listing of detailed solutions
  (List or default = NoList)
- min = Required minimum probability for classifiability in %
  (default = 75.0)
- limit = Maximum sigma of fit allowed for classification
  (default = 10.0)
- var = Variability prior to be used?
  (default = varyes)

Examples:

MCCLASS/CLASSIFY flux16h_Dez.tbl
MCCLASS/CLASSIFY flux9h_27Apr98.tbl ? fea_4filt.tbl
MCCLASS/CLASSIFY flux9h_27Apr98.tbl 0.03 ? List
MCCLASS/CLASSIFY flux16h_Dez.tbl ? ? ? 90.0

Remarks:

The object table has to be run through MCCLASS/COLOURS first in order to calculate the colour indices and their errors for all objects. Furthermore, a :class -column is required as well as a number of other columns which are to incorporate the final interpretation (see below). The parameter file defining the comparative colours has to be in the directory where the command is called.

The probability values for the classification are based on a template-fit procedure making use of three different kinds of information which are:

- the previous knowledge in the form of colour libraries of stars, white dwarfs, galaxies and quasars
- the measured value for the colours of the objects
- the measuring error for the colours of the objects

Moreover, the classification estimates the redshifts for the extragalactic classes which are determined with two procedures: (1) the minimum error variance method (MEV) and (2) the maximum likelihood method (ML). During the process, the programme first calculates the solution for the star interpretation, estimates subsequently the WD/BHB/sdB case, the galaxy case and then the quasar case. Then, the probabilities for the individual classes are compared and the final interpretation is entered into the result columns.

Explanation of the result columns:
1. General results
• **prob_star** probability in %, that an object is a star

• **prob_wd** probability in %, that an object is a wD/BHB/sdB star

• **prob_gal** probability in %, that an object is a galaxy

• **prob_qso** probability in %, that an object is a quasar The sum of the three probabilities is 100%. It is assumed that there exists no other kind of object.

• **MC_class** Final classification of the object determined by using colour and morphology as well as by respecting the minimum probability required (stars and WD/BHBs combined need to reach $p_{\text{min}}$):
  - **Star** = definite stars (colour of star, stellar shape if $z < 0.2$)
  - **WDwarf** = definite WD/BHB/sdB star (colour of WD/BHB/sdB, shape if $z < 0.2$)
  - **Galaxy** = definite galaxies (colour of galaxy, shape irrelevant)
  - **Galaxy (Star?)** = most likely galaxy at $z < 0.2$ (star colour, but extended shape and $z < 0.2$ - overlap in colour space!
  - **Galaxy (Uncl!)** = colour undecided (statistically almost always a galaxy)
  - **QSO** = definite QSOs (colour of QSO, stellar shape)
  - **QSO (Gal?)** = colour of QSOs, extended shape (low-L AGN or contaminating galaxy)
  - **Strange Object** = very strange spectrum (strong photometric artifacts or uncorrected strong variability or unusual spectrum, e.g. strong ELs)

2. Minimum error variance results for the classified interpretation:

• **MC_z** redshift of the object corresponding to classification. z-value of MEV estimate, if galaxy or quasar, *null*, if star or WD or no MEV estimate possible.

• **MC_sz** error estimate for redshift (sigma of $p(z)$ distribution)

• **MC_z_bin** interval, from which the solution for $p(z)$ is taken: $0$ : There is no solution. $10$ : There is exactly one solution. $21$ : There are two solutions. The one at lower $z$ is given. It is more likely. $22$ : There are two solutions. The one at higher $z$ is given. It is more likely.

• **MC_gSED** spectral type of the object, if galaxy ($0..5$): $0..2$ means dust-free (mostly bulges) $1..2$ means increasing extinction

• **MC_sgSED** error estimate for the SED (sigma of distribution)

• **MC_gSED2** spectral type of the object, if galaxy ($0..59$): from 0 to 59 the mean age of the population increases from 50 Myr to 15 Gyr for a tau=1 Gyr exponential SFH $SED = 0..25$ means starbursts $SED = 25..42$ means typical disk galaxies $SED = 42..59$ means typical bulges

• **MC_sgSED2** error estimate for the SED2 (sigma of distribution)

3. Maximum likelihood results for the classified interpretation:

• **MC_z_ml** redshift of the object corresponding to classification. z-value, if galaxy or quasar, *null*, if star.

• **MC_gSED_ml** spectral type, if galaxy ($0..5$), else *null

• **MC_gSED2_ml** spectral type, if galaxy ($0..59$), else *null

4. Maximum likelihood results for the nearest library object:

• **MC_nextlib** number of the library object. The name of the library is given in the descriptor $MC-library1$, $MC-library2$ and $MC-library3$.

• **MC_next_0** parameter index 0 of the object (defines position of

• **MC_next_1** ... 1 the object within

• **MC_next_2** ... 2 the library array)
5.3. CONTEXT: CLASSY

- **MC\_next\_3** ... 3
- **MC\_dist** distance to the nearest library object (kind of sigma), needs rescaling. Presently, strange objects are best selected by **MC\_dist.gt.6.0**

Further results are used for internal success checks and detailed analyses:

- **sdmg** distance from the star library (units as in **MC\_dist**)
- **wdmg** distance from the WD/BHB library (units as in **MC\_dist**)
- **gdmg** distance from the galaxy library (units as in **MC\_dist**)
- **qdmg** distance from the quasar library (units as in **MC\_dist**)
- **spI** probability integral of the star library
- **vpI** probability integral of the WD/BHB library
- **gpI** probability integral of the galaxy library
- **gpIa** partial integral of the lower z-interval
- **gpIb** partial integral of the upper z-interval
- **qpI** probability integral of the quasar library
- **qpIa** partial integral of the lower z-interval
- **qpIb** partial integral of the upper z-interval
- **stype** MEV estimate of the star type (meaningless since 1999!)
- **sstype** error of this estimate (meaningless since 1999!)
- **sdmg3** parameter 3 of the nearest star (= row number in the star library)
- **wdmg3** parameter 3 of the nearest WD/BHB (= row number in the WD/BHB library)
- **gz** MEV estimate of the galaxy redshift
- **gsz** error of this estimate
- **gza** estimate of the redshift in the lower interval
- **gsza** error of this estimate
- **gzb** estimate of the redshift in the upper interval
- **gszb** error of this estimate
- **gSED** MEV estimate of the galaxy SED
- **gsSED** error of this estimate
- **gSeda** estimate of the SED in the lower interval
- **gsSeda** error of this estimate
- **gSedb** estimate of the SED in the upper interval
- **gsSedb** error of this estimate
- **gSED2** MEV estimate of the galaxy SED2
- **gsSED2** error of this estimate
• \texttt{gSED2a} estimate of the SED2 in the lower interval
• \texttt{gsSED2a} error of this estimate
• \texttt{gSED2b} estimate of the SED2 in the upper interval
• \texttt{gsSED2b} error of this estimate
• \texttt{gdmg0} parameter 0 of the nearest galaxy
• \texttt{gdmg1} ... 1
• \texttt{gdmg2} ... 2
• \texttt{gdmnr} row number of this galaxy in the galaxy library
• \texttt{qz} MEV estimate of the quasar redshift
• \texttt{qsz} error of this estimate
• \texttt{qza} estimate of the redshift in the lower interval
• \texttt{qsza} error of this estimate
• \texttt{qzb} estimate of the redshift in the upper interval
• \texttt{qszb} error of this estimate
• \texttt{qalph} MEV estimate of the quasar spectral index
• \texttt{qsalph} error of this estimate
• \texttt{qepsi} MEV estimate of the quasar emission line strength
• \texttt{qsepsi} error of this estimate
• \texttt{qdmg0} parameter 0 of the nearest quasar
• \texttt{qdmg1} ... 1
• \texttt{qdmg2} ... 2
• \texttt{qdmnr} row number of this quasar in the quasar library

\textbf{ATTENTION:} All integer numbers follow C-convention, first entry = 0

\textbf{Colour libraries}

\textit{Version 23 June 2003}

For stars, WD/BHBs/sdBs, galaxies and quasars, there is one colour library, respectively, containing about 70 colour indices including those for the SDSS, ODT, CADIS and COMBO-17 surveys, as well as Johnson, WFPC2 and other bands. For the classification only those colour indices are used which can be found simultaneously with identical names in three places: the colour library, the object table and the parameter file. In general, the colour libraries are organized as four-dimensional arrays; in the case of a simpler data structure, some dimensions collapse. In analogy to the sequential storage of array variables in programming languages, the library arrays are sequentially organized as MIDAS tables. The index of the parameter 0 has the slowest upward count, the index for parameter 3 changes in every line. Parameter 0 is generally reserved for the redshift being the only one which, in the case of fuzzy distributions, can split into two partial intervals by the classification programme. The three colour libraries currently recommended are found in /photo/user/tbl/COLORLIB/ and are explained here:
5.3. CONTEXT: CLASSY

- fG_UK96.tbl
  The star library, derived from the UK spectra of the Pickles catalogue (1998 PASP 110, 863). The 96 stars of spectral types F,G,K and M having luminosity classes I to V are used. The library has an unsorted structure.

  Weakness: It contains no L-stars and no white dwarf/M-dwarf-binaries.

- fG_WDDwarf.tbl
  The WD/BHB/sdB library, derived from the theoretical spectra and provided by Detlev Koester (more details follow soon) covering DA types with temperatures ranging from 6000 K to 40000 K and surface gravities from log g = 9 to log g = 6 (sub-dwarfs sdB/sdA). The library has an unsorted structure.

- fG_KICA100_M2002.tbl
  The older alternative:
  The galaxy library, derived from the template spectra by Kinney, Calzetti et al. (1996, ApJ 467, 38). The ten templates for the different galaxy-SEDs (E up to S1 = *B1) from 125 nm up to 800 nm have been adopted almost unchanged (assumed artefacts were removed at 290 to 340 nm, at 540 nm and beyond 680 nm). Towards the K-band the $F_{\nu}$ue spectra were extrapolated by a simple power law, with the spectral index -0.333 for the types E to Sb, and with the spectral index -1.0 for star burst galaxies. Within the flux space, the ten templates were extended to 100 spectral types by interpolation. The library is organized as a two-dimensional array. Parameter 0 is the redshift, which assumes 460 values in equidistant steps of 0.002 on a log(1+z)-axis within the interval from z = 0.00 up to 1.502. Parameter 1 is the spectral type, whose translation key into local morphological types is explained above. Thus, the library contains 46000 objects.

  Weakness: The one-dimensional spectral type cannot perfectly reproduce the different ratios of lines which, in the case of galaxies with two or more emission lines in medium-band filters, may lead to a wrong solution (e.g. z = 0.25 vs. 0.66 in CADIS filter set; here, the emission line analysis should offer a more reliable result). Also, Seyfert-2-galaxies have occasionally different combinations of continuum slopes not realized in this library, so their redshifts can be wrong.

- fG_REx1000_SMC6.tbl
  The newest galaxy library:
  A galaxy library calculated with the PEGASE population synthesis package in a 2-D format with 60 ages times 6 extinction levels. The ages range from 50 Myr to 15 Gyr and contain some interpolated steps besides the age steps produced by PEGASE itself. The SFR follows an exponential decline with a tau of 1 Gyr. The model involves standard parameters, a Kroupa IMF, an initial metallicity of 0.01 and is extinction-free. The extinction is applied afterwards as a screen following the SMC law defined by Pei (1992) with six values of E(B-V)=[0,0.0,0.1...0.5]. The SMC law is most appropriate for higher-z galaxies, while at z<0.5 the 220 nm bump of higher metallicity galaxies is not visible in our filter set anyway, so its absence from the SMC law is irrelevant. 177 steps in redshift cover the range from 0.0..1.4 at 0.005 resolution on a log(1+z) scale.

- fG_QSO_sdss05.tbl
The quasar library, derived from the SDSS template spectrum. The template is varied in intensity, added to a power law continuum and multiplied by a redshift-dependent throughput function modelling the Hydrogen absorption bluewards of the Lyman alpha line. The library is organized as a three-dimensional array. Parameter 0 is the redshift which adopts 155 values in equidistant steps of 0.01 on a log(1+z)-scale within the interval z = 0.504 up to 5.96. The template does not cover sufficiently red wavelengths to be used at lower z. Parameter 1 ist the spectral index of the continuum and assumes 20 different values. Parameter 2 is the intensity of the emission line shape and assumes eight values of intensity relative to the mean template (more details follow soon). Thus, the library contains 30600 objects.

The differentiation between galaxy and QSO depends on the ratio of nuclear luminosity to host luminosity. Effectively, it appears, as if all objects of $M_B < -21.5$ are classified as QSO, provided their apparent magnitudes are bright enough, including a few lower-luminosity objects.
Parameter files

The parameter files contain information about the colour indices, the libraries to be used for comparison and the magnitude used for a classification improved by a MAP approach taking class abundance known a-priori into account. This file allows to switch off specific colour indices for the classification or switch off the MAP1-classification stage or use custom-made color libraries.

Example:

! feature_cadis.par  C. Wolf 06.03.2002
! example parameter file for CADIS colors
!
 feature=:bmr
 feature=:rmi
 feature=:imk
 !
 feature=:b531mr
 feature=:b612mr
 ! feature=:rmb700
 feature=:b752mi
 feature=:imb855
 feature=:imb909
 feature=:bmv465
 !
 MAP_mag=_M_I__E
!
 library1=/photo/user/tbl/COLORLIB/fG_UK96.tbl
 library2=/photo/user/tbl/COLORLIB/fG_KICA100_Feb02.tbl
 library3=/photo/user/tbl/COLORLIB/fG_QSO_sdss.tbl
 library4=/photo/user/tbl/COLORLIB/fG_WDwarf.tbl

Lines beginning with ! are ignored as they are regarded as comment lines. The command feature:= announces a comparative feature to the classification programme. Actually it is only used, when it is also available in both the library and the object table. The command library{i}= announces the path and filename of libraries to be used. Library1 must be for stars, library2 for galaxies, library3 for quasars and library4 for WD/BHBs. If the command for any library is omitted, the corresponding recommended library (see above) is used.

List option:

The list option is a remnant from testing times by means of which the classification of individual objects can be traced. The output of the programme is then also recorded in the MIDAS logfile. If this option is used, the programme is processed more slowly.

Last update Oct 30, 2003, CW
MCCLASS / COLORS

Purpose:

Calculation of colour indices and their errors for an object table, using the flux data $F_\nu$ and their errors $s_F$. These colour indices are also used for the calibration tests with the stellar main sequence using MCCLASS/MAINSEQ and for the classification with MCCLASS/CLASSIFY. For COMBO-17 usage, see particulars below.

Syntax:   MCCLASS/COLORS [flux-table]

  flux-table = Name of object table
  (no default)

Example:   MCCLASS/COLORS flux16h_Dec00.tbl

Remarks:

Colour indices are in units of CD magnitudes. The CD magnitude is defined as $CDmag = -2.5 \times \log_{10}(F_\nu) - \text{const}$. The AB magnitude contains $F_\nu$ as an argument for the logarithm and the ST-Magnitude has $F_\lambda$ as an argument of the logarithm. As a result, the CD magnitude is always the mean of AB and ST magnitude. These three magnitudes are physical definitions. Astronomical magnitudes have the ratio of object flux and Vega flux as an argument of the logarithm. All magnitude systems are calibrated such, that Vega has $ABmag = CDmag = STmag = mag = ... = 0.0$ at 548 nm wavelength. The CADIS filter set has the following colour indices, e.g.:

:bmr
:rmi
:imk
:bmb489 :v396mb
:b531mr :bmv465
:b612mr :v522mr
:rmb700 :v535mr
:b752mi :v611mr
:imb855 :v628mr
:imb909 :rmv683
:rmv702

The column names of the colour indices are always constructed along the same scheme. They are denoted $\{\text{bluer filter}\}m\{\text{redder filter}\}$.

COMBO-17 use:

In COMBO-17 every survey field needs its own colour-calculating prg to allow for good variability correction in the survey, which is essential for high-quality photometric redshifts of QSOs. The respective commands for the fields are collected in the directory MCC and are:

- @ MCC:mc_colors_A901
- @ MCC:mc_colors_A901_Asinh
- @ MCC:mc_colors_AXAF
The *Asinh.prg versions are to be used for a new classification due in summer 2005, while the other versions are the basis of the commonly used catalogues in the J2003 series.

_Last update July 19, 2005, CW_
MCCLASS / ERRORS

Purpose:

Calculation of depth reached in every filter and plot of their flux error distribution for an object table, using the magnitude data \( M \) and their errors \( sM \). These error plots are used as diagnostic tools to search for hidden problems in the data reduction. This command also determines Asinh magnitudes (luptitudes) and errors for all objects which are the basis for improved colour indices and classification results at faint magnitudes and among dropout objects.

Syntax:  

```
MCCLASS/ERRORS [flux-table] [plots]
```

```
flux-table = Name of object table  
(no default)
```

```
plots = Plots for all individual filters?  
(YES=default / NO)
```

Example:  

```
MCCLASS/ERRORS flux16b_Dec00
```

```
MCCLASS/ERRORS fluxA901_J2003 NO
```

Remarks:

The resulting plot pages are saved as postscript files named FE0_{table}.ps, FE1_{table}.ps, etc., where the page FE0 contains a full filter list with depth information and potential warnings. The following pages FE1, FE2,... hold four panels each, one for each filter in the table, and are only created if not deselected with the NO parameter.

Last update July 19, 2005, CW
5.3. CONTEXT: CLASSY

MCCLASS/MAINSEQ or @@ MCC:mc_mainseq_COMBO

Purpose:

Plots diagrams for calibration check of object tables on the basis of colour indices calculated with MCCLASS/COLORS before. A good calibration is necessary for a reliable classification with MCCLASS/CLASSIFY. For COMBO-17 you need a different call than for CADIS, see title.

Syntax: MCCLASS/MAINSEQ [flux-table] [R-limit] [color] [rad] [library]

- flux-table = name of object table  
  (no default)
- R-limit = R-band limit for object selection  
  (default = 22)
- color = screen color used for library stars  
  (default = 3 = Gr(\"u\n)
- rad = limit for radial distance from image center in object selection (only CADIS, not COMBO)  
  (default = 400)
- library = MIDAS table with colour library of stars  
  (default = /photo/user/tbl/COLORLIB/fG_UK131)

Example: MCCLASS/MAINSEQ flux16h_Dec.tbl  
  MCCLASS/MAINSEQ flux_16h_14Jan00.tbl 21 ? 300  

MCCLASS/MAINSEQ ---> prints on-line help text

Remarks:

Colour indices are in units of CD magnitudes. In the CADIS application, the command plots four pages in the graphics window one after another, each with four color-color diagrams containing the library stars (in color green) as well as the point sources from the object table (color black). The selection of these point sources is also constrained by an upper limit in the radial distance from the image center and by an R-band limiting magnitude. The selection for radial position prevents bad objects along the image border from messing up the plot. The magnitude limit prevents a broad scatter in the plot due to photometric errors. Faint point sources are not always stars, but at R=22 about 75% of all objects are still stars, 20% are compact galaxies and 5% are QSOs. So, down to this limit calibration checks should be possible. The four graphics pages are copied into four Postscript files, with names being constructed by the four prefixes HR1, HR2, HR3, and HR4, followed by the name of the object table itself. These files can also be printed in black and white since green library stars will clearly show up as grey.

An on-line help text is available at the MIDAS prompt by sending the command without parameters.

Last update July 19, 2005, CW
MCCLASS / PLOT

Purpose:

Plot the measured SED and the template for an object from a flux table style catalogue. Such plots allow to spot inconsistencies between measured data and best-fitting templates found in the classification.

Syntax: MCCLASS/PLOT [tab] [i] [templ] [class] [z] [ax] [tsize]

  tab      = Name of input object catalogue
            (default: flux_table)
  i        = Number of object in the table
            (default: 1)
  templ    = plot template or not?
            (default: template)
  class    = desired template class name
            (default: :mc_class, i.e. take from column!)
  z        = desired template redshift value
            (default: :mc_z, i.e. take from column!)
  ax       = axis definition for plot panel, reset by
            commands mc_plot_table1/table4/table6.prg
            (default: -160,-60,18,50)
  tsize    = legend text size, reset by
            commands mc_plot_table1/table4/table6.prg
            (default: 1.5)

Examples:  MCCLASS/PLOT cat_A901_J2003c 11756
           MCCLASS/PLOT cat_A901_J2003c 11756 notemp
           MCCLASS/PLOT cat_A901_J2003c 11756 ? star
           MCCLASS/PLOT cat_A901_J2003c 11756 ? qso 5.0

Remarks:

This command looks for the spectral libraries mc_plot_*_spec.tbl in the MID_WORK directory. These are required if template overplotting is desired (get them from $MCC). They should be destroyable copies of the template libraries used for generating colour libraries, because they will suffer from write-access by the plot program and can not be turned into colour libraries any more after this happens. The command creates a plot in the MIDAS graphics window and a postscript file for an A4 size paper print. A number of further parameters can be changed, if the user creates a local copy of the MCC:mc_plot_f04.prg file which then provides write access. The setup variables in the header of the prg available for user manipulation include choices of

- the spectral template libraries for overplotting
- linear or logarithmic wavelength axis
- linear (flux) or logarithmic (Asinh magnitude, needs to exist before plotting) vertical axis
- manual or automatic (driven by filters) plotted wavelength range
- manual or automatic (driven by fluxes) plotted flux range
- manual or automatic (driven by flux errors) weighting for template normalization
- paper or ppt (Powerpoint) style in colours, fonts and label sizes
• values for xmin, xmax, xstep, ymin, ymax in manual range settings

• up to three flux+error columns chosen for the template normalization

• a magnitude column and label to be reported in the text labels accompanying the plot

The command @@ MCC:mc_plot_table1 as well as @@ MCC:mc_plot_table4 and @@ MCC:mc_plot_table6 loop over an entire table and plot all contained objects sequentially. 1/4/6 refers to the number of objects per A4-size page. At six objects per page, the textual data about the objects are dropped to save space for the plot panels.

Sometimes the colour indices in the table are not directly calculated from the flux or magnitude data, but involve corrections for interstellar reddening or for residual calibration errors. These corrections should be specified by the descriptor “cal_corr”. It will also be used to correct the flux or magnitude data in the plot, in order to resemble the colour indices for comparison with the templates.

_Last update July 25, 2006, CW_
Calculating restframe magnitudes and colours for galaxies and QSOs

The multi-colour classification performed in the COMBO-17 and MANOS surveys produces estimates for the redshift and spectral type of galaxies and QSOs which are best fits to the measured multi-band SED. Given the spectral coverage from U- to I-band, it is possible to estimate restframe luminosities within this wavelength range directly from the measured photometry. If a restframe band falls outside the U-to-I range because of the redshift, then estimating the luminosity involves an estimated extrapolation of the SED beyond the range probed by the photometry. This is called K-correction.

The determination of luminosities involves two parts, a cosmological part unique to the redshift value, and a spectrophotometric part unique to the object. The former consists of calculating a distance modulus based on a set of cosmological parameters. In COMBO-17, we have so far published values for \((\Omega_m,\Omega_k,H_0)=(0.3,0.7,100)\). Changing the Hubble constant means only shifting the zeropoint of the distance modulus scale, e.g. \(H_0=70\) means that all distance moduli have to be increased by \(-0.775\) mag. In contrast, changing the cosmological parameters means that the distance moduli change in a non-trivial redshift-dependent way. The prgs for restframe calculation in COMBO-17 also calculate luminosity distances and luminosities for \((\Omega_m,\Omega_k,H_0)=(1.0,0.0,100)\) and \((\Omega_m,\Omega_k,H_0)=(0.2,0.0,100)\).

The spectrophotometric part unique to every object places the best-fitting template into the observed SED by linking it to the measured broadband nearest to the given redshifted restframe band in question (with a bias towards the red, where we expect higher and more reliably measured flux from the object). Here, a K correction table is used which in design looks entirely like a colour library and contains all galaxy (or QSO) templates for the necessary range of redshifts as one object per row with the colour indices in the columns. However, the colour indices in the K correction table are not \(^\text{observed-frame}\)-\(^\text{observed-frame}\) like in the colour library used for the classification; instead they are \(^\text{restframe}\)-\(^\text{observed-frame}\), linking various interesting restframe bands (Jonhson, Bessell, SDSS, ...) to the various observed-frame ones that appear in the survey data (WFI-UBVRI).

The practical procedure to calculate restframe luminosities is then as follows:

- **MCC:mc_restframe_opt03c** (in COMBO-17) or **MCC:mc_restframe_NIR03c** (in COMBO-17+4)
  
  This calculates the (to-be-)observed photon flux in the redshifted restframe band, i.e. the K correction

- **MCC:mc_dls_luminos_03c**

  This calculates luminosity distances, and calculates absolute magnitudes and errors on the basis of cosmological parameters and photometric Vega zeropoints for the bands.

As a result you get luminosities and restframe colours. The errors for the luminosities take into account a constant minimum error for absolute calibration of 0.1 mag, the error of mag_best which defines the total magnitude and hence the aperture correction, and the error of the nearest observed-frame band which drives the normalisation of the template (all added in quadrature). The errors of the restframe colours are just the quadratically added errors from the two observed-frame bands near the two restframe bands combined in the colour index and a 0.1 mag error floor added in again. But the aperture correction to total magnitudes is not used as it does not affect the colour in the absence of colour gradients. With colour gradients being present, it is not clear how to estimate the error from the existing data easily, and we hope that the effect is accounted for by the error floor. Also, the error in mag_best is not included as it does not affect the colours either.

Last update July 27, 2005, CW
5.3. CONTEXT: CLASSY

Description of COMBO-17 Catalogue Columns

Purpose:

COMBO-17 provides object catalogues in FITS format tables, containing more than hundred data columns of information for every object with one object per row. The data columns are listed and briefly explained below, while the columns created by the classification process are explained in more detail in the documentation for the MCCLASS/CLASSIFY command. The column number or order can change, but their labels will remain unique. All object magnitudes (but not the colour indices) are listed in Vega magnitudes.

1. General object description:

<table>
<thead>
<tr>
<th>Col. #</th>
<th>Description</th>
<th>Format</th>
<th>Unit:</th>
<th>Label:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Nr</td>
<td>object number</td>
<td>I8</td>
<td>I*4</td>
<td>object number</td>
</tr>
<tr>
<td>2: ID</td>
<td></td>
<td>A10</td>
<td>C*1</td>
<td></td>
</tr>
<tr>
<td>3: CLASS</td>
<td>morphology class = s/g/u</td>
<td>A12</td>
<td>C*8</td>
<td></td>
</tr>
<tr>
<td>4: X</td>
<td>x-position on deep R-frame</td>
<td>F10.2</td>
<td>R*4</td>
<td>x-position on deep R-frame</td>
</tr>
<tr>
<td>5: Y</td>
<td>y-position on deep R-frame</td>
<td>F10.2</td>
<td>R*4</td>
<td>y-position on deep R-frame</td>
</tr>
<tr>
<td>6: CLASS_VAL</td>
<td>stellarity index 0..1000</td>
<td>I9</td>
<td>I*4</td>
<td></td>
</tr>
<tr>
<td>7: MORPH1</td>
<td>major axis</td>
<td>F7.2</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>8: MORPH2</td>
<td>minor axis</td>
<td>F7.2</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>9: MORPH3</td>
<td>axis angle</td>
<td>F7.1</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>10: FLAG_all</td>
<td>warning flags, 0.7=mostly ok</td>
<td>I8</td>
<td>I*4</td>
<td></td>
</tr>
<tr>
<td>227: mag_best</td>
<td>error of MAG_BEST in mag</td>
<td>F7.3</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>30: add_m_best</td>
<td>Aperture correction (mag)</td>
<td>F12.4</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>31: varflag</td>
<td>variability flag 0 / &gt;=1</td>
<td>F11</td>
<td>I*4</td>
<td></td>
</tr>
<tr>
<td>32: r_a</td>
<td>right ascension (J2000)</td>
<td>R12.7</td>
<td>R*8</td>
<td></td>
</tr>
<tr>
<td>33: dec</td>
<td>declination (J2000)</td>
<td>S12.6</td>
<td>R*8</td>
<td></td>
</tr>
<tr>
<td>229: mu_max</td>
<td>R-band cen. surf. brightness</td>
<td>E12.6</td>
<td>R*4</td>
<td></td>
</tr>
</tbody>
</table>

2. Classification columns (see MCCLASS/CLASSIFY)

<table>
<thead>
<tr>
<th>Col. #</th>
<th>Description</th>
<th>Format</th>
<th>Unit:</th>
<th>Label:</th>
</tr>
</thead>
<tbody>
<tr>
<td>11: MC_class</td>
<td>object class by multi-colour</td>
<td>A16</td>
<td>C*16</td>
<td></td>
</tr>
<tr>
<td>12: MC_z</td>
<td>mean redshift (galaxy/QSO only)</td>
<td>F7.3</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>13: MC_sz</td>
<td>redshift error (1-sigma)</td>
<td>F7.3</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>14: MC_z_bin</td>
<td>redshift bimodality flag</td>
<td>I8</td>
<td>I*4</td>
<td></td>
</tr>
<tr>
<td>15: MC_gSED</td>
<td>mean SED type (galaxy only)</td>
<td>F8.1</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>16: MC_sgSED</td>
<td>SED type error (1-sigma)</td>
<td>F8.1</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>515: MC_gSED2</td>
<td>mean SED2 type (galaxy only)</td>
<td>F8.1</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>516: MC_sgSED2</td>
<td>SED2 type error (1-sigma)</td>
<td>F8.1</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>17: MC_z_ml</td>
<td>probability peak in redshift</td>
<td>F8.3</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>18: MC_gSED_ml</td>
<td>probability peak in SED</td>
<td>I10</td>
<td>I*4</td>
<td></td>
</tr>
<tr>
<td>518: MC_gSED2_ml</td>
<td>probability peak in SED2</td>
<td>I10</td>
<td>I*4</td>
<td></td>
</tr>
<tr>
<td>12: MC_z2</td>
<td>2nd guess redshift if ambiguous</td>
<td>F7.3</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>13: MC_sz2</td>
<td>2nd guess redshift error</td>
<td>F7.3</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>477: chi2red</td>
<td>chi-squared/Nf of best-fit</td>
<td>F7.3</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>19: MC_dist</td>
<td>strangeness in colour space, normalised to sigma(obj-templ)</td>
<td>F8.2</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>20: prob_star</td>
<td>probability to be a FGKM star</td>
<td>F9.1</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>21: prob_gal</td>
<td>probability to be a galaxy</td>
<td>F9.1</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>22: prob_qso</td>
<td>probability to be a quasar</td>
<td>F9.1</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>25: prob wd</td>
<td>probability to be a blue star</td>
<td>F9.1</td>
<td>R*4</td>
<td></td>
</tr>
<tr>
<td>23: MC_nextlib</td>
<td>library ID of best-fitting</td>
<td>I11</td>
<td>I*4</td>
<td></td>
</tr>
</tbody>
</table>
### 3. Galaxy restframe luminosities

<table>
<thead>
<tr>
<th>Col.#</th>
<th>Unit</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>34</td>
<td>d1, 37</td>
<td>Mpc</td>
<td>F8.1 R*8 luminosity distance 0.3/0.7</td>
</tr>
<tr>
<td>35</td>
<td>d1, 01</td>
<td>Mpc</td>
<td>F8.1 R*4 luminosity distance 0.2/0.0</td>
</tr>
<tr>
<td>36</td>
<td>d1, 05</td>
<td>Mpc</td>
<td>F8.1 R*4 luminosity distance 1.0/0.0 all for ( H_0 = 100 \text{ km/sec/Mpc} )</td>
</tr>
<tr>
<td>337</td>
<td>muc, 37</td>
<td>mag</td>
<td>F7.2 R*8 ( M_{280/40} ) (Vega-mag)</td>
</tr>
<tr>
<td>338</td>
<td>muj, 37</td>
<td>mag</td>
<td>F7.2 R*8 ( M_{Johnson-U} )</td>
</tr>
<tr>
<td>339</td>
<td>mbj, 37</td>
<td>mag</td>
<td>F7.2 R*8 ( M_{Johnson-B} )</td>
</tr>
<tr>
<td>340</td>
<td>mvj, 37</td>
<td>mag</td>
<td>F7.2 R*8 ( M_{Johnson-V} )</td>
</tr>
<tr>
<td>341</td>
<td>mus, 37</td>
<td>mag</td>
<td>F7.2 R*8 ( M_{SDSS-u} )</td>
</tr>
<tr>
<td>342</td>
<td>mgs, 37</td>
<td>mag</td>
<td>F7.2 R*8 ( M_{SDSS-g} )</td>
</tr>
<tr>
<td>343</td>
<td>mrs, 37</td>
<td>mag</td>
<td>F7.2 R*8 ( M_{SDSS-r} )</td>
</tr>
<tr>
<td>344</td>
<td>mub, 37</td>
<td>mag</td>
<td>F7.2 R*8 ( M_{Bessell-U} )</td>
</tr>
<tr>
<td>345</td>
<td>mbb, 37</td>
<td>mag</td>
<td>F7.2 R*8 ( M_{Bessell-B} )</td>
</tr>
<tr>
<td>346</td>
<td>mvb, 37</td>
<td>mag</td>
<td>F7.2 R*8 ( M_{Bessell-V} )</td>
</tr>
<tr>
<td>537</td>
<td>smuc, 37</td>
<td>mag</td>
<td>F7.2 R*8 error of ( M_{280/40} ) (Vega-mag)</td>
</tr>
<tr>
<td>538</td>
<td>smuj, 37</td>
<td>mag</td>
<td>F7.2 R*8 error of ( M_{Johnson-U} )</td>
</tr>
<tr>
<td>539</td>
<td>smbj, 37</td>
<td>mag</td>
<td>F7.2 R*8 error of ( M_{Johnson-B} )</td>
</tr>
<tr>
<td>540</td>
<td>smvj, 37</td>
<td>mag</td>
<td>F7.2 R*8 error of ( M_{Johnson-V} )</td>
</tr>
<tr>
<td>541</td>
<td>smus, 37</td>
<td>mag</td>
<td>F7.2 R*8 error of ( M_{SDSS-u} )</td>
</tr>
<tr>
<td>542</td>
<td>smgs, 37</td>
<td>mag</td>
<td>F7.2 R*8 error of ( M_{SDSS-g} )</td>
</tr>
<tr>
<td>543</td>
<td>smrs, 37</td>
<td>mag</td>
<td>F7.2 R*8 error of ( M_{SDSS-r} )</td>
</tr>
<tr>
<td>544</td>
<td>smub, 37</td>
<td>mag</td>
<td>F7.2 R*8 error of ( M_{Bessell-U} )</td>
</tr>
<tr>
<td>545</td>
<td>smb, 37</td>
<td>mag</td>
<td>F7.2 R*8 error of ( M_{Bessell-B} )</td>
</tr>
<tr>
<td>546</td>
<td>smvb, 37</td>
<td>mag</td>
<td>F7.2 R*8 error of ( M_{Bessell-V} )</td>
</tr>
<tr>
<td>601</td>
<td>rf, uv</td>
<td>mag</td>
<td>F7.2 R*8 restframe colour U-V Johnson</td>
</tr>
<tr>
<td>601</td>
<td>smrf, uv</td>
<td>mag</td>
<td>F7.2 R*8 error of restframe colour U-V</td>
</tr>
</tbody>
</table>

... some more of those colour indices, their errors are different from the squared sum of the restframe luminosities as the total mag does not matter.

#### 3b. Quasar restframe luminosities

<table>
<thead>
<tr>
<th>Col.#</th>
<th>Unit</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>337</td>
<td>m145, 37</td>
<td>mag</td>
<td>F7.2 R*8 ( M_{145/4} ) (Vega-mag)</td>
</tr>
<tr>
<td>337</td>
<td>sm145, 37</td>
<td>mag</td>
<td>F7.2 R*8 error of ( M_{145/4} ) (mag)</td>
</tr>
</tbody>
</table>

#### 4. More classification columns, also needed for plotting filter spectra

<table>
<thead>
<tr>
<th>Col.#</th>
<th>Unit</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>42</td>
<td>gz, ml</td>
<td>Unitless</td>
<td>F5.3 R*4 see classification for details</td>
</tr>
<tr>
<td>43</td>
<td>qz, ml</td>
<td>Unitless</td>
<td>E12.6 R*4</td>
</tr>
<tr>
<td>44</td>
<td>sdmg</td>
<td>sigma</td>
<td>F6.1 R*4</td>
</tr>
<tr>
<td>45</td>
<td>gdmg</td>
<td>sigma</td>
<td>F6.1 R*4</td>
</tr>
<tr>
<td>46</td>
<td>qdmg</td>
<td>sigma</td>
<td>F6.1 R*4</td>
</tr>
<tr>
<td>47</td>
<td>wdmg</td>
<td>sigma</td>
<td>F6.1 R*4</td>
</tr>
</tbody>
</table>
5. Photometric measurements

- for every observed filter there is a block of columns like the example given below
- identifiers (420E e.g.) are composed of a filter name (420, R__, B__, 646, etc.) and a letter denoting the observing run (A = Feb 1999 ... G = Feb 2001)
- observing run "S" denotes combined sum measurement of all runs for a given filter
- a given filter observed in different runs can show variation due to long-term variability

...followed by several more 4-column blocks for all other filters.

Extension:
----------

The full catalogue versions known as "FLUX" tables contain many more columns of rather technical meaning and they are usually not relevant for team members. However, if you do require to use them, consult the classification page or the responsible people creating the columns. In the following list column numbers are not to be relied upon as they are taken from various tables.

6. Classification columns for emission-line galaxies (by Hans and Klaus)
CHAPTER 5. DESCRIPTION OF COMMANDS

Col. # 43: maxFsg_B  Unit:       Format: F8.2  R*4
Col. # 44: ELflag_B  Unit:       Format: I8   I*4
Col. # 45: maxsgn_B  Unit:       Format: F8.2  R*4
Col. # 46: ELclass_B Unit:       Format: A12  C*12
Col. # 47: EL_z_B    Unit:       Format: F8.3  R*4
Col. # 48: Sline_B   Unit: W/m^2  Format: E12.4 R*4
Col. # 49: flag_C    Unit:       Format: I6   I*4
Col. # 50: maxFsg_C  Unit:       Format: F8.2  R*4
Col. # 51: ELflag_C  Unit:       Format: I8   I*4
Col. # 52: maxsgn_C  Unit:       Format: F8.2  R*4
Col. # 53: ELclass_C Unit:       Format: A12  C*12
Col. # 54: EL_z_C    Unit:       Format: F8.3  R*4
Col. # 55: Sline_C   Unit: W/m^2  Format: E12.4 R*4
Col. # 56: rflx_NR   Unit:       Format: A8   C*8
Col. # 57: rflx_ID   Unit:       Format: A8   C*8
Col. # 58: rflx_DX   Unit: arcsec Format: F8.2  R*4
Col. # 59: rflx_DY   Unit: arcsec Format: F8.2  R*4
Col. # 60: rflx_LIM  Unit: ph/m^2/s/nm Format: E10.4 R*4
Col. # 61: rflx_FLX  Unit: ph/m^2/s/nm Format: E10.4 R*4
Col. # 62: SCRATCH1  Unit:       Format: G12.4 R*4
Col. # 63: SCRATCH2  Unit:       Format: G12.4 R*4

7. More photometric measurements

- for every observed filter there is a block of columns like the example given below
- identifiers (420E e.g.) are composed of a filter name (420, R__, B__, 646, etc.)
  and a letter denoting the observing run (A = Feb 1999 ... G = Feb 2001)
- observing run "S" denotes combined sum measurement of all runs for a given filter
- a given filter observed in different runs can show variation due to long-term variability

Col. # 64: _F_ 420E  Unit: phot/m^2/s/nm Format: E12.4 R*4 measured aperture flux
Col. # 65: sF_ 420E  Unit: phot/m^2/s/nm Format: E12.4 R*4 pure photon noise
Col. # 66: sFc420E  Unit: phot/m^2/s/nm Format: E12.4 R*4 most likely error
Col. # 67: _Fm_420E Unit: phot/m^2/s/nm Format: E12.4 R*4
Col. # 68: sM_ 420E  Unit: mag Format: F10.3 R*4 aperture magnitude (Vega)
Col. # 69: sc1420E  Unit: Format: E12.4 R*4
Col. # 70: sc2420E  Unit: Format: E12.4 R*4

8. Colour indices and their errors (s_*)

- constructed from the photometric measurements above (see MCCLASS/COLORS)
- corrected for long-term variability and galactic extinction

Col. # 495: wufmwb  Unit: CDmag Format: F6.2 R*4
<table>
<thead>
<tr>
<th>Col. #</th>
<th>Column Name</th>
<th>Unit</th>
<th>Format</th>
<th>R*4</th>
</tr>
</thead>
<tbody>
<tr>
<td>496</td>
<td>wugmwb</td>
<td>CDmag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>497</td>
<td>s_wufmwb</td>
<td>mag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>498</td>
<td>s_wugmwb</td>
<td>mag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>499</td>
<td>wumwb</td>
<td>CDmag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>s_wumwb</td>
<td>mag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>501</td>
<td>wbmvw</td>
<td>CDmag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>502</td>
<td>wvmwr</td>
<td>CDmag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>503</td>
<td>wrmwi</td>
<td>CDmag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>504</td>
<td>w418mb</td>
<td>CDmag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>505</td>
<td>bmw464</td>
<td>CDmag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>506</td>
<td>bmw486</td>
<td>CDmag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>507</td>
<td>v520mr</td>
<td>CDmag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>508</td>
<td>v573dmr</td>
<td>CDmag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>509</td>
<td>v573emr</td>
<td>CDmag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>510</td>
<td>s_w573dmr</td>
<td>mag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>511</td>
<td>s_w573emr</td>
<td>mag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>512</td>
<td>w573mr</td>
<td>CDmag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>513</td>
<td>s_w573mr</td>
<td>mag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>514</td>
<td>w605mr</td>
<td>CDmag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>515</td>
<td>w643mr</td>
<td>CDmag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>516</td>
<td>rmw697</td>
<td>CDmag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>517</td>
<td>rmw752</td>
<td>CDmag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>518</td>
<td>rmw816</td>
<td>CDmag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>519</td>
<td>imw856</td>
<td>CDmag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>520</td>
<td>imw915</td>
<td>CDmag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>521</td>
<td>s_wbmwv</td>
<td>mag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>522</td>
<td>s_vvmwr</td>
<td>mag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>523</td>
<td>s_vrmwi</td>
<td>mag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>524</td>
<td>s_w418mb</td>
<td>mag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>525</td>
<td>s_bmw464</td>
<td>mag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>526</td>
<td>s_bmw486</td>
<td>mag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>527</td>
<td>s_w520mr</td>
<td>mag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>528</td>
<td>s_w605mr</td>
<td>mag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>529</td>
<td>s_w643mr</td>
<td>mag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>530</td>
<td>s_rmw697</td>
<td>mag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>531</td>
<td>s_rmw752</td>
<td>mag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>532</td>
<td>s_rmw816</td>
<td>mag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>533</td>
<td>s_imw856</td>
<td>mag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>534</td>
<td>s_imw915</td>
<td>mag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>582</td>
<td>w486mv</td>
<td>CDmag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>583</td>
<td>w520mv</td>
<td>CDmag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>584</td>
<td>s_w486mv</td>
<td>mag</td>
<td>F6.2</td>
<td></td>
</tr>
<tr>
<td>585</td>
<td>s_w520mv</td>
<td>mag</td>
<td>F6.2</td>
<td></td>
</tr>
</tbody>
</table>

Last update Feb 5, 2004, CW
5.4 Context: synphot

▷ integrate standard star spectra over instrumental response
PHOT_INTEG table = epsilons objects filter mode,SP_table integ_mode redshift

Purpose:
Integrate standard star spectra over instrumental response function.
The standard star spectra are retrieved from a MIDAS table. The instrumental response function is
built from the individual transmission-/response curves like mirror, filter, optics, detector . . . , which are
provided also in the form of MIDAS tables.

Parameters:
\[ p_1 \ \text{out\_table} \] Name of the output table. If table exists, output will be appended.
\[ p_2 = \]
\[ p_3 \ \text{epsilon} \] List of epsilon-vectors. The elements in the list, separated by commas, are identical to the names of the corresponding MIDAS tables.
\[ p_4 \ \text{range} \] Range of objects in the spectrophotometric table \ or 
FIELD,OBJECT names (< 9 characters long. [default: 1,175].
\[ p_5 \ \text{filter} \] Filter name \ or 
filtername, wavelength range [nm] for resulting total efficiency \( e.g. \ V,335,920. \) Default: XXXX
\[ p_6 \ \text{mode} \] Format of the spectrophotometric input table (see 209 and its name. Default: 4,GSATLAS
\[ p_7 \ \text{integration} \] Integration mode is one of AVER,FL, AVER,FN, AVER,CR, PH,GRAV, D,REFRA.
Default: AVER,CR
\[ AVER,FL \ [\text{ergs/cm}^2/s/nm] = \int \epsilon \times CR \times h \nu \, d\lambda / \int \epsilon \, d\lambda \]
\[ AVER,FN \ [\text{mJy}] = \int \epsilon \times CR \times h \lambda \, d\lambda / \int \epsilon \, d\lambda \]
\[ AVER,CR \ [\text{photons/cm}^2/sec/nm] = \int \epsilon \times CR \, d\lambda / \int \epsilon \, d\lambda \]
\[ \text{PH,GRAV} \ [\text{nm}] = \int \lambda \times \epsilon \times CR \, d\lambda / \int \epsilon \times CR \, d\lambda \]
\[ \text{D,REFRA} = \int (n-1) \times \epsilon \times CR \, d\lambda / \int \epsilon \times CR \, d\lambda \]
\[ p_8 \ \text{redshift} \] default: 0.

Examples:

PHOT/INTEG test = mirror,mirror,wfi_B,wfi_optics,wfi_CCD ? B

Use the table GSATLAS to obtain average count rate for each of the 175 objects in
the table.

Actions taken:

1. Create / append output table with integration results

2. The effective \( \epsilon \) is stored in descriptors LAMBIiiii and EPSIIiiii. The effective central wavelength,
FWHM and starting and ending wavelength are stored in descriptors LAMDA,C(i), FWHM(i),
LAM_LOW8i), LAM_HIGH(i), i being the sequence number of the call to this command with the
same output table. (see page 199 for details.)
LAM_LOW and LAM_HIGH should not be confused with LAMDA(3),LAMDA(4), which specify the begin-
ingning and end of the wavelength vector!

3. If only the wavelength / epsilon information is wanted, parameters \( P4 \) and higher may be defaulted.
Although the command will probably crash, the output table will be created and relevant descriptors
will be filled.
Related commands:

Used to calculate library colours for the multi-colour classification.
5.5  Context: ametry

- commands to handle MIDAS context ASTROMET (by R. West)
- utilities to get reference frame
5.5.1 Astrometrie mittels DSS

1. `set/cont ametry`
   Achtung: Der Objektname darf höchstens 6 Zeichen lang sein!

**Plattenlösung auf Schmidtplatte für PPM-Sterne**

2. DSS FITS-File für Objektfeld (typischer Ausschnitt 15') vom Web holen: `Objekt_dss.fits`
   [http://archive.eso.org/cgi-bin/dss/](http://archive.eso.org/cgi-bin/dss/)
   Rotaufnahme vom DSS-II verwenden (Eigenbewegungen)!

3. Auswahl der PPM-Sterne (Koordinaten J2000)
   `ametry/selppm Objekt RAh RAm RAs DECd DECm DECs rad,lim_mag`
   → `Objekt_PPMP_RADEC.tbl`
   Ausgabe der Information auch in die Files `Objekt_PPMP_RADEC.lis / txt` (DOS)
   ASCII-File (DOS-Format) der PPM-Sterne kann auch nochmals erstellt werden durch
   `ametry/listppm Objekt`

4. Ungeeignete Sterne aus `Objekt_PPMP_RADEC.tbl` entfernen
   Sterne mit Flags P (problem), C (critical), D (double) sollten wenn möglich nicht verwendet werden.

5. FITS-Files für PPM-Sterne (Ausschnitt 1') ebenfalls vom Web holen (DSS-II wo immer möglich):
   Hinweis:
   Koordinaten und Nummern aus PPM.lis mit Cut/Paste übertragen oder PPM-Nummer von Simbad auflösen lassen!
   Namen der Files: `PPMi#####.fits`, ##### = PPM-Nummer

6. Katalog der PPM-Fits-Files erzeugen: `create/icat PPM PPM*.fits`

7. Überprüfen, ob alle PPM-Sterne von der gleichen Schmidt-Platte kommen wie das Objektfeld:
   (Descriptor PLTLABEL vergleichen) über
   `ametry/checkppm Objekt`
   Files der Sterne von anderen Schmidtplatten in `Objekt_PPMP_RADEC.tbl` werden dabei umbenannt
   (PPM....fits.wrong), die Tabelle bereinigt und der Katalog der PPM*.fits-Files neu erzeugt.

8. BDFs (Ausschnitte um PPM-Sterne und Feld) erzeugen
   `ametry/inppm Objekt`
   Erzeugt Bild vom Feld/Objekt `Objekt_dss.bdf`, `PPMi#####.bdf` sowie Katalog PPM der BDFs.

9. Positionen der PPM-Sterne (kleine Bilder) messen und in absolute Koordinaten auf der Schmidt-
    platte umrechnen
   `ametry/cenppm Objekt`
   Output geht nach `Objekt_PPMP_pos`

10. Plattenlösung finden über
    `ametry/prim Objekt`

11. Primäre Standards u.U. editieren über
    `ametry/ed_prim Objekt (plot)`
    Dabei muss die SEQ und nicht die ID angegeben werden!
    Einfachere Selektion anhand des durch ametry/prim erzeugten Plots:
    `sel/tab Objekt_PPMP_RADEC :deviat.lt.1`
    `comp/tab Objekt_PPMP_RADEC :std = sel`
5.5. CONTEXT: AMETRY

RA und DEC der sekundären Referenzsterne bestimmen

Vor diesem Schritt müssen folgende Files vorhanden sein:

- Tabelle mit PPM-Sternen und Plattenlösung (*Objekt_PPM_pos.tbl*)
- DSS-Bild des Feldes (*Objekt_dss.bdf*)
- CCD-Bild des Feldes, für das eine Plattenlösung gesucht wird (Bild kleiner 2.5k x 2.5k) (*Objekt_ccd.bdf*)
- Tabelle mit Positionen der Programmobjekte und der sekundären Referenzsterne im Koordinatensystem des CCD-Bildes (Name beliebig, flux_in)

12. Tabelle mit sekundären Referenzsternen aufbereiten

Die Tabelle muss folgende Spalten enthalten: :Nr, :X, :Y, :mag0

Namen dieser Tabelle: *Objekt_pos*

Kopie aus z.B. der CADIS flux-Tabelle:

`project/tab flux in Objekt_pos :Nr,:X,:Y,:mag0;class;class_val`

13. Sekundäre Referenzsterne auf Objektbild auswählen und deren Nummern feststellen:

Dazu Tabelle *Objekt_pos* etwa wie folgt selektieren:

`sel/tab Objekt_pos :class.eq."star*".and.:mag0.ge.17.and.:mag0.le.20`

Zur Identifikation von Hand (optional):

Bild laden, das im Koordinatensystem der Tabelle *Objekt_pos* ist. Danach

`ametry/idsec Objekt_pos Nr`

Sekundäre Referenzsterne und deren Nummern auf Aufsuchkarte vermerken (Hardcopy):

`load/tab Objekt_pos :X :Y :Nr 1 10 3 -1`

14. Positionen der sekundären Standards im DSS-Bild messen

`ametry/censec Objekt modus`

Dieses Kommando stellt drei Modi zur Ausführung bereit:

- **AUTO** (default)

- **SEMI** (noch nicht implementiert)
  Hierbei soll durch manuelle Auswahl einiger Anhaltssterne mit dem MIDAS-Kommando align/image die obige Transformation bestimmt und dann wie oben mit center/gauss die transformierten Koordinaten verbessert werden.

- **MANUAL**
  Zur Ausführung wird die oben erwähnte Aufsuchkarte der sekundären Objekte im DSS-Feld benötigt. Das Bild *Objekt_dss* muß vorher schon so geladen sein, daß der zu nutzende Bereich voll im Display ist! Die in der Tabelle *Objekt_pos* selektierten Objekte werden interaktiv markiert und zentriert.

Ausgabetabelle wird in allen Fällen *Objekt_SEC_pos*.

Die Kontrolle, ob alles korrekt gelaufen ist erfolgt z.B. durch Laden des Bildes und Markierung der Objekte im Overlay-Kanal:

`load Objekt_dss`

`load/tab Objekt_SEC_pos :x_dss :y_dss :ident 1 10 3 -1`

15. RA und DEC der sekundären Standards anhand der Plattenlösung auf dem DSS (Punkt 10) berechnen

`ametry/secrd Objekt`

Ausgabe ist Tabelle *Objekt_SEC_RADEC*, die nicht mehr die PPM-Sterne enthält.
RA und DEC für die Programmobjekte bestimmen

16. Tabelle mit den Koordinaten der Programmobjekte auf der CCD-Aufnahme erzeugen
   ametry/objpos Objekt
   Die Eingabetabelle ist wieder Objekt_pos und kann sämtliche Objekte auf der CCD-Aufnahme enthalten. Die Programmobjekte (am besten inklusive sekundärer Standardsterne) werden über SEL/TAB entsprechend ausgewählt, d.h. gegenüber Schritt 13 ist das Select zu ändern! Wie oben erwähnt, muß die Tabelle die Spalten :Nr, :X, :Y, :mag0 haben. mag_0 ist dabei eine Helligkeitssangabe.
   Ausgabetabelle ist Objekt_OBJ_pos.
   Achtung: Es sieht so aus, als ob bei großer Anzahl Programmobjekte, ein Softwareproblem vorliegt. Die sekundären Standardsterne werden nicht gefunden. In diesem Fall ist es empfehlenswert, in diesem Schritt nur die sekundären Standardsterne zu verarbeiten und die Programmobjekte gemäß Punkt 20 später zu behandeln.

17. Astrometrie auf dem CCD durchführen (Plattenlösung auf dem CCD):
   ametry/field Objekt Grad
   Grad kann dabei LIN (default), QUA oder CUB sein.

18. U.U. durch ametry/ed_sec Objekt (plot) optimieren. Dabei muss die SEQ und nicht die ID angegeben werden!

19. RA und DEC der Feldobjekte bestimmen
   ametry/field Objekt Grad alle
   Das Ergebnis wird in Objekt_RADEC.tbl abgelegt.

20. Soll die Astrometrie auf eine andere Eingabetabelle desselben Feldes angewandt werden, so muß die neue Tabelle den Namen Objektneu_pos erhalten. Erstellen aus z.B. der CADIS-flux-Tabelle: project/tab flux... Objektneu_pos :Nr,:X,:Y,:mag0,:class,:class_val
   Damit wird wieder eine Tabelle der Koordinaten erstellt über ametry/objpos Objektneu.
   Dann müssen die Descriptoren der Tabelle Objekt_OBJ_pos auf die neue Tabelle umkopiert und die astrometrische Berechnung für die neue Tabelle durchgeführt werden. Dazu dient das Kommando ametry/recalc Objektneu = Objekt
   Hier werden im wesentlichen die zur Astrometrie benötigten Descriptoren umkopiert und die Koordinaten mit dieser Plattenlösung errechnet.
   Das Ergebnis ist dann wie im Originalfall eine neue Tabelle names Objektneu_RADEC.
5.5. CONTEXT: AMETRY

Input / output Files

<table>
<thead>
<tr>
<th>Schritt</th>
<th>Filename</th>
<th>Kommando</th>
<th>Beschreibung</th>
</tr>
</thead>
<tbody>
<tr>
<td>in</td>
<td>13h.ccd.bdf</td>
<td></td>
<td>Kopie von sum_BR2_D186.bdf</td>
</tr>
<tr>
<td>in</td>
<td>flux_13h_06Feb02.tbl</td>
<td></td>
<td>CADIS Flußtabelle</td>
</tr>
<tr>
<td>in</td>
<td>sum_BR2_D186.bdf</td>
<td></td>
<td>CADIS Summenbild</td>
</tr>
<tr>
<td>3</td>
<td>13h_PPM_RADEC.lis</td>
<td>ametry/selppm</td>
<td>Listen der PPM-Sterne</td>
</tr>
<tr>
<td>3</td>
<td>13h_PPM_RADEC.tbl</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>13h_PPM_RADEC.txt</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>get_PPM_13h.dat</td>
<td></td>
<td>Batch-File um PPM vom DSS zu holen</td>
</tr>
<tr>
<td>in</td>
<td>PPM159951.fits</td>
<td>get_PPM_13h.dat</td>
<td>Inputfiles, Quelle ESO DSS</td>
</tr>
<tr>
<td>in</td>
<td>PPM160075.fits</td>
<td></td>
<td></td>
</tr>
<tr>
<td>in</td>
<td>13h.dss.fits</td>
<td></td>
<td>Input, Quelle ESO DSS</td>
</tr>
<tr>
<td>6</td>
<td>PPM.cat</td>
<td></td>
<td>Katalog der PPM-Files</td>
</tr>
<tr>
<td>7</td>
<td>13h_PPM_RADEC_orig.tbl</td>
<td>ametry/checkppm</td>
<td>PPM-Stern liegt nicht auf der gleich Platte</td>
</tr>
<tr>
<td>7</td>
<td>PPM160079.fits.wrong</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>13h.dss.bdf</td>
<td>ametry/inppm</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>PPM159951.bdf</td>
<td>ametry/inppm</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>13h_PPM_pos.tbl</td>
<td>ametry/cenppm</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>13h_pos.tbl</td>
<td>project/tab</td>
<td>Erzeugt z.B. aus CADIS-Flußtabelle</td>
</tr>
<tr>
<td>14</td>
<td>13h_SEC_pos.tbl</td>
<td>ametry/censec</td>
<td>Hilfstabelle</td>
</tr>
<tr>
<td>14</td>
<td>13h_ccd.tbl</td>
<td>ametry/censec</td>
<td>Hilfstabelle</td>
</tr>
<tr>
<td>15</td>
<td>13h_SEC_RADEC.tbl</td>
<td>ametry/secrd</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>13h_OBJ_pos.tbl</td>
<td>ametry/objpos</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>13h_RADEC.tbl</td>
<td>ametry/field</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>13hall_pos.tbl</td>
<td>project/tab</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>13hall_OBJ_pos.tbl</td>
<td>ametry/objpos</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>13hall_RADEC.tbl</td>
<td>ametry/recalc</td>
<td></td>
</tr>
<tr>
<td>out</td>
<td>ametry_13h_06Feb02.tbl</td>
<td></td>
<td>Endergebnis, Kopie von 13hall_RADEC.tbl</td>
</tr>
</tbody>
</table>

DSS files im Batch mode

Datenfile
10h_01_ 10 00 57.32 2 48 11.0 35 35
10h_02_ 9 59 02.53 2 48 11.0 35 35
10h_03_ 10 00 57.30 2 19 31.0 35 35
10h_04_ 9 59 02.55 2 19 31.0 35 35
10h_05_ 10 00 57.30 1 50 51.1 35 35
10h_06_ 9 59 02.58 1 50 51.1 35 35
10h_07_ 10 00 57.29 1 22 11.1 35 35
10h_08_ 9 59 02.59 1 22 11.1 35 35

Abruf der Files vom ESO DSS-Server (hier dss2 red) mit
dss2 red -i filename
Die Files heißen dann 10h_01_irgendwas
CHAPTER 5. DESCRIPTION OF COMMANDS

Figure 5.2: Flow chart for astrometric reduction.
Chapter 6

WFI-specific commands

This is a collection of commands specific to WFI. Only WFI/debias and WFI/mosaic are adopted from the original WFI pipeline. These new scripts are not part of the official MPIAPHOT package but are only meant as a suggestion on how one might tackle the data.

Note:
Commands start with wfi/qual instead of qual/wfi as in the WFI pipeline. Another important change is the definition of the mask content. Previously bad pixels or gaps had a value of zero and valid pixels of one. This is now reversed for clarity and to differentiate from the use in the pipeline.

6.1 Outline of WFI data reduction

- data input (wfi/indisk or name/hirocs)
- subtract bias and correct non-linearity (wfi/debias)
- create a gap mask (@@ WFI:WFI_gaps)
- insert 8 individual images belonging to one exposure into a single frame (wfi/mosaic) using the gap mask
- create frame-table (frame/create)
- flag short exposures (frame/short)
- set keyword EPSILIST, write descriptors FILTER and LAMBDA to all mosaic frames
- run check/descr on all mosaic frames
- create input frames for mask creation from LAMP,BETA images using bias/extra. See below on how to retrieve the LAMP,BETA images.
- create bad column masks using the results from bias/extra on the LAMP,BETA images:
  - @@ WFI:WFI_analyse to analyse the bad pixels
  - @@ WFI:WFI_create_mask to create a bad pixel mask
  - @@ WFI:WFI_create_bad_column_mask to isolate bad columns and remove them from the bad-pixel-mask
  - @@ WFI:WFI_create_column_offsets to create the mask for the correctable columns from the const output from bias/extr.
    Use @@ WFI:WFI_flatten_chips on the const file to flatten it.
  - run PREP/CCD on the LAMP,beta images to subtract the column offsets. Then calculate representative ratios to see the remaining bad columns.
use REPLACE/IMA to create a mask for the bad pixels and run WFI:WFI_create_bad_column_mask to create the bad-columns masks for different background levels.

- WFI:WFI_define_bad_columns to interactively improve the really-bad-columns mask (especially at the end of the bad columns).
- add up the various bad-pixel and bad-column masks for the final result.

A procedure, which combines the above sequence is found at WFI:WFI_prepare_column_offsets and WFI:WFI_make_column_offsets. This worked fine for me but may have to be adjusted to other needs!

- run PREP/CCD to subtract the column offsets from all images.
  To correct for the (constant) column offsets do not use the mask parameter in the call to PREP/CCD but use the column offset image as the DARK0-file to subtract the column-offsets for all input images.

- apply the bad-column-mask to all images, depending on the exposure level (WFI:WFI_patch_bad_pixels)

- create twilight flats (flat/aver)
- flatfield-correct images (corr/ima)
- flatten images (flat/back or WFI:WFI_flatten_chips)
- determine orientation of (sub)images with find/obj, undistort/table and find/move
- fill frame-table with results (frame/fill)
- create gnomonic images (mosaic/gnom)
- construct median image for sub-sets of about equal seeing (±0.2″...0.3″) from gnomonic images (mosaic/median)
- correct cosmics in gnomonic images (mosaic/rmcos) and check cosmic correction in the vicinity of stars.
- transfer cosmic information to original (flat) images (mosaic/copycos)
- build sum frame (mosaic/sumup)

Please note that bad frames should be flagged throughout the reduction process in the frame table (frame/flag).

6.2 How to retrieve the LAMP,BETA images

Go to the ESO archive page http://archive.eso.org/eso/eso_archive_main.html.

- In the blue field in column IMAGING select 2.2/WFI
- In the yellow field at upper right in row TYPE → User defined input insert LAMP,BETA
- to reduce the large number of entries give a limit in START and END date (pink area at upper right)
- run query

Please note that the keyword LAMP,BETA was in use only since August 2004. To retrieve earlier data (since about mid 2001) set Instrumental Setup → Filter in the yellow area to MISC#_BTLGHT_WFI.
Figure 6.1: Web page for retrieval of LAMP.BETA images.
wfi/indisk OUT_name = IN_list flag

Purpose:
Read WFI FITS file and convert to header file plus 8 data files in BDF-format.

Parameters:

\[ p_1 \text{ out} \text{name} \] first four characters (root) for name(s) of output images

\[ p_2 = \]

\[ p_3 \text{ IN_list:} \] list of input header frames (see page 195)

\[ p_4 \text{ naming flag} \] Renumber files according to input list or keep file names. Default: keep

Examples:

```
    wfi/indisk raw_ = all_fits:
```

Actions taken:

1.

Related commands:
wfi/debias OUT_name = IN_list

**Purpose:**
Subtract the bias and convert from counts to electrons. This command is identical to the one in the WFI pipeline.

**Parameters:**
- \( p_1 \) out_name: first four characters for name(s) of output images
- \( p_2 = \)
- \( p_3 \) IN_list: list of input header frames (see page 195)

**Examples:**
- wfi/debias debi = headers:

**Actions taken:**
1.

**Related commands:**
wfi/mosaic OUT_name = IN_list mask

**Purpose:**
Create a single mosaic file from the header and the 8 data files. This command is different to the one in the WFI pipeline! The images are no longer skewed because the different chip orientation is taken care of by the mosaic commands.

**Parameters:**
- $p_1$ **out_name**: first four characters (root) for name(s) of output images
- $p_2$ =
- $p_3$ **IN_list**: list of input header frames (see page 195)
- $p_4$ **mask**: If name of a mask is given, the mask will be used to patch bad pixels (*i.e.* set their value to DATA_LIMIT. Default: no_mask

**Examples:**
- wfi/mosaic mosa = headers:

**Actions taken:**
1.

**Related commands:**
6.2. \textit{HOW TO RETRIEVE THE LAMP, BETA IMAGES}

\texttt{@@ WFI:WFI\_cp\_date\_time IN\_list}

\textbf{Purpose:}
Set \textit{MPIAPHOT} conform date and time values from header. This prg is called automatically from check/dscr if the input files are WFI data.

\textbf{Parameters:}
\begin{itemize}
  \item \texttt{p1 IN\_list:} list of input frames (see page 109)
\end{itemize}

\textbf{Examples:}
\texttt{@@ WFI:WFI\_cp\_date\_time all\_frames:}

\textbf{Actions taken:}
\begin{itemize}
  \item 1.
\end{itemize}

\textbf{Related commands:}
check/dscr
@@ WFI:WFI_analyse IN_name hist_range ord_range chip cursor significance gap.mask

Purpose:
Use output from bias/extrapolate to isolate bad pixels via the graphic display.
Optionally (parameter 5) the graphics cursor may be used to define the good range. In this case the global keyword WFI_RANGE is filled with the upper and lower boundary. This keyword can then be used in WFI_create_mask.

Parameters:

\( p_1 \) **IN_name** Full name of input image (= on of the output images of bias/extra

\( p_2 \) **hist_range** range for histogram [no default]

\( p_3 \) **ord_range** Start and end value for ordinate on graphics display [default: auto]

\( p_4 \) **chip** Range of detectors or single chip number [default: 1,8]

\( p_5 \) **cursor** Cursor input desired or not [default: no_cursor]

\( p_6 \) **significance** Desired significance in units of rms for screen output only [default: 5]

\( p_7 \) **gap.mask** Name of mask file for gaps [default: wfi_gaps]

Examples:

@@ WFI:WFI_analyse Nov04.rms 0,500 0,1.5e5
   Plot for all detectors

@@ WFI:WFI_analyse Nov04.rms 0,500 0,1.5e5 3
   Plot for detector #3

Actions taken:

1. Keyword WFI_RANGE is written in interactive mode.

Related commands:
@ WFI:WFI_create_mask OUT_name = reference_image chip_no range

Purpose:
Create a mask from a reference frame (usually one of the output frames of bias/extr with information provided by WFI_analyse) for given detector and insert it into the mask. An existing mask is updated.)

Parameters:
- $p_1$ file_name: Name of mask to be created or updated.
- $p_2 = p_3$ reference: Name of reference file, according to which the mask is created. Usually this is one of the output files of bias/extrapolate.
- $p_4$ detector: Detector number (1 to 8), clockwise starting at upper left
- $p_5$ range: Range of good values in reference frames. Values outside this range will be masked (i.e. set to 1). If content of keyword WFI_RANGE is to be used, enter the string keyword instead of numerical values.

Examples:
- @ WFI:WFI_create_mask mask_Nov04 = Nov04_slope 1 68,491

Actions taken:
1. Creates and removes scratch files scr_replace.bdf and scr_mask.bdf

Related commands:
@@ WFI:WFI_create_bad_column_mask bad-column-mask = bad-pixel-mask thresholds

Purpose:
Analyses bad-pixel-mask created by WFI_create_mask and search for bad columns. Eliminate the bad columns in the input file and create a separate mask for bad columns.

Parameters:
- $p_1$ bad-column mask Name of bad-column mask (bcm), good pixels = 0, bad pixels = 1.
- $p_2 =$
- $p_3$ bad-pixel mask Name of bad-pixel mask (bpm), good pixels = 0, bad pixels > 0
- $p_3$ thresholds Lower limit for fraction of good pixels in window, window size. [default: 0.98,200]

Examples:
@@ WFI:WFI_create_bad_column_mask bcm_Nov04 = bpm_Nov04

Actions taken:
1.

Related commands:
@@ WFI:WFI_create_mask
6.2. HOW TO RETRIEVE THE LAMP,BETA IMAGES

@@ WFI:WFI_define_bad_columns mask zoom

**Purpose:**
Interactively identify bad areas with start and end pixel and create/update mask. Image on the display is used as source.

This command should be used after the mask has been created with @@ WFI:WFI_create_bad_column_mask, because the result of the latter is only a first approximation.

As the cursor reading precedes the entry of the action code, this command is a bit tricky to use. Make cursor readings slowly and wait for display to refresh. Depending on the action code, the cursor reading may be discarded, *e.g.* if the overlay channel is to be cleared via action "o" or if the column mode is to be entered. Hitting the right mouse button twice in the display window brings back the options menu.

**Parameters:**

- $p_1$ mask
  
  Name of mask to be created or updated

- $p_2$ zoom
  
  The zoom factor for the zoom window.

**Examples:**

@@ WFI:WFI_define_bad_columns bcm_Nov04

**Actions taken:**

1. On the display (zoom window) mark first lower left and then upper right corner of bad area

2. Following keys are functional:
   
   - e = exit
   - s = skip, *i.e.* do no use current reading(s)
   - o = clear/chan over
   - CR = enter marked areas to table. More than area may be marked within the zoom window.
   - n = enter values for one area numerically
   - c = column mode: The cursor reading is discarded and the column mode is entered.
   - First select lower end region of bad column and then mark starting pixel in zoom-window.
   - Next select upper end region and then mark end pixel in zoom-window.
   - The whole range between these markings will be masked.

3. **Important:** Image is updated directly! This is an irreversible process!

**Related commands:**

bias/extra  @@ WFI:WFI_create_bad_column_mask
@@ WFI:WFI_create_column_offsets offset-mask = input thresholds

**Purpose:**
Analyze *e.g.* `const` output file from `bias/extra` and search for columns with constant offset. This mask can then be applied with `prep/ccd` as the `dark0`-file.

**Parameters:**
- $p_1$ offset mask Name of offset mask
- $p_2 =$
- $p_3$ bad-pixel mask Name of input file, usually the `cons`-file from `bias/extra`
- $p_4$ thresholds Lower limit for fraction of good pixels in window, window size, noise limit. [default: 0.5, 200, 10]

**Examples:**
@@ WFI:WFI_create_bad_column_offsets Nov04_offset = Nov04_cons

**Actions taken:**
1.

**Related commands:**
`bias/extrapolation`
6.2. HOW TO RETRIEVE THE LAMP, BETA IMAGES

@@ WFI:WFI_check_bad_columns IN_list flatfield mask

Purpose:
Utility to check bad-column-mask on image display. Input images will be flatfielded and displayed together with the mask to check the level of flatfield accuracy. The zoom-window is used. Used normally to check lower part of northern chips and upper part of southern chips, i.e. around the horizontal gap where bad columns start.

Parameters:
- \( p_1 \) IN_files List of images with long and short exposure times
- \( p_2 \) flatfield Name of flatfield file.
- \( p_3 \) mask Name of bad-column-mask to be checked.

Examples:
```
@@ WFI:WFI_check_bad_columns check: flat mask
```

Actions taken:
1. Cursor functions
   - \( y \) continue with next horizontal stripe
   - \( c \) display cursor to measure position
   - \( d \) display again
   - \( a \) abort
   - \( n \) next frame
   - \( b \) back one step

Related commands:
@@ PM:find_satur_pixels mask = input-list threshold

Purpose:
Check for pixels which are saturated in > total number of images × threshold of the images. These are flagged in the output image.

Parameters:
- $p_1$ output mask  Name of output mask
- $p_2 =$
- $p_3$ input list  Name of input files
- $p_4$ threshold  Lower limit for unsaturated fraction of pixels [default: 0.5]

Examples:

@@ PM:find_satur_pixels mask_Nov04_add = Nov04_prep: 0.7
Check all frames in catalogue Nov04_prep and flag a pixel if it is saturated in more than 70% of the frames.

Actions taken:
1.

Related commands:
6.2. HOW TO RETRIEVE THE LAMP, BETA IMAGES

@@ WFI:WFI_patch_bad_pixels IN_list mask value

Purpose:
Set bad pixels to value.

Parameters:

- $p_1$ IN_files: List of images to be patched.
- $p_2$ mask: Name of bad-pixel-mask.
- $p_3$ mask: Value to which bad pixels are to be set. Default: -1000.

Examples:

@@ WFI:WFI_patch_bad_pixels mosa: mask_Nov04

Actions taken:

1.

Related commands:
Purpose:
Determine coordinates of detector coordinates for given image and detector number. Output will be written to keywords INPUTC and INPUTi, INPUTr and INPUTd (see PM:corners for details).

Parameters:
- $p_1$ file_name: Name of images
- $p_2$ detector: Detector number (1 to 8, clockwise starting at upper left)

Examples:
@@ WFI:WFI_chip_corners raw_0001 3

Actions taken:
1. see PM:corners

Related commands:
@@ PM:corners
@@ WFI:WFI_gaps OUT_name value_chip value_gap

**Purpose:**
Create a mask to flag pixels in the gaps. This is a single mosaic file with chip area set to `value_chip` and gaps set to `value_gap`.

**Parameters:**
- $p_1$ `out_name` Name of gap mask image [default: wfi_gaps]
- $p_2$ `chip` Data value for chip areas [default: 0.]
- $p_3$ `gaps` Data value for gaps [default: 1.]

**Examples:**
@@ WFI:WFI_gaps wfi_gaps_map

**Actions taken:**
1. 

**Related commands:**
@@ WFI:WFI_aver_row IN_name

Purpose:
Use MIDAS command aver/row for each detector individually.

Parameters:

\[ p_1 \text{ IN_name} \]
Full name of input image (= on of the output images of bias/extra).

Examples:

@@ WFI:WFI_aver_row Nov04_rms
Create files Nov04_rms_row1 ... Nov04_rms_row8, one file for each detector.

Actions taken:

1.

Related commands:
6.2. HOW TO RETRIEVE THE LAMP, BETA IMAGES

6.2.1 Elapse times for WFI data reduction

The following table gives some information about the run-time for MPIAPHOT-task during a WFI reduction on aida72. The data input consisted of a total of 672 images, 566 of which were processed until the find/obj stage. After this the images were restricted to 202 long and 88 short exposures of one square degree collected for HIROCS in 6 observing runs.

<table>
<thead>
<tr>
<th>Task Description</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>mosaic</td>
<td>1.5 h</td>
</tr>
<tr>
<td>prep/ccd (subtraction of column offsets)</td>
<td>1.0 h</td>
</tr>
<tr>
<td>application of mask</td>
<td>1.0 h</td>
</tr>
<tr>
<td>create dusk flats</td>
<td>1.4 h</td>
</tr>
<tr>
<td>correct</td>
<td>1.0 h</td>
</tr>
<tr>
<td>flat/back</td>
<td>5.0 h</td>
</tr>
<tr>
<td>find/obj / undistort</td>
<td>0.8 h</td>
</tr>
</tbody>
</table>

566 up to here, 202 + 88 images thereafter

<table>
<thead>
<tr>
<th>Task Description</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>find/move</td>
<td>0.1 h</td>
</tr>
<tr>
<td>mosaic/gnom</td>
<td>1.7 h</td>
</tr>
<tr>
<td>mosaic/median (194 images)</td>
<td>0.9 h</td>
</tr>
<tr>
<td>mosaic/rmcos</td>
<td>2.0 h</td>
</tr>
<tr>
<td>mosaic/copycos</td>
<td>1.0 h</td>
</tr>
<tr>
<td>mosaic/sumup</td>
<td>1.0 h</td>
</tr>
</tbody>
</table>

total 18.4 h
Chapter 7

O2k-specific commands

This is a collection of commands specific to O2k. They are not part of the "official" \textit{MPIAPHOT} software but have been written by HJR to facilitate the adoption of the O2k pipeline output to \textit{MPIAPHOT} conventions. Some additional features unique to OMEGA2000 are also covered. Please check the header of the PRGs directly for further documentation.

\textbf{Note:}

No special commands are provided. Use @@ PM:... instead! It may be necessary that you modify these little PRGs to meet your needs.

Please keep in mind that the O2k-pipeline knows nothing / does not care about \textit{MPIAPHOT} and its descriptors or keywords.

7.1 Suggestion how to reduce OMEGA2000 data

Details are also to be found in the diploma thesis \textsuperscript{[2]} and in the PhD thesis \textsuperscript{[3]} of René Faßbender and in the OMEGA2000 manual.

- Data input (\texttt{name/hirocs})

- Fix the header keywords with @@ PM:O2k\_fix\_descrs. This command does a check/descr at the end. Wrong \texttt{O\_POS} should be fixed with @@ PM:O2k\_corr\_opos before this stage.

  \textbf{Warning:} Do not use any MIDAS commands on the FITS-files before the header is fixed this way. Otherwise you may loose the DATE\_OBS and O\_POS information!

- Create frame table (\texttt{frame/crea}) holding all images.

- Check for bad images (you might want to use @@ PM:movie for this, see page \textsuperscript{[39]} and flag them (\texttt{frame/flag}).

- Create a bad-pixel mask from dome flat and dark series with increasing exposure time (use output from \texttt{bias/extrapolate}).

- Create multiplicative flatfield from twilight flats using the domeflats as a "catalyst" to remove the fixed-pattern noise before averaging them. Alternatively construct the flat from the science frames themselves. Again, a domeflat should be used as "catalyst" to remove the fixed-pattern-noise. See @@ PM:O2k\_create\_science\_flats as an example. To conserve the count level the flatfield used in the pipeline has to be normalized to 1 !

- No dark frames are needed as the dark is best subtracted together with the sky. Use a dummy file to supply it to the pipeline.

- Create the object catalogue (\texttt{frame/cat}) for the current subset from the frame table.
• Run the Faßbender pipeline `o2k/science` (context `omega2k`).
  Sometimes the pipeline fails. In these cases a change in the sequence of files may help. Another reason may be that `OPOS` is wrong, as this is used as a first approximation to the dither offsets (see above).
  Check the output text files for any omitted files.

• Check the object mask and adjust threshold if necessary.

• Experience has shown that the multiplicative flatfield corrected images come at best to within 20% of the photon noise in the background. Sometimes the noise is considerably higher. Thus subtraction of the sky images will remove the pixel-to-pixel noise additively thus mimicking a background noise which is far too low. It is thus best to smooth the sky images before subtraction (see `PM:O2k_smooth_sky.prg`).

• Convert the pipeline output (sky-subtracted images from second pass) to the `MPIAPHOT` standards with `PM:O2k_re2corr`. This does the following:
  – Create HIROCS names.
  – Runs `PREP/CCD` to convert from counts to electrons.
  – Runs `CORR/IMA` to convert to world coordinates.
  – Runs `PM:O2k_flatten_quadrants` to remove any quadrant pattern.

• Run `find/obj` and `find/move` on the single images.

• Check the photometric quality of the flatfielding with the output of `find/move`.
  (column :mag using e.g. `PM:check_flat`)

• Flag images for which `find/obj` and/or `find/move` failed (e.g. bad image shape due to telescope jump etc.).

• Fill the frame-table with the results from `find/object` and `find/move` (frame/fill). Flag frames with bad seeing, low transmission, and high background (use `plot/table` and `stat/table` with a suitably selected frame table).

• Produce the gnomonic projection images (`mosaic/gnomima`).

• Create the median image (`mosaic/median`), probably for subsets with similar seeing ($\approx \pm 0.3\arcmin$).
  Then remove the cosmic ray events and replace bad pixels based on the bad-pixel mask (see `mosaic/rmcos` on page 76 for details on how to include the bad-pixel-mask in this process.).

• Check that the cosmic correction did not affect the centres of stars and that the number of cosmic events detected is reasonable.

• Create sum image (`mosaic/sum`) in gnomonic projection.

• Transfer the cosmic correction information to the original, flat field corrected ("corr") frames (`mosaic/copycos`).

• Create suitable sub-sums of the cosmic-corrected (non-gnomonic) images using
  `o2k/pipe icat=... sum=11,1,2 kappa=1.e20 cuts=0.5,3.0` (adjust parameters to your needs).
  Important is the high `kappa` to suppress cosmic removal and the 2. parameter in `sum`, which is set to 1 for "summation only". Check the pipeline documentation for further details.
  Alternatively the command `sum/ima` may be used, which does a straight-forward pixel-wise summation, taking care of descriptors like exposure time and saturation.
  Rename the sub-sums according to HIROCS standards and update their descriptors.

• Follow standard `MPIAPHOT` data reduction from here on.

Please note that bad frames should be flagged throughout the reduction process in the frame table.
Chapter 8

LBC-specific commands

This is a collection of commands specific to LBC.

To be able to reduce LBC-data with \textit{MPIAPHOT} two commands have been provided by Sebastian Jester. The first standardizes the header information and converts the FITS-file to MIDAS-bdf. The second generates a mosaic of the 4 LBC-images which are from thereon treated as a conventional image. LBC images are treated analog to WFI images.
LBC/INGEST list campaign catalogue FITS-filter FITS-OB FITS-instr FITS-typ

Purpose:
Read the LBC header and convert keywords to standard \textit{MPIAPHOT} descriptors. The FITS-files are converted to BDF.

Parameters:

\begin{itemize}
\item \texttt{p1 list} list of FITS input files (see page 195).
\item \texttt{p2 campaign} 4-character identifier of the campaign the data were taken. Format YYMM, no default.
\item \texttt{p3 IN_list:} Name of catalogue to be created with names of newly created images. The catalogue only holds the frames for chip 1, as needed by LBC/prep.
\item \texttt{p4 filter} Name of FITS header keyword specifying the filter used [default: \texttt{FILTER}].
\item \texttt{p5 OB name} Name of FITS header keyword specifying the OB executed [default: \texttt{LBCOBNAM}].
\item \texttt{p6 instrument} Name of FITS header keyword providing the instrument’s name [default: \texttt{INSTRUME}].
\item \texttt{p7 instrument} Name of FITS header keyword giving the image type [default: \texttt{IMAGETYP}].
\end{itemize}

Examples:

\begin{itemize}
\item \texttt{LBC/ingest all.fits: 0803 raw}
\end{itemize}

Takes LBC-FITS-files in catalogue \texttt{all.fits}, extracts the 4 files to BDF and adjusts the descriptors to the \textit{MPIAPHOT} conventions. Output catalogue \texttt{raw} is created, ready for input to LBC/prep.

Actions taken:

\begin{enumerate}
\item Convert FITS to BDF
\item Update descriptors
\item Create 28-digit \textit{MPIAPHOT}-name
\item Create output catalogue of raw frames (chip #1 only)
\end{enumerate}

Following this command the user needs to specify keyword \texttt{EPSILIST} and then call \texttt{CHECK/DESCR}.

Related commands:

\texttt{check/dscr LBC/prep}
LBC/PREP out = list catalogue

**Purpose:**
Subtract the bias according to the overscan and create a mosaic image from the 4 CCDs making up an LBC image.

**Parameters:**
- \( p_1 \text{ out\_root} \)  
  First four characters (root) for name(s) of output images
- \( p_2 = \)  
- \( p_3 \text{ IN\_list:} \)  
  List of input frames (see page 195)
- \( p_4 \text{ catalogue} \)  
  Name of catalogue which will hold the names of the newly created images.

**Examples:**
```plaintext
LBC/prep prep = raw: prep
```
Use all images, whose chip #1 frame is listed in catalogue `raw` and create an LBC mosaic. The name of the mosaic images will start with `prep`. These names are assembled in catalogue `prep`.

**Actions taken:**
1. determine bias level from overscan for each of the 4 images and subtract bias from data.
2. extract the illuminated area for each detector
3. assemble the 4 images into a mosaic with the images 1 to 3 at the bottom and image #4 rotated and placed above them.

**Related commands:**
LBC/ingest
Chapter 9

Glossary

**CCD**  charge coupled device  
**MIDAS** Munich Image Data Analysis System  
**dithering** Observing mode where telescope is moved by a small amount between multiple images of the same field. Thus objects in the individual frames do not always fall onto the same area on the detector.  
**Savart plate** Double calcite plate  
**Synthetic photometry** A model of the system is analysed to obtain the conversion factor from observed photons to flux of the object outside the atmosphere
Bibliography


Appendix A

Utilities

A.1 Frame_list

MIDAS catalogues are used to create lists of input frames/tables. The catalogues have to be created via standard MIDAS commands like `create/icat mycat *.bdf` or via the frame table (see page 54).

If a path is included in the specification of the input files, the path is also included in the catalogue. Some — but not all — *MPIAPHOT* routines can handle catalogues with paths included. This is convenient if input files come from different sub-directories, like in the *mosaic*-commands. Maximum length of file name (including path) is currently 64 characters.

The string specifying the frame list may contain one of the following:

- catalogue name followed by a colon (:) (→ frame list will contain all frames in catalogue) (If all frames from the active catalogue are to be used, only a colon should be given!)
- cat_name:list (e.g. list=1,3–7,9,15–19) to select from cat. If only a list preceded by a colon is given, frames are taken from the active catalogue.
- frame list, names separated by commata Here frames from active catalogue may also be referenced via #no!

Blanks in the string are not allowed! If a catalogue is used the frames have to be specified in ascending catalogue number.

Examples:

: use all of active catalogue
:1,5-7,10 use frames 1, 5 to 7 and 10 from active cat.
CATAL: use all of CATAL.CAT
CATAL:1,5-7,10 same as 2. example, but for CATAL.CAT
FRAME1,FRAME5,FRAME6,FRAME7,FRAME10 use frames given
#1,#5,#6,#7,#10 same as 2. and 5. example, if active cat. holds FRAME1 to FRAME10

Looping a MIDAS command over a frame list:

This only works if the input frame/table is the first argument of the MIDAS command. Only 7 arguments of a MIDAS command can be used!

Examples:

Read descriptor EXP_TIME for all frames in a frame list: `proc/ima read/descr cat:1-3,7-9 step`

Read the number of rows used in a table with `proc/tab read/descr tblcat: tblcontr/i/4/1`

Error codes:

STATUS = 0 all okay
9999 too many frames (max. number possible taken for computation)
−9999 fatal error, should not occur!
−1 catalogue not found or empty
−2 syntax error in frame list
= or > 1 not all frames found

If STATUS > 0 the caller may continue without danger.

Related commands: LIST/NAMES, GET/NAMES

A.2 File name conventions

A.2.1 HIROCS file name convention

File-names are 28 characters long and divide into two logical parts:

- first 4 characters change during reduction process (raw data: raw.)
- last 24 characters always remain and describe the data set.

Convention: filename = xxxxFFFF_ffhcppc_IIoooo_iiii

xxxx identify the reduction stage, e.g. prep

FFFF identifies the filter used (use B, R, i, z, H, I314)

ffh is the HIROCS field (03h, 10h, 12h, 16h, 22h)

For LAICA and OMEGA2000:

pp is the LAICA pointing (1 ... 12)

(pi for relative photometric calibration files, i = 1,...0)

c is the LAICA chip (also used for OMEGA2000) a, b, c, d

 a b

 c d

(p for relative photometric calibration files)

For WFI:

pp field (A, B or C) followed by an underscore

c subfield a,b,c or d (LAICA orientation)

III instrument used (WFI, LAI, O2k)

ooooo identifies the campaign by its date YYMM

iiii is the frame number in that period.

For short exposures substitute the last underscore (before iiii) by an s.

In case of calibration files etc. the ffhcppc is replaced by

Darks dark0_ or darkX_ or dark_

Flats dome_ dusk_ dawn_ flat_

Focus focus_

Standard star phot_

Comparison lamp arc_

Sky field sky_

Test test_ or other_

Astrometry astro_

Acquisition AQ__
A.3. ERROR TREATMENT

For LAICA the last underscore is replaced by the chip indicator (a, b, c, d).
For OMEGA2000 the pipeline results are pipe... and the sub-sums sub....

Examples:

```
raw_H_03h12c_02k0309_1234.bdf
raw_I814_dome_a_LAI0309_1235.bdf
```

### A.3 Error treatment

If an *MPIAPHOT* task is called within a user’s own prg there are several possibilities to check, if all went well. In case of an error the MIDAS descriptor `PROGSTAT(3)` is always set to the error code. If `PROGSTAT(3)` is zero, all went well.

In many instances one has to run a command over a large number of frames. Currently the philosophy is to abort a task if e.g. a descriptor is missing, instead of just skipping this frame.

A few task have now been equipped with a new feature. In case of problems the task is not aborted but problematic frames are entered into the catalogue `bad_frames`. The IDENT column of this catalogue contains a hint to what went wrong, instead of the real frame identifier. To check for problems use `@@ PM:check_err`. Upon return the character (!) keyword `Q1` contains information about the catalogue `bad_frames`. It is "o.k." if the catalogue is empty, i.e. no frames were skipped. Otherwise it contains the number of entries in the catalogue. Keyword `Q2` returns the content of `PROGSTAT(3)`, which should be zero in case of no problems. You may use this as follows, e.g. to check if find/obj was successful:

```
find/obj test
@@ PM:check_err
if Q1(1:4) .ne. "o.k." .or. {Q2} .ne. 0 then
   write/out "Error in find/obj"
   goto exit
endif
```

You may then just search the log-file for the word "Error" to see, if in a long script something went wrong.

To deal with the problematic cases one just needs to rename the catalogue `bad_frames` and call the task again with improved parameters and the new catalogue as framelist.

Tasks currently supporting the `bad_frames` catalogue:

- stat/back
- find/obj
- find/move
- mosaic/rmcosmics
- mosaic/copycos
- find/ring
- flux/selnorm
- flux/norm
Appendix B

Descriptors

The following is a list of descriptors that are checked for their existence in CHECK/DESCR or that are created by the MPIAPHOT procedures. Use the MIDAS command show/descr to see a description of the descriptor content (in most cases):

<table>
<thead>
<tr>
<th>Name</th>
<th>type</th>
<th>n</th>
<th>content</th>
</tr>
</thead>
<tbody>
<tr>
<td>NR</td>
<td>I</td>
<td>1</td>
<td>unique object number</td>
</tr>
<tr>
<td>IDENT</td>
<td>C</td>
<td>72</td>
<td>identifier (must be present)</td>
</tr>
<tr>
<td>DATE</td>
<td>C</td>
<td>8</td>
<td>date file is created (or checked if not present)</td>
</tr>
<tr>
<td>ROT_POS</td>
<td>I</td>
<td>1</td>
<td>position angle of rotator (from CCD-program)</td>
</tr>
<tr>
<td>DETECTOR</td>
<td>C</td>
<td>8</td>
<td>detector identification</td>
</tr>
<tr>
<td>CCD_BINN</td>
<td>I</td>
<td>2</td>
<td>binning of CCD in X and Y</td>
</tr>
</tbody>
</table>
| CCD_PARA      | R    | 5  | EPC, RON, DC_OFF, upper end of dynamic range (~original saturation value),
|               |      |    | global saturation [cts] or [e-] depending on reduction stage.           |
| CCD_REFE      | C    | 32 | source of information in CCD_PARA                                     |
| PIX_SIZE      | R    | 2  | pixel size in X and Y [micron]                                         |
| BAD_COL       | I    | *  | seq.numbers of really bad columns (< 6) (absolute column numbers !)    |
| FR_AREA       | D    | 4  | X_s,Y_s,X_e,Y_e [world] of statistic window                            |
| INSTR_ID      | C    | 8  | ID for instrument used                                                 |
| TELES_ID      | C    | 8  | ID for telescope used                                                  |
| OBS_SITE      | R    | 3  | observatory long., lat. elev. [deg.,m]                                 |
| TEL_DIAM      | R    | 1  | effective telescope diameter [m]                                       |
| INSTR_SC      | R    | 2  | scale provided by instrument in X and Y                               |
| UNITS_SC      | C    | 32 | units used in INSTR_SC [arcsec/mm]                                    |
| CUNIT         | C    | 72 | set to counts pixels pixels                                            |
| HISTORY       | C    | *  | filled to 80 characters per line                                       |
| DATE-OBS      | C    | 8  | civil date at exposure [dd/mm/yy]                                      |
| TM_START      | R    | 1  | start of exposure [UT sec. since midnight]                             |
| TM_END        | R    | 1  | end of exposure [UT sec. since midnight]                               |
| EXP_TIME      | R    | 1  | exposure time [sec]                                                    |
| POSTN_RA      | D    | 1  | RA [deg]                                                               |

... continued on next page
<table>
<thead>
<tr>
<th>Name</th>
<th>type</th>
<th>n</th>
<th>content</th>
</tr>
</thead>
<tbody>
<tr>
<td>POSTN DE</td>
<td>D</td>
<td>1</td>
<td>DEC [deg]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>POSTN DE (real), O_POS(2) are recognized</td>
</tr>
<tr>
<td>SUN</td>
<td>R</td>
<td>(2)</td>
<td>elevation and azimuth of sun [deg], if elevation &gt; −18°</td>
</tr>
<tr>
<td>MOON</td>
<td>R</td>
<td>(4)</td>
<td>elevation, azimuth, angular distance and phase of moon if it is above horizon</td>
</tr>
<tr>
<td>FLAT BKG</td>
<td>R</td>
<td>1</td>
<td>contribution to background level that is not giving rise to fringes, i.e. pure continuum radiation as e.g. moonlight</td>
</tr>
<tr>
<td>JUL_DATE</td>
<td>D</td>
<td>1</td>
<td>Julian date (including time information) if JUL_DATE (char.) is present, it is deleted.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(1) year</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(2) month</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(3) fully elapsed days since beginning of year</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(4) modified JD (JUL_DATE - 2.400.000.5)</td>
</tr>
<tr>
<td>O TIME</td>
<td>D</td>
<td>8</td>
<td>(5) TM_START</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(6) SID_TIME</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(7) EXP_TIME</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(8) offset of UT-clock [real hours]</td>
</tr>
<tr>
<td>AIRMASS</td>
<td>R</td>
<td>1</td>
<td>airmass at mid exposure</td>
</tr>
<tr>
<td>FILTER</td>
<td>C</td>
<td>32</td>
<td>description of filter used</td>
</tr>
<tr>
<td>LAMBDA</td>
<td>R</td>
<td>4</td>
<td>central wavelength, FWHM of filter, lower, upper wavelength limit for transmission curve [nm], defining the integration boundaries!</td>
</tr>
<tr>
<td>EPSILIST</td>
<td>C</td>
<td>80</td>
<td>list of table names specifying the instrumental response</td>
</tr>
<tr>
<td>ANALYSER</td>
<td>C</td>
<td>32</td>
<td>description of analyser used (CHECK/POLAR only)</td>
</tr>
<tr>
<td>PA ANALY</td>
<td>R</td>
<td>1</td>
<td>position angle of analyser (CHECK/POLAR only)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ROT_POS (integer) used as default in prompt</td>
</tr>
<tr>
<td>PA INSTR</td>
<td>R</td>
<td>1</td>
<td>position angle of instrument (degrees, relative to zero point, where PA CASS=0. Counted from N over E)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(is preset to value given in INSTRUMENT.PAR)</td>
</tr>
<tr>
<td>PA CASS</td>
<td>R</td>
<td>1</td>
<td>position angle of Cassegrain flange [deg]</td>
</tr>
<tr>
<td>PA ROTA</td>
<td>R</td>
<td>1</td>
<td>same function as PA CASS</td>
</tr>
<tr>
<td>PA IMAGE</td>
<td>R</td>
<td>1</td>
<td>position angle of the image according to (PA INSTR + PA CASS). If PA CASS or PA ROTA are not present angle is assumed to be zero.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>This is the position angle of the +Y-axis of the detector (after optional flip), counted from North over East</td>
</tr>
<tr>
<td>OLD COOR</td>
<td>D</td>
<td>3</td>
<td>Holds original START before CORRECT, 3. element gives angle by which images was rotated in CORRECT</td>
</tr>
<tr>
<td>TEL OFFS</td>
<td>R</td>
<td>2</td>
<td>telescope offsets in RA and DEC [XY system in arcsec]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>User can also enter offset in RA/DEC-system</td>
</tr>
<tr>
<td>XY RADEC</td>
<td>D</td>
<td>6</td>
<td>astrometric reference point and astrometric error [X, Y, RA, DEC, error X, error Y]</td>
</tr>
<tr>
<td>OA PIX</td>
<td>D</td>
<td>2</td>
<td>position of optical axis [pixels!] with respect to lower left corner of image</td>
</tr>
<tr>
<td>OA PIXII</td>
<td>D</td>
<td>2×n</td>
<td>same for individual detectors in case of multi-detector images like e.g. WFI.</td>
</tr>
<tr>
<td>CHIP EDGESII</td>
<td>I</td>
<td>4×n</td>
<td>chip edges of individual detectors for multi-detector images</td>
</tr>
<tr>
<td>N DETECT</td>
<td>I</td>
<td>1</td>
<td>number of detectors for multi-detector image</td>
</tr>
<tr>
<td>DISTORT</td>
<td>R</td>
<td>7</td>
<td>distortion parameters (polynomial expansion of up to 6th degree), specifying the radially symmetric optical distortion of the instrument.</td>
</tr>
</tbody>
</table>

... continued on next page
<table>
<thead>
<tr>
<th>Name</th>
<th>type</th>
<th>n</th>
<th>content</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA_LIMIT</td>
<td>R</td>
<td>1</td>
<td>Optional: Pixels with values below this limit are ignored.</td>
</tr>
<tr>
<td>MOSAIC</td>
<td>I</td>
<td>1</td>
<td>set to &gt; 0 if image is a mosaic (needed for CLASS/EVAL use 1 for HIROCS and 2 for MANOS-deep)</td>
</tr>
<tr>
<td>SURVEY</td>
<td>C</td>
<td>8</td>
<td>Identifies from which sky survey the table is coming from (SDSS, 2MASS), used in flux/survey.</td>
</tr>
<tr>
<td>BPM_ROOT</td>
<td>C</td>
<td>4</td>
<td>Root of original file in case a bad-pixel-mask is used in mosaic/rmosics</td>
</tr>
<tr>
<td>BPM_ID</td>
<td>C</td>
<td>80</td>
<td>identifier for bad-pixel-mask</td>
</tr>
<tr>
<td>M_CLEAN</td>
<td>I</td>
<td>1</td>
<td>number of bright stars to be cleaned in their surrounding</td>
</tr>
<tr>
<td>NR_CLEAN</td>
<td>I</td>
<td>m_clean</td>
<td>object number of these stars</td>
</tr>
<tr>
<td>RAD_CLEAN</td>
<td>R</td>
<td>m_clean</td>
<td>cleaning radius for each star</td>
</tr>
<tr>
<td>M_LARGE</td>
<td>I</td>
<td>1</td>
<td>number of excessively large objects to be flagged</td>
</tr>
<tr>
<td>NR_LARGE</td>
<td>I</td>
<td>m_large</td>
<td>object numbers</td>
</tr>
<tr>
<td>BRIGHT_PSF</td>
<td>R</td>
<td>1</td>
<td>upper magnitude limit of bright stars to be set to the PSF (skip if ≤ 0)</td>
</tr>
<tr>
<td>STELLAR_PSF</td>
<td>R</td>
<td>1</td>
<td>lower limit of stellarity index (:DX_LINE).</td>
</tr>
<tr>
<td>BRIGHT_SPUR</td>
<td>R</td>
<td>1</td>
<td>upper magnitude limit for spurious objects (spikes etc.)</td>
</tr>
<tr>
<td>STELLAR_SPUR</td>
<td>R</td>
<td>1</td>
<td>lower limit of stellarity index for spurious objects.</td>
</tr>
<tr>
<td>RM_PATCHES</td>
<td>I</td>
<td>1</td>
<td>number of rectangular patches within which objects will be flagged with W by CLEAN/SEX</td>
</tr>
<tr>
<td>RM_CORNERS</td>
<td>R</td>
<td>4×RM_PATCHES</td>
<td>edges of the patches (world coordinates)</td>
</tr>
</tbody>
</table>

Descriptors created or modified by MPIAPHOT procedures

<table>
<thead>
<tr>
<th>Name</th>
<th>type</th>
<th>n</th>
<th>content</th>
</tr>
</thead>
<tbody>
<tr>
<td>FR_STAT</td>
<td>R</td>
<td>6</td>
<td>statistics of background in window FR_AREA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1  background level (median)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2  rms-scatter</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3  rms-scatter after subtraction of smooth model</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4  lower bound</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5  upper bound</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6  count rate in background</td>
</tr>
<tr>
<td>FF_NORM</td>
<td>R</td>
<td>7</td>
<td>statistics for flat field normalization</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1  median level in FR_AREA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2  EPC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3  RON [counts]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4  window X_lower</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5  window Y_lower</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6  window X_upper</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>7  window Y_upper</td>
</tr>
<tr>
<td>PSF_MEAN</td>
<td>R</td>
<td>4</td>
<td>aperture radius [pixels], e-folding width [pixels] in X, Y, position angle of major axis with respect to the +X-axis (!)</td>
</tr>
<tr>
<td>PSF_FIND</td>
<td>R</td>
<td>4</td>
<td>same as PSF_MEAN but from FIND/OBJECT</td>
</tr>
<tr>
<td>PSF_FWHM</td>
<td>R</td>
<td>1</td>
<td>width in arcsec</td>
</tr>
<tr>
<td>EPSILIST</td>
<td>C</td>
<td>80</td>
<td>string with names of optical elements/detectors, separated by commata</td>
</tr>
<tr>
<td>DARK_0</td>
<td>R</td>
<td>2</td>
<td>bias and rms from overscan</td>
</tr>
<tr>
<td>N_OBJ</td>
<td>I</td>
<td>1</td>
<td>number of objects found in find/obj</td>
</tr>
<tr>
<td>REF_TAB</td>
<td>C</td>
<td>80</td>
<td>reference table for merging of object list</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>table to which XY_move is referenced</td>
</tr>
<tr>
<td>REF_FRAME</td>
<td>C</td>
<td>80</td>
<td>frame to which XY_move is referenced</td>
</tr>
<tr>
<td>REF_POS</td>
<td>D</td>
<td>3</td>
<td>reference position RA, DEC, equinox</td>
</tr>
</tbody>
</table>

... continued on next page
### APPENDIX B. DESCRIPTORS

<table>
<thead>
<tr>
<th>Name</th>
<th>type</th>
<th>n</th>
<th>content</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABFORM</td>
<td>C</td>
<td></td>
<td>8 if present in a sky survey table signals to find/move that the reference is not an image with an associated MARK table but the survey table. Its content is the name of the filter to be used for magnitude information.</td>
</tr>
<tr>
<td>XY_MOVE</td>
<td>R</td>
<td></td>
<td>6 transformation matrix (shift, scale) determined by FIND/MOVE</td>
</tr>
<tr>
<td>XY_M1</td>
<td>R</td>
<td></td>
<td>6 transformation matrix (shift, scale) for 1. call</td>
</tr>
<tr>
<td>XY_M2</td>
<td>R</td>
<td></td>
<td>6 transformation matrix (shift, scale) for 2. call</td>
</tr>
<tr>
<td>BORDERS</td>
<td>R</td>
<td></td>
<td>4 lower left and upper right corner [gnomonic (!) world coordinates] encompassing the whole image (see master/section).</td>
</tr>
<tr>
<td>BORDERii</td>
<td>R</td>
<td>4*n</td>
<td>for each detector (in case of multiple detectors)</td>
</tr>
<tr>
<td>REL_SCALE</td>
<td>R</td>
<td></td>
<td>1 ratio of exposure level to that of reference frame (&gt; 1 if this image is exposed deeper)</td>
</tr>
<tr>
<td>LIN_SCALE</td>
<td>R</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>ANGLE</td>
<td>R</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>N_MATCH</td>
<td>I</td>
<td></td>
<td>1 number of matches found in find/move</td>
</tr>
<tr>
<td>RMS_ARCS</td>
<td>R</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>MOVE_PAR</td>
<td>R</td>
<td></td>
<td>3 included for backwards compatibility</td>
</tr>
<tr>
<td>XY_MOVEii</td>
<td>R</td>
<td>6</td>
<td>as above but for multiple detectors, ii = 1...8</td>
</tr>
<tr>
<td>REL_SCALEii</td>
<td>R</td>
<td>1</td>
<td>dito</td>
</tr>
<tr>
<td>LIN_SCALEii</td>
<td>R</td>
<td>1</td>
<td>dito</td>
</tr>
<tr>
<td>ANGLEii</td>
<td>R</td>
<td>1</td>
<td>dito</td>
</tr>
<tr>
<td>N_MATCHii</td>
<td>I</td>
<td>1</td>
<td>dito</td>
</tr>
<tr>
<td>RMS_ARCSii</td>
<td>R</td>
<td>1</td>
<td>dito</td>
</tr>
<tr>
<td>MOVE_PARii</td>
<td>R</td>
<td>3</td>
<td>dito</td>
</tr>
<tr>
<td>SUM_FRAMES</td>
<td>C*64</td>
<td>n</td>
<td>names of images which entered into sum frame</td>
</tr>
<tr>
<td>SUM_WGHT</td>
<td>R</td>
<td>n</td>
<td>weights for each frame entered into the sum frame</td>
</tr>
</tbody>
</table>

Descriptors created during flux calibration

#### flux/integ

<table>
<thead>
<tr>
<th>Name</th>
<th>type</th>
<th>n</th>
<th>content</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAMBDA_C</td>
<td>R</td>
<td>1</td>
<td>central wavelength of overall transmission [nm] (created in flux/unite)</td>
</tr>
<tr>
<td>FWHM</td>
<td>R</td>
<td>1</td>
<td>Full-width-at-half-maximum of overall transmission [nm]</td>
</tr>
<tr>
<td>LAM_LOW</td>
<td>R</td>
<td>1</td>
<td>50% point in the rising flank of the overall transmission [nm]</td>
</tr>
<tr>
<td>LAM_HIGH</td>
<td>R</td>
<td>1</td>
<td>50% point in the falling flank of the overall transmission [nm]</td>
</tr>
<tr>
<td>LAMBi</td>
<td>R</td>
<td>n</td>
<td>n data points specifying the wavelength vector of the overall transmission curve [nm]</td>
</tr>
<tr>
<td>EPSi</td>
<td>R</td>
<td>n</td>
<td>n data points for the corresponding epsilon vector [%]</td>
</tr>
<tr>
<td>CAL_ffff_IND_ii</td>
<td>R</td>
<td>n</td>
<td>Calibration parameter for filter ffff (expected CR / normalized and adjusted observed CR) for standard star ii for n blocks</td>
</tr>
<tr>
<td>CAL_ffff_INDS_ii</td>
<td>R</td>
<td>n</td>
<td>Error of CAL_F_IND_ii</td>
</tr>
<tr>
<td>CAL_ffff_AV</td>
<td>R</td>
<td>m</td>
<td>Calibration parameter averaged over all blocks for m standard stars</td>
</tr>
<tr>
<td>CAL_ffff_AV_S</td>
<td>R</td>
<td>m</td>
<td>Error of CAL_F_AV</td>
</tr>
</tbody>
</table>

... continued on next page
<table>
<thead>
<tr>
<th>Name</th>
<th>type</th>
<th>n</th>
<th>content</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALIBm_all</td>
<td>R</td>
<td>1</td>
<td>1 Calibration factor averaged over all standard stars. flux [photons/m²/s/nm] = CR [photons/s] × CALIBm_all × normalization factor × mosaic factor.</td>
</tr>
<tr>
<td>CALIBs_all</td>
<td>R</td>
<td>1</td>
<td>1 Error of calibration factor averaged over all standard stars</td>
</tr>
<tr>
<td>CALIB_F</td>
<td>R</td>
<td>n</td>
<td>n Calibration factor averaged over all standard stars for each block, transformed from first block calibration including normalization and mosaic adjustment</td>
</tr>
<tr>
<td>CALIB_S</td>
<td>R</td>
<td>n</td>
<td>n Error of calibration factor</td>
</tr>
<tr>
<td>INT_EPS</td>
<td>R</td>
<td>1</td>
<td>1 Integral over total epsilon</td>
</tr>
<tr>
<td>INT_EPSL</td>
<td>R</td>
<td>1</td>
<td>1 Integral over total epsilon × lambda</td>
</tr>
<tr>
<td>N_STDS</td>
<td>I</td>
<td>1</td>
<td>1 total number of standard stars used</td>
</tr>
</tbody>
</table>

**flux/mosaic**

<table>
<thead>
<tr>
<th>Name</th>
<th>type</th>
<th>n</th>
<th>content</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAG_ZERO</td>
<td>R</td>
<td>1</td>
<td>1 zero-point for magnitude difference mag(eval) - mag(survey)</td>
</tr>
<tr>
<td>MAGD_iii</td>
<td>R</td>
<td>n</td>
<td>n Magnitude offsets for n survey stars in normblock iii</td>
</tr>
<tr>
<td>MAGE_iii</td>
<td>R</td>
<td>n</td>
<td>n Error of magnitude offsets</td>
</tr>
<tr>
<td>MAG_OFF</td>
<td>R</td>
<td>m</td>
<td>m Average magnitude offset for m normalization blocks</td>
</tr>
<tr>
<td>MAG_OFF_S</td>
<td>R</td>
<td>m</td>
<td>m Error of average offsets</td>
</tr>
<tr>
<td>MAG_OFF_R</td>
<td>R</td>
<td>m</td>
<td>m Scatter of average offsets</td>
</tr>
<tr>
<td>MAG_OFF_A</td>
<td>R</td>
<td>m</td>
<td>m Average offset from all objects (no clipping)</td>
</tr>
<tr>
<td>MAG_OFF_N</td>
<td>I</td>
<td>m</td>
<td>m Number of objects used (after clipping)</td>
</tr>
<tr>
<td>MAG_OFF_I</td>
<td>I</td>
<td>m</td>
<td>m Total number of objects available</td>
</tr>
</tbody>
</table>

**flux/norm**

<table>
<thead>
<tr>
<th>Name</th>
<th>type</th>
<th>n</th>
<th>content</th>
</tr>
</thead>
<tbody>
<tr>
<td>NORM_FAC</td>
<td>R</td>
<td>n</td>
<td>n Normalization factor for n blocks (CR(norm) / CR(current))</td>
</tr>
<tr>
<td>NORM_ERR</td>
<td>R</td>
<td>n</td>
<td>n Error of normalization factors</td>
</tr>
<tr>
<td>NORM_NUM</td>
<td>I</td>
<td>n</td>
<td>n Number of objects used for calculation of normalization factor</td>
</tr>
<tr>
<td>NORM_BLOCKS</td>
<td>I</td>
<td>n</td>
<td>n Number of normalization block to which current block is normalized to</td>
</tr>
</tbody>
</table>

**flux/selnorm**

<table>
<thead>
<tr>
<th>Name</th>
<th>type</th>
<th>n</th>
<th>content</th>
</tr>
</thead>
<tbody>
<tr>
<td>NORM_STARS</td>
<td>I</td>
<td>1</td>
<td>1 Number of normalization stars selected</td>
</tr>
</tbody>
</table>

**flux/setnorm**

<table>
<thead>
<tr>
<th>Name</th>
<th>type</th>
<th>n</th>
<th>content</th>
</tr>
</thead>
<tbody>
<tr>
<td>NORM_IMA</td>
<td>I</td>
<td>1</td>
<td>1 flag of normalization image</td>
</tr>
</tbody>
</table>

**flux/unite**

<table>
<thead>
<tr>
<th>Name</th>
<th>type</th>
<th>n</th>
<th>content</th>
</tr>
</thead>
<tbody>
<tr>
<td>REF_TAB</td>
<td>C</td>
<td></td>
<td>Name of reference table used</td>
</tr>
<tr>
<td>FILTER</td>
<td>C</td>
<td></td>
<td>Filter designation</td>
</tr>
<tr>
<td>STORED_TAB</td>
<td>C</td>
<td>n</td>
<td>n Name of evaluate table stored in each block</td>
</tr>
<tr>
<td>PSF_X</td>
<td>R</td>
<td>n</td>
<td>n major axis of PSF for n blocks</td>
</tr>
<tr>
<td>PSF_Y</td>
<td>R</td>
<td>n</td>
<td>n minor axis of PSF</td>
</tr>
<tr>
<td>PSF_A</td>
<td>R</td>
<td>n</td>
<td>n position angle of PSF</td>
</tr>
</tbody>
</table>

... continued on next page
### APPENDIX B. DESCRIPTORS

<table>
<thead>
<tr>
<th>Name</th>
<th>type</th>
<th>n</th>
<th>content</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAM_MIN</td>
<td>R</td>
<td>n</td>
<td>lower wavelength bound for integral over epsilon</td>
</tr>
<tr>
<td>LAM_MAX</td>
<td>R</td>
<td>n</td>
<td>upper wavelength bound for integral over epsilon</td>
</tr>
<tr>
<td>N_ENTRY</td>
<td>I</td>
<td></td>
<td>Number of blocks stored in table</td>
</tr>
<tr>
<td>NO_NORM</td>
<td>I</td>
<td></td>
<td>Number of normalization blocks (first blocks following leading columns)</td>
</tr>
</tbody>
</table>

**flux/combine**

<table>
<thead>
<tr>
<th>VEGA_ZERO</th>
<th>R</th>
<th>n</th>
<th>Zero-points for calculation of Vega magnitudes from fluxes for each block:</th>
</tr>
</thead>
</table>

\[
m = -2.5 \log(VEGA\_ZERO/F) + 20
\]

\[
dm = 2.5/\ln10dF/F = 1.0857dF/F
\]
Appendix C

Keywords

\textit{MPIAPHOT} specific keywords. The following is a list of keywords, which are created via context mpiaphot. Their content is calculated or inferred from the parameter files \texttt{INSTRUMENT.PAR} and \texttt{DETECTOR.PAR}.

Table C.1: Keywords specific to \textit{MPIAPHOT}. The user need not be familiar in detail with the keywords listed in the second block.

<table>
<thead>
<tr>
<th>Name</th>
<th>type</th>
<th>n</th>
<th>content</th>
</tr>
</thead>
<tbody>
<tr>
<td>DETECTOR</td>
<td>C</td>
<td>8</td>
<td>detector identification</td>
</tr>
<tr>
<td>DET_INFO</td>
<td>C</td>
<td>16</td>
<td>name of detector info file (extension &quot;.dat&quot;), resides in $PM/../tbl</td>
</tr>
<tr>
<td>N_DETECT</td>
<td>I</td>
<td>1</td>
<td>number of detectors</td>
</tr>
<tr>
<td>IMA_SIZE</td>
<td>I</td>
<td>2</td>
<td>total pixel number in X and Y</td>
</tr>
<tr>
<td>CCD_BINN</td>
<td>I</td>
<td>2</td>
<td>CCD binning factor in X and Y</td>
</tr>
<tr>
<td>CCD_PARA</td>
<td>R</td>
<td>5</td>
<td>EPC, RON, DC_OFF, upper end of dynamic range ((\approx)original saturation value), global saturation([e^-/ct], [e-], [cts], [cts], [cts])</td>
</tr>
<tr>
<td>CCD_REFE</td>
<td>C</td>
<td>32</td>
<td>reference for CCD_PARA</td>
</tr>
<tr>
<td>PIX_SIZE</td>
<td>R</td>
<td>2</td>
<td>pixel size in X and Y [micron]</td>
</tr>
<tr>
<td>GOOD_PIX</td>
<td>I</td>
<td>4</td>
<td>window of usefull area of chip [world coordinates = pixels](X_s,Y_s,X_e,Y_e)</td>
</tr>
<tr>
<td>OVERSCAN</td>
<td>I</td>
<td>4</td>
<td>window of overscan area (world coordinates!) to be used for determination of DARK_0 (Bias) (X_s,Y_s,X_e,Y_e)</td>
</tr>
<tr>
<td>BAD_COL</td>
<td>I</td>
<td></td>
<td>seq. numbers of really bad columns (6)</td>
</tr>
<tr>
<td>FR_AREA</td>
<td>R</td>
<td>4</td>
<td>(X_s,Y_s,X_e,Y_e) [world] of statistic window</td>
</tr>
<tr>
<td>INSTR_ID</td>
<td>C</td>
<td>8</td>
<td>ID for instrument used</td>
</tr>
<tr>
<td>TELES_ID</td>
<td>C</td>
<td>8</td>
<td>ID for telescope used</td>
</tr>
<tr>
<td>OBS_SITE</td>
<td>R</td>
<td>3</td>
<td>observatory long., lat. elev. [deg.,deg,m]</td>
</tr>
<tr>
<td>TEL_DIAM</td>
<td>R</td>
<td>1</td>
<td>effective telescope diameter [m]</td>
</tr>
<tr>
<td>INSTR_SC</td>
<td>R</td>
<td>2</td>
<td>scale provided by instrument in X and Y</td>
</tr>
<tr>
<td>UNIT_SC</td>
<td>C</td>
<td>32</td>
<td>units used in INSTR_SC ([e.g.\text{ arcsec/mm}])</td>
</tr>
<tr>
<td>INSTR_AP</td>
<td>I</td>
<td>4</td>
<td>instrument aperture [pixels]: Circle: (X_c, Y_c, \text{Radius, } 0) Rectangle: (X_c, Y_c, \text{ half sidelength in } X, Y)</td>
</tr>
<tr>
<td>PA_INSTR</td>
<td>R</td>
<td>1</td>
<td>PA of instrument (degrees)</td>
</tr>
<tr>
<td>PA_INSTR</td>
<td>R</td>
<td>1</td>
<td>PA of instrument (degrees)</td>
</tr>
<tr>
<td>EPSILIST</td>
<td>C</td>
<td>80</td>
<td>list of table names specifying the instrumental response</td>
</tr>
<tr>
<td>DATA_LIMIT</td>
<td>R</td>
<td>1</td>
<td>Pixels with values at or below this value are ignored by some of \textit{MPIAPHOT}’s routines (e.g.\text{ evaluate, find/obj etc..}). Default: -1000.</td>
</tr>
</tbody>
</table>

... continued on next page
<table>
<thead>
<tr>
<th>Name</th>
<th>type</th>
<th>n</th>
<th>content</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ERR_CTRL</strong></td>
<td>I</td>
<td>3</td>
<td>error control in PRG-files</td>
</tr>
<tr>
<td><strong>BG_FLAG</strong></td>
<td>C</td>
<td>10</td>
<td>fg: perform command in fg process</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>bg: store command in bg procedure</td>
</tr>
<tr>
<td><strong>CTRL_KEY</strong></td>
<td>C</td>
<td>8</td>
<td>used for program control</td>
</tr>
<tr>
<td><strong>DEL_KEY</strong></td>
<td>C</td>
<td>9</td>
<td>controls deleting of INPUT frames</td>
</tr>
<tr>
<td><strong>OUT_CAT</strong></td>
<td>C</td>
<td>60</td>
<td>default catalogue for output FRAMES</td>
</tr>
<tr>
<td><strong>OUT_CATT</strong></td>
<td>C</td>
<td>60</td>
<td>default catalogue for output TABLES</td>
</tr>
<tr>
<td><strong>NO_NAMES</strong></td>
<td>I</td>
<td>1</td>
<td>number of entries in NAMES.LIST</td>
</tr>
<tr>
<td><strong>I_PART</strong></td>
<td>I</td>
<td>1</td>
<td>pointer to retrieve name from NAMES.LIST</td>
</tr>
<tr>
<td><strong>PART_NN</strong></td>
<td>C</td>
<td>64</td>
<td>holds name retrieved from NAMES.LIST</td>
</tr>
<tr>
<td><strong>TOL_WORLD</strong></td>
<td>D</td>
<td>1</td>
<td>tolerance in arcsec, if world coordinates are to be compared for equality</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(default = 0.01)</td>
</tr>
<tr>
<td><strong>MAX_FRAMES</strong></td>
<td>I</td>
<td>2</td>
<td>Maximum number of frames handled in frames lists by application programs</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>and by listname (PRG level)</td>
</tr>
<tr>
<td><strong>N_DETECT</strong></td>
<td>I</td>
<td>1</td>
<td>number of detectors making up the image</td>
</tr>
<tr>
<td><strong>CHIP_EDGES</strong></td>
<td>I</td>
<td>4*n_detect</td>
<td>edges of the detectors in the mosaic</td>
</tr>
<tr>
<td><strong>OA_PIX_det</strong></td>
<td>D</td>
<td>2*n_detect</td>
<td>position of the optical axis in the pixel coordinate system of each detector</td>
</tr>
</tbody>
</table>
Appendix D

Astrometric transformation

The astrometric transformation to a reference frame or a reference table is contained in descriptor \texttt{XY\_MOVE}.

\texttt{XY\_MOVE} has 6 elements and is associated with each image. It specifies the transformation matrix to calculate coordinates \(x', y'\) in this image from the coordinates \(x, y\) in the reference image:

\[
\begin{align*}
x' &= \texttt{XY}(1) + \texttt{XY}(2) \cdot x + \texttt{XY}(3) \cdot y \\
y' &= \texttt{XY}(4) + \texttt{XY}(5) \cdot x + \texttt{XY}(6) \cdot y
\end{align*}
\]

The trivial matrix is \(0, 1, 0, 0, 0, 1\).

The inverse transformation is given by

\[
\begin{align*}
x &= \texttt{XY}'(1) + \texttt{XY}'(2) \cdot x' + \texttt{XY}'(3) \cdot y' \\
y &= \texttt{XY}'(4) + \texttt{XY}'(5) \cdot x' + \texttt{XY}'(6) \cdot y'
\end{align*}
\]

with the following definition

\[
\begin{align*}
det &= (\texttt{XY}(2) \cdot \texttt{XY}(6) - \texttt{XY}(3) \cdot \texttt{XY}(5)) \\
\texttt{XY}'(1) &= (\texttt{XY}(3) \cdot \texttt{XY}(4) - \texttt{XY}(1) \cdot \texttt{XY}(6))/det \\
\texttt{XY}'(2) &= \texttt{XY}(6)/detc \\
\texttt{XY}'(3) &= -\texttt{XY}(3)/det \\
\texttt{XY}'(4) &= (\texttt{XY}(1) \cdot \texttt{XY}(5) - \texttt{XY}(2) \cdot \texttt{XY}(4))/det \\
\texttt{XY}'(5) &= -\texttt{XY}(5)/det \\
\texttt{XY}'(6) &= \texttt{XY}(2)/det
\end{align*}
\]

Average enlargement with respect to reference image is

\[
\text{enlf} = \sqrt{\texttt{XY}'(2) \cdot \texttt{XY}'(2) + \texttt{XY}'(3) \cdot \texttt{XY}'(3)}
\]

Rotation against reference image

\[
\text{rotf} = 57.296 \cdot \arctan(\texttt{XY}'(3), \texttt{XY}'(2))
\]

Shear [\%] with respect to reference image

\[
\text{shearf} = (1 - \sqrt{\texttt{XY}'(2) \cdot \texttt{XY}'(2) + \texttt{XY}'(3) \cdot \texttt{XY}'(3)})/c(\texttt{XY}'(5) \cdot \texttt{XY}'(5) + \texttt{XY}'(6) \cdot \texttt{XY}'(6))) \cdot 100
\]
Appendix E

Photometric data tables

Photometric data (standard star fluxes, filter curves, detector efficiencies etc.) are stored under

```
$PM/../tbl/PHOTDATA
$PM/../tbl/PHOTSTDS.
```

Sky survey data are stored under $PM/../tbl/2MASS and $PM/../tbl/SDSS.

Currently the user has to copy the sky survey tables into his current directory. All other tables are
first searched for in the local directory and then the environment variables PHOTDATA and PHOTSTDS are
used to locate the tables in the photometry area. A message is issued if tables outside the current user
directory is used.

E.1 Standard star tables

There are 5 different modi for the format of the standard star table. For FLUX/INTEG only mode 5 is
implemented. Each input table has to hold a descriptor MODE which specifies the mode to be used with
this table.

Please note the different units for mode = 4 and mode = 5 !! This is especially important if a table
constructed for mode = 4 is to be used with mode = 5. Sorry for this, which has its roots in the historical
development.

1 Spectrum is defined via broadband magnitudes.
   Table (UBVRISTD) columns:
   FIELD: name of field
   OBJECT: name of object
   LOWER, U, B, V, R, I, UPPER: magnitudes at wavelengths given by descriptor LAMB_STD [nm]
   with conversion factors to microJY as given by descriptor CONV_FLX. Flux at a lower and upper
   boundary is also taken from that table in order to cover optical range completely.
   INCLUDE : flag to suppress object in calculating the mean for the conversion factor per field in
   PHOTO/SYNTH
   NULL in FIELD indicates end of data!

2 Spectrum is defined by wavelength and AB magnitude
   Table (SPSSDATA) columns:
   LAMB_i:ii: Wavelength [nm]
   _name_: Label is name of standard star as taken from IDENT
   (first 8 none-blanks) or from input OBJECT_IN. In this case FIELD_IN is ignored!
   Entries are AB magnitudes : \( AB = -2.5 \log (f_n[\text{microJy}]) + 23.90 \)
   Searched is done via label "name". From there backwards a column is search with label starting with
   LAMB_..., which is taken as the corresponding wavelength column. This way a block of standard
   stars can be entered with only one wavelength column. NULL in either column terminates data
   input!

3 Spectrum is taken from a spectrophotometric atlas via cross reference.
   Columns of table (NORMFLUX):
FIELD: name of field
OBJECT: name of object
SP_REF: cross-reference to atlas spectrum.
Either atlas sequence number in the format #iii (e.g. #012), or name of the standard star (C*8).
LAMBDA: normalization wavelength [nm]
F_NUE: flux [in AB magnitudes] at that wavelength
INCLUDE: 1 = include, 0 = exclude for photometry (PHOT/SYNTH)
Logical NULL values are skipped and do NOT terminate input.
However, if LAMBDA or F_NUE are logical NULL, no normalization is done, but the values as given are taken [microJy]. Use of Gunn-Striker spectrophotometric atlas is currently implemented. The name of the atlas tables is held in descriptor SP_ATLAS of table.
(NORMFLUX): e.g. SP_ATLAS/C/1/2 GS will use the tables GSATLAS (lambda [nm], relative f_nues) and GSINFO (column 1: OBJECT [name]). The latter is only used, if object identification is done via name and not via sequence number. Information upon entry numbers is stored in descriptor N_ATLAS/I/1/2 N_ATLAS(1) = number of objects N_ATLAS(2) = number of points per spectrum
Search table (NORMFLUX) for entry as specified in FIELD_IN and OBJECT_IN. Sequence number is used to get normalized F_nue from atlas table. The spectrum from that catalogue will be re-normalized as defined by the specified wavelength and flux.

4 Take data directly from spectrophotometric atlas as specified by TABLE.
FIELD_IN = irrelevant,
OBJECT_IN = object number (I3)
MAX contains the number of wavelength entries in TABLE.
Logical NULL values are skipped and do NOT terminate input. The first column of TABLE should contain the wavelength in nanometer, successive columns flux f_nue in µJy for each object.

5 Individual table for each object
FIELD_IN = irrelevant,
OBJECT_IN = object name
Table name with flux is found via cross-reference from table given in call. Format of cross-reference table:
object: name of object (< 17 char., case-insensitive as it will be converted to upper case, blanks ignored)
table: full path and name (upper case only) of table to be used for object
(max. 64 characters).
Descriptor MODE must be present and set to 5.
Format for flux tables:
lambda wavelength [Angstrom]
f_nue flux [mJy]

E.2 Epsilon tables
These tables specify the transmission / detection properties of the optical elements of the instruments. For each element a table has to be supplied with the following properties:
Tables contain two columns each with wavelength in the 1. column [units nm] and epsilon in the 2. column [units %]. The data are terminated when END OF COLUMN is encountered, i.e. NULL values are skipped. Column labels should be :wavelength and :DQE.
The name of the table specifies the optical element. It is first searched in the current working directory. If it is not found, the environment variable PHOTDATA is consulted. If it exists it should point to a directory, where the table is searched for.

Examples: mirror, LAICA_R, FPA77
Appendix F

Internals

F.1  Kappa-sigma clipping

When averages are to be calculated the subroutine averclip.b is used. This routine has the following input/output variables:

Input
X  vector with values
S  vector with sigmas (enter -1, if no sigma available)
N  number of data points provided
MODE  mode of error determination
       STD/WEIGHT or MIN/NO_WEIGHT and combinations
       STD = standard error determination
       sigma = scatter/sqrt(n), in case of no weighting
       = 1/sqrt(sum(1/sigma**2)) in case of weighting
       MIN = do not use sigma, but calculate scatter around mean.
       Find entry, whose removal produces smallest scatter
       (including weighting by 1/sigma**2).
       Use this reduced scatter as sigma for clipping
       WEIGHT = do weighting of data ~ 1 / sigma**2
       NO_WEIGHT = no weighting even when sigmas are provided
KAPPA  kappa for kappa-sigma clipping (=0 for no clipping)
LOOP  number of iterations for kappa-sigma-clipping
SIG_MIN  lower bound for sigma of the mean
       (if < 0, stop clipping if relative change in average is < SIG_MIN)
MIN_NUM  minimum number of data points to be retained

Output
CLIPPED  vector with clipping flags (=1 clipped, 0=retained)
MEAN  mean value
MEANALL  mean value of total sample (without clipping)
SIG_1  un-clipped sigma of the mean
SIG_2  clipped sigma of the mean
SIG_3  un-clipped corrected sigma
       = sqrt( sqrt[((CHI^2)/<CHI^2>_n)**2-CHI^2/<CHI^2>_n+1)*SIG_1
SIG_4  clipped corrected sigma
       = sqrt( sqrt[((CHI^2)/<CHI^2>_n)**2-CHI^2/<CHI^2>_n+1)*SIG_2
CHI2  CHI^2 calculated from the data
CHI2C  CHI^2 calculated from the clipped data
SCATTER  scattering of values
       = sqrt(n/(n-1)*(sum(x^2)/sum(1/s^2)-mean*mean))
N_USED  used number of data points
STATUS  exit status
    Error codes : STATUS = 0  okay
    1  clipping reduces N to less than MIN_NUM
    2  N < MIN_NUM upon entry
    3  mean_old = 0, sorry
    4  not enough iterations

In most routines default values are provided in the code for MODE, KAPPA, LOOP, SIG_MIN and MIN_NUM. These default values may be overwritten by the user if he provides a MIDAS keyword KS_MODE (character) and/or KS_CLIP (4 real values specifying KAPPA, LOOP, SIG_MIN, MIN_NUM. Please remember to delete this keyword once it is no longer needed. Otherwise future averages in the same MIDAS session will be affected as well!

Routines supporting this feature:

- flux/selnorm
- flux/norm
- flux/unite
- flux/mosaic
- flux/apcorr
Appendix G

Cosmic ray removal on single frames

If dithered images are available cosmic ray events are most conveniently removed with command `cosmic/median` (see page 50). However, if only single frames are available, the median image cannot be constructed and the cosmics have to be detected within a single frame. This is accomplished by analysing the gradient from each pixel to its neighbours and comparing this gradient with what is allowed by the photon statistics. Manfred Stickel has provided a set of MIDAS commands, which allow to derive a likelihood map for a pixel being a cosmic ray event. With a convenient threshold a cosmic mask can be derived from this likelihood map.

This package is activated by `set/cont badpixel`.

For this to work, the frames have to have a proper statistics descriptor `FR_STAT` (see `STATISTICS/BACKGROUND`). Descriptor `CCD_PARA` must also be present.
BADPIX/LIKE root for OUT_names = IN_list  threshold

Purpose:
Creates a likelihood map for each input image. This specifies the likelihood for each pixel being a cosmic ray event.

Parameters:

- $p_1$ root of out_names first for characters for name(s) of output images
- $p_2$ =
- $p_3$ IN_list: list of input frames (see page 195)
- $p_4$ threshold The threshold value beyond which a pixel is regarded as a cosmic ray event. In this call it is only used to calculate the potential number of cosmic ray affected pixels.

Examples:
badpix/like like = inlist:

Actions taken:
1.

Related commands:
badpix/cmask  badpix/remove
Index

MPIAPHOT
  environment variable PM, 3
  DETECTOR_new.PAR, 4
  INSTRUMENT_new.PAR, 4

ANALYSE
  FILL, 124
  FIT, 126
  STORE, 128

ARTIFICIAL
  IMAGE, 35
  astrometry, 160

badpixel
  like, 214
  BIAS/EXTRAPOLATE, 38

CHECK
  DESCRIPTOR, 40
  CHKLRG/SEXTRACTOR, 83
  CLASS/COMBINE, 93
  CLASS/EVAL, 91
  CLEAN/SEXTRACTOR, 84
  context
    astrometry, 159
    classy, 131
    flux_cal, 100
    main, 32
    synphot, 156

CORRECT
  FLIP, 46
  IMAGE, 46

COSMIC
  MEDIAN, 50
  REFER, 51

cosmic ray removal
  single frames, 213

CREATE/SEXTRACT, 82

DARK
  AVER, 44
  data limit, 14
  descriptors
    FLAT_BKG, 200
    XY_MOVE, 207

  epsilon tables, 210
  error treatment, 197
  EVALUATE, 96
    MC, 99
    evaluate
      set, 94

  file names, 9
  FIND/MOVE, 69
  FIND/OBJECT, 66
  find/ring, 47
  FIND/SEXTRACT, 80
  FLAT
    AVER, 44
    BACKGROUND, 73

FLUX
  ANALYSE, 116
  APCORR, 130
  AVERAGE, 126
  BLOCK, 122
  CLEANUP, 121
  COMBINE, 117
  CORRECT, 129
  INTEG, 115
  MOSAIC, 114
  NORM, 111
  PRINTCAL, 123
  REFER, 101
  SELNORM, 103
  SETNORM, 105
  STD, 102
  STORED, 122
  SURVEY, 112
  UNITE, 107
  VERIFY, 106

frame list, 195
frame table, 52
  b2f, 55
  badflag, 57
  catalogue, 54
  copyflags, 58
  cpdt, 61
  cpkt, 62
  create, 52
  f2b, 55
  fill, 53
  flag, 56

  215
rename, 63
reset, 59
short, 60
statistics, 64

INPUT
FITS, 33
installation, 3
copyphot, 3

kappa-sigma clipping, 211

LBC
ingest, 188
prep, 189
LBC reduction, 187

MARK/DISP, 37
MASTER
IMAGE, 88
SECTION, 90
SET, 87
TABLE, 89

MOSAIC
COPYCOS, 78
GNOMIMA, 74
MEDIAN, 75
RM COSMICS, 76
SUMUP, 79
movie, 39

NAME
HIROCS, 34

OMEGA2k reduction, 185

PATCH/SEXTRACTOR, 86
PHOT_integ, 157
Photometric tables, 209

PREP
CCD, 41
Profile
Gauss, 36
remove/ring, 48
saturation, 14
set/refpos, 65
standard star tables, 209
STAT/BAC, 43
subtract/ring, 49
sum
ima, 72
synthetic photometry
PHOT_integ, 157

UNDISTORT/TABLE, 68