example7

May 7, 2018

1 Using the calibration database

The calibration database is an important part of the DRS.

1.1 Loading the calibration database (with unique entries for each key)

By default the calibration database should be opened in the setup procedure of the recipe.

For this to happen it requires an acquisition time (to make a choice between the calibration database files).

Therefore during setup a file must be marked as the main fits file.

This is done by default is using the standard recipe options (i.e. RECIPE NIGHT_NAME FILES) - the first file is used as the main fits file (and therefore defines "fitsfilename")

For custom arguments one has to set this file name manually (using the "mainfitsfile" arguement in "LoadArguments" where "mainfitsfile" must be one of the names (see example 4). Also recall that if one uses custom arguments one must manually load he calibration database (spirouStartup.LoadCalibDB(p)).

Below we use cal_loc_RAW_spirou.py as an example

```
In [1]: # notebook only setup run time arguments via sys.argv
        import sys
        sys.argv = ('cal_loc_RAW_spirou.py 20170710 flat_dark02f10.fits flat_dark03f10.fits flat
                    'flat_dark05f10.fits flat_dark06f10.fits'.split(' '))

In [2]: # imports and setup
        from __future__ import division
        import numpy as np
        import os

        from SpirouDRS import spirouBACK
        from SpirouDRS import spirouCDB
        from SpirouDRS import spirouConfig
        from SpirouDRS import spirouCore
        from SpirouDRS import spirouImage
        from SpirouDRS import spirouLOCOR
        from SpirouDRS import spirouStartup

        # =====================================================================
```

```python
# Define variables
# ==============================================================================
# Get Logging function
WLOG = spirouCore.wlog
# Get plotting functions
sPlt = spirouCore.sPlt
# Name of program
__NAME__ = 'cal_loc_RAW_spirou.py'
# Get version and author
__version__ = spirouConfig.Constants.VERSION()
__author__ = spirouConfig.Constants.AUTHORS()
__date__ = spirouConfig.Constants.LATEST_EDIT()
__release__ = spirouConfig.Constants.RELEASE()
# Custom parameter dictionary
ParamDict = spirouConfig.ParamDict


# calls to main function (missing here) and set to None to use run time arguments
night_name = None
files = None


# ------------------------------------------------------------------------
# Set up
# ------------------------------------------------------------------------
# get parameters from config files/run time args/load paths + calibdb
p = spirouStartup.Begin()
p = spirouStartup.LoadArguments(p, night_name, files)
# run specific start up
params2add = dict()
params2add['dark_flat'] = spirouLOCOR.FiberParams(p, 'C')
params2add['flat_dark'] = spirouLOCOR.FiberParams(p, 'AB')
p = spirouStartup.InitialFileSetup(p, kind='localisation',
                                   prefixes=['dark_flat', 'flat_dark'],
                                   add_to_p=params2add, calibdb=True)
```

```
15:38:55.0 -   || ****************************************
15:38:55.0 -   || * SPIROU @(#) Geneva Observatory (0.1.032)
15:38:55.0 -   || ****************************************
15:38:55.0 -   ||(dir_data_raw)     DRS_DATA_RAW=/scratch/Projects/spirou_py3/data/raw
15:38:55.0 -   ||(dir_data_reduc)   DRS_DATA_REDUC=/scratch/Projects/spirou_py3/data/reduced
15:38:55.0 -   ||(dir_calib_db)     DRS_CALIB_DB=/scratch/Projects/spirou_py3/data/calibDB
15:38:55.0 -   ||(dir_data_msg)     DRS_DATA_MSG=/scratch/Projects/spirou_py3/data/msg
15:38:55.0 -   ||(print_level)      PRINT_LEVEL=all      %(error/warning/info/all)
15:38:55.0 -   ||(log_level)        LOG_LEVEL=all        %(error/warning/info/all)
15:38:55.0 -   ||(plot_graph)       DRS_PLOT=1           %(def/undef/trigger)
15:38:55.0 -   ||(used_date)        DRS_USED_DATE=undefined
15:38:55.0 -   ||(working_dir)      DRS_DATA_WORKING=/scratch/Projects/spirou_py3/data/tmp
15:38:55.0 -   ||                   DRS_INTERACTIVE is not set, running on-line mode
15:38:55.0 -   |cal_loc_RAW_spirou:02f10+[...]|Now running : cal_loc_RAW_spirou on file(s): flat
```

```
15:38:55.0 -    |cal_loc_RAW_spirou:02f10+[...]|On directory /scratch/Projects/spirou_py3/data/ra
15:38:55.0 -    |cal_loc_RAW_spirou:02f10+[...]|ICDP_NAME loaded from: /scratch/Projects/spirou_p
15:38:55.0 - * |cal_loc_RAW_spirou:02f10+[...]|Correct type of image for localisation (dark_flat
15:38:55.0 -    |cal_loc_RAW_spirou:02f10+[...]|Calibration file: 20170710_flat_flat02f10_badpixe
15:38:55.0 -    |cal_loc_RAW_spirou:02f10+[...]|Calibration file: 20170710_flat_dark02f10_blaze_A
15:38:55.0 -    |cal_loc_RAW_spirou:02f10+[...]|Calibration file: 20170710_dark_flat02f10_blaze_C
15:38:55.0 -    |cal_loc_RAW_spirou:02f10+[...]|Calibration file: 20170710_dark_dark02d406.fits a
15:38:55.0 -    |cal_loc_RAW_spirou:02f10+[...]|Calibration file: 20170710_flat_dark02f10_flat_AB
15:38:55.0 -    |cal_loc_RAW_spirou:02f10+[...]|Calibration file: 20170710_dark_flat02f10_flat_C.
15:38:55.0 -    |cal_loc_RAW_spirou:02f10+[...]|Calibration file: 20170710_flat_dark02f10_loco_AB
15:38:55.0 -    |cal_loc_RAW_spirou:02f10+[...]|Calibration file: 20170710_dark_flat02f10_loco_C.
15:38:55.0 -    |cal_loc_RAW_spirou:02f10+[...]|Calibration file: 20170710_flat_dark02f10_order_p
15:38:55.0 -    |cal_loc_RAW_spirou:02f10+[...]|Calibration file: 20170710_dark_flat02f10_order_p
15:38:55.0 -    |cal_loc_RAW_spirou:02f10+[...]|Calibration file: 20170710_fp_fp02a203_tilt.fits
15:38:55.0 -    |cal_loc_RAW_spirou:02f10+[...]|Calibration file: spirou_wave_ini3.fits already e
15:38:55.0 -    |cal_loc_RAW_spirou:02f10+[...]|Calibration file: 2017-10-11_21-32-17_hcone_hcone
15:38:55.0 -    |cal_loc_RAW_spirou:02f10+[...]|Calibration file: spirou_wave_ini3.fits already e
15:38:55.0 -    |cal_loc_RAW_spirou:02f10+[...]|Calibration file: 2017-10-11_21-32-17_hcone_hcone
```

As mentioned above the calibration database was loaded automatically here:

```
In [10]: p['calibDB']

Out[10]: {'BADPIX': ['20170710', '20170710_flat_flat02f10_badpixel.fits'],
          'BLAZE_AB': ['20170710', '20170710_flat_dark02f10_blaze_AB.fits'],
          'BLAZE_C': ['20170710', '20170710_dark_flat02f10_blaze_C.fits'],
          'DARK': ['20170710', '20170710_dark_dark02d406.fits'],
          'FLAT_AB': ['20170710', '20170710_flat_dark02f10_flat_AB.fits'],
          'FLAT_C': ['20170710', '20170710_dark_flat02f10_flat_C.fits'],
          'LOC_AB': ['20170710', '20170710_flat_dark02f10_loco_AB.fits'],
          'LOC_C': ['20170710', '20170710_dark_flat02f10_loco_C.fits'],
          'ORDER_PROFILE_AB': ['20170710',
           '20170710_flat_dark02f10_order_profile_AB.fits'],
          'ORDER_PROFILE_C': ['20170710',
           '20170710_dark_flat02f10_order_profile_C.fits'],
          'TILT': ['20170710', '20170710_fp_fp02a203_tilt.fits'],
          'WAVE_A': ['20170710', 'spirou_wave_ini3.fits'],
          'WAVE_AB': ['AT4-04/2017-10-11_21-32-17',
           '2017-10-11_21-32-17_hcone_hcone02c406_wave_AB.fits'],
          'WAVE_B': ['20170710', 'spirou_wave_ini3.fits'],
          'WAVE_C': ['AT4-04/2017-10-11_21-32-17',
           '2017-10-11_21-32-17_hcone_hcone02c406_wave_C.fits']}

In [8]: # Note the acquisition time used is stored in two forms:
        print('max_time_human = {MAX_TIME_HUMAN}'.format(**p))
        print('max_time_unix = {MAX_TIME_UNIX}'.format(**p))
```

```
max_time_human = 2017-07-10-13:04:34.44
max_time_unix = 1499691874.44
```

Also note that the acquisition time decides which calibDB files are used. This depends on the constants.

- If `calib_db_match="older` the recipes will only allow calibDB entries that are OLDER than the time in the main fits file (i.e. < `max_time_human` and < `max_time_unix`. This is the setup in the old DRS.

- If `calib_db_match="closest` the recipes will used the closest entry in time to `max_time_human` and `max_time_unix`.

Note for this reason there is a check that `max_time_human` and "max_time_unix"' are the same and that the human time and unix time in the calibration database file are the same!

1.2 Updating the calibration database

This is very similar to before and involves a "PutFile" command to move a file into the calibration database folder and a "UpdateMaster" command to updated the master calibration database file (note one would have to re-run `LoadCalibDB` to update the in-memory version of the calibration database.

```
In [21]: # need a header for acquisition time in master calibration database file
         data, hdr, cmt, nx, ny = spirouImage.ReadImage(p, p['fitsfilename'], kind='TESTDATA')
         # print header ACQTIME
         wargs = ['ACQTIME', hdr['ACQTIME'], cmt['ACQTIME']]
         WLOG('', p['log_opt'], '{0} = {1} /{2}'.format(*wargs))
         # test output file
         outputfilename = '20170710_dark_dark02d406.fits'
         outputfile = os.path.join(p['REDUCED_DIR'], outputfilename)
         # set database key
         keydb = 'TEST'
         # copy localisation file to the calibDB folder
         spirouCDB.PutFile(p, outputfile)
         # update the master calib DB file with new key
         spirouCDB.UpdateMaster(p, keydb, outputfilename, hdr)
```

```
16:01:51.0 -   |cal_loc_RAW_spirou:02f10+[...]|Reading TESTDATA Image /scratch/Projects/spirou_p
16:01:51.0 -   |cal_loc_RAW_spirou:02f10+[...]|TESTDATA Image 2048 x 2048 loaded
16:01:51.0 -   |cal_loc_RAW_spirou:02f10+[...]|ACQTIME = 2457945.0448379633 /UTC Julian time
16:01:51.0 - * |cal_loc_RAW_spirou:02f10+[...]|Updating Calib Data Base with TEST
```

And the output in the master calibration database file is as follows:

```
TEST 20170710 20170710_dark_dark02d406.fits 2017-07-10-13:04:34.440000 1499691874.44
```