example3

May 7, 2018

1 Debug mode log error exception handling

Below we show how the logger (WLOG) is handled in debugging mode, using cal_DARK_spirou as an example:

In [14]:
```
# simulate cal_DARK_spirou
from __future__ import division

from SpirouDRS import spirouConfig
from SpirouDRS import spirouCore
from SpirouDRS import spirouStartup

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

# Notebook only code
import sys
sys.argv = ['cal_DARK_spirou.py']
night_name, files = None, None

# ----------------------------------------------------------------------
# Set up
# ----------------------------------------------------------------------
# get parameters from config files/run time args/load paths + calibdb
p = spirouStartup.Begin()
```

```
        p = spirouStartup.LoadArguments(p, night_name, files)
        p = spirouStartup.InitialFileSetup(p, kind='dark', prefixes=['dark_dark'])

13:44:04.0 -   || ****************************************
13:44:04.0 -   || * SPIROU @(#) Geneva Observatory (0.1.029)
13:44:04.0 -   || ****************************************
13:44:04.0 -   ||(dir_data_raw)      DRS_DATA_RAW=/scratch/Projects/spirou_py3/data/raw
13:44:04.0 -   ||(dir_data_reduc)    DRS_DATA_REDUC=/scratch/Projects/spirou_py3/data/reduced
13:44:04.0 -   ||(dir_calib_db)      DRS_CALIB_DB=/scratch/Projects/spirou_py3/data/calibDB
13:44:04.0 -   ||(dir_data_msg)      DRS_DATA_MSG=/scratch/Projects/spirou_py3/data/msg
13:44:04.0 -   ||(print_level)       PRINT_LEVEL=all         %(error/warning/info/all)
13:44:04.0 -   ||(log_level)         LOG_LEVEL=all         %(error/warning/info/all)
13:44:04.0 -   ||(plot_graph)        DRS_PLOT=1            %(def/undef/trigger)
13:44:04.0 -   ||(used_date)         DRS_USED_DATE=undefined
13:44:04.0 -   ||(working_dir)       DRS_DATA_WORKING=/scratch/Projects/spirou_py3/data/tmp
13:44:04.0 -   ||                    DRS_INTERACTIVE is not set, running on-line mode
13:44:04.0 -   ||                    DRS_DEBUG is set, debug mode level:1
13:44:04.0 -   |cal_DARK_spirou|Now running : cal_DARK_spirou on file(s):
13:44:04.0 -   |cal_DARK_spirou|On directory /scratch/Projects/spirou_py3/data/raw/
13:44:04.0 -   |cal_DARK_spirou|ICDP_NAME loaded from: /scratch/Projects/spirou_py3/spirou_py3/I
13:44:04.0 - ! |cal_DARK_spirou|Argument Error: No fits file defined at run time argument
13:44:04.0 - ! |cal_DARK_spirou|    format must be:
13:44:04.0 - ! |cal_DARK_spirou|    >>> cal_DARK_spirou.py [FOLDER] [FILES]
        Error found and running in DEBUG mode


        An exception has occurred, use %tb to see the full traceback.


        SystemExit: 1


/scratch/bin/anaconda3/lib/python3.6/site-packages/IPython/core/interactiveshell.py:2918: UserWa
  warn("To exit: use 'exit', 'quit', or Ctrl-D.", stacklevel=1)
```

In the console the following message will be displayed:

```
"Enter python debugger? [Y]es or [N]o?"

If "N" is selected the recipe will exit. If "Y" is selected the recipe will go into a python deb


        Error found and running in DEBUG mode

        Enter python debugger? [Y]es or [N]o?   Y
```

```
==== DEBUGGER ====

- type "list" to list code
- type "up" to go up a level
- type "interact" to go to an interactive shell
- type "print(variable)" to print variable
- type "print(dir())" to list available variables
- type "continue" to exit
- type "help" to see all commands

==================


> /scratch/Projects/spirou_py3/spirou_py3/INTROOT/SpirouDRS/spirouCore/spirouLog.py(217)debu
-> print(cc + '\n\nCode Exited' + nocol)
(Pdb)
```

This starts a basic python debugger with command such as: - list: which lists the last 10 lines of code before this error was generated (will normally just show the "error" catching in the logger (WLOG) function)

- up: this command goes "up" a level in the code, going up a couple of levels should get to the WLOG message which caused the crash.
- interact: this starts a python interactive session which can be used to access all variables currently stored in the memory and to see the currently assigned values of each variable

Shown below is a typical use for the debug based on the code above (unfortunately notebooks cannot run interactive code so this is copied as text)

- Here the code is "listed" which caused the recipe to exit (this level is where the debugger was entered so isn't the level we want to debug).

As one can see we are in the **spirouLog.py debug_start()** function

```
> /scratch/Projects/spirou_py3/spirou_py3/INTROOT/SpirouDRS/spirouCore/spirouLog.py(217)
(Pdb) list
212                         '\n\n\t ==================\n\n' + nocol)
213
214                 import pdb
215                 pdb.set_trace()
216
217  ->             print(cc + '\n\nCode Exited' + nocol)
218                 EXIT_TYPE(1)
219             else:
220                 EXIT_TYPE(1)
221         except:
222             EXIT_TYPE(1)
```

- Here the "up" command was used to go to up a level (the function that called the **debug_start()** function (**spirouLog.py logger()** function) this function is the logger function thus like **debug_start()** this isn't the level we want to debug.

```
(Pdb) up
> /scratch/Projects/spirou_py3/spirou_py3/INTROOT/SpirouDRS/spirouCore/spirouLog.py(136
-> debug_start()

(Pdb) list
131                 TDATA_WARNING = 0
132
133         # deal with errors (if key is in EXIT_LEVELS) then exit after log/print
134         if key in EXIT_LEVELS:
135             if spirouConfig.Constants.DEBUG():
136  ->             debug_start()
137             else:
138                 EXIT_TYPE(1)
139
140
141     def printlogandcmd(message, key, human_time, dsec, option):
```

- Here again we have used the "up" command (as the **logger()** function level is not useful for debugging the exception. Thus we are now at the call to the **logger()** functino via the alias **WLOG()**. This shows what was happening before the log "error" was raised. (In this case it is due to "fits_fn" having a value of "None").

```
(Pdb) up
> /scratch/Projects/spirou_py3/spirou_py3/INTROOT/SpirouDRS/spirouStartup/spirouStartup
-> WLOG('error', log_opt, [wmsg1, wmsg2, emsg.format(p['program'])])

(Pdb) list
254         # check that fitsfilename exists
255         if fits_fn is None:
256             wmsg1 = 'Argument Error: No fits file defined at run time argument'
257             wmsg2 = '    format must be:'
258             emsg = '    >>> {0}.py [FOLDER] [FILES]'
259  ->         WLOG('error', log_opt, [wmsg1, wmsg2, emsg.format(p['program'])])
260         if not os.path.exists(fits_fn):
261             WLOG('error', log_opt, 'File : {0} does not exist'.format(fits_fn))
262         # ------------------------------------------------------------------------
263         # if we have prefixes defined then check that fitsfilename has them
264         # if add_to_params is defined then add params to p accordingly
```

- At this stage it may be useful to be in interactive mode. Typing "interact" allows a python console to be opened (or ipython if it was detected that we were running in ipython). Below as with a standard python console we can explore the parameter space and indeed see that "fits_fn" has a value of "None". Obviously this example is trivia as we intentionally forgot to add "night_name" and "files" at run time.

```
(Pdb) interact
*interactive*

>>> log_opt
'cal_DARK_spirou'

>>> fits_fn is None
True
```