SPIROU-4800-LAM-UM-00961

SPIRou Data Reduction Software

Developer Guide

0.0.1

For DRS SPIRou0.0.1

N. Cook, F. Bouchy, E. Artigau, I. Boisse, M. Hobson, C. Moutou

2017-11-27

Abstract

This is the guide to coding the DRS (including installation, running, rules and stardisation approaches). This document is not intended for the general used of the DRS, instead it is intended for those who wish to develop the software further and understand the changes between this version and previous versions.

Contents

Introduction

Chapter 1

Installation

1.1 Introduction

Once finialised the installation should just be a download, run setup.py and configure the DRS directories, however, during development the following stages are required.
Currently the download repositry on github is private and requires a github account, and the user to be added to the list of collaborators. To be added to the collaborators please email neil.james.cook@gmail.com with your github username.

1.2 Download

Get the latest version of the DRS (for SPIRouversion 0.0.1). Use any of the following ways:

- manually download from here: https://github.com/njcuk9999/spirou_py3
- use Git:

```
git checkout https://github.com/njcuk9999/spirou_py3.git
```

- use SVN:

```
svn checkout  https://github.com/njcuk9999/spirou_py3.git
```

- use ssh:

```
scp -r git@github.com:njcuk9999/spirou_py3.git
```

1.3 Prerequisites

It is recommended to install the latest version of Anaconda python distribution, available for Windows, macOS and Linux (here: https://www.anaconda.com/download/). However one can run the DRS on a native python installation.

We recommend python 3 over python 2 for long term continued support (however the latest version of the DRS supports the newest versions of python 2.7).

Before installing the DRS you must have one of the following:

EITHER (RECOMMENDED)

- A valid version of the Anaconda python distribution (for python2 or python 3) Currently tested version of python are:
  - Python 2.7.13 and Anaconda 4.4.0
  - Python 3.6.3 and Anaconda 5.0.1 — RECOMMENDED

OR

- An up-to-date version of python and the following python modules (for either python 2 or python 3)
  - Python 3.6
    - ASTROPY (tested with version 2.0.2)
    - MATPLOTLIB (tested with version 2.1.0)
    - NUMPY (tested with version 1.13.3)
    - and the following built-in modules (comes with python): DATETIME, FILECMP, GLOB, OS, PKG_RESOURCES, SHUTIL, SYS, TIME, WARNINGS
  - Python 2.7
    - astropy (tested with version 1.3.2)
    - matplotlib (tested with version 2.0.2)
    - numpy (tested with version 1.12.1)
    - and the following built-in modules (comes with python): __FUTURE__, COLLECTIONS, DATETIME, FILECMP, GLOB, OS, PKG_RESOURCES, SHUTIL, SYS, TIME, WARNINGS

1.4 Installation Linux and macOS

Currently the DRS has to be installed manually. This involves the following steps:

1. Extraction (Section 1.4.1)
2. Modify environmental settings (Section 1.4.2)
3. Make recipes executable (Section 1.4.3)

1.4.1 Extraction

The first step is to extract the DRS into a folder (the {INSTALL_DIR}).
Do this by using the following commands:

```
cd INSTALL_DIR
unzip DRS.zip
```

1.4.2 Modify environmental settings

The next step is to modify your PATH and PYTHONPATH environmental variables (to include the {INSTALL_DIR}. This depends which shell you are using (type '`echo $0`' to find out which).

- In bash open the '.bashrc' text file in your home (~) directory (or create it if it doesn't exist)

```
export PATH={INSTALL_DIR}/bin/:$PATH

export PYTHONPATH={INSTALL_DIR}:{INSTALL_DIR}/bin/:$PYTHONPATH
```

- In csh /tcsh open the '.cshrc' or '.tcshrc' text file in your home (~) directory (or create it if it doesn't exist)

```
setenv PATH {INSTALL_DIR}/bin/:${PATH}

setenv PYTHONPATH {INSTALL_DIR}:{INSTALL_DIR}/bin/:${PYTHONPATH}
```

1.4.3 Make recipes executable

To run the recipes from the command line (without starting python) one must make them executable. Do this by using the following command:

```
chmod +x {INSTALL_DIR}/bin/*.py
```

1.5 Installation Windows

This is very similar currently to the Linux/macOS installation (in the future a '.exe' file will be given).

1. Extract to {INSTALL_DIR}with your favourite unzipping softwear.
2. Add {INSTALL_DIR}to your PYTHONPATH (Section 1.5.1)

1.5.1 How to modify environmental settings in windows

This process is a little more convoluted than on Linux or macOS system.

1. Go to 'My computer > Properties > Advanced System Settings > Enviromental Variables'.
2. if under system variable 'PythonPath' exists click edit and add '{INSTALL_DIR};' to the end.

   i.e.

   ```
   C:\Python27;{INSTALL_DIR};
   ```

3. if under system variables 'PythonPath' does not exist create a new variable called 'PythonPath' and add:

   ```
   %PYTHONPATH%;{INSTALL_DIR};{INSTALL_DIR}\bin\;
   ```

For problems/troubleshooting see here: https://stackoverflow.com/questions/3701646/how-to-add-to-the-pythonpath-in-windows-7.

1.6 Setting up the DRS

Before running the DRS one must set the data paths.

The 'config.txt' file is located in the {INSTALL_DIR}in the config folder.

i.e. at {INSTALL_DIR}/config/config.txt

The following keywords must be changed (and must be a valid path):

{TDATA}	=	/drs/data/	/	Define the DATA directory
{DRS_ROOT}	=	/drs/INTROOT/	/	Define the installation directory ({INSTALL_DIR})
{DRS_DATA_RAW}	=	/drs/data/raw	/	Define the folder with the raw data files in
{DRS_DATA_REDUC}	=	/drs/data/reduced	/	Define the directory that the reduced data should be saved to/read from
{DRS_CALIB_DB}	=	/drs/data/calibDB	/	Define the directory that the calibration files should be saved to/read from
{DRS_DATA_MSG}	=	/drs/data/msg	/	Define the directory that the log messages are stored in
{DRS_DATA_WORKING}	=	/drs/data/tmp/	/	Define the working directory

The directories here are for linux and macOS systems another example would be '/home/user/INTROOT' for the {INSTALL_DIR}directory. On Windows machines this would be equivalent to 'C:\Users\<username>\INTROOT' in Windows Vista, 7, 8 and 10 or 'C:\Documents and Settings\<username>\INTROOT' on early versions of Windows.

The following keywords can be changed:

{DRS_PLOT}	=	1	/	Whether to show plots
{PRINT_LEVEL}	=	"all"	/	Level at which to print
{LOG_LEVEL}	=	"all"	/	Level at which to log in log file

For the '{PRINT_LEVEL} and {LOG_LEVEL} keywords the values are set as follows:

- "all" – prints all events
- "info" – prints info, warning and error events
- "warning" – prints warning and error events
- "error" – print only error events

1.7 Validating Installion on Linux and macOS

Note one must install the DRS (Section 1.4) AND set up the DRS (Section 1.6).

There are four ways to run the DRS in Linux and macOS (thus four ways to verify installation was correct).

- To validate running from command line type:

```
cal_validate_drs.py
```

- To validate running from python/ipython from the command line type:

```
python cal_validate_drs.py

ipython cal_validate_drs.py
```

- To validate running from ipython, open ipython and type:

```
run cal_validate_drs.main()
```

- To validate running from import from python/ipython, open python/ipython and type:

```
import cal_validate_drs

cal_validate_drs.main()
```

If validation is successful the following should appear:

```
22:11:44.1 -   || ****************************************
22:11:44.1 -   || * SPIROU @(#) Geneva Observatory (0.0.1)
22:11:44.1 -   || ****************************************
22:11:44.1 -   ||(dir_data_raw)      DRS_DATA_RAW=/scratch/Projects/spirou_py3/data/raw
22:11:44.1 -   ||(dir_data_reduc)    DRS_DATA_REDUC=/scratch/Projects/spirou_py3/data/
    reduced
22:11:44.1 -   ||(dir_calib_db)      DRS_CALIB_DB=/scratch/Projects/spirou_py3/data/calibDB
22:11:44.1 -   ||(dir_data_msg)      DRS_DATA_MSG=/scratch/Projects/spirou_py3/data/msg
22:11:44.1 -   ||(print_level)       PRINT_LEVEL=all         %(error/warning/info/all)
22:11:44.1 -   ||(log_level)         LOG_LEVEL=all         %(error/warning/info/all)
22:11:44.1 -   ||(plot_graph)        DRS_PLOT=1            %(def/undef/trigger)
22:11:44.1 -   ||(used_date)         DRS_USED_DATE=undefined
22:11:44.1 -   ||(working_dir)       DRS_DATA_WORKING=/scratch/Projects/spirou_py3/data/tmp
    /
22:11:44.1 -   ||                    DRS_INTERACTIVE is not set, running on-line mode
22:11:44.1 -   ||
22:11:44.1 -   ||Validation successful. DRS installed corrected.
```

1.8 Validating Installion on Windows

Note one must install the DRS (Section 1.5) AND set up the DRS (Section 1.6).

In windows there are currently 3 ways to run the RS (running in python/ipython).

- To validate running from python/ipython from the command line type:

```
python cal_validate_drs.py

ipython cal_validate_drs.py
```

- To validate running from ipython, open ipython and type:

```
run cal_validate_drs.main()
```

- To validate running from import from python/ipython, open python/ipython and type:

```
import cal_validate_drs

cal_validate_drs.main()
```

If validation is successful the following should appear:

```
22:11:44.1 -   || ****************************************
22:11:44.1 -   || * SPIROU @(#) Geneva Observatory (0.0.1)
22:11:44.1 -   || ****************************************
22:11:44.1 -   ||(dir_data_raw)      DRS_DATA_RAW=/scratch/Projects/spirou_py3/data/raw
22:11:44.1 -   ||(dir_data_reduc)    DRS_DATA_REDUC=/scratch/Projects/spirou_py3/data/
    reduced
22:11:44.1 -   ||(dir_calib_db)      DRS_CALIB_DB=/scratch/Projects/spirou_py3/data/calibDB
22:11:44.1 -   ||(dir_data_msg)      DRS_DATA_MSG=/scratch/Projects/spirou_py3/data/msg
22:11:44.1 -   ||(print_level)       PRINT_LEVEL=all         %(error/warning/info/all)
22:11:44.1 -   ||(log_level)         LOG_LEVEL=all         %(error/warning/info/all)
22:11:44.1 -   ||(plot_graph)        DRS_PLOT=1            %(def/undef/trigger)
22:11:44.1 -   ||(used_date)         DRS_USED_DATE=undefined
22:11:44.1 -   ||(working_dir)       DRS_DATA_WORKING=/scratch/Projects/spirou_py3/data/tmp
    /
22:11:44.1 -   ||                   DRS_INTERACTIVE is not set, running on-line mode
22:11:44.1 -   ||
22:11:44.1 -   ||Validation successful. DRS installed corrected.
```

Chapter 2

Data Architecture

Chapter 3

Using the DRS

Chapter 4

Summary of changes (from version AT-4)

Chapter 5

Coding rules and standardisation practises

Chapter 6

Required input header keywords

Chapter 7

Description of Constants and Variables

Chapter 8

The Recipes

Chapter 9

The DRS Module