



Code Inspection

Name of the Reviewer: Ahsan Zia

Code Under Review: Main interface (oec_sync.py & oec_demo.py)

Date of the Review: November 24, 2016

Name of the Author of the Code under review: Ian Ferguson

Bugs:

- No bugs found.
- Good error checking/logging.

Poor code logic:

- Could move some of the main functions into separate files.
- Could break up main run and init functions into multiple more specific smaller functions.
- Code logic is correct everything functions as expected.

Poor coding style:

- run and init functions could be made smaller by breaking up into smaller functions.
- Good use of encapsulation/modular functions (i.e. classes called could be exported to other projects).
- Good variable/function names.
- Pep8 compliant.

Missing documentation:

- Fully documented.
- Good user command line help documentation.

Unreadable code:

- Good naming/commenting.
- Longer sections of code could be broken up for easier reading.

Vulnerabilities in code:

- Data is checked on input.
- All errors caught, or helpful error message is printed to console if given improper input.

Poor testing:

- Fully tested however automated testing is limited.
- Recommend additional automated testing.
- Note: As this is the main function this requires system level testing and validation test.
- Add evidence of validation testing activities.



Code Inspection

Name of the Reviewer: **Lucy Xing**

Code Under Review: (**git branch creation/updating**) `git_repo.py`

Date of the Review: **November 24, 2016**

Name of the Author of the Code under review: **Ahsan Zia**

Bugs:

- No bugs found.
- May fail to update specific branch if user manually adds branch of identical name to downstream GitHub repo via web interface (does not crash or effect other branches but will prevent creation of that specific branch). See lines 221 in `oec_sync.py` and `checkout` method in `git_repo.py`

Poor code logic:

- `cleanup_branches` method will attempt to delete remote branches even if they are already deleted (not error but unnecessary). However, it is unclear whether this is more or less efficient than checking for existence and then deleting only if present. Can you justify current approach.

Poor coding style:

- Pep8 compliant.
- Good encapsulation.
- `Stage` and `issue_pull_requests` methods are both dependant on `SystemXML` object class; an additional layer of abstraction could make it more reusable/modular.

Missing documentation:

- Fully documented/good commenting.

Unreadable code:

- Code is clear and readable.

Vulnerabilities in code:

- Does not check that upstream repo is valid. This is currently checked in `oec_sync.py` `init` function so will not effect overall program but does potentially effect reusability.
- Cannot create branches if user manually creates a branch of same name in downstream repo through GitHub website. Prevents update for that specific branch.

Poor testing:

- Good overall testing but lacks some testing for methods that interact with GitHub API.



Code Inspection

Name of the Reviewer: **Jubin Patel**

Code Under Review: **Map Aliases (in xml_functions.py)**

Date of the Review: **November 23, 2016**

Name of the Author of the Code under review: **Lucy Xing**

Bugs:

- Program does not crash, and operates correctly.

Poor code logic:

- Code logic is good

Poor coding style:

- Good coding style used.
- Pep8 compliant.
- xml_functions.py could be partially converted to SystemXML methods.
- The child classes of SystemXML are not very extensible beyond this project (that said parent class is and it is reasonable for classes like Star and Exoplanet to not be overly extensible).

Missing documentation:

- Code is fully documented/useful comments.

Unreadable code:

- Code is easily readable.

Vulnerabilities in code:

- All potential errors are handled.
- If xml is improperly formatted then will simply skip and log error, which is acceptable.

Poor testing:

- Good testing coverage.
- Somewhat limited variety of testing for outputted xml documents, may want to create additional test xml to compare against.



Code Inspection

Name of the Reviewer: Marhababanu Chariwala

Code Under Review: save_load_io.py

Date of the Review: November 22, 2016

Name of the Author of the Code under review: Jubin Patel

Bugs:

- No bugs found
- Exceptions are logged and errors are handled gracefully

Poor code logic:

- No redundant code logic
- No dead code
- Functionalities separated accordingly into proper functions

Poor coding style:

- Pep8 is followed
- Comments provide sufficient insight

Missing documentation:

- Fully documented/commented

Unreadable code:

- Internal comments and Docstrings provided for each function

Vulnerabilities in code:

- Data input is not checked, may lead to error(s)
- Unable to test Python's pickle library integrated with our code to ensure fully functioning code

Poor testing:

- Testing is not automated
- Testing is trivial compared to other components of the system
- Testing is limited due to the code



Code Inspection

Name of the Reviewer: Ian Ferguson

Code Under Review: Crontab (Daily Scheduler)

Date of the Review: November 24, 2016

Name of the Author of the Code under review: Marhababanu Chariwala

Bugs:

- No bugs are found
- Exception are handled correctly

Poor code logic:

- No redundant or duplicate code
- No dead code
- Good use to OS's utility

Poor coding style:

- Follows pep8 style
- Could improve naming style so it more understandable to the reader

Missing documentation:

- Docstring and internal comment are provided

Unreadable code:

- No unreadable code found
- Code is readable

Vulnerabilities in code:

- Data inputs not checked might lead to error
- Third-party library (python-crontab) is used but not all errors are caught
- Invalid parameters not handled

Poor testing:

- Testing does not have good coverage
- Adding more test case using stub to check the dependencies