User Manual

for the ISS Check Status Engine

PREPARED BY

Daniel Wooten

For Allied Financial

A Quick Introduction

Welcome to the ISS Check Status Engine!

The ISS Check Status Engine is a Python 3 program.

In this user manual you will find three sections, installation, use, and testing.

Installation will cover how to install the ISS Check Status Engine.

Use will cover how to use the ISS Check Status Engine.

Testing will cover how to test the ISS Check Status Engine.

If you feel lost at this point, read on! Or, find the README file in the top most directory of this package and give that a read, it might be more of what you're looking for.

Once you've got the basics down, which won't take long at all, use of this software will allow you to easily view the current location and crew roster of the ISS as well as predict when and for how long the ISS will pass overhead a pair of coordinates you supply.

I hope you find this software package as fun and easy to use as I did to build.

Yours Truly,

Daniel Wooten

Installation

The following python packages must be installed on your system and locatable on your PYTHONPATH environment variable...

1. sys
2. os
3. urllib
4. json
5. math
6. datetime
7. ephem
8. calendar
9. redis
10. Time

The following programs must be installed on your system and located on your PATH environment variable...

1. redis-server

Once the above dependencies have been installed on your system and this package

is as well, go into the source/ subdirectory found inside of the src/ directory

and adjust the permissions of the 'main.py' file to be executable.

Important!!

In the very first line of the main.py file replace the existing file path with your system's absolute path to its Python 3 interpreter.

Congratulations!! You have successfully installed the ISS Check Status Engine!

Use

The ISS Check Status Engine is invoked by calling the executable *main.py* from the command line. This invocation must be followed by one of three optional command arguments. These options are detailed below as well as their resulting behavior.

Option 1: loc

From the command line execute the following command…

```
./main.py loc
```

…to retrieve and print to stdout the current latitude and longitude of the ISS.

Option 2: people

From the command line execute the following command…

```
./main.py people
```

…to retrieve and print to stdout the current crew roster of the ISS.

Option 3: pass

From the command line execute the following command…

```
./main.py pass [latitude] [longitude]
```

…to retrieve and print to stdout when and for how long the ISS will be overhead of the coordinates specified by *[latitude]* and *[longitude]*.

[latitude] should be a floating point number in the range of [-90, 90].

[longitude] should be a floating point number in the range of [-180, 180].

Testing

The ISS Check Status Engine was developed in a test driven development environment. As such there is a suite of tests for the ISS Check Status Engine.

Unit Testing

To run the unit testing suite make the file, *run_unit_tests.py*, executable. This file is found in the src/ directory. Additionally, you must change the absolute file path found at the top of the file to point towards your system's Python 3 interpreter. Upon executing this file the unit test suite will be run and the results will be printed to an ASCII file named *unit_Tests.test* which will be created in the executing directory of the *run_unit_tests.py* file.

Integration Testing

To run the integration testing suite make the file, *run_integration_tests.py*, executable. This file is found in the src/ directory. Additionally, you must change the absolute file path found at the top of the file to point towards your system's Python 3 interpreter. Upon executing this file the integration test suite will be run and the results will be printed to an ASCII file named *integration_Tests.test* which will be created in the executing directory of the *run_integration_tests.py* file.