

A Quick Start to `rss_ringoccs`

©Team Cassini at Wellesley College
Richard G. French*, Sophia R. Flury, Jolene W. Fong,
Ryan J. Maguire, and Glenn J. Steranka

**Cassini Radio Science Team Leader,
rfrench@wellesley.edu*

July 1, 2019

This is a step-by-step outline for setting up and using `rss_ringoccs` software. For details on the use of `rss_ringoccs` or the inner workings of the software, see the User's Guide and online documentation at <https://rss-ringoccs.readthedocs.io/en/master/>.

Setup

- Note that all commands should be run using the bash shell.
- Navigate to the local directory where you would like to store and run `rss_ringoccs`. Download the `rss_ringoccs` software package by cloning from GitHub repository using

```
git clone https://github.com/NASA-Planetary-Science/rss_ringoccs.git
```

Note that, when downloading the zip file instead of cloning from GitHub, the top-level directory will be `rss_ringoccs-master` instead of `rss_ringoccs`.

- Navigate to the `rss_ringoccs` directory and run the configuration shell script to set up the appropriate modules and software packages

```
cd rss_ringoccs  
./rss_ringoccs_config.sh
```

Example Processing

- Navigate to the `rss_ringoccs/examples` directory

```
cd rss_ringoccs/examples
```

- Acquire relevant files needed for processing example

```
./get_example_files.sh
```

- Run end-to-end processing example script

```
yes | python e2e_run.py
```

- Edit the `rss_ringoccs/examples/quick_look_example.py` script by replacing the 'YYYYMMDD' in line 15 to match the date of the *.TAB files output by the `e2e_run.py` script run in the previous step. Then run the quick-look processing example script

```
python quick_look_example.py
```

Batch Processing

- Navigate to the `rss_ringoccs/pipeline/` directory

```
cd rss_ringoccs/pipeline
```

- From the `pipeline` directory, acquire all reference files needed for total set of occultations. Download time may vary with bandwidth. Anticipate at least 15 mins for these to complete.

```
./get_all_kernels.sh
```

- From the `pipeline` directory, acquire all 1 kHz RSR files prior to USO failure and some 1 kHz RSR files post-USO failure. Download time may vary with bandwidth. Anticipate at least 1 hr for these to complete.

```
./get_1kHz_rsr_files_preUSOfailure.sh  
./get_1kHz_rsr_files_postUSOfailure.sh
```

- From the `pipeline` directory, run this batch file for a 1 km reconstruction of occultations prior to USO failure. Execution time may vary with local hardware. Anticipate at least 3.5-4 hrs for this script to run.

```
yes | python e2e_batch_1km.py
```

- From the `pipeline` directory, run this batch file for a 500 m reconstruction of occultations prior to USO failure. Execution time may vary with local hardware. Anticipate at least 15 hrs for this script to run.

```
yes | python e2e_batch_500m.py
```

- For a 1 km reconstruction of occultations post-USO failure, run this batch file. Execution time may vary with local hardware. Anticipate at least 10 hrs for this script to run.

```
yes | python e2e_batch_postUSO_1km.py
```

- For a customized reconstruction of a particular set of RSR files and user inputs, edit the `rss_ringoccs/pipeline/e2e_batch_args.py` file and run the corresponding `e2e_batch.py` script

```
yes | python e2e_batch.py
```

Scattered Signal Processing

- To run the script that processes and outputs the incoherent, or scattered, signal, navigate to the `rss_ringoccs/pipeline/` and run the end-to-end scattered signal batch file. Execution time may vary with local hardware. Anticipate at least 3.5-4 hrs for this script to run.

```
cd rss_ringoccs/pipeline  
yes | python e2e_scattered.py
```