RST Release Guidelines

During the SuperDARN Data Analysis Working Group (DAWG) telecon on 19 March 2018, it was agreed that guidelines should be drafted to govern the versioning and timeframe of future releases of the Radar Software Toolkit (RST - https://github.com/SuperDARN/rst). This document is a formalization of those guidelines.

Note that this document is not intended to address the testing procedure for individual commits or pull requests. For details on recommended testing procedures, please see the RST Testing Guidelines documentation (TBD, but in the meantime see Section 1 of the minutes of the SD'18 DAWG meeting: https://homepage.usask.ca/~pbp672/DAWG_minutes_20180604_Banyuls.pdf).

1. Versioning

Semantic Versioning 2.0.0 (https://semver.org/spec/v2.0.0.html) should be used as a starting point for assigning version numbers to new RST releases. Generally speaking, the version number should follow the format MAJOR.MINOR.PATCH, where

- MAJOR is incremented for a major refactoring/redesign of the entire package
- MINOR is incremented for addition of new functionality (default or optional)
- PATCH is incremented for bug fixes and hardware/radar table updates

As an example, the MAJOR version number was incremented from VTRST3.5 to v4.0 when the rpkg and ROS software were removed and significant changes were made to the compilation scripts. The MINOR version number was incremented from v4.0 to v4.1 when new features were added such as FITACF3.0, the Chisham virtual height model, the PSR10 and CS10 statistical models, etc. To date we have not implemented PATCH version numbers, although this has been proposed in response to bugs breaking the functionality of the "gridtogrdmap" and "maptocnvmap" binaries in v4.1.

If the MINOR version number is incremented, then the PATCH version number should be reset to zero. Similarly, if the MAJOR version number is incremented then both the MINOR and PATCH version numbers should be reset to zero.

2. Release Schedule

For a new release to become official, a release branch must first be created from the develop branch. This release branch must then be merged into the master branch of the RST. Finally, a new release must be drafted and tagged on Github corresponding to the master branch at that time. This procedure follows the git branching model guidelines adopted from https://nvie.com/posts/a-successful-git-branching-model/. Note that pull requests should not be made directly from the develop branch to the master branch.

There is no regular schedule which needs to be followed regarding new releases of the RST, except for these considerations:

- At least one month of testing should be performed on each new MAJOR release branch before merging into the master branch
- No more than two weeks of testing should be performed on each new MINOR release branch before merging into the master branch
- No more than one week of testing should be performed on each new PATCH release branch before merging into the master branch
- No more than 6 months should pass without a new MINOR release if at least one new feature has been merged into the develop branch
- No more than 3 months should pass without a new PATCH release if at least one bug fix has been merged into the develop branch, however depending on the severity of the bug fix the PATCH release should be merged as quickly as possible

Note that commits which do not modify any functionality (e.g., fixing typos in comments or descriptions) do not qualify as a bug fix and thus do not warrant an eventual PATCH release on their own.

3. Distribution

An email announcing the new RST release should be sent to the SuperDARN community. Currently, the darn-users mailing list is the distribution mechanism however an alternative mailing list (such as a new “rst-users”) should be explored. Members of the DAWG should promptly update their own websites to link to the zip or tar.gz archives of the newest RST release.

Aside from the release tag, a more permanent URL for the latest master branch code can be found at https://api.github.com/repos/superdarn/rst/zipball/master for a zipball or https://api.github.com/repos/superdarn/rst/tarball/master for a tarball. Note that these download links point to the current master branch only and will not necessarily correspond to an official release of the software associated with a particular version number and citable DOI.

4. Archiving

A Github repository (https://github.com/SuperDARN/rst-archive) has been created to store all previous and current versions of the RST. When new versions of the RST are released, the zip/tar archive of the new release should be unpacked and added to a new branch of the rst-archive repository. Branches should be created from the master branch and follow the naming convention “rst.MAJOR.MINOR{.PATCH}”.