RHEOS Versioning - A Brief Overview

Abstract

This document is a short explanation of how the RHEOS versioning procedure works, and how it relates to documentation, CI testing and the RHEOS central package repository.

1 Basic Versioning Procedure

1.1 Tools and their Use

The basic versioning procedure relies on the following three components: the Project.toml file that lives in the RHEOS root directory, the Julia TagBot GitHub app, and the JuliaTeam Registrator GitHub app.

The above components are used for RHEOS versioning in the following way. A number of changes are made to RHEOS directly to the master branch, at some point it will make sense to release a new version. When this moment arrives, a final commit should be made that contains an updated Project.toml file. The information added to the Project.toml commit is the version number (for example `version = "0.9.2"` at time of writing) and any compatibility information (e.g. the version(s) of Julia that the package is compatible with and any specific version numbers required of dependencies). For information on exactly how this information should be formatted in the Project.toml file, see this link. **IMPORTANT NOTE: The Julia versions which the package is specified to be compatible with should be in the CI testing matrix on Linux and Windows and should be passing, see section 2 for more info**. After this commit is pushed, you need to navigate to that commit in GitHub by going to the main RHEOS page, clicking commits, and then on the relevant commit. You can then invoke the JuliaRegistrator bot by creating a comment saying `@JuliaRegistrator register()`, as for example this link. This will create a pull request on the Julia central package repositry. **You do not need to manually tag the release, this will be done automatically by the Julia TagBot app.**

So a high level overview is the following:

- Make changes on master branch, add tests as necessary and make sure all tests are passing in required Julia versions
- When a version is ready to be finalised, update Project.toml, then commit
- Invoke Registrator.jl which pushes necessary information to the central package repository
- TagBot will then automatically create a release on the RHEOS repository

Note: it is best practice to update the testing matrix as soon as possible, so that the appropriate versions of Julia are being tested on *during development*.

1.2 Version Number Guidelines

There is some debate in the programming community as to the best version numbering system. Generally speaking, for version X.Y.Z, an increment in Z is a minor update which contains small changes and bug-fixes. An increment in Y would contain one or more new features. Finally, an increment in X could be a major update which significantly changes the way users interact with the program, e.g. major syntax changes. Agreement should be sought with main commiters before a version is finalised. For now, we should try and follow Semantic Versioning 2.0.0, a description of which can be found at https://semver.org/.

2 Testing Considerations

As described in the testing documentation, the RHEOS github repository is set up with Travis CI and Appveyor to automatically run tests on Linux and Windows servers respectively. If any of the tests fail, this will be indicated on the appropriate badge in the github repository README page.

If we make changes in the Project.toml file saying that the package is compatible with particular version(s) of Julia then all RHEOS tests should be passing on those versions of Julia. Generally we will want compatibility on as many Julia versions as is practically feasible.

For an example of how to specify which version of Julia RHEOS will be tested on, you can check the appveyor.yml and .travis.yml in the RHEOS repository, or many other Julia packages.

3 Relation to Documentation

Documenter.jl will make a version of the documentation for each version of RHEOS. The most recent release of RHEOS is considered the 'stable' documentation and the current master branch build is the 'dev' documentation. There

is a link to both versions on the RHEOS readme page. For more information on this, see the Documenter.jl documentation, or this link more specifically under the section 'Documentation Versions'.