VERA C. RUBIN OBSERVATORY

rubin-env

Kian-Tat Lim

NSF AURA U.S. DEPARTMENT OF ENERGY SLAC CHARLES AND LISA SIMONYI FUND FOR ARTS AND SCIENCES LSST CORPORATION

What Is rubin-env?

- conda metapackage (no contents, only references to other packages)
- Contains all conda-forge-published third-party dependencies
  - Except eigen and ones under development by Rubin developers
  - Others where we need to patch upstream would be similar
- Most are unpinned; exceptions at https://confluence.lsstcorp.org/display/DM/DM+Third+Party+Software
  - C++ interfaces are pinned at major version (or sometimes minor)
  - Unpinning gives flexibility when installing other packages on top
  - Actual versions are recorded for builds and releases in Jenkins and eups.lsst.codes (lsstinstall.sh will give access)
- No build-time or runtime dependency of stack code on rubin-env version

Semantically Versioning rubin-env

- rubin-env build version bumps:
  - compatible with previous and new code; same versions as before
  - < pins for existing dependencies
- rubin-env patch version bumps:
  - compatible with previous and new code; code can rely on fixes
  - > pins for existing dependencies (should probably use this more)
  - < pins if to earlier than previous patch
- rubin-env minor version bumps:
  - compatible with previous and new code; new packages/features
  - Add new dependencies
- rubin-env major version bumps:
  - compatible with new code only
  - Remove dependencies or major dependency updates

Using Upgraded Dependencies

- Dependency bumps:
  - Canaries: Jenkins lsst_distrib and ci_hsc clean builds always use fresh environment
  - stack-os-matrix also uses fresh environment but shouldn't and may not rebuild LSST packages
  - Release always uses fresh environment but may not rebuild LSST packages
  - lsstsqre/centos + RSP containers and shared stack never use fresh environments
- rubin-env build bumps:
  - Could manually update container base and shared stack
- rubin-env patch/minor/major bumps:
  - rubin-env default version coded into newinstall.sh, lsstsw/deploy, and Jenkins configuration
  - Shared stack needs to be manually updated with a new environment (although theoretically a new rubin-env could be installed in the same environment up to minor bumps)
- If we cannot upstream patches, create an eups TaP package or forked package