

# DM Third Party Software





## rubin-env conda metapackage

Most of the Science Pipelines third-party dependencies are obtained from conda-forge via the [rubin-env metapackage](#). This applies to both C++ and Python libraries. Packages included in rubin-env are intended to be the *minimal* set that is necessary and appropriate for use in all environments, including production, development, and the Rubin Science Platform. Each of those environments can add additional packages as needed for its users.

Packages linked to from C++, including numpy, are pinned via the conda\_build\_config file. We need to do this to ensure binary compatibility with C++ shared libraries. It is usually sufficient to do this at the major version level assuming semantic versioning is used, assuming that we generally use older binaries with newer dependency libraries. Some packages pin at the minor or exact version level due to their versioning practices.

The following packages have unusual version constraints due to recognized incompatibilities or other reasons. Rows in [light blue](#) are present for Rubin development standardization reasons. Rows highlighted in [light green](#) seem eligible for removal in future releases of the metapackage. Jira issues blocking constraint removal are listed where known. Constraints that eliminate only particular versions that were incompatible but do not constrain future versions (!=) as well as minimum constraints imposed by the package's use in LSST code (>=) are not listed.

Historic information for old pins may be found in previous versions of this page.

Package	Constraint	Reason	Last Checked
flake8	= 6.0	Define a syntax standard; avoid "blind-siding" caused by new rules.  We have stopped using pytest-flake8 and will start using ruff instead.	N/A
python	= 3.11	Officially supported version	N/A
ruff	= 0.1.7	Define a syntax standard; avoid "blind-siding" caused by new rules.	N/A
sysroot_linux-64	= 2.17	Obtain CentOS 7-compatible libraries; enables CLOCK_TAI	N/A
boost	= 1.82	Recommendation is to pin only at even versions less than or equal to what is in <a href="#">the global pinnings</a>	2023-12-19
htcondor	>=23.0,<23.1	23.0 is the long-term support branch; new features have 23.x versions.	2023-12-19
moto	>=3,<5	5.0 breaks resources tests. <div> <a href="#">DM-42698</a> - Jira project doesn't exist or you don't have permission to view it.</div>	2024-01-27
parsl	=2023.06.12	Parsl seems to be releasing on a weekly cadence now with no guarantees of compatibility from one to the next, so we have to do an = pin at what we've tested.	2023-12-19
pybind11	<2.11	2.11 breaks afw tests. <div> <a href="#">DM-40043</a> - Jira project doesn't exist or you don't have permission to view it.</div>	2023-12-19
pytest	<8.1.0	8.1.0 breaks all tests.  <a href="#">DM-43144</a> - pytest 8.1.0 breaks Science Pipelines builds <span>IN PROGRESS</span>	2024-03-03
scipy	<1.12.0	1.12 breaks shapelet test. <div> <a href="#">DM-42582</a> - Jira project doesn't exist or you don't have permission to view it.</div>	2024-01-21

## Third-party packages in eupS

There are two kinds of third-party packages distributed via eupS: tarball-and-patch (TaP) packages and source code packages based on forks with `lsst-dev` branches. We have eliminated all of the TaP packages, replacing them with conda-forge packages.

These packages are under active development by Rubin developers, so we distribute them as forked source code packages:

- `fgcm`

- `kht`
- `proxmin`
- `scarlet`
- `scarlet_extensions`
- `spectractor`

## rubin-env-rsp

This package includes all dependencies that can be installed by conda/mamba from conda-forge for the Rubin Science Platform containers.

Package	Constraint	Reason	Last Checked
jupyterlab	>=3,<4	Protect against potential future incompatibility.	
jupyterlab_execute_time	>=2,< 3	Protect against potential future incompatibility.	

## rubin-env-developer

This package includes all dependencies that can be installed by conda/mamba from conda-forge for developer installations such as the "shared stack" at the USDF.

## "Extras" packages

Additional packages beyond those needed by batch production jobs are added to a conda-forge metapackage output called `rubin-env-extras`. This metapackage is not intended to be installed anywhere, but it ensures that there are compatible packages available when subsets are installed in other contexts: Rubin Science Platform container images, the "shared stack" developer installation, and the rubin-sim conda environment.

Package	Source	Constraint	Comment	Last Checked
---------	--------	------------	---------	--------------