Python 3 at the Large Synoptic Survey Telescope

Tim Jenness (AURA/LSST, Tucson, AZ, USA)

INTRODUCTION

The LSST software systems make extensive use of Python, with almost all of it initially being developed solely in Python 2. Since LSST will be commissioned when Python 2 is end-of-lifed it is critical that we have all our code support Python 3 before commissioning begins. Over the past year we have made significant progress in migrating the bulk of the code from the Data Management system onto Python 3. This poster presents our migration methodology, and the current status of the port, with our eventual aim to be running completely on Python 3 by early 2018.

PYTHON 3 PORTING

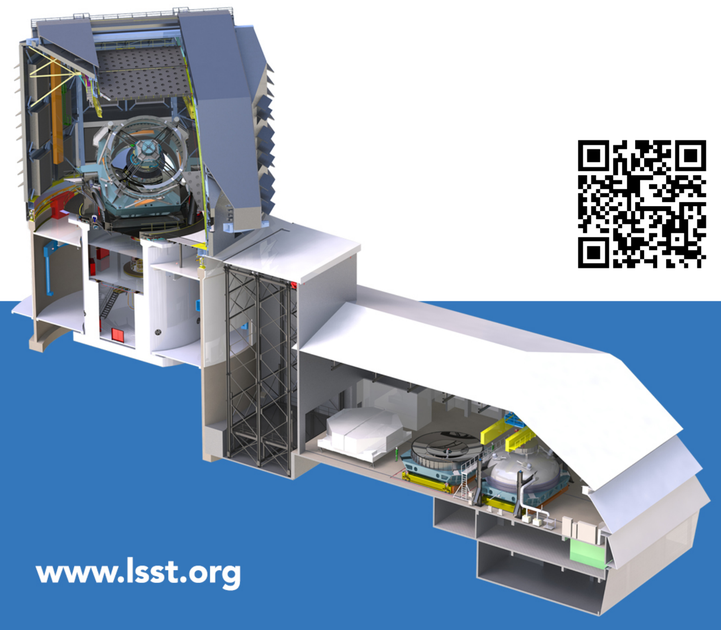
We recommend using the Python "future" package to port Python 2 software to Python 3 and to aid development when supporting both Python versions. We chose "future" because it is designed to make compatible code look like it is written for Python 3 natively and minimizes explicit checks for Python version.

The "futurize" command worked really well and support for the two stage conversion was important. Stage 1 modernizes code to use Python 2.7 constructs such as modern exception catching, checking if a key is in a dict using "in" rather than "has_key", and use of __future__ for print function.

Stage 2 does more extensive rewrites to support Python 3 changes to builtins and the standard library. It also replaces map(filter(lambda()) constructs with more readable list comprehensions.

We found that using the Unicode "str" object added more complication than was desired so we have left string objects as their native type in many places. Proper handling of bytes and strings is the biggest headache when switching Python versions.

The LSST baseline Python 3 is version 3.6.



LSST PYTHON SOFTWARE

Python is used widely at LSST. They key software is:

- Science Pipelines.
- VO and Data Access Services
- Simulations software.
- LSST Scheduler.
- Wavefront Sensor data processing.

The biggest user of Python is Data Management (DM) which has approximately 150,00 lines of Python, and the bulk of this software was ported to Python 3 in the second half of 2016. DM are regularly running Python 3 now for science pipelines, data access services and the Qserv database administration scripts. DM currently use some DESDM infrastructure at NCSA; those will be ported by early 2018.

The "sims" software has been ported to Python 3 and has no remaining issues.

The Scheduler software has had an initial modernization pass to support Python 3 but has not yet been tested. The Scheduler uses the Software Abstraction Layer (SAL) message system (based on DDS) and the Python 3 bindings were not available until October 2017. We expect the Scheduler to be running with Python 3 by the end of the 2017.

The wavefront sensor processing software uses DM software and will be running Python 3 when it is completed.

OTHER CHANGES

In addition to supporting Python 3, we have made two major improvements to the Python infrastructure for DM and Simulations:

- We have replaced our SWIG bindings with pybind11. pybind11 requires that the interface be defined manually, but does result in a cleaner separation of C++, interface, and Python code.
- We have adopted pytest as our test execution framework. This has significantly enhanced the reporting of test metrics in Jenkins CI and plugin support has allowed us to add automated flake8 testing, parallel execution with xdist, and code coverage.

DROPPING PYTHON2

We have been supporting Python 3 and Python 2 in the Data Management software since Summer 2016, and this has given our user community time to become accustomed to Python 3. There is though a cost to supporting Python 2, with extra Continuous Integration resources, source code distractions ("cruft") with constructs that are not needed in Python 3, and developers having to understand that they can't use Python 3 features and either running with two local installations or discovering late that their working code fails on Python 2. To simplify our development roadmap and give clarity to the community, LSST DM will drop support for Python 2 following the release of v15.0 of the DM Pipelines Stack in Spring 2018. We will support critical bug fixes to v15.0 until the release of v16.0 in late 2018. This timeline is consistent with the release of Astropy v3.0; their first release without Python 2 support.

DM Pipelines v14.0

Version 14.0 of the DM Science Pipelines Stack was released in October 2017. This is the first release using pybind11. This release also begins to use the Starlink AST library for WCS handling.

Full release notes can be found at:
https://pipelines.lsst.io/releases/notes.html

SUMMARY

LSST will be using Python 3.6 internally starting in 2018. This includes data challenges, and integration and testing activities. The Data Management software will drop support for Python 2 following the release of v15.0 in Spring 2018.

More Information

LSST DM Python 3 Porting Guide: https://sqr-014.lsst.io
LSST Data Management Overview: arXiv:1512.07914
LSST Design Overview: arXiv:0805.2366
LSST Science Book: arXiv:0912.0201
LSST Data Products: http://ls.st/LSE-163
DM Applications Design: http://ls.st/LDM-151
Key Numbers: http://lsst.org/scientists/keynumbers

www.lsst.org

This material is based upon work supported in part by the National Science Foundation through Cooperative Agreement 1258333 managed by the Association of Universities for Research in Astronomy (AURA), and the Department of Energy under Contract No. DE-AC02-76SF00515 with the SLAC National Accelerator Laboratory. Additional LSST funding comes from private donations, grants to universities, and in-kind support from LSSTC Institutional Members.