

Erik M. Bray

Space Telescope Science Institute, 3700 San Martin Drive, Baltimore, MD 21218, USA

Introduction

The Astropy Project is a community effort to develop an open source Python package of common data structures and routines for use by other, more specialized astronomy software in order to foster interoperability.

The project encompasses the "core" astropy Python package, "affiliated packages" that strive to implement Astropy's coding standards and interoperability with other affiliated packages, and a broader community aimed at implementing Pythonic solutions to astronomy computing problems while minimizing duplication of effort.

Here we present an overview of the key features of the core package and some of the free and open source technologies and services we use to manage a large project with a diverse community of contributors.

Astropy Affiliated Packages

An affiliated package is a Python package that is not part of the core astropy package, but is under the umbrella of the Astropy Project. Some affiliated packages may be moved into the core package once they reach a reasonable level of maturity. Others may be more specialized or have enough external dependencies (GUI libraries, for example) that they make more sense as separate packages. In general we hope that becoming an affiliated package will be seen as a good way for new and existing packages to gain exposure.

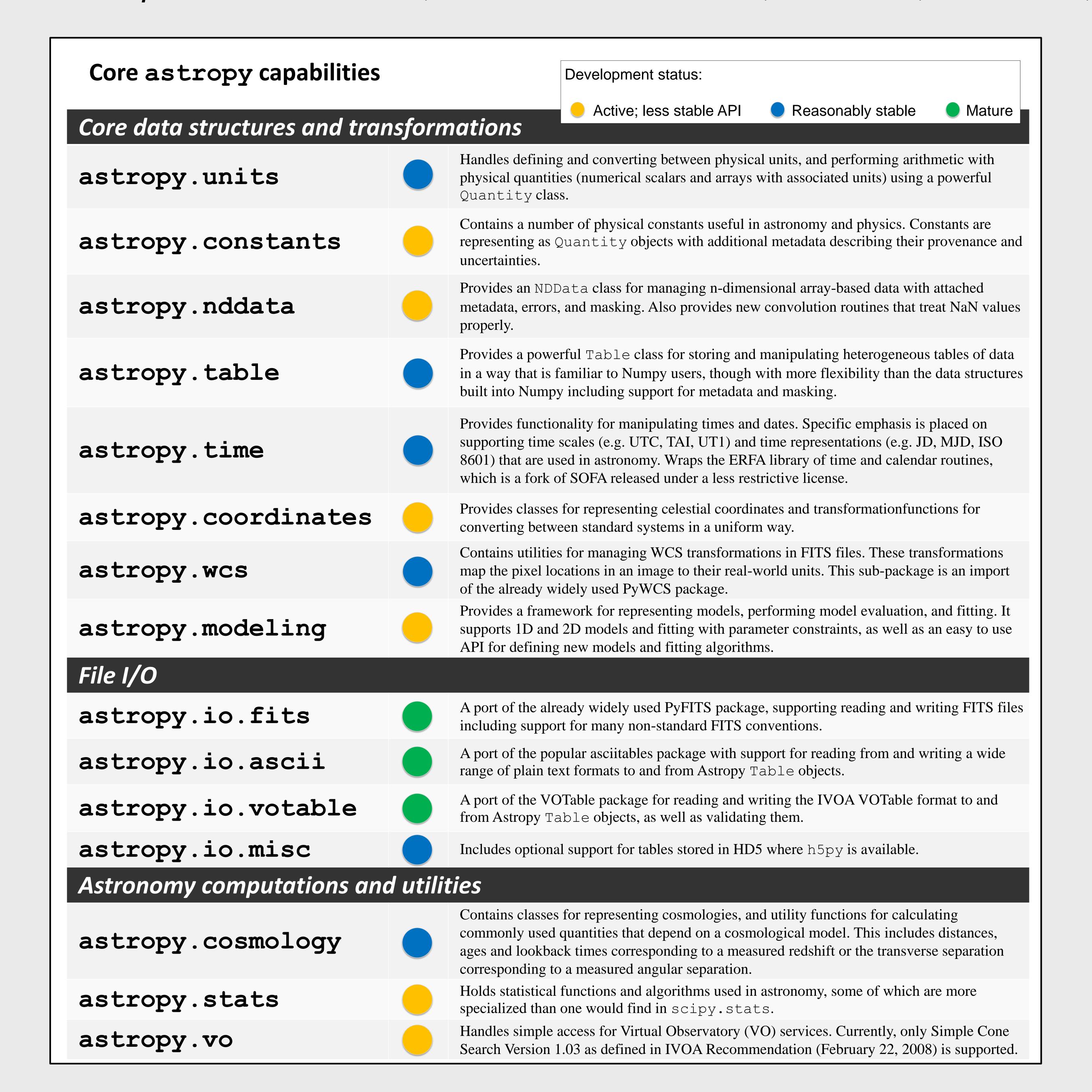
Some currently featured affiliated packages include Montage-wrapper, Ginga, APLpy, astroML, and Astropysics. A few additional affiliated packages that are currently under development are photutils, astroquery, specutils, and kcorrect.

Citing Astropy

If you use Astropy for work/research presented in a publication, we would be grateful if you could include the following citation:

The Astropy Collaboration, Robitaille T. P., Tollerud E. J., Greenfield P., et al. 2013, ArXiv e-prints. 1307.6212

which is a citation to this paper, on which this poster is based: http://arxiv.org/abs/1307.6212



Community and Collaboration

A primary guiding principle behind Astropy has been that it is developed by and for the astronomy community, and that it strives to include contributions from users and developers from around the world and from all sub-disciplines. It also aims to encourage best practices from software engineering in the hope of creating a clean, consistent, and stable API; thorough and well-maintained documentation; and constant feedback through careful issue tracking and continuous integration testing. All of our development practices and guidelines are documented so that other projects—whether affiliated with Astropy or not—may adopt them without having to develop their own standards from scratch.

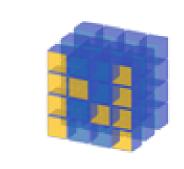
In order to better enable collaboration we have adopted the GitHub code hosting service which provides an issue tracker and a git source code repository. GitHub is designed around collaboration and enables a low barrier of entry to submitting new code and bug fixes to open source projects via "pull requests".

Astropy's development process ensures that any bug fixes or new features that are implemented are accompanied by tests, and any relevant updates to the documentation. Free documentation building and hosting by Read the Docs ensures that the latest documentation builds and is accessible online. All new pull requests are tested against multiple Python and NumPy versions using the continuous integration system Travis. Comprehensive testing against multiple OS platforms (Linux, MacOS X, and Windows) is made possible by Jenkins with hosting from Shining Panda.

Having an accessible development workflow has attracted source code contributions from over 50 developers, and many more have contributed design, feedback, and testing.

Technologies and Services We Use

python







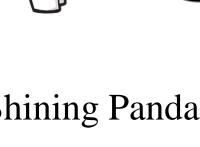












Python, NumPy, SciPy, Matplotlib, Sphinx, Git, GitHub, Read the Docs, Travis, Jenkins, Shining Panda, py.test (not pictured)