easyaccess: Enhanced SQL command line interpreter for astronomical surveys

Matias Carrasco Kind[1], **Alex Drlica-Wagner**[2], **Audrey Koziol**[1], **and Don Petravick**[1]

1 National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign. 1205 W Clark St, Urbana, IL USA 61801 **2** Fermi National Accelerator Laboratory, P. O. Box 500, Batavia,IL 60510, USA

Summary

`easyaccess` is an enhanced command line interpreter and Python package created to facilitate access to astronomical catalogs stored in SQL Databases. It provides a custom interface with custom commands and was specifically designed to access data from the Dark Energy Survey Oracle database, although it can easily be extended to another survey or SQL database. The package was completely written in Python and support customized addition of commands and functionalities. Visit https://github.com/mgckind/easyaccess to view installation instructions, tutorials, and the Python source code for `easyaccess`.

The Dark Energy Survey

The Dark Energy Survey (DES) (DES Collaboration 2005; DES Collaboration 2016) is an international, collaborative effort of over 500 scientists from 26 institutions in seven countries. The primary goals of DES are reveal the nature of the mysterious dark energy and dark matter by mapping hundreds of millions of galaxies, detecting thousands of supernovae, and finding patterns in the large-scale structure of the Universe. Survey operations began on on August 31, 2013 and will conclude in early 2019. For about 500 nights, DES has been taking thousands of deep images of southern sky, which are transferred and processed at the National Center for Supercomputing Applications (NCSA). The images are processed to produce catalogs of astronomical sources with hundreds of millions of entries (billions in the case of individual detections), describing the sources found within the images and other relevant metadata. A significant subset of the DES data was recently made public (DES Collaboration 2018) and can be accessed through several mechanisms including `easyaccess` and web interfaces that run `easyaccess` as a backend. This public release includes information for almost 400 million astrophysical sources and complementary tables to allow scientific analysis.

DES users

The first release of `easyaccess` was on February 17th, 2015 and since then, over 300 DES Collaborators have used it to access the DES databases (Figure 1). We note that roughly 800 DES accounts exist, but this includes all database users including those that created accounts before the release of `easyaccess`. In August 2018 (version 1.4.4), we added support for the public DES data release, and since then we have increased the number of public users.
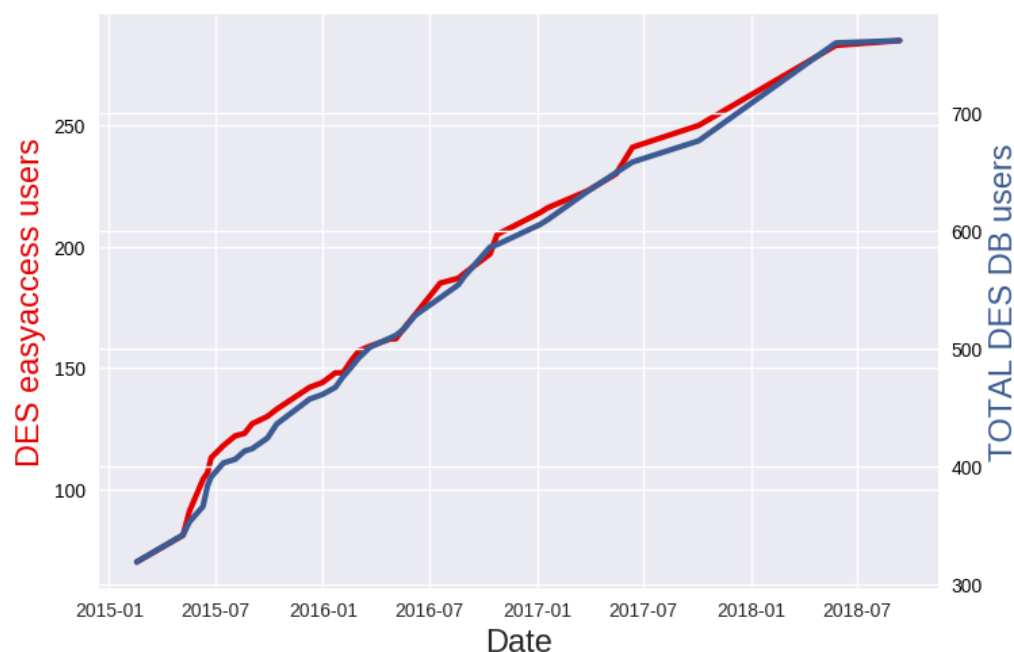
Figure 1: Number of user since first version

easyaccess

`easyaccess` is a command line interpreter that is heavily based on `termcolor` (Lepa 2018) and the [cmd](#) Python core module. It interfaces with `cx_Oracle` (Oracle Corp. 2018) to communicate with Oracle, `requests` (Reitz 2012–2018) for external URL requests, and other external open source libraries, including NumPy (Oliphant 2006), `pandas` (McKinney 2010), `fitsio` (Sheldon 2018) and `h5py` (Collette 2013) to handle and transform array data. Figure 2 shows an example of the welcome screen as seen by a DES user.

Features

`easyaccess` has a variety of features including a history of past commands and smart tab auto-completion for commands, functions, columns, users, tables, and paths. Tables can be written directly into comma-separated-value (CSV) or white-space separated text files, FITS (Wells, Greisen, and Harten 1981) files, and HDF5 (The HDF Group 1997–2018) files. It provides an iteration scheme to avoid memory constraints when retrieving large tables. Tables can also be displayed on the command line and most of the formatting is done using `pandas`. Similarly, privileged users can easily upload tables to the database from any of the file format described above in order to share data with other users. The uploading mechanism is done chunk-wise, allowing large tables to be loaded while keeping memory usage low.

In addition, there are a variety of customized functions to search and describe the tables, search for users and user tables, check quota usage, check the Oracle execution plan, and soon the ability to run asynchronous jobs through a dedicated server. There are dozens of other minor features that allow for a seamless experience while exploring and discovering data within the hundreds of tables inside the DB.

One can also load SQL queries from a file into the database, or run SQL queries inside the `easyaccess` python module in another IDE. Most of the features are also exposed

Figure 2: Welcome screenshot

through a Python API and can be run inside a Jupyter (Kluyver et al. 2016) notebook or similar tool alongside scientific analyses.

Users can also use `easyaccess` to submit and request cutouts around specific positions or objects which are generated from the images. This allows better integration with other data services for a richer scientific workflow.
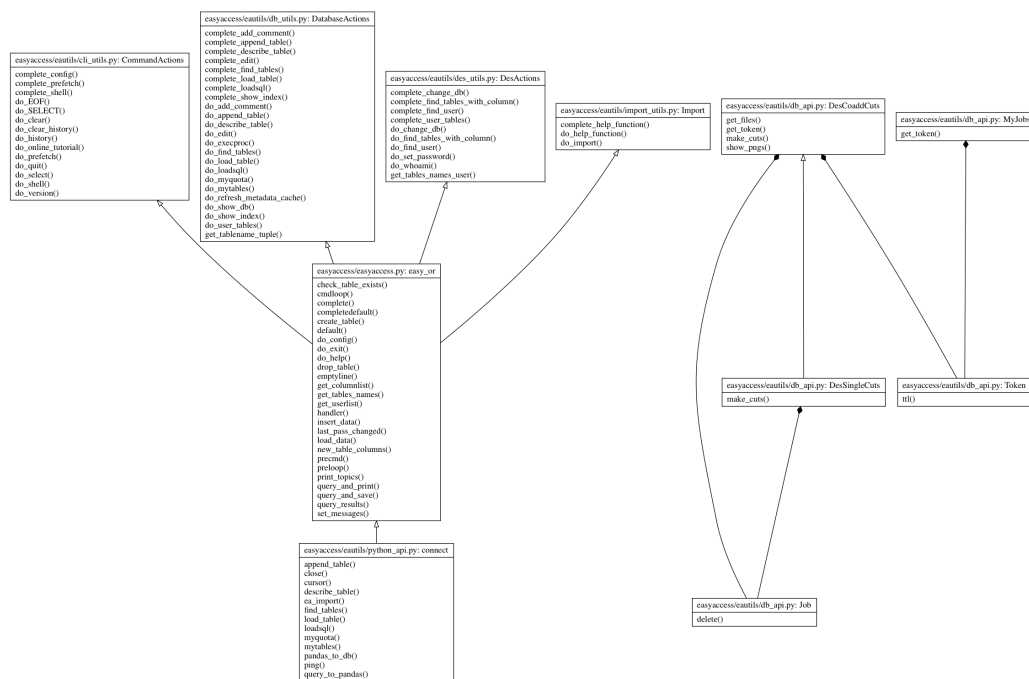
Architecture

We have included a simplified UML diagram describing the architecture and dependencies of `easyaccess` (Figure 3). Figure 3 shows only the different methods for a given class and the name of the file hosting a given class. The main class, `easy_or()`, inherits all methods from all different subclasses, making this model flexible and extendable to other surveys or databases. These methods are then converted to command line commands and functions that can be called inside `easyaccess`. Given that there are some DES specific functions, we have moved DES methods into a separate class `DesActions()`.

Installation

To download easyaccess you can clone the source code from GitHub at https://github.com/mgckind/easyaccess or follow any of the standard installation channels described below.

- From source

  ```
  python setup.py install
  ```

- conda

  ```
  conda install easyaccess -c mgckind
  ```

- Docker

Figure 3: easyaccess architecture diagram

```
docker pull mgckind/easyaccess
```

- pip

```
pip install easyaccess
```

Acknowledgments

References

Collette, Andrew. 2013. *Python and Hdf5*. O'Reilly.

DES Collaboration. 2005. "The Dark Energy Survey." *ArXiv Astrophysics E-Prints*, October.

———. 2016. "The Dark Energy Survey: more than dark energy - an overview." *MNRAS* 460 (August): 1270–99. doi:10.1093/mnras/stw641.

———. 2018. "The Dark Energy Survey Data Release 1." *ArXiv E-Prints*, January.

Kluyver, Thomas, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, et al. 2016. "Jupyter Notebooks – a Publishing Format for Reproducible Computational Workflows." Edited by F. Loizides and B. Schmidt. IOS Press.

Lepa, Konstantin. 2018. "Termcolorr." *PyPi Repository*. https://pypi.org/project/termcolor/; PyPi.

McKinney, Wes. 2010. "Data Structures for Statistical Computing in Python." In *Proceedings of the 9th Python in Science Conference*, edited by Stéfan van der Walt and Jarrod Millman, 51–56.

Oliphant, Travis E. 2006. *A Guide to Numpy*. Trelgol Publishing.

Oracle Corp. 2018. "CxOracle." *GitHub Repository*. https://github.com/oracle/python-cx_Oracle; GitHub.

Reitz, Kenneth. 2012–2018. "Requests: HTTP for Humans." http://docs.python-requests.org/.

Sheldon, Erin. 2018. "Fitsio." *GitHub Repository*. https://github.com/esheldon/fitsio; GitHub.

The HDF Group. 1997–2018. "Hierarchical Data Format, version 5."

Wells, D. C., E. W. Greisen, and R. H. Harten. 1981. "FITS - a Flexible Image Transport System" 44 (June): 363.