The Journal of Open Source Software

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

DOI: 00.00000/joss.00000

Software

- Review
- Repository
- Archive

Submitted: 00 January 0000
Published: 00 January 0000

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 gerenerally, it can easily be adapted to extend 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.

Dark Energy Survey

The Dark Energy Survey (DES) (DES Collaboration 2005; DES Collaboration et al. 2016) is an international, collaborative effort of over 500 scientists from 26 institutions in seven countries. One of the main DES the goals is to map hundreds of millions of galaxies, detect thousands of supernovae, and find patterns in the large-scale structure within the cosmic web with the objective to reveal the nature of the mysterious dark matter and dark energy that is accelerating the expansion of our Universe. Survey operations of the Southern skies began on on August 31, 2013, and will conclude early 2019. For about 500 nights, DES has been taking thousands of images of the deep sky which are transferred and processed at the National Center for Supercomputing Applications (NCSA) where immense catalogs of sources and metadata are created with hundreds of millions of entries (billions in the case of individual detections), describing all sources found within the images as well as other relevant information about the data. A significant subset of this data is, for now, only accessible to the DES collaboration but a significant sub-set was recently made public (DES Collaboration et al. 2018) and can be accessed through several mechanisms including easyaccess and web interfaces which have easyaccess running in the backend. This public release includes information for almost 400M 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 users have used it to access the DES databases within the DES collaboration as shown in Figure 1. We note that the number of DES accounts is almost 800, but this is considering

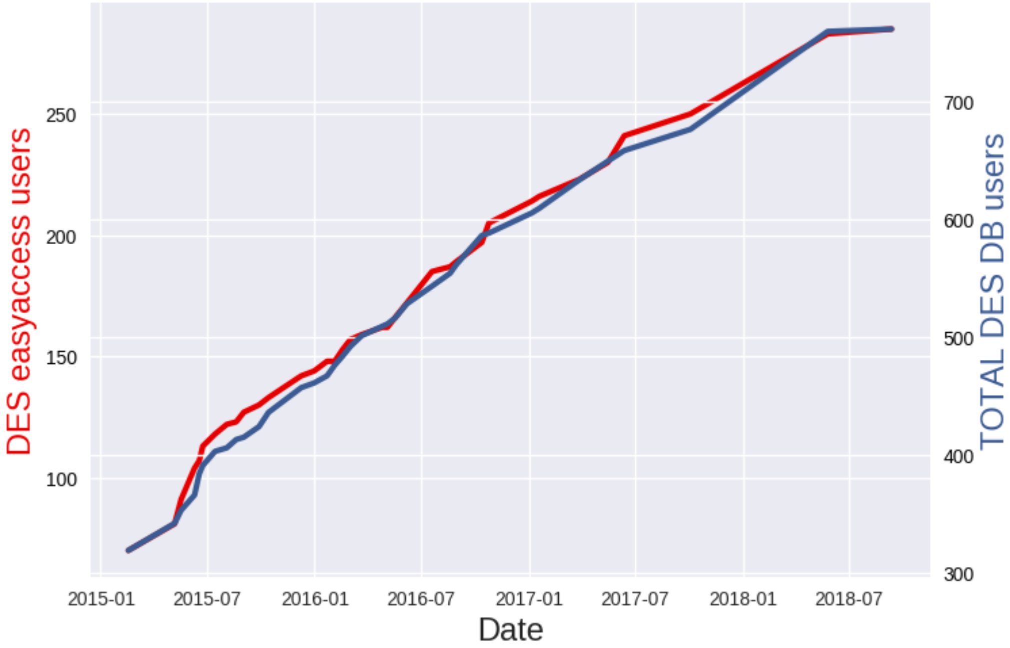


Figure 1: Number of user since first version

all users that had an account including those before the first released version. Recently in August 2018, with version 1.4.4 we added support for the public release and since then we have increased the number of public users.

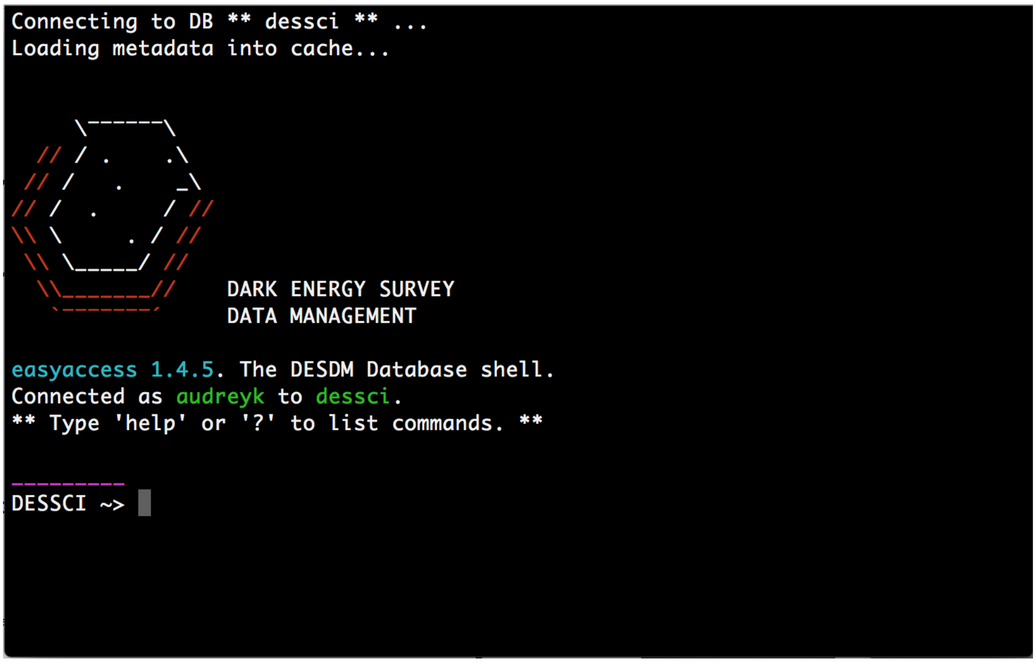
`easyaccess`

`easyaccess` is a command line interpreter which is heavily based in the `cmd` Python core module and `termcolor` (Lepa 2018), at its core as well other external and open sourced libraries including NumPy (Oliphant 2006), `pandas` (McKinney 2010), `fitsio` (Sheldon 2018) and `h5py` (Collette 2013) to handle and transform array data coming to or from the DB, `cx_Oracle` (Oracle Corp. 2018) to handle the Oracle communication and `requests` (Reitz 2012–2018) for external URL requests. Figure 2 shows an example of the welcome screen as seen as 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 as being typed. Tables can be written directly into comma-separated-values (CSV) files (or white-space separated), FITS (Wells, Greisen, and Harten 1981), and HDF5 (The HDF Group 1997–2018) files and an iterator is provided to avoid memory constraints when retrieving large tables. Tables can also be displayed on the screen and most of the formatting is done using `pandas`. Similarly, users with DB space can easily upload tables from any of the file format described above and share 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 for quota, check the Oracle execution plan, and



```
Connecting to DB ** dessci ** ...
Loading metadata into cache...


     \------\
  // / .    .\
 // /   .    _\
// /  .     / //
\\ \     . / //
 \\ \_____/ //
  \\_______//    DARK ENERGY SURVEY
   `-------'     DATA MANAGEMENT

easyaccess 1.4.5. The DESDM Database shell.
Connected as audreyk to dessci.
** Type 'help' or '?' to list commands. **

_________
DESSCI ~>
```

Figure 2: Welcome screenshot

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 through a Python API and can be run inside a Jupyter (Kluyver et al. 2016) notebook or similar tool along with the scientific code analysis.

While also using `easyaccess`, users can 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

As complement information, we have added a simplified UML diagram, in Figure 3, of the `easyaccess` architecture with dependencies. 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 follow the provided options or clone the source code from the GitHub repository at https://github.com/mgckind/easyaccess. We also provide other means to install `easyaccess` using the standard channels as described below.

- From source

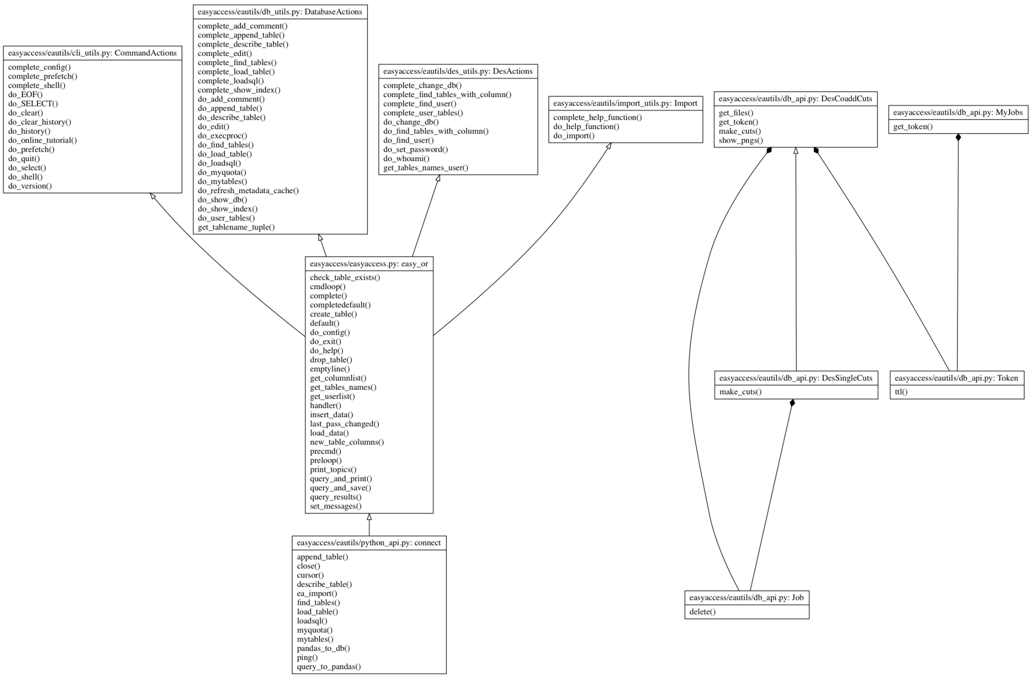


Figure 3: `easyaccess architecture diagram`

```
python setup.py install
```

- conda

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

- Docker

  ```
  docker pull mgckind/easyaccess
  ```

- pip

  ```
  pip install easyaccess
  ```

Acknowledgments

The DES Data Management System is supported by the National Science Foundation under Grant NSF AST 07-15036 and NSF AST 08-13543

References

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

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

DES Collaboration, T. M. C. Abbott, F. B. Abdalla, S. Allam, A. Amara, J. Annis, J. Asorey, et al. 2018. "The Dark Energy Survey Data Release 1." *ArXiv E-Prints*, January.

DES Collaboration, T. Abbott, F. B. Abdalla, J. Aleksić, S. Allam, A. Amara, D. Bacon, et al. 2016. "The Dark Energy Survey: more than dark energy - an overview" 460 (August): 1270–99. doi:10.1093/mnras/stw641.

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.