Tutorial: Using Command Line wget to Download ECCO Datasets from PO.DAAC

Version 1.0 2021-06-25

By Jack McNelis and Ian Fenty

Step 1: Create an account with NASA Earthdata

Please visit https://urs.earthdata.nasa.gov/home to make an account and be ready with your EOSDIS login and password.

The Earthdata Login provides a single mechanism for user registration and profile management for all EOSDIS system components (DAACs, Tools, Services). Your Earthdata login also helps the EOSDIS program better understand the usage of EOSDIS services to improve user experience through customization of tools and improvement of services. EOSDIS data are openly available to all and free of charge except where governed by international agreements.

> **Note!** *some Earthdata password characters may cause problems depending on your system*. To be safe, do not use any of the following characters in your password: backslash (\), space, hash (#), quotes (single or double), or greater than (>). Set/change your Earthdata password here: https://urs.earthdata.nasa.gov/change_password

Step 2: Set up your `netrc` and `urs_cookies` files

1. Create a file called `.netrc` in your home directory (linux, Mac):

```
/home/<username>/.netrc
```

or `_netrc` (Windows):

```
C:\Users\<username>\_netrc
```

The `netrc` file must have the following structure and must include your Earthdata account login name and password:

```
machine urs.earthdata.nasa.gov
    login <your username>
    password <your password>
```

2. Set permissions on your `netrc` file to be readable only by the current user. If not, you will receive the error "netrc access too permissive."

```
$ chmod 0600 ~/.netrc
```

3. Create an `urs_cookies` "cookie" file. This will be used to persist sessions across individual cURL/Wget calls, making it more efficient.

```
  > cd ~
> touch .urs_cookies
```

Step 3: Prepare a list of granules (files) to download

Now the only step that remains is to get a list of URLs to pass to *wget* or *curl* for downloading. There's a lot of ways to do this -- even more so for ECCO datasets data because the files/datasets follow well-structured naming conventions -- but we will rely on Earthdata Search to do this from the browser for the sake of simplicity.

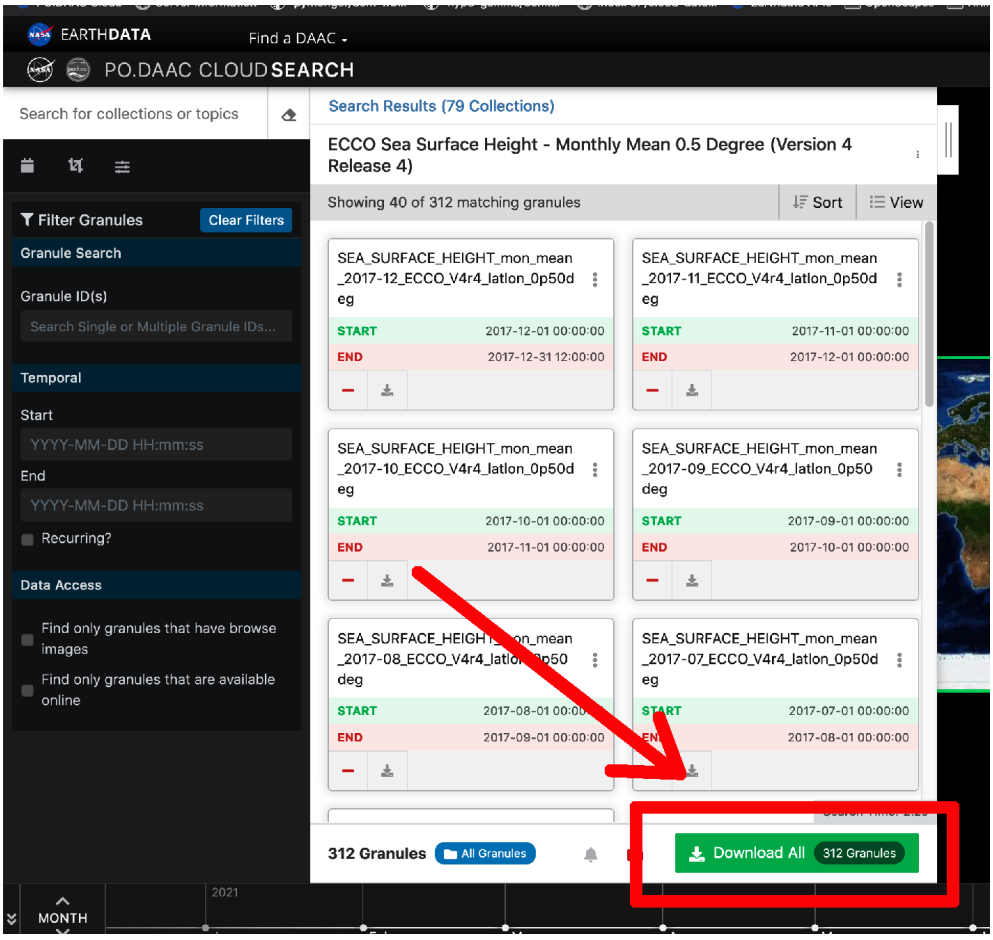
1. Find the collection/dataset of interest in Earthdata Search.

Start from this complete list of ECCO collections in Earthdata Search, and refine the results until you see your dataset of interest.

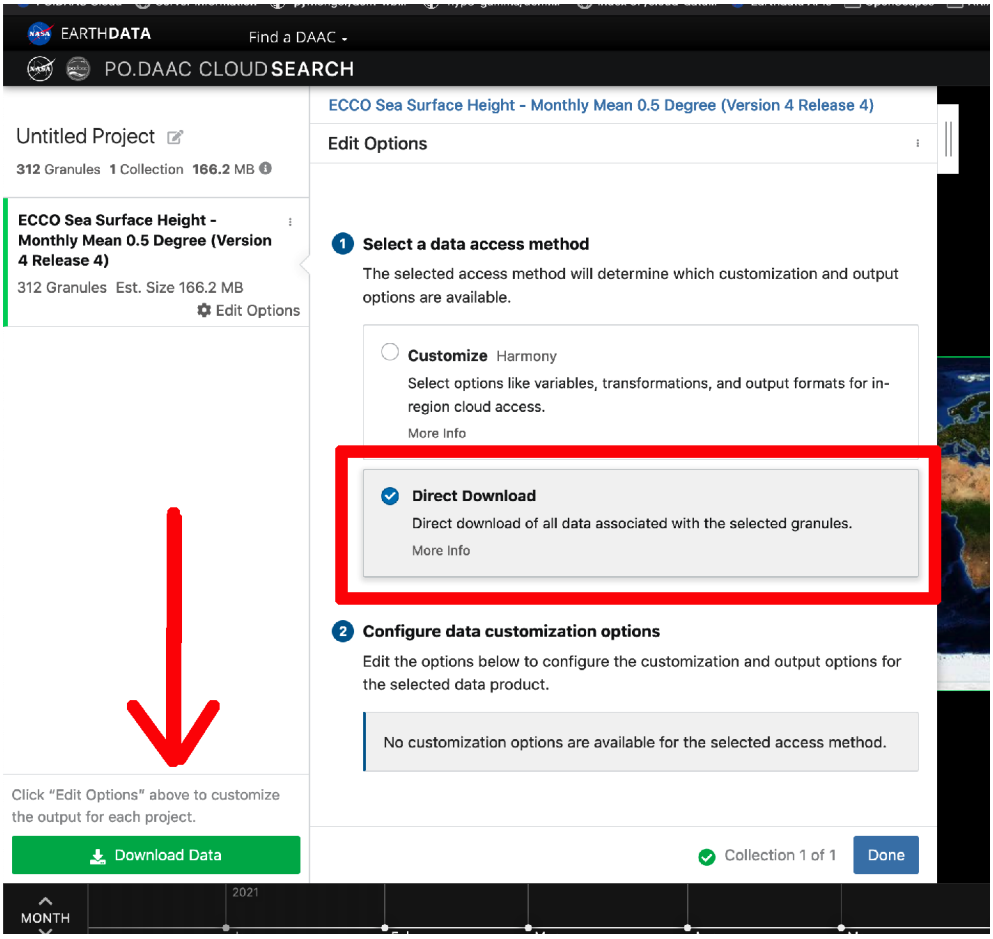
In this example we will download all of the granules for the collection ECCO Version 4 Release 4 (V4r4) monthly sea surface height on a 0.5 degree lat-lon grid.

2. Choose your collection, then click the green *Download All* button on the next page.

Click the big green button identified by the red arrow/box in the screenshot below.



That will add all the granules in the collection to your "shopping cart" and then redirect you straight there and present you with the available options for customizing the data prior to download. We will ignore those because they're mostly in active development and because we want to download all data in the collection.

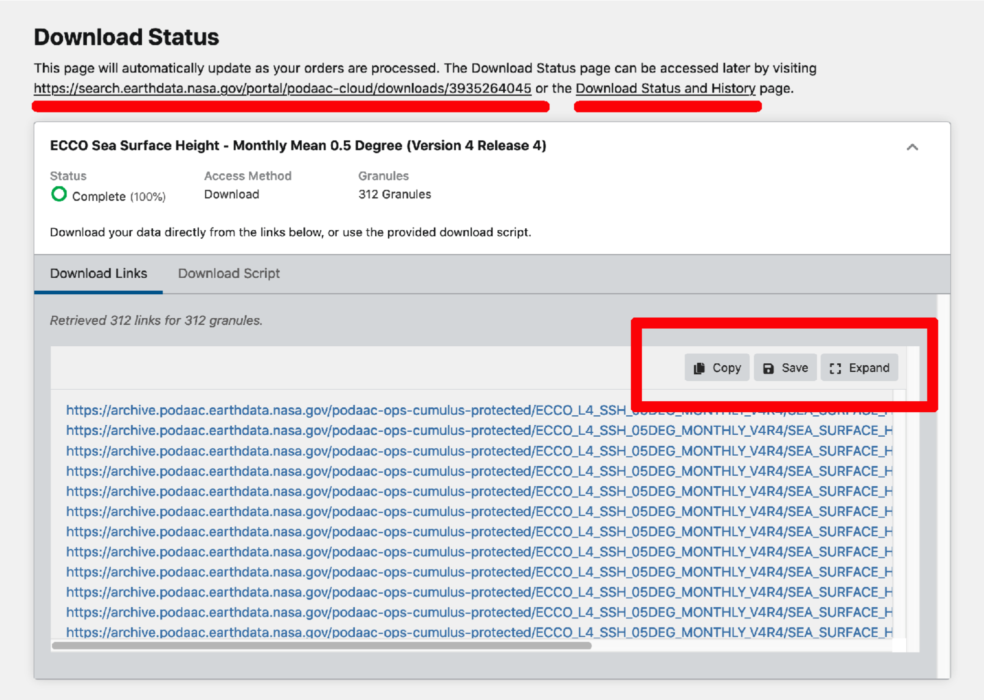


The screenshot above shows the download customization interface (i.e. "shopping cart")

3. Click *Download Data* to get your list of download urls (bottom-left, another green button)

The *Download Data* button takes you to one final page that provides the list of urls from which to download the files matching your search parameters and any

customization options that you selected in the steps that followed. This page will be retained in your User History in case you need to return to it later.



There are several ways that you could get the list of urls into a text file that's accessible from Jupyter or your local shell. I simply clicked the save button in my browser and downloaded them as a text file to a subdirectory called *resources* inside this workspace. (You could also copy them into a new notebook cell and write them to a file like we did with the netrc file above.)

Step 3: Download files in a batch with GNU *Wget*

I find *wget* options to be convenient and easy to remember. There are only a handful that I use with any regularity.

The most important wget option for our purpose is set by the `-i` argument, which takes a path to the input text file containing our download urls. Another nice feature of wget is the ability to continue downloads where you left of during a previously-interrupted download session. That option is turned on by passing the `-c` argument.

Go ahead and create a *data/* directory to keep the downloaded files, and then start the downloads into that location by including the `-P` argument:

```
mkdir -p data

wget --no-verbose \
     --no-clobber \
     --continue \
     -i 5237392644-download.txt -P data/
```