esgcet

Release 5.0.0a

May 09, 2022

Contents

CHAPTER 1

Installation

You can install esgcet one of three ways: pip, conda, or git.

1.1 Conda & Required Packages

We recommend creating a conda env before installing `esgcet`:

```
conda create -n esgf-pub -c conda-forge -c esgf-forge pip cmor autocurator esgconfigparser
conda activate esgf-pub
```

Note: the command above creates a new environment for the publisher. This is recommended rather than attempting to reuse an existing environment if you wish to upgrade a previous version of the publisher. If you installed esgcet using conda above, the cmor package (different from tables) should be installed at the time you install esgcet automatically, and having cmor in your env may cause conflicts (but not always).

1.2 esgcet install:

To install esgcet by pip run the following (note the version tag is requried):

```
pip install esgcet==5.0.0a12
```

Note: You must specify the version as the v5.0.x is under pre-release. Installing `esgcet` will install the previous major version (v3.xx).

To install esgcet into an existing environment using conda run the following:

```
conda install -c conda-forge -c esgf-forge esgcet
```

Of course, you can also just tack on `esgcet` to the list of packages in the command above when creating your new conda environment as well.

To install esgcet by cloning our github repository (useful if you want to modiy the software): first, you should ensure you have a suitable python in your environment (see above for information on conda, etc.), and then run:

```
git clone http://github.com/ESGF/esg-publisher.git -b gen-five-pkg
cd esg-publisher
cd pkg
python3 setup.py install
```

Now you will be able to call all commands in this package from any directory. A default config file, `esg.ini` will populate in `$HOME/.esg` where `$HOME` is your home directory.

NOTE: if you are intending to publish CMIP6 data, the publisher will run the PrePARE module to check all file metadata. To enable this procedure, it is necessry to download CMOR tables before the publisher will successfully run. See those pages for more info.

1.3 Config

The default config file will look like this:

```
[DEFAULT]
note = IMPORTANT: please configure below in the [user] section, that is what the publisher will use to read configured settings. The below are marked as necessary or optional variables.
version = 5.0.0a2
data_node = * necessary
index_node = * necessary
cmor_path = * necessary, and must be an absolute path (not relative)
autoc_path = autocurator * optional, default is autocurator conda binary, can be replaced with a file path, relative or absolute
data_roots = * necessary, must be in json loadable dictionary format
cert = ./cert.pem * optional, default assumes cert in current directory, override to change
test = false * optional, default assumes test is off, override to change
project = none * optional, default will be parsed from mapfile name
set_replica = false * optional, default assumes replica publication off
globus_uuid = none * optional
data_transfer_node = none * optional
pid_creds = * necessary
silent = false * optional
verbose = false * optional

[example]
data_node = esgf-data1.llnl.gov
index_node = esgf-node.llnl.gov
cmor_path = /export/user/cmor/Tables
autoc_path = autocurator
data_roots = {"/esg/data": "esgf_data"}
cert = ./cert.pem
test = false
project = CMIP6
```

(continues on next page)

(continued from previous page)

```
set_replica = true
globus_uuid = none
data_transfer_node = none
pid_creds = [{"url": "aims4.llnl.gov", "port": 7070, "vhost": "esgf-pid", "user":
↪"esgf-publisher", "password": "<password>", "ssl_enabled": true, "priority": 1}]
silent = false
verbose = false


[user]
data_node =
index_node =
cmor_path =
autoc_path = autocurator
data_roots =
cert = ./cert.pem
test = false
project = none
set_replica = false
globus_uuid = none
data_transfer_node = none
pid_creds =
silent = false
verbose = falsee
```

Fill out the necessary variables, and either leave or override the optional configurations. Note that the section the publisher reads is the `user` section, not the default nor example.

If you have an old config file from the previous iteration of the publisher, you can use `esgmigrate` to migrate over those settings to a new config file which can be read by the current publisher. See that page for more info.

1.4 Run Time Args

If you prefer to set certain things at runtime, the `esgpublish` command has several optional command line arguments which will override options set in the config file. For instance, if you use the `--cmor-tables` command line argument to set the path to the cmor tables directory, that will override anything written in the config file under `cmor_path`. More details can be found in the `esgpublish` section.

CHAPTER 2

Autocurator

2.1 Install

If you do not wish to install autocurator via conda, the option also exists to clone and install it from git:

```
git clone http://github.com/sashakames/autocurator.git
cd autocurator
make
```

After running this, there should be an autocurator executable saved as `.../autocurator/bin/autocurator`. You will need to update the config if you choose to do this with the correct path to the autocurator folder, as the default is just the `autocurator` command.

2.2 Running Autocurator

Before running `autocurator` (if you are not using the conda installed version) you must first run the following command:

```
export LD_LIBRARY_PATH=$CONDA_PREFIX/lib
```

This command helps autocurator locate and open shared libraries within the current conda environment. It will not work if this is not run. This also goes for running the `esgpublish` command if, in your config, you have listed a direct path instead of simply the autocurator command.

If you want to run `autocurator` as a stand alone, use the following format:

```
bash autocurator.sh <path to autocurator executable> <full mapfile path> <scan file name (output file)>
```

The executable itself can also be run like so:

```
bin/autocurator --out_pretty --out_json <scan file name> --files <dataset directory>
```

However, this mode is sometimes difficult as specifying multiple files requires using a `dir/*.nc` format which sometimes causes issues. Overall, we recommend using the script above as it cleans up a few things. You can also use the conda install as above, but the path/command will just be "autocurator". Once you have your scan file, you can use that to run `esgmkpubrec` (see that page for more info).

CHAPTER 3

CMOR

Before running the publisher, you will also need to obtain a directory of CMOR tables, used by PrePARE to check the metadata of your files. You can get this directory either using esgprep or by cloning the git repository.

3.1 esgprep

You can install esgprep using pip:

```
pip install esgprep
```

You can also clone their git repository and run setup.py:

```
git clone git://github.com/ESGF/esgf-prepare.git
cd esgf-prepare
python setup.py install
```

NOTE: esgprep uses python 2.6 or greater, but less than python 3.0. Configure your virtual environment as needed.

Following install, simply run:

```
esgfetchtables
```

You can specify project using --project and the output directory using --table-dir like so:

```
esgfetchtables --project CMIP6 --table-dir <path>
```

Once you have fetched the tables, you can update the cmor_path variable in your config file, or specify it at run time in the command line.

3.2 Clone Git Repository

Clone the repository:

```
git clone https://github.com/PCMDI/cmip6-cmor-tables.git
```

Your tables will be in the folder `cmip6-cmor-tables/Tables` (unless you specify a different target directory name for the clone). You can now update the `cmor_path` variable in your config file, or specify it at run time in the command line.

CHAPTER 4

esgmigrate

The esgmigrate command migrates old config settings from the old publisher into a new config file formatted for the current new publisher. The output will be found in ~/.esg/esg.ini which is the default config file path the publisher will read from.

4.1 Usage

esgmigrate is used with the following syntax:

```
esgmigrate <ini_directory_path>
```

Where <ini_directory_path> is an optional argument specifying a directory to an old esg.ini file to migrate. The default directory path is /esg/config/esgcet.

CHAPTER 5

esgpublish

The `esgpublish` command publishes a record from start to finish using the mapfile(s) passed to it. On success, it will display a success message in the output of the last two steps. If an error occurs, a helpful statement will be printed explaining which step went wrong and why.

5.1 Usage

`esgpublish` is used with the following syntax:

```
esgpublish --map <mapfile>
```

You can also use `--help` to see:

```
$ esgpublish --help
usage: esgpublish [-h] [--test] [--set-replica] [--no-replica] [--json JSON] [--data-
↪node DATA_NODE] [--index-node INDEX_NODE] [--certificate CERT] [--project PROJ]
                  [--cmor-tables CMOR_PATH] [--autocurator AUTOCURATOR_PATH] --map MAP

Publish data sets to ESGF databases.

optional arguments:
  -h, --help            show this help message and exit
  --test                PID registration will run in 'test' mode. Use this mode␣
↪unless you are performing 'production' publications.
  --set-replica         Enable replica publication.
  --no-replica          Disable replica publication.
  --json JSON           Load attributes from a JSON file in .json form. The␣
↪attributes will override any found in the DRS structure or global attributes.
  --data-node DATA_NODE
                        Specify data node.
  --index-node INDEX_NODE
                        Specify index node.
  --certificate CERT, -c CERT
```

(continues on next page)

(continued from previous page)

```
                        Use the following certificate file in .pem form for␣
↪publishing (use a myproxy login to generate).
  --project PROJ        Set/overide the project for the given mapfile, for use with␣
↪selecting the DRS or specific features, e.g. PrePARE, PID.
  --cmor-tables CMOR_PATH
                Path to CMIP6 CMOR tables for PrePARE. Required for CMIP6 only.
  --autocurator AUTOCURATOR_PATH
                        Path to autocurator repository folder.
  --map MAP             mapfile or file containing a list of mapfiles.
  --ini CFG, -i CFG     Path to config file.
```

This command can handle a singular mapfile passed to it, or a file containing a list of mapfiles (with full paths). If optional command line arguments are used, they will override anything set in the config file. NOTE: If, in your config file, you have specified a directory for `autocurator` rather than the default command, ie you are using a different `autocurator` than the one installed using conda, you must run the following command prior to running `esgpublish`:

```
export LD_LIBRARY_PATH=$CONDA_PREFIX/lib
```

If you do not run this and are not using the conda installed `autocurator`, the program will not work.

CHAPTER 6

esgmapconv

The `esgmapconv` command executes the first step of the publishing protocol by converting metadata from a mapfile into json data. That data is the input to the `esgmkpubrec` command.

6.1 Usage

`esgmapconv` is used with the following syntax:

```
esgmapconv --map <mapfile>
```

where `<mapfile>` is the absolute path to a single mapfile. The output will be printed to stdout, but can be easily redirected to a chosen file using the `--out-file` option.

You can also use the other command line options for additional configuration:

```
usage: esgmapconv [-h] [--project PROJ] --map MAP [--out-file OUT_FILE]

Publish data sets to ESGF databases.

optional arguments:
    -h, --help           show this help message and exit
    --project PROJ       Set/overide the project for the given mapfile, for use with␣
↪selecting the DRS or specific features, e.g. PrePARE, PID.
    --map MAP            Mapfile ending in .map extension, contains metadata about␣
↪the record.
    --out-file OUT_FILE  Output file for map data in JSON format. Default is printed␣
↪to standard out.
```

Using the command line option `-h` will display the above message. The above options (excluding `--map`) can be defined in the config file instead of the command line if you choose.

CHAPTER 7

esgmkpubrec

The esgmkpubrec command uses the output data from esgmapconv to populate metadata for the dataset and file records. This command also requires the output of the autocurator command, which populates additional metadata using the mapfile and puts it into a separate json file. This output is the input to the esgpidcitepub command.

7.1 Usage

esgmkpubrec is used with the following syntax:

```
esgmkpubrec --scan-file <scan file> --map-data <JSON file>
```

where <JSON file> is the aforementioned output from esgmapconv and <scan file> is the output of autocurator<https://github.com/lisi-w/autocurator>`_. The output is again defaulted to stdout, but can easily be redirected using the ``--out-file option.

The other command line options are as follows:

```
usage: esgmkpubrec [-h] [--set-replica] [--no-replica] --scan-file SCAN_FILE [--json␣
↪JSON] [--data-node DATA_NODE] [--index-node INDEX_NODE] --map-data MAP_DATA [--ini␣
↪CFG]
               [--out-file OUT_FILE]

Publish data sets to ESGF databases.

optional arguments:
    -h, --help            show this help message and exit
    --set-replica         Enable replica publication.
    --no-replica          Disable replica publication.
    --scan-file SCAN_FILE
                          JSON output file from autocurator.
    --json JSON           Load attributes from a JSON file in .json form. The␣
↪attributes will override any found in the DRS structure or global attributes.
    --data-node DATA_NODE
```

(continues on next page)

(continued from previous page)

```
                        Specify data node.
  --index-node INDEX_NODE
                        Specify index node.
  --map-data MAP_DATA   Mapfile json data converted using esgmapconv.
  --ini CFG, -i CFG     Path to config file.
  --out-file OUT_FILE   Optional output file destination. Default is stdout.
```

CHAPTER 8

esgpidcitepub

The `esgpidcitepub` command connects to a PID server using credentials defined in the config file. It then assigns a PID to the dataset. This step is necessary for all CMIP6 data records. The output of this command is the input to both the `esgupdate` command as well as the `esgindexpub` command.

8.1 Usage

`esgpidcitepub` is used with the following syntax:

```
esgpidcitepub --pub-rec <JSON file>
```

where `<JSON file>` is the output of the `esgmkpubrec` command. The output of this command is by default printed to stdout, but can easily be redirected using the `--out-file` option.

The other command line options are as follows:

```
usage: esgpidcitepub [-h] [--data-node DATA_NODE] --pub-rec JSON_DATA [--ini CFG] [--
↪out-file OUT_FILE]

Publish data sets to ESGF databases.

optional arguments:
    -h, --help            show this help message and exit
    --data-node DATA_NODE
                          Specify data node.
    --pub-rec JSON_DATA   Dataset and file json data; output from esgmkpubrec.
    --ini CFG, -i CFG     Path to config file.
    --out-file OUT_FILE   Optional output file destination. Default is stdout.
```

You can also define the above options (aside from `--pub-rec`) in the config file if you choose.

CHAPTER 9

esgupdate

The `esgupdate` command checks to see if the dataset being published is already in our database. If it is, it uses the metadata produced by the other commands to update the record. The output is the published data along with a success message upon success.

9.1 Usage

`esgupdate` is used with the follwing syntax:

```
esgupdate --pub-rec <JSON file>
```

where `<JSON file>` is the output of the `esgpidcitepub` command.

Additional command line options are as follows:

```
usage: esgupdate [-h] [--index-node INDEX_NODE] [--certificate CERT] --pub-rec JSON_
↪DATA [--ini CFG]

Publish data sets to ESGF databases.

optional arguments:
    -h, --help            show this help message and exit
    --index-node INDEX_NODE
                          Specify index node.
    --certificate CERT, -c CERT
                          Use the following certificate file in .pem form for␣
↪publishing (use a myproxy login to generate).
    --pub-rec JSON_DATA   JSON file output from esgpidcitepub or esgmkpubrec.
    --ini CFG, -i CFG     Path to config file.
```

You can also define most of these options in the config file if you choose.

CHAPTER 10

esgindexpub

The `esgindexpub` command publishes the data record using the metadata produced by the other commands to the `index_node` defined in the config file. The output of this command will display published data along with a success message upon success.

10.1 Usage

`esgindexpub` is used with the following syntax:

```
esgindexpub --pub-rec <JSON file>
```

where `<JSON file>` is the output of the `esgpidcitepub` command.

You can also use the other command line options to configure some variables outside of the config file (or to define where to find the config file):

```
usage: esgindexpub [-h] [--index-node INDEX_NODE] [--certificate CERT] --pub-rec JSON_
↪DATA [--ini CFG]

Publish data sets to ESGF databases.

optional arguments:
    -h, --help            show this help message and exit
    --index-node INDEX_NODE
                          Specify index node.
    --certificate CERT, -c CERT
                          Use the following certificate file in .pem form for␣
↪publishing (use a myproxy login to generate).
    --pub-rec JSON_DATA   JSON file output from esgpidcitepub or esgmkpubrec.
    --ini CFG, -i CFG     Path to config file.
```

Use the command line option `-h` to see the message above.

CHAPTER 11

Contributing

Please document your pull requests so we can understand how to test your changes. We don't want changes to affect publishing of ongoing projects.

11.1 Updates to this document

Please install the Sphinx package. Also you will need to *pip install sphinx-glpi-theme* in your environment.

Esgcet is a package of publisher commands for publishing to the ESGF search database.