
esgcet

Release 5.1.0b11

Jul 10, 2023

Contents

1	TL;DR	3
1.1	Publisher Introduction	3
1.2	Release Notes	4
1.3	Installation	6
1.4	Autocurator	9
1.5	CMOR	10
1.6	esgmigrate	11
1.7	esgpublish	12
1.8	esgmapconv	13
1.9	esgmkipubrec	14
1.10	esgpdcitepub	15
1.11	esgupdate	16
1.12	esgindexpub	16
1.13	esgunpublish	17
1.14	Troubleshooting & Tips	18
1.15	Contributing	19

Esgcet is a package of publisher commands for publishing to the [ESGF](#) search database.

if you have conda you can install the publisher with the following into a fresh environment, and update to the latest version:

```
conda create -n esgf-pub -c conda-forge -c esgf-forge esgcet
conda activate esgf-pub
pip install esgcet
esgpublish --version # Ensure you have upgraded to v5.2.0
esgpublish # will print the usage information.
```

You may also look at the initial `~/ .esg/esg.yaml` and update to fit your site configuration.

1.1 Publisher Introduction

The **esg-publisher** or `esgcet` Python package contains a collection of command-line utilities to scan, manipulate and push dataset metadata to an *ESGF index node*. The basic publication process includes several basic steps and sometimes *optional* steps. Publisher functionality is available via several submodules/classes in the package.

The publisher software has undergone a significant change starting with v5.* of the software. Prior versions involved storage of dataset metadata in the legacy ESGF data node PostgreSQL database and generation of *THREDDS* catalogs. The actual publication to the ESGF index occurred via catalog harvesting. Instead, the more recent publisher simplifies the process with the following phases:

1. Local scan of datasets (featuring the `autocurator` package by default)
2. Record generation using scan, mapfile and auxiliary (json) information/files as input
3. Update check of existing dataset, previous version manipulation.
4. Push/publish of record(s) to ESGF index

And several *optional* project-specific phases:

- Automatic metadata checking with PrePARE (CMIP6-only as of today)
- PID registration and citation URL generation (CMIP6 and input4MIPs)

For those familiar with the previous publisher, please be aware of the following distinctions between earlier versions and v5.*

- A Python3 conda environment is required (most prior versions have run Python2)
- the configuration (.ini) file format is new and have been vastly simplified. Note that the old format for project-specific .ini files are still used by the esgf-prepare tools (eg. esgmapfile). The v5. publisher has the ability to migrate the needed settings from the previous ini files.
- Prior invocation of esgpublish required use of `--thredds` and `--publish` stages. Those arguments are eliminated. In the general case, you can run esgpublish in a single command. Advanced users may chose to run the individual publishing steps separately to create workflows, for instance, in the use of an external workflow manager.

1.1.1 Prerequisites

- conda eg. [Miniconda](#) installation.
- Mountpoint to located data on the same host as publisher software installation, so the publisher scan utility (eg. `autocurator`) has access.
- Basic dataset information provided via the esg mapfile format. The most popular approach is using the [esgf-prepare/esgmapfile](#) utility.

1.2 Release Notes

1.2.1 v5.2.0

- Migrated configuration from .ini format to .yaml. Use `esgmigrate` to convert existing .ini files.
- Added XArray for NetCDF file reading. Disable autocurator in settings to use or add `-xarray`
- Additionally refactoring done to support the above features.

1.2.2 b5.1.0-b13

- **BUGFIX:** corrected file URL format for PID/Handle publishing (previously published URLs via v5.* were malformed).
- CMIP6 Cloned project support
- **NOTE:** this version is unavailable on Conda (`esgf-forge` channel), please use `pip install esgset` and confirm the upgrade with `esgpublish --version`.

1.2.3 b5.1.0-b11

- Updated arguments for esgunpublish
- XML archive functionality (see [Archiving Info.](#))
- bugfix for use of lower case cmip6 (should become case-insensitive)

1.2.4 b5.1.0-b10

- **CRITICAL:** esgunpublish checks dataset id argument for publication prior to unpublication to prevent server-side erroneous deletions.

1.2.5 v5.1.0-b9

- Improved Controlled-vocabulary agreement checks and upgraded rules (for CMIP6)
- Bug fix for input4MIPs (omit CMOR tables load)

1.2.6 v5.1.0-b8

- Change `set-replica` semantics with respect to PrePARE and add `force_prepare` option.
 1. Default behavior is to run PrePARE for non-replica but not for replica.
 2. With `force_prepare=True`, PrePARE is *always* run.
- esgunpublish now unpublishes PID from handle database.
- Allow for custom gridftp ports (specify with `<hostname>:<port>`).
- Correct file `instance_id` and `master_id`.

1.2.7 v5.1.0-b7

- Bug fix and refactoring: improved data root handling for paths that contain multiple instances of the project name in the path
- Bug fix for the `skip_prepare` argument (applies to CMIP6 replica publishing to bypass PrePARE)
- Feature to ensure that file tracking_ids are never duplicated within a dataset

1.2.8 v5.1.0-b6

- **CRITICAL:** corrected File record ID format to include `|data_node` to conform to prior specification
- Support for data root specifications that include the project string in the root
- Bug fixes: citation case for command line project path, support tilde for homedir in cmor path property in config file

1.2.9 v5.1.0-b5

- Update to support input4MIPs project
- Added `--version` argument
- Additional arguments for esgunpublish
- Halt publishing if a file listed in the mapfile isn't found by autocurator

1.3 Installation

1.3.1 Conda & Required Packages

We recommend creating a conda env before installing esgset

```
conda create -n esgf-pub -c conda-forge -c esgf-forge pip libnetcdf cmor autocurator_↵  
↵esgconfigparser  
conda activate esgf-pub
```

You will also need to install esgfpid using pip:

```
pip install esgfpid
```

NOTE: you will need a functioning version of autocurator in order to run the publisher, in addition to downloading the CMOR tables. See those pages for more info. The autocurator package in the esgf-forge conda channel provides a working albeit not the most recent version of this module.

1.3.2 Pip Install

Use the following command to install esgset into a previously created conda environment:

```
conda activate esgf-pub  
pip install esgset  
esgpublish --version # Ensure you have upgraded to v5.2.0
```

1.3.3 Installing esgset via git

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

```
git clone http://github.com/ESGF/esg-publisher.git  
cd esg-publisher  
git checkout refactor-esgf # NOTE this is a temporary fix prior to a merge into the_↵  
↵master branch  
cd src/python  
pip install -e . # You can modify the source in place  
esgpublish --version # check v5.2.0 has been installed
```

Now you will be able to call all commands in this package from any directory. A default config file, esg.yaml 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 necessary to download CMOR tables before the publisher will successfully run. See those pages for more info.

1.3.4 Config File (esg.yaml)

The config file will contain the following settings:

- **data_node**
 - Required. This is the ESGF node at which the data is stored that you are publishing. It will be concatenated with the dataset_id to form the full id for your dataset.

- **index_node**
 - Required. This is the ESGF node where your dataset will be published and indexed. You can then retrieve it or see related metadata by using the ESGF Search API at that index node.
- **cmor_path**
 - Required for CMIP6. This is a full absolute path to a directory containing CMOR tables, used by the publisher to run PrePARE to verify the structure of CMIP6 data. Example: /usr/local/cmip6-cmor-tables/Tables
- **autoc_path**
 - Optional. This is the path for the autocurator executable. The default assumes that you have installed it via conda. If you have not installed it via conda, please replace with a file path to your installed binary. If set to `none` or removed, the publisher will default to scanning data using XArray.
- **data_roots**
 - Required. Must be in a json string loadable by python. Maps file roots to names that appears in urls.
- **mountpoint_map**
 - Optional. Must be in yaml dictionary format. Changes specified sym link file roots in mapfile to actual file roots like so: /symlink/dir: “/actual/path”
- **cert**
 - Required, unless running in `--no-auth` mode. This is the full path to the certificate file used for publishing. Default assumes a file “cert.pem” in your current directory. Replace to override.
- **test**
 - Optional. This can be set to True or False, and it will run the esgfpid service in test mode. Default assumes False. Override if you are not doing production publishing.
- **project**
 - Optional. ESGF project to which your data belongs. Default will be parsed from the mapfile name.
- **non_netcdf**
 - Optional. Enable or disable publication settings for non NetCDF data, default assumes False.
- **set_replica**
 - Optional. Enable or disable replica publication settings. Default assumes False, or replica publication off.
- **globus_uuid**
 - Optional. Specify the UUID for your site Globus endpoint as configured in the Globus webapp. Default leaves out Globus URL from dataset metadata.
- **data_transfer_node**
 - Optional. If you run the GridFTP service, set the hostname of that node, whether it the same as your data node or a sepearte Data Transfer Node for gsiftp urls in file records. Default of “none” will omit.
- **pid_creds**
 - Settings and credentials for RabbitMQ server access for the PID sefvce, required for some projects (CMIP6, input4MIPs).
- **user_project_config**

- Optional. If using a self-defined project compatible with our generic publisher, put DRS and CONST_ATTR in a dictionary designated by project.
- **silent**
 - Optional. Enable or disable silent mode, which suppresses all INFO logging messages. Errors and messages from sub-modules are not suppressed. Default is False, silent mode disabled.
- **verbose**
 - Optional. Enable or disable verbose mode, which outputs additional DEBUG logging messages. Default is False, verbose mode disabled.
- **enable_archive**
 - Optional. Enable the writeout of dataset/file record in xml files to a local file system. (see [Archiving Info](#))
- **archive_location**
 - Optional. (Required when enable_archive = True) Path on local file system to build directory tree and write xml files for record archive.
- **archive_depth**
 - Optional. (Required when enable_archive = True) sets the directory depth of subdirectories to create/use in the xml archive. (see [Archiving Info](#))

Fill out the necessary variables, and either leave or override the optional configurations. Example config settings can be found in the default esg.ini config file which will be created at `$HOME/.esg/esg.yaml` when you install esgset. Note that while the `cmor_path` variable points to a directory, other filepaths must be complete, such as `autoc_path` and `cert`. This applies to the command line arguments for these as well. Additionally, a *required* setting if omitted can be satisfied via inclusion as ccommand line arguments.

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.

1.3.5 Project Configuration

You may define a custom project in several ways. First, using the `user_project_config` setting, specify an alternate *DRS* and constant attribute values (`CONST_ATTR`) for your project. *DRS* is followed an array with the components. *version* is *always* the ultimate component of the dataset.

If your project desires to use the features of CMIP6 included extracted Global Attributes use the `cmip6_clone` config file property and assign to your custom project name within the `user_project_config`. The project name must be overridden using `CONST_ATTR project` setting (see example below). If you CMIP6 project wishes to register PIDs, you must assign a `pid_prefix` within config settings.

Example Config

The following contains example `.yaml` code and configures the *primavera* project as a user-defined *cloned* project:

```
autoc_path: autocurator
cmip6_clone: primavera
cmor_path: /path/to/cmip6-cmor-tables/Tables
data_node: esgf-fake-test.llnl.gov
data_roots:
  /Users/ames4/datatree: data
data_transfer_node: aimsdtn2.llnl.gov
```

(continues on next page)

(continued from previous page)

```

force_prepare: 'false'
globus_uuid: 415a6320-e49c-11e5-9798-22000b9da45e
index_node: esgf-fedtest.llnl.gov
pid_creds:
  aims4.llnl.gov:
    password: password
    port: 7070
    priority: 1
    ssl_enabled: true
    user: esgf-publisher
    vhost: esgf-pid
project: none
set_replica: 'true'
silent: 'false'
skip_prepare: 'true'
test: 'true'
cmip_clone: primavera
user_project_config:
  primavera:
    CONST_ATTR:
      project: primavera
      pid_prefix: '21.14100'
verbose: 'false'

```

1.3.6 Run Time Args

If you prefer to set your configuration to publish 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`.

If you used the old (v4 or earlier) version of the publisher, you should note that the command line argument `--config` which points to your config file must be a complete path, not the directory as it was in the previous version. More details can be found in the `esgpublish` section. Some settings are not available on the command line and must be placed in the config file, such as the xml “archive” utility.

1.4 Autocurator

`Autocurator` is an optional tool for scanning data. In some test cases it has shown to be faster than the Python-centric approach of using `Xarray`. In the default workflow `esgpublish` uses a subprocess to call the executable over each input file then open its output in `.json` format. Additionally it can be called in custom workflows using the individual CLI publishing modules.

1.4.1 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.

1.4.2 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).

1.5 CMOR

Before running the publisher for CMIP6, you will 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.

1.5.1 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.

1.5.2 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.

1.6 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 `$HOME/.esg/esg.yaml` which is the default config file path the publisher will read from.

1.6.1 Usage

`esgmigrate` is used with the following syntax:

```
esgmigrate
```

By default, `esgmigrate` will attempt to read the old config file at `/esg/config/esgcet` and will write the new config file to `$HOME/.esg/esg.yaml`. To override these defaults, use the optional command line arguments below.

Additional command line options are as follows:

```
usage: esgmigrate [-h] [--old-config CFG] [--silent] [--verbose]
                [--project PROJECT] [--destination DEST]

Migrate old config settings into new format.

optional arguments:
  -h, --help            show this help message and exit
  --old-config CFG      Full path to old config file to migrate.
  --silent              Enable silent mode.
  --verbose             Enable verbose mode.
  --project PROJECT     Name of a particular legacy project to migrate.
  --destination DEST    Destination for new config file.
```

Note that `--old-config` should point to a directory, not the file itself; however, `--destination` should be a complete file path including the file name.

1.7 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.

1.7.1 Usage

`esgpublish` is used with the following syntax:

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

The mapfile (`--map`) is the only truly *required* argument, as other are typically supplied through the config file. You can also use `--help` to see:

```
$ esgpublish --help
usage: esgpublish [-h] [--test] [--set-replica] [--no-replica] [--esgmigrate]
                  [--json JSON] [--data-node DATA_NODE]
                  [--index-node INDEX_NODE] [--certificate CERT]
                  [--project PROJ] [--cmor-tables CMOR_PATH]
                  [--autocurator AUTOCURATOR_PATH] --map MAP [MAP ...]
                  [--config CFG] [--silent] [--verbose] [--no-auth] [--verify]
                  [--version] [--xarray]
```

Publish data sets to ESGF databases.

options:

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
                          Use the following certificate file in .pem form for
↳ publishing (use a myproxy login to generate).
--project PROJ            Set/override 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                Required. mapfile or file containing a list of mapfiles.
--ini CFG, -i CFG        Path to config file.
--silent                 Enable silent mode.
--verbose                 Enable verbose mode.
```

(continues on next page)

(continued from previous page)

<code>--no-auth</code>	Run publisher without certificate, only works on certain_
<code>↪index nodes.</code>	
<code>--verify</code>	Toggle verification for publishing, default is off.
<code>--xarray</code>	Use Xarray to extract metadata even if Autocurator is_
<code>↪configured.</code>	

This command can handle a singular mapfile passed to it, a file containing a list of mapfiles (with full paths), a directory of mapfiles, or a directory of lists of mapfiles. You do not need to specify how you are passing mapfiles, but all of them must be for the same project in order for them to be published with the correct metadata. 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.

Note: Using the `--xarray` argument will override `autocurator` whether specified in the config file or the `--autocurator` argument.

Warning: Please do not attempt to run *esg-publisher* commands with a legacy `esg.ini` file using the `-i` argument. You will need to migrate the config using *esgmigrate*.

1.7.2 Archiving Info

Dataset records (metadata) can be preserved in xml form for future use if the need arises to rebuild an index. (This functionality replaces the ability to reharvest THREDDS catalog that was available with the prior ESGF/publisher architecture). XML files are created for both the dataset and every file record: one file per each record, eg. if there are *two* files for a dataset, *three* xml files are generated in total. There are three config file options that must be set in order to enable the archive:

- **enable_archive**
 - Set to True to enable the feature
- **archive_location**
 - Path on local file system to build directory tree and write xml files for record archive.
- **archive_depth**
 - Controls the directory depth of subdirectories to create/use in the xml archive

The `esgindexpub` subcommand has the `--xml-list` option. Supply a file containing a list of paths to xml files within the archive in order to push the recods to the index node.

1.8 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.

1.8.1 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] [--config CFG]

Publish data sets to ESGF databases.

optional arguments:
  -h, --help            show this help message and exit
  --project PROJ        Set/override 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.
  --config CFG, -cfg CFG Path to config file.
```

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.

1.9 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.

1.9.1 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] [--json JSON]
                  --scan-file SCAN_FILE --map-data MAP_DATA
                  [--out-file OUT_FILE] [--data-node DATA_NODE]
                  [--index-node INDEX_NODE] [--project PROJ]
                  [--config CFG] [--silent] [--verbose]
```

Publish data sets to ESGF databases.

(continues on next page)

(continued from previous page)

```

optional arguments:
  -h, --help                show this help message and exit
  --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.
  --scan-file SCAN_FILE    JSON output file from autocurator.
  --map-data MAP_DATA       Mapfile json data converted using esgmapconv.
  --out-file OUT_FILE       Optional output file destination. Default is stdout.
  --data-node DATA_NODE    Specify data node.
  --index-node INDEX_NODE   Specify index node.
  --project PROJ            Set/override the project for the given mapfile, for use with
  ↪ selecting the DRS or specific features, e.g. PrePARE, PID.
  --config CFG, -cfg CFG    Path to config file.
  --silent                  Enable silent mode.
  --verbose                 Enable verbose mode.

```

NOTE: esgmkpubrec has customized settings and features depending on the project. If the project is undefined, it will use default settings which may not work for your project and could result in errors. It is highly recommended to specify your project, and also use the config file to specify if it is non-netcdf data.

1.10 esgpidcitemapub

The esgpidcitemapub 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.

1.10.1 Usage

esgpidcitemapub is used with the following syntax:

```
esgpidcitemapub --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: esgpidcitemapub [-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.

```

(continues on next page)

(continued from previous page)

```
--config CFG, -cfg 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.

1.11 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.

1.11.1 Usage

`esgupdate` is used with the following 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 [--config CFG] [--silent]  
               [--verbose] [--no-auth] [--verify]  
  
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.  
  --config CFG, -cfg CFG     Path to config file.  
  --silent                  Enable silent mode.  
  --verbose                 Enable verbose mode.  
  --no-auth                 Run publisher without certificate, only works on certain   
↪ index nodes.  
  --verify                  Toggle verification for publishing, default is off.
```

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

1.12 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.

1.12.1 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 [--config CFG] [--silent]
                  [--verbose] [--no-auth] [--verify]

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 esgmkipubrec.
  --config CFG, -cfg CFG     Path to config file.
  --silent                  Enable silent mode.
  --verbose                  Enable verbose mode.
  --no-auth                  Run publisher without certificate, only works on certain
  →index nodes.
  --verify                  Toggle verification for publishing, default is off.
  --xml-list                 Publish directly from xml files listed (supply a file
  →containing paths to the files).
```

Use the command line option `-h` to see the message above. Note that the `--xml-list` option is intended to be used following the use of the “enable_archive” setting and the presence of “archived” publication records in xml format (see [Archiving Info](#)). Before use of the `esgindexpub` command in this context, create a list of these files to supply to the command.

1.13 esgunpublish

The `esgunpublish` command retracts, or, upon specification, deletes a specified dataset(s). The output of this command is either a success or failure message accompanied with the id of the dataset that was retracted. Exercise caution when deleting datasets as, if replicas have been made or if you will be republishing, you should retract rather than delete outright. There are three input methods for specifying input dataset(s).

1.13.1 Usage

For a single dataset `esgunpublish` is used with the following syntax:

```
esgunpublish --dset-id <dataset_id>
```

The <dataset_id> can be either the `instance_id` or the full `dataset_id` corresponding to the dataset. If `instance_id` is used, the program will use the `data-node` option, from CLI or config file, to create the full `dataset_id`.

For multiple datasets there are two additional options. **Option 1:** use a list in a text file with `--use-list`.

```
esgunpublish --use-list /path/to/textfile
```

Option 2: Specify the mapfile or a path to a directory containing mapfile(s). A datanode must be specified as mapfiles don't contain the datanode in the dataset id:

```
esgunpublish --map /path/to/mapfiles
```

esgunpublish supports the following command line arguments:

```
usage: esgunpublish [-h] [--index-node INDEX_NODE] [--data-node DATA_NODE]
                  [--certificate CERT] [--delete] [--dset-id DSET_ID]
                  [--map MAP [MAP ...]] [--use-list DSET_LIST] [--ini CFG]
                  [--version] [--no-auth] [--silent] [--verbose]

Unpublish data sets from ESGF databases.

optional arguments:
  -h, --help                show this help message and exit
  --index-node INDEX_NODE    Specify index node.
  --data-node DATA_NODE     Specify data node.
  --certificate CERT, -c CERT
                             Use the following certificate file in .pem form for
                             unpublishing (use a myproxy login to generate).
  --delete                  Specify deletion of dataset (default is retraction).
  --dset-id DSET_ID         Dataset ID for dataset to be retracted or deleted.
  --config CFG, -cfg CFG    Path to config file.
  --map MAP [MAP ...]       Path(s) to a mapfile or directory(s) containing
                             mapfiles.
  --use-list DSET_LIST      Path to a file containing list of dataset_ids.
  --version                 Print the version and exit
  --no-auth                 Run publisher without certificate, only works on
                             certain index nodes.
  --silent                  Enable silent mode.
  --verbose                 Enable verbose mode.
```

You can see this message above by running `esgunpublish -h`. For the `--ini`, `-i` option, the path may be relative but it must point to the file, not to the directory in which the config file is.

1.14 Troubleshooting & Tips

If you encounter issues running any of the esgctet commands, try looking for common issues:

- If you encounter issues processing arguments (variables are undefined but you included them either in the command line or ini file), try checking your ini file for syntax issues. The error messages should be clear for the most part, but for variable issues the config file is a good place to start.
- If the program fails to create the dataset, check to see if autocurator exited without error.
- If you are using a custom project and encounter errors, try using the individual commands one at a time instead of `esgpublish`. If your project requires customization, feel free to open a github issue and request that support for your project is added.
- For example commands and test scripts, see our [test suite repository](#).

- For unexpected behavior, output, or errors, please open a [github issue](#).

1.15 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.

1.15.1 Updates to this document

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