
mesh_part_prep Documentation

Release 0.1.0

Paul Gierz

Oct 11, 2020

Contents

1	Preperation Plugin for FESOM Mesh Partitioning	1
1.1	Installation	1
1.2	Usage	1
2	Command Line Interface	3
3	ESM Tools Plugin	5
4	Python Library Usage	7
5	Indices and tables	9
	Index	11

Preperation Plugin for FESOM Mesh Partitioning

This plugin contains functionality to prepare files for the mesh partioner after the Triangle program has run.

1.1 Installation

To install the plugin, use:

```
pip install --user git+https://github.com/esm-tools-plugins/mesh_part_prep
```

1.2 Usage

From the command line:

```
$ mesh_part_prep <PATH_TO_MESH_FOLDER>
```

From an `esm_tools` script (as a plugin, which you can include in your recipe as `mesh_part_prep` and `mesh_part_finish`):

```
general:
    # Turn on the plugin (seperate from actually loading it):
    mesh_part_prep: True
    # Turn on copying the mesh to a desired location after completion
    mesh_part_finish: True

fesom_mesh_part:
    # Specify the raw mesh directory:
    mesh_dir: /some/path/to/mesh/dir
```

(continues on next page)

(continued from previous page)

```
# Where the finished, partitioned mesh should be placed:  
result_mesh_dir: /some/path/where/the/finished/mesh/should/be
```

Command Line Interface

The command line interface for `mesh_part_prep` allows you to run preparatory steps needed to transform the output files generated by the [Triangle](#) program into [METIS](#) compatible input files.

After installation, you can use:

```
mesh_part_prep --help

usage: mesh_part_prep [-h] [-l {DEBUG,INFO,WARNING,ERROR,CRITICAL}] mesh_dir

positional arguments:
  mesh_dir              Path to the unpartitioned mesh directory

optional arguments:
  -h, --help            show this help message and exit
  -l {DEBUG,INFO,WARNING,ERROR,CRITICAL}, --log {DEBUG,INFO,WARNING,ERROR,CRITICAL}
                        Set the logging level
```

The only required argument is the path to the mesh:

```
mesh_part_prep /some/path/goes/here
```

In this case, the level information is prepended to the `aux3d.out` file.

CHAPTER 3

ESM Tools Plugin

There is also a esm-tools plugin to prepare files for the FESOM Mesh Generator.

This consists of two parts, first to prepare the `aux3d.out` file, and secondly to move the resulting mesh files into a common directory.

In your recipe, you can include the following elements after the installation:

```
1 compute:
2   recipe:
3     - "mesh_part_prep"
4     - "_create_setup_folders"
5     - "_create_component_folders"
6     - "initialize_experiment_logfile"
7     - "_write_finalized_config"
8     - "copy_tools_to_thisrun"
9     - "_copy_preliminary_files_from_experiment_to_thisrun"
10    - "_show_simulation_info"
11    - "copy_files_to_thisrun"
12    - "modify_namelists"
13    - "modify_files"
14    - "create_new_files"
15    - "prepare_coupler_files"
16    - "add_batch_hostfile"
17    - "copy_files_to_work"
18    - "write_simple_runscript"
19    - "report_missing_files"
20    - "database_entry"
21    - "submit"
22    - "mesh_part_finish"
```

Your configuration should include the following to set up the plugin correctly:

```
general:
  mesh_part_prep: True
  mesh_part_finish: True
```

(continues on next page)

(continued from previous page)

```
fesom_mesh_part:
  mesh_dir: /path/to/your/Triangle/output/dir
  result_mesh_dir: /some/path
  part: 288 # For example, you could set a different partitioning here
```

CHAPTER 4

Python Library Usage

The Python module allows you to prepare a Triangle output for use with the mesh partitioner.

A minimal example:

```
>>> from mesh_part_prep import RawMesh
>>> path = "/some/path/with/a/triangle/output"
>>> rm = RawMesh(path)
>>> rm.process()
```


CHAPTER 5

Indices and tables

- `genindex`
- `modindex`
- `search`

M

`mesh_part_prep.cli` (*module*), [3](#)

`mesh_part_prep.mesh_part_prep` (*module*), [7](#)

`mesh_part_prep.plugin` (*module*), [5](#)