Guide: Using PGC GitHub: pgcdemtools
URL: https://www.pgc.umn.edu/guides/pgc-coding-and-utilities/using-pgc-github-pgcdemtools/
Last Modified: December 6, 2019
Export Date: June 14, 2026

Introduction

In this guide, you will learn what software you need to run the pgcdemtools scripts, where to access the required software, and how to use the multiple scripts.

The PGC hosts several open source codes on GitHub, a company that hosts software development. Projects, source codes, changes and versions can be accessed in PGC's Repositories online.

Pgcdemtools is a package of tools used to index, shelve, copy, and modify SETSM and ASP Digital Elevation Models (DEM). There are 11 utilities bundled together for a single download on the Polar Geospatial Center's GitHub page. This guide will provide step-by-step instructions on how to access and use the scripts within the pgcdemtools package.

Quick Links

- Polar Geospatial Center GitHub Homepage
- Polar Geospatial Center GitHub Pgcdemtools
- Mac and Linux: Installation of PGC's GDAL Stack
- Windows: OSGeo4W Installation

Requirements

- Personal computing environment (desktop computer) or cluster computer environment
- Linux HPC cluster running Maui/Torque for queue management (preferred) or Windows platform

> Please note that changes made to scripts may not be reflected in this guide and the code is tightly coupled to the systems on which it was developed. You should have no expectation of running it on another system without some patching.

Software

This tool is built on the GDAL/OGR image processing API using Python. GDAL 2.1 is required for some these tools to function.

If you are using a Linux system you will need to download the PGC optimized GDAL toolchain. The list of software installed with the optimized GDAL toolchain can be found here. A script is provided to install all required packages. If you have not ran a shell script in a Linux terminal follow this guide here. After installation of the PGC optimized GDAL toolchain, the tools can be ran through the Linux terminal. There are plenty of free, online tutorials for Linux terminal if you are new to command line interfaces.

If you are using a Windows system, it is recommended that you use OSGeo4W. This will provide a Windows environment to use the tool. You can get the installers here. The express installation will provide the most high profile OSGeo4W packages. However, it will not allow for control over install location, proxies, and cache directory selection. The advanced install will allow for more control. The DEM tools will run with either install type. After installation of OSGeo4W, the tools can be ran through the OSGeo4W Shell. As with Linux, there are numerous online

resources for using a Windows command line interface.

Bundle Details

The pgcdemtools bundle runs multiple processes or submits them to a PBS or SLURM HPC cluster for processing. The download page can be found here. This bundle contains 11 scripts:

Indexing:

1. index_setsm.py – Builds a geodatabase (gdb) or shapefile (shp) index of a single SETSM DEM or a directory of DEMs.
2. index_setsm_tiles.py – Builds a geodatabase (gdb) or shapefile (shp) index of a single SETSM DEM mosaic tile or a directory of tiles.

Packaging:

3. package_setsm.py – Builds an index of a SETSM DEM or directory of DEMs and packages all auxiliary files into a tar.gz archive.
4. package_setsm_tiles.py – Builds an index of a SETSM DEM mosaic tiles or directory of tiles and packages all auxiliary files into a tar.gz archive.

Shelving:

5. shelve_setsm_by_date.py – Move or copy a SETSM DEM or a directory of DEMs into folders based on acquisition date.
6. shelve_setsm_by_geocell.py – Move or copy a SETSM DEM or a directory of DEMS into folders based on the geocell that intersects the DEM centroid. Identified by the lower left geocell center.
7. shelve_setsm_by_shp.py – Move or copy a SETSM DEM or a directory of DEMs into folders based on a custom shapefile index.

Retrieval:

8. copy_dems.py – Copy DEMs using a subset of the DEM index built using the index_setsm.py utility.

Miscellaneous:

9. apply_setsm_registration.py – If GCP registration information is included with a SETSM DEM, this utility applies the offset and outputs a new raster.
10. resample_setsm.py – Resamples the SETSM DEMs to a lower resolution.
11. divide_setsm_tiles.py – Divide SETSM DEM mosaic tiles into subtiles.

Submission scripts to PBS and SLURM can be found at the PGC GitHub page. Including the command **“- -pbs”** will submit the task to PBS, and including the command **“- -slurm”** will submit the task to SLURM. When submitting a job to a cluster where there is storage local to the processing node it is recommended to include the **“- -wd”** command. This will allow you to set a local working directory which will allow for increased processing time. The parallel processing option **(- -parallel-processes)**, if available, allows the given tool to operate on several tasks at once.

Information regarding common commands are detailed in the sample workflow below. Additionally, the script description can be viewed in a command terminal by using the **“-h”** or **“- -help”** command.
For example: `C:\>python user\pathname\pgcdemtools\resample_setsm.py -h`

Script Details

Before you begin you will need to gather all your image files and place them in a single folder. Note for these commands that Linux users will have forward slashes (/) in the pathname, and pathnames will begin with "/mnt/" instead of using the drive name as in Windows (D:\).

All scripts are accessed through typing "Python" into a command terminal followed by the pathname for the script. This can be accomplished by dragging and dropping the script into the command terminal.

This section will work through commands for each script:

index_setsm.py

This script builds a geodatabase (gdb) or shapefile (shp) index of a single SETSM DEM or a directory of DEMs.

```
<C:\>python user\pathname\pgcdemtools\index_setsm.py --epsg 3031 --log pathname\log_file_destination --overwritepathname\source_dem_location pathname\output_destination
```

Above is an example of all available commands within this script. The EPSG is denoted with the **"- -epsg"** command followed by the EPSG number. The **"- -log"** command specifies the output location for a log file. The **"- -overwrite"** command should only be used if you are looking to overwrite an existing index. The last two inputs into the index_setsm.py tool are the source directory for the DEMs and the output location for the index. Include the file type extension with the output destination.

index_setsm_tiles.py

This script builds a geodatabase (gdb) or shapefile (shp) index of a single SETSM DEM mosaic tile or a directory of tiles. This has the same commands as the index_setsm.py script discussed above. An example of proper command syntax is below:

```
C:\>python user\pathname\pgcdemtools\index_setsm_tiles.py --epsg 3031 --log pathname\log_file_destination --overwrite pathname\source_dem_location pathname\output_destination
```

package_setsm.py

This tool builds an index of a SETSM DEM or directory of DEMs and packages all auxiliary files into a tar.gz archive.

```
C:\>python user\pathname\pgcdemtools\package_setsm.py --mdf-only --lsf --filter-dems --force-filter-dems -v --overwrite --tasks-per-job 10 pathname\input_directory pathname\scratch_space
```

Above is an example of the available commands for this tool. Note that you will not need to use each command at the same time. The **"- -mdf-only"** builds only mdf and readme files. To package the LSF DEM instead of the original DEM use the **"- -lsf"** command. This also includes the metadata flag. Including the **"- -filter-dems"** command will filter DEMs with area less than 5.6 square kilometers and a density less than 0.1. The **"- -force-filter-dems"** will filter DEMs where the tar has already been built. It will use the same criteria as the **"- -filter-dems"** command (with area less than 5.6 square kilometers and a density less than 0.1). The **"-v"** command denotes a verbose output. If the job is being submitted to pbs, there is an option to specify the number of tasks to bundle into a single job. The **"- -tasks-per-job"** command followed by the number of jobs will express how many jobs should be bundled. This command requires the **"- -pbs"** command to submit the job to pbs. Including the **"- -**

dryrun” command will print the actions of the tool without executing.

package_setsm_tiles.py

This script builds an index of a SETSM DEM mosaic tiles or directory of tiles and packages all auxiliary files into a tar.gz archive.

```
C:\>python user\pathname\pgcdemtools\package_setsm_tiles.py --log pathname\log --epsg 3031 -v --overwrite pathname\input_dem_tiles pathname\scratch_folder
```

An example of available commands is seen above. The **“- -log”** command specifies the output location for a log file. An EPSG is denoted with the **“- -epsg”** command followed by the EPSG number. The **“-v”** command denotes a verbose output. The **“- -overwrite”** command should only be used if you are looking to overwrite an existing index. The last two inputs into the package_setsm_tiles.py tool are the source directory for the DEM tiles and the output location for the tar.gz archive.

shelve_setsm_by_date.py

This tool moves or copies a SETSM DEM or a directory of DEMs into folders based on acquisition date. The file structure is ///. Below is an example of available commands:

```
C:\>python user\pathname\pgcdemtools\shelve_setsm_by_date.py --try-link --log pathname\log_file --overwrite pathname\input_directory pathname\output_directory
```

To try linking files instead of copying them, include the **“- -try-link”** command. The **“- -log”** command followed by a pathname will output a log file to that location. If you are looking to overwrite the existing index, include the **“- -overwrite”** command. Lastly, include the pathname to the input directory of DEM along with an output destination directory.

shelve_setsm_by_geocell.py

This utility moves or copies a SETSM DEM or a directory of DEMS into folders based on the geocell that intersects the DEM centroid. Identified by the lower left geocell center. Below is an example of available commands.

```
C:\>python user\pathname\pgcdemtools\shelve_setsm_by_geocell.py --res {2m,8m} --try-link --log pathname\log_file --overwrite pathname\input_directory pathname\output_directory
```

If you are looking to shelve DEMs of a certain resolution, include the **“- -res”** command followed by the resolution. The example above shelves DEMs with a resolution of 2m or 8m. To try linking files instead of copying them, include the **“- -try-link”** command. The **“- -log”** command followed by a pathname will output a log file to that location. If you are looking to overwrite the existing index, include the **“- -overwrite”** command. Lastly, include the pathname to the input directory of DEM along with an output destination directory.

shelve_setsm_by_shp.py

This tool will move or copy a SETSM DEM or a directory of DEMs into folders based on a custom shapefile index. The files will be shelved based on the centroid location relative to the shapefile index. Below is an example of available commands:

```
C:\>python user\pathname\pgcdemtools\shelve_setsm_by_shp.py --try-link --log pathname\log_file --overwrite pathname\input_directory pathname\output_directory
```

```
pathname\shapefile_index shapefile_field_name
```

To try linking files instead of copying them, include the **“- -try-link”** command. The **“- -log”** command followed by a pathname will output a log file to that location. If you are looking to overwrite the existing index, include the **“- -overwrite”** command. The source directory pathname for the DEMs should be included followed by the pathname for the desired destination directory. Lastly, the pathname to the shapefile index defining the grid scheme is included followed by the field in the shapefile index that contains the grid name.

copy_dems.py

This script can be used to move or copy ASP deliverable files. Below is an example of available commands:

```
C:\>python user\pathname\pgcdemtools\copy_dems.py -m --exclude-drg --dems-only --no-dirs --tar-only --exclude-err --include-pc --include-fltr --include-logs pathname\input_directory pathname\output_directory
```

If you are looking to move DEMs instead of copy, include the **“-m”** command. To exclude copying or moving the DRG/Ortho files, insert the “**“- -exclude-drg”** command. Using the **“- -dems-only”** command will copy or move only the DEMs. This command will override the other exclude and include options, except for the “**“–include-fltr”** command. By default this tool will create pairname subdirectories for overlaps. Including the “**“- -no-dirs”** command will stop the creation of the subdirectories. The **“- -tar-only”** command will copy or move only the tar archive in the source directory. This command will override the other exclude or include command options.

The copy_dems.py script offers four other exclude/include options for ASP created DEMs. First it the “**“- -exclude-err”** command. This will exclude the copying or moving of the intersection Err raster. The **“- -include-pc”** command will include the copying or moving of the point cloud generated during the ASP process. The **“- -include-fltr”** command will include the non-interpolated DEM. Lastly, the **“- -include-logs”** command will include the stereo logs with the copying or moving.

The final two commands to include are for the input and output directories. The input can be the source directory of the DEMs or a shapefile of the ASP DEMs.

apply_setsm_registration.py

This script will apply the offset and outputs a new raster if there is GCP registration information included with the SETSM DEM. The strip ArcticDEM files contain IceSAT altimetry offset information within the metadata, and this tools allows for the adjustment of SETSM DEMs with that information. Below is an example of available commands:

```
C:\>python user\pathname\pgcdemtools\apply_setsm_registration.py --dstdir pathname\output_location --overwrite pathname\input_directory
```

To specify the output directory, use the **“- -dstdir”** command followed by the output directory pathname. If you are looking to overwrite the existing index, include the **“- -overwrite”** command. Lastly, include the pathname to the input directory. This script can be run on a cluster utilizing the cluster submit commands mentioned above.

resample_setsm.py

This utility allows for the resampling of a SETSM DEM to a lower resolution. Available commands include:

```
C:\>python user\pathname\pgcdemtools\resample_setsm.py -c {matchtag,dem} -r 16 --overwrite pathname\input_directory
```

To specify the SETSM DEM to resample, use the **“-c”** command followed by the DEM matchtag. The **“-r”** command allows for the specification of the output DEM. The default option if not included is 16m. If you are looking to overwrite the existing index, include the **“- -overwrite”** command. Lastly, include the source directory pathname or the pathname for the specific DEM being resampled.

divide_setsm_tiles.py

This tool divides SETSM mosaics into subtiles. Below is an example of available commands:

```
C:\>python user\pathname\pgcdemtools\divide_setsm.py --log pathname\log_file --num-rows 100 --num-cols 100 --tiles xxxxx -version v1.2 --cutline-loc pathname\cutline.shp --build-ovr pathname\input_directory
```

The **“- -log”** command followed by a pathname will output a log file to that location. The **“- -num-rows”** command specifies the number of rows in the subtiles, while the **“- -num-cols”** command specifies the number of columns in the subtiles. To denote the specific tiles to process, use the **“- -tiles”** command followed by a list of tiles to process. The list is comma delimited. The **“- -version”** command allows for the specification of the version string to use. Areas of bad data can be cut using the **“- -cutline-loc”** command followed the pathname to the directory containing cutline shapefiles indicating the areas of bad data. The **“- -build-ovr”** command can be used to build overviews. Lastly, include the pathname to the source directory or DEM.