

# Atmospheric Profile Builder

Description	APBuilder User Guide
Author(s)	Raul Viera-Mercado, Keehoon Kim
LLNL Release Number	LLNL-CODE-2012226
Software Version	1.0.0
Document Last Updated	October 13, 2025
DOI	<a href="https://doi.org/10.11578/dc.20251009.4">https://doi.org/10.11578/dc.20251009.4</a>

# Table of Contents

[1 Home](#)

[2 Install](#)

[3 Usage](#)

[4 How It Works](#)

[5 About](#)

[6 FAQ](#)

# 1 Atmospheric Profile Builder (APBuilder)

The APBuilder is a tool to create [AC2Dr](#) atmosphere profiles by downloading a weather model and making transformations to generate the 1D or 2D binary profile. The binary profile is used as an input to AC2Dr.

To install APBuilder, please [visit this page](#).

# 2 Installing APBuilder

If you are looking to upgrade apbuilder, you can skip to the [Upgrade](#) section.

## 2.1 Types of Installation

There are 2 ways to install the application.

1. [Native install](#) using conda
2. [Container install](#)

## 2.2 Native Install

### 2.2.1 Pre-requisites

Create conda virtual environment

```
conda create -y --name apbuilder python=3.11
conda activate apbuilder
```

Before installing the tool, [PyGMT](#) must be installed in the system.

```
conda install -y -c conda-forge 'pygmt=0.12' libgdal-grib libgdal-netcdf
```

### 2.2.2 Installation

Install the Python application, including all dependencies, using the following command:

```
pip install apbuilder
```

### 2.2.3 Verify

You can run a selfcheck to make sure there are no errors with 3rd party dependencies.

```
apbuilder info --selfcheck
```

## 2.3 Container Install

APBuilder can be run as a container. We provide a docker compose file for ease of use.

You can follow the steps below to configure and run the container.

### 2.3.1 Pre-requisites

To install the container image, you will need a container runtime, for example Docker or Podman. These instructions are written using docker.

## 2.3.2 Download

Download the docker compose and volume override files.

1. Docker Compose
2. Volume Override

## 2.3.3 Configure

In the `volume-overrides.yaml` change the following:

1. `<data-source>` to the full root path of where you want to store the data. A `data` and `output` directory will be automatically created in this root path.
2. `<cacert-source>` to the full path where your ca-certificate PEM file is located to allow SSL.

## 2.3.4 Verify

Run the following command to verify installation and configuration.

It should print the apbuilder help message.

NOTE: The first time it will take a little while because it will download the container image.

```
docker compose \
-f apbuilder-local-docker-compose.yaml \
-f volume-overrides.yaml run \
--rm apbuilder
```

## 2.3.5 Usage

You can run with docker compose using the following command:

```
docker compose \
-f apbuilder-local-docker-compose.yaml \
-f volume-overrides.yaml run \
--rm apbuilder \
apbuilder info --selfcheck
```

The actual command we are telling the container to execute is `apbuilder info --selfcheck`, which you can change to execute any apbuilder command.

For example, try replacing it with `info -h`

The following prefix is always needed to run apbuilder with docker compose.

```
docker compose \  
-f apbuilder-local-docker-compose.yaml \  
-f volume-overrides.yaml \  
run --rm apbuilder
```

It is recommended to create an alias to simplify usage:

```
alias apbuilder='  
docker compose \  
-f apbuilder-local-docker-compose.yaml \  
-f volume-overrides.yaml \  
run --rm apbuilder apbuilder'
```

NOTE: The word apbuilder is mentioned twice at the end of the command. This is not a mistake. The first instance is the name of the container. The second instance is the command we are executing inside the container.

Then you can easily run with:

```
apbuilder info -h
```

To learn more on how to use apbuilder, please visit the [Using APBuilder page](#).

## 2.4 Upgrade

To upgrade APBuilder along with herbie-data, use the following command. Note this will not update transitive dependencies. It will only update the explicit dependencies in APBuilder.

```
pip install apbuilder --upgrade
```

# 3 Using APBuilder

## 3.1 Validate Installation

You can run a selfcheck to make sure there are no errors with 3rd party dependencies.

No errors should be reported. If there is any error, please double check you installed everything correctly.

```
apbuilder info --selfcheck
```

## 3.2 Test Run

You can run a basic 1D profile build with the following command:

```
apbuilder build1d 20240601 37.1149 243.9309 38.1149 244.9309
```

Here is the command to test a 2D profile build:

```
apbuilder build2d 20160815 48.3200 16.8700 47.9183 19.8908 0.5
```

## 3.3 Usage

**If NCEI source does not work on VPN, please disconnect VPN and try again.**

```
$ apbuilder --help
usage: apbuilder [-h] [-v] {build1d,build2d,info} ...

Atmospheric Profile Builder (APBuilder)

positional arguments:
  {build1d,build2d,info}
    build1d          build 1D profiles
    build2d          build 2D profiles
    info             additional information

options:
  -h, --help        show this help message and exit
  -v, --version     show program's version number and exit

LLNL
```

```

$ apbuilder build1d --help
usage: apbuilder build1d [-h] [-wm {GFS,HRRR,IFS,RAP,ERA5}] [-e] [-nd]
                         [-c {model,forecast,forecast-only}] [-lf filename] [-out out_dir]
                         [-data data_dir] [-pof prefix] [-clim min max] [-dlim min max]
                         [-wlim min max]
                         datetime [-90, 90] [-180, 360] [-90, 90] [-180, 360]

positional arguments:
  datetime           date of data in ISO 8601 or epoch format
  [-90, 90]          latitude of the profile
  [-180, 360]        longitude of the profile
  [-90, 90]          latitude of the point to which direction the wind speed is calculated
  [-180, 360]        longitude of the point to which direction the wind speed is calculated

options:
  -h, --help          show this help message and exit
  -wm {GFS,HRRR,IFS,RAP,ERA5}, --weather-model {GFS,HRRR,IFS,RAP,ERA5}
                      weather model (default: GFS)
  -e, --epoch         specify if datetime is in epoch format (default: False)
  -nd, --no-download  test out the parameters without downloading data (default: False)
  -c {model,forecast,forecast-only}, --cycle {model,forecast,forecast-only}
                      the model or forecast cycle type to download the data (default: model)
  -lf filename, --local-filename filename
                      name of the local file to read instead of downloading from remote server
                      (default: None)
  -out out_dir, --output-directory out_dir
                      specify full path directory to save output files (default:
                      C:\Users\myuser\apbuilder\output)
  -data data_dir, --data-directory data_dir
                      specify full path directory to save the weather models data files (default:
                      C:\Users\myuser\apbuilder\data)
  -pof prefix, --prefix-output-file prefix
                      Prefix for the output binary files (default: None)
  -clim min max      min and max values for sound speed profile plots (default: [0, 0])
  -dlim min max      min and max values for density profile plots (default: [0, 0])
  -wlim min max      min and max values for wind profile plots (default: [0, 0])

```

```

$ apbuilder build2d --help
usage: apbuilder build2d [-h] [-wm {GFS,HRRR,IFS,RAP,ERA5}] [-e] [-nd]
                         [-c {model,forecast,forecast-only}] [-lf filename] [-out out_dir]
                         [-data data_dir] [-pof prefix] [-clim min max] [-dlim min max]
                         [-wlim min max]
                         datetime [-90, 90] [-180, 360] [-90, 90] [-180, 360] [[0, 360]] [[0, ]]

positional arguments:
  datetime           date of data in ISO 8601 or epoch format
  [-90, 90]          latitude of the starting point of the 2D section
  [-180, 360]         longitude of the starting point of the 2D section
  [-90, 90]          latitude of the end point of the 2D section
  [-180, 360]         longitude of the end point of the 2D section
  [0, 360]           horizontal resolution of 2D slice in degree (default: 0)
  [0, ]              vertical resolution of 2D slice in meter (default: 100)

options:
  -h, --help          show this help message and exit
  -wm {GFS,HRRR,IFS,RAP,ERA5}, --weather-model {GFS,HRRR,IFS,RAP,ERA5}
                      weather model (default: GFS)
  -e, --epoch         specify if datetime is in epoch format (default: False)
  -nd, --no-download  test out the parameters without downloading data (default: False)
  -c {model,forecast,forecast-only}, --cycle {model,forecast,forecast-only}
                      the model or forecast cycle type to download the data (default: model)
  -lf filename, --local-filename filename
                      name of the local file to read instead of downloading from remote server
                      (default: None)
  -out out_dir, --output-directory out_dir
                      specify full path directory to save output files (default:
                      C:\Users\myuser\apbuilder\output)
  -data data_dir, --data-directory data_dir
                      specify full path directory to save the weather models data files (default:
                      C:\Users\myuser\apbuilder\data)
  -pof prefix, --prefix-output-file prefix
                      Prefix for the output binary files (default: None)
  -clim min max       min and max values for sound speed profile plots (default: [0, 0])
  -dlim min max       min and max values for density profile plots (default: [0, 0])
  -wlim min max       min and max values for wind profile plots (default: [0, 0])

```

```
$ apbuilder info --help
usage: apbuilder info [-h] [-sb COLUMN_NAME] [-ro] (-mi | -gmt | -sc)

options:
-h, --help            show this help message and exit
-sb COLUMN_NAME, --sort-by COLUMN_NAME
                      column name to sort the available weather models table. Allowed values:
{Model, Grid (Deg), Time Period, Model Cycle, Forecast Cycle, Geographic
Extent} (default: Model)
-ro, --reverse-order  Reverse the sorting order of the available weather models table (default:
False)

exclusive options:
-mi, --models-info   print information about the available weather models
-gmt, --gmt-info     print information about GMT
-sc, --selfcheck      check the installation of the tool and dependencies
```

# 4 How it works?

APBuilder uses `herbie-data` to download grib and inventory files from remote servers. APBuilder also supports reading local grib files without an inventory file. This application will check if the grib file exists locally, if not it will download it from a remote server. Then process the file accordingly. The next time you run this application with the datetime and weather model, it will use the local copy of the file, saving time as there is no need to download the file. You can also specify a local grib filename to use instead of downloading from a remote server.

Then APBuilder will generate the atmospheric profile files to be used in AC2Dr. In addition, it generates images to visualize the data. The output is saved to the default location or the directory specified by the user with the `-out` parameter.

## 4.1 Directory Structure

APBuilder uses a directory structure created by `herbie-data`, which is the following:

- `out_dir` is the output directory set by the user in the command
- `weather_model` is the weather model set by the user in the command, or default if not set
- `yyyymmdd` is the date of the files to search set by the user in the command
- `filename` is the filename of the grib2 file. This filename has a defined structure, see more info below.

```
|-- out_dir  
|   |-- weather_model  
|       |-- yyyymmdd  
|           |-- filename
```

Here is a concrete example.

```
|-- data  
|   |-- gfs  
|       |-- 20201029  
|           |-- gfs_4_20201029_0000_000.grb2
```

## 4.2 Adding local files manually

You can use a local grib file instead of downloading from a remote server. The file must be placed inside a specific folder as denoted above on the `Directory Structure` section.

To use the local file with APBuilder, make sure to specify the `-lf` or `--local-filename` flag with the full path to the file.

For example:

```
apbuilder buildId 20200804T00:00:00 50.6565 6.8033 53.2534 8.68980 \
C:\\\\Users\\\\myuser\\\\apbuilder\\\\data \
-lf C:\\\\Users\\\\myuser\\\\apbuilder\\\\data\\\\ifs\\\\20200804\\\\ECMWF_ERA5_20200804_1200.grib \
-wm ERA5
```

# 5 Atmospheric Profile Builder (APBuilder)

The APBuilder is a tool to create [AC2Dr](#) atmosphere profiles by downloading a weather model and making transformations to generate the 1D or 2D binary profile. The binary profile is used as an input to AC2Dr.

AC2Dr is a 2-D numerical solver for the acoustic wave equation using the finite difference method.

## 5.1 License

See [LICENSE](#) and [NOTICE](#) files for details.

LLNL-CODE-2012226

# 6 Frequently Asked Questions (FAQ)

## 6.1 What is Volume Mapping in Containers?

Volume mapping allows you to link a directory or file from your host machine (the `source`) to a directory inside a container (the `target`). This makes it possible for the container to read and write files that are stored on your local filesystem. Any changes made by the container in the mapped (`target`) directory are immediately reflected on your host (`source`), and vice versa. This is useful for sharing data, saving output, or persisting files even after the container stops. Here is example volume mapping:

```
services:  
  apbuilder:  
    volumes:  
      - type: bind  
        source: /tmp/apbuilder/data  
        target: /home/apbuilder/apbuilder/data
```

## 6.2 The data is not saved to the specified output directory when using the container

When using the container image, you must specify in `volume-overrides.yaml` the `source` (a path on your local machine) that will be mapped to the `target` (the path inside the container). The output data is saved to the `target` directory as seen by the container. However, because of the volume mapping, any data written to the `target` directory inside the container is actually stored in your `source` directory on your local filesystem.

Therefore, the `-out` directory must be the absolute path **inside the container** (the `target` path), not the host (`source`) path. For example, consider the following `volume-overrides.yaml`:

```
services:  
  apbuilder:  
    volumes:  
      - type: bind  
        source: /tmp/apbuilder/data  
        target: /home/apbuilder/apbuilder/data
```

If you want the output data to be accessible on your local machine at `/tmp/apbuilder/data/myanalysis`, you should set the `-out` parameter to `/home/apbuilder/apbuilder/data/myanalysis` (the `target` path inside the container). The container will write to this directory, and the data will appear in `/tmp/apbuilder/data/myanalysis` on your host machine.

### 6.2.1 Table Summary

Parameter	Value inside container	Value on host machine
-out argument	/home/apbuilder/apbuilder/data/myanalysis	/tmp/apbuilder/data/myanalysis
Data physically stored	/home/apbuilder/apbuilder/data/myanalysis (container)	/tmp/apbuilder/data/myanalysis (host)

### In summary:

Always use the `target` (container path) for the `-out` parameter. The data will be available on your host in the `source` directory you mapped.

## 6.3 Using Windows Git Bash is modifying my output directory when using the container

The specified `-out` directory can be automatically modified when using Git Bash for Windows to run the container. This is because Git Bash tries to "translate" Unix-style paths (like `/home/...`) to Windows-style paths (`C:/Program Files/Git/home/...`). When you run a command that includes a path starting with `/`, Git Bash assumes you mean a Windows absolute path, and prepends the Git installation directory if it can't resolve the path. Using `//` at the start (double slash) can prevent Git Bash from translating the path:

```
-out //home/apbuilder/apbuilder/data/raul-data
```

## 6.4 Getting SSL errors when downloading data

If you are connected to VPN and having SSL errors, you will have to configure an environment variable with the SSL certificate.

Here is an example to configure on Linux. Use the defined approach for your OS to set the environment variable accordingly.

```
export REQUESTS_CA_BUNDLE=/path/to/cacert.pem
```

Documentation built with [MkDocs](#).