ncpag2s-clc

Release 1.0.0

Claus Hetzer

Oct 10, 2024

CONTENTS

CHAPTER

ONE

OVERVIEW

The **ncpag2s-clc** package provides a convenient command-line interface for retrieving data from the NCPA-G2S system. Prior to development, data retrieval required filling in multiple web forms and waiting for a batch process to complete and a download link to be sent by email. Now, data requests can be submitted directly (and anonymously) to the server, which will fulfill them synchronously.

1.1 Requirements

The **ncpag2s-clc** package requires Python version 3.9 or greater. All package dependencies are part of the base Python distribution.

1.2 Installation

Just download the code tree from its GitHub repository and optionally add the root directory of the repository to your PATH. No additional packages are required.

1.3 Operation

The client is operated through the ``**ncpag2s.py**`` script, with the first argument being the subcommand and subsequent arguments being the flags and options necessary. The client will parse the arguments, generate and submit the necessary HTTP requests, and format the responses as requested. Subcommands can be listed by running `ncpag2s.py help`, with more detailed help text available for each subcommand from `ncpag2s.py help [subcommand]`. Options and flags will vary between subcommands but will be consistent when the meaning is roughly the same.

1.4 Output

Output is ASCII text and can be specified in JSON or NCPAprop format. Grid requests may also be output in InfraGA format.

1.5 Quick Example

With `ncpag2s.py` in your path, enter:

```
ncpag2s.py point --date 2023-07-04 --hour 12 --lat 37.867 --lon -122.259
```

All request commands will be in similar formats, with the switchboard program `ncpag2s.py`, the subcommand (in this case `point`), and flags following.

CHAPTER

TWO

USER GUIDE

This section will go into greater detail about the use of the various available subcommands and their flags. Each subcommand has help text that can be accessed with `ncpag2s.py help [subcommand]`.

2.1 Time Conventions

G2S data is generally available hourly after 1 January 2011, and every 6 hours between 1 January 2003 and 31 December 2010. Thus, there is no reason to specify times to any precision greater than 1 hour. In all cases, subcommands require dates and times to be specified, either singly or as a range. In each case, dates and hours are specified separately. Dates can be in any ISO8601-compliant format (e.g. 2023-08-01) while hours are specified as integers from 0-23. All times are in UTC.

2.2 Output Formats

In all cases, `json` and `ncpaprop` outputs are available. Others may be available for specific subcommands. For details on these formats, see Data Formats.

2.3 `checktime`: Confirm data availability

This subcommand queries the G2S database to verify that data is available for the date and time requested. The command will return either "Time [datetime] is available" or "Time [datetime] is not available".

Usage is:

```
ncpag2s.py checktime --date [date] --hour [hour]
```

where `[flags]` is one or more of the following, with flags that require each other grouped together:

Required: `--date [date]` `--hour [hour]`

The UTC date and hour for the grid file.

Example Command `ncpag2s.py checktime --date 2023-08-01 --hour 8`

```
ncpag2s.py checktime --date 2008-08-01 --hour 8
```

2.4 `point`: A single geographic point

This subcommand will extract profile(s) at one or more times at a single geographic point (i.e. one latitude/longitude pair). Usage is:

`ncpag2s.py point [flags]`

where `[flags]` is one or more of the following, with flags that require each other grouped together:

Required:

`--latitude [latitude]` `--longitude [longitude]`

The latitude/longitude coordinates at which to extract the profile. Coordinates are expected to be in the ranges [-90,90] and [-180,180], respectively.

One of the following groups is required:

`--date [date]` `--hour [hour]`

The UTC date and hour at which to extract the profile.

OR

`--startdate [date]` `--starthour [hour]` `--enddate [date]` `--endhour [hour]` `--every [interval]`

The starting and ending dates and times for a range of multiple times, extracted at the requested interval (in hours).

Optional:

`--outputformat [json|ncpaprop]` Specifies the format in which the profile(s) are output. The default is `json`.

`--output [destination]` The destination for the output data. The default is to print to the screen. For `json` output this should be a file. For `ncpaprop` output this should be a file for single-time requests, or a directory for multi-time requests.

Example Commands `ncpag2s.py point --date 2023-07-04 --hour 12 --lat 37.867 --lon -122.259`

```
ncpag2s.py point --date 2023-07-04 --hour 12 --lat 37.867 --lon -122.259
--outputformat ncpaprop --output rasputin.dat
```

2.5 `line`: An evenly-spaced line of points

This subcommand will extract an evenly-spaced great-circle line of profiles between two latitude/longitude pairs at one or more times. Usage is:

`ncpag2s.py line [flags]`

where `[flags]` is one or more of the following, with flags that require each other grouped together:

Required:

`--startlatitude [latitude]` `--startlongitude [longitude]` `--endlatitude [latitude]` `--endlongitude [longitude]`

The starting and ending latitude/longitude coordinates of the great-circle line. Coordinates are expected to be in the ranges [-90,90] and [-180,180], respectively.

`--points` The number of points to extract in the line.

One of the following groups is required:

`--date [date]` `--hour [hour]`

The UTC date and hour at which to extract the profile.

OR

`--startdate [date]` `--starthour [hour]` `--enddate [date]` `--endhour [hour]` `--every [interval]`

The starting and ending dates and times for a range of multiple times, extracted at the requested interval (in hours).

Optional:

`--outputformat [json|ncpaprop]` Specifies the format in which the profile(s) are output. The default is `json`.

`--output [destination]` The destination for the output data. The default is to print to the screen. For `json` output this should be a file. For `ncpaprop` output this should be a directory.

Example Commands `ncpag2s.py line --date 2023-08-10 --hour 0 --startlat 34.39 --startlon -89.51 --endlat 35.23 --endlon -106.66 --points 21`

`ncpag2s.py line --date 2023-08-10 --hour 0 --startlat 34.39 --startlon -89.51 --endlat 35.23 --endlon -106.66 --points 21 --outputformat ncpaprop --output /tmp/ms_to_abq`

2.6 `grid`: An evenly-spaced grid of points

This subcommand will extract an evenly-spaced grid of profiles in Mercator projection (i.e. evenly-spaced latitude and longitude intervals, not necessarily in physical distance). Usage is:

`ncpag2s.py grid [flags]`

where `[flags]` is one or more of the following, with flags that require each other grouped together:

Required:

`--startlatitude [latitude]` `--startlongitude [longitude]` `--endlatitude [latitude]` `--endlongitude [longitude]`

The starting and ending latitude/longitude coordinates of the grid. These will correspond to the lower-left (i.e. southwesternmost) and upper-right (i.e. northeasternmost) corners of the grid, respectively. Coordinates are expected to be in the ranges [-90,90] and [-180,180], respectively.

`--latpoints` `--lonpoints`

The number of latitude and longitude points to extract in the grid.

One of the following groups is required:

`--date [date]` `--hour [hour]`

The UTC date and hour at which to extract the profile.

OR

`--startdate [date]` `--starthour [hour]` `--enddate [date]` `--endhour [hour]` `--every [interval]`

The starting and ending dates and times for a range of multiple times, extracted at the requested interval (in hours).

Optional:

--outputformat [json|ncpaprop|infraga] Specifies the format in which the profile(s) are output. The default is `json`.

--output [destination] The destination for the output data. The default is to print to the screen. For `json` output this should be a file. For `ncpaprop` or `infraga` output this should be a directory.

Example Commands `ncpag2s.py grid --date 2023-08-10 --hour 0 --startlat 34.0 --startlon -89.0 --endlat 40.0 --endlon -96.0 --latpoints 13 --lonpoints 15`

```
ncpag2s.py grid --date 2023-08-10 --hour 0 --startlat 34.39 --startlon -89.51
--endlat 35.23 --endlon -106.66 --points 21 --outputformat infraga --output /tmp/
testgrid
```

2.7 `raw`: A raw G2S coefficient file

This subcommand will return a raw G2S coefficient file, for use if you are authorized to possess the G2S extraction software. Usage is:

`ncpag2s.py raw [flags]`

where `[flags]` is one or more of the following, with flags that require each other grouped together:

Required: `--date [date]` `--hour [hour]`

The UTC date and hour for the grid file.

Optional:

--outputfile [filename] Write the coefficients to this filename. Default will let the server decide.

Example Command `ncpag2s.py raw --date 2023-07-04 --hour 12 --outputfile tester.bin`

CHAPTER

THREE

DATA FORMATS

3.1 JSON Format

The default JSON format contains a "metadata" structure, a "data" structure containing one or more "parameter" structures, and some bookkeeping entries at the end. In more detail, the format is structured as follows:

Individual Profiles

An example of a single G2S profile in JSON format is shown here:

```
{
  "metadata": {
      "sourcefile": "G2SGCSB2023012404.bin",
      "nz": 1501,
      "time": {
          "datetime": "2023-01-24T04:00:00",
          "format": "%Y-%m-%dT%H:%M:%S",
          "__extended_json_type__": "datetime"
      },
      "location": {
          "latitude": 19.59,
          "longitude": -155.89,
          "__extended_json_type__": "Location"
      },
      "parameters": 7
  },
  "data": [
          {
              "parameter": "Z0",
              "description": "Ground Height",
              "units": "km",
              "n": 1,
              "values": [
                  1.052
              ]
          },
          {
              "parameter": "Z",
              "description": "Height",
              "units": "km",
              "n": 1501,
```

(continues on next page)

(continued from previous page)

```
            "values": [
                0.0,
                0.1,
                ...
```

etc.

JSON files containing multiple profiles will generally be organized into profile sets, which may be `line``s, ``grid``s, or other logical groupings. In these cases there will be an outer JSON block with its own metadata structure, which will vary by the type of grouping. So a ``line` grouping might look like this:

```
{
  "metadata": {
      "type": "line",
      "points": 21,
      "reference_location": {
          "latitude": 19.59,
          "longitude": -155.89,
          "__extended_json_type__": "Location"
      },
  },
  "points": [
      {
      <individual profile structure>
      },
      {
      <individual profile structure>
      },
```

etc. In these cases, additional fields will often be added to the metadata block for each constituent profile as appropriate; for example in the case of the line above, each profile's metadata block will also have a "range" entry with the distance to the reference location in km.

This nesting behavior is extended as needed. For example if lines are requested for multiple times, the individual lines will be presented as a comma-separated series of the above structure, enclosed within square brackets.

3.2 NCPAprop Format

This is the format used by the ncpaprop package. It consists of a series of comments, interspersed with formatted header lines that describe the data structure, followed by a set of space-separated columns of ASCII data, one column per atmospheric property. An example of the format would be:

```
# Data Source: G2SGCSB2023070412.bin
# Model Time 2023-07-04T12:00:00
# Location = [ 37.8670, -122.2590 ]
# Fields = [ Z(km), T(K), U(m/s), V(m/s), R(g/cm3), P(mbar) ]
# Ground Height = 0.076 km
# The following lines are formatted input for ncpaprop
#% 0, Z0, km, 0.076
#% 1, Z, km
#% 2, T, K
```

(continues on next page)

(continued from previous page)

```
#% 3, U, m/s
#% 4, V, m/s
#% 5, RHO, g/cm3
#% 6, P, mbar
  0.000      2.85880e+02       1.40070e+00       1.26130e+00       1.23280e-03       1.
↪01150e+03
  0.100      2.85530e+02       1.59140e+00       1.33830e+00       1.21970e-03       9.
↪99540e+02
  0.200      2.85200e+02       1.78140e+00       1.41670e+00       1.20670e-03       9.
↪87710e+02
  0.300      2.85250e+02       1.95350e+00       1.53070e+00       1.19220e-03       9.
↪76030e+02
  0.400      2.86050e+02       2.09570e+00       1.67470e+00       1.17480e-03       9.
↪64500e+02
...
```

etc. In this example there are 6 columns of data. The formatted header lines beginning with `#%` associate the column number with a label and its units. Header lines with column number 0 indicate scalar quantities associated with the profile overall.

Because this format was designed for single-location profiles only, multi-profile requests have their own formats. Line requests are output as a directory `profiles` containing a series of single-profile files, one per point, and a summary file associating each profile with its distance from the beginning of the line. Grid requests are returned similarly, except the summary file associates each profile with its latitude/longitude coordinates.

3.3 InfraGA Format

This data format is available for grid requests only and is intended to work with the range-dependent modules of the LANL InfraGA geometric acoustics package. It is similar to the `ncpaprop` format, except that two summary files are created, one with the latitude points and one with the longitude points, and the filename structure is changed to that expected by the range-dependent InfraGA modules. A third file `flags.txt` is also generated, providing the flags to use with InfraGA in order to read the grid files.

CHAPTER

FOUR

AUTHORSHIP, ACKNOWLEDGEMENTS, AND REFERENCES

The NCPA-G2S Command Line Client was developed by Claus Hetzer at the National Center for Physical Acoustics, the University of Mississippi, with the support of the Sandia National Laboratories under award #2515096.

For questions about installation, usage, or features, contact:

Claus Hetzer claus@olemiss.edu

4.1 Citing NCPA-G2S and the CLI

Publications that make use of data from the NCPA-G2S system are kindly requested to cite one or more of the following references, as appropriate:

For the G2S model itself:

- Drob, D.P., Picone, J.M., & Garces, M. (2003). "Global morphology of infrasound propagation". *Journal of Geophysical Research*, **108** (D21), 4680. doi: 10.1029/2002JD003307.
- Drob, D.P., Garces, M., Hedlin, M., & Brachet, N. (2010). "The temporal morphology of infrasound propagation". *Pure & Applied Geophysics*, **167** (4-5), 437 – 453. doi: 10.1007/s00024-010-0080-6.
- Drob, D.P. (2019). "Meteorology, climatology, and upper atmospheric composition for infrasound propagation modeling". in Le Pichon, A., Blanc, E., and Hauchecorne, A. (eds.), *Infrasound Monitoring for Atmospheric Studies, 2nd. ed.*, Springer, doi: 10.1007/978-3-319-75140-5.

For data from the NCPA-G2S System:

- Hetzer, C.H., Drob, D.P., & Zabel, K. (2019). "The NCPA-G2S request system". https://g2s.ncpa.olemiss.edu.
- Hetzer, C.H. (2024). "The NCPAG2S Command-Line Client". https://github.com/chetzer-ncpa/ncpag2s-clc. doi: 10.5281/zenodo.13345069.

4.2 G2S Data Sources

The NCPA G2S system generates its models using the following data sources and models:

- National Centers for Environmental Prediction/National Weather Service/NOAA/U.S. Department of Commerce. 2015, updated daily. "NCEP GFS 0.25 Degree Global Forecast Grids". National Center for Atmospheric Research, Computational and Information Systems Laboratory. doi:10.5065/D65D8PWK.
- Rienecker, M.M., Suarez, M.J., Todling, R., Bacmeister, J., Takacs, L., Liu, H.-C., Gu, W., Sienkiewicz, M., Koster, R.D., Gelaro, R., et al. (2008). "The GEOS-5 Data Assimilation System – Documentation of Versions

5.0.1, 5.1.0, and 5.2.0". *NASA Technical Report Series on Global Modeling and Data Assimilation*, **27**, NASA Technical Report NASA/TM-2008-104606.

- Bosilovich, M., Akella, S., Coy, L., Cullather, R., Draper, C., Gelaro, R., Kovach, R., Liu, Q., Molod, A., Norris, P., et al. (2015) "Merra-2: initial evaluation of the climate". *NASA Technical Report Series on Global Modeling and Data Assimilation*, **43**, NASA Technical Report NASA/TM-2015 104606.
- Drob, D.P., Emmert, J.T., Meriwether, J.W., Makela, J.J., Doornbos, E., Conde, M., Hernandez, G., Noto, J., Zawdie, K.A., McDonald, S.E., et al. (2015). "An update to the horizontal wind model (HWM): the quiet time thermosphere". *Earth Space Science* **2** (7), 301 – 319.
- Picone, J., Hedin, A., Drob, D.P., Aikin, A. (2002). "NRLMSISE-00 empirical model of the atmosphere: Statistical comparisons and scientific issues". *Journal of Geophysical Research Space Physics*, **107** (A12).