RFC: nagg like-a-file (-l) option to specify characteristics of output files

Larry Knox
Albert Cheng

1. Purpose

The intent of this document is to describe the proposed *Like* option for the nagg tool. (The reader should consult the Nagg tool reference document first if he is not familiar with the tool.) Current options –n and –A set the number of data granules in each aggregated nagg output file. The –t option specifies a list of data products to be processed from the input files. The proposed option will set both of these properties according to the file specified by the *Like* option.

Section 2 shows a user example using the Like option and the characteristics of Nagg output files. Section 3 specifies the requirements of the Like option. Section 4 describes the Implementation design. Section 5 lists the document changes for the added Like option. Section 6 describes the Testing requirements. The last section summarizes recommendations given in this document.

2. Introduction

2.1 User Example

A Nagg user may want to package the daily downloads of all of the VIIRS Moderate Resolution Band data in aggregations spanning one hour of data collection each. This can be done with "nagg –A 3600 –t SVM01,SVM02,SVM03,SVM04,SVM05,SVM06,SVM07, SVM08,SVM09,SVM10,SVM11,SVM12,SVM13,SVM14,SVM15,SVM16 <input files>". The -l <*file*> option will enable the user to create files once with this command, select an output file that is not a partial aggregation from this initial run, and subsequently run "nagg -l < *file*> <*input files*>". The output files should be the same provided the selected file matches the original command.

The first and last files in the original output files should be avoided or checked for number of granules, since they may be partial files with less than the full number of granules. Using a partial aggregation file as the selected file for -l will result in smaller aggregations than using a full aggregation file.

2.2 Nagg output files characteristics

Two properties are specified in the first nagg command above. The first is the number of data granules of each product that will be included in each full aggregation output file. In the command above it is determined by –A 3600, which for the VIIRS products requires 43 granules per full aggregation. This would be the same if –n 43 were specified. The other specified property is the list of products that follows –t. These products will be aggregated in the output files. As there are no other options specified, the other properties will be determined by nagg defaults.

3. Requirement Specification

Each JPSS file contains enough information to determine the products to be included and the number of granules of each product in each full aggregation. When the proposed *Like* option is invoked, nagg will inspect the specified example file, find the products and number of granules per product in the file, and create its output files with the same products and aggregation number as those in the file. Either of these properties can be overridden by command line options -n, -A, or -t. Other properties will be determined by command line options or default nagg behavior.

Note that the current implementation does not support use of the *Like* option for GEO only files. The –g <geoproduct> option and –n or –A options must be used for GEO only files.

3.1 Nagg Option Syntax

The *Like* option is specified via the following syntax:

-l *<file>*, --like=*<file>*

If -l *<file>* is found in the nagg command, open *<file>* and determine the aggregation number and products for the output files. *<file>* may be a filename or a path to a file.

If any of the command line options –A, -n, or -t are also found, override either of the characteristics according to the command line option.

Return an appropriate error message <file> is omitted, cannot be opened, or does not contain a required characteristic.

3.2 Overriding of the properties set by -l <file>

Any of the following command options when used with -l <file> will override the corresponding property from <file>.

-n *N*

-A *seconds*

-t *<product list>*

Overriding both the aggregate number and the product types on the command line will render the –l option of no effect. An error message telling the user that the –l option has no effect with the two overriding options may reduce user confusion.

3.3 Partially filled NPP files

Partially filled output files are created when the first or last granule does not begin or end a full aggregation. The file contains attributes AggregateGranuleNumber which according to the Common Data Format Control Book V "provides a count of the valid granules in the HDF5 file" and AggregateBeginningTime, "based on the BeginningTime of the individual granules that are included in the aggregation" and AggregateEndingTime, "based on the EndingTime of the individual granules that are included in the aggregation". Since there are no attributes in a file that identify it as a full or partial aggregation, nagg can only use the number of granules per product in the file as the aggregation number. The user must provide a file with the desired number of granules and avoid partially filled files that may be at the beginning or end of a series of files.

3.4 Table of example and input file elements, conditions and expected nagg

Element	Example and input file conditions determining outcome	nagg command option equivalent / successful outcome or error
Product groups	All products in example file are found in input files	-t <product list> aggregate matching products
	Example file has one or more products not found in input files	ERROR
Aggregation number	0 < aggr_num < nagg granule_limit in example file	-n aggr_num create aggregations of size aggr_num
	0 >= aggr_num or aggr_num > granule_limit in example file	ERROR

4. Implementation Design

4.1 Products List

The products to be included will be determined from the subgroups of /Data_Products in the example file, which are named with the same names as the products. These can be converted to DPIDs for the –t list.

4.2 Number of granules

The number of granules of each product in the file is stored in an attribute of each <product name>_Aggr dataset named "AggregateNumberGranules".

4.3 Override by –t, -n or -A

The products found will be overridden by –t in the nagg command as will the aggregation number by –n or –A in the nagg command

5. Required documentation updates

5.1 Update for Reference Manual

These are the current and proposed reference manual entries for the –l option. The idea of using the first file encountered as the example file has been eliminated.

5.1.1 Current entry

`-l file`

(To be supported in future implementation.)

Package like the example `file` in number or type list. Options on the command line override the example. If both `-l` and `-t` are omitted, then the first NPP data product file encountered will be used as the example file.

5.1.2 Proposed entry

`-l file`

Package like the example `file` in number of granules and type list. Options on the command line override the aggregation number and product list from the example file.

5.2 User Examples document additions

These examples are intended to demonstrate the common behavior of the –l <example_file> option and the behavior with possible command options that override a part of the common behavior.

5.2.1 Example file option with no overriding command options:

nagg –l *<file> <input_files>*

5.2.2 Example file option with –A command option override:

nagg –l *<file>* -A 300, –t *<type list> <input_files>*

5.2.3 Example file option with –n command option overrides:

nagg –l *<file> –n 4 <input_files>*

5.2.4 Example file option with –t command option overrides:

nagg –l *<file>* -t *<type list> <input_files>*

6 Test Specification

The nagg tool creates output files that match the example file in terms of products included, and aggregation number.

Tests 1 – 4 are for scenarios with the nagg –l option in this command:
nagg –l < *file*> <*input_files*>

1. Example file has:
   Products also found in input files
   Any reasonable number of granules
   Packaged geolocation product
   Output:
   Packaged files produced with products and aggregation number matching example file
2. Example file has:
   One or more products not found in input files
   Output:
   nagg: ***ERROR*** nagg_get_granules(): The number of product types for which
   granules were found was less than the number of products requested.
3. Example file has:
   1 data product and N_GEO_Ref attribute in example file
   Matching and available GEO product for input files
   Output:
   Packaged data product and GEO product output files
4. Example file has:
   1 data product
   No GEO product available for input files
   Output:
   nagg: ***ERROR*** nagg_get_granules(): no granules found for geoproduct.

Tests 5 – 7 are for the –l option with other command options that override the effects of the example file.

5. Example file characteristics are overridden by –A 300
   Command: nagg –l <*file*> –A 300 <*input_files*>
   Output:
   Files with aggregation size according to –A 300, products according to <type list> and other characteristics as determined by the example file.
6. Example file characteristics are overridden by –n4
   Command: nagg –l <*file*> -n4, -g no <*input_files*>
   Output:

Files with aggregation size 4, no GEO granules, and other characteristics as determined by the example file.

7. Example file characteristics are overridden by –t *<type list>*
   Command: nagg –l *<file>* –t *<type list>* *<input_files>*
   Output:
   Files with products according to *<type list>* and other characteristics as determined by the example file.

Tests 8 – 9 are for the –l option with the example file name missing.

8. Command: nagg –l –S <input files>
   Output: nagg: ***ERROR*** Missing argument for –l option
9. Command : nagg –l <input file>
   Output: nagg: ***ERROR*** parse_options(): no input file given

7 Conclusion

The *File* option will provide a useful alternative for nagg to create files with a specific number of granules per files and a particular set of products. This will be most useful for aggregating additional files of a product in the same manner as was done previously or for aggregating files of a product or products on a recurring basis.

Nagg will not be able to identify whether <file> is a full or partial aggregation and will therefore use whatever number of granules it finds as the aggregation number. Some of the other properties that can be set on the command line could be determined by the *File* option, but doing so may add complexity and possibly user confusion, especially since the defaults will be commonly used.