RFC: nagg options to put aggregation in one output file

Larry Knox

This document summarizes proposed command line options for the NPP aggregating and packaging tool (nagg). We propose to implement the “--onefile” option to instruct nagg to place each output aggregation in one file. When the flag is specified the original bucket boundaries will not be considered, the size of the aggregation will be the number of input granules, and the option will be exclusive with options -A, -n or --number=<N>. A second option, “--nofill” would be potentially useful when the granules to be aggregated are not entirely contiguous.

Nagg (<http://www.hdfgroup.org/projects/npoess/nagg_index.html>) is a tool for aggregating NPP data granules from existing files into new files with a different number of granules or different combinations of compatible products than in the original files.When given a number of granules per aggregation with -n or --number, or a minimum time span per aggregation with -A, nagg determines where the boundaries between aggregations would have been if the aggregations had been produced by IDPS according to DDS Aggregation Methodology.

We are proposing two new options “--onefile” and “--nofill” to modify aggregation methodology to accommodate a new use case.

An example of this use case is to aggregate each day’s granules for a particular geographic area such as the Gulf of Mexico and the southern US. The granules covering the area are from more than one pass of the Suomi-NPP satellite. The granules between passes that are outside the area are not needed for certain purposes, and should not be replaced by fill granules when aggregating the granules for the day. Aggregating a subset of several satellite passes with nagg is currently not possible without contiguous swaths of actual or fill granules and an aggregation size that cannot be predicted without calculating the bucket boundaries determined by the granule size and the time since January 1, 1958. The proposed options will allow the user to aggregate only the most interesting files into a useful aggregation, putting them in one file and/or controlling the addition of fill granules to the aggregation.

# --onefile option

Users of nagg are sometimes interested in aggregating an arbitrary collection of granules for their own purposes, and do not want the resulting aggregation to be divided into multiple files because the collection spans a bucket boundary. The proposed --onefile option will allow aggregating all input granules for each product in a single file. Packaging of products will not be affected by this option. The --onefile option cannot be used with -n <aggregation number> or -A <seconds>.

# --nofill option

nagg versions through 1.6.1 create fill granules for all missing granules between input granules to be aggregated, except that files consisting entirely of fill granules are not created. Fill granules are also created for packaged files containing multiple products where one of the products is missing an input granule to match an input granule for another product.

If a user needs to aggregate granules from a geographic area as in the example use case given above, it is likely that granules from adjacent satellite passes will be included. The user may not want fill granules created around the other side of the earth connecting the adjacent granules in the area. Therefore nagg will provide an option to not create fill granules, --nofill. This option can be used independently of the --onefile option.

The desirability of fill granules may depend largely on the behavior of applications that are expected to use the output aggregation files. For example, both IDV and Panoply will connect the end of one block of ganules to the beginning of a second, separate block of granules when displaying them in the following examples. Adding the connecting fill granules will correct the display when using IDV, and the extra 60 fill granules increase the storage size by a mere 0.02%. On the other hand, on a Windows 7 laptop with 8GB RAM, Panoply will not display the file at all when it contains fill granules produced by nagg.

# Example

Input files:

GMODO-SVM01\_npp\_d20130506\_t1825525\_e1831329\_...\_XXXX\_XXX.h5

GMODO-SVM01\_npp\_d20130506\_t1831341\_e1837145\_...\_XXXX\_XXX.h5

GMODO-SVM01\_npp\_d20130506\_t2002365\_e2008169\_...\_XXXX\_XXX.h5

GMODO-SVM01\_npp\_d20130506\_t2008181\_e2013585\_...\_XXXX\_XXX.h5

These files contain 2 sets of contiguous data granules in 2 adjacent Suomi-NPP satellite passes over the Gulf (of Mexico) Coast area of the southeastern US on May 6,2013. The first set contains the data from 18:12:525 through 18:37:145 GMT, the second set from 20:02:365 through 20:13:585 GMT.

Command: nagg -n 16 -t SVM01 GMODO-SVM01\_npp\_d20130506\_t\*

Output:

Produced 16 granules in GMODO-SVM01\_npp\_d20130506\_t1825525\_e1848212\_...\_XXXX\_XXX.h5

Produced 12 granules in GMODO-SVM01\_npp\_d20130506\_t1956380\_e2013585\_...\_XXXX\_XXX.h5

Users often wonder why 28 output granules are produced in 2 files when aggregating 16 input granules with an aggregation size of 16. The reasons are that nagg creates aggregations according to the JPSS aggregation methodology[[1]](#footnote-1), dividing aggregations according to predetermined “bucket boundaries”, creating fill granules to fill spaces between input granules, not producing leading or trailing fill granules, and not producing files containing only fill granules. If a large aggregation size is used, a total of 76 granules will be produced, the first 8 input granules followed by 60 fill granules and finally by the second 8 input granules. Provided the aggregation size is larger than the total number of granules produced, the output may be in one file or in two files, depending on whether or not a bucket boundary for the aggregation size chosen falls within the aggregation.

The --onefile option will insure that all input granules are aggregated in one output file. Packaging will continue to be an independent option.

Command: nagg --onefile -t SVM01 GMODO-SVM01\_npp\_d20130506\_t\*

Output with --onefile option:

Produced 76 granules in GMODO-SVM01\_npp\_d20130506\_t1825525\_e2013585\_...\_XXXX\_XXX.h5

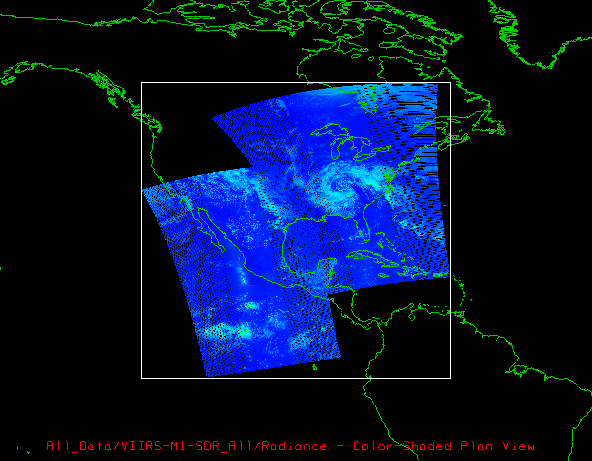
The user may not want the file to contain the 60 fill granules that represent the satellite data between the first 8 granules around the globe to the second 8 granules. The --nofill option will suppress the creation of these fill granules. The resulting output file will contain 16 granules that have an unmarked division. An application using such a file will need to detect the division from either the granule time data or the geo-location data.

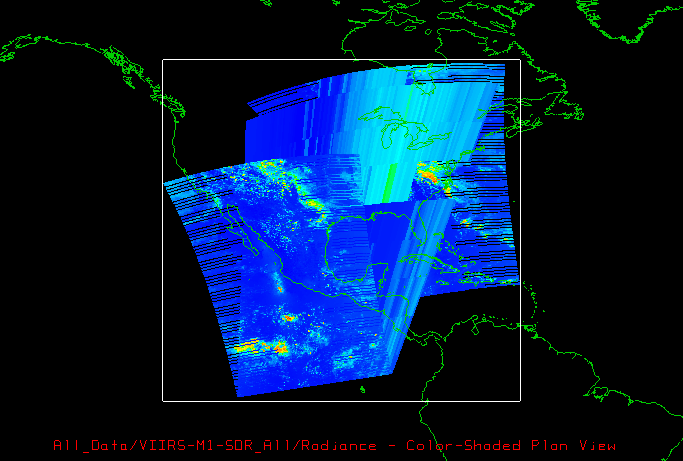
Command: nagg --onefile --nofill -t SVM01 GMODO-SVM01\_npp\_d20130506\_t\*

Output with --onefile --nofill options:

Produced 16 granules in GMODO-SVM01\_npp\_d20130506\_t1825525\_e2013585\_...\_XXXX\_XXX.h5

The 16 granules produced are displayed in the following images produced using IDV. The first image contains 60 fill granules. The second image has no fill granules, but their absence results in the diagonal streaks from the bottom of the left block of granules to the top of the right block.





# Revision History

|  |  |
| --- | --- |
| *December 29, 2014:January 27, 2015:* | Version 1 circulated for comment within The HDF Group.  Version 2 with --nofill option and example. |

1. Joint Polar Satellite System Common Data Format Control Book – External Volume I, Section 3.5.12 DDS Aggregation Methodology [↑](#footnote-ref-1)