
NPP Aggregation Tool for JPSS Product Files (nagg)

Version 1.2

May 2012



Copyright Notice and License Terms for HDF5 (Hierarchical Data Format 5) Software Library and Utilities

HDF5 (Hierarchical Data Format 5) Software Library and Utilities

Copyright 2006-2012 by The HDF Group.

NCSA HDF5 (Hierarchical Data Format 5) Software Library and Utilities

Copyright 1998-2006 by the Board of Trustees of the University of Illinois.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted for any purpose (including commercial purposes) provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the following disclaimer in the documentation and/or materials provided with the distribution.
3. In addition, redistributions of modified forms of the source or binary code must carry prominent notices stating that the original code was changed and the date of the change.
4. All publications or advertising materials mentioning features or use of this software are asked, but not required, to acknowledge that it was developed by The HDF Group and by the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign and credit the contributors.
5. Neither the name of The HDF Group, the name of the University, nor the name of any Contributor may be used to endorse or promote products derived from this software without specific prior written permission from The HDF Group, the University, or the Contributor, respectively.

DISCLAIMER:

THIS SOFTWARE IS PROVIDED BY THE HDF GROUP AND THE CONTRIBUTORS "AS IS" WITH NO WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED. In no event shall The HDF Group or the Contributors be liable for any damages suffered by the users arising out of the use of this software, even if advised of the possibility of such damage.

Contributors: National Center for Supercomputing Applications (NCSA) at the University of Illinois, Fortner Software, Unidata Program Center (netCDF), The Independent JPEG Group (JPEG), Jean-loup Gailly and Mark Adler (gzip), and Digital Equipment Corporation (DEC).

Portions of HDF5 were developed with support from the Lawrence Berkeley National Laboratory (LBNL) and the United States Department of Energy under Prime Contract No. DE-AC02-05CH11231.

Portions of HDF5 were developed with support from the University of California, Lawrence Livermore National Laboratory (UC LLNL). The following statement applies to those portions of the product and must be retained in any redistribution of source code, binaries, documentation, and/or accompanying materials:

This work was partially produced at the University of California, Lawrence Livermore National Laboratory (UC LLNL) under contract no. W-7405-ENG-48 (Contract 48) between the U.S. Department of Energy (DOE) and The Regents of the University of California (University) for the operation of UC LLNL.

DISCLAIMER:

This work was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately- owned rights. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

Contents

1. Introduction	5
1.1. What does nagg do?	5
1.2. Filenames, fill granules and bucket boundaries	6
2. Building, Installing, and Running nagg	7
2.1. Building and Testing the Program	7
2.2. Command Syntax	8
2.3. Command Line Options and Parameters:	10
3. Examples of nagg Usage.....	12
3.1. De-aggregating an aggregate file	13
3.2. Aggregating single granule files	16
3.3. Reaggregating to an aggregation number different from the aggregated input files	19
3.4. Packaging compatible products from single granule input files	21
3.5. Packaging compatible products from single granule input files – 2 input files missing	23
3.6. Packaging compatible products from single granule input files – with “-g strict” option	26
3.7. Unpackaging and de-aggregating packaged and aggregated input files	27
3.8. Extracting a product from packaged files containing other products	29
3.9. Aggregating or de-aggregating GEO files	30

1. Introduction

Nagg is a tool for aggregating JPSS data granules from existing files into new files with a different number of granules per file or different combinations of compatible products than in the original files. The tool was created to provide individual users the ability to rearrange NPP product data granules from downloaded files into new files with aggregations or packaging that are better suited as input for a particular application.

Nagg is needed to address the flexibility of NPP products, especially swaths products. A swath is a ribbon of data collected as the satellite orbit sweeps across the Earth. As an orbit is continuous, the swath is continuous, wrapping around the Earth over and over again like a yarn on a ball. In most NASA heritage missions, the ribbon of swath is snipped into equal size pieces called granules, and one granule is placed in each file. The size of the snipped piece, while usually set for each mission, varies from mission to mission based mostly on the amount of data and therefore the size of the file that can be conveniently manipulated.

For NPP, the concept of a granule is independent of the packaging of the granule in a file, so granules are made relatively small in order to conveniently manipulate them, but can be packaged into arbitrary size for the convenience of the particular application. For example, when comparing a MODIS 5 minute data granules to the VIIRS data, it is useful to package four VIIRS 86 second granules together to make the piece of swath similar in size to the MODIS granule. Some data analysis tools analyze by full orbit, so it is convenient to package OMPS data that way. Nagg makes it possible for the analyst to package NPP data into files in the way that best suits the tool to be used.

1.1. What does nagg do?

- Copies data granules from input files to new files with the number of data granules per file specified by -n. The number can be higher or lower or the same, limited by the current nagg limit of 10,000 granules for a nagg run on the high end and a minimum of 1 granule per file on the low end.
- Copies NPP data product granules and corresponding GEO product granules from separate input files to one output file.
- Copies NPP data product granules and corresponding GEO product granules from one input file to separate output files.
- Copies data granules from separate input files to new output files containing more than one compatible NPP product. Compatible products share geolocation data, which is optional for inclusion, but if geolocation data is included it will be copied to the output file.
- Copies data granules from one input file containing compatible NPP products to separate output files for each product. Geolocation data may be excluded.
- Copies geolocation granules corresponding to sensor data products to new geolocation files or groups. If geolocation granules are not available or unwanted, the product granules may be aggregated or packaged without geolocation with the “-g no” option.

- Copies geolocation granules alone to new geolocation files without product granules using the “-g <GEO product>” option.
- Nagg always copies existing data granules from input data files to new data files that it creates. No changes are made to input files.

1.2. Filenames, fill granules and bucket boundaries

- Output file names are determined by the products and timestamps of granules in the file, the creation time of the file, and the -O and -D command line parameters. See section 3.4.1, “File Naming Convention for NPP/NPOESS Data Products”, in the JPSS Common Data Format Control Book – External Volume I, p 22 for complete details.
- Fill granules are added according to section 3.5.6.1 of JPSS Common Data Format Control Book – External Volume I, p 106, whenever there is a gap larger than the size of a granule between 2 input granules, or to match a granule of another product at a given time in a packaged file. Fill granules are not added at the beginning or end of an aggregation, and files that would consist entirely of fill granules are not produced.
- The beginning and end of each aggregation of any particular size for each NPP product is predetermined by integer numbers of aggregation of that size since t0 (1/1/1958). As a result, aggregating a series of contiguous input granules the size of an aggregation or smaller may result in either 1 or 2 output files, depending on whether the series falls within the predetermined bucket boundaries or straddles a bucket boundary. See JPSS Common Data Format Control Book – External Volume I, pp. 131-133 for a detailed explanation of bucket boundaries.

For More Information

The nagg tool endeavors to produce files that match what the JPSS System provides as described in the Joint Polar Satellite System (JPSS) Common Data Format Control Book – External - Volumes I, III, IV and V. These documents may be found at <http://jointmission.gsfc.nasa.gov/science/documents.html>.

Comments

Send questions or comments to the HDF Group Help Desk via help@hdfgroup.org.

2. Building, Installing, and Running nagg

The following are some notes about how to build, install, and run nagg.

A copy of the nagg source files can be downloaded from
<ftp://ftp.hdfgroup.uiuc.edu/pub/outgoing/JPSS/source/NAGG/>.

nagg is a standalone tool that is run from the command line. The currently available version of the program can be built to work on 32- and 64-bit Linux systems.

The installation process is a file copy operation: copy the file to the directory where you want to keep the program.

In this version of the program, there is no file with configuration settings. To adjust the program's behavior, add one or more of the options to the command line statement when the program is executed. The available options are described in the "Command Syntax" section beginning on page 8.

To run the program, run it from the directory where it is kept, or put in the path the directory where the program is stored.

2.1. Building and Testing the Program

Here are some notes that will help you build and test the nagg program.

Before you start:

- Make sure HDF5 version 1.8.5 or later is installed on your system. Source code or binaries can be downloaded from <http://www.hdfgroup.org/HDF5/release/obtain5.html>.
- Install High-Level library for handling object and region references hdf5_hl_region v.1.1.1; the source code can be found at <ftp://ftp.hdfgroup.uiuc.edu/pub/outgoing/NPOESS/source/>. For installation instructions see the README.txt file in the source distribution.

To build:

Assuming gmake and h5cc (HDF5 C compiler) are in your \$PATH, just do

```
% ./configure --with-hlregion=/path-to-hdf5_hl_region-library  
% gmake
```

The default C compiler is set to h5cc.

If you want to use a specific h5cc compiler, do

```
% env CC=$mypath/h5cc ./configure
```

To build with different compiler flags, use the CFLAGS variable, e.g.

```
% env CFLAGS=-O2 ./configure
```

This command will set the compiling to "h5cc -O2".

To run tests:

```
% gmake check
```

To install:

```
% gmake install
```

The nagg install files go to nagg/bin of the build directory by default. If you want to install it somewhere else such as /usr/local/bin/nagg,

```
% gmake prefix=/usr/local install
```

2.2. Command Syntax

Tool Name: nagg

Syntax:

```
nagg -h  
nagg [-n number] [-t list | -l file] [-g criterion] [-S] [-A  
seconds]      [-d directory] [-O origin] [-D domain] INPUT...
```

Purpose:

nagg is NPP granule aggregation and packaging utility.

Description:

The command line utility nagg is the NPP granule aggregation utility for grouping NPP data record granules into aggregate and packaged files.

Aggregates include time contiguous and gap-filled series of the same type of granules together in the same file. Packages include different, but related type granule aggregates with same temporal and spatial extent together in the same file.

Input parameter `INPUT` is a list of one or more files.

The output of nagg is a file or set of files named according to the NPP file naming convention. Output files are aligned with the bucket boundaries as defined in the Control Book. Fill granules are added when there are no granules available for "slots" in a bucket. Leading and trailing fill granules are not generated for the first and last output files respectively. Therefore, first and last output files may be partial files that consist of fewer granules than requested.

The default behavior of nagg is to aggregate corresponding Geolocation granules in an external Geolocation file whose name is in the input file's `N_GEO_Ref` attribute, producing output Geolocation files with granules that correspond to those of the primary product. Nagg will fail if none of the specified input Geolocation files are available. This behavior can be overridden with

the “-g no” command option, which directs nagg to not use or look for Geolocation input granules or files and to not produce Geolocation output granules or files.

Geolocation file names can be approximately matched such that only the beginning parts up to creation date are matched. E.g., if the sensor data file defines its geolocation file as: “GMTCO_npp_d20100906_t0701368_e0703013_b00004_c20111024161933653314_noaa_ops.h5”, the tool will accept any file matching the following pattern and will use the one with the latest creation timestamp.

GMTCO_npp_d20100906_t0701368_e0703013_b00004_c*.h5

The “-g strict” command option will enforce that the external Geolocation file name must exactly match the N_GEO_Ref value.

nagg is non-destructive. It will not overwrite existing files.

Limits defined:

NAGG_Granule_info_max 10000	Maximum number of granules managed
NAGG_Product_list_max 30	Maximum number of products requested
NAGG_outputfiles_max 30	Maximum number of output files

See the NPP Aggregation Tool Components document Appendix 2 for a complete list of size definitions in the nagg tool.

Other considerations:

1. nagg can deaggregate files by setting -n 1 and operating on aggregate files.
2. nagg can aggregate files by setting -n N and operating on a directory of single granule files.
3. nagg can reaggregate by setting -n N to a number different from the number of granules in the input files.

2.3. Command Line Options and Parameters:

- h, --help
Print command syntax; also list valid types and packaging groupings.

- t *list*, --type=*list*
list specifies a comma separated list of NPP record type mnemonics. Unless -S is specified, the granule types will be packaged together. Types must be compatible to be packaged together. (Use -h to list valid package groupings). If -t is not specified, -g <product> must be given to aggregate only the Geolocation product granules.

- n *N*, --number=*N*
N is the number of granules of each product in each aggregate file; must be greater than zero. If not specified, default is 1 (single granule files).

- g *criterion*, --geolocation=*criterion*
criterion is the criterion for searching the Geolocation granules
 - no | 0: aggregate product files without Geolocation input or output
 - yes | 1: allow approximate matching of Geolocation input filenames (default)
 - strict | 2: require exact matching of geolocation input filenames
 <product>: aggregate Geolocation product <product> only. This excludes the use of -t.

- 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.

- A *seconds*
(To be supported in future implementation.)
The number of seconds in each aggregate file. Aggregation Size is a fixed number of granules per file equal to seconds divided by the period of the granule rounded up to the next integer number of granules
If -l, -t, -n and -A are omitted, then the first NPP data product file encountered will be used to determine the -t list and -n number.

- S, --simple
Simple aggregates are produced. Each type is packaged separately. Default is not set, that is, all types including geolocation products are packaged in one file.

- d *directory*, --directory=*directory*
(To be supported in future implementation.)
Directory where the output files should be placed. Default is present working directory.

- O *origin*
The origin identifier, a four-character string in the output filename. Only the first four characters will be used. If less than four characters, the origin will be pre-pended with the character "X". Default is "XXXX".

`-D domain`

The domain identifier, a three-character string in the output filename. Only the first three characters will be used. If less than three characters, the origin will be pre-pended with the character "x". Default is "XXX".

`--debug`

Read in all granules in the input files, including those not specified by the `-t` list. Display all the granules and end the execution without generating the normal output files.

Note that this option is intended for tool debugging. Files or output generated should not be used for production purpose. Its behavior may change from version to version.

INPUT

INPUT is a list of one or more NPP files.

Exit Status:

0 Succeeded.
>0 An error occurred.

3. Examples of nagg Usage

These examples demonstrate the use of available nagg command options in a variety of combinations. The input files for the examples are available in the test/testfiles directory of the nagg source files. The path in the commands will need modification depending on the directory chosen to run nagg.

The examples contain a table that shows how the granules are arranged in the input and output files. They also describe displaying the contents of the input and output files using the HDF5 utility h5ls, found in the bin directory of the HDF5 install. The command “h5ls -r <filename>” recursively displays the full group path to every object in the file. For dataset objects, the name is followed by the dataset dimensions in the form {<current dimension>/<maximum dimension>, ...} as shown in example 3.1. To see the differences described for the other examples, run “h5ls -r <filename>” or use HDFView or some other utility to examine the file contents.

A granule in the notes sections will often refer to both the sensor data granule and the geolocation granule. For example when in example 3.2 it is said that the first file has only 1 granule, the statement refers to both the first SATMS product file and also to the first GATMO product file with the corresponding geolocation data.

This section includes the following examples demonstrating various options.

Example	See this page:
De-aggregating an aggregate file with options -S -O -D (default -g yes)	13
Aggregating single granule files with options -n2 -S (default -g yes)	16
Reaggregating to an aggregation number different from the aggregated input files with options -n3 -S (default -g yes)	19
Packaging compatible products from single granule input files with options -n5 (defaults -g yes, packaged output)	21
Packaging compatible products from single granule input files – 2 input files missing with options -n5 (defaults -g yes, packaged output)	23
Packaging compatible products from single granule input files – with “-g strict option” with options -n5 -g strict	26
Unpackaging and de-aggregating packaged and aggregated input files with options -tSVM03,SVM04,SVM05 -S -g no	27
Extracting a product from packaged files containing other products with options – n4 -t SVM04 -g no	29
Aggregating or de-aggregating GEO files with option -g <GEO product>	30

3.1. De-aggregating an aggregate file

Command:

```
nagg -t SOMPS -S -O unkn -D dev
test/testfiles/SOMPS_npp_d20120508_t0333549
_e0336247_b02735_c20120522134023064953_unkn_XXX.h5
```

Example 3.1	Total granules	Granules per aggregation for each product	Files	Hyperslabs in each raw data dataset for a full aggregation
Input aggregations	8	4	2	4
Output aggregations	8	1	8	1

Command effects:

- The nagg command without -n <number> produces default aggregations of one granule each.
- -t SOMPS directs nagg to process SOMPS product granules
- nagg with no -g option specified processes SOMPS product granules and by default the corresponding GEO product granules either from the input files specified for the SOMPS product or from files matching the filename in the N_GEO_Ref attribute in those files. For the default behavior the creation field in the GEO filename is replaced with * for matching (GONPO_npp_d20120508_t0333549_e0336247_b02735_*_unkn_dev.h5).
- The -S option puts all products including the geolocation granules in separate files.
- -O unkn sets the origin field of the output filenames to “unkn”. The default if no -O option is provided is “XXXX”
- -D dev sets the domain description field of the output filenames to “dev”. The default without the -D option is “XXX”.
- Output: 4 files with 1 SOMPS product granule each and 4 files with 1 GONPO geo-location product granule each. The part of the output file names beginning with _c is determined by the creation time of the output file.

Notes:

The SOMPS input file has 4 SOMPS granules (see “Display of input file objects with h5ls -r, below”) and a reference to the geolocation file which contains the corresponding geolocation granules. In this case the 4 granules in the SOMPS file specified as input for the command plus 4 granules in the GONPO geolocation file it references result in 8 output files: 4 with one SOMPS granule each and 4 with the corresponding GONPO granules.

The contents of the input and one output files as displayed by “h5ls -r” are shown below. The input file has 4 granules (highlighted in blue), the output file shown has 1 granule (the other SOMPS output files have the same structure, each with one granule but with different data, and the 4 GONPO geolocation granules in the GONPO input file are similarly distributed into 4 output files, with a different collection of datasets for the geolocation data). The datasets in the input files have 4 hyperslabs each, 1 for each

granule. The datasets in the output files each have 1 of these hyperslabs for the file's 1 granule. Consequently the current dimensions (highlighted in yellow) for the first dimension of the datasets in the output files are 1/4 the size of the same dimension in the input files (also highlighted in yellow). The time range from t0333549 to e0336247 reflected in the input file names is distributed among the 4 output files for each product.

Display of input file objects with "h5ls -r

```
SOMPS_npp_d20120508_t0333549_e0336247_b02735_c20120522134023064953_unkn_xxx.h5"
```

```
/
/All_Data                               Group
/All_Data/OMPS-NP-SDR_All               Group
/All_Data/OMPS-NP-SDR_All/BadCal Dataset {4/Inf}
/All_Data/OMPS-NP-SDR_All/Bias1 Dataset {4/Inf}
/All_Data/OMPS-NP-SDR_All/Cal Dataset {20/Inf, 200/Inf}
/All_Data/OMPS-NP-SDR_All/DarkCurrentEarth Dataset {24/Inf, 200/Inf}
...
/Data_Products                          Group
/Data_Products/OMPS-NP-SDR              Group
/Data_Products/OMPS-NP-SDR/OMPS-NP-SDR_Aggr Dataset {21}
/Data_Products/OMPS-NP-SDR/OMPS-NP-SDR_Gran_0 Dataset {21/Inf}
/Data_Products/OMPS-NP-SDR/OMPS-NP-SDR_Gran_1 Dataset {21/Inf}
/Data_Products/OMPS-NP-SDR/OMPS-NP-SDR_Gran_2 Dataset {21/Inf}
/Data_Products/OMPS-NP-SDR/OMPS-NP-SDR_Gran_3 Dataset {21/Inf}
```

Display of objects in 1 of 8 output files with "h5ls -r

```
SOMPS_npp_d20120508_t0333549_e0334324_b02735_c20120509170435469674_xxxX_xxx.h5"
```

```
/
/All_Data                               Group
/All_Data/OMPS-NP-SDR_All               Group
/All_Data/OMPS-NP-SDR_All/BadCal Dataset {1/Inf}
/All_Data/OMPS-NP-SDR_All/Bias1 Dataset {1/Inf}
/All_Data/OMPS-NP-SDR_All/Cal Dataset {5/Inf, 200/Inf}
/All_Data/OMPS-NP-SDR_All/DarkCurrentEarth Dataset {6/Inf, 200/Inf}
...
/Data_Products                          Group
/Data_Products/OMPS-NP-SDR              Group
/Data_Products/OMPS-NP-SDR/OMPS-NP-SDR_Aggr Dataset {21}
/Data_Products/OMPS-NP-SDR/OMPS-NP-SDR_Gran_0 Dataset {21/Inf}
```

Input files (2):

```
SOMPS_npp_d20120508_t0333549_e0336247_b02735_c201205 22134023064953_unkn_xxx.h5
```

Geolocation file:

```
GONPO_npp_d20120508_t0333549_e0336247_b02735_c201205 22134023064953_unkn_dev.h5
```

Command runtime output:

Produced 1 granules in
SOMPS_npp_d20120508_t0333549_e0334324_b02735_c20120509174130237345_unk
n_dev.h5
Produced 1 granules in
GONPO_npp_d20120508_t0333549_e0334324_b02735_c20120509174130237345_unk
n_dev.h5
Produced 1 granules in
SOMPS_npp_d20120508_t0334324_e0335098_b02735_c20120509174130259000_unk
n_dev.h5
Produced 1 granules in
GONPO_npp_d20120508_t0334324_e0335098_b02735_c20120509174130259000_unk
n_dev.h5
Produced 1 granules in
SOMPS_npp_d20120508_t0335098_e0335472_b02735_c20120509174130278264_unk
n_dev.h5
Produced 1 granules in
GONPO_npp_d20120508_t0335098_e0335472_b02735_c20120509174130278264_unk
n_dev.h5
Produced 1 granules in
SOMPS_npp_d20120508_t0335473_e0336247_b02735_c20120509174130297695_unk
n_dev.h5
Produced 1 granules in
GONPO_npp_d20120508_t0335473_e0336247_b02735_c20120509174130297695_unk
n_dev.h5

3.2. Aggregating single granule files

Command:

```
nagg -n 2 -t SATMS -S test/testfiles/SATMS_npp_d20100906*.h5
```

Example 3.2	Total granules	Granules per aggregation for each product	Files (containing full and partial aggregations)	Hyperslabs in each raw data dataset for a full aggregation
Input aggregations	14	1	14	1
Output aggregations	14	2	8 (6 full, 2 partial)	2

Command effects:

- The nagg command with -n 2 produces aggregations of two granules each.
- -t SATMS directs nagg to process SATMS product granules
- nagg with no -g option specified processes SATMS product granules and by default the corresponding GEO product granules either from the input files specified for the SATMS product or from files matching the filename in the N_GEO_Ref attribute in those files. For the default behavior the creation date, origin and domain description fields in the GEO filename are replaced with * for matching files with the corresponding GEO product that have the same date, begin time, end time and orbit number (GATMO_npp_d20100906_t0750210_e0750524_b00005_*_noaa_ops.h5).
- The -S option puts all products including the geolocation granules in separate files.
- Output: 3 files with 2 SATMS product granules each and 3 files with 2 GATMO geo-location product granules each, 1 file with 1 SATMS granule and 1 file with 1 GATMO granule.

Notes:

Why does the first file have only 1 granule when the command option dictated 2 granules per file? nagg duplicates the behavior of the JPSS DDS, calculating predetermined bucket boundaries according to integer multiples of the aggregation size beginning at t0 (12:00 am, 1/1/1958). For a more detailed explanation, see section 1 of this document or section 3.5.12 of the Common Data Format Control Book - External, Volume I, page 129. Since the first granule in these input files is in the second of 2 granules in a bucket, and since nagg does not produce leading or trailing fill granules (see example 3.5) the first file has only the one granule.

The input file string “test/testfiles/SATMS_npp_d20100906*.h5” matches 7 SATMS files in the test/testfiles directory, and those 7 files each have an N_GEO_Ref attribute with the name of a GATMO file in the same directory. Unless “-g no” is supplied in the command line, nagg will aggregate the geolocation granules in those 7 files, provided that they exist. If all of the geolocation files are missing, nagg will fail. However, the product files can be aggregated without geolocation using the “-g no” option.

Occasionally the creation time field of the geolocation filenames will not match the strings in the product files’ N_GEO_Ref attributes. Nagg allows for this, using the last created GEO file that matches the rest of the string. Exact matches can be enforced with the “-g strict” option.

Input and output files may be examined using “h5ls -r” as shown in example 3.1. HDFview and h5dump are other HDF5 tools that may be useful for examining the contents of HDF5 files.

The effect on dataset dimensions is the reverse of that for example 3.1, since granules are being aggregated rather than deaggregated. With “-n 2” the datasets in the /All_Data group of the output files will have first dimensions that are twice the size of the first dimensions in the input files. This reflects the 2 hyperslabs from 2 granules in the complete aggregations.

Input files (14):

```
test/testfiles/SATMS_npp_d20100906_t0750210_e0750524_b00005_c20111024161819049592_noaa_
ops.h5
test/testfiles/SATMS_npp_d20100906_t0750530_e0751244_b00005_c20111024161819049592_noaa_
ops.h5
test/testfiles/SATMS_npp_d20100906_t0751250_e0751564_b00005_c20111024161819049592_noaa_
ops.h5
test/testfiles/SATMS_npp_d20100906_t0751570_e0752284_b00005_c20111024161819049592_noaa_
ops.h5
test/testfiles/SATMS_npp_d20100906_t0752290_e0753004_b00005_c20111024161819049592_noaa_
ops.h5
test/testfiles/SATMS_npp_d20100906_t0753010_e0753324_b00005_c20111024161819049592_noaa_
ops.h5
test/testfiles/SATMS_npp_d20100906_t0753330_e0754044_b00005_c20111024161819049592_noaa_
ops.h5
test/testfiles/GATMO_npp_d20100906_t0750210_e0750524_b00005_c20111024161819049592_noaa_
ops.h5
test/testfiles/GATMO_npp_d20100906_t0750530_e0751244_b00005_c20111024161819049592_noaa_
ops.h5
test/testfiles/GATMO_npp_d20100906_t0751250_e0751564_b00005_c20111024161819049592_noaa_
ops.h5
test/testfiles/GATMO_npp_d20100906_t0751570_e0752284_b00005_c20111024161819049592_noaa_
ops.h5
test/testfiles/GATMO_npp_d20100906_t0752290_e0753004_b00005_c20111024161819049592_noaa_
ops.h5
test/testfiles/GATMO_npp_d20100906_t0753010_e0753324_b00005_c20111024161819049592_noaa_
ops.h5
test/testfiles/GATMO_npp_d20100906_t0753330_e0754044_b00005_c20111024161819049592_noaa_
ops.h5
```

Command runtime output:

```
Produced    1 granules in
SATMS_npp_d20100906_t0750210_e0750524_b00005_c20120516210214226724_XXX
X_XXX.h5
Produced    1 granules in
GATMO_npp_d20100906_t0750210_e0750524_b00005_c20120516210214226724_XXX
X_XXX.h5
```

```
Produced 2 granules in
SATMS_npp_d20100906_t0750530_e0751564_b00005_c20120516210214283013_XXX
X_XXX.h5
Produced 2 granules in
GATMO_npp_d20100906_t0750530_e0751564_b00005_c20120516210214283013_XXX
X_XXX.h5
Produced 2 granules in
SATMS_npp_d20100906_t0751570_e0753004_b00005_c20120516210214381909_XXX
X_XXX.h5
Produced 2 granules in
GATMO_npp_d20100906_t0751570_e0753004_b00005_c20120516210214381909_XXX
X_XXX.h5
Produced 2 granules in
SATMS_npp_d20100906_t0753010_e0754044_b00005_c20120516210214481149_XXX
X_XXX.h5
Produced 2 granules in
GATMO_npp_d20100906_t0753010_e0754044_b00005_c20120516210214481149_XXX
X_XXX.h5
```

3.3. Reaggregating to an aggregation number different from the aggregated input files

Command:

```
nagg -n 3 -t SATMS -S SATMS_npp_d20100906*_XXXX_XXX.h5
```

Example 3.3	Total granules	Granules per aggregation for each product	Files (containing full and partial aggregations)	Hyperslabs in each raw data dataset for a full aggregation
Input aggregations	14	2	8 (6 full, 2 partial)	2
Output aggregations	14	3	6 (4 full, 2 partial)	3

Command effects:

- The nagg command with -n 3 produces aggregations of 3 granules each.
- -t SATMS directs nagg to process SSATMS product granules
- nagg without a -g option behaves the same as “-g yes” and processes the GATMO GEO product granules corresponding to the SATMS product either from the input files specified for the SATMS product or from files matching the filename in the N_GEO_Ref attribute in those files. For the default behavior the creation field in the GEO filename is replaced with * for matching (GATMO_npp_d20100906_t0750210_e0750524_b00005_*_XXXX_XXX.h5).
- The -S option puts all products including the geolocation granules in separate files.
- Output: 2 files with 3 SATMS product granules each and 2 files with 3 GATMO geo-location product granules each, 1 file with 1 SATMS granule and 1 file with 1 GATMO granule.

Notes:

This example uses the output files from example 3.2. As it happens, the granule bucket boundaries for 2 and 3 granule aggregations coincide between the 1st and 2nd granules in this set of granules, so the 1st file once again has only the last granule since nagg does not produce leading fill granules. For this set of granules the first granule is the only granule of a partial aggregation.

Input files (8):

```
SATMS_npp_d20100906_t0750210_e0750524_b00005_c20120516210214226724_XXXX_XXX.h5
SATMS_npp_d20100906_t0750530_e0751564_b00005_c20120516210214283013_XXXX_XXX.h5
SATMS_npp_d20100906_t0751570_e0753004_b00005_c20120516210214381909_XXXX_XXX.h5
SATMS_npp_d20100906_t0753010_e0754044_b00005_c20120516210214481149_XXXX_XXX.h5
GATMO_npp_d20100906_t0750210_e0750524_b00005_c20120516210214226724_XXXX_XXX.h5
GATMO_npp_d20100906_t0750530_e0751564_b00005_c20120516210214283013_XXXX_XXX.h5
GATMO_npp_d20100906_t0751570_e0753004_b00005_c20120516210214381909_XXXX_XXX.h5
GATMO_npp_d20100906_t0753010_e0754044_b00005_c20120516210214481149_XXXX_XXX.h5
```

Command runtime output:

```
Produced    1 granules in
SATMS_npp_d20100906_t0750210_e0750524_b00005_c20120516210313080299_XXX
X_XXX.h5
Produced    1 granules in
GATMO_npp_d20100906_t0750210_e0750524_b00005_c20120516210313080299_XXX
X_XXX.h5
Produced    3 granules in
SATMS_npp_d20100906_t0750530_e0752284_b00005_c20120516210313138410_XXX
X_XXX.h5
Produced    3 granules in
GATMO_npp_d20100906_t0750530_e0752284_b00005_c20120516210313138410_XXX
X_XXX.h5
Produced    3 granules in
SATMS_npp_d20100906_t0752290_e0754044_b00005_c20120516210313277669_XXX
X_XXX.h5
Produced    3 granules in
GATMO_npp_d20100906_t0752290_e0754044_b00005_c20120516210313277669_XXX
X_XXX.h5
```

3.4. Packaging compatible products from single granule input files

Command:

```
nagg -n 5 -t SATMS,TATMS test/testfiles/SATMS_npp_d2012040[34]*.h5
test/testfiles/TATMS_npp_d2012040[34]*.h5
```

Example 3.4	Total granules	Granules per aggregation for each product	Files (containing full and partial aggregations)	Hyperslabs in each raw data dataset for a full aggregation
Input aggregations	15	1	15	1
Output aggregations	15	5	2 partial	5

Command effects:

- The nagg command with -n 5 produces aggregations of 5 granules each.
- -t SATMS,TATMS indicates to nagg that those 2 products + their common geolocation are to be aggregated.
- nagg with no -g option processes granules for the products specified with -t and by default for the corresponding GEO product granules either from the input files specified for the -t products or from files matching the filename in the N_GEO_Ref attribute in those files. For the default behavior the creation field in the GEO filename is replaced with * for matching (GONPO_npp_d20120508_t0333549_e0336247_b02735*_unkn_dev.h5).
- Without the -S option all products including the geolocation granules are put in a single file for each aggregation.
- Output: 2 files, one with 4 SATMS and 4 TATMS product granule each and 4 GATMO geo-location product granules, the other with 1 granule of each product.

Notes:

The [34]* wildcards in the command input filenames matches all SATMS_npp_d20120403... and SATMS_npp_d20120404... filenames in the input directory, as well as those with the same filename patterns beginning with TATMS. The corresponding GATMO files are included based on the /N_GEO_Ref attributes in the files for both products. For packaged files with multiple products, only one corresponding GEO granule will be included in the output, as all packaged products must have the same corresponding GEO granules, and duplicate granules are added only once.

The command asked for a 5 granule aggregation. In this case, 5 input granules for each product were supplied, but a predetermined aggregation boundary for 5 granules of these products falls after the first 4 input granules. Since leading and trailing fill granules are not produced, the aggregation produces 2 files, the first with 4 sets of granules and the second with 1 set of granules.

Examining the contents of the input and output files will show that the first dimension of the datasets in the /All_Data group of the output file are 5 times the size of the first dimension of the datasets with the same name in the input files, since the output files raw data datasets contain hyperslabs for each of 5 granules, compared to 1 hyperslab in the input files' datasets.

Input files (15):

```
test/testfiles/SATMS_npp_d20120403_t2359123_e2359439_b02250_c20120405155447242391_noaa_
ops.h5
...
test/testfiles/SATMS_npp_d20120404_t0001203_e0001519_b02250_c20120405155447242391_noaa_
ops.h5
test/testfiles/TATMS_npp_d20120403_t2359123_e2359439_b02250_c20120405155457899734_noaa_
ops.h5
...
test/testfiles/TATMS_npp_d20120404_t0001203_e0001519_b02250_c20120405155457899734_noaa_
ops.h5
test/testfiles/GATMO_npp_d20120403_t2359123_e2359439_b02250_c20120405155441643487_noaa_
ops.h5
...
test/testfiles/GATMO_npp_d20120404_t0001203_e0001519_b02250_c20120405155441643487_noaa_
ops.h5
```

Command runtime output:

```
Produced    4 granules in SATMS-TATMS-
GATMO_npp_d20120403_t2359123_e0001199_b02250_c20120510175020763627_XXX
X_XXX.h5
Produced    1 granules in SATMS-TATMS-
GATMO_npp_d20120404_t0001203_e0001519_b02250_c20120510175020886072_XXX
X_XXX.h5
```

3.5. Packaging compatible products from single granule input files – 2 input files missing

Command:

```
nagg -n 5 -t SATMS,TATMS test/testfiles/SATMS_npp_d2012040[34]*.h5
test/testfiles/TATMS_npp_d20120404*.h5
```

Example 3.4.a	Total granules	Granules per aggregation for each product	Files (containing full and partial aggregations)	Hyperslabs in each raw data dataset for a full aggregation
Input aggregations	13	1	15	1
Output aggregations	15	5	2 partial	5

Command effects:

- The nagg command with -n 5 produces aggregations of 5 granules each.
- -t SATMS,TATMS indicates to nagg that those 2 products + their common geolocation are to be aggregated.
- nagg with no -g option processes granules for the products specified with -t and by default for the corresponding GEO product granules either from the input files specified for the -t products or from files matching the filename in the N_GEO_Ref attribute in those files. For the default behavior the creation field in the GEO filename is replaced with * for matching (GONPO_npp_d20120508_t0333549_e0336247_b02735*_unkn_dev.h5).
- Without the -S option all products including the geolocation granules are put in a single file for each aggregation.
- Output: 2 files, one with 4 SATMS and 4 TATMS product granule each and 4 GATMO geo-location product granules, the other with 1 granule of each product.

Notes:

The [34]* wildcards in the command input filenames matches all SATMS_npp_d20120403... and SATMS_npp_d20120404... filenames in the input directory. Those beginning with TATMS match only TATMS_npp_d20120404... filenames. The corresponding GATMO files are included as in the previous example, and the aggregation boundaries, numbers of granules in output files and dimensions of raw data datasets is the same.

The difference in output from the previous example is that with 5 input SATMS and GATMO granules but only 3 input TATMS granules, nagg produces 2 TATMS fill granules to complete the packaging. Nagg will also produce fill granules for granules missing from between granules in a temporal aggregation.

These fill granules can be identified by examining the output files to check the N_Granule_ID attributes of the granule datasets with HDF5 tools HDFview or h5dump as shown below. h5dump is another HDF5 utility found in the bin directory of the HDF5 install. Using h5dump to view the N_Granule_ID of the first SATMS granule and the first TATMS granule shows the difference in N_Granule_IDs for a real and a fill granule, the real granule ID being a 15 character string beginning with "NPP" while the fill granule ID is

"N/A". Fill granules will have the same amount of data as real granules, but the data will be fill values as specified in the JPSS Common Data Format Control Book.

Command:

```
h5dump -a /Data_Products/ATMS-SDR/ATMS-SDR_Gran_0/N_Granule_ID SATMS-
TATMS-
GATMO_npp_d20120403_t2359123_e0001199_b02250_c20120529123052142977_XXX
X_XXX.h5
```

Output:

```
HDF5 "SATMS-TATMS-
GATMO_npp_d20120403_t2359123_e0001199_b02250_c20120529123052142977_XXX
X_XXX.h5" {
  ATTRIBUTE "/Data_Products/ATMS-SDR/ATMS-SDR_Gran_0/N_Granule_ID" {
    ...
    DATA {
      (0,0) : "NPP000141695514"
    }
  }
}
```

Command:

```
h5dump -a /Data_Products/ATMS-TDR/ATMS-TDR_Gran_0/N_Granule_ID SATMS-
TATMS-
GATMO_npp_d20120403_t2359123_e0001199_b02250_c20120529123052142977_XXX
X_XXX.h5
```

Output:

```
HDF5 "SATMS-TATMS-
GATMO_npp_d20120403_t2359123_e0001199_b02250_c20120529123052142977_XXX
X_XXX.h5" {
  ATTRIBUTE "/Data_Products/ATMS-TDR/ATMS-TDR_Gran_0/N_Granule_ID" {
    ...
    DATA {
      (0,0) : "N/A"
    }
  }
}
```

Input files (13):

```
test/testfiles/SATMS_npp_d20120403_t2359123_e2359439_b02250_c20120405155447242391_noaa_
ops.h5
...
test/testfiles/SATMS_npp_d20120404_t0001203_e0001519_b02250_c20120405155447242391_noaa_
ops.h5
test/testfiles/TATMS_npp_d20120404_t0000163_e0000479_b02250_c20120405155457899734_noaa_
ops.h5
...
test/testfiles/TATMS_npp_d20120404_t0001203_e0001519_b02250_c20120405155457899734_noaa_
ops.h5
```


test/testfiles/GATMO_npp_d20120403_t2359123_e2359439_b02250_c20120405155441643487_noaa_
ops.h5

...

test/testfiles/GATMO_npp_d20120404_t0001203_e0001519_b02250_c20120405155441643487_noaa_
ops.h5

Command runtime output:

```
Produced    4 granules in SATMS-TATMS-  
GATMO_npp_d20120403_t2359123_e0001199_b02250_c20120529123052142977_XXX_  
X_XXX.h5  
Produced    1 granules in SATMS-TATMS-  
GATMO_npp_d20120404_t0001203_e0001519_b02250_c20120529123052282010_XXX_  
X_XXX.h5
```

3.6. Packaging compatible products from single granule input files – with “-g strict” option

Command:

```
nagg -n 5 -t SATMS,TATMS -g strict
test/testfiles/SATMS_npp_d2012040[34]*.h5
test/testfiles/TATMS_npp_d2012040[34]*.h5
```

This is the same command and same set of input files as example 3.4, but adding `-g strict` to the command will enforce that geofile names must exactly match those in the `/N_GEO_Ref` attribute of the SATMS and TATMS files. Since the name of the GATMO file with the GEO granule corresponding to the first SATMS or TATMS granule processed, `GATMO_npp_d20120403_t2359123_e2359439_b02250_c20120405155441643487_noaa_ops.h5`, has a creation time that doesn't match that in the attribute, the command fails. With “-g yes” (default) instead of “-g strict”, the creation time, origin and domain fields are masked when searching for the geo file named in the attribute. The “-g no option” processes the data files without processing GEO information.

Command runtime output:

```
nagg: ***ERROR*** find_geo_granules(): Failed to get geolocation
granules.
test/testfiles/GATMO_npp_d20120403_t2359123_e2359439_b02250_c201204051
55447242391_noaa_ops.h5 does not exist or otherwise cannot be accessed
for reading.
nagg: ***ERROR*** nagg_get_granules(): Failed to get geo-location
granules
nagg: ***ERROR*** nagg_get_granules(): no granules found for
geoproduct.
geoproduct: GATMO
nagg: ***ERROR*** Failure encountered in Get Granules.
```

3.7. Unpackaging and de-aggregating packaged and aggregated input files

Command:

```
nagg -t SVM03,SVM04,SVM05 -S -g no test/testfiles/SVM03-SVM04-
SVM05_npp_d20120424_t0010313_e0016117_b02534_c20120521190943233596_unk
n_XXX.h5
```

Example 3.5	Total granules	Granules per aggregation for each product	Files	Hyperslabs in each raw data dataset for a full aggregation
Input aggregations	12	4	1	4
Output aggregations	12	1	12	1

Command effects:

- The nagg command without -n <number> produces default aggregations of one granule each.
- The -S option puts all product granules in separate files.
- -t SVM03,SVM04,SVM05 indicates to nagg that granules of those products are to be deaggregated and unpackaged
- The “-g no” option tells nagg to ignore the corresponding geolocation granules, neither trying to find them for input nor to produce geolocation information in output files. This was done for this example because the GMODO geolocation files for VIIRS products are rather large for distributing with the source code. The “-g no” option may also be useful for using nagg with product files when the GEO files are unavailable.
- Output: 12 files, 4 for each of the three products.

Notes:

Examining the contents of the input and output files will show that the first dimension of the datasets in the All_Data group of the output file are 1/4 the size of the first dimension of the corresponding dataset in the input files. The input file’s raw data datasets contain hyperslabs for each of 4 granules, where the output files’ datasets contain a hyperslab for only 1 granule.

Input file:

```
test/testfiles/SVM03-SVM04-
SVM05_npp_d20120424_t0010313_e0016117_b02534_c20120521190943233596_unkn_XXX.h5
```

Command runtime output:

```
Produced 1 granules in
SVM03_npp_d20120424_t0010313_e0011555_b02534_c2
0120521192724535803_XXXX_XXX.h5
Produced 1 granules in
SVM04_npp_d20120424_t0010313_e0011555_b02534_c20120521192724535803_XXX
X_XXX.h5
```

Produced 1 granules in
SVM05_npp_d20120424_t0010313_e0011555_b02534_c20120521192724535803_XXX
X_XXX.h5
Produced 1 granules in
SVM03_npp_d20120424_t0011567_e0013209_b02534_c20120521192724857446_XXX
X_XXX.h5
Produced 1 granules in
SVM04_npp_d20120424_t0011567_e0013209_b02534_c20120521192724857446_XXX
X_XXX.h5
Produced 1 granules in
SVM05_npp_d20120424_t0011567_e0013209_b02534_c20120521192724857446_XXX
X_XXX.h5
Produced 1 granules in
SVM03_npp_d20120424_t0013221_e0014463_b02534_c20120521192725180119_XXX
X_XXX.h5
Produced 1 granules in
SVM04_npp_d20120424_t0013221_e0014463_b02534_c20120521192725180119_XXX
X_XXX.h5
Produced 1 granules in
SVM05_npp_d20120424_t0013221_e0014463_b02534_c20120521192725180119_XXX
X_XXX.h5
Produced 1 granules in
SVM03_npp_d20120424_t0014475_e0016117_b02534_c20120521192725504013_XXX
X_XXX.h5
Produced 1 granules in
SVM04_npp_d20120424_t0014475_e0016117_b02534_c20120521192725504013_XXX
X_XXX.h5
Produced 1 granules in SVM05_npp_d20120424_t0014475_e0016117_b0253
4_c20120521192725504013_XXXX_XXX.h5

3.8. Extracting a product from packaged files containing other products

Command:

```
nagg -n 4 -t SVM04 -g no test/testfiles/SVM03-SVM04-
SVM05_npp_d20120424_t0010313_e0016117_b02534_c20120521190943233596_unk
n_XXX.h5
```

Example 3.6	Total granules	Granules per aggregation for each product	Files	Hyperslabs in each raw data dataset for a full aggregation
Input aggregations	12	4	1	4
Output aggregations	4	4	1	4

Command effects:

- The nagg command with -n 4 produces aggregations of 4 granules each.
- -t SVM04 indicates to nagg that only granules product SVM04 are to be aggregated and packaged in the output file.
- The “-g no” option tells nagg to ignore the corresponding geolocation granules, neither trying to find them for input nor to produce geolocation information in output files. This was done for this example because the GMOD0 geolocation files for VIIRS products are rather large for distributing with the source code. The “-g no” option may also be useful for using nagg with product files when the GEO files are unavailable.
- Output: 1 file with 4 granules of the SVM04 product only.

Notes:

This example uses the same input file as example 3.7, but only one of the products is listed for the -t option. The aggregation size (4) is the same as in the input file, so nagg in effect copies the listed product only into the output file.

For this example the granules per aggregation of each product, and consequently the number of hyperslabs in each dataset are the same in the output file as in the input file. This time the total number of granules in the output file is smaller than the original because only one of three products is selected for aggregation in the output file.

Input file:

```
test/testfiles/SVM03-SVM04-
SVM05_npp_d20120424_t0010313_e0016117_b02534_c20120521190943233596_unkn_XXX.h5
```

Command runtime output:

```
Produced 4 granules in
SVM04_npp_d20120424_t0010313_e0016117_b02534_c20120521202951397965_XXX
X_XXX.h5
```

3.9. Aggregating or de-aggregating GEO files

Command:

```
nagg -n32 -g GCRI0 ../test/testfiles/GCRI0_npp_d20030125*.h5
```

Example 3.7	Total granules	Granules per aggregation for each product	Files (containing full and partial aggregations)	Hyperslabs in each raw data dataset for a full aggregation
Input aggregations	79	16	5 (4 full 1 partial)	16
Output aggregations	64	32	3 (1 full, 2 partial)	32

Command effects:

- The nagg command with -n 32 produces aggregations of 32 granules each.
- The -g <GEO product> option is for aggregating/deaggregating granules of a GEO product directly, without any corresponding sensor date product granules. -t is not used with -g <GEO product>
- Output: 3 files with 16 or 32 granules of the GCRI0 product only.

Notes:

This option is provided for processing aggregations of only GEO granules.

In this example the total number of granules in the output files is smaller than the total number of granule in the input files. The second input file with the name ending in anyt_ing.h5 contains duplicates of granules in the first and third input files. nagg selects only one of the duplicates for the output aggregation, eliminating the 15 duplicates.

Input files:

```
test/testfiles/GCRI0_npp_d20030125_t0702533_e0711257_b00014_c20111025170507184515_unkn_a
da.h5
test/testfiles/GCRI0_npp_d20030125_t0705359_e0713337_b00014_c98765432109876543210_anyt_in
g.h5
test/testfiles/GCRI0_npp_d20030125_t0711279_e0719577_b00014_c20111025170507194756_unkn_a
da.h5
test/testfiles/GCRI0_npp_d20030125_t0719599_e0728297_b00014_c20111025170507197657_unkn_a
da.h5
test/testfiles/GCRI0_npp_d20030125_t0728319_e0737011_b00014_c20111025170507201941_unkn_a
da.h5
```

Command runtime output:

```
Produced 16 granules in
GCRI0_npp_d20030125_t0702533_e0711257_b00993_c20120523155444015673_XXX
X_XXX.h5
```

```
Produced 32 granules in
GCRIO_npp_d20030125_t0711279_e0728297_b00014_c20120523155445226530_XXX
X_XXX.h5
Produced 16 granules in
GCRIO_npp_d20030125_t0728319_e0737011_b00014_c20120523155447399983_XXX
X_XXX.h5
```