NPP Aggregation Tool for JPSS Product Files (nagg)

Version 1.4.0

August 2012

The HDF Group

Copyright Notice and License Terms for HDF NPP Aggregation Tool (nagg)

The HDF Group

Contents

1. Introduction

This document provides a brief introduction to the `nagg` tool, its uses, how to build and install it, the `nagg` command syntax, and a variety of examples.

1.1. The NPP Granule Aggregation and Packaging Utility (nagg)

`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 for 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. Being able to rearrange granules means users only have to request and download data once.

`nagg` is needed to address the flexibility of NPP products, especially swath 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, each 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. Suppose an analyst needs to compare MODIS and VIIRS data. With `nagg`, four VIIRS 86 second granules can be packaged together to make the piece of swath similar in size to the MODIS 5 minute data granule. Some data analysis tools analyze by full orbit, so it is possible 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 being used.

1.2. nagg's Capabilities

The following is a list of `nagg`'s current and future capabilities:

- Copies data granules from input files to new files with the number of data granules per file specified by the `-n` command line option. The number data granules copied can range from 1 to 10,000.
- Copies data granules from input files to new files with the number of seconds of data provided by the granules in each file specified by the `-A` command line option. `nagg` will convert the number of seconds specified to the smallest integer number of data granules that will provide the seconds of data specified.

The HDF Group

- Creates output files in the current directory or in a directory specified by the −d command line option.
- Copies NPP data product granules and corresponding geolocation product granules from separate input files to one output file.
- Copies NPP data product granules and corresponding geolocation 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 command line option.
- Copies geolocation granules alone to new geolocation files without product granules using the −g <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.

See the "Command Line Options and Parameters" section on page 10 for more information.

1.3. Notes

The following are some notes regarding output file names, fill granules, and bucket boundaries.

- **Output file names** are determined by the products and timestamps of the granules in the file, the creation time of the file, and the –O and –D command line options. 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 more information.
- **Fill granules** are added whenever there is a gap larger than the size of a granule between two input granules. Fill granules might also be used 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. See section 3.5.6.1 of *JPSS Common Data Format Control Book – External Volume I*, p. 106 for more information.
- 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). Aggregating a series of contiguous input granules the size of an aggregation or smaller may result in either one or two 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 more information.

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.

Run the program 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 the 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. An example is

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

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

To run tests:

To run tests, use the following:

```
% gmake check
```

To install:

To install, use the following:

```
% gmake install
```

The `nagg` install files go to `nagg/bin` of the build directory by default. If you want to install it somewhere else, for example `/usr/local/bin/nagg`, use the following:

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

2.2. Command Syntax

The command line syntax for the `nagg` tool is described below. See the "Command Line Options and Parameters" section on page 10 for more information.

Tool Name: `nagg`

Syntax

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

Purpose

To manipulate granules stored in NPP files into more easily processed NPP files.

Description

The command line utility `nagg` is the utility for grouping NPP granules into aggregate and package 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 the same temporal and spatial extent together in the same file.

Input parameter `INPUT` is a list of one or more NPP files with the granules to aggregate and/or package.

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 boundary. For more information on bucket boundaries, see *JPSS Common Data Format Control Book – External Volume I*, pp. 131-133. Fill granules are generated when there are no data granules available in the input files 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 contain fewer granules than requested.

The default behavior of `nagg` is to aggregate corresponding geolocation granules stored in an external geolocation file(s). The names of the geolocation file(s) can be found from the value(s) of an N_GEO_Ref attribute in the input file. The names of the output geolocation file(s) correspond to those of the primary product. `nagg` will fail if the input geolocation file is unavailable. The aggregating of geolocation granules 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 approximate such that only the beginning parts up to the creation date are matched. Suppose the sensor data file defines its geolocation file as: `GMTCO_npp_d20100906_t0701368_e0703013_b00004_c20110124161933653314_noaa_ops.h5`. `nagg` 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 can be used to tell `nagg` that the external geolocation file name must exactly match the N_GEO_Ref value.

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

Tool Limits

Some limits for the `nagg` tool are defined below.

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 "Appendix 2" in the *NPP Aggregation Tool Components* document for a complete list of size definitions.

Notes

The following are some notes about aggregating and de-aggregating files:

- `nagg` can de-aggregate files by setting the command line option `-n` to `1` and operating on aggregate files.

- `nagg` can aggregate files by setting the command line option `-n` to `N` and operating on a directory of single granule files.
- `nagg` can re-aggregate files by setting the command line option `-n` to a number different from the number of granules in the input files.

2.3. Command Line Options and Parameters

The command line options and parameters available with the `nagg` tool are described below. See the "Command Syntax" section above for more information.

`-h, --help`

> **Help**. Using this command line option will display the command syntax. This can also be used to list valid granule types and packaging groupings.

`--version`

> Displays the version of the nagg tool.

`-t list, --type=list`

> **Granule Type**. `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`

> **Granule Number**. `N` is the number of granules of each product in each aggregate file and must be greater than zero. If this option not specified, the default value of 1 (single granule files) will be used.

`-g criterion, --geolocation=criterion`

> **Geolocation Criterion**. `criterion` is the criterion used for searching the geolocation granules. Valid values for `criterion` are the following:

> | | |
> |---|---|
> | `no \| 0` | Aggregate product files without geolocation input or output |
> | `yes \| 1` | Allow approximate matching of geolocation input file names (default) |
> | `strict \| 2` | Require exact matching of geolocation input file names |
> | `<product>` | Aggregate geolocation product *<product>* only. This excludes the use of –t. |

`-l file`

> **Example File**. `file` is used by `nagg` as an example of how to process data. Options on the command line override the example file.

`-A seconds`

> **Aggregation Size**. `seconds` is the number of seconds in each aggregate file. The aggregation's 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. The formula for this is

The HDF Group

NumGranules/File = *seconds*/GranulePeriod

where

NumGranules/File is the number of granules in the aggregated file,

seconds is the number of seconds that are used to determine the size of the aggregated file, and

GranulePeriod is the number of seconds worth of data held by each granule.

If, for example, *seconds* is set to 60 and GranulePeriod is .7 seconds per granule, then the following steps show the calculation:

Calculation	Comments
NumGranules/File = 60 seconds/.7seconds/granule	This is the setting of this parameter divided by the granule period.
NumGranules/File = 85.71 granules	This is the actual value of the calculation.
NumGranules/File = 86 granules	This is the value of the calculation rounded up to the next integer.

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.

This command line option has not yet been implemented.

`-S, --simple`
Simple Aggregates. Simple aggregates are produced. Each type is packaged separately. All granule types including geolocation products are packaged in one file.

`-d directory, --directory=directory`
Output Directory. Directory where the output files should be placed. The default is the current working directory. The tool will fail if the directory does not exist.

`-O origin`
Origin Identifier. The origin identifier is a four-character string in the output file name. Only the first four characters will be used. If fewer than four characters are specified, the origin will be pre-pended with the character "X". The default is "XXXX".

`-D domain`
Domain Identifier. The domain identifier is a three-character string in the output file name. Only the first three characters will be used. If fewer than three characters are specified, the domain will be pre-pended with the character "X". The default is "XXX".

`--debug`

Debug Mode. In debug mode, all of the granules in the input files including those not specified by the $-t$ list are read in, all of the granules are displayed, and the execution is ended 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 purposes: the behavior of debug mode 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

The examples in this chapter demonstrate the use of `nagg` command line 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`.

Each example has a number of sub-sections. The sub-sections that are included vary by the action of the example. All of the examples have the following sub-sections: "Command," "Command Line Elements," "Output," "Data Granules in Input and Output Files," and "Notes." The "Command" sub-section shows the command line used in the example. The "Command Line Elements" sub-section describes how each element in the command line effects the results. The "Output" sub-section describes the results of the command line. The "Data Granules in Input and Output Files" sub-section has a table that summarizes how the data granules are arranged in the input and output files. The "Notes" sub-section has items worth paying attention to.

The examples might have other sub-sections that further describe the input files, the output files, or any status messages produced by `nagg` during processing.

The contents of the input and output files can be viewed with the HDF5 utility `h5ls`. `h5ls` can be 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>, ...}. See the "De-aggregating an Aggregate File" section on page 18 for an example. 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" sub-sections will often refer to both the sensor data granule and the geolocation granule. See the example in the "Aggregating Single Granule Files" section on page 15: when it says that the first file has only one granule, the statement refers to both the first SATMS product file and also to the first GATMO product file with the corresponding geolocation data.

The table below lists the examples in this chapter.

Example	See this page:
Aggregating single granule files	15
De-aggregating an aggregate file	18
Re-aggregating by number of data granules	22
Re-aggregating by number of seconds of data	24
Packaging compatible products from single granule input files	26
Packaging compatible products from single granule input files and fill granule files	28
Packaging compatible products from single granule input files using exact matching of geolocation input file names	32
Unpackaging and de-aggregating packaged and aggregated input files	33

The HDF Group

Example	See this page:
Extracting a product from packaged files containing other products	36
Aggregating or de-aggregating geolocation files	38
Re-aggregating granules from input files like an example file	39
-with -n command line override	41
-with -A command line override	43
-with -t command line override	44

3.1. Aggregating Single Granule Files

This example shows how to use `nagg` to aggregate single granule files.

Command

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

Command Line Elements

- 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 geolocation product granules either from the input files specified for the SATMS product or from files matching the file name in the N_GEO_Ref attribute in those files. For the default behavior, the creation date, origin, and domain description fields in the geolocation file name are replaced with * for matching files with the corresponding geolocation products 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

The output of this command is three full files with two SATMS product granules each, three full files with two GATMO geolocation product granules each, one partial file with one SATMS granule, and one partial file with one GATMO granule.

Data Granules in Input and Output Files

Aggregations	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

Notes

Why does the first file have only one granule when the command line option dictated two granules per file? `nagg` duplicates the behavior of the JPSS DDS: it calculates 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 the "Notes" section on page 5 of this document or section 3.5.12 of the *JPSS Common Data Format Control Book - External, Volume I*, page 129. Since the first granule in these input files is in the second of two granules in a bucket, and since `nagg` does not produce leading or trailing fill granules, the first file has only the one granule. See the "Packaging Compatible Products from Single Granule Input Files and Fill Granule Files" section on page 28 for more information.

The input file string `test/testfiles/SATMS_npp_d20100906*.h5` matches seven SATMS files in the test/testfiles directory, and those seven 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 seven 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 file names will not match the strings in the product files' N_GEO_Ref attributes. `nagg` handles this difficulty by using the last created geolocation 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 on page 19. HDFview and h5dump are other HDF5 tools that may be useful for examining the contents of HDF5 files.

The effect on dataset dimensions in this example is the reverse of that for the example in the "3.2. De-aggregating an Aggregate File" section starting on page 18: granules in this example are being aggregated rather than de-aggregated. 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 two hyperslabs from two granules in the complete aggregations.

Input Files (14)

The input files are listed below.

```
test/testfiles/SATMS_npp_d20100906_t0750210_e0750524_b00005_c2011102416181904
9592_noaa_ops.h5
test/testfiles/SATMS_npp_d20100906_t0750530_e0751244_b00005_c2011102416181904
9592_noaa_ops.h5
test/testfiles/SATMS_npp_d20100906_t0751250_e0751564_b00005_c2011102416181904
9592_noaa_ops.h5
test/testfiles/SATMS_npp_d20100906_t0751570_e0752284_b00005_c2011102416181904
9592_noaa_ops.h5
test/testfiles/SATMS_npp_d20100906_t0752290_e0753004_b00005_c2011102416181904
9592_noaa_ops.h5
test/testfiles/SATMS_npp_d20100906_t0753010_e0753324_b00005_c2011102416181904
9592_noaa_ops.h5
test/testfiles/SATMS_npp_d20100906_t0753330_e0754044_b00005_c2011102416181904
9592_noaa_ops.h5
```

```
test/testfiles/GATMO_npp_d20100906_t0750210_e0750524_b00005_c2011102416181904
9592_noaa_ops.h5
test/testfiles/GATMO_npp_d20100906_t0750530_e0751244_b00005_c2011102416181904
9592_noaa_ops.h5
test/testfiles/GATMO_npp_d20100906_t0751250_e0751564_b00005_c2011102416181904
9592_noaa_ops.h5
test/testfiles/GATMO_npp_d20100906_t0751570_e0752284_b00005_c2011102416181904
9592_noaa_ops.h5
test/testfiles/GATMO_npp_d20100906_t0752290_e0753004_b00005_c2011102416181904
9592_noaa_ops.h5
test/testfiles/GATMO_npp_d20100906_t0753010_e0753324_b00005_c2011102416181904
9592_noaa_ops.h5
test/testfiles/GATMO_npp_d20100906_t0753330_e0754044_b00005_c2011102416181904
9592_noaa_ops.h5
```

Command Runtime Output

The status messages produced by the program during processing are listed below.

```
Produced   1 granules in
SATMS_npp_d20100906_t0750210_e0750524_b00005_c20120516210214226724_XXXX_XXX.h
5
Produced   1 granules in
GATMO_npp_d20100906_t0750210_e0750524_b00005_c20120516210214226724_XXXX_XXX.h
5
Produced   2 granules in
SATMS_npp_d20100906_t0750530_e0751564_b00005_c20120516210214283013_XXXX_XXX.h
5
Produced   2 granules in
GATMO_npp_d20100906_t0750530_e0751564_b00005_c20120516210214283013_XXXX_XXX.h
5
Produced   2 granules in
SATMS_npp_d20100906_t0751570_e0753004_b00005_c20120516210214381909_XXXX_XXX.h
5
Produced   2 granules in
GATMO_npp_d20100906_t0751570_e0753004_b00005_c20120516210214381909_XXXX_XXX.h
5
Produced   2 granules in
SATMS_npp_d20100906_t0753010_e0754044_b00005_c20120516210214481149_XXXX_XXX.h
5
Produced   2 granules in
GATMO_npp_d20100906_t0753010_e0754044_b00005_c20120516210214481149_XXXX_XXX.h
5
```

The HDF Group

3.2. De-aggregating an Aggregate File

This example shows how to use `nagg` to de-aggregate an aggregate file.

Command

```
nagg -t SOMPS -S -O unkn -D dev test/testfiles/SOMPS_npp_d20120508_t0333549
_e0336247_b02735_c201205 22134023064953_unkn_xxx.h5
```

Command Line Elements

- 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 geolocation product granules either from the input files specified for the SOMPS product or from files matching the file name in the N_GEO_Ref attribute in those files. For the default behavior, the creation field in the geolocation file name 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 file names to "unkn." The default if no `-O` option is provided is "XXXX."
- `-D dev` sets the domain description field of the output file names to "dev." The default without the `-D` option is "XXX."

Output

The output of this command is four files with one SOMPS product granule each and four files with one GONPO geolocation product granule each. The part of the output file names beginning with _c is determined by the creation time of the output file.

Data Granules in Input and Output Files

Aggregations	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

Notes

The SOMPS input file has four SOMPS granules (see the "Input File Objects" section below) and a reference to the geolocation file which contains the corresponding geolocation granules. In this case, the four granules in the SOMPS file specified as input for the command plus four granules in the GONPO geolocation file it references result in eight output files: four with one SOMPS granule each and four 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 four granules (highlighted in blue and bold), the output file shown has one granule. The other SOMPS output files have the same structure, each with one granule but with different data, and the four GONPO geolocation granules in the GONPO input file are similarly distributed into four output files with a different collection of datasets for the geolocation data. The datasets in the input files have four hyperslabs each, one for each granule. The datasets in the output files each have one of these hyperslabs for the file's one granule. Consequently, the current dimensions (highlighted in yellow and underlined) for the first dimension of the datasets in the output files are one quarter the size of the same dimension in the input files (also highlighted in yellow and underlined). The time range from t0333549 to e0336247 reflected in the input file names is distributed among the four output files for each product.

Input File Objects

The listing of input file objects with the `h5ls -r` SOMPS_npp_d20120508_t0333549_e0336247_b02735_c20120522134023064953_unkn_xxx.h5 command line is shown below.

```
/                          Group
/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}
```

Output File Objects

The listing of objects in 1 of 8 output files with the `h5ls -r`
`SOMPS_npp_d20120508_t0333549_e0334324_b02735_c20120509170435469674_XXXX_XXX.h`
`5` command line is shown below.

```
/                            Group
/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 Data File

The input data file is listed below.

```
SOMPS_npp_d20120508_t0333549_e0336247_b02735_c201205
22134023064953_unkn_xxx.h5
```

Input Geolocation File

The input geolocation data file is listed below.

```
GONPO_npp_d20120508_t0333549_e0336247_b02735_c201205
22134023064953_unkn_dev.h5
```

Command Runtime Output

The status messages produced by the program during processing are listed below.

```
Produced   1 granules in
SOMPS_npp_d20120508_t0333549_e0334324_b02735_c20120509174130237345_unkn_dev.h
5
Produced   1 granules in
GONPO_npp_d20120508_t0333549_e0334324_b02735_c20120509174130237345_unkn_dev.h
5
Produced   1 granules in
SOMPS_npp_d20120508_t0334324_e0335098_b02735_c20120509174130259000_unkn_dev.h
5
```

```
Produced   1 granules in
GONPO_npp_d20120508_t0334324_e0335098_b02735_c20120509174130259000_unkn_dev.h
5
Produced   1 granules in
SOMPS_npp_d20120508_t0335098_e0335472_b02735_c20120509174130278264_unkn_dev.h
5
Produced   1 granules in
GONPO_npp_d20120508_t0335098_e0335472_b02735_c20120509174130278264_unkn_dev.h
5
Produced   1 granules in
SOMPS_npp_d20120508_t0335473_e0336247_b02735_c20120509174130297695_unkn_dev.h
5
Produced   1 granules in
GONPO_npp_d20120508_t0335473_e0336247_b02735_c20120509174130297695_unkn_dev.h
5
```

The HDF Group

3.3. Re-aggregating by Number of Data Granules

This example shows how to use `nagg` to re-aggregate files by changing the number of data granules in each aggregated file from the original aggregated file.

Command

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

Command Line Elements

- The `nagg` command with `-n 3` produces aggregations of three granules each.
- `-t SATMS` directs `nagg` to process SATMS product granules.
- `nagg` without a `-g` option behaves the same as "`-g yes`" and processes the GATMO geolocation product granules corresponding to the SATMS product either from the input files specified for the SATMS product or from files matching the file name in the N_GEO_Ref attribute in those files. For the default behavior, the creation field in the geolocation file name 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

The output of this command is two full files with three SATMS product granules each, two full files with three GATMO geolocation product granules each, one partial file with one SATMS granule, and one partial file with one GATMO granule.

Data Granules in Input and Output Files

Aggregations	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

Notes

This example uses the output files from the example in the "Aggregating Single Granule Files" section on page 15. As it happens, the granule bucket boundaries for two and three granule aggregations coincide between the first and second granules in this set of granules, so the first 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)

The input files are listed below.

```
SATMS_npp_d20100906_t0750210_e0750524_b00005_c20120516210214226724_XXXX_XXX.h
5
SATMS_npp_d20100906_t0750530_e0751564_b00005_c20120516210214283013_XXXX_XXX.h
5
SATMS_npp_d20100906_t0751570_e0753004_b00005_c20120516210214381909_XXXX_XXX.h
5
SATMS_npp_d20100906_t0753010_e0754044_b00005_c20120516210214481149_XXXX_XXX.h
5
GATMO_npp_d20100906_t0750210_e0750524_b00005_c20120516210214226724_XXXX_XXX.h
5
GATMO_npp_d20100906_t0750530_e0751564_b00005_c20120516210214283013_XXXX_XXX.h
5
GATMO_npp_d20100906_t0751570_e0753004_b00005_c20120516210214381909_XXXX_XXX.h
5
GATMO_npp_d20100906_t0753010_e0754044_b00005_c20120516210214481149_XXXX_XXX.h
5
```

Command Runtime Output

The status messages produced by the program during processing are listed below.

```
Produced   1 granules in
SATMS_npp_d20100906_t0750210_e0750524_b00005_c20120516210313080299_XXXX_XXX.h
5
Produced   1 granules in
GATMO_npp_d20100906_t0750210_e0750524_b00005_c20120516210313080299_XXXX_XXX.h
5
Produced   3 granules in
SATMS_npp_d20100906_t0750530_e0752284_b00005_c20120516210313138410_XXXX_XXX.h
5
Produced   3 granules in
GATMO_npp_d20100906_t0750530_e0752284_b00005_c20120516210313138410_XXXX_XXX.h
5
Produced   3 granules in
SATMS_npp_d20100906_t0752290_e0754044_b00005_c20120516210313277669_XXXX_XXX.h
5
Produced   3 granules in
GATMO_npp_d20100906_t0752290_e0754044_b00005_c20120516210313277669_XXXX_XXX.h
5
```

3.4. Re-aggregating by Number of Seconds of Data

This example shows how to use `nagg` to re-aggregate data by number of seconds of data rather than by number of data granules.

Command

```
nagg –A 300 –t REDRO –d npp-out test/testfiles/REDRO_npp_d20030125*.h5
```

Command Line Elements

- The `nagg` command with `–A 300` for the REDRO product produces aggregations of ten granules each. This is determined by dividing 300 by the 31.997 second granule duration and rounding up to the next integer (10).
- `–t REDRO` directs `nagg` to process REDRO product granules.
- `nagg` without a `–g` option behaves the same as "`–g yes`" and processes the GCRIO geolocation product granules corresponding to the REDRO product either from the input files specified for the REDRO product or from files matching the file name in the N_GEO_Ref attribute in those files. For the default behavior, the creation field in the geolocation file name is replaced with * for matching (`GCRIO_npp_ d20030125_t0702533_e0711257_b00014_*_unkn_ada.h5`).
- Without the `–S` option the SDR data granules and the corresponding geolocation granules are put in one output file for each aggregation.
- With the `–d <directory>` command line option, the output files are created in the specified npp-out directory.

Output

The output of this command is six full files with ten REDRO product granules and ten GCRIO geolocation granules in each file and one partial file with four REDRO product granules and four GCRIO geolocation product granules.

Data Granules in Input and Output Files

Aggregations	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	128	16	8	16
Output aggregations	128	10	7 (6 full, 1 partial)	10

Notes

For this set of files, the first granule in the first input file happens to be the first granule in a bucket for a 16 granule aggregation and also for a 10 granule aggregation. The last file is a partial aggregation with only four granules.

Either relative or absolute paths are allowed for the directory specified by -d.

Input Files (8)

The input files are listed below.

```
test/testfiles/REDRO_npp_d20030125_t0702533_e0711257_b00014_c2011102517050719
0780_unkn_ada.h5
test/testfiles/REDRO_npp_d20030125_t0711279_e0719577_b00014_c2011102517050719
5067_unkn_ada.h5
test/testfiles/REDRO_npp_d20030125_t0719599_e0728297_b00014_c2011102517050719
7969_unkn_ada.h5
test/testfiles/REDRO_npp_d20030125_t0728319_e0737011_b00014_c2011102517050720
4040_unkn_ada.h5
test/testfiles/GCRIO_npp_d20030125_t0702533_e0711257_b00014_c2011102517050718
4515_unkn_ada.h5
test/testfiles/GCRIO_npp_d20030125_t0711279_e0719577_b00014_c2011102517050719
4756_unkn_ada.h5
test/testfiles/GCRIO_npp_d20030125_t0719599_e0728297_b00014_c2011102517050719
7657_unkn_ada.h5
test/testfiles/GCRIO_npp_d20030125_t0728319_e0737011_b00014_c2011102517050720
1941_unkn_ada.h5
```

Command Runtime Output

The status messages produced by the program during processing are listed below.

```
Produced  10 granules in npp-out/GCRIO-REDRO
_npp_d20030125_t0702533_e0708137_b00993_c20120712162901513435_XXXX_XXX.h5
Produced  10 granules in npp-out/GCRIO-REDRO_
npp_d20030125_t0708159_e0713337_b00014_c20120712162901631182_XXXX_XXX.h5
Produced  10 granules in npp-out/GCRIO-REDRO
_npp_d20030125_t0713359_e0718537_b00014_c20120712162901795401_XXXX_XXX.h5
Produced  10 granules in npp-out/GCRIO-REDRO_
npp_d20030125_t0718559_e0724137_b00014_c20120712162901920678_XXXX_XXX.h5
Produced  10 granules in npp-out/GCRIO-REDRO
_npp_d20030125_t0724159_e0729337_b00014_c20120712162902111364_XXXX_XXX.h5
Produced  10 granules in npp-out/GCRIO-REDRO
_npp_d20030125_t0729359_e0734531_b00014_c20120712162902271830_XXXX_XXX.h5
Produced   4 granules in npp-out/GCRIO-REDRO
_npp_d20030125_t0734531_e0737011_b00993_c20120712162902398737_XXXX_XXX.h5
```

3.5. Packaging Compatible Products from Single Granule Input Files

This example shows how to use `nagg` to package 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
```

Command Line Elements

- The `nagg` command with `-n 5` produces aggregations of five granules each.
- `-t SATMS,TATMS` indicates to `nagg` that those two products and their common geolocation granules are to be aggregated.
- `nagg` with no `-g` option processes granules for the products specified with `-t` and by default for the corresponding geolocation product granules either from the input files specified for the `-t` products or from files matching the file name in the N_GEO_Ref attribute in those files. For the default behavior, the creation field in the geolocation file name 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

The output of this command is one partial file with four SATMS and four TATMS product granules and four GATMO geolocation product granules and one other partial file with one granule of each product.

Data Granules in Input and Output Files

Aggregations	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

The HDF Group

Notes

The [34]* wildcards in the `-t` option in the command line matches all SATMS_npp_d20120403… and SATMS_npp_d20120404… file names in the input directory; the wildcards also match those with the same file name 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 geolocation granule will be included in the output: all packaged products must have the same corresponding geolocation granules, and duplicate granules are added only once.

The command asked for a five granule aggregation. In this case, five input granules for each product were supplied, but a predetermined aggregation boundary for five granules of these products falls after the first four input granules. Since leading and trailing fill granules are not produced, the aggregation produces two files: the first with four sets of granules and the second with one 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 five times the size of the first dimension of the datasets with the same name in the input files. The output files raw data datasets contain hyperslabs for each of five granules compared to one hyperslab in the input files' datasets.

Input Files (15)

The input files are listed below.

```
test/testfiles/SATMS_npp_d20120403_t2359123_e2359439_b02250_c2012040515544724
2391_noaa_ops.h5
…
test/testfiles/SATMS_npp_d20120404_t0001203_e0001519_b02250_c2012040515544724
2391_noaa_ops.h5
test/testfiles/TATMS_npp_d20120403_t2359123_e2359439_b02250_c2012040515545789
9734_noaa_ops.h5
…
test/testfiles/TATMS_npp_d20120404_t0001203_e0001519_b02250_c2012040515545789
9734_noaa_ops.h5
test/testfiles/GATMO_npp_d20120403_t2359123_e2359439_b02250_c2012040515544164
3487_noaa_ops.h5
…
test/testfiles/GATMO_npp_d20120404_t0001203_e0001519_b02250_c2012040515544164
3487_noaa_ops.h5
```

Command Runtime Output

The status messages produced by the program during processing are listed below.

```
Produced   4 granules in SATMS-TATMS-
GATMO_npp_d20120403_t2359123_e0001199_b02250_c20120510175020763627_XXXX_XXX.h
5
Produced   1 granules in SATMS-TATMS-
GATMO_npp_d20120404_t0001203_e0001519_b02250_c20120510175020886072_XXXX_XXX.h
5
```

3.6. Packaging Compatible Products from Single Granule Input Files and Fill Granule Files

This example shows how to use `nagg` to package compatible products from single granule input files and shows the use of fill granule files.

Command

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

Command Line Elements

- The `nagg` command with `–n 5` produces aggregations of five granules each.
- `-t SATMS,TATMS` indicates to `nagg` that those two products and their common geolocation data are to be aggregated.
- `nagg` with no `-g` option processes granules for the products specified with `-t` and by default for the corresponding geolocation product granules either from the input files specified for the `-t` products or from files matching the file name in the N_GEO_Ref attribute in those files. For the default behavior, the creation field in the geolocation file name 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

The output of this command is one partial file with four SATMS and four TATMS product granules and four GATMO geolocation product granules and one other partial file with one granule of each product.

Data Granules in Input and Output Files

Aggregations	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

The HDF Group

Notes

The [34]* wildcards in the −t option in the command line matches all SATMS_npp_d20120403… and SATMS_npp_d20120404… file names in the input directory. Those beginning with TATMS match only TATMS_ npp_d20120404… file names. The corresponding GATMO files are included as they are in the previous example (see page 26), and the aggregation boundaries, numbers of granules in output files, and dimensions of raw data datasets are also the same.

The difference in output from the previous example is that with five input SATMS and GATMO granules but only three input TATMS granules, nagg produces two 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 difference is the real granule ID has 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*.

Real Granule Command

The following h5dump command line will retrieve a real granule ID.

```
h5dump -a /Data_Products/ATMS-SDR/ATMS-SDR_Gran_0/N_Granule_ID SATMS-TATMS-
GATMO_npp_d20120403_t2359123_e0001199_b02250_c20120529123052142977_XXXX_XXX.h
5
```

Real Granule Output

The output from the real granule command is below.

```
HDF5 "SATMS-TATMS-
GATMO_npp_d20120403_t2359123_e0001199_b02250_c20120529123052142977_XXXX_XXX.h
5" {
ATTRIBUTE "/Data_Products/ATMS-SDR/ATMS-SDR_Gran_0/N_Granule_ID" {
   …
   DATA {
   (0,0): "NPP000141695514"
   }
}
```

Fill Granule Command

The following `h5dump` command line will retrieve a fill granule ID.

```
h5dump -a /Data_Products/ATMS-TDR/ATMS-TDR_Gran_0/N_Granule_ID SATMS-TATMS-
GATMO_npp_d20120403_t2359123_e0001199_b02250_c20120529123052142977_XXXX_XXX.h
5
```

Fill Granule Output

The output from the fill granule command is below.

```
HDF5 "SATMS-TATMS-
GATMO_npp_d20120403_t2359123_e0001199_b02250_c20120529123052142977_XXXX_XXX.h
5" {
ATTRIBUTE "/Data_Products/ATMS-TDR/ATMS-TDR_Gran_0/N_Granule_ID" {
   …
   DATA {
   (0,0): "N/A"
   }
}
}
```

Input Files (13)

The input files are listed below.

```
test/testfiles/SATMS_npp_d20120403_t2359123_e2359439_b02250_c2012040515544724
2391_noaa_ops.h5
…
test/testfiles/SATMS_npp_d20120404_t0001203_e0001519_b02250_c2012040515544724
2391_noaa_ops.h5
test/testfiles/TATMS_npp_d20120404_t0000163_e0000479_b02250_c2012040515545789
9734_noaa_ops.h5
…
test/testfiles/TATMS_npp_d20120404_t0001203_e0001519_b02250_c2012040515545789
9734_noaa_ops.h5
test/testfiles/GATMO_npp_d20120403_t2359123_e2359439_b02250_c2012040515544164
3487_noaa_ops.h5
…
test/testfiles/GATMO_npp_d20120404_t0001203_e0001519_b02250_c2012040515544164
3487_noaa_ops.h5
```

Command Runtime Output

The status messages produced by the program during processing are listed below.

```
Produced    4 granules in SATMS-TATMS-
GATMO_npp_d20120403_t2359123_e0001199_b02250_c20120529123052142977_XXXX_XXX.h
5
Produced    1 granules in SATMS-TATMS-
GATMO_npp_d20120404_t0001203_e0001519_b02250_c20120529123052282010_XXXX_XXX.h
5
```

3.7. Packaging Compatible Products from Single Granule Input Files using Exact Matching of Geolocation Input File Names

This example shows how to use `nagg` to package compatible products from single granule input files and how to specify the exact matching of geolocation input file names.

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 the example in the "Packaging Compatible Products from Single Granule Input Files" on page 26 except for the `-g strict` command line option. Adding `-g strict` to the command will enforce that geolocation file 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 geolocation 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 does not match that in the attribute, the command fails. With `-g yes` (the default) instead of `-g strict`, the creation time, origin, and domain fields are masked when searching for the geolocation file named in the attribute. The `-g no` option processes the data files without processing the geolocation information.

Command Runtime Output

The status messages produced by the program during processing are listed below.

```
nagg: ***ERROR*** find_geo_granules():  Failed to get geolocation granules.
test/testfiles/GATMO_npp_d20120403_t2359123_e2359439_b02250_c2012040515544724
2391_noaa_ops.h5 does not exist or otherwise cannot be accessed for reading.
nagg: ***ERROR*** nagg_get_granules(): Failed to get geolocation granules
nagg: ***ERROR*** nagg_get_granules(): no granules found for geoproduct.
geoproduct:  GATMO
nagg: ***ERROR*** Failure encountered in Get Granules.
```

The HDF Group

3.8. Unpackaging and De-aggregating Packaged and Aggregated Input Files

This example shows how to use `nagg` to unpackage and de-aggregate 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_unkn_xxx.h
5
```

Command Line Elements

- The `nagg` command without the `-n <number>` command line option 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 de-aggregated 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 geolocation files are unavailable.

Output

The output of this command is 12 files: four for each of the three products.

Data Granules in Input and Output Files

Aggregations	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

The HDF Group

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 one quarter 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 four granules, and the output files' datasets contain a hyperslab for only one granule.

Input File

The input file is listed below.

```
test/testfiles/SVM03-SVM04-
SVM05_npp_d20120424_t0010313_e0016117_b02534_c20120521190943233596_unkn_xxx.h
5
```

Command Runtime Output

The status messages produced by the program during processing are listed below.

```
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_XXXX_XXX.h
5
Produced   1 granules in
SVM05_npp_d20120424_t0010313_e0011555_b02534_c20120521192724535803_XXXX_XXX.h
5
Produced   1 granules in
SVM03_npp_d20120424_t0011567_e0013209_b02534_c20120521192724857446_XXXX_XXX.h
5
Produced   1 granules in
SVM04_npp_d20120424_t0011567_e0013209_b02534_c20120521192724857446_XXXX_XXX.h
5
Produced   1 granules in
SVM05_npp_d20120424_t0011567_e0013209_b02534_c20120521192724857446_XXXX_XXX.h
5
Produced   1 granules in
SVM03_npp_d20120424_t0013221_e0014463_b02534_c20120521192725180119_XXXX_XXX.h
5
Produced   1 granules in
SVM04_npp_d20120424_t0013221_e0014463_b02534_c20120521192725180119_XXXX_XXX.h
5
Produced   1 granules in
SVM05_npp_d20120424_t0013221_e0014463_b02534_c20120521192725180119_XXXX_XXX.h
5
Produced   1 granules in
SVM03_npp_d20120424_t0014475_e0016117_b02534_c20120521192725504013_XXXX_XXX.h
5
```

```
Produced   1 granules in
SVM04_npp_d20120424_t0014475_e0016117_b02534_c20120521192725504013_XXXX_XXX.h
5
Produced   1 granules in SVM05_npp_d20120424_t0014475_e0016117_b0253
4_c20120521192725504013_XXXX_XXX.h5
```

3.9. Extracting a Product from Packaged Files Containing Other Products

This example shows how to use `nagg` to extract 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_unkn_xxx.h
5
```

Command Line Elements

- The `nagg` command with the `-n 4` command line option produces aggregations of four granules each.
- `-t SVM04` indicates to `nagg` that only granules of the SVM04 product 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 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 geolocation files are unavailable.

Output

The output of this command is one file with four granules of the SVM04 product only.

Data Granules in Input and Output Files

Aggregations	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

Notes

This example uses the same input file as the example in the "Unpackaging and De-aggregating Packaged and Aggregated Input Files" section on page 33, 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

The input file is listed below.

```
test/testfiles/SVM03-SVM04-
SVM05_npp_d20120424_t0010313_e0016117_b02534_c20120521190943233596_unkn_xxx.h
5
```

Command Runtime Output

The status message produced by the program during processing is listed below.

```
Produced   4 granules in
SVM04_npp_d20120424_t0010313_e0016117_b02534_c20120521202951397965_XXXX_XXX.h
5
```

The HDF Group

3.10. Aggregating or De-aggregating Geolocation Files

This example shows how to use `nagg` to aggregate or de-aggregate geolocation files.

Command

```
nagg –n32 –g GCRIO ../test/testfiles/GCRIO_npp_d20030125*.h5
```

Command Line Elements

- The `nagg` command with the `–n 32` command line option produces aggregations of 32 granules each.
- The `–g <product>` option is for aggregating or de-aggregating granules of a geolocation product directly without any corresponding sensor date product granules. `–t` is not used with `–g <product>`.

Output

The output of this command is three files with 16 or 32 granules of the GCRIO product only.

Data Granules in Input and Output Files

Aggregations	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

Notes

The `–g <product>` command line option is provided for processing aggregations that contain only geolocation 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; the other 15 duplicates are ignored.

Input Files

The input files are listed below.

```
test/testfiles/GCRIO_npp_d20030125_t0702533_e0711257_b00014_c2011102517050718
4515_unkn_ada.h5
test/testfiles/GCRIO_npp_d20030125_t0705359_e0713337_b00014_c9876543210987654
3210_anyt_ing.h5
test/testfiles/GCRIO_npp_d20030125_t0711279_e0719577_b00014_c2011102517050719
4756_unkn_ada.h5
test/testfiles/GCRIO_npp_d20030125_t0719599_e0728297_b00014_c2011102517050719
7657_unkn_ada.h5
test/testfiles/GCRIO_npp_d20030125_t0728319_e0737011_b00014_c2011102517050720
1941_unkn_ada.h5
```

Command Runtime Output

The status messages produced by the program during processing are listed below.

```
Produced  16 granules in
GCRIO_npp_d20030125_t0702533_e0711257_b00993_c20120523155444015673_XXXX_XXX.h
5
Produced  32 granules in
GCRIO_npp_d20030125_t0711279_e0728297_b00014_c20120523155445226530_XXXX_XXX.h
5
Produced  16 granules in
GCRIO_npp_d20030125_t0728319_e0737011_b00014_c20120523155447399983_XXXX_XXX.h
5
```

3.11. Aggregating and Packaging Files Like an Example File

This example shows how to use `nagg -l or --like <example file>` to aggregate or package granules from input files with a product list and aggregation number like those in the example file. Note that the aggregation number will be the number of granules per product in the file, and if the file is a partial file, the aggregation number will be less than the original aggregation number. Avoid beginning and ending files to insure the full number of granules per file in the original aggregation.

Command

```
nagg -l  ../test/testfiles/GATMO-SATMS-
TATMS_npp_d20120403_t2359123_e0000159_b02250_c20120830155722719166_unkn_xxx.h
5 ../test/testfiles/SATMS_npp_d2012040*.h5
../test/testfiles/TATMS_npp_d2012040
```

Command Line Elements

- -l example file GATMO-SATMS-
  TATMS_npp_d20120403_t2359123_e0000159_b02250_c20120830155722719166_unkn_xxx.

Output

The output of this command is three files with 2 or 1 granules of the SATMS, TATMS, and GATMO products.

Data Granules in Input and Output Files

Aggregations	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	2	3 (2 full, 1 partial)	2

Notes

The example file GATMO-SATMS-TATMS_npp_d20120403_t2359123_e0000159_b02250_c20120830155722719166_unkn_xxx.h5has 2 granules for each of these products: SATMS, TATMS and the GATMO GEO product. The `nagg` command with the `-l` GATMO-SATMS-TATMS_npp_d20120403_t2359123_e0000159_b02250_c20120830155722719166_unkn_xxx.h5 command line option produces aggregations of 2 granules each with SATMS and TATMS granules found in the input files plus corresponding GEO granules.

The -l <example file> option determines the number of granules per output file and the list of sensor data products to be aggregated/packaged by inspection of the example file. All other properties are determined by other command line options or nagg defaults. Use of the -l option with geolocation only files is not supported. The -A, -n, or -t command options will override the corresponding value from the example file.

Input Files

The input files are listed below.

```
../test/testfiles/SATMS_npp_d20120403_t2359123_e2359439_b02250_c2012040515544
7242391_noaa_ops.h5
../test/testfiles/SATMS_npp_d20120403_t2359443_e0000159_b02250_c2012040515544
7242391_noaa_ops.h5
../test/testfiles/SATMS_npp_d20120404_t0000163_e0000479_b02250_c2012040515544
7242391_noaa_ops.h5
../test/testfiles/SATMS_npp_d20120404_t0000483_e0001199_b02250_c2012040515544
7242391_noaa_ops.h5
../test/testfiles/SATMS_npp_d20120404_t0001203_e0001519_b02250_c2012040515544
7242391_noaa_ops.h5
../test/testfiles/TATMS_npp_d20120403_t2359123_e2359439_b02250_c2012040515545
7899734_noaa_ops.h5
../test/testfiles/TATMS_npp_d20120403_t2359443_e0000159_b02250_c2012040515545
7899734_noaa_ops.h5
../test/testfiles/TATMS_npp_d20120404_t0000163_e0000479_b02250_c2012040515545
7899734_noaa_ops.h5
../test/testfiles/TATMS_npp_d20120404_t0000483_e0001199_b02250_c2012040515545
7899734_noaa_ops.h5
../test/testfiles/TATMS_npp_d20120404_t0001203_e0001519_b02250_c2012040515545
7899734_noaa_ops.h5
../test/testfiles/GATMO_npp_d20120403_t2359123_e2359439_b02250_c2012040515544
1643487_noaa_ops.h5
../test/testfiles/GATMO_npp_d20120403_t2359443_e0000159_b02250_c2012040515544
1643487_noaa_ops.h5
../test/testfiles/GATMO_npp_d20120404_t0000163_e0000479_b02250_c2012040515544
1643487_noaa_ops.h5
../test/testfiles/GATMO_npp_d20120404_t0000483_e0001199_b02250_c2012040515544
1643487_noaa_ops.h5
../test/testfiles/GATMO_npp_d20120404_t0001203_e0001519_b02250_c2012040515544
1643487_noaa_ops.h5
```

Command Runtime Output

The status messages produced by the program during processing are listed below.

```
Produced   2 granules in GATMO-SATMS-
TATMS_npp_d20120403_t2359123_e0000159_b02250_c20120831023912403398_XXXX_XXX.h
5
Produced   2 granules in GATMO-SATMS-
TATMS_npp_d20120404_t0000163_e0001199_b02250_c20120831023912464828_XXXX_XXX.h
5
Produced   1 granules in GATMO-SATMS-
TATMS_npp_d20120404_t0001203_e0001519_b02250_c20120831023912550539_XXXX_XXX.h
5
```

3.11.1. Using the –n override

The -l or –like option in the previous example can be used in combination with -n, -A, or -t. When one of these options is also in the command line, it will override the conflicting directive from the –like <example file> option. The following example uses the same example file and input files as in the previous example.

Command

```
nagg -n 3 --like ../test/testfiles/GATMO-SATMS-
TATMS_npp_d20120403_t2359123_e0000159_b02250_c20120830155722719166_unkn_xxx.h
5 ../test/testfiles/SATMS_npp_d2012040*.h5
../test/testfiles/TATMS_npp_d2012040
```

Command Line Elements

- -l example file GATMO-SATMS-
  TATMS_npp_d20120403_t2359123_e0000159_b02250_c20120830155722719166_unkn_xxx.
- -n 3 overrides the aggregation number from the –l option.

Output

The output of this command is two files with 3 or 2 granules of the SATMS, TATMS, and GATMO products.

Data Granules in Input and Output Files

Aggregations	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	3	2 (1 full, 1 partial)	3

Notes

- With the addition of –n 3 the `nagg` command with the `-l` GATMO-SATMS-
  TATMS_npp_d20120403_t2359123_e0000159_b02250_c20120830155722719166_unkn_xxx.h5
  command line option produces aggregations of 3 granules each with SATMS and TATMS
  granules found in the input files plus corresponding GEO granules.

Command Runtime Output

The status messages produced by the program during processing are listed below.

```
Produced   3 granules in GATMO-SATMS-
TATMS_npp_d20120403_t2359123_e0000479_b02250_c20120831160042219489_XXXX_XXX.h
5
```

```
Produced  2 granules in GATMO-SATMS-
TATMS_npp_d20120404_t0000483_e0001519_b02250_c20120831160042440599_XXXX_XXX.h
5
```

3.11.2. Using the –A override

The -l or –like option in the previous example can be used in combination with -n, -A, or -t. When one of these options is also in the command line, it will override the conflicting directive from the –like <example file> option. The following examples use the same example file and input files as in the example above.
Suppose 2 minutes of SATMS, TATMS and associated GATMO data were desired instead of the 2 granules in the example file. –A 120 will provide an aggregation of that size, with the products according to the example file.

Command

```
nagg –A 120 --like ../test/testfiles/GATMO-SATMS-
TATMS_npp_d20120403_t2359123_e0000159_b02250_c20120830155722719166_unkn_xxx.h
5 ../test/testfiles/SATMS_npp_d2012040*.h5
../test/testfiles/TATMS_npp_d2012040
```

Command Line Elements

- -l example file GATMO-SATMS-
  TATMS_npp_d20120403_t2359123_e0000159_b02250_c20120830155722719166_unkn_xxx.
- -A 120 overrides the aggregation number from the –l option.

Output

The output of this command is two files with 4 or 1 granules of the SATMS, TATMS, and GATMO products.

Data Granules in Input and Output Files

Aggregations	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	4	2 (1 full, 1 partial)	4

Notes

- With the addition of –n 120 the `nagg` command with the `-l` GATMO-SATMS-TATMS_npp_d20120403_t2359123_e0000159_b02250_c20120830155722719166_unkn_xxx.h5 command line option produces aggregations of 4 granules each with SATMS and TATMS granules found in the input files plus corresponding GEO granules.

Command Runtime Output

The status messages produced by the program during processing are listed below.

```
Produced   4 granules in GATMO-SATMS-
TATMS_npp_d20120403_t2359123_e0001199_b02250_c20120831161244023811_XXXX_XXX.h
5
Produced   1 granules in GATMO-SATMS-
TATMS_npp_d20120404_t0001203_e0001519_b02250_c20120831161244313132_XXXX_XXX.h
5
```

3.11.3. Using the –t override

The -l or –like option in the previous example can be used in combination with -n, -A, or -t. When one of these options is also in the command line, it will override the conflicting directive from the –like <example file> option. The following examples use the same example file and input files as in the example above.
Overriding the -l option with a –t <product list> option will use the aggregation number from the file, but the –t product list.

Command

```
nagg -t TATMS --like ../test/testfiles/GATMO-SATMS-
TATMS_npp_d20120403_t2359123_e0000159_b02250_c20120830155722719166_unkn_xxx.h
5 ../test/testfiles/SATMS_npp_d2012040*.h5
../test/testfiles/TATMS_npp_d2012040
```

Command Line Elements

- -l example file GATMO-SATMS-TATMS_npp_d20120403_t2359123_e0000159_b02250_c20120830155722719166_unkn_xxx.
- -t TATMS overrides the product list from the –l option.

Output

The output of this command is three files with 2 or 1 granules of the TATMS, and GATMO products.

Data Granules in Input and Output Files

Aggregations	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	10	1	10	1
Output aggregations	10	2	3 (2 full, 1 partial)	2

Notes

- This example uses a subset of the products in the example file. It would be possible to use entirely different products if they were also in the input files.
- The GATMO GEO product is associated with both SATMS and TATMS, and therefore will be packaged with both products or with either product alone.

Command Runtime Output

The status messages produced by the program during processing are listed below.

```
Produced   2 granules in GATMO-
TATMS_npp_d20120403_t2359123_e0000159_b02250_c20120831162351301821_XXXX_XXX.h
5
Produced   2 granules in GATMO-
TATMS_npp_d20120404_t0000163_e0001199_b02250_c20120831162351458586_XXXX_XXX.h
5
Produced   1 granules in GATMO-
TATMS_npp_d20120404_t0001203_e0001519_b02250_c20120831162351559138_XXXX_XXX.h
5
```