The HDF-EOS5 Augmentation Tool User’s Guide

This document describes version 2.4 of the aug_eos5 augmentation tool. The following major sections are included: an overview of the augmentation process, installation and configuration instructions, a user’s guide for aug_eos5, and a user’s guide for the validation tools.

The HDF Group

For more information about The HDF Group, see www.hdfgroup.org.

Contents

1. How the Augmentation Process Works 5
1.1. Background 5
1.2. The Issue 6
1.3. Augmentation 7

2. Installation 9
2.1. Download and Uncompress 9
2.2. Required and Optional Libraries 9
2.3. Configure and Build the Tools 10
2.3.1. Configure Using h5cc 10
2.3.2. Configure for the gcc Compiler 11
2.3.3. Make and Install 11
2.3.4. Building aug_eos5 by Modifying makefile.augment 11
2.3.5. Building the Validation Executables 12

3. Using the aug_eos5 Tool 13
3.1. Command-line Syntax 14
3.2. Command-line Options 15
3.2.1. The Default Command-line Option 16
3.2.2. The File Command-line Option 17
3.2.3. Index 18
3.3. Mapping Files 19
3.3.1. Name Rows 19
3.3.2. Data Rows 20
3.3.3. Notes 20
3.4. Working with Some Grid Data Examples 21
3.4.1. Using the Default Command-line Option 21
3.4.2. Using the Index Command-line Option 21
3.4.3. Using the File Command-line Option 22
3.4.4. Working with Multiple Grids in a Single File 22
3.5. Working with Some Swath and Zonal Average Data Examples 24
3.5.1. Using the Default Command-line Option 24
3.5.2. Using the Index Command-line Option 24
3.5.3. Using the File Command-line Option 24
3.5.4. Working with Multiple Swaths or Zonal Averages 26
3.6. Notes 27
3.6.1. Known Problems 27
3.6.2. Tested Data Products 27
3.6.3. Program Limitations 27
3.6.3.1. Invisible HDF5 Objects Are Not Augmented 27
3.6.3.2. Object Names 28
3.6.3.3. HDF-EOS5 Swath Dimension Maps Are Not Supported 28
3.7. Testing Notes 28
3.7.1. Test with Fake Files 28
3.7.2. Batch Test for NASA Files (Optional) 29

4. Validating Augmented Files 30

4.1. check_c ... 30
4.1.1. Description ... 30
4.1.2. Usage ... 31
4.1.2.1. Swath ... 31
4.1.2.1.1. A Special Example ... 31
4.1.2.1.2. A General Example ... 31
4.1.2.2. Grid ... 32
4.1.2.3. Zonal Average ... 32
4.2. check_f and check_za_f ... 33
4.3. Limitations and Notes ... 33
4.3.1. Data Fields and Attributes ... 33
4.3.2. Data Types ... 34
4.3.3. Pure Dimensions ... 34
5. Other Resources ... 35

1. How the Augmentation Process Works

The HDF-EOS5 Augmentation Tool, aug_eos5, is a utility program that can be used to augment existing HDF-EOS5 files so that applications built with the netCDF-4 library can read the augmented files. This program mainly adds HDF5 dimension scales and associates them with the corresponding HDF5 datasets. Augmented files can still be read by the HDF-EOS5 library.

As of December 2013, the latest version of the netCDF-4 is 4.3. Applications built with this version of the netCDF-4 library can access augmented files.

To understand how this program works, users should have some knowledge of HDF5, HDF-EOS5, and netCDF-4.

1.1. Background

The HDF5 Library defines two primary types of objects: groups and datasets. An HDF5 group is a structure containing zero or more HDF5 objects, and an HDF5 dataset is a multidimensional array of data elements. Both HDF5 datasets and HDF5 groups can have HDF5 attributes. An HDF5 attribute is a small metadata object describing the nature and/or usage of an HDF5 object.

An HDF5 dimension scale is a special HDF5 dataset that is associated with a dimension of another HDF5 dataset to provide temporal or spatial information such as time, longitude, or latitude.

Both HDF-EOS5 and netCDF-4 are built on HDF5. The HDF-EOS5 Library defines a few data types - grid, swath, point, and zonal average – that have been customized for NASA's earth observing system data. The HDF-EOS5 Library also defines groups, fields, attributes, and dimensions. For example, an instance of the grid data type consists of one group, multiple fields, multiple attributes, and multiple dimensions. An HDF-EOS5 group, an HDF-EOS5 field, and an HDF-EOS5 attribute are represented using an HDF5 group, an HDF5 dataset, and an HDF5 attribute, respectively. However, an HDF-EOS5 dimension is not implemented according to the *HDF5 Dimension Scale Specification and Design Notes*.

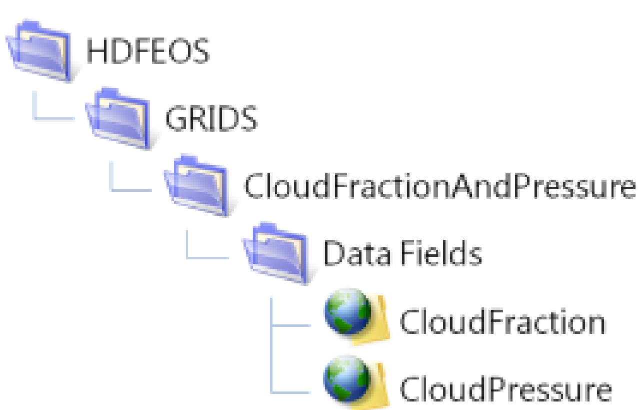

Figure 1. File structure of an HDF-EOS5 file viewed as an HDF5 file

Figure 1 shows the file structure of a typical HDF-EOS5 file viewed as an HDF5 file. In this figure, CloudFraction and CloudPressure are HDF5 datasets corresponding to two HDF-EOS5 fields. CloudFractionAndPressure is an HDF5 group corresponding to an HDF-EOS5 group. Other HDF5 groups including HDFEOS, GRIDS, and Data Fields are groups created by the HDF-EOS5 Library but are invisible to HDF-EOS5 applications.

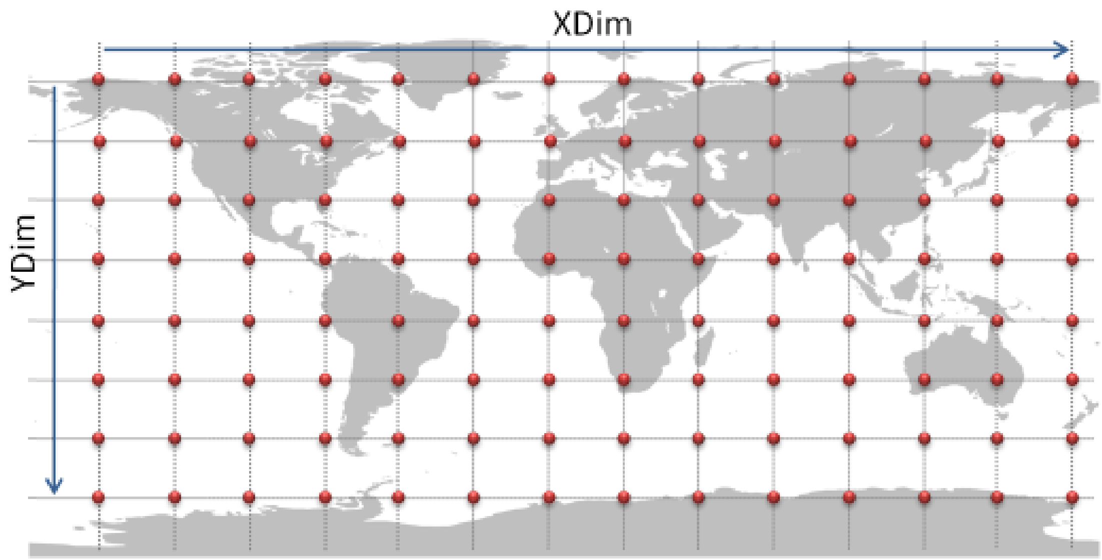

Figure 2. An example of grid data

Figure 2 shows a typical HDF-EOS5 grid data. In this example, the grid is two-dimensional and is defined by XDim and YDim. Since both XDim and YDim are standard dimensions defined by HDF-EOS5, all grid objects have these dimensions and most data fields in grid objects refer to them.

The netCDF-4 library defines groups, variables, attributes, and dimensions. A netCDF-4 group, a netCDF-4 variable, and a netCDF-4 attribute are implemented using an HDF5 group, an HDF5 dataset, and an HDF5 attribute, respectively. A netCDF-4 dimension is implemented as an HDF5 dimension scale by following the directions in *HDF5 Dimension Scale Specification and Design Notes*. The netCDF-4 library requires that every dimension of any single netCDF-4 variable is associated with a netCDF-4 dimension.

1.2. The Issue

Although the two HDF-EOS5 dimensions - XDim, YDim, and their dimension scales - can be retrieved from the HDF-EOS5 Library, they cannot be recognized by the netCDF-4 data model. Therefore, to the netCDF-4 library, the HDF-EOS5 dimension information is actually missing. The lack of dimension information to netCDF-4 is the main reason why netCDF-4 applications cannot read an HDF-EOS5 file by following the netCDF-4 data model.

1.3. Augmentation

To deal with the lack of dimension information, the HDF-EOS5 Augmentation Tool was created to add dimension information to existing HDF-EOS5 files. The result of augmentation is shown in Figure 3 below.

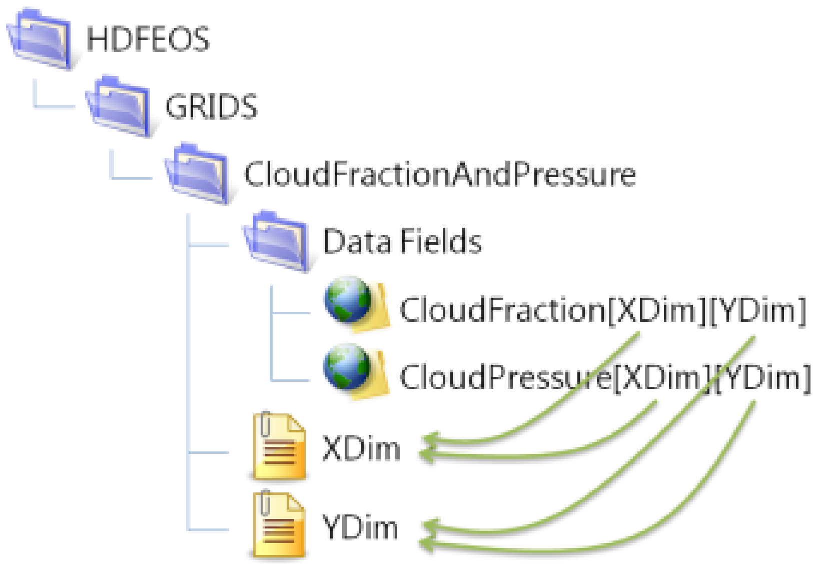

Figure 3. File structure of the augmented HDF-EOS5 file viewed as an HDF5 file

In the augmentation process, two additional HDF5 objects, XDim and YDim are added, and two existing HDF5 datasets, CloudFraction and CloudPressure, are now associated with XDim and YDim. We will briefly explain how existing HDF-EOS5 files are augmented below.

Creating an HDF5 Dimension Scale for Each HDF-EOS5 Dimension

A netCDF-4 dimension is an HDF5 dimension scale. Creating HDF5 dimension scales will make netCDF-4 applications able to recognize netCDF-4 dimensions. For each HDF-EOS5 dimension, the augmentation tool creates an HDF5 dimension scale under the HDF5 group that corresponds to the HDF-EOS5 grid object. For example, XDim and YDim in Figure 3 are created as HDF5 dimension scales. They are created under CloudFractionAndPressure which corresponds to the HDF-EOS5 grid object.

Associating Created HDF5 Dimension Scales with Corresponding HDF5 Datasets

Just creating HDF5 dimension scales by itself does not guarantee that netCDF-4 will associate them with netCDF-4 variables. The association is made through an HDF5 attribute. See the *HDF5 Dimension Scale*

Specification and Design Notes document for more information. The HDF5 dataset that refers to HDF5 dimension scales should have an attribute named DIMENSION_LIST, and the HDF5 dimension scales associated with HDF5 datasets should have an attribute named REFERENCE_LIST.

The DIMENSION_LIST and REFERENCE_LIST attributes are attached during augmentation. This association is described as arrows in Figure 3. With these attributes, netCDF-4 will recognize the association, and the file would become structurally correct.

Fill Two Special HDF5 Dimension Scales, XDim and YDim

Although the previous steps make the augmented file structurally correct, the file is not very useful because it does not contain any longitude or latitude values. These values are mostly missing in HDF-EOS5 grid objects because they can be calculated from several parameters contained in an HDF5 dataset predefined by the HDF-EOS5 Library. In fact, this is the way HDF-EOS5 keeps the file size smaller.

To make the augmented file more useful, the augmentation program calculates longitude and latitude values using HDF-EOS5 APIs and stores them at XDim and YDim in Figure 3. For the grid file shown in Figure 1, the number of elements for XDim is 14 and the number of elements for YDim is 8. Therefore, XDim will have 14 longitude values, and YDim will have eight latitude values.

2. Installation

The installation process has the following general steps. These steps are described in the sections below.

- Download and Uncompress
- Required and Optional Libraries
- Configure and Build the Tools

2.1. Download and Uncompress

The first step in the installation process is to download the HDF-EOS5 augmentation tool files at www.hdfeos.net/software/aug_eos5/. The tool files are kept in a compressed file named aug_eos5-x.y.tar.gz where x and y are version numbers.

After uncompressing the file, three directories - `src`, `test`, and `validprog` - will be created under the installation directory, and an `examples` sub-directory will be created under `validprog`. These are listed in the table below.

Directory	Comments
src	The `src` directory includes the source code files, Makefile, and a makefile template. The makefile template can be used to build the tool in case the configuration script does not work on a user's computer system.
test	The `test` directory includes a few shell scripts that can be used to test the tool with real HDF-EOS5 files.
validprog	The `validprog` directory includes the source code files and sample HDF-EOS5 files for the simple validation programs.
validprog/examples	The `examples` sub-directory under `validprog` holds some sample HDF-EOS5 files and mapping files. The sample HDF-EOS5 files are `single_swath_2.he5` and `single_zonal_average.he5`. The sample mapping files are `single_swath_2_mapping_file.txt` for `single_swath_2.he5` and `single_zonal_average_mapping_file.txt` for `single_zonal_average.he5`. See the "Mapping Files" section on page 19.

2.2. Required and Optional Libraries

The required and optional libraries are listed below. Confirm that the required libraries are installed and that the optional libraries are installed if they will be needed.

The following libraries are required to build the aug_eos5 augmentation tool:

- HDF-EOS5 C Library (1.14 or higher)
  - Download at newsroom.gsfc.nasa.gov/sdptoolkit/TKDownload.html
- HDF5 C Library (1.8, prefer 1.8.11 or higher)
  - Download at www.hdfgroup.org/downloads/index.html

The following libraries are required to build the aug_eos5 tool if these libraries were used to build the HDF5 Library:

- szlib
  - Download at www.hdfgroup.org/release4/obtain.html
- zlib

The netCDF-4 library is not needed to build aug_eos5. The netCDF-4 library is required to build the simple validation programs check_c, check_f, and check_za_f. See the "Validating Augmented Files" section on page 30 for more information. The netCDF-4 library is required to build any application that might be used to confirm that an augmented file can be read by a netCDF-4 application.

- netCDF-4
  - Download at www.unidata.ucar.edu/downloads/netcdf/index.jsp

2.3. Configure and Build the Tools

The sub-sections below describe ways to configure and build aug_eos5 and the validation tools.

2.3.1. Configure Using h5cc

The recommended way to configure a system to build aug_eos5 is to use the h5cc helper script.

Assume the HDF5 library path is `/path/to/hdf5`.

Set CC to be `/path/to/hdf5/bin/h5cc`.

Use `setenv` for csh and `export` for bash.

Execute the configuration script under the top directory with the appropriate path:

```
. /configure --with-hdfeos5=/path/to/hdfeos5
```

For more information on h5cc, see www.hdfgroup.org/HDF5/doc/RM/Tools.html#Tools-H5CC in the *HDF5 C Reference Manual*.

For more information on configuring and building the HDF-EOS5 C Library, see the "HDF-EOS5" section on the "How to Build HDF-EOS" page at hdfeos.org/software/hdfeos.php#ref_sec:hdf-eos5.

2.3.2. Configure for the gcc Compiler

If the gcc compiler will be used, execute the configuration script below with the appropriate path names:

```
. /configure \
--with-zlib=/path/to/zlib \
--with-szlib=/path/to/szlib \
--with-hdf5=/path/to/hdf5 \
--with-hdfeos5=/path/to/hdfeos5
```

If the HDF5 library was not built with szlib, there is no reason to include `--with-szlib`.

On most Linux platforms, the zlib library is installed in their system path. If this is the case, there is no reason to include `--with-zlib` either.

The above configuration script will build a dynamically linked executable. Adding `LDFLAGS=-static` to your configuration script will build a statically linked executable instead.

2.3.3. Make and Install

After successful configuration, type the following command to build aug_eos5:

```
make install
```

The executable file `aug_eos5` will be located under `/path/to/augmentation/aug_eos5/bin` and `/path/to/augmentation/src` or under `/path/to/installation/bin` if you use the prefix option during the configuration.

2.3.4. Building aug_eos5 by Modifying makefile.augment

If the configuration scripts shown above do not work on your computer system, you could also generate the aug_eos5 executable file by executing a makefile template named `makefile.augment` under the `/src` directory as follows:

```
make -f makefile.augment
```

The executable `aug_eos5` will be generated under the `/src` directory.

For `makefile.augment` to work properly, modify the library path based on your own system configuration, such as

```
HDF5_DIR=/path/to/hdf5
HDFEOS5_DIR=/path/to/hdfeos5
……
```

For Mac users, ignore warnings such as the following:

```
run.c: In function 'gridaugalloc_info':
run.c:56: warning: format '%4d' expects type 'int', but argument 3 has type
'size_t'
run.c:56: warning: format '%4d' expects type 'int', but argument 4 has type
'size_t'
```

The executable file will work properly in spite of these warnings.

2.3.5. Building the Validation Executables

The validation executables are used to check the accessibility of an augmented file by the netCDF-4 APIs. The executables can be built by typing the following command under the `validprog` directory:

```
make -f makefile.check
```

This generates three executable files - check_c, check_f, and check_za_f - which are capable of validating whether the netCDF-4 C or Fortran APIs can access augmented files as expected. For `makefile.check` to work properly, you should modify the library path based on your own system configuration. The following is an example:

```
NCCONFIG=/path/to/netCDF4/bin/nc-config
```

3. Using the aug_eos5 Tool

This chapter describes how to use the HDF-EOS5 augmentation tool, aug_eos5. This command-line tool can be used to change HDF-EOS5 files so that netCDF-4 APIs are able to read the files. HDF-EOS5 swath, grid, and zonal average objects can be augmented.

A grid object and a grid dataset are the same. A grid file is a file with one or more grid datasets. The same can be said for swath and zonal average objects. A swath object can also be called a swath dataset. A swath file is a file with one or more swath datasets. A zonal average object is a zonal average dataset. A zonal average file is a file with one or more zonal average datasets.

The process of augmenting grid objects is slightly different than the process of augmenting swath and zonal average objects. The tool creates dimension scales for each grid, swath, and zonal average dataset and associates the dimension scales with the datasets via attributes. With grid data files, the tool also generates longitude and latitude data via the HDF-EOS5 APIs. The longitude and latitude data are already in files with swath and zonal average objects. See the "Augmentation" section on page 7 for more information.

Since aug_eos5 changes an existing HDF-EOS5 file during the augmentation process, the specified file should be writable by the program user, and a backup copy of the file should be made.

Verifying whether an augmented file can be read by the netCDF-4 library can be done through the use of `ncdump` or the validation tools we provide. `ncdump` is a netCDF dumper tool distributed with the netCDF-4 library. See the "Validating Augmented Files" section on page 30 for more information on how to use our validation tools.

3.1. Command-line Syntax

The command-line syntax for aug_eos5 is

```
aug_eos5 commandlineoption [mappingfilename] eos5filename
```

The elements of the command-line are described in the table below.

Element	Comments
`aug_eos5`	`aug_eos5` is the name of the tool.
`commandlineoption`	`commandlineoption` specifies the action the tool will take. Possible values are: • **blank** for the Default option • **-f** for the File option • **-i** for the Index option See the "Command-line Options" section below for more information.
`mappingfilename`	`mappingfilename` is the name of a plain text mapping file that is only required by the File command-line option. See the "Mapping Files" section on page 19 for more information.
`eos5filename`	`eos5filename` is the name of the HDF-EOS5 file that will be changed. The HDF-EOS5 file may contain grid or swath/zonal average data.

Note that datasets are not specified in the command line. If a file has more than one dataset, the tool will augment every dataset.

3.2. Command-line Options

The tool will act according to the command-line option specified on the command line. Three options are possible: Default, File, and Index.

Command-line Option	Purpose	For more information, see this page:
Default	Use this option to augment grid data files and swath/zonal average data files if users do not know the relationships between dimensions and dimension scales in the HDF-EOS5 files.	16
File	Use this option if users know the relationships between dimensions and dimension scales in the HDF-EOS5 files. The tool will use the information provided by users to correctly generate the dimension scales so that the augmented file can be accessed via netCDF-4 by following the netCDF-4 data model. We strongly recommend users to choose this option if possible.	17
Index	Use this option to augment data files if you want dimension scales to be filled in with natural numbers. The only exception is the XDim and YDim dimensions for grid objects. XDim and YDim dimension scales are filled in the same way as the Default command-line option.	18

3.2.1. The Default Command-line Option

This option benefits users who do not have detailed knowledge of their files. This option is easier to use, but the downside is that a user has no control over the augmentation process. So, we only recommend this option for the users who do not have detailed knowledge of their files.

For grid data, the implied longitude and latitude values are put into the XDim and YDim dimensions. If the data array has more than two dimensions, the other dimensions will be treated as pure dimensions by netCDF-4 APIs.

Swath and zonal average data objects have longitude and latitude data already and only need the dimension scales and attributes set up.

The dimension scales are created under each dataset separately. If a file has more than one dataset, the tool will augment the file with dimension scales for each dataset. Every dataset will be augmented. Datasets that have already been augmented will not be processed again, and an error message will be generated by aug_eos5.

To use the Default option, do not include a command-line option on the command line. You only need to enter `aug_eos5` and the name of the file you wanted augmented on the command line. The command-line syntax using the Default option is shown below.

Default Command-line Syntax

```
aug_eos5 <eos5filename>
```

3.2.2. The File Command-line Option

The File command-line option relies on a mapping file supplied by the user. See the "Mapping Files" section on page 19 for more information.

Use this option if users can provide dimension scales for some dimensions in the HDF-EOS5 files. The tool will use the dimension scale information provided by the users to correctly generate those dimension scales. The tool will follow the Default command-line option to handle the rest of the dimensions. We recommend that users choose this option if possible.

`XDim` and `YDim` for grid files are added in the same way as the Default command-line option if the mapping file does not provide any information for `XDim` and `YDim`. If the mapping file does provide mappings for `XDim` and `YDim`, then they are processed just like the other dimensions according to the mapping file.

To use the File option, include –f and the name of the mapping file on the command line. The following is the command-line syntax where the File option is specified:

File Command-line Syntax

```
aug_eos5 -f mappingfilename eos5filename
```

3.2.3. Index

If you cannot use the File command-line option, use the Index command-line option as an alternative for the Default command-line option especially if you need to augment grid data files with more than two dimensions. The X and Y dimensions are filled with longitude and latitude data in the same way as the Default command-line option. Other dimensions are filled with natural numbers with this command-line option; the Default command-line option fills other dimensions with zeroes. This command-line option does not use a mapping file.

The Index option provides another alternative solution for users who cannot use the File option due to a lack of detailed knowledge of their files. Like the Default option, this option provides an easier to use way of augmenting datasets than the File option. The ease of use comes with the same lack of control that the Default option has.

To use the Index option, include the "-i" argument. The command-line syntax using the Index option is shown below.

Index Command-line Syntax

```
aug_eos5 –i <eos5filename>
```

3.3. Mapping Files

The mapping file plays an important role in the augmentation process for the File command-line option. It consists of information aug_eos5 uses to create the HDF5 dimension scales.

Grid, swath, and zonal average data are stored in arrays of two or more dimensions. In order for these dimensions to be visible to netCDF-4 applications, dimension scales must be associated with the dimensions. A mapping file is used by aug_eos5 to connect data dimensions with dimension scales.

Mapping files are plain text files. Each row conveys information. Each element in a row is separated by a blank space from other elements in the row.

There are two types of rows: name and data. These are described in the sections below.

See the "Working with Multiple Grids in a Single File" section on page 22 and the "Working with Multiple Swaths or Zonal Averages in a Single File" section on page 26 for more information.

3.3.1. Name Rows

A name row identifies the type of dataset and the name of the dataset that will be augmented. A name row is needed for each grid, swath, or zonal average dataset in the file. If there is only one dataset in a file, then no name row is needed.

The general format of a name row is

```
type name: ObjectName
```

where

`type` is the type of dataset (use either grid or swath),

`name` is the word name, and

`ObjectName` is the name of the dataset.

There are three types of datasets: grid, swath, and zonal average. The following are some examples of name rows:

```
grid name: NadirDayGrid
grid name: NadirNightGrid
swath name: Swath1
swath name: Swath2
zonal name: ZonalAverage1
zonal name: ZonalAverage2
```

3.3.2. Data Rows

A data row identifies a dimension that needs a dimension scale and specifies where the dimension scale information is located. There can be in a mapping file multiple data rows: one for every dimension that needs a dimension scale. With grid datasets, the `XDim` and `YDim` dimensions are assumed: rows can be added to grid mapping files for the `XDim` and `YDim` dimensions, but are not needed.

The general format of a data row is the following:

```
LocationType dimension_name location
```

where

`LocationType` defines the type of location,

`dimension_name` is the name of the dimension in the HDF-EOS5 file, and

`location` is the place where the dimension scale data resides.

In the `LocationType` column, valid values will be 0, 1, or 2. Use 0 if the location is a data field in the same file as the dataset that is being augmented. Use 1 if the location is another file. See the "External File Notes" section below for more information. Use 2 if the location column will be blank. A 2 means the dimension scale will be filled with natural numbers starting with 1.

If a dimension exists in the augmented dataset and is not included in the mapping file, then the dimension will be treated as a pure dimension by netCDF-4.

External File Notes

This section describes the external file used to hold dimension scale information. This file will be a text file with values. Each value will be separated from other values by commas, newlines, spaces, or tabs. The following is an example:

```
…
11.111
12.22
…
```

The number of values in the file must match the corresponding dimension size. Otherwise, the augmentation tool will generate an error message: "The required data length is different from the length in the data file. Please check the data file to eliminate the inconsistency."

3.3.3. Notes

The following are some other notes regarding mapping files.

- Users do not need to provide any dimension information in a mapping file if that dimension is not used by any data field.
- Users can add some comments in a mapping file to increase readability. Any line starting with "#" is considered a comment. aug_eos5 will ignore such a line.

3.4. Working with Some Grid Data Examples

HDF-EOS5 grids are datasets that have longitude and latitude information implied in the dataset. aug_eos5 through the HDF-EOS5 APIs can generate the longitude and latitude information along with the attributes needed so that netCDF-4 applications can see the data. After augmentation, longitude information is stored in the XDim dimension if the projection is geographic, and latitude information is stored in the YDim dimension if the projection is geographic.

The sample HDF-EOS5 data files and mapping files referred to in this section can be found in the validprog/examples directory. Examples can also be found on our ftp server at ftp://ftp.hdfgroup.uiuc.edu/pub/outgoing/NASAHDFTOOLfiles/aug_eos5/sample_files_for_doc/.

If you run the command lines shown below, augment copies of the original sample file.

3.4.1. Using the Default Command-line Option

To use the Default command-line option to augment single_grid.he5 with grid data, enter at the command prompt the following command:

```
aug_eos5 single_grid.he5
```

Note in the augmented output file that `XDim` and `YDim` are filled with longitude and latitude data values. The other dimensions are treated as pure dimensions by netCDF-4.

3.4.2. Using the Index Command-line Option

To use the Index command-line option to augment single_grid.he5 with grid data, enter at the command prompt the following command:

```
aug_eos5 -i single_grid.he5
```

Note in the augmented output file that `XDim` and `YDim` are filled with longitude and latitude data values in the same way as the Default command-line option. The other dimension scale is filled with natural numbers.

3.4.3. Using the File Command-line Option

Before the File command-line option can be used, a mapping file must be made. For this example, insert the following text in a file named `single_grid_mapping_file.txt`:

```
0 ZDim Pressure
```

Each element in the row is separated from another element by a blank space. The `0` in the first column means the value of column 3 will be the name of the data field in the file where the dimension scale information can be found. The value of column 2, `ZDim`, is the name of the dimension to which the scale will be applied. The value of column 3, `Pressure`, is the name of the data field in the file that will be augmented where the dimension scale can be found. See the "Mapping Files" section on page 19 for more information.

To use the File command-line option to augment single_grid.he5, a file with grid data, enter at the command prompt the following command:

```
aug_eos5 -f single_grid_mapping_file.txt single_grid.he5
```

Note in the augmented output file that `XDim` and `YDim` are filled with longitude and latitude data values in the same way as the Default command-line option. The other dimension scale, `ZDim,` is filled with the values obtained from `Pressure` instead. `Pressure` is a data field in the HDF-EOS5 file. Obtaining the dimension scale from a field inside an HDF-EOS5 file is the most common case a user may encounter.

If for some reason a dimension scale needs to be obtained from an external text file, add the following line for that dimension to the mapping file:

```
1 ZDim data_file
```

`data_file` is a plain text file where the values of a dimension scale field are stored.

Generally, longitude and latitude should be calculated via HDF-EOS5 APIs. Under such circumstances, the mapping file does not need to provide any information about `XDim` and `YDim`.

3.4.4. Working with Multiple Grids in a Single File

aug_eos5 can be used to augment files that have multiple grids (multiple grid datasets).

When the **Default** or **Index** command-line options are used to augment a file with a single grid, the dimension scales of `XDim` and `YDim` are created under the grid dataset. See the figure in the "Augmentation" section on page 7 for more information. When these options are used to augment a file with multiple grid datasets, the dimension scales of `XDim` and `YDim` are created under each grid dataset.

When the **File** command-line option is used with a file with multiple grids, the mapping file will have entries for each grid. The following example shows how to construct such a file.

Suppose a file called `two_grids.he5` needs to be augmented. This file contains two grids, `NadirDayGrid` and `NadirNightGrid`. Each grid has a dimension called `nLevels`. For this example, we are assuming that users want to fill the corresponding dimension scale with the values from a data field.

Before running aug_eos5, we should write the mapping file. Each grid will first be identified in a row in the mapping file. The next row will specify where the dimension scale data will be found (in this case a data field in the file), the name of the dimension, and the name of the data field that holds the dimension scale data.

In this example, there will be four rows: two rows identify the grids, and two rows specify where aug_eos5 can find the dimension scale. The first row in this example identifies the grid as NadirDayGrid, and the third row identifies NadirNightGrid. The second and fourth rows have a `0` in the first column since the dimension data will come from a data field in the file. The dimension name, `nLevels`, is specified in the second column. The data field name, `Pressure`, is specified in the third column. The following shows the layout of the `two_grids_mapping_file.txt` mapping file:

```
grid name: NadirDayGrid
0 nLevels Pressure
grid name: NadirNightGrid
0 nLevels Pressure
```

To augment `two_grids.he5`, enter the following command at a command prompt:

```
aug_eos5 -f two_grids_mapping_file.txt two_grids.he5
```

3.5. Working with Some Swath and Zonal Average Data Examples

Swath datasets and zonal average datasets are handled in the same manner. The example command lines in this section augment swath data files. Zonal average data files can be substituted for swath files.

The sample HDF-EOS5 data files and mapping files referred to in this section can be found in the validprog/examples directory. The example files have two dimensions: `NDim` and `ZDim`. If you run the command lines shown below, augment copies of the original sample file.

3.5.1. Using the Default Command-line Option

At the command prompt, enter the following command to augment `single_swath.he5` with the Default command-line option:

```
aug_eos5 single_swath.he5
```

Note that for the HDF5 dataset, the `NDim` and `ZDim` dimensions, if seen with HDFView, are filled with zeros. However, the netCDF-4 library will treat these two dimensions as dimensions without having dimension scales (in other words, pure netCDF-4 dimensions).

3.5.2. Using the Index Command-line Option

At the command prompt, enter the following command to augment `single_swath.he5` with the Index command-line option:

```
aug_eos5 -i single_swath.he5
```

Note that `NDim` and `ZDim`, the newly added dimension scales, are filled with natural numbers.

3.5.3. Using the File Command-line Option

Two examples of using the File command-line option with swath (and zonal average) data files are described in this section.

Example 1

Before the File command-line option can be used, a mapping file must be made. For this example, insert the following text in a file named `single_swath_mapping_file.txt`:

```
0 NDim Latitude
0 ZDim Pressure
```

There are two rows in this mapping file. A row is needed for each dimension that must be augmented. Each element in the rows is separated from another element by a blank space. The `0` in the first column means the value of column 3 will be the name of the data field in the file where the dimension scale information can be found. The values in column 2, `NDim` and `ZDim`, are the names of the dimensions to which the scales will be applied. The values of column 3, `Latitude` and `Pressure`, are the names of the data fields where the dimension scales can be found. See the "Mapping Files" section on page 19 for more information.

To use the File command-line option to augment `single_swath.he5`, a file with swath data, enter at the command prompt the following command:

```
aug_eos5 –f single_swath_mapping_file.txt single_swath.he5
```

Note that two dimension scale fields, `NDim` and `ZDim`, have been added. `NDim` and `ZDim` are filled with the values obtained from `Latitude` and `Pressure` respectively. `Latitude` and `Pressure` are data fields in the HDF-EOS5 file. Obtaining the dimension scale from a field inside an HDF-EOS5 file is the most common case a user may encounter.

Example 2

In this example, a file with three dimensions is augmented. The name of the file is `single_swath_2.he5`. The dimensions are `nTimes`, `nLevels` and `nCloudTypes`.

The mapping file for this example is `single_swath_2_mapping_file.txt`. The contents of the file are shown below:

```
0 nTimes Time
0 nLevels Pressure
```

Enter the following command to augment `single_swath_2.he5`:

```
aug_eos5 –f single_swath_2_mapping_file.txt single_swath_2.he5
```

Note that three dimension scales, `nTimes`, `nLevels`, and `nCloudTypes`, have been added. `nTimes` and `nLevels` have been filled with the values obtained from the data fields called `Time` and `Pressure` in the HDF-EOS5 file respectively. In this example, there is no information in the mapping file on how to handle `nCloudTypes`. The augmentation tool currently makes `nCloudTypes` a pure dimension for netCDF-4 applications. If the dimension scale of `nCloudTypes` should be filled with natural numbers, a user should add the following line in the mapping file:

```
2 nCloudTypes
```

The third column in this line is blank.

If for some reason a dimension scale needs to be obtained from an external text file, add the following line for that dimension to the mapping file:

```
1 dimension_name data_file
```

`dimension_name` is the name of the dimension, and `data_file` is a plain text file where the values of a dimension scale field are stored.

3.5.4. Working with Multiple Swaths or Zonal Averages

aug_eos5 can be used to augment files that have multiple swath or zonal average datasets.

When the **Default** or **Index** command-line options are used to augment a file with multiple swath or zonal average datasets, the dimension scales are created under the swath or zonal average dataset. See the figure in the "Augmentation" section on page 7 for more information.

When the **File** command-line option is used with a file with multiple swath or zonal average datasets, the mapping file will have entries for each dataset. The following example shows how to construct such a file.

Suppose a file called `two_swaths.he5` needs to be augmented. This file contains two swaths, `Swath1` and `Swath2`. Each swath has dimensions called `NDim` and `ZDim`. For this example, we are assuming that users want to fill the corresponding dimension scale with the values from data fields.

Before running aug_eos5, we should write the mapping file. Each swath will first be identified in a row in the mapping file. The next row will specify where the dimension scale data will be found (in this case a data field in the file), the name of the dimension, and the name of the data field that holds the dimension scale data.

In this example, there will be six rows: two rows identify the swaths, and four rows specify where aug_eos5 can find the dimension scales. The first row in this example identifies the swath as Swath1, and the fourth row identifies Swath2. The other rows have a `0` in the first column since the dimension data will come from a data field in the file. The dimension names, `NDim` and `ZDim`, are specified in the second column. The data field names, `Latitude` and `Pressure`, are specified in the third column. The following shows the layout of the `two_swaths_mapping_file.txt` mapping file:

```
swath name: Swath1
0 NDim Latitude
0 ZDim Pressure
swath name: Swath2
0 NDim Latitude
0 ZDim Pressure
```

To augment `two_swaths.he5`, enter the following command at a command prompt:

```
aug_eos5 -f two_swaths_mapping_file.txt two_swaths.he5
```

3.6. Notes

3.6.1. Known Problems

The only known problem is the following:

There is a bug in the HDF-EOS5 library (version HDF-EOS5.1.14) when compiling on a MacOS 10.9 (Mavericks) system, and this bug will cause the augmentation tool failed. We reported this to the HDF-EOS5 developer and were told that this bug will be fixed in the new HDF-EOS5 library release.

3.6.2. Tested Data Products

We have tested aug_eos5 on Aura HIRDLS, MLS, TES, OMI, and NASA MEaSUREs GSSTF data products on 32-bit and 64-bit Linux systems and on 64-bit Mac systems.

A set of guidelines is available at hdfeos.org/software/aug_eos5/Aug-NASA-EOS5-guide.pdf to help users augment the NASA HDF-EOS5 products listed above.

3.6.3. Program Limitations

The known limitations and known problems of aug_eos5 are listed below.

3.6.3.1. Invisible HDF5 Objects Are Not Augmented

This program assumes that all HDF5 datasets are accessible via the HDF-EOS5 APIs. This limitation is caused by the fact that this program uses the HDF-EOS5 APIs to collect all information. If an HDF-EOS5 file contains an HDF5 dataset that cannot be accessed by the HDF-EOS5 APIs, this program cannot add HDF5 dimension scales for that dataset, which may cause the netCDF-4 library to fail to read the file even after augmentation.

3.6.3.2. Object Names

Object names in HDF-EOS5 can contain white spaces and some special characters allowed by HDF5. Some of these characters are forbidden to be used in netCDF-4 variables, but aug_eos5 does not try to make the name safe. This tool attempts to keep the original file as much as possible.

We also assume that HDF-EOS5 object, field, and attribute names in an HDF-EOS5 file contain only alphanumeric characters and underscores since other characters are not allowed by the netCDF-4 library.

3.6.3.3. HDF-EOS5 Swath Dimension Maps Are Not Supported

HDF-EOS5 swaths may have dimension maps. aug_eos5 cannot handle these maps, and the program will return an error message that states a dimension map exists in the HDF-EOS5 file.

If this limitation matters, the HDF-EOS5 to netCDF-4 Converter tool can be used: it can handle dimension maps correctly. However, the file generated by the HDF-EOS5 to netCDF-4 Converter can no longer be accessed by HDF-EOS5 APIs. More information is available at http://hdfeos.org/software/convert_hdfeos5.php.

3.7. Testing Notes

Some testing scripts are described below. These scripts when run will confirm that aug_eos5 has been built correctly.

3.7.1. Test with Fake Files

The purpose of this test is to run the tool with all options on HDF-EOS5 files created for this tool.

Go to the `test` directory test and run test.sh. Use the following command line:

```
./test.sh
```

The output should include the augmenting information for the testing files.

The output should also include OK, passed, or 'correctly handles', and you should not see "FAILED" in the output.

See the "Download and Uncompress" section on page 9 for more information on the `test` directory.

3.7.2. Batch Test for NASA Files (Optional)

This is an optional sanitized test for users who want to check to see if this tool works properly with typical NASA HDF-EOS5 files. Generally users do not need to use this test. The test described in the section above should be sufficient.

This test essentially compares the augmented sample HDF-EOS5 files generated by users with the augmented sample HDF-EOS5 files generated by The HDF Group.

For proper testing, the directory containing the h5diff binary should be included in the system path.

Go to the `test` directory, and then go through the following steps:

- Change to the `nfile_test` directory
- Run the scripts
    - On Linux, run download_default.sh and download_file_index.sh
    - On Mac, run download_default_mac.sh and download_file_index_mac.sh

    Use the following command lines:
    ```
    ./download_default.sh
    ./download_file_index.sh
    ```

    These scripts will download the original sample HDF-EOS5 files and the expected augmented files. The original files are under the directory `default_option` and `file_index_option`. The expected augmented files are under the directory `default_option/augmented_files` or `file_index_option/augmented_files`.

    The original sample HDF-EOS5 files are GSSTF, MOPITT, HIRDLS, MLS, OMI and TES files.

- Check to see if the tool correctly augments the file. Use the following command lines:

    ```
    ./test_nfile_default.sh
    ./test_nfile_file_index.sh
    ```

    The output should include the augmenting information for the testing files. The output should also include OK, passed, and you should not see "FAILED" in the output.

4. Validating Augmented Files

We also provide executable files for users to validate whether an augmented file can be accessed by the netCDF-4 library. The names of these executable files are check_c, check_f, and check_za_f.

check_c uses the netCDF-4 C APIs. check_f and check_za_f use the netCDF-4 Fortran APIs. check_c provides more flexibility to users. By providing command line parameters, a user can use check_c to validate any 32-bit floating point data field. The data field is a field under the Data Fields group in an HDF-EOS5 file. In addition, if the data field contains attributes "Units" and "Title", the values of these attributes will also be printed out. However, the data field to be validated is hardcoded in the source code of check_f and check_za_f.

A makefile template is provided to help users generate these executable files. See the "Configure and Build the Tools" section on page 10 for more information.

4.1. check_c

4.1.1. Description

The syntax of check_c is as follows:

```
check_c [OPTION]… [AUGMENTED FILE]
```

OPTION:

`-t, --file type`
    0(swath)/1(grid)/2(zonal average)
`-g, --HDF-EOS5 object (group) name`
    HDF-EOS5 object (grid/swath/zonal average) to be validated
`-v, --variable name`
    HDF-EOS5 data field to be validated

AUGMENTED FILE: the name of an augmented file

See "Using the aug_eos5 Tool" chapter on page 13 for more information.

The variable to be validated should be a data field of an HDF-EOS5 object (grid/swath/zonal average) in an augmented file. Quotation mark should be used to delimit an HDF-EOS5 object name or a variable name if the name contains spaces.

4.1.2. Usage

4.1.2.1. Swath

4.1.2.1.1. A Special Example

To address a user's request, we provide a special example to test an augmented swath file with check_c. Users do not need to specify any command-line parameters to check a data field in this file. The name of the augmented HDF-EOS5 swath file for this example is `single_swath_2.he5`, and the file can be found under the `validprog` directory. The data field to be validated is `CFC11`, and it is under a swath called `HIRDLS`. At the command prompt, type the following command:

```
./check_c
```

The output is as follows:

```
No. of dimension(s): 2

Dim[0]: nTimes 5554

38.73  39.48  40.50  41.25  42.28  43.02  44.05  44.79  45.82  46.55  47.59
48.31

Dim[1]: nLevels 145

1000.00  908.52  825.40  749.89  681.29  618.97  562.34  510.90  464.16
421.70  383.12  348.07

Values:

-999.0000000000  -999.0000000000  -999.0000000000  -999.0000000000  -
999.0000000000  -999.0000000000  -999.0000000000  -999.0000000000  -
999.0000000000  -999.0000000000

Attributes:

Units=vmr

Title=CFC11
```

4.1.2.1.2. A General Example

Here we provide a general example on how to use check_c for an augmented swath file. Assume that a user wants to validate the data field `BrO` under the swath `BrO` for an HDF-EOS5 swath file `MLS-Aura_L2GP-BrO_v02-23-c01_2010d255.he5`. This file can be obtained from ftp://ftp.hdfgroup.uiuc.edu/pub/outgoing/NASAHDFTOOLfiles/aug_eos5/sample_files_for_batch_test/augmented_files/file-index/MLS-Aura_L2GP-BrO_v02-23-c01_2010d255.he5.

At the command prompt, type the following the command:

```
./check_c -t 0 -g BrO -v BrO MLS-Aura_L2GP-BrO_v02-23-c01_2010d255.he5
```

The output is as follows:

```
No. of dimension(s): 2
  Dim[0]: nTimes 3495
    558403200.00  558403264.00  ……
  Dim[1]: nLevels 37
    1000.00  681.29  ……
Values:
    0.0000000000  0.0000000000  ……
Attributes:
  Units=vmr
  Title=BrO
```

4.1.2.2. Grid

Assume that a user wants to validate the data field `TotalColumnDensity` under the object `NadirGrid` for an augmented grid file `TES-Aura_L3-O3_r0000011812_C01_F01_07.he5`.This file can be obtained from

ftp://ftp.hdfgroup.uiuc.edu/pub/outgoing/NASAHDFTOOLfiles/aug_eos5/sample_files_for_batch_test/augmented_files/file-index/TES-Aura_L3-O3_r0000011812_C01_F01_07.he5

At the command prompt, type the following command:

```
./check_c -t 1 -g NadirGrid -v TotalColumnDensity TES-Aura_L3-
O3_r0000011812_C01_F01_07.he5
```

The output is similar to the swath example described in "A Special Example" on page 31. Note that we do not provide a special example to validate a grid field like the swath field described in "A Special Example."

4.1.2.3. Zonal Average

Assume that a user wants to validate the data field `Day` under the zonal average `HIRDLS` for an augmented zonal average file `HIR3ZA-test2.he5`. The file can be obtained from

ftp://ftp.hdfgroup.uiuc.edu/pub/outgoing/NASAHDFTOOLfiles/aug_eos5/sample_files_for_batch_test/augmented_files/file-index/HIR3ZA-test2.he5

At the command prompt, type the following command:

```
./check_c -t 2 -g HIRDLS -v Day HIR3ZA-test2.he5
```

The output is similar to the swath example described in "A Special Example" on page 31. Note we do not provide a special example to validate a zonal average field like the swath field in "A Special Example."

4.2. check_f and check_za_f

Unlike check_c, check_f and check_za_f do not need command-line parameters. Instead the hard-coded parameters are provided in the source code. In our example, check_f validates the data field `CFC11` under the swath `HIRDLS` for an HDF-EOS5 swath file `single_swath_2.he5`. This file is the same as the one used by check_c. check_za_f validates the data field `Day` under the object `HIRDLS` for an HDF-EOS5 zonal average file `single_zonal_average.he5`.

At the command prompt, typing the following command to validate the swath file `single_swath_2.he5`:

```
./check_f
```

The output is similar to the C example described in 2.2.1.1. One can also modify the parameters in the source code of check_f to validate a grid file.

At the command prompt, type the following command to validate the zonal average file `single_zonal_average.he5`:

```
./check_za_f
```

The output is similar to the C example described in "A Special Example" on page 31.

4.3. Limitations and Notes

4.3.1. Data Fields and Attributes

In this release, check_c only validates a data field and its two attributes: Units and Title. If these two attributes exist, their values will be printed out together with the values of the data field; otherwise the "NetCDF: Attribute not found" error will be reported. Be aware that this is not an error if these two attributes do not exist. A user can also use check_c to validate other attributes by modifying the code itself. In the similar way, check_f and check_za_f can also be used to validate other data fields and attributes if a user modifies the corresponding source code.

4.3.2. Data Types

Without modifying the source code, the data type for a data field to be validated is limited to 32-bit floating point only; the data type for an attribute to be validated is limited to string only.

4.3.3. Pure Dimensions

After aug_eos5 is run with the Default command-line option, some dimensions in the augmented file may not have dimension scales. For such a dimension (aka a pure dimension), only the dimension name and dimension length will be printed in check_c output.

5. Other Resources

Some other resources are listed below.

The "Investigation Report for Allowing netCDF-4 to Access HDF-EOS5 Files" is a technical report that explains that there are three different ways of augmenting the HDF-EOS5 files and why the HDF-EOS5 augmentation tool uses the solution described in the "How it works" section. The report can be found at http://www.hdfeos.net/software/aug_eos5/doc/Investigation Report for Allowing netCDF-4 to Access HDF-EOS5 Files.pdf

See the *HDF5 Dimension Scale Specification and Design Notes* document at http://www.hdfgroup.org/HDF5/doc/HL/H5DS_Spec.pdf for more information.