Request For Comments (Draft)
HDF5 Augmentation Tool for NPOESS/JPSS Product Files
Requirements, Design, and Mapping Specifications

Elena Pourmal, Larry Knox
The HDF Group

This document discusses requirements and design for the HDF5 augmentation tool. The tool will modify NPOESS/JPSS product files to be accessible by netCDF-4 applications. File modification will be done according to the proposed specification.

It is assumed that the reader is familiar with the major concepts of HDF5, netCDF-4, and NPOESS product files and the ways they will be delivered to the customers.

This version of the document focuses on augmenting a non-aggregated and non-packaged NPOESS product file.[1]

[1] A non-aggregated and non-packaged NPOESS product file is a product file that contains one granule; geolocation information is located in a separate file. Both product file and geolocation product file are delivered to a user.

1 Introduction

NPOESS/JPSS data is critical for long-range weather and climate forecasting. The processed data will be distributed in the HDF5 file format that addresses the volume and complexity of the data and the enormously high speed at which it must be processed.

Many popular data analysis and visualization applications in climate and weather forecasting research communities are netCDF-based. While the new version of the netCDF library, netCDF-4, uses HDF5 as its storage layer, it cannot read an arbitrary HDF5 file but only the files that satisfy the netCDF-4 profile; for more on the topic see [1]. The netCDF-4 library cannot access NPOESS product files unless they are modified or until the netCDF-4 library implements all HDF5 features used to create the files. Examples of HDF5 elements that must be changed include region and object reference datatypes and multidimensional attributes.

To complicate the problem, data analysis and visualization tools need more than just a netCDF-4 accessible file. Tools rely on metadata such as dimension variables, geolocation information and miscellaneous attributes to interpret and display the data. Since required metadata information is not present in the NPOESS/JPSS product files and is available in the NPOESS XML product files and the NPOESS geolocation files, more modifications are needed before data can be analyzed and visualized.

The HDF Group's developers were tasked to create a tool that would modify an NPOESS product file to make the file readable by the netCDF-4 library and tools. As a part of this task the developers wrote this Request for Comment document (RFC) to solicit comments on requirements, overall design, and on the proposed NPOESS XML-to-HDF5 mapping.

The RFC is organized as follows: Section 2 discusses the tool's purpose, assumptions, requirements, use cases, and design overview; Section 3 focuses on the tool's functional requirements; Section 4 specifies a mapping from the NPOESS product data schema file to the HDF5 objects.

The NPOESS product schema file, the NPOESS XML product file, and the NPOESS product and geolocation files (HDF5 files) used in the examples in this document can be found at http://www.hdfgroup.uiuc.edu/ftp/pub/outgoing/NPOESS/augmentation-tool/design-files/.

2 Purpose, Assumptions, Requirements, and Use Cases

2.1 Purpose

The purpose of the HDF5 augmentation tool is to modify an NPOESS product file with information found in the NPOESS XML product file and in the NPOESS geolocation file to make the file accessible by the netCDF-4.1 library and later.

2.2 Assumptions

1. The following three files are available to a user to be used with the tool:
    a. NPOESS product file
    b. NPOESS XML product file with the metatdata for the NPOESS product file

   c. NPOESS geolocation product file that has geolocation information for the NPOESS product file

2. The tool doesn't check the validity of the three files in section 2.2. 1.a-c. It is the user's responsibility to provide the correct data files.
3. A modified NPOESS product file may not be restored to its original state.
4. The tool does not check CF compliance of the NPOESS metadata.
5. It is acceptable that some HDF5 objects (groups, datasets, and their attributes) in the original file are not accessible in the augmented file.
6. It is acceptable that an augmented file may require additional modifications to enable data to be visualizee with a tool such as Unidata's Integrated Data Viewer (IDV http://www.unidata.ucar.edu/software/idv/). In other words, an augmented file might not work with an application such as IDV.

2.3 Requirements

The tool should satisfy the following requirements:

1. The tool is a command-line tool
2. The tool shall augment the NPOESS product file using information found in the NPOESS XML product file and the NPOESS geolocation product file.
3. The tool shall verify information found in the NPOESS XML product file against information present in the NPOESS product file.
4. The augmented file shall satisfy the following:
   a. All objects in the augmented file are readable by the netCDF-4.1 library and later.
   a. The augmented file has meaningful dimensional and other metadata information to interpret the data by the netCDF-4.1 library and later.
   b. The augmented file has geolocation information available in the same location as product raw data.

2.4 Use Cases

The following scenarios describe how NPOESS data consumers may use the tool.

2.4.1 Read a Variable Using a C Program

A user would like to open an NPOESS data product file and read data of the "Radiance" variable using the following C program.

```
#include <netcdf.h>
#define FILE_NAME "SVI05_aqu_grav_dev.h5"
int
main()
{
    int ncid, varid1, grp1id, grp2id;

    size_t radiance_index[] = {0,0};

    unsigned short data;

    int retval;

    nc_open(FILE_NAME, NC_NOWRITE, &ncid);

    nc_inq_ncid(ncid, "All_Data", &grp1id);

    nc_inq_ncid(grp1id, "VIIRS-I5-SDR_All", &grp2id);

    nc_inq_varid(grp2id, "Radiance", &varid1);

    nc_get_var1_ushort(grp2id, varid1, radiance_index, &data);

    printf("The first data value is %u.\n", data);

    nc_close(ncid);

    return 0;
}
```

The program fails as shown.

```
./nc_read_my_data

NetCDF: Bad type ID
```

The user runs the HDF5 augmentation tool on the NPOESS data product file and reruns the program that now succeeds.

```
./nc_read_my_data

The first data value is 65533.
```

2.4.2 Display NPOESS Data Product File Using ncdump

A user downloads an NPOESS data product file and tries to display the file with the netCDF-4 ncdump utility. The application fails to access the data as shown.

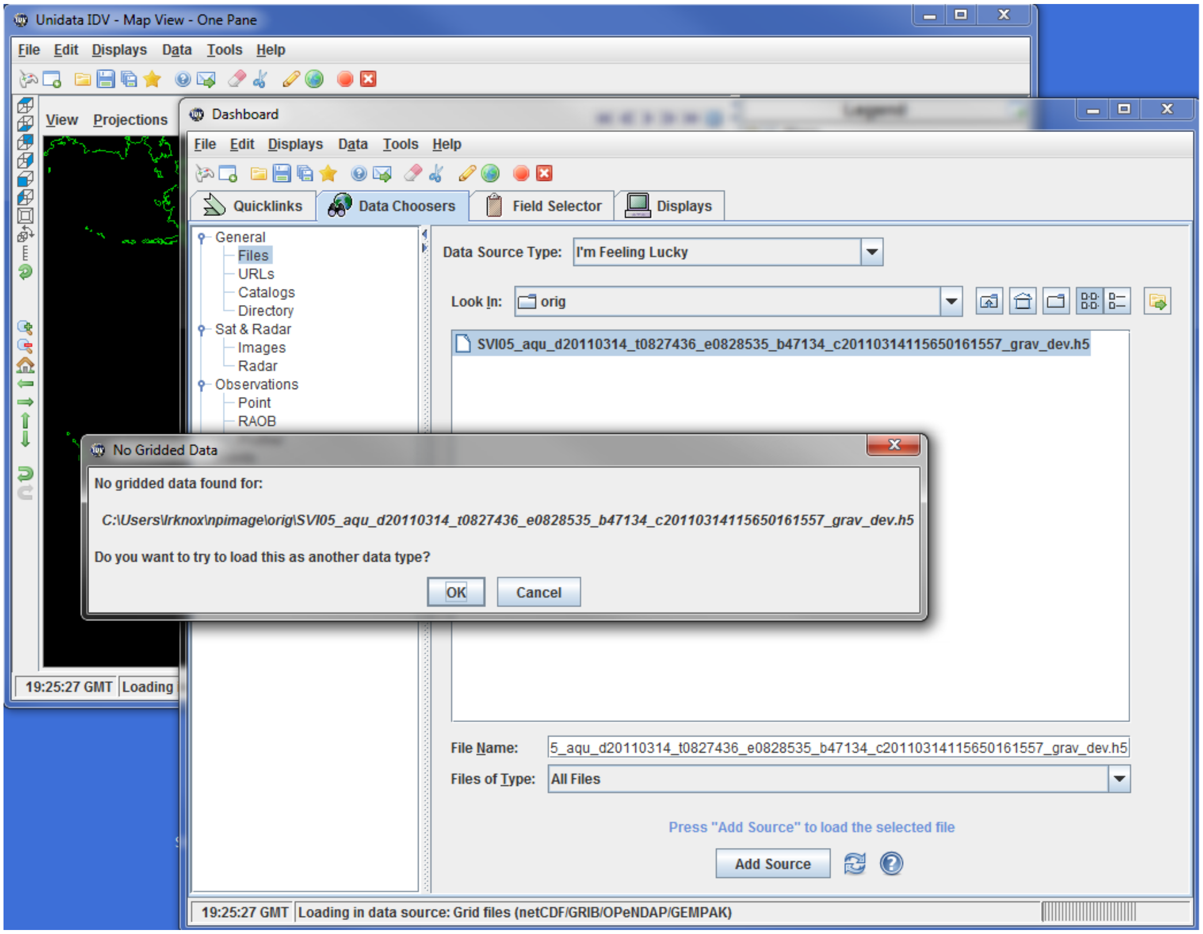
```
./ncdump     SVM07_ter_d20101206_t2009584_e2011083_b0000-1_c20101206231443705497_grav_dev.h5:
NetCDF: Bad type ID
```

The user modifies the file with the HDF5 augmentation tool and reruns the application. The second time the ncdump utility succeeds displaying:

```
netcdf SVM07_ter_d20101206_t2009584_e2011083_b0000-1_c20101206231443705497_grav_dev {
dimensions:
        AlongTrack = 768 ;
        CrossTrack = 3200 ;
        Detector = 16 ;
        …
variables:
        int AlongTrack(AlongTrack) ;
        int CrossTrack(CrossTrack) ;
        int Detector(Detector) ;
        ….
         ……
        float Radiance(AlongTrack, CrossTrack) ;
                Radiance:Description = "Calibrated Top of Atmosphere (TOA) Radiance for each
VIIRS pixel" ;
                Radiance:DatumOffset = 0 ;
                Radiance:Scaled = 1 ;
                Radiance:ScaleFactorName = "RadianceFactors" ;
                …
       // global attributes:
              string:N_GEO_Ref=
"GMODO_ter_d20101206_t2009584_e2011083_b00000_c20101206225316640547_grav_dev.h5" ;
              string :Distributor = "grav" ;
              string :Mission_Name = "NPP_Proxy" ;
              ….
       }
```

2.4.3 Visualize Data with IDV

The user would like to visualize the "Radiance" variable with the IDV tool. He runs the HDF5 augmentation tool to make the file accessible by the netCDF-4 library and to add geolocation information to the file. IDV fails as shown on Figure 1 due to the absence of the required HDF5 attributes on the "Latitude", "Longitude", and "Radiance" variables.



No Gridded Data

No gridded data found for:

C:\Users\lrknox\npimage\orig\SVI05_aqu_d20110314_t0827436_e0828535_b47134_c20110314115650161557_grav_dev.h5

Do you want to try to load this as another data type?

Data Source Type: I'm Feeling Lucky

Look In: orig

SVI05_aqu_d20110314_t0827436_e0828535_b47134_c20110314115650161557_grav_dev.h5

File Name: 5_aqu_d20110314_t0827436_e0828535_b47134_c20110314115650161557_grav_dev.h5

Files of Type: All Files

Press "Add Source" to load the selected file

19:25:27 GMT Loading in data source: Grid files (netCDF/GRIB/OPeNDAP/GEMPAK)

Figure 1: IDV fails to show data in the augmented file due to the absence of the required attributes.

The user runs the h5edit tool to add the attributes required by IDV to properly display the data of the “Radiance” variable. Added attributes and their properties are shown in Table 1. The “scale_factor” and “add_offset” attributes to the “Radiance” variable are necessary to make the value range correct (2 to 8.8 instead of 3 to 31168), and the “valid_min” and “valid_max” attributes affect the range, the colors, and the display of some fill values.[2]

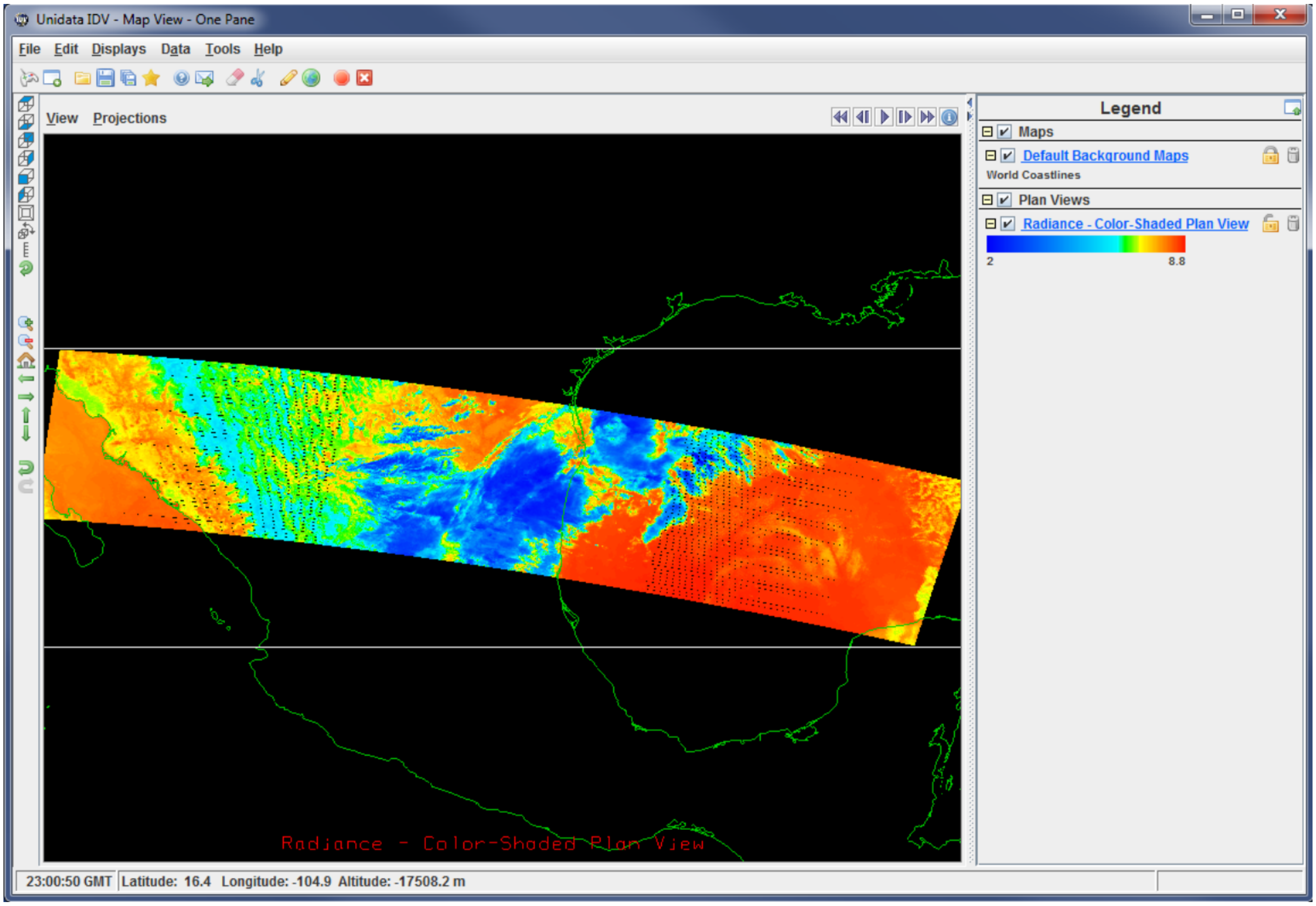
.

Variable	Required Attribute	Type	Value
Latitude	units	string	degrees_north
Longitude	units	string	degrees_east
Radiance	coordinates	string	Latitude Longitude
	add_offset	float	-0.08
	scale_factor	float	2.8339462E-4
	valid_min	ushort	0
	valid_max	ushort	65527

Table 1: Attributes on the “Latitude”, “Longitude”, and “Radiance” variable required by IDV.

The data can be visualized now as shown on Figure 2.

[2] To display the data as shown on Figure 2 the authors had not only to add attributes as discussed, but also to create links in the Root group to all datasets found under the /All_Data group and then unlink the /All_Data group from the group hierarchy. This extra step was done because the current version of IDV doesn’t work with the extended netCDF model that allows groups.


Unidata IDV - Map View - One Pane
File Edit Displays Data Tools Help
View Projections
Legend
Maps
Default Background Maps
World Coastlines
Plan Views
Radiance - Color-Shaded Plan View
2
8.8
Radiance - Color-Shaded Plan View

23:00:50 GMT Latitude: 16.4 Longitude: -104.9 Altitude: -17508.2 m

Figure 2: “Radiance” variable shown by IDV in the augmented file.

2.5 Design Overview

The HDF5 augmentation tool will take as input two files: an NPOESS product file and an NPOESS XML product file. The tool will apply the following augmentation process to the NPOESS product file:

- Modify the NPOESS product file to make it accessible by the netCDF-4 library.
- Parse the XML file to extract dimensional and other information and add it to the NPOESS product file according to the NPOESS XML-to-HDF5 specification.
- Add geolocation information from the NPOESS geolocation product file to the NPOESS product file.

One should emphasize that the augmentation file **DOES** change the original product file as shown on Figure 3.

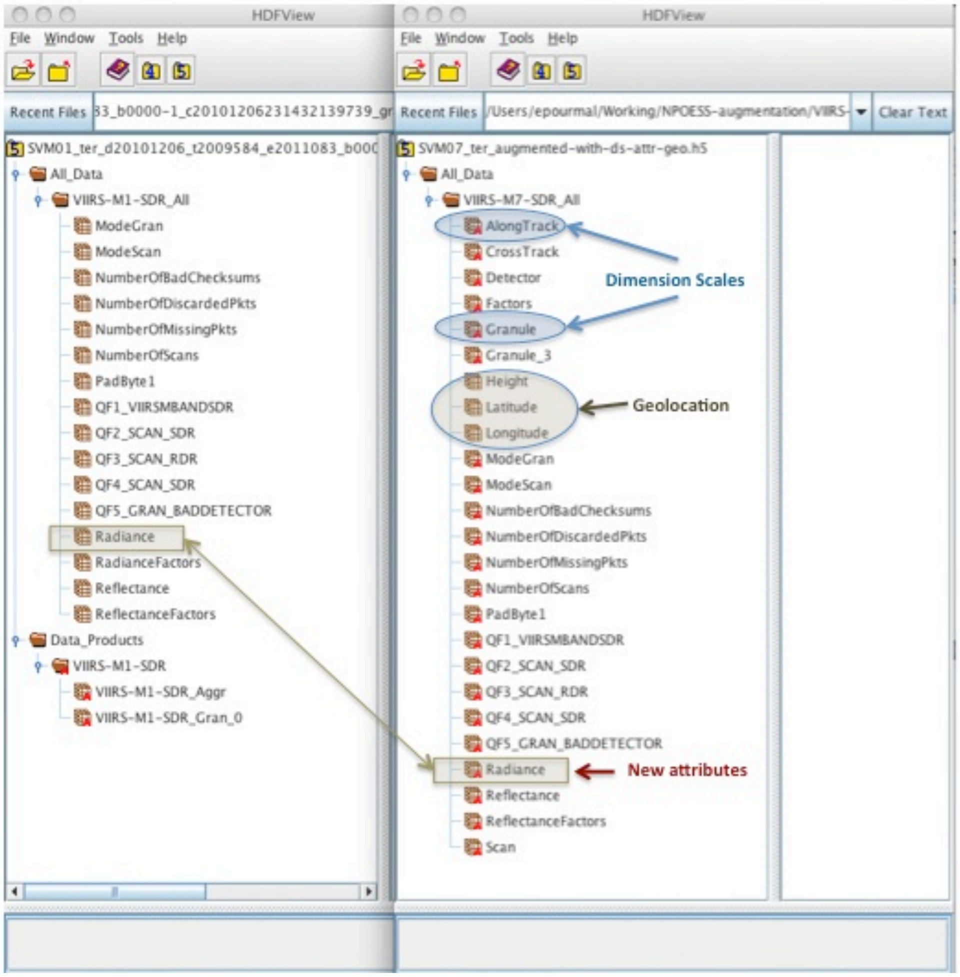


Figure 3: File tree structure in the original NPOESS product file (on the left) and the augmented file (on the right) in HDFView.

Notice that there is no /Data_Products group in the augmented file. The tool removes the link to make file accessible by the netCDF-4 library. This is a critical step of the augmentation process that can be avoided only by enhancing the netCDF-4 library. The library has to be modified to skip "unknown" objects such as datasets with object and region references and objects with the multidimensional attributes. In our example, such datasets are located under the /Data_Products group and the VIIRS-M7-SDR group has a multidimensional attribute.

Information about removed links is stored in the file, and the augmentation tool can restore the "presence" of the group in the file if requested. Section 3 discusses the requirements for the HDF5 augmentation tool in a greater detail.

One can also see that the augmented file now has dimension scales datasets such as "AlongTrack" and "Granule" under the /All_Data/VIIRS-M7-SDR_All group that original datasets like "Radiance" have new attributes added from the XML product file; and that internal HDF5 attributes are created when dimensions scales are attached to a dataset.

3 Augmentation Tool Design

This section is organized as follows. Section 3.1 discusses the tool's general architecture. Section 3.2 introduces three independent augmentation steps and their affect on the file content. Section 3.3 gives an overview of the input to the tool. Augmentation steps are discussed in great detail in section 3.4. Error Handling and exit codes are covered in section 3.5. Finally, non-functional requirements are presented in section 3.6.

3.1 Overview of the Tool Architecture

The HDF5 augmentation tool (h5augjpss) is a command line tool. Its functionality is implemented with public APIs (TBD in another document). For example, the tool will use public functions[3] to perform different augmentation steps (see 3.2) or to restore the original group structure in the augmented file (see sections 3.3.2.4).

3.2 Augmentation Steps

The tool is designed to perform three augmentation steps:

1. Augment a file to make it readable by the netCDF-4 library and ncdump utility.
2. Augment the file with dimensional information and other metadata found in the NPOESS XML product file to allow tools like ncdump to display meaningful information (see example in section 2.3.2 and compare it with example in section 3.3.1).
3. Augment the file with the geolocation information needed for visualization tools based on the netCDF library.

Step 1 does not require any input files and creates a minimum number of new objects in the file. This step does require some changes to the file structure. A file can be easily restored to its original state. See section 3.3.1 for detailed discussion.

Step 2 adds new datasets and attributes. The file is augmented with datasets and attributes. While the file size does not increase much, restoration of the file to its original state becomes very complex.

Step 3 adds several datasets with the geolocation information. This noticeably increases the size of the original file.

3.3 Input

As was discussed in 2.2 the following files are required for the h5augjpss tool:

1. NPOESS product file (an HDF5 file to be augmented)

   Example: SVM07_aqu_d20101130_t1343575_e1345074_b45616_c20101201065150846446_grav_dev.h5

[3] The library with public functions will be built and distributed with the h5augjpss tool. API specifications will be provided after the requirements and functionality for the h5augjpss tool are finalized.

2. NPOESS XML product file (required for level 2 augmentation)

   Example: D34862-03_NPOESS-CDFCB-X-Vol-III_D_VIIRS-M7-SDR-PP.xml

3. NPOESS geolocation product file (required for level 3 augmentation)

   Example: GMODO_ter_d20101206_t2009584_e2011083_b00000_c20101206225316640547_grav_dev.h5

3.3.1 Input Files

The tool shall require at least one file: an NPOESS product file. More than one NPOESS product file may be specified to augment data files of the same product.

Levels 2 and 3 shall require at least two files: an NPOESS product file and an NPOESS XML product file. More than one NPOESS product file may be specified to augment files of the same product.

3.3.2 User Options

3.3.2.1 Control File Option

The tool shall provide an option to read the input from a command file.

3.3.2.2 Help and Version Option

The tool shall provide a command-line option for displaying a help page and a tool version number that corresponds to the version of the NPOESS XML-to-HDF5 specification (see section 4).

3.3.2.3 Augmentation Level Option

The tool shall provide a command-line option to specify an augmentation corresponding to each of the augmentation levels (see section 3.2).

This option will allow users to choose an appropriate level of augmentation to avoid unnecessary changes to the NPOESS product files.

3.3.2.4 Restore Option

The tool shall provide a command-line option to restore the link to the /Data_Products group.

3.4 Augmentation Steps

The h5augjpss tool shall perform three major steps to augment an NPOESS product file. Each of the steps listed below is independent of each other and can be specified by the command line option discussed in 3.3.2.3. Figure 4 shows the steps performed in a sequence (the tool's default behavior) to augment a file to be readable by netCDF- based visualization tools.

1. Hide HDF5 data structures that are not recognized by the current netCDF-4 library and store information about to allow restoring the data structures later. This step is discussed in section 3.4.1.
2. Copy information from the associated NPOESS product XML file to the NPOESS product file according to the specification discussed in section 3.4.2.
3. Add geolocation information (and CF compliant metadata) to the NPOESS product file as described in sections 3.4.3 and 3.4.4.

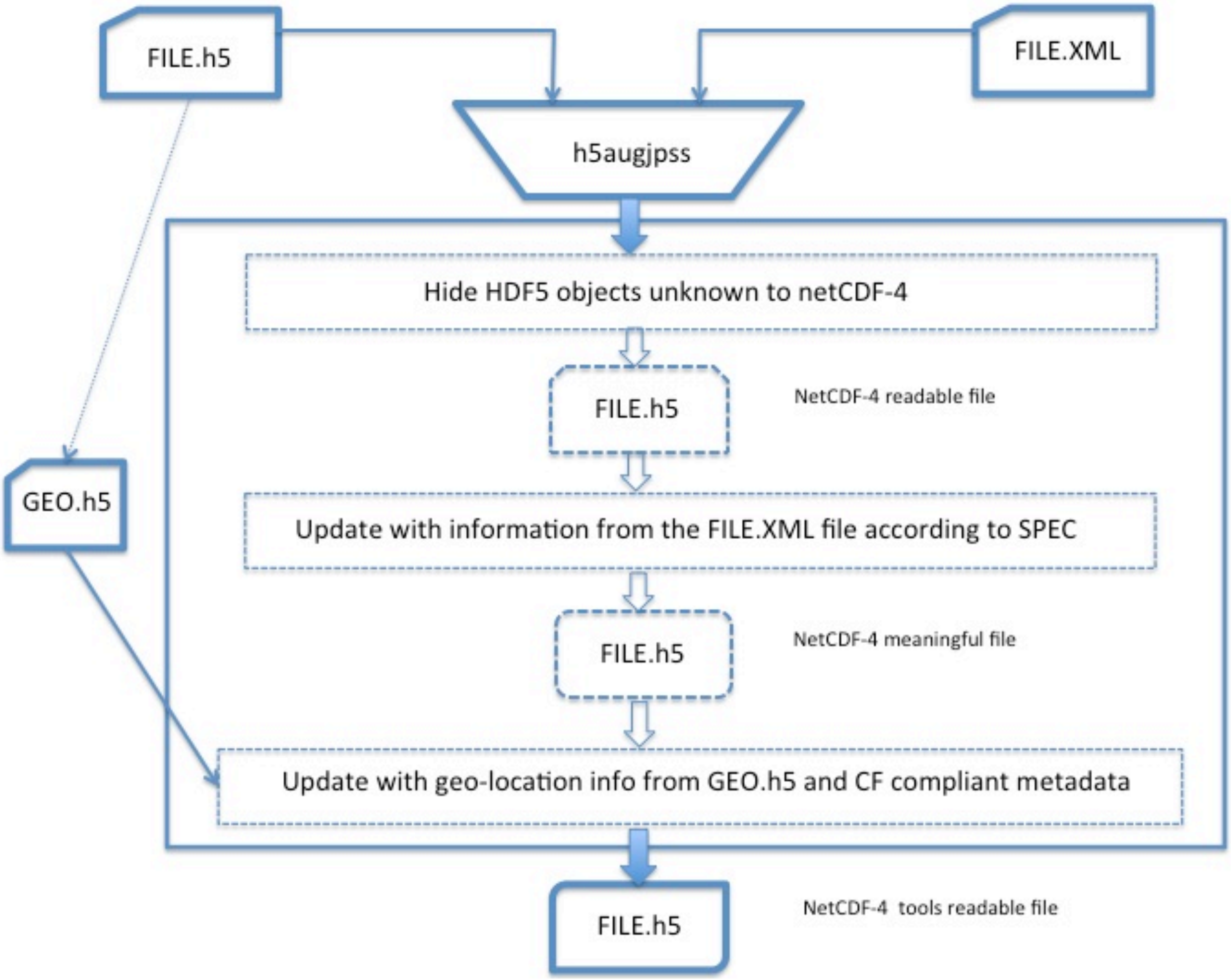

FILE.h5
FILE.XML
h5augjpss
Hide HDF5 objects unknown to netCDF-4
FILE.h5
NetCDF-4 readable file
GEO.h5
Update with information from the FILE.XML file according to SPEC
FILE.h5
NetCDF-4 meaningful file
Update with geo-location info from GEO.h5 and CF compliant metadata
FILE.h5
NetCDF-4 tools readable file

Figure 4: Default augmentation process: augmentation steps are performed in the order shown. The augmented file is not only readable by the netCDF-4 library but also has meaningful dimensional and geolocation information that is used by tools like ncdump and IDV.

The next sub-sections describe each step in greater detail.

3.4.1 Hiding Objects Unknown to the netCDF-4 Library

The purpose of this step is to make an NPOESS file readable by the netCDF-4 library.

The netCDF-4 library cannot handle an arbitrary HDF5 file. For example, the current version of the netCDF-4 library[4] fails to open an HDF5 file that has groups or datasets with two-dimensional attributes or datasets with HDF5 object or region references.

Unfortunately, every NPOESS file has a “Data_Products” group which contains objects that netCDF-4 library does not handle. As a result, the netCDF-4 library fails to open the file. Currently, the only remedy to this problem is to modify every NPOESS product file to hide the “Data_Products” group.

[4] netCDF-4.1.1 and release candidate netCDF-4.1.2-rc1

This is accomplished by deleting a path to the “Data_Products” group in the file hierarchy, which makes this group invisible in the file to netCDF-4. Information about the hidden group is stored as attributes on the Root group of the NPOESS product file according to the specification shown in the Appendix and can be used to restore access to the group (see 3.3.2.4).

After the group is hidden, for example, the netCDF tool ncdump can read the file and display its content as shown:

```
netcdf SVM07_ter_augmented-step1 {

// global attributes:
                string:N_GEO_Ref                                                   =
"GMODO_ter_d20101206_t2009584_e2011083_b00000_c20101206225316640547_grav_dev.h5" ;
                string :Distributor = "grav" ;
                string :Mission_Name = "NPP_Proxy" ;
                ….

group: All_Data {

  group: VIIRS-M7-SDR_All {
    dimensions:
        ….
        phony_dim_3 = 768 ;
        phony_dim_4 = 3200 ;
        ….
    variables:
        ubyte ModeGran(phony_dim_0) ;
        ubyte ModeScan(phony_dim_1) ;
        ….
        ubyte QF4_SCAN_SDR(phony_dim_3) ;
        ubyte QF5_GRAN_BADDETECTOR(phony_dim_5) ;
        float Radiance(phony_dim_3, phony_dim_4) ;
        ushort Reflectance(phony_dim_3, phony_dim_4) ;
        float ReflectanceFactors(phony_dim_6) ;
    } // group VIIRS-M7-SDR_All
  } // group All_Data
}
```

Since there is no dimension information in the original NPOESS product file, ncdump shows “phony” dimensions for the variables (highlighted in yellow above). Dimension information is present in the NPOESS XML Product file and has to be added to the NPOESS product file for the netCDF-4 library to present the data in a meaningful way. We would like to emphasize here that the absence of the dimension information doesn’t prevent a netCDF-4 application from reading the variables stored in the file. Therefore, in Figure 4 the augmented file is shown as “NetCDF readable”.

Dimension information is added during the next augmentation step described in section 3.4.2

3.4.2 Updating with Information from the NPOESS XML File

The purpose of this step is to read information present in the NPOESS XML product file and store it in the NPOESS product file.

An NPOESS XML product file contains information that is applicable to any corresponding HDF5 NPOESS product file. This information includes the name of the product, the name of the collection the product belongs to, a product identifier, dimension information for the variables stored in the

NPOESS product file, and other miscellaneous data. Examples of miscellaneous data include measurement units and fill values. See the excerpt below:

```
    <ProductName>VIIRS Moderate Resolution Band 7 SDR</ProductName>
        <CollectionShortName>VIIRS-M7-SDR</CollectionShortName>
        <DataProductID>SVM7</DataProductID>
        <ProductData>
                <DataName>VIIRS M-Band SDR Data Product Profile</DataName>
                <Field>
                        <Name>Radiance</Name>
                        <Dimension>
                                <Name>AlongTrack</Name>
                                <GranuleBoundary>1</GranuleBoundary>
                                <Dynamic>0</Dynamic>
                                <MinIndex>768</MinIndex>
                                <MaxIndex>768</MaxIndex>
                        </Dimension>
                        ….
                        <Datum>
                        …
                                <DatumOffset>0</DatumOffset>
                                <Scaled>1</Scaled>
                                <ScaleFactorName>RadianceFactors</ScaleFactorName>
                                <MeasurementUnits>W/(m^2 µm sr)</MeasurementUnits>
                                <DataType>unsigned 16-bit integer</DataType>
                                <FillValue>
                                        <Name>NA_UINT16_FILL</Name>
                                        <Value>65535</Value>
                                </FillValue>
                                <FillValue>
                                        <Name>MISS_UINT16_FILL</Name>
                                        <Value>65534</Value>
                                </FillValue>
                                …
```

The augmentation tool reads information from the XML file and writes it according to the specification described in section 4. Since some information found in the XML file is already present in the original file, for example, as a name of a group, it may be desirable to not map all XML elements. This practice helps avoid extra file modification operations and avoids redundant data.

We are seeking input from the user community on which information from the XML file should be saved in the HDF5 NPOESS product file. Redundant information is noted in the corresponding sub-sections of section 4.

If this step is done after Step 1, the netCDF-4 library finds the necessary dimensional information that allows tools like ncdump to display the file content in a more meaningful way as shown below. Examples of the dimensional information and other metadata brought from the XML file are highlighted in yellow; an example of redundant information is highlighted in grey. Notice that "phony" dimensions are replaced with information found in the XML file that allows better interpretation of, for example, the "Radiance" variable. Therefore in Figure 4, the file created after steps 1 and 2 is called a "netCDF-4 meaningful file".

```
netcdf SVM07_ter_augmented-step1-step2 {

// global attributes:
```

```
                ……..

group: All_Data {
  group: VIIRS-M7-SDR_All {
    dimensions:
        AlongTrack = 768 ;
        CrossTrack = 3200 ;
        …
    variables:
        int AlongTrack(AlongTrack) ;
        …
        ….
        float Radiance(AlongTrack, CrossTrack) ;
                …
                Radiance:DatumOffset = 0 ;
                Radiance:Scaled = 1 ;
                string Radiance:MeasurementUnits = "W/(m^2 µm sr)" ;
        ushort Reflectance(AlongTrack, CrossTrack) ;
                string Reflectance:Description = "Calibrated Top of Atmosphere (TOA) Reflectance … ;
                ….
} // group VIIRS-M7-SDR_All
  } // group All_Data
}
```

The presence of dimensional information is not sufficient for tools like IDV to visualize data, and the third step described in the next section is required.

3.4.3 Updating with Geolocation Information

Geolocation information is required in the file if a user would like to visualize and analyze data with many netCDF-based applications. In this step, geolocation information becomes available in the NPOESS product file in the location where visualization tools like IDV expect to find it.

For non-packed NPOESS data, geolocation information is stored in a separate file as shown on Figure 4. An NPOESS data product file contains information about that file, and therefore, the augmentation tool does not require the name of the file on input.

The augmentation tool copies datasets with the names “Longitude”, “Latitude”, and, if present, “Height” from the geolocation file to the group /All_Data/<XXX>_All in the NPOESS product file. In the group name, <XXX> is a value of the CollectionShortName XML element found in the XML file. For example, in the file shown in Figure 3, this information would be copied to the /All_Data/VIIRS-M7-SDR_All group.

Since geolocation datasets may be very large, this step will increase file size and will affect the tool’s performance.

3.4.4 Future Enhancements

One should note here that for many tools more modification is required. For example, to visualize dataset “Radiance” with IDV one has to add the attributes shown in Table 1. Also all datasets have to be under the Root group.

Those types of modifications depend on visualization tool requirements. We recommend leaving further modifications out of the initial implementation. Additional changes can be added later when the set of desired tools and their requirements are identified.

Attributes can currently be added using tools such as HDFView or a new tool under development h5edit [5] as shown below:

```
% h5edit -c "CREATE /Radiance scale_factor {H5T_IEEE_F32LE SIMPLE(1) DATA{2.8339462E-4}};" f.h5
% h5edit -c "CREATE /Longitude units {H5T_STRING { STRSIZE 12 } DATA {'degrees_east'}};” f.h5
```

3.4.5 CF Conventions

Metadata stored in the NPOESS XML and NPOESS product files is not CF compliant. As a result, the file augmented as described in the sections 3.4.1 – 3.4.3, while readable by the netCDF-4 library, may not be fully accessible by the netCDF tools that rely on the CF conventions to find, visualize and analyze data. Ideally the XML and product files should have metadata that follows CF conventions, instead of the requiring the augmentation tool to fix this problem.

For a more comprehensive discussion of CF conventions we refer the reader to [6]. Here also we recommend leaving the CF conventions outside the scope of the initial implementation of the augmentation tool.

3.5 Exit Codes and Error Handling

3.5.1 Exit Codes

The tool shall exit with the exit code 0 if no failures occur during execution.

The tool shall exit with the exit code 1 if a failure occurs during execution.

3.5.2 Error Handling

3.5.2.1 API Failures

The tool shall exit if an API call used to implement the tool’s functionality returns an error. The name of the failed API, the location of the call, and an informative message will be displayed.

3.5.2.2 Verification Failures

The tool shall exit with an error if verification of XML data against the data in the NPOESS product file fails. XML data to be verified is found in section 4.

3.6 Other Considerations (Non-functional Requirements and Design Constraints)

3.6.1 Memory Considerations

The tool will read and parse NPOESS XML product files in memory. Since the files are small (about 50K), the parsing step is not memory intensive. Processing geolocation files may require some care if the datasets to be added are large. If so, the tool may provide a tunable parameter such as a buffer size for performing buffered I/O.

3.6.2 Dependencies on the Third-party Libraries

The tool is to be built with HDF5 1.8.6 or later and the libxml2 library (http://xmlsoft.org/).

3.6.3 Operating Systems

The minimum set of operating systems that the tool will work on includes Linux 32- and 64-bit and AIX 32-bit. Other operating systems may be added later.

3.6.4 Tool Testing

The tool will be tested using automated procedures on the Linux 32- and 64-bit platforms and on the AIX 32-bit platforms using NPOESS/JPSS product files downloaded from the GRAVITE system and the netCDF-4.1 library (and later). The tool will also be tested manually with netCDF-4 based visualization tools (TBD).

3.6.5 Build System and Packaging

The tool will be built using the Makefile and h5cc compiler script that comes with the HDF5 1.8.6 (or later) binaries.

The tool will be distributed from The HDF Group web site along with other tools created for the NPOESS/JPSS project. Both source code and pre-built binaries will be distributed.

3.6.6 Documentation

Documentation will include a help page, a User’s Guide, and Reference Manual material.

3.6.7 License

Tools will come with the standard license for HDF software
http://www.hdfgroup.org/products/licenses.html

4 Mapping Specifications Version 1.0

This section describes the NPOESS XML to HDF5 mapping used by the augmentation tool to modify an NPOESS product file with the information from an NPOESS XML product file. The version of the mapping specification should be stored in the file as an attribute on the Root group; specific mapping will be added in the next revision of this document.

4.1 NPOESS XML to HDF5 Mapping Considerations

The mapping design was chosen to bypass some restrictions of netCDF-4 and/or HDF5:

- Choice of dataspace for HDF5 attributes

  NetCDF-4 does not support scalar dataspaces for numeric attributes; therefore, the simple dataspace type is used even when the attribute is a scalar. If in the future the netCDF-4 library lifts this restriction, mapping can be changed to use the scalar dataspace type for all attributes with scalar values.

- Naming convention for HDF5 dimension scale dataset

  XML elements mapped to the HDF5 dimension scales may not have unique names (the value of the XML “Name” attribute). If those elements have different values for the “MaxIndex” attribute, they cannot be mapped to the same HDF5 dimension scale. In this case each of those elements is mapped to an HDF5 dimension scale dataset with a unique name as specified in 4.2.4. The value of the XML “Name” attribute is stored in the “Label” attribute of the dimension scale. See section 4.3 in [2] for discussions of dimension scale labels.

- Naming convention for HDF5 attributes

  XML elements mapped to HDF5 attributes may not have unique names. If those elements are mapped to the HDF5 attributes attached to the same HDF5 object, names are constructed to assure uniqueness of the HDF5 attributes’ names as required by the HDF5 library. Sections 4.2.5 – 4.2.7 provide examples of the chosen naming conventions.

4.2 NPOESS XML to HDF5 Mapping Specification

The following sub-sections describe a mapping between XML elements found in the NPOESS product profile schema file and HDF5 objects.

To specify the mapping we used a table that should be read as follows:

- The first row contains the name of the element or complex type as present in the NPOESS product profile schema file.
- The second row shows a corresponding excerpt from the NPOESS product profile schema file.

- The rest of the table specifies an element using the XPath convention in the left column and the corresponding HDF5 object and its properties in the right column as shown in Table 2. The HDF5 object description follows the notation adopted in [3]. See the "References" section on page 36.

Element or Complex Type Name		
<Excerpt from the NPOESS product profile XSD file>		
XPath	**HDF5 Object**	
/Path	<Key word>: Description of required or optional HDF5 object	
	[property]	property
	[property]	property
	…..	……
	…..	…..

Table 2: Mapping table template

Excerpts from the NPOESS XML file D34862-03_NPOESS-CDFCB-X-Vol-III_D_VIIRS-M7-SDR-PP.xml and h5dump output of the augmented file SVM07_ter_d20101206_t2009584_e2011083_b0000-1_c20101206231443705497_grav_dev.h5 are used in the section below to illustrate the mapping.

4.2.1 Mapping Elements in the NPP/NPOESS Data Product

The primary NPOESS XML element "NPOESSDataProduct" is mapped by mapping its child elements "ProductName", "CollectionShortName", "DataProductID", and "ProductData".

"ProductName", "CollectionShortName", "DataProductID" elements are mapped to the HDF5 attributes on the Root group with the properties as shown in Table 3, Table 4, and Table 5.

The element "ProductData" is mapped as a complex type (see Table 6) and is discussed in section 4.2.2.

Element ProductName		
`<xs:element name="ProductName" type="xs:string" minOccurs="1" maxOccurs="1" />`		
XPath	**HDF5 Object**	
/ProductName	Required: HDF5 Attribute attached to Root group "/"	
	[dataspace]	Scalar H5S_SCALAR
	[type]	Fixed-length C string H5T_S1_C
	[name]	"Product name"
	[value]	Value Example: "VIIRS Moderate Resolution Band 7"

Table 3: Mapping "ProductName" element.

Element CollectionShortName		
<xs:element name="CollectionShortName" type="xs:string" minOccurs="1" maxOccurs="1" />		
XPath	**HDF5 Object**	
/CollectionShortName[5]	Required: HDF5 Attribute attached to Root group "/"	
	[dataspace]	Scalar H5S_SCALAR
	[type]	Fixed-length C string H5T_S1_C
	[name]	"Collection short name"
	[value]	Value Example: "VIIRS-M7-SDR"

Table 4: Mapping "CollectionShortName" element

Element DataProductID		
<xs:element name="DataProductID" type="xs:string" minOccurs="1" maxOccurs="1" />		
XPath	**HDF5 Object**	
/DataProductID	Required: HDF5 Attribute attached to Root group "/"	
	[dataspace]	Scalar H5S_SCALAR
	[type]	Fixed-length C string H5T_S1_C
	[name]	"Data Product ID"
	[value]	Value Example: "SVM7"

Table 5: Mapping "DataProductID" element.

Element ProductData	
<xs:element name="ProductData" type="ProductDataType" minOccurs="3" maxOccurs="3" />	
XPath	**HDF5 Object**
/ProductData	Required: Mapped by mapping complex "ProductDataType" type

Table 6: Mapping "ProductData" element.

[5] We should note here that if the element "CollectionShortName" is mapped, information becomes redundant. The original NPOESS product file has a group with the same name as a value of that element under the /Data_Products group. For example, in Figure 1, HDFview shows the VIIRS-M7-SDR group in the NPOESS product file, and the value of the CollectionShortName element is "VIIRS-M7-SDR" as it is shown in the excerpt from the NPOESS XML product file in Figure 5.

Figure 5 and Figure 6 below show the instances of those elements in the NPOESS product XML file and their representation as HDF5 attributes on a Root group in the augmented file.

XML Example:

Figure 5: Instances of the “ProductName”, “CollectionShortName” and “DataProductID”.

```
<NPOESSDataProduct xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="NPOESS_Product_Profile.xsd">

        <ProductName>VIIRS Moderate Resolution Band 7 SDR</ProductName>
        <CollectionShortName>VIIRS-M7-SDR</CollectionShortName>
        <DataProductID>SVM7</DataProductID>
        <ProductData>…..
```

h5dump Output:

```
        GROUP "/" {
        ……….
         ATTRIBUTE "Product name" {
              DATATYPE  H5T_STRING {
                    STRSIZE 33;
                    …
                    CTYPE H5T_C_S1;
                 }
              DATASPACE  SCALAR
              DATA {
                 (0):  " VIIRS Moderate Resolution Band 7"
                 }
        }
        ATTRIBUTE "Collection short name" {
              DATATYPE  H5T_STRING {
                    STRSIZE 13;
                    …
                    CTYPE H5T_C_S1;
                 }
              DATASPACE  SCALAR
              DATA {
                 (0):  " VIIRS-M7-SDR"
                 }

        }
        ATTRIBUTE "Data Product ID" {
              DATATYPE  H5T_STRING {
                    STRSIZE 5;
                    …
                    CTYPE H5T_C_S1;
                 }
              DATASPACE  SCALAR
              DATA {
                 (0):  " SVM7"
                 }

        }
        …….
```

Figure 6: Instances of the elements “ProductName”, “CollectionShortName”, “DataProductID” shown as attributes on the Root group in the augmented file.

4.2.2 Mapping Product Data Types

The NPOESS complex ProductDataType type is mapped by mapping its child elements as shown in Table 7. For the "Field" element there is always an existing HDF5 dataset in the NPOESS product file, so the element is mapped but not written.

Complex "ProductDataType" Type

```
<xs:element name="DataName" type="xs:string" minOccurs="1" maxOccurs="1" />
<xs:element name="Field" type="FieldType" minOccurs="1" maxOccurs="unbounded" />
```

XPath	HDF5 Object	
/ProductData/DataName	Required: **HDF5 Attribute** attached to the group "/All_Data/<CollectionShortName>_All" Example: /All_Data/ VIIRS-M7-SDR_All	
	[dataspace]	Scalar H5S_SCALAR
	[type]	Fixed-length C string H5T_S1_C
	[name]	"Data Name"
	[value]	Value; Example: "VIIRS M-Band SDR Data Product Profile"
/ProductData/Field	Required: **HDF5 Dataset** under the group "/All_Data/<CollectionShortName>_All" with the name that is a value of the /ProductData/Field/Name element Example: /All_Data/ VIIRS-M7-SDR_All/Radiance	
	[dataspace]	As of the existing dataset
	[type]	As of the existing dataset
	[name]	As of the existing dataset
	[value]	As of the existing dataset

Table 7: Mapping of the ProductDataType complex type.

XML example:

```
<ProductData>
       <DataName>VIIRS M-Band SDR Data Product Profile</DataName>
       <Field>
              <Name>Radiance</Name>
…….
```

Figure 7: Instance of the ProductData element

h5dump example:

```
….
GROUP "All_Data" {
   GROUP "VIIRS-M7-SDR_All" {
   ……
   ATTRIBUTE "DataName" {
               DATATYPE  H5T_STRING {
                     STRSIZE 37;
                     ….
                  }
               DATASPACE  SCALAR
               DATA {
               (0): " VIIRS M-Band SDR Data Product Profile"
               }
            }
DATASET "Radiance" {
….
```

Figure 8: Instance of the ProductData element shown as the HDF5 attribute "DataName" on the /All_Data/VIIRS-M7-SDR_All group and the dataset "Radiance.

4.2.3 Mapping Field Type

NPOESS XML elements of complex type “FieldType” are mapped as shown in Table 8.

Complex “FieldType” Type

```
<xs:element name="Name" type="xs:string" minOccurs="0" maxOccurs="1" />
<xs:element name="Dimension" type="DimType" minOccurs="0" maxOccurs="unbounded" />
<xs:element name="DataSize" type="DataSizeType" minOccurs="1" maxOccurs="1" />
<xs:element name="Datum" type="DatumType" minOccurs="1" maxOccurs="unbounded" />
```

XPath	HDF5 Object	
/ProductData/Field/Name	Required: **HDF5 Dataset** under the group “/All_Data/<CollectionShortName>_All” with the name that is a value of the /ProductData/Field/Name element Example: /All_Data/ VIIRS-M7-SDR_All/Radiance	
	[dataspace]	As of the dataset
	[type]	As of the dataset
	[name]	As of the dataset
	[value]	As of the dataset
/ProductData/Field/Dimension	See 3.3.1.4	
/ProductData/Field/DataSize	Not mapped; values of Count and type child elements can be used to verify size of [type] above	
/ProductData/Field/Datum	See 3.3.1.5	

Table 8: Mapping of the FieldType complex type

XML example:

```
<Name>Radiance</Name>
                        <Dimension>
                                <Name>AlongTrack</Name>
                                …
                        </Dimension>
                        <Dimension>
                                <Name>CrossTrack</Name>
                                 …
                        </Dimension>
                       ….
                        <Datum>
                                <Description>Calibrated Top of Atmosphere (TOA) Radiance for each
VIIRS pixel</Description>
```

Figure 9: Instance of the FieldType complex type

h5dump output:

```
          DATASET "AlongTrack" {
             DATATYPE  H5T_STD_I32LE
             DATASPACE  SIMPLE { ( 768 ) / ( 768 ) }
          ….
          DATASET "Radiance" {
              …..
             ATTRIBUTE "DIMENSION_LIST" {
                DATATYPE  H5T_VLEN { H5T_REFERENCE}
                DATASPACE  SIMPLE { ( 2 ) / ( 2 ) }
                DATA {
                (0): (DATASET 17306712 /All_Data/VIIRS-M7-SDR_All/AlongTrack ),
                (1): (DATASET 17309248 /All_Data/VIIRS-M7-SDR_All/CrossTrack )
                }
            }
```

Figure 10: Dataset "Radiance" has a dimension called "AlongTrack". The dimension is represented as an HDF5 dataset with the name "AlongTrack". The dimension dataset is in the same group as the "Radiance" dataset.

4.2.4 Mapping Dimension Type

Dimension information stored in the NPOESS XML file is brought to the NPOESS product file using the mapping shown in Table 9.

Complex "DimType" Type

```
<xs:element name="Name" type="xs:string" minOccurs="0" maxOccurs="1" />
<xs:element name="GranuleBoundary" type="xs:boolean" minOccurs="0" maxOccurs="1" />
<xs:element name="Dynamic" type="xs:boolean" minOccurs="1" maxOccurs="1" />
<xs:element name="MinIndex" type="xs:integer" minOccurs="1" maxOccurs="1" />
<xs:element name="MaxIndex" type="xs:integer" minOccurs="1" maxOccurs="1" />
```

XPath	HDF5 Object		
/ProductData/Field/Dimension/Name	Optional: **HDF5 Dataset** under the group "/All_Data/<CollectionShortName>_All"; used as a dimension scale for a dataset that corresponds to the /ProductData/Field/Name element		
	[dataspace]	[rank]	1
		[current dims]	Value of /ProductData/Field/Dimension/MaxIndex
		[max dims]	Value of /ProductData/Field/Dimension/MaxIndex
	[type]	32-bit integer H5T_NATIVE_INT	
	[name]	<Value of /ProductData/Field/Dimension/Name>[_<Value of maxdims>] $_{\text{opt}}$ Example: Granule, Granule_3	
	[value]	None	
/ProductData/Field/Dimension/Granule/ GranuleBoundary	Required: **HDF5 Attribute** attached to dataset above		
	[dataspace]	[rank]	1
		[current dims]	1
		[max dims]	1
	[type]	32-bit integer H5T_NATIVE_INT	
	[name]	"GranuleBoundary"	
	[value]	0 or 1	

Continued on the next page

Complex "DimType" Type (cont.)

```
<xs:element name="Dynamic" type="xs:boolean" minOccurs="1" maxOccurs="1" />
<xs:element name="MinIndex" type="xs:integer" minOccurs="1" maxOccurs="1" />
<xs:element name="MaxIndex" type="xs:integer" minOccurs="1" maxOccurs="1" />
```

XPath	**HDF5 Object**		
/ProductData/Field/Dimension/Dynamic	Required: **HDF5 Attribute** attached to dataset above		
	[dataspace]	[rank]	[dataspace]
		[current dims]	1
		[max dims]	1
	[type]	32-bit integer H5T_NATIVE_INT	
	[name]	"Dynamic"	
	[value]	0 or 1	
/ProductData/Field/Dimension/MinIndex	Not mapped; used in DS		
/ProductData/Field/Dimension/MaxIndex	Not mapped; should be the same as MinIndex		

Table 9: Mapping of the DimType complex type

XML example:

```
         <Dimension>
                 <Name>AlongTrack</Name>
                 <GranuleBoundary>1</GranuleBoundary>
                 <Dynamic>0</Dynamic>
                 …
          </Dimension>
```

Figure 11: Dimension with the name of AlongTrack is mapped to a dataset (see Figure 12)

h5dump output:

```
         DATASET "AlongTrack" {
          …
            }
            ATTRIBUTE "Dynamic" {
          …
            }
            ATTRIBUTE "GranuleBoundary" {
          ….
            }
```

Figure 12: Dimension is represented by a dataset. Child elements of Dimension are represented by HDF5 attributes on the Dimension dataset.

4.2.5 Mapping Datum Type

The DatumType complex type is mapped by mapping its child elements as shown in Table 10. All of the child elements are mapped to the HDF5 attributes on the dataset that corresponds to the “Field” element. Some attributes are optional as noted in the table.

Complex “DatumType” Type

```
<xs:element name="Description" type="xs:string" minOccurs="1" maxOccurs="1" />
<xs:element name="DatumOffset" type="xs:integer" minOccurs="1" maxOccurs="1" />
<xs:element name="Scaled" type="xs:boolean" minOccurs="1" maxOccurs="1" />
<xs:element name="ScaleFactorName" type="xs:string" minOccurs="0" maxOccurs="1" />
<xs:element name="MeasurementUnits" type="xs:string" minOccurs="0" maxOccurs="1" />
<xs:element name="RangeMin" type="xs:decimal" minOccurs="0" maxOccurs="1" />
<xs:element name="RangeMax" type="xs:decimal" minOccurs="0" maxOccurs="1" />
<xs:element name="DataType" type="xs:string" minOccurs="1" maxOccurs="1" />
<xs:element name="FillValue" type="FillValueType" minOccurs="0" maxOccurs="unbounded" />
<xs:element name="LegendEntry" type="LegendEntryType" minOccurs="0" maxOccurs="unbounded"
/>
```

XPath	**HDF5 Object**		
/ProductData/Field/Datum/Description	Required: **HDF5 Attribute** attached to dataset that corresponds to the /ProductData/Field/Name element		
	[dataspace]	Scalar H5S_SCALAR	
	[type]	Fixed-length C string H5T_S1_C	
	[name]	“Description”	
	[value]	Value	
/ProductData/Field/Datum/DatumOffset	Required: **HDF5 Attribute** attached to dataset that corresponds to the /ProductData/Field/Name element		
	[dataspace]	[rank]	1
		[current dims]	1
		[max dims]	1
	[type]	32-bit integer H5T_NATIVE_INT	
	[name]	“<Datum#_>DatumOffset”	
	[value]	Value	

Continued on the next page

Complex “DatumType” Type (cont.)

```
<xs:element name="Scaled" type="xs:boolean" minOccurs="1" maxOccurs="1" />
<xs:element name="ScaleFactorName" type="xs:string" minOccurs="0" maxOccurs="1" />
<xs:element name="MeasurementUnits" type="xs:string" minOccurs="0" maxOccurs="1" />
```

XPath	HDF5 Object		
/ProductData/Field/Datum/Scaled	Required: **HDF5 Attribute** attached to dataset that corresponds to the /ProductData/Field/Name element		
	[dataspace]	[rank]	1
		[current dims]	1
		[max dims]	1
	[type]	32-bit integer H5T_NATIVE_INT	
	[name]	“Scaled”	
	[value]	Value	
/ProductData/Field/Datum/ScaleFactorName	Optional: **HDF5 Attribute** attached to dataset that corresponds to the /ProductData/Field/Name element		
	[dataspace]	Scalar H5S_SCALAR	
	[type]	Fixed-length C string H5T_S1_C	
	[name]	““<Datum#_>ScaleFactorName”	
	[value]	Value	
/ProductData/Field/Datum/MeasurementsUnits	Optional: **HDF5 Attribute** attached to dataset that corresponds to the /ProductData/Field/Name element		
	[dataspace]	Scalar H5S_SCALAR	
	[type]	Fixed-length C string H5T_S1_C	
	[name]	““<Datum#_>MeasurementUnits”	
	[value]	Value	

Continued on the next page

Complex “DatumType” Type (cont.)

```
<xs:element name="RangeMin" type="xs:decimal" minOccurs="0" maxOccurs="1" />
<xs:element name="RangeMax" type="xs:decimal" minOccurs="0" maxOccurs="1" />
<xs:element name="DataType" type="xs:string" minOccurs="1" maxOccurs="1" />
<xs:element name="FillValue" type="FillValueType" minOccurs="0" maxOccurs="unbounded" />
<xs:element name="LegendEntry" type="LegendEntryType" minOccurs="0" maxOccurs="unbounded"
/>
```

XPath	HDF5 Object		
/ProductData/Field/Datum/RangeMin	Optional: **HDF5 Attribute** attached to dataset that corresponds to the /ProductData/Field/Name element		
	[dataspace]	[rank]	1
		[current dims]	1
		[max dims]	1
	[type]	32-bit integer H5T_NATIVE_INT	
	[name]	““<Datum#_>RangeMin”	
	[value]	Value	
/ProductData/Field/Datum/RangeMax	Optional: **HDF5 Attribute** attached to dataset that corresponds to the /ProductData/Field/Name element		
	[dataspace]	[rank]	1
		[current dims]	1
		[max dims]	1
	[type]	32-bit integer H5T_NATIVE_INT	
	[name]	““<Datum#_>RangeMax”	
	[value]	Value	
/ProductData/Field/Datum/DataType	Not mapped; can be used for verification purposes to verify correct datatype of the dataset that corresponds to the /ProductData/Field/Name element		
/ProductData/Field/Datum/FillValue	See Section 3.3.1.6		
/ProductData/Field/Datum/LegendEntry	See Section 3.3.1.7		

Table 10: Mapping of the DatumType type

XML example:

```
<Datum>
   <Description>Calibrated Top of Atmosphere (TOA) Radiance for each VIIRS pixel</Description>
   <DatumOffset>0</DatumOffset>
   <Scaled>1</Scaled>
   <ScaleFactorName>RadianceFactors</ScaleFactorName>
   <MeasurementUnits>W/(m^2 µm sr)</MeasurementUnits>
     <DataType>unsigned 16-bit integer</DataType>
     <FillValue>
               <Name>NA_UINT16_FILL</Name>
               <Value>65535</Value>
     </FillValue>
```

Figure 13: Datum element and its child elements in the XML file; child elements are mapped to the attributes on an HDF5 datasets. Datum element mapped to a dataset (see “Radiance” on Figure 14)

h5dump output:

```
DATASET "Radiance" {
…
    }
   ATTRIBUTE "DatumOffset" {
        …
   ATTRIBUTE "Description" {
   …
   ATTRIBUTE "MeasurementUnits" {
   ….
   ATTRIBUTE “FillValue_NA_UINT16_FILL {
```

Figure 14: Dataset “Radiance” corresponds to the Datum element of “Radiance” Field. Child elements of Datum (see Figure 13) are mapped to the HDF5 attributes

4.2.6 Mapping the Fill Value Type

The FillValueType complex type is mapped by mapping its child elements as shown in Table 11.

Complex "FillValueType" Type

```
<xs:element name="Name" type="xs:string" minOccurs="1" maxOccurs="1" />
<xs:element name="Value" type="xs:double" minOccurs="1" maxOccurs="1" />
```

XPath	HDF5 Object		
/ProductData /Field/Datum/FillValue/Name	Required: **HDF5 Attribute** attached to the dataset corresponding to the /ProductData/Field/Name element Example: /All_Data/ VIIRS-M7-SDR_All/Radiance		
	[dataspace]	[rank]	1
		[current dims]	1
		[max dims]	1
	[type]	Matches [type] in dataset	
	[name]	""<Datum#_>$_{opt}$FillValue_VALUE", where VALUE is a value of /ProductData/DataName/Field/Datum/FillValue/Name; Example "FillValue_MISS_UINT16_FILL"	
	[value]	Value of /ProductData/DataName/Field/Datum/FillValue/Value	
/ProductData/DataName/Field/Datum/FillValue/Value	Used above for [value]		

Table 11: Mapping of the FillValueType type

XML example:

```
          <FillValue>
                   <Name>NA_UINT16_FILL</Name>
                   <Value>65535</Value>
           </FillValue>
           <FillValue>
                    <Name>MISS_UINT16_FILL</Name>
                    <Value>65534</Value>
           </FillValue>
           <FillValue>
                    <Name>ONBOARD_PT_UINT16_FILL</Name>
                    <Value>65533</Value>
            </FillValue>
```

Figure 15: The Fill value instances

h5dump example:

```
DATASET "Radiance" {
…..

    ATTRIBUTE "FillValue_NA_UINT16_FILL" {
       …
    }
    ATTRIBUTE "FillValue_MISS_UINT16_FILL" {
       …
    }
    ATTRIBUTE "FillValue_ONBOARD_PT_UINT16_FILL" {
       …
    }
```

Figure 14: Fill value attributes

4.2.7 Mapping Legend Entry Type

The LegendEntryType complex type is mapped by mapping its child elements as shown in Table 12.

Complex “LegendEntryType” Type			
<xs:element name="Name" type="xs:string" minOccurs="1" maxOccurs="1" /> <xs:element name="Value" type="xs:double" minOccurs="1" maxOccurs="1" />			
XPath	**HDF5 Object**		
/ProductData /Field/Datum/LegendEntry/Name	Required: **HDF5 Attribute** attached to the dataset corresponding to the /ProductData/Field/Name element Example: /All_Data/ VIIRS-M7-SDR_All/ModeScan		
	[dataspace]	[rank]	1
		[current dims]	1
		[max dims]	1
	[type]	double H5T_NATIVE_DOUBLE	
	[name]	““<Datum#_>$_{opt}$LegendEntry_VALUE”, where VALUE is a value of /ProductData/DataName/Field/Datum/LegendEntry/Name; Example “LegendEntry_Night”	
	[value]	Value of /ProductData/DataName/Field/Datum/LegendEntry/Value	
/ProductData /Field/Datum/ LegendEntry /Value	Used above for [value]		

Table 12: Mapping of the LegendEntry Complex Type

XML example:

```
     <Field>
           <Name>ModeScan</Name>
           …….
           <LegendEntry>
                      <Name>Night</Name>
                      <Value>0</Value>
           </LegendEntry>
            <LegendEntry>
                      <Name>Day</Name>
                  <Value>1</Value>
            </LegendEntry>
```

Figure 16: Instance of the LegendEntry Complex Type

h5dump output:

```
   DATASET "ModeScan" {
   …….
      ATTRIBUTE  “LegendEntry_Night" {
         …
      }
      ATTRIBUTE "LegendEntry_Day" {
```

Figure 17: h5dump output for LegendEntry

Revision History:

March 10, 2011:	Version 1 circulated for comment to Mike and Larry.

References

1. The netCDF-4 Format. http://www.unidata.ucar.edu/software/netcdf/docs/netcdf/NetCDF_002d4-Format.html#NetCDF_002d4-Format
2. The HDF Group. ""HDF5 Dimension Scale Specification" http://www.hdfgroup.org/HDF5/doc/HL/H5DS_Spec.pdf
3. Mike Folk, Gerd Herber, Quincey Koziol. "HDF5 Information Set", to be published.
4. HDF5 Command Line Tools http://www.hdfgroup.org/products/hdf5_tools/
5. H5edit tool (under development) http://www.hdfgroup.org/projects/npoess/
6. Kent Yang. "CF support for JPSS augmented files", http://www.hdfgroup.uiuc.edu/ftp/pub/outgoing/NPOESS/augmentation-tool/CF-JPSS.pdf.

Appendix: Specifications for Storing Information of a Hidden Group

HDF5 Link information			
HDF5 Attribute on Root group	[dataspace]	[rank]	1
		[current dims]	1
		[max dims]	1
	[type]	64-bit unsigned integer H5T_NATIVE_ULLONG	
	[name]	“HDF5_interal_address_of_disconnected_group_with_reference_types”	
	[value]	Hard link value (address of the group’s object header)	
HDF5 Attribute on Root group	[dataspace]	Scalar	
	[type]	Fixed-length C string H5T_S1_C	
	[name]	“HDF5_interal_name_of_disconnected_group_with_reference_types”	
	[value]	Path to the group (Example: “/Data_Producs”)	

Table 13: Storing information about a hidden group in HDF5 file

Both attributes are given names that could easily distinguish them from other attributes stored in the NPOESS product file.