Request for Comments
HDF5 Augmentation Tool for JPSS Product Files
Requirements, Design and Mapping Specifications

Mike Folk, Elena Pourmal, Larry Knox
The HDF Group

JPSS data products will be distributed in the HDF5 file format. HDF5 was chosen to accommodate the volume and complexity of JPSS data and the enormously high speed at which it must be processed. The distributed data will be used for weather forecasting and in climate research. However, the majority of applications in this area employ the netCDF data model and file format for data storage, and therefore cannot easily access the distributed JPSS products.

The HDF Group was charged with the task to make JPSS data available to netCDF-based applications and tools without rewriting JPSS data products into the netCDF file format.

The solution became feasible because the new version of the netCDF library, netCDF-4, uses the HDF5 file format for data storage and the HDF5 library for I/O while maintaining backward compatibility with the netCDF APIs.

The solution requires that two obstacles be removed first: 1) JPSS data products do not conform to the netCDF classical model or to the netCDF-4 models, and therefore cannot be accessed by the netCDF-4 library as is; 2) many netCDF-based visualization tools require a presence of additional metadata information to interpret and visualize data.

The HDF Group developers propose to create an HDF5 augmentation tool to remove those obstacles. This document identifies requirements for the tool, describes its design and specifies changes done by the tool to a JPSS product file to make it accessible by the netCDF library and netCDF-based tools.

This version of the document focuses on augmenting non-aggregated and non-packaged JPSS product files.[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 Overview 3
2 Use Cases 4
2.1 Display the contents of a JPSS data product using ncdump 4
2.2 Hide objects blocking netCDF-4 opening the file, then display contents with ncdump 5
2.3 Add important metadata, then display with ncdump 6
2.4 Hide HDF5 groups for netCDF-4 tools and applications requiring netCDF-4 Classic format files. 7
2.5 Add geolocation information, then display with geographic background map. 9
2.6 Special needs: visualizing data with IDV 10
3 Approach: A Tool to Modify JPSS Product Files for NetCDF Accessibility 14
4 Functional Requirements and Assumptions for a JPSS Augmentation Tool 16
4.1 Functional requirements 16
4.2 Assumptions 17
5 Design 19
5.1 Command-line options and public API 19
5.2 Overview of augmentation steps 19
5.3 Augmentation steps in detail 24
5.3.1 NetCDF readable: hiding objects unknown to the netCDF-4 library 24
5.3.2 NetCDF meaningful: updating with information from XML file 25
5.3.3 NetCDF classic conformant 27
5.3.4 Geolocation conformant: updating with geolocation information 27
5.4 Non-functional requirements and design constraints 28
5.4.1 Exit codes and error handling 28
5.4.2 Memory considerations 28
5.4.3 Dependencies on third-party libraries 29
5.4.4 Operating systems 29
5.4.5 Tool Testing 29
5.4.6 Build system and packaging 29
5.4.7 Documentation 29
5.4.8 License 29
6 Mapping Specifications Version 1.0 30
6.1 JPSS XML-to-HDF5 mapping considerations 30
6.2 JPSS XML-to-HDF5 mapping specification 31
6.2.1 Mapping NPP/JPSS Data Product 32
6.2.2 Mapping Product Data Types 35
6.2.3 Mapping Field Type 37
6.2.4 Mapping Dimension Type 39
6.2.5 Mapping Datum Type 41
6.2.6 Mapping FillValue Type 44
6.2.7 Mapping LegendEntry Type 46
Appendix: Specifications for Storing Information of a Hidden Group 48
References 49

1 Overview

Problem and approach. JPSS data products will be distributed in the HDF5 file format. However, many applications that will use this data are *netCDF-based*. That is, many popular data analysis and visualization applications used by climate and weather forecasting and research communities assume that their data conform to the netCDF data model. In addition, many programmers prefer to implement applications using a netCDF application programming interface (API).

Fortunately, the new version of the netCDF library, netCDF-4, uses HDF5 as its storage layer. This capability gives hope that a remedy might be found that would allow both netCDF and HDF5 applications to access JPSS data products. In order to exploit this, we need to overcome some obstacles, and that is the purpose of this project.

The first obstacle is that netCDF-4 cannot read arbitrary HDF5 files. Applications that use the netCDF-4 API can only read files that conform to the netCDF-4 data model. (For more on this, see "The netCDF Format" [1].) Since JPSS product files do not conform to the netCDF-4 data model, the netCDF-4 library cannot access them. This means, in turn, that applications that use the netCDF-4 library also cannot access JPSS product files.

To complicate matters, many applications need more than just a netCDF-4 accessible file. These tools rely on metadata such as dimension variables, geolocation information and miscellaneous attributes to interpret and display the data. Since this metadata is not present in JPSS product files, additional modifications may be needed before data can be meaningfully analyzed and visualized. Fortunately much of this information is available in JPSS XML product files and in JPSS geolocation files.

Finally, it must be noted that there are two variants of netCDF, the *netCDF classic model* and the *netCDF-4 model.* Both models are supported by netCDF-4. The netCDF classic model is more restrictive than the netCDF-4 model. Any application that can work with the netCDF-4 model can also work with the netCDF classic model, but the converse is not true. This means that simply modifying a JPSS product file to make it readable by netCDF-4 may not be sufficient. For some applications the file may further need to be modified to make it compliant with the netCDF classic model.

We have decided to address these obstacles by developing an augmentation tool that can modify JPSS files in ways that satisfy the needs of many netCDF-based applications.

What the tool does not do. As will be seen, the tool will not be sufficient for all cases, and other remedies may be needed. In this document, we give an example using IDV in which additional information must be added to a JPSS product file before it can be viewed.

Furthermore, we do not currently address compliance with climate and forecast (CF) metadata conventions.[2] Because metadata stored in the JPSS XML and JPSS product files are not CF compliant, files modified by the tool may not be fully accessible by netCDF tools that rely on the CF conventions to find, visualize and analyze data. For a more comprehensive discussion of CF conventions, refer to "CF Support for JPSS Augmented Files" [5]. Here, we recommend leaving the CF conventions outside the scope of the initial implementation of the augmentation tool.

Organization of the document. This document is organized as follows: Section 2 discusses use cases that justify the development of the augmentation tool; Section 3 gives a brief overview of the tool's requirements; Section 4 discusses in detail assumptions and functional requirements; Section 5 focuses on the tool's design; Section 6 specifies a mapping from the JPSS product data schema file to the HDF5 objects used by the tool.

Required background. It is not necessary to know HDF5 or netCDF in order to follow this material, but a basic understanding of netCDF and HDF5 would be helpful. For a primer on the two formats, see *The NetCDF Tutorial* (http://www.unidata.ucar.edu/software/netcdf/docs/netcdf-tutorial.html) and *Introduction to HDF5* (http://www.hdfgroup.org/HDF5/doc/H5.intro.html).

Location of the h5augjpss software and sample files. The prototype implementation discussed in this document can be downloaded from http://www.hdfgroup.org/projects/npoess/h5augjpss_index.html. The JPSS product schema file, the JPSS XML product file, and JPSS 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/h5augjpss_example/.

2 Use Cases

The following scenarios describe problems that can occur when netCDF-based applications access JPSS data products.

2.1 Display the contents of a JPSS data product using ncdump

Ncdump is a utility that writes a text representation of a netCDF file to standard output. Ncdump uses the netCDF library to access files.

If a user downloads a JPSS data product file and tries to display the file with the netCDF-4 ncdump utility, ncdump will fail as shown:

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

[2] See "The CF Metadata Standard," by Jonathan Gregory, 2003: http://cf-pcmdi.llnl.gov/documents/other/cf_overview_article.pdf. See also http://en.wikipedia.org/wiki/Climate_and_Forecast_Metadata_Conventions.

Ncdump fails because the file contains data objects that are not part of the netCDF data model, and hence are unknown to the netCDF library. More specifically, the JPSS product file contains a “Data_Products” group, which contains objects that the netCDF-4 library does not recognize. As a result, the netCDF-4 library fails to open the file.

2.2 Hide objects blocking netCDF-4 opening the file, then display contents with ncdump

Let us suppose that we remove those objects in the JPSS product file that caused the error. We can delete the link to the “Data_Products” group in the HDF5 file hierarchy, which makes this group and its contents invisible in the file to netCDF-4.[3]

After the group is hidden, 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
}
```

Figure 1: ncdump view of “netCDF readable” file, after removing objects that would cause netCDF to fail to open.

We have cleared our first hurdle, which is to modify the HDF5 in such a way that a netCDF-based application can read it. We will refer to this type of file as “netCDF readable”.

[3] Notice that we have not removed the group. We have only removed the link to the group. Information about the hidden group can be preserved by storing it as attributes in the JPSS product file, in case we would later want to recreate the original file.

However, notice the highlighted lines in the dump shown on Figure 1. The JPSS product file contained no dimension information (coordinate variables, in netCDF parlance) for the variables. Because of this, ncdump creates dimensions for the variables with the names “phony_dim_3” and “phony_dim_4.”

2.3 Add important metadata, then display with ncdump

Dimension information can be important in order to have an understanding of the data, and it is an important component of the netCDF data model. Fortunately, this dimension information is in fact available, though not in the product file. JPSS stores this information in a separate metadata file called the JPSS “XML product file.”

The XML product file contains more than just dimension information, however. It also includes the name of the product, the name of the collection the product belongs to, a product identifier, measurement units, and fill values.

In the same way that we removed information from the HDF5 file in the previous step, we can add information in this step. That is, we can add dimension information to the JPSS product file in such a way that netCDF associates it appropriately with the proper variables. We can also copy other useful information from the XML product file to the JPSS product file.

If this step is done in addition to the previous step, ncdump finds the dimensional information needed to display the file content in a more meaningful way. Other metadata brought from the XML file enhance the meaning of the data even further. The results of doing this are illustrated in the fragment below (see Figure 2), where the new information is highlighted.

```
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
}
```

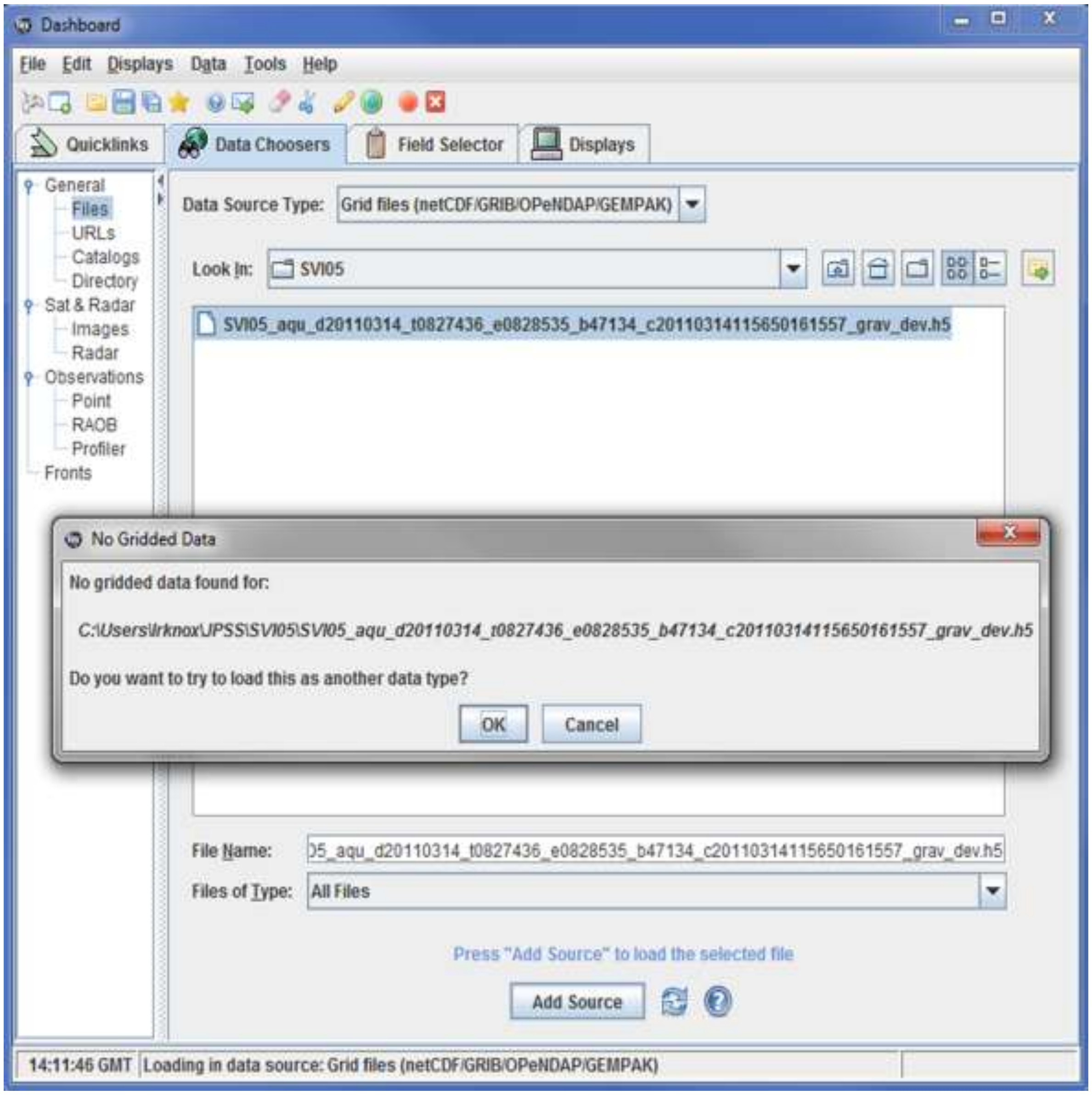
Figure 2: ncdump view of an HDF5 file modified to be "netCDF meaningful". Examples of new dimension information and other metadata brought from the XML file are highlighted in yellow; an example of redundant information is highlighted in grey. Notice that "phony" dimensions are replaced by information found in the XML file that allows better interpretation of, for example, the "Radiance" variable.

We will refer to a file that is both "netCDF readable" and has dimension and other information inserted in this way as a "netCDF-4 meaningful file".

2.4 Hide HDF5 groups for netCDF-4 tools and applications requiring netCDF-4 Classic format files.

NetCDF-4-based applications that require netCDF-4 classic format files will not open HDF5 files that have groups. This inability includes versions of IDV prior to 3.0b1 and Panoply prior to 3.04a, as well as any application that carries the NetCDF expectation that all variables are at the top level of the file. Using the augmentation tool to hide all groups in JPSS files removes the HDF5 obstacles blocking these applications from opening the files. Individual applications may require additional attributes to visualize a dataset.

Figure 3 shows the error thrown by IDV when opening an original NPP file. After a file is augmented to hide the group hierarchy and to add geolocation data, IDV successfully opens the file and displays the "Radiance" dataset as shown on Figure 4.



Data Source Type: Grid files (netCDF/GRIB/OPeNDAP/GEMPAK)

Look In: SVI05

SVI05_aqu_d20110314_t0827436_e0828535_b47134_c20110314115650161557_grav_dev.h5

No Gridded Data

No gridded data found for:

C:\Users\lrknox\JPSS\SVI05\SVI05_aqu_d20110314_t0827436_e0828535_b47134_c20110314115650161557_grav_dev.h5

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

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

14:11:46 GMT Loading in data source: Grid files (netCDF/GRIB/OPeNDAP/GEMPAK)

Figure 3: IDV throws an error when opening an HDF5 file that has groups

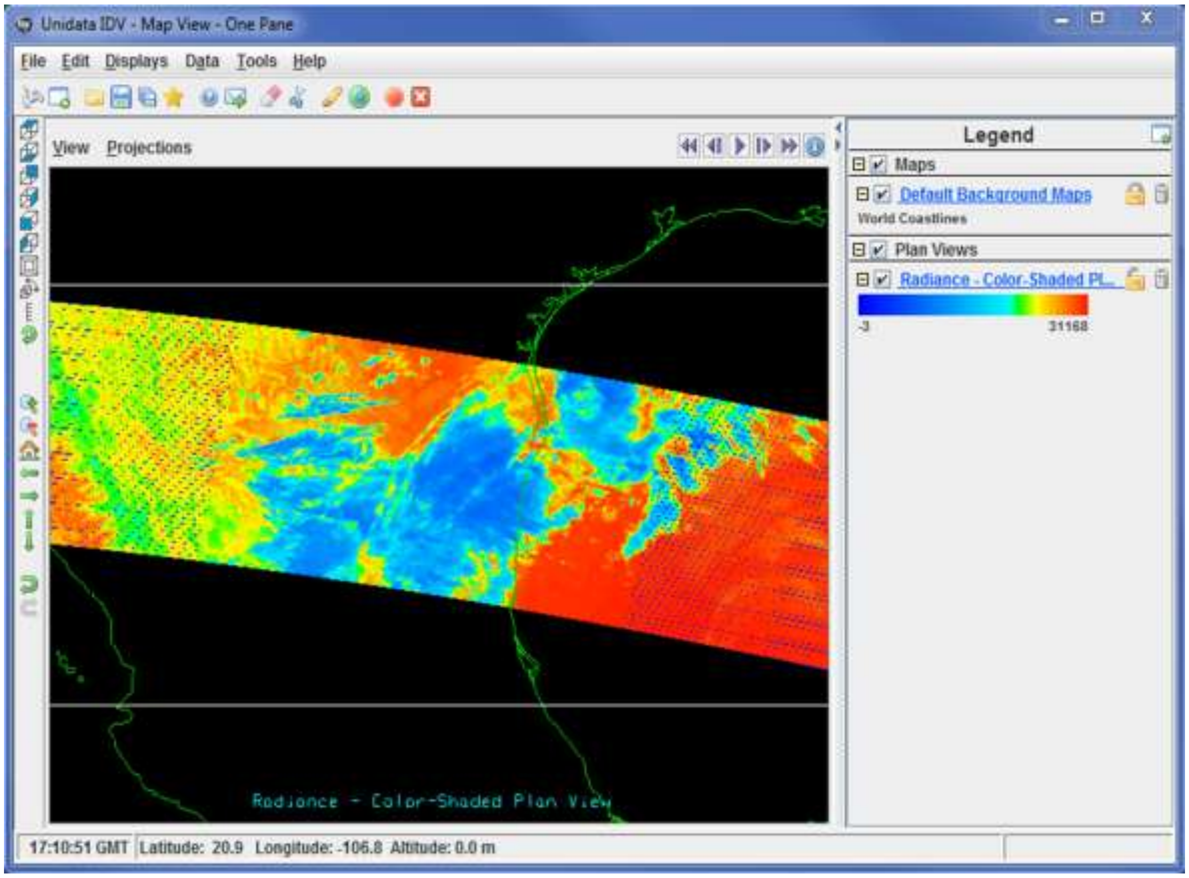

Unidata IDV - Map View - One Pane
File Edit Displays Data Tools Help
View Projections
Legend
Maps
Default Background Maps
World Coastlines
Plan Views
Radiance - Color-Shaded Pl...
-3
31168
Radiance - Color-Shaded Plan View
17:10:51 GMT Latitude: 20.9 Longitude: -106.8 Altitude: 0.0 m

Figure 4: IDV displays "Radiance" dataset after the file was augmented to remove group hierarchy and to add geolocation datasets.

2.5 Add geolocation information, then display with geographic background map.

Applications such as IDV and Panoply that provide visualization of datasets (variables) in the context of a geographic map will not display the data without the geolocation information. The augmentation tool can be used to add the geolocation information to the JPSS product file.

With the latest versions, IDV 3.0b1 and Panoply 3.04a, nothing other than "adding geolocation information" augmentation is required to display the data in the correct geographic location, as shown on Figure 5. Scientifically meaningful visualization of the datasets requires adding information from the XML file such as the dimension scales along with the attributes discussed in the following section.

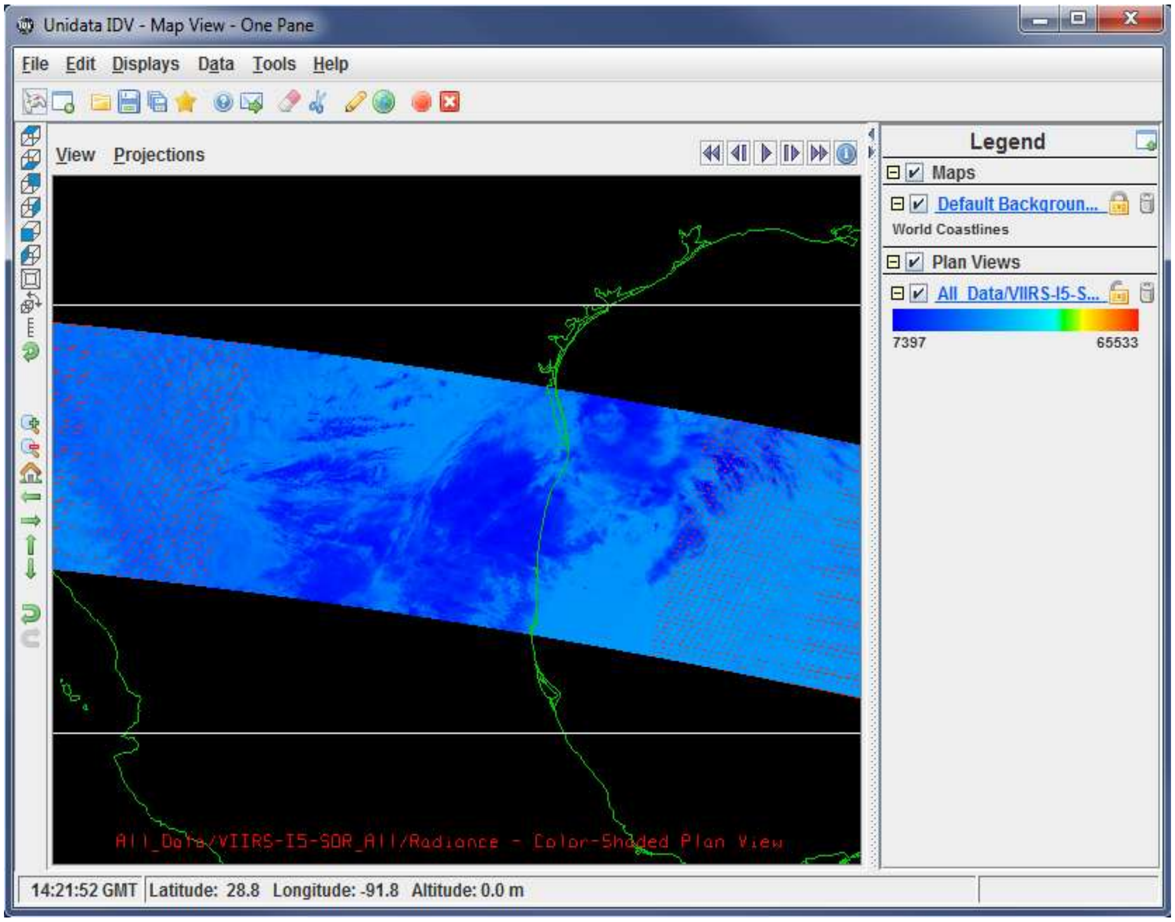
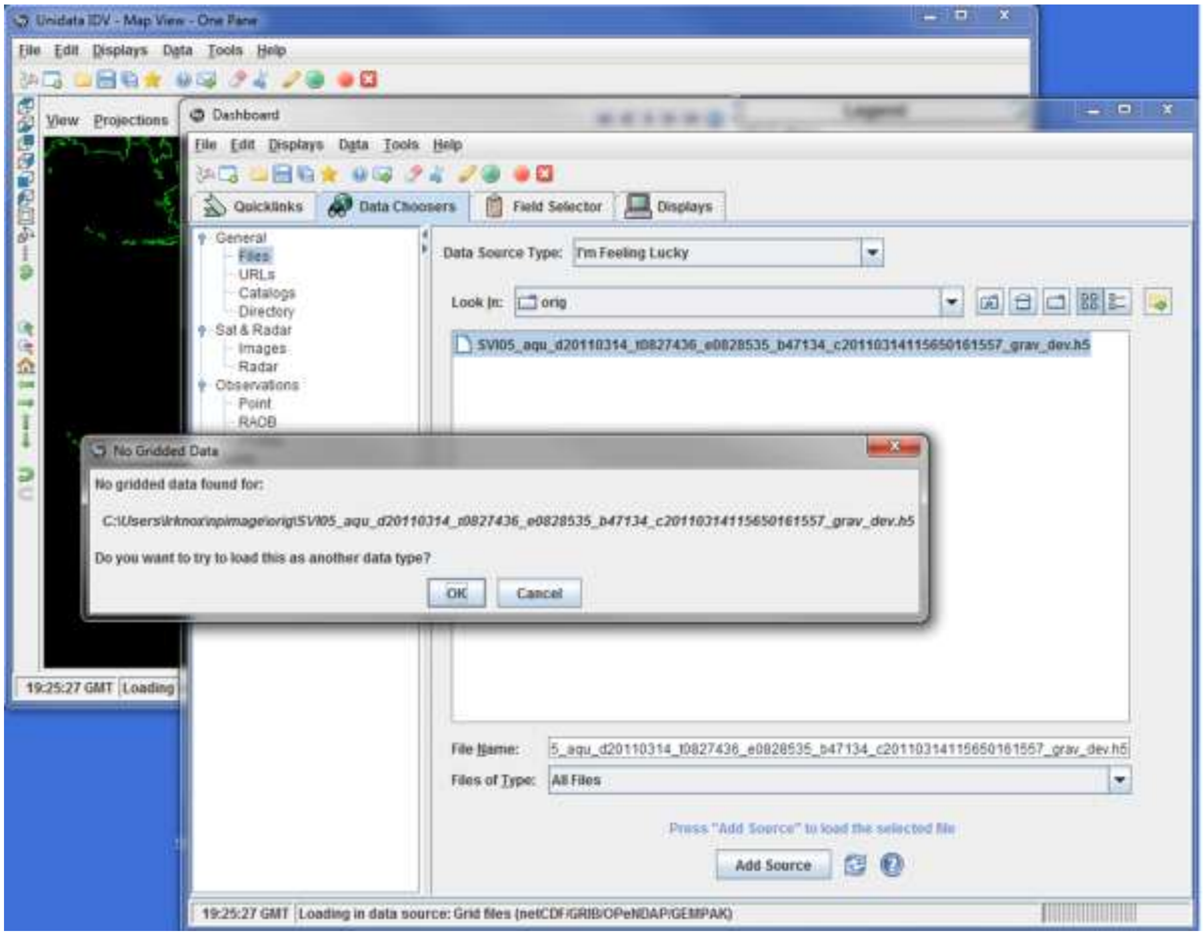

Unidata IDV - Map View - One Pane
File Edit Displays Data Tools Help
View Projections
Legend
Maps
Default Backgroun...
World Coastlines
Plan Views
All Data/VIIRS-I5-S...
7397
65533
All_Data/VIIRS-I5-SDR_All/Radiance - Color-Shaded Plan View
14:21:52 GMT Latitude: 28.8 Longitude: -91.8 Altitude: 0.0 m

Figure 5: IDV version 3.0b1 requires geolocation datasets to display data in the correct location.

2.6 Special needs: visualizing data with IDV

Note: *the following use case is included because it illustrates an important step that may be needed for some applications to work with JPSS data products. This case is* *<u>not</u>* *completely addressed with the h5augjpss tool that is the focus of this document. However, remedies are often available, such as the remedy applied in this use case.*

Sometimes applications have special metadata requirements that are not satisfied by the preceding steps. For example, suppose users want to visualize the “Radiance” variable in our file with the IDV (Integrated Data Viewer, http://www.unidata.ucar.edu/software/idv/). They run the HDF5 augmentation tool to make the file accessible by the netCDF-4 library and to add geolocation information to the file. Nevertheless, IDV fails, as shown on Figure 6 due to the absence of certain required attributes on the “Latitude”, “Longitude”, and “Radiance” variables.


Unidata IDV - Map View - One Pane
File Edit Displays Data Tools Help
View Projections
Dashboard
Quicklinks
Data Choosers
Field Selector
Displays
General
Files
URLs
Catalogs
Directory
Sat & Radar
Images
Radar
Observations
Point
RAOB
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
Add Source
19:25:27 GMT
Loading
19:25:27 GMT Loading in data source: Grid files (netCDF/GRIB/OPeNDAP/GEMPAK)
No Gridded Data
OK
Cancel

No gridded data found for:

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

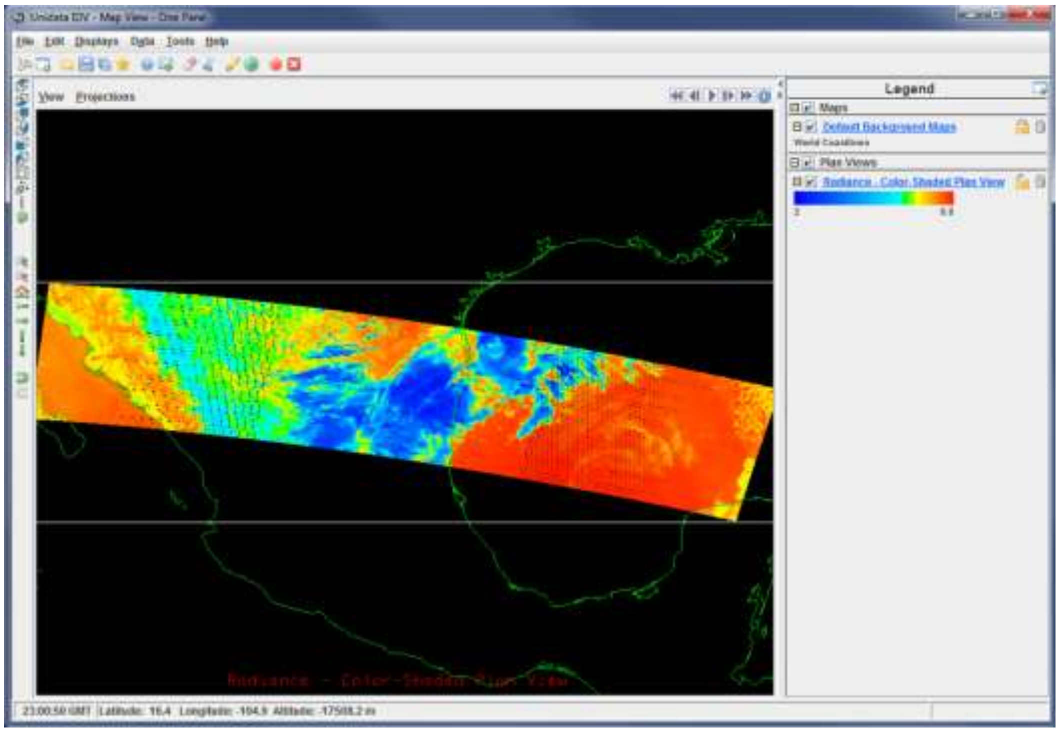
Do you want to try to load this as another data type?

Figure 6: IDV fails to show data in the augmented file due to the absence of required attributes.

If the user adds the attributes required by IDV, then IDV can properly display the data of the "Radiance" variable as shown on Figure 7. These attributes and their properties are shown in Table 1[4].

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" variables requried by IDV


Unidata IDV - Map View - One Panel
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 7: "Radiance" variable shown by IDV in the augmented file.

A new tool called h5edit is under development that can add such attributes to an HDF5 file. For instance, the "scale_factor" and "units" attributes can be added as follows:

[4] 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. .

```
% 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 Approach: A Tool to Modify JPSS Product Files for NetCDF Accessibility

Our approach to achieving interoperability with netCDF applications will be to provide a command-line tool called h5augjpss to modify JPSS product files in the ways described in the foregoing use cases.

Note: *As the last use case ("visualizing data in IDV") shows, there will be cases in which the information needed is specific to a certain tool, and cannot have been predicted. In many such cases, the addition of one or more netCDF attributes will be sufficient to achieve the desired purpose. When that is the case, it is recommended that tools be used such as HDFView or the new tool, which is under development, h5edit [4].*

The remainder of this document will focus on h5augjpss.

The types of JPSS product file modification can be categorized as follows:

NetCDF-4 readable. The netCDF-4 version library can open the file without failure.
NetCDF meaningful. Ncdump can display information that is scientifically meaningful and conforms to the netCDF-4 data model.
NetCDF classic conformant. The file conforms to the requirements of netCDF classic.
Geolocation conformant: The file contains information allowing variables to be mapped to earth coordinates.

These modifications take the following general forms:

1. Removal of objects from JPSS HDF5 product files.
2. Rearrangement of objects in JPSS HDF5 product files.
3. Addition of information to JPSS HDF5 product files from the XML product file, the geolocation file, and possibly other sources as needed by a particular application.

H5augjpss is to perform modifications in ordered steps, each step being a command-line option:

Step 1. Hide HDF5 objects unknown to netCDF-4 to make the file netCDF readable.
Step 2. Add dimensional information and other metadata to the file, obtained from the JPSS XML product file to make the fine netCDF meaningful.
Step 3. Add geolocation information to make the file geolocation conformant.
Step 4. Remove group structure to make the file netCDF classic compliant.

Figure 8 illustrates the relationships among these steps when no options are given on the command line (default behavior) and corresponding input and output files.

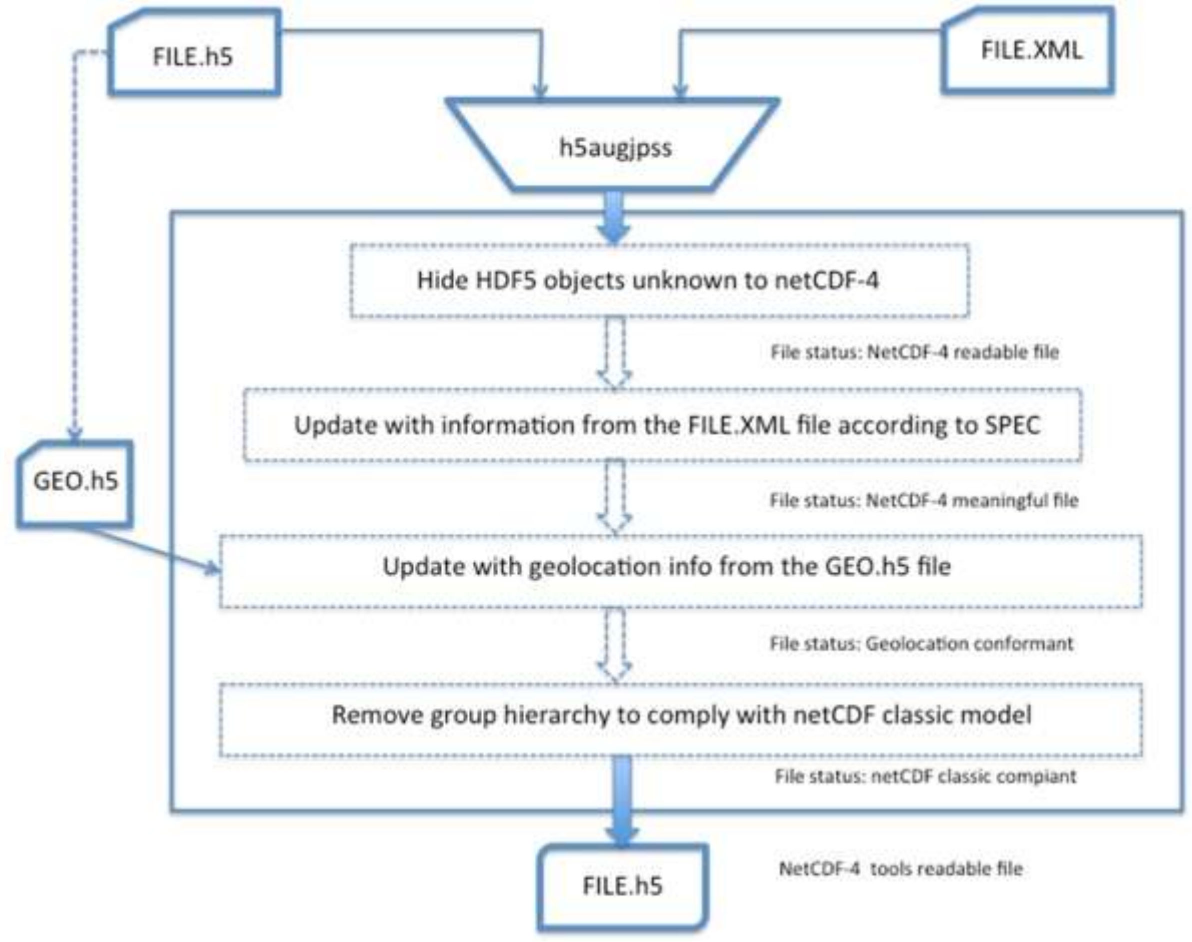

FILE.h5
FILE.XML
h5augjpss
Hide HDF5 objects unknown to netCDF-4
File status: NetCDF-4 readable file
Update with information from the FILE.XML file according to SPEC
File status: NetCDF-4 meaningful file
GEO.h5
Update with geolocation info from the GEO.h5 file
File status: Geolocation conformant
Remove group hierarchy to comply with netCDF classic model
File status: netCDF classic compiant
FILE.h5
NetCDF-4 tools readable file

Figure 8: 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 such as ncdump and IDV.

Each of the steps described above is specified by a command line option and can be performed independently as described in section 5.2.

4 Functional Requirements and Assumptions for a JPSS Augmentation Tool

The functional requirements and assumptions are summarized here. Non-functional requirements and design constraints are included in Section 5.

4.1 Functional requirements

The following functional requirements specify the changes to be made to JPSS files by the augmentation tool, and the behavior of the tool.

1. The tool shall be capable of modifying (augmenting) a JPSS product file so that it is
   - **NetCDF-4 readable**. The netCDF-4 version 4.1.3 library shall be able to open the file without failure.
   - **NetCDF meaningful**. Ncdump version 4.1.3 shall display information that is scientifically meaningful and in accordance with the netCDF-4 data model. This includes adding meaningful dimension names, as well as certain key metadata as described in Section 6.
   - **NetCDF classic conformant.** The file shall conform to the requirements of netCDF classic, as described in *The NetCDF Users' Guide*.[5]
   - **Geolocation conformant.** The file shall contain information allowing variables to be mapped to earth coordinates.
2. The "netCDF readable" and "netCDF classic conformant" modifications shall be reversible. That is, it shall be possible to revert all changes made to the product file that were made in order to achieve these states.
3. More than one JPSS product file may be specified to augment data files of the same product.
4. The tool shall be capable of operating on the following JPSS product files:

   - ♦ VIIRS
     - ▪ SVI01-05
     - ▪ SVM01-16
   - ♦ CrIS/ATMS
     - ▪ SCRIS
     - ▪ SATMS
     - ▪ TATMS

5. The tool shall be a command-line utility.
6. Options available on the command-line shall include the following:
   - Control file – reads the names of the input files from a separate file, rather than from the command line (Not implemented in version 1.0.0.)

5 http://www.unidata.ucar.edu/software/netcdf/docs/netcdf/.

- Help page – displays a help page
- Version number – displays a version number that corresponds to the version of the JPSS XML-to-HDF5 specification (see Section 6)
- Augmentation step – specifies augmentation step
- Restore – restores the link to the /Data_Products group or to the /All_Data group

7. The tool shall add dimension information to the geolocation datasets.[6]

4.2 Assumptions

The aforementioned requirements are based on the following assumptions.

1. Input files:
   1.1. For all modification requirements, at least one JPSS product file shall be supplied.
   Example: SVM07_aqu_d20101130_t1343575_e1345074_b45616_c20101201065150846446_grav_dev.h5
   1.2. For the "netCDF meaningful" requirement, a JPSS XML product file shall be supplied.
   Example: D34862-03_JPSS-CDFCB-X-Vol-III_D_VIIRS-M7-SDR-PP.xml
   1.3. For the "geolocation conformant" requirement, a JPSS geolocation product file shall be supplied with the JPSS product file.
   Example: GMODO_ter_d20101206_t2009584_e2011083_b00000_c20101206225316640547_grav_dev.h5
   1.4. The JPSS product files, the corresponding JPSS XML product file and the corresponding JPSS geolocation product file shall all be present in the directory in which the tool is executed, or else full paths shall be specified for JPSS product files and the corresponding JPSS XML product files. Geolocation files shall be located in the same directory as the JPSS product files.
2. The JPSS product file shall have the following structure:

```
Root Group "/"

        Group "All_Data"

                Group "<CollectionShortName>_all"

                        All datasets specified by <Field> tags in the corresponding JPSS
                        XML product file.  All instrument data is assumed to be in these datasets.

        Group "Data_Products"

                Any objects of reference type or attributes not supported by
                NetCDF-4 shall be in this group
```

[6] This feature was not implemented in the h5augjpss version 1.0.0 release but may be required in the future for tools to work properly on the augmented file.

3. It is acceptable that the “Data_Products” group and its contents not be accessible in an augmented file to either the netCDF or HDF5 libraries, and hence not accessible to applications that use the netCDF-4 and HDF5 libraries. The contents of the “Data_Products” group can include:
    3.1. Objects of type H5T_REFERENCE
    3.2. Multidimensional attributes
    3.3. SCALAR non-string attributes
4. It is acceptable that the following information be added to a JPSS product file:
    4.1. Attributes with names or addresses of objects hidden by h5augjpss
    4.2. Dimension scales (datasets, references and labels) and other attributes specified by the corresponding JPSS XML product file
5. Any geolocation information present in the JPSS product file is assumed to be correct and complete. “Height”, “Latitude” or “Longitude” datasets will not be added from the geolocation file specified if there is an existing dataset in the file.
6. In order to be made netCDF meaningful, the file shall also be made netCDF accessible.
7. In order to be made netCDF classic conformant, the file shall also be made netCDF meaningful.
8. In order to be made geolocation conformant, the file shall also be made netCDF meaningful.

5 Design

This section describes the design of the tools. It includes the list of command-line options to be provided, an overview and a detailed description of the operations performed by each step, and a list of non-functional requirements and design considerations.

5.1 Command-line options and public API

The h5augjpss tool will implement the user options described in Section 4.1, namely:

1. **Augmentation level** – specifies an augmentation corresponding to each of the augmentation steps
2. **Control file** – reads the names of the input files from a separate file, rather than from the command line (Not implemented in version 1.0.0.)
3. **Help page** – displays a help page
4. **Version number** – displays a version number that corresponds to the version of the JPSS XML-to-HDF5 specification (See Section 6.)

This functionality is to be implemented with a public API. (To be determined in another document.) For example, the tool will use public functions[7] to perform different augmentation steps or to restore the original group structure in the augmented file.

5.2 Overview of augmentation steps

The HDF5 augmentation tool will take two input files: a JPSS product file and a JPSS XML product file. The tool will provide options to make the following changes to the JPSS product file. Figure 9 shows the steps performed in a sequence (the tool's default behavior) to augment a file to be readable by the netCDF-based visualization tools. It needs to be emphasized that the augmentation <u>does</u> change the original product file as shown in

NetCDF readable. To make a file netCDF readable, delete the path to the "Data_Products" group in the HDF5 file hierarchy, making this group inaccessible to both netCDF-4 and HDF5 libraries. This step is not a prerequisite to performing the other steps, but it is necessary to allow netCDF4 to open the file; therefore, this step is necessary for any use of the file with the current netCDF-4 C library[8].

This option requires JPSS HDF5 product file as an input file. It creates a small number of new objects in the file and performs some changes to the file structure. The file can be easily restored to its original state from this modification. The process is shown

[7] 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.

[8] The 4.2.x netCDF-Java libraries do not have this restriction

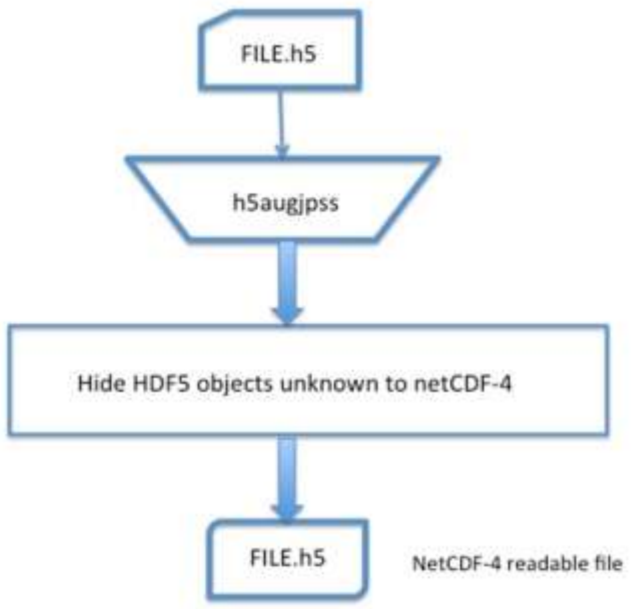


Figure 9: h5augjpss takes as input HDF5 product file and modifies it to become accessible by the netCDF-4 library

NetCDF meaningful. To make a file netCDF meaningful, the tool shall make the following changes to the product file.

1. Replace dimensions labeled "phony" in the JPSS project file with the dimension information from the XML product file as described in the JPSS XML-to-HDF5 specification, described in Section 6. Do this in such a way that ncdump will display these dimensions as coordinate variables.
2. Insert the metadata elements from the JPSS XML product file element JPSSDataProduct, described in Section 6.

This option adds new datasets and attributes. The file is augmented with many datasets and attributes. While file size does not increase much, restoration of the file to its original state becomes very complex. There is no provision for this restoration in the tool. The process is shown in Figure 10.

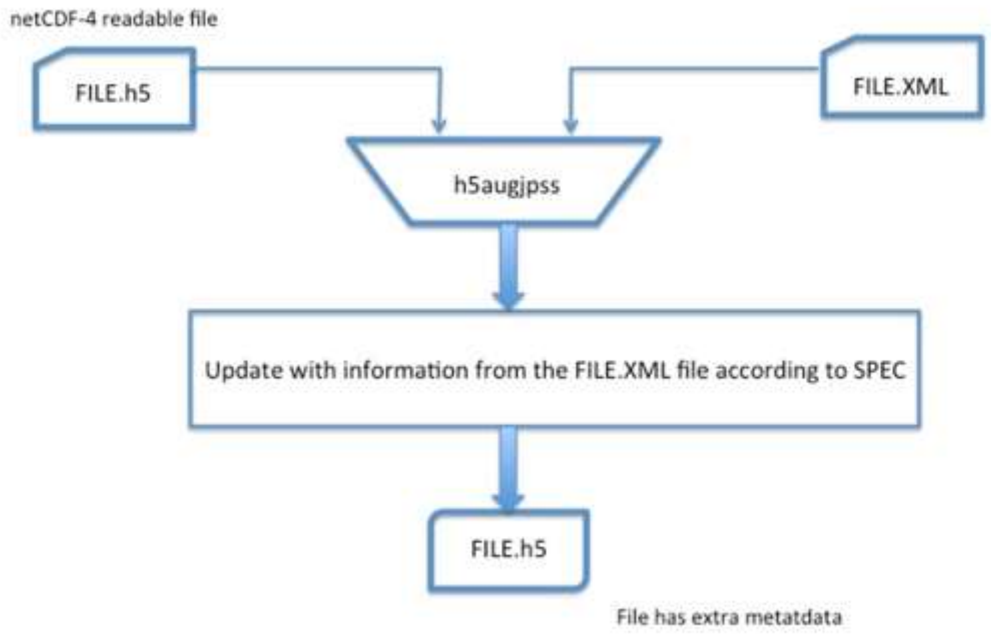


Figure 10: Adding metadata information from the XML file requires two input files - HDF5 product file and XML product file

NetCDF classic conformant. To make a file netCDF classic conformant, remove multi-level grouping so that file structure becomes flat. Since all the instrument data is contained in the datasets in one group (/All_Data/<CollectionShortName>_all), links can be created from the top level or '/' group to each of their datasets, making them all appear in the top level of the file. The /All_Data group structure can then be hidden. The netCDF-readable step hid the /Data_Products group, which has only attributes and references to the data in the datasets that will now be accessible at the top level. Hiding both groups leaves a "flat" file with no group structure. The process is shown on Figure 11.

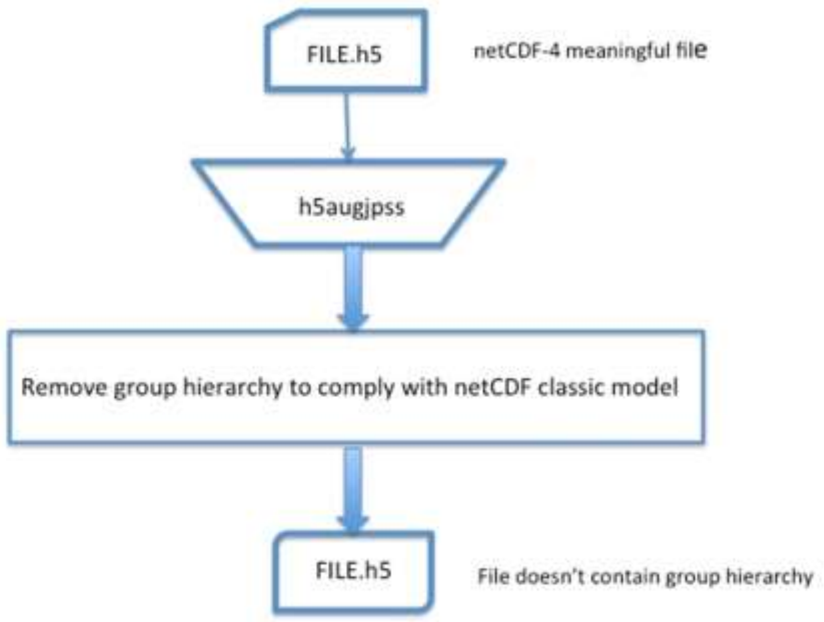

FILE.h5
netCDF-4 meaningful file
h5augjpss
Remove group hierarchy to comply with netCDF classic model
FILE.h5
File doesn't contain group hierarchy

Figure 11: h5augjpss removes group hierarchy from the netCDF-4 meaningful file

Geolocation conformant. To make a file geolocation conformant, copy “Height”, “Latitude” and “Longitude” datasets from the file specified in the JPSS product file’s N_GEO_Ref attribute. The process is shown on Figure 12.

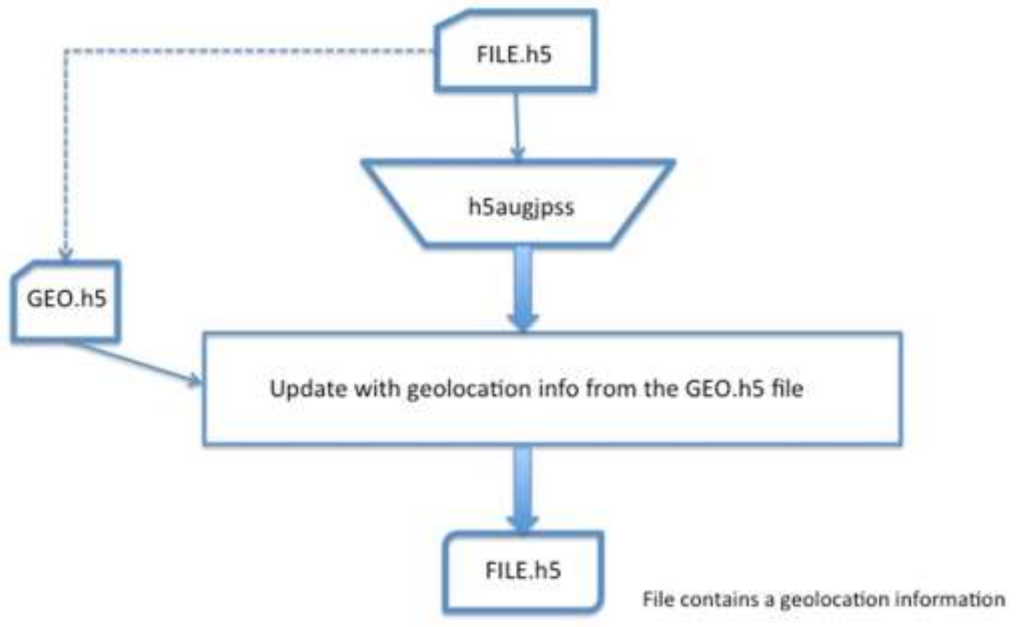
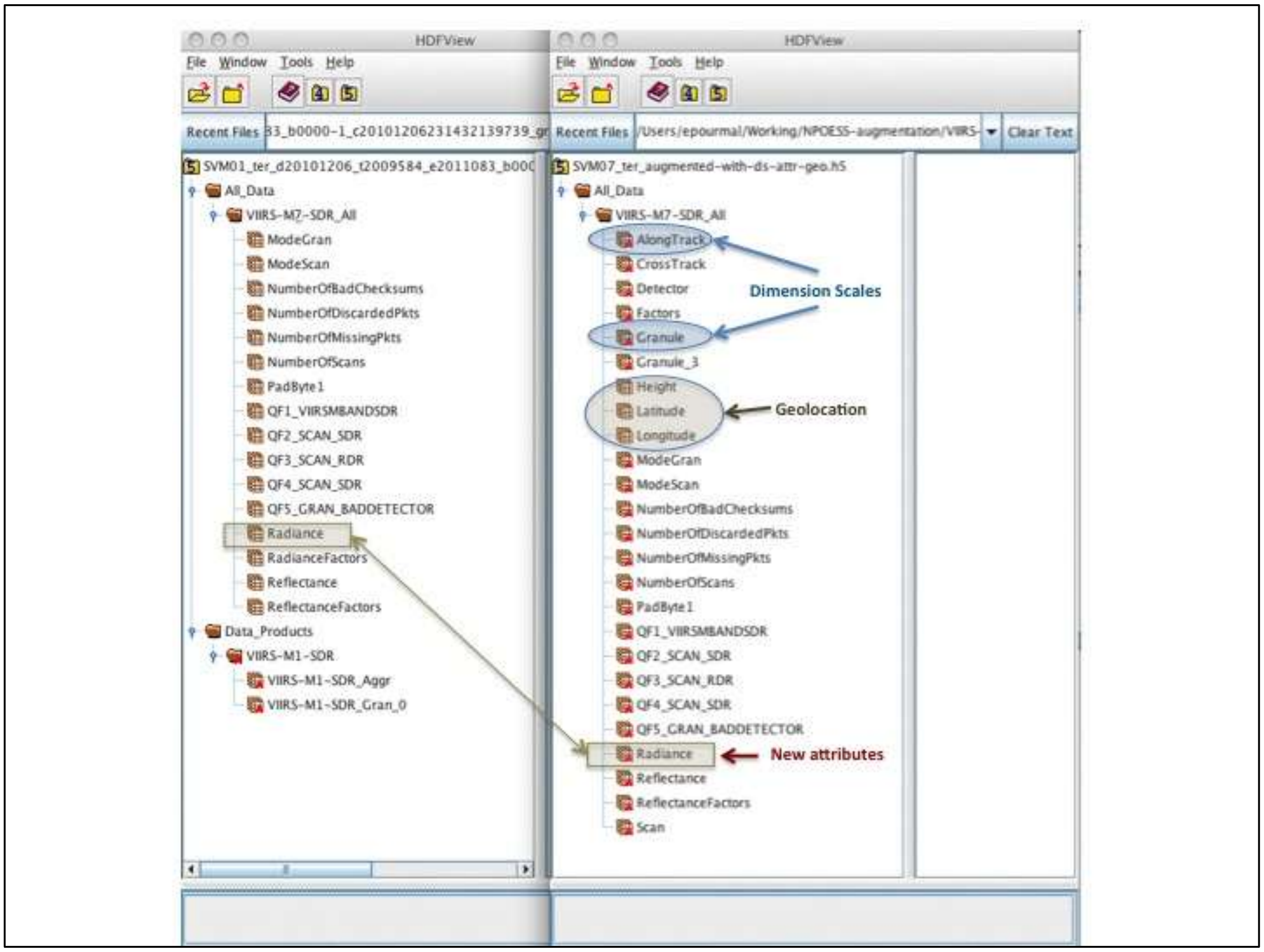

FILE.h5
h5augjpss
GEO.h5
Update with geolocation info from the GEO.h5 file
FILE.h5
File contains a geolocation information

Figure 12: h5augjpss adds geolocation information from the GEO.h5 file

The option adds several datasets with the geolocation information noticeably increasing the size of the original file. Figure 13 shows original tree structure of JPSS file on the left and the structure of the file after the augmentation step on the right.



```
SVM01_ter_d20101206_t2009584_e2011083_b000
  All_Data
    VIIRS-M7-SDR_All
      ModeGran
      ModeScan
      NumberOfBadChecksums
      NumberOfDiscardedPkts
      NumberOfMissingPkts
      NumberOfScans
      PadByte1
      QF1_VIIRSMBANDSDR
      QF2_SCAN_SDR
      QF3_SCAN_RDR
      QF4_SCAN_SDR
      QF5_GRAN_BADDETECTOR
      Radiance
      RadianceFactors
      Reflectance
      ReflectanceFactors
  Data_Products
    VIIRS-M1-SDR
      VIIRS-M1-SDR_Aggr
      VIIRS-M1-SDR_Gran_0
```

```
SVM07_ter_augmented-with-ds-attr-geo.h5
  All_Data
    VIIRS-M7-SDR_All
      AlongTrack
      CrossTrack
      Detector
      Factors
      Granule
      Granule_3
      Height
      Latitude
      Longitude
      ModeGran
      ModeScan
      NumberOfBadChecksums
      NumberOfDiscardedPkts
      NumberOfMissingPkts
      NumberOfScans
      PadByte1
      QF1_VIIRSMBANDSDR
      QF2_SCAN_SDR
      QF3_SCAN_RDR
      QF4_SCAN_SDR
      QF5_GRAN_BADDETECTOR
      Radiance
      Reflectance
      ReflectanceFactors
      Scan
```

Figure 13: File tree structure on original JPSS product file (on the left) and the augmented file (on the right) in HDFView

5.3 Augmentation steps in detail

5.3.1 NetCDF readable: hiding objects unknown to the netCDF-4 library

The purpose of this step is to make a JPSS file that the netCDF-4 library can read.

The current version of the netCDF-4 library[9] fails to open an HDF5 file that has groups or datasets with either two-dimensional attributes or datasets with HDF5 object or region references. Since most JPSS files have a "Data_Products" group containing one or more of these objects, the netCDF-4 library fails to open them. Currently, the only remedy to this problem is to modify every JPSS product file in

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

order to hide the “Data_Products” group. This modification is accomplished by deleting a path to the “Data_Products” group in the file hierarchy, making this group invisible to netCDF-4.

Information about the hidden group is stored as attributes in the Root group of the JPSS product file, according to the specification shown in Section 6. These attributes can be used to restore access to the group. The group’s internal reference count is also incremented by the tool in order to prevent the deletion of the group from the file when there are no links to it.

After the group is hidden, the netCDF-4 tool ncdump can read the file and display its content as shown in Section 2.2

Example: The following command hides the objects unknown to the netCDF-4 library in the *datafile.h5* file.

“h5augjpss -o1 datafile.h5”

5.3.2 NetCDF meaningful: updating with information from XML file

The purpose of this step is to copy meaningful dimension names and certain key metadata from the JPSS XML product file to the JPSS product file.

A JPSS XML product file contains information that is applicable to any corresponding HDF5 JPSS product file. This information includes the name of the product, the name of the collection the product belongs to, and a product identifier. It also has dimension information for the variables stored in the JPSS product file and other miscellaneous data, such as measurement units and fill values, as shown on Figure 14.

```
<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>
                            …
```

Figure 14: Examples of information (highlighted) from JPSS XML product file used to make HDF5 file netCDF meaningful.

The augmentation tool reads information from the XML file and writes it according to the specification described in Section 6. Since some information found in the XML file is already present in the original file, such as a name of a group, it may not be required to map all XML elements. This practice helps avoid extra file modification operations and avoids redundant data.

If this step is done after Step 1, netCDF-4 library finds necessary dimensional information that allows tools like ncdump to display the file content in a more meaningful way as shown on Figure 2.

We are seeking input from the user community on which information from the XML file should be saved in the HDF5 JPSS product file. Redundant information is noted in the corresponding subsections of Section 4.

Example: The following command adds information from the specified XML file; if *datafile.h5* was previously modified as in 5.3.1, ncdump displays meaningful information.

"h5augjpss -o2 productprofile.xml datafile.h5"

5.3.3 NetCDF classic conformant

The netCDF-4 library offers a CLASSIC_MODEL mode flag when creating a file. For a file created with this flag, the netCDF-4 library will limit features used to those compatible with NetCDF, creating an HDF5 file compatible with applications not using netCDF-4 enhanced model features. One of these features is the HDF5 group. The absence of groups is a necessary, but not sufficient, condition for making a file conform to the netCDF-4 Classic Model Format. The absence of groups will be sufficient as long as there are no other HDF5 features used that netCDF-4 cannot understand, which appears to be true for JPSS Data Product files.

Example: The following command removes groups from *datafile.h5*. If *datafile.h5* was previously modified using *-01* and *-02* flags as shown in 5.3.1 and in 5.3.2, then *datafile.h5* becomes netCDF classic conformant.

"h5augjpss -o4 datafile.h5"

5.3.4 Geolocation conformant: updating with geolocation information

The purpose of adding geolocation information to the JPSS product file is to enable netCDF-based applications to map variables to earth coordinates so that they can meaningfully visualize and analyze the data.

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

For non-packed JPSS data, geolocation information is stored in a separate file as shown on Figure 12. A JPSS data product file contains information about that file. 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 JPSS product file, where <XXX> is a value of the CollectionShortName XML element found in the XML file. For example, in the file shown on the left side of Figure 13, this information would be copied to the /All_Data/VIIRS-M7-SDR_All group shown on the right of the same figure.

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

As noted earlier, for many tools another extra step is required. For example, to visualize the dataset "Radiance" with IDV, one has to add attributes shown in Table 1. Also, all datasets have to be under the Root group.

Example: The following command adds geolocation information to the file.

"h5augjpss -o3 datafile.h5"

5.4 Non-functional requirements and design constraints

5.4.1 Exit codes and error handling

5.4.1.1 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.

5.4.1.2 Error handling

5.4.1.2.1 API failures

The tool shall exit if 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. (Not implemented in version 1.0.0.)

5.4.1.2.2 Verification failures

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

Match Failure	Result
CollectionShortName vs. /All_Data subgroup name	Exit
No Dataset to match <Field>	Warning – attributes not added
Size of type in dataset vs. <DataSize>/<Count>	Warning – attributes added
Dataset type vs. <Field>/<Datum>/<DataType>	(Not implemented in 1.0.0)
	Warning – attributes added

Radiance dataset types that do not match the type in the associated XML file have been noted where the dataset type is 32-bit float and the XML file specifies unsigned 16-bit integer. The RadianceFactors dataset has also been missing in these files. It is speculated that the scale factor and offset have already been applied.

5.4.2 Memory considerations

The tool will read and parse JPSS XML Product files in memory. Since the files are small, ~50K, the parsing step is not memory intensive. Adding the processing of 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.

5.4.3 Dependencies on third-party libraries

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

5.4.4 Operating systems

The tool shall work on the operating systems that are to be determined later. However, the minimum set includes Linux 32- and 64-bit, and AIX 32-bit platforms.

5.4.5 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 JPSS product files downloaded from the GRAVITE system and the netCDF-4.1 library (and later). The tool will also be tested manually with the netCDF-4 based visualization tools (to be determined).

5.4.6 Build system and packaging

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

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

5.4.7 Documentation

Documentation will include the help page, User's Guide and Reference Manual material.

5.4.8 License

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

6 Mapping Specifications Version 1.0

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

6.1 JPSS XML-to-HDF5 mapping considerations

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

- Choice of dataspace for HDF5 attributes

  NetCDF-4 does not support SCALAR dataspace for numeric attribute; therefore, SIMPLE dataspace is used even when the attribute is scalar. If in the future the netCDF-4 library lifts this restriction, mapping can be simplified by using SCALAR dataspace for all attributes with scalar values.

- Naming convention for HDF5 Dimension Scale Dataset

  XML elements mapped to the HDF5 dimensions 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 subsection 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 *HDF5 Dimension Scale Specification and Design Notes* [2] for discussions of dimension scale labels.

- Naming convention for HDF5 Attributes

XML elements mapped to the 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 6.2.5, 6.2.6, and 6.2.7 provide examples of the chosen naming conventions.

6.2 JPSS XML-to-HDF5 mapping specification

The following subsections describe a mapping between XML elements found in the JPSS Product Profile schema file and HDF5 objects.

To specify the mapping we used a table (Table 2) that should be read as follows:

- The first row contains a name of the element or complex type as present in the JPSS Product Profile schema file.
- The second row shows a corresponding excerpt from the JPSS 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. HDF5 object description follows notation adopted in [3].

Table 2: Mapping Table Template

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

Excerpts from the JPSS XML file D34862-03_JPSS-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 to illustrate the mapping. Highlighted text in a table and in the following examples will help to follow the mapping.

6.2.1 Mapping NPP/JPSS Data Product

The primary JPSS XML element “JPSSDataProduct” 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 shown in Table 3, Table 4, and Table 5, respectively.

The element “ProductData” is mapped as a complex type (see Table 6). The mapping is discussed in Section 6.2.2.

Table 3: Mapping the "ProductName" Element

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 4: Mapping the "CollectionShortName" Element

Element CollectionShortName		
`<xs:element name="CollectionShortName" type="xs:string" minOccurs="1" maxOccurs="1" />`		
XPath	**HDF5 Object**	
/CollectionShortName[10]	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”

[10] We should note here that if the element “CollectionShortName” is mapped, information becomes redundant. The original JPSS product file has a group with the same name as a value of that element under the /Data_Products group. For example, in Section 2.2 HDFView shows the VIIRS-M7-SDR group in the JPSS product file, and the value of the CollectionShortName element is “VIIRS-M7-SDR” as it is shown in the excerpt from the JPSS XML product file in Figure 14.

Table 5: Mapping the "DataProductID" 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 6: Mapping the "ProductData" Element

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

The two examples below show the instances of the "ProductName", "CollectionShortName" and "DataProductID" elements in the JPSS product XML file and their representation as HDF5 attributes in a Root group in the augmented file.

The following code is the XML example:

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

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

The following h5dump output shows instances of the elements “ProductName”, “CollectionShortName”, and “DataProductID” displayed as attributes in the Root group in the augmented file:

```
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"
         }

}
…….
```

6.2.2 Mapping Product Data Types

The JPSS complex type ProductDataType 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 JPSS product file, so the element is mapped but not written.

Table 7: Mapping of the "ProductDataType" Complex Type

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

The following XML example shows an instance of the “ProductData” element:

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

The following h5dump output shows an instance of the ProductData element shown as the HDF5 attribute “DataName” in the /All_Data/VIIRS-M7-SDR_All group and the dataset “Radiance":

```
….
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" {
….
```

6.2.3 Mapping Field Type

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

Table 8: Mapping of the "FieldType" Complex Type

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	

The following XML example depicts an instance of the “FieldType” complex type:

```
<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>
```

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, as represented in the following 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 )
      }

  }
       …..
```

6.2.4 Mapping Dimension Type

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

Table 9: Mapping of the "DimType" Complex Type

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>] $_{opt}$ 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" />
```

Continued on the next page

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		

Dimension with name AlongTrack is mapped to a dataset, as shown in the following XML example.

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

Dimension is represented by a dataset. Child elements of Dimension are represented by HDF5 attributes on the Dimension dataset, as shown by the following h5dumple output:

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

6.2.5 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 datasets that corresponds to the “Field” element. Some attributes are optional as noted in the table.

Table 10: The Mapping of the "DatumType" Complex Type

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	

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	

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		

The following is an XML example of “Datum” element attributes:

```
        <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>
```

“Datum” attributes are stored as attributes of a dataset. The following is the h5dump output:

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

6.2.6 Mapping FillValue Type

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

Table 11: Mapping of the "FillValueType" Complex Type

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]		

FillValue instances are shown in the following 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>
```

The following h5dump output shows the FillValue attributes:

```
 DATASET "Radiance" {
…..

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

6.2.7 Mapping LegendEntry Type

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

Table 12: Mapping of the "LegendEntryType" Complex Type

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]		

The following is an XML example of the instance of “LegendEntry”:

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

The repeated XML element “LegendEntry” is stored as an HDF5 attribute with a modified name. The h5dump output is:

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

Appendix: Specifications for Storing Information of a Hidden Group

Both attributes are given names that could be easily distinguished from other attributes stored in the JPSS product file (Table 13).

Table 13: Storing Information about Hidden Group in HDF5 File

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”)	

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 and Design Notes* http://www.hdfgroup.org/HDF5/doc/HL/H5DS_Spec.pdf
3. Mike Folk, Gerd Heber, Quincey Koziol. "HDF5 Information Set", to be published.
4. "IPO/JPSS Support (Formerly NPOESS)", H5edit tool (under development) http://www.hdfgroup.org/projects/npoess/
5. Kent Yang. "CF support for JPSS augmented files", ftp://ftp.hdfgroup.uiuc.edu/pub/outgoing/JPSS/RFC/CF-JPSS.pdf.