CF Support for JPSS-augmented Files

Kent Yang

Many users who are familiar with netCDF would benefit if netCDF-4 library and tools could access JPSS HDF5 files. One way to achieve this benefit is by augmenting the JPSS HDF5 files. Climate and Forecast Metadata conventions (CF conventions) are widely accepted by earth science user communities and are followed by several popular netCDF tools to display and analyze netCDF version 4 (netCDF-4) data. Additionally, JPSS XML files contain some metadata information defined by CF conventions. This document provides overviews of CF conventions, netCDF tools and JPSS file organizations. It reviews JPSS metadata related to CF conventions in detail and thoroughly discusses the possible mapping to CF attributes in the augmented file.

1 Introduction

JPSS data are useful to improve the understanding of global change and the accuracy of weather forecasts and climate predictions. NetCDF library and tools are widely used by some JPSS user communities, especially by weather and climate model communities. JPSS data use HDF5 as their storage format. Therefore, if netCDF library and tools could access JPSS files, many users familiar with netCDF would benefit. Because netCDF-4 is built on HDF5, it is possible to augment JPSS HDF5 files so that netCDF-4 APIs and netCDF-4 tools can access these files. The HDF Group is working on such a tool. CF conventions are designed to promote the processing and sharing of files created with the netCDF library. Some popular netCDF tools can only display and analyze netCDF files that have key CF attributes in their metadata descriptions. So it is crucial to figure out the CF requirements of the netCDF-4 files that have an organization similar to the augmented JPSS data. It is also crucial to figure out if JPSS data products can provide the information to feed in the required CF attributes. This document is a starting point for an ongoing effort to address these crucial elements.

This document provides overviews of CF conventions, some netCDF tools and JPSS file organizations, and includes a section that shows how CF attributes affect the visualization of netCDF files. JPSS metadata related to CF conventions are reviewed in detail. The requirement of CF attributes in the augmented file is thoroughly discussed. To facilitate the discussions, we also demonstrate CF requirements for JPSS VIIRS-like data with fake netCDF-4 classic files via a popular netCDF visualization tool, in Appendix A. The dump of a JPSS VIIRS SDR data XML file is listed in Appendix B. The dump of a JPSS VIIRS EDR geolocation XML file is listed in Appendix C. While investigating JPSS files, we also find some issues and have some questions that we hope the JPSS team can clarify or correct. The summary of these questions is listed in Appendix D.

2 Introduction to CF conventions

CF conventions [1] are metadata conventions for earth science data, intended to promote the processing and sharing of files created with netCDF APIs. However, most of its ideas relate to metadata design in general, not specifically to netCDF. CF metadata could be contained in other formats, such as XML. The conventions define metadata that are included in the same file as the data, that provide a definitive description of what the data in each variable represents, and of the spatial and temporal properties of the data. The definitive description of data enables users of data from different sources to decide which data are comparable and allows for the building of applications with powerful extraction, regridding, and display capabilities.

The CF conventions extend the COARDS conventions, which served a similar purpose before CF conventions were generated. COARDS conventions are efforts to provide conventions for the standardization of netCDF files from a program sponsored by the Cooperative Ocean/Atmosphere Research Data Service (COARDS), a NOAA/university cooperative for the sharing and distribution of global atmospheric and oceanographic research data sets. Since COARDS is still widely used by some netCDF tools, CF is backward-compatible with COARDS: applications which understand CF can also process COARDS datasets, and CF datasets will not break applications based on COARDS. The motivation for developing CF was the need for extra features, which include conventions for grid-cell boundaries, horizontal grids other than latitude–longitude, recording common statistical operations, standardized identification of physical quantities, non-spatiotemporal axes, climatological statistics and data compression.

Although CF was designed primarily to address gridded data types such as numerical model outputs and binned climatologies, it is applicable to many classes of observational data.

CF metadata use values of attributes to represent the following:

- Data provenance: title, institution, contact, source, history, references, comment
- Description of associated activity: project, experiment
- Description of data: units, standard_name, long_name, auxiliary_variables, missing_value, valid_range, flag_values, flag_meanings
- Description of coordinates: coordinates, bounds, grid_mapping (with formula_terms); time specified with reference_time ("time since T0") and calendar attributes
- Meaning of grid cells: cell_methods, cell_measures, and climatological statistics

“Description of data” and some part of “description of coordinates” are especially relevant to understanding the CF requirements of some popular netCDF tools in order to access the JPSS data. More information is listed in section 7.

3 Introduction to netCDF Tools

Generally, users can use some high-level languages such as IDL [2], MATLAB [3] and NCL [4] to access, display and analyze any netCDF data. However, some users get used to displaying and analyzing netCDF data via free visualization and analysis tools. Mostly these tools can only visualize a netCDF variable in the user-preferred way when the netCDF file follows CF conventions. If some attributes of a netCDF variable do not follow CF conventions, some tools cannot display the variable. In this section we briefly introduce several such tools and loosely divide the tools into two categories according to the usage of the tools by general users: Graphic User Interface (GUI) tools and Command-line tools. GUI tools are listed in section 3.1. Command-line tools are listed in section 3.2. Be aware that all the tools we list here can also support data in other data formats. These tools may work best with netCDF data. Detailed information about these tools can be found on their websites.

3.1 GUI Tools

In this section, we briefly introduce the Integrated Data Viewer (IDV) [5], Panoply [6] and MciDAS-V [7], which are the three most popular free netCDF GUI tools. These tools are built on the netCDF Java library [8], implemented by Unidata. These tools may be popular because they have the smallest learning curve for beginners and general users.

The IDV uses the VisAD [9] and other Java-based utility packages as well as the netCDF Java library. It allows users to interactively slice and probe data to create cross-sections, profiles, animations, data subsets, value readouts and aggregations of multi-dimensional data sets. Since the first release, around the year 2001, many powerful features have been added to the package.

Panoply is implemented by NASA Goddard Institute for Space Studies (GISS). It can slice and plot specific latitude-longitude, latitude-vertical, or time-latitude arrays from larger multidimensional variables. It supports more than 75 map projections.

McIDAS-V is the fifth generation of McIDAS (Man computer Interactive Data Access System) visualization and data analysis software package. It is implemented by the Space Science Engineering Center (SSEC) at the University of Wisconsin. It is actually built on VisAD and IDV and incorporates the functionality of McIDAS-X and HYperspectral-viewer for Development of Research Applications (HYDRA).

3.2 Command-line tools

Unlike GUI tools, many Command-line tools require users to input the commands and parameters using their keyboards in order to visualize and analyze the data. However, these tools may provide more powerful features for data analysis, so they may be flexible to generate various plots users define. The Grid Analysis and Display System (GrADS) [10] and Ferret [11] are two widely used C-based tools.

GrADS is an interactive desktop tool that is used for easy access, manipulation, and visualization of earth science data, especially gridded data. It is implemented by the Center for Ocean-Land-Atmosphere Studies (COLA).

Ferret is also an interactive computer visualization and analysis tool of earth science data. Implemented by Pacific Marine Environmental Laboratory (PMEL), it is specially designed to meet the needs of oceanographers and meteorologists analyzing large and complex gridded data sets.

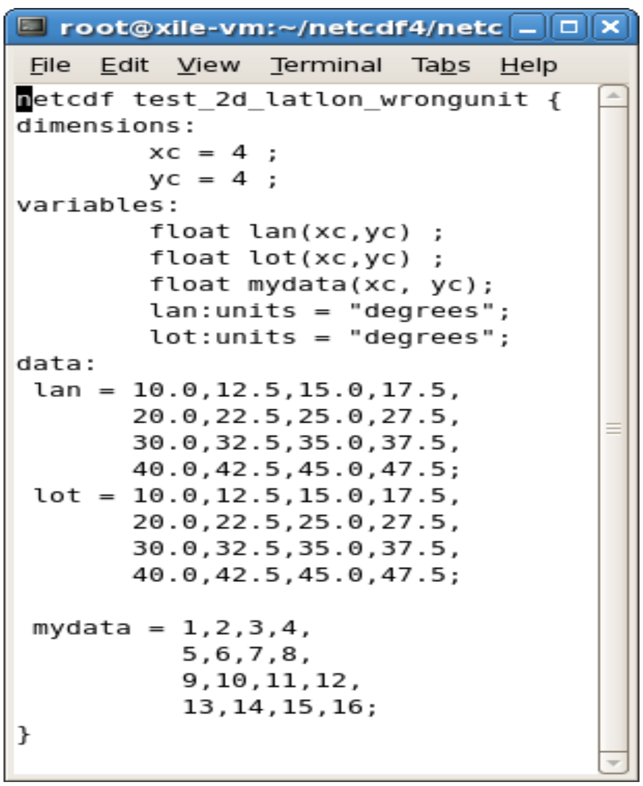
4 Access data via OPeNDAP

Open-source Project for a Network Data Access Protocal (OPeNDAP) [12] is a client-server software that supports remote access of data via Data Access Protocol (DAP). All the tools introduced in section 3 support the access of data via OPeNDAP. These tools can visualize and analyze the data via a netCDF OPeNDAP client library that receives the data from the DAP server. Generally these tools see the data accessed by OPeNDAP as netCDF data. For this reason, the CF requirements for data served by OPeNDAP are the same as those for netCDF data. To facilitate the discussion, this document uses the netCDF data to cover both the data in netCDF format and data via OPeNDAP.

5 Demonstration of the CF impact on visualizing netCDF data

To demonstrate how CF impacts the access of netCDF data via popular netCDF tools, we used IDV to display a variable of two fake similar netCDF-4 classic files. The main difference between these two files is that the attributes of variables describing the latitude and longitude of the first file do not follow CF conventions, but those of the second file do. These two files are similar to typical JPSS VIIRS files in terms of the organization of the variables.

Figure 1 shows the Common Data Language (CDL) output file of the first netCDF-4 classic file. The variable “lan” stores the latitude values of the variable “mydata”. The variable “lot” stores the longitude values of the variable “mydata”. Note that the “units” of both lan and lot are “degrees”, which don’t follow CF conventions. Figure 2 shows the output of using IDV to display this file. IDV cannot open the file and shows a “No Gridded Data” error message.




```
netcdf test_2d_latlon_wrongunit {
dimensions:
        xc = 4 ;
        yc = 4 ;
variables:
        float lan(xc,yc) ;
        float lot(xc,yc) ;
        float mydata(xc, yc);
        lan:units = "degrees";
        lot:units = "degrees";
data:
 lan = 10.0,12.5,15.0,17.5,
       20.0,22.5,25.0,27.5,
       30.0,32.5,35.0,37.5,
       40.0,42.5,45.0,47.5;
 lot = 10.0,12.5,15.0,17.5,
       20.0,22.5,25.0,27.5,
       30.0,32.5,35.0,37.5,
       40.0,42.5,45.0,47.5;

 mydata = 1,2,3,4,
          5,6,7,8,
          9,10,11,12,
          13,14,15,16;
}
```

Figure 1. CDL output of a netCDF-4 file that does not follow CF conventions

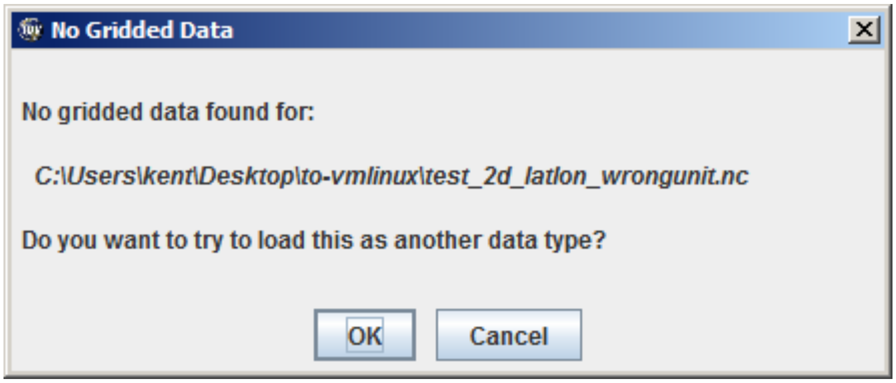
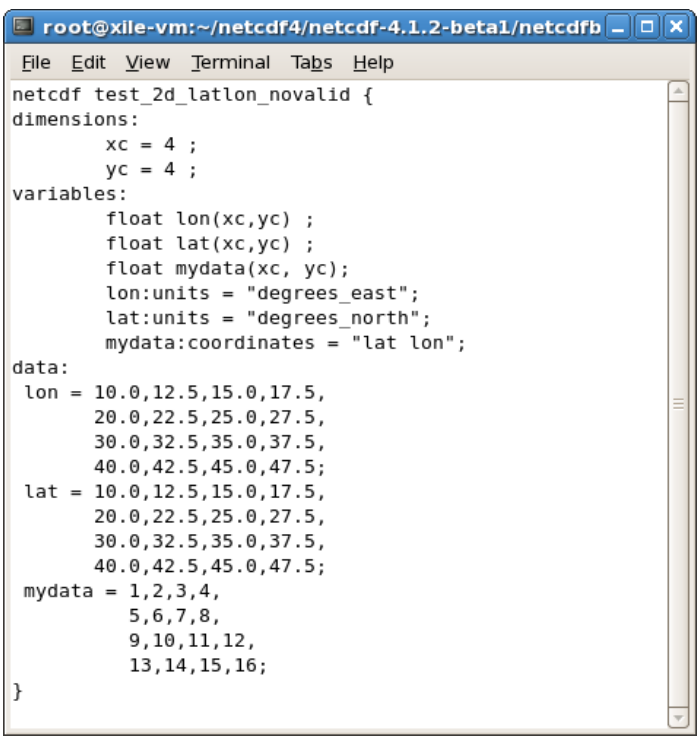


Figure 2. IDV output of a netCDF-4 file that does not follow CF conventions

Figure 3 shows the CDL output of the second netCDF-4 classic file. In this file, variable "lon" stores the longitude values of the variable "mydata". Variable "lat" stores the latitude values of the variable "mydata". Note that the attribute "units" of "lon" is "degrees_east" and the attribute "units" of "lat" is "degrees_north"; the attribute "coordinates" of "mydata" is "lat lon". The values of all these attributes follow CF conventions. We provide more information about CF attributes in section 7. Figure 4 displays the distribution of "mydata" in IDV's "value plots" option. Compared with the values of "mydata", "lat" and "lon", one can find that IDV correctly displays the distribution of the variable "mydata". From these two simple examples, one can see the importance of CF conventions regarding the visualization of netCDF data.




```
netcdf test_2d_latlon_novalid {
dimensions:
        xc = 4 ;
        yc = 4 ;
variables:
        float lon(xc,yc) ;
        float lat(xc,yc) ;
        float mydata(xc, yc);
        lon:units = "degrees_east";
        lat:units = "degrees_north";
        mydata:coordinates = "lat lon";
data:
 lon = 10.0,12.5,15.0,17.5,
       20.0,22.5,25.0,27.5,
       30.0,32.5,35.0,37.5,
       40.0,42.5,45.0,47.5;
 lat = 10.0,12.5,15.0,17.5,
       20.0,22.5,25.0,27.5,
       30.0,32.5,35.0,37.5,
       40.0,42.5,45.0,47.5;
 mydata = 1,2,3,4,
          5,6,7,8,
          9,10,11,12,
          13,14,15,16;
}
```

Figure 3. CDL output of a netCDF-4 file that follows CF conventions

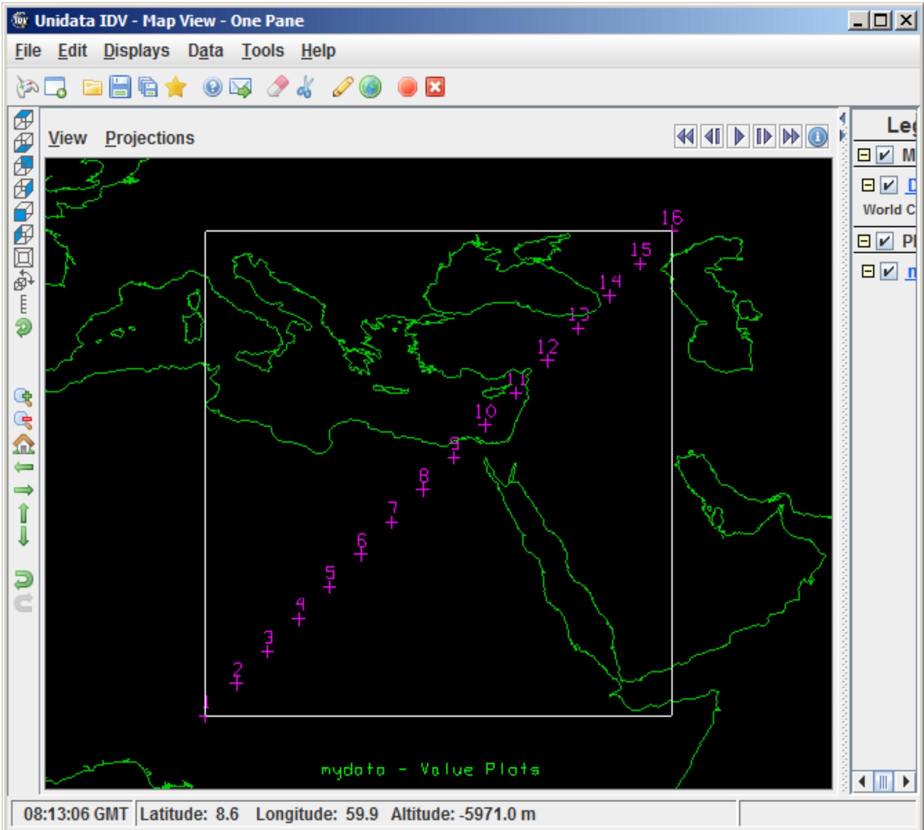


Figure 4. IDV output of a netCDF-4 file that follows CF conventions

6 Overview of JPSS VIIRS file organizations – our understanding

It is beyond the scope of this project to have a complete study of the JPSS HDF5 file structure since there are many JPSS data products and only limited information is available about these data products. The statements about the JPSS file organization in the following paragraph are based on our understanding of the "NPOESS Common Data Format Control Book – External Volume V – Metadata", the VIIRS SDR Data XML description file, the VIIRS EDR Geolocation XML description file, and several simulated JPSS HDF5 files from the NOAA's GRAVITE system. We omit some details of our observations of these files and focus only on the area of the file organization related to CF conventions.

To facilitate the further discussions, we want to clarify several terms. A JPSS field is equivalent to a CF variable. Hereafter in this document, the terms *field* and *variable* may be used interchangeably. For many cases, a piece of JPSS XML metadata information is also equivalent to a CF attribute. Sometimes in this document, we may also refer to a piece of XML metadata information as an attribute. Since it is time consuming and practically impossible to examine all kinds of JPSS files and documents, we made some assumptions based on our current best knowledge. We also have some questions and speculations that may need the JPSS team to clarify or correct. To clearly demonstrate these assumptions, questions and speculations, we use **bold,** *italicized* text (in the rest of the document) to highlight them. A summary of these assumptions, questions and speculations are listed in Appendix D.

Based on our current understanding, the organization of JPSS data is not trivial. The metadata information and data values are often spread in several files. One typical data organization is as follows:

- Raw data and some metadata stored in one HDF5 file
- Some key metadata information of the raw data stored in an XML file
- The geolocation data (latitude, longitude etc.) stored in another HDF5 file
- Some key metadata information of the geolocation data stored in another XML file

Although we also find other types of JPSS data organization, we only use the JPSS data organization listed above to facilitate this discussion.

To demonstrate the file organization of JPSS VIIRS data, we use one JPSS data HDF5 file and one JPSS geolocation HDF5 file, both obtained from NOAA's GRAVITE system; and a VIIRS SDR XML file and a VIIRS EDR GTM Geolocation XML file. Ideally we should find the corresponding JPSS geolocation HDF5 file of a JPSS data HDF5 file and then find the corresponding JPSS data XML file and the corresponding geolocation XML file. However, such files cannot be found when this document was written. According to Example A-4 of "NPOESS Data Product Profile XML Example" of "NPOESS Common Data Format Control Book – External Volume V – Metadata", ***it seems that different JPSS data products may share the similar XML profiles, even for different sensors***.

The JPSS geolocation HDF5 example file name we use for discussion is GMODO_aqu_d20101115_t1829460_e1830558_b45401_c20101115210512589018_grav_dev.he5

The JPSS data HDF5 example file name is SVM03_aqu_d20101119_t1223029_e1224128_b45455_c20101119155937553093_grav_dev.he5

We believe that the "aqu" inside the file name strings represents the AQUA satellite which generates MODIS products. Since the VIIRS data products aim to replace MODIS data products, ***we assume that the JPSS data HDF5 file and the JPSS geolocation HDF5 file obtained from NOAA's GRAVITE system are VIIRS products.*** So the information from a VIIRS SDR XML file and the JPSS VIIRS geolocation XML file can be used to understand the file structure of JPSS VIIRS files.

Inside the JPSS geolocation HDF5 example file, both latitude and longitude are 768 by 3200 two-dimensional arrays.

Inside the JPSS data HDF5 example file, two physical variables Radiance and Reflectance are also 768 by 3200 two-dimensional arrays.

This may indicate that the latitude and longitude of the corresponding physical variables are two-dimensional arrays. By checking the VIIRS SDR data XML file shown in Appendix B, we can verify that radiance and reflectance have two dimensions. The dimension names are AlongTrack and CrossTrack. The EDR geolocation XML file also shows that the latitude and longitude have two dimensions, the dimension names are also AlongTrack and CrossTrack.

"AlongTrack" and "CrossTrack" are like the typical dimension names used to describe MODIS swath data. MODIS latitude and longitude fields are also two-dimensional arrays of which the dimension names are like "AlongTrack" and "CrossTrack". ***So it seems that the representations of the latitude and longitude fields of VIIRS SDR data are like typical MODIS swath***. Currently the only way to make the netCDF GUI tools display MODIS swath is via CF conventions. To our best knowledge, it is difficult for the netCDF Command-line tools to display MODIS swath data. So in this document, we focus on the CF requirements of using GUI tools to display JPSS VIIRS data. The detailed CF requirements are listed in the next section. We also investigated the impact of CF attributes on using netCDF tools to access JPSS VIIRS-like netCDF-4 classic files. The results are summarized in Appendix A, which shows examples created by using IDV to access JPSS VIIRS-like netCDF-4 classic files with and without CF attributes. The impacts of CF attributes can be clearly observed from the CDL files and the corresponding screenshots.

7 CF requirements for JPSS VIIRS data

CF conventions version 1.5 [1] lists most CF attributes. In this document, we only list the CF attributes required or recommended for netCDF tools to access JPSS VIIRS data. Table 1 summarizes the information of CF required attribute “units” for several coordinate variables. Table 2 summarizes other required or recommended CF attributes. Table 3 summarizes other CF requirements.

Table 1: CF-required attribute "units" for several coordinate variables

Coordinate Type	Coordinate Variable	"units" Value	Note
Horizontal coordinate	Latitude	degrees_north	Also acceptable are degree_north, degree_N, degrees_N, degreeN, and degreesN
	Longitude	degrees_east	Also acceptable are degree_east, degree_E, degrees_E, degreeE, and degreesE
Vertical coordinate	Pressure	hPa	Also commonly used is bar, millibar, decibar, atmosphere (atm), pascal (Pa).
	Height or depth	Meter(m) and kilometer(km)	The units can be other values but can NOT be any "units" value of Pressure (hPa etc.). Another attribute called "Positive" must be used with "units" in this case. The "Positive" value is either "up" or "down". It is used to distinguish if the vertical coordinate is Height (up) or depth (down).

Table 2: Other required or recommended CF attributes

Attribute	Attribute Requirement	Description
coordinates	The current IDV release (version 2.9) and Panoply release (2.9.4) seem not to require this attribute. Since both tools currently only have limited support for variables that are three-dimensional (or higher) arrays, we expect future releases of these tools will enhance their support for multiple-dimensional arrays by using this attribute to identify the coordinate variables. We strongly recommend using this attribute. .	The "coordinates" attribute of a variable consists of a list of the associated coordinate variable names of the variable. For example, the associated coordinate variables of a variable "temperature" are "latitude", "longitude", and "pressure". The "coordinates" attribute of "temperature" should be stored as a string list: coordinates = "latitude longitude pressure"
_FillValue	Required if there are missing or undefined values	This attribute can only represent one missing or undefined value.
valid_min	Strongly recommended	Smallest valid value of a variable This attribute and attribute "valid_max" are required if all the following conditions apply: • There are multiple missing or undefined values or other special values • These values should be smaller than valid_min and larger than valid_max • "valid_range" is not presented
valid_max	Strongly recommended	Largest valid value of a variable This attribute and attribute "valid_min" are required if all the following conditions apply: • There are multiple missing or undefined values or other special values • These values should be smaller than valid_min and greater than valid_max

		• "valid_range" is not presented
valid_range	Strongly recommended	Smallest and largest valid values of a variable This attribute can replace valid_min and valid_max.
scale_factor	Strongly recommended	If present for a variable, the data are to be multiplied by this factor after the data are read by an application
add_offset	Strongly recommended	If present for a variable, this number is to be added to the data after it is read by an application. If both scale_factor and add_offset attributes are present, the data are first scaled before the offset is added.
long_name	Strongly recommended	A descriptive name that indicates a variable's content.
standard_name	Strongly recommended for latitude and longitude coordinates	For JPSS VIIRS data, some tools may need to use this attribute to identify that the variable is the coordinate variable latitude or longitude.

Table 3: Other CF requirements

Items	Requirements	Description
Special characters	Should not be used inside variables, dimensions and attribute names	Variable, dimension and attribute names should begin with a letter and be composed of letters, digits, and underscores. Other characters are treated as special characters.

8 Mapping JPSS metadata to CF

The JPSS XML files consist of geolocation XML files and data XML files. Not all the information inside the XML files is related to CF conventions. In this section, we focus only on the information that is related to CF conventions. Appendix B shows a VIIRS SDR data XML file. Appendix C shows a VIIRS EDR GTM geolocation XML file. The CF-related metadata in Appendices B and C include the following names: "Unit of Measure", "Scale Factor", "Offset", "Description" and "Fill Values" of all fields. These are the names we use in this document.

We are aware that the name "MeasurementUnits" is used in some XML files to represent the same content as the name "Unit of Measure". Similar situations may occur for other metadata, which may imply that some JPSS XML files use different names to represent the same metadata listed here.

We also discuss in this section the relation of the geolocation fields "Latitude", "Longitude", "Height" and "Time" to CF conventions, as well as dimensions. Although there are no CF attributes related to dimensions of a variable, we still discuss the impact of dimensions for tools to access netCDF data because the dimensions connect the coordinate variables described by the JPSS geolocation data and XML files with the physical variables described by the JPSS raw data and XML files.

We begin by examining the metadata information related to CF in the JPSS Product XML file and then list the CF-equivalent information. Since some CF-required information is either incomplete or unclear to us, we list this information in this section. We hope that the JPSS team can confirm, clarify, or correct our understandings. Finally, we present tables that summarize the possible mappings of CF-equivalent attributes to the augmented files and issues related to the mappings. Again, only VIIRS XML files are used for discussions.

8.1 CF-equivalent XML metadata

8.1.1 Unit of Measure

The metadata “Unit of Measure” is equivalent to the CF attribute “units”; therefore, the CF attribute “units” should be added to the augmented file. After a thorough examination, we found that the “Unit of Measure” of latitude and longitude described by the VIIRS EDR Geolocation XML file is different than that of other fields. We discuss the “Unit of Measure” of latitude and longitude separately.

8.1.1.1 VIIRS EDR Geolocation XML – Latitude and Longitude

The “Unit of Measure” of both latitude and longitude is “degrees”. CF conventions require that the units of latitude be “degrees_north” and the units of longitude be “degrees_east”. The value of units of latitude and longitude must be changed to follow CF conventions so that netCDF tools display the VIIRS JPSS data. Cases 1 and 2 in Appendix A verify this.

8.1.1.2 Other fields

To the best our knowledge, it appears that the contents of “Unit of Measure” of any other fields do follow CF conventions. Therefore, the contents of “Unit of Measure” of other fields can be the value of the CF attribute “units”.

8.1.2 Description

We suggest mapping the metadata “Description” for a JPSS variable to the CF attribute “long_name”. Since the CF attribute long_name is a descriptive name that indicates a variable’s content. So it is equivalent to “Description” of the variable name in the JPSS XML file.

8.1.3 Fill Values

According to the VIIRS SDR Data XML file shown in Appendix B and the VIIRS EDR Geolocation XML file shown in Appendix C, several JPSS data fields and geolocation fields have multiple fill values. In fact, we have not found any JPSS field that has a single fill value. Currently, the only way to handle multiple fill values in CF conventions is via the following CF attributes: valid_min, valid_max or valid_range. (Cases 6 and 7 in Appendix A verify this.) Be aware that only a certain type of multiple fill value cases can be handled via these CF attributes. In examining the fill values of JPSS variables, we found that more information is required to determine if CF attributes valid_min, valid_max or valid_range can be used to handle multiple fill values of JPSS variables. We need the JPSS team to provide more information for us.

8.1.3.1 VIIRS SDR Data XML – Radiance and Reflectance

Table 4 shows the section of table "VIIRS I-Band SDR Data Product Profile" in Appendix B that describes the metadata of a VIIRS SDR data field "Radiance". Examine the "Fill Values" column under the Datum heading in Table 4 to see multiple fill values named NA_UINT16_FILL, MISS_UINT16_FILL, ONBOARD_PT_UINT16_FILL, ONGROUND_PT_UINT16_FILL, ERR_UINT16_FILL, VDNE_UINT16_FILL, SOUB_UINT16_FILL. Their values are 65535, 65534, 65533, 65532, 65531, 65529, 65528, respectively. The data type is unsigned 16-bit integer. Since the maximum value of unsigned 16-bit integer is 65535, every valid data value should be smaller than NA_UINT16_FILL, MISS_UINT16_FILL, ONBOARD_PT_UINT16_FILL, ONGROUND_PT_UINT16_FILL, ERR_UINT16_FILL. However, the data value can be 65530, which is greater than VDNE_UINT16_FILL, SOUB_UINT16_FILL. We need to know whether 65530 can be a valid data value. If a valid data value cannot be 65530, we can add two CF attributes valid_min and valid_max to make netCDF tools not display the multiple fill values, as shown in Case 7 of Appendix A. The value of valid_min should be 0 and the value of valid_max should be 65527. If 65530 can be a valid data value, to the best of our knowledge, there is currently no CF support to handle this case. This will cause the tools to treat special values as normal data values and, thus, cannot display the distribution of data properly. ***We need the JPSS team to clarify if 65530 can be used as a valid data value or if it is reserved for variable Radiance.***

The fill values of the variable "Reflectance" are exactly the same as those for "Radiance" and should be handled the same way

Table 4: Metadata of a VIIRS SDR data field "Radiance"

Fields		
Name	**Data Size**	**Description**
Radiance	2byte(s)	(see Dimension(s) and Datum below)

Dimension(s)

Dimension Name	Granule Bounding	Dynamic	Max Index	Min Index
AlongTrack	Yes	No	1536	1536
CrossTrack	No	No	6400	6400

Datum

Description	Offset	Scale Factor	Unit of Measure	Range Min	Range Max	Data Type	Fill Values
Calibrated Top of Atmosphere (TOA) Radiance for each VIIRS pixel	0	Scaled using RadianceFactors	W/(m^2 sr µm)			unsigned 16-bit integer	(see Fill Values below)

Fill Values

Name	Value
NA_UINT16_FILL	65535
MISS_UINT16_FI L	65534
ONBOARD_PT_UINT16_FILL	65533

									ONGROUND_PT_UINT16_FILL	65532
									ERR_UINT16_FILL	65531
									VDNE_UINT16_FILL	65529
									SOUB_UINT16_FILL	65528

8.1.3.2 VIIRS SDR Data XML – ModeGran and ModeScan

Variables ModeGran and ModeScan shown in Appendix B have the following multiple fill values: 254,251 and 249. The data type of ModeGran and ModeScan is unsigned 8-bit char. The maximum possible value is 255. So to use CF attributes valid_min and valid_max, we need to know whether 253,252 and 250 can be valid data values. If 253,252 and 250 are valid data values, there is currently no CF support to handle this case. This lack of support will cause the tools to treat 253,252 and 250 as normal data values and, thus, cannot display the distribution of data properly. ***Again we need the JPSS team to clarify whether 253,252 and 250 can be valid data values for ModeGran and ModeScan.*** If 253,252 and 250 are valid data values, we can assign valid_min to be 0 and valid_max to be 248. It seems that ModeGran and ModeScan are not physical variables measured by JPSS satellites. So it may not be necessary to display the distribution of these variables. If this is the case, it may not be important to make these variables follow CF conventions.

8.1.3.3 VIIRS SDR Data XML – NumberOfMissingPkts, NumberOfDiscardedPkts and NumberOfBadChecksums

Variables NumberOfMissingPkts, NumberOfDiscardedPkts and NumberOfBadChecksums shown in Appendixes B also have multiple fill values. Their fill values are -998 and -993. The data type of these variables is 32-bit integer. According to descriptions of these variables***, it seems that the valid data values of these variables may always be no less than 0.*** If this is true, it is not necessary to provide any CF attributes to handle fill values.

8.1.3.4 VIIRS EDR Geolocation XML – Latitude and Longitude

Based on our experience with netCDF tools, we strongly recommend that there are no fill values inside latitude and longitude because some tools still follow COARDS conventions, which require that the latitude and the longitude do not contain fill values. However, we still find multiple fill values of latitude and longitude described inside the JPSS XML file.

Table 5 shows the section from the table “VIIRS NCC-Band Imagery GTM Geolocation Product Profile" in Appendix C that describes the metadata of field names “Latitude” and “Longitude”. Given the Datum information for Latitude (Unit of Measure: “degree”, Description: “Latitude of each pixel (positive North)”, Scale Factor: “unscaled”, and Offset: “0”) and the Datum for Longitude (Unit of Measure: “degree”, Description: “Longitude of each pixel (positive East)”, Scale Factor: “unscaled”, and Offset: “0”), we can firmly assert that the latitude value of VIIRS JPSS data is from -90 to 90 and the longitude value of VIIRS JPSS data is from -180 to 180. Since multiple fill values of latitude and longitude are -999.9, -999.8, -999.5, -999.4 and -999.3, which are all outside the valid range of latitude and longitude, we strongly suggest that the following CF attributes be provided inside the augmented file:

- Latitude: valid_min= -90.0 valid_max = 90.0
- Longitude: valid_min = -180.0 valid_max = 180.0

Table 5: Metadata of latitude and longitude described by the VIIRS EDR Geolocation XML file

<table>
<tr><th colspan="3">Fields</th></tr>
<tr><th>Name</th><th>Data Size</th><th>Description</th></tr>
<tr><td>Latitude</td><td>4byte(s)</td><td>
<table>
<tr><th colspan="5">Dimension(s)</th></tr>
<tr><th>Dimension Name</th><th>Granule Bounding</th><th>Dynamic</th><th>Max Index</th><th>Min Index</th></tr>
<tr><td>AlongTrack</td><td>Yes</td><td>No</td><td>771</td><td>771</td></tr>
<tr><td>CrossTrack</td><td>No</td><td>No</td><td>4121</td><td>4121</td></tr>
</table>
<table>
<tr><th colspan="8">Datum</th></tr>
<tr><th>Description</th><th>Offset</th><th>Scale Factor</th><th>Unit of Measure</th><th>Range Min</th><th>Range Max</th><th>Data Type</th><th>Fill Values</th></tr>
<tr><td>Latitude of each pixel (positive North)</td><td>0</td><td>Unscaled</td><td>degree</td><td></td><td></td><td>32-bit floating point</td><td>
<table>
<tr><th>Name</th><th>Value</th></tr>
<tr><td>NA_FLOAT32_FILL</td><td>-999.9</td></tr>
<tr><td>MISS_FLOAT32_FILL</td><td>-999.8</td></tr>
<tr><td>ERR_FLOAT32_FILL</td><td>-999.5</td></tr>
<tr><td>ELINT_FLOAT32_FILL</td><td>-999.4</td></tr>
<tr><td>VDNE_FLOAT32_FILL</td><td>-999.3</td></tr>
</table>
</td></tr>
</table>
</td></tr>
<tr><td>Longitude</td><td>4byte(s)</td><td>
<table>
<tr><th colspan="5">Dimension(s)</th></tr>
<tr><th>Dimension Name</th><th>Granule Bounding</th><th>Dynamic</th><th>Max Index</th><th>Min Index</th></tr>
<tr><td>AlongTrack</td><td>Yes</td><td>No</td><td>771</td><td>771</td></tr>
<tr><td>CrossTrack</td><td>No</td><td>No</td><td>4121</td><td>4121</td></tr>
</table>
<table>
<tr><th colspan="8">Datum</th></tr>
<tr><th>Description</th><th>Offset</th><th>Scale Factor</th><th>Unit of Measure</th><th>Range Min</th><th>Range Max</th><th>Data Type</th><th>Fill Values</th></tr>
</table>
</td></tr>
</table>

		Longitude of each pixel (positive East)	0	**Unscaled**	degree			32-bit floating point

Name	Value
NA_FLOAT32_FILL	-999.9
MISS_FLOAT32_FILL	-999.8
ERR_FLOAT32_FILL	-999.5
ELINT_FLOAT32_FILL	-999.4
VDNE_FLOAT32_FILL	-999.3

8.1.3.5 VIIRS EDR Geolocation XML – Time

Table 6 shows the section in the table "VIIRS NCC-Band Imagery GTM Geolocation Product Profile" in Appendix C that describes the metadata of "Time". "Time" has multiple fill values, but it also has two other attributes, "Range Min" and "Range Max". According to the meaning of "Range Min" and "Range Max", ***"Range Min" is possibly equivalent to CF attribute "valid_min", and "Range Max" is possibly equivalent to CF attribute "valid_max".*** We can also verify that multiple fill values are outside the valid range of the data values between valid_min and valid_max. So we can safely map "Range Min" to the CF attribute "valid_min" and "Range Max" to "valid_max".

Time: valid_min = 1483228832000000, valid_max = 2272147232000000

Table 6: Metadata of Field "Time" described by the VIIRS EDR Geolocation XML file

Fields		
Name	**Data Size**	**Description**
Time	8byte(s)	

Dimension(s)

Dimension Name	**Granule Bounding**	**Dynamic**	**Max Index**	**Min Index**
AlongTrack	Yes	No	771	771

Datum

Description	**Offset**	**Scale Factor**	**Unit of Measure**	**Range Min**	**Range Max**	**Data Type**	**Fill Values**
Time of the	0	**Unscaled**	microsecond	1483228832000000	2272147232000000	64-bit	

Name	**Val**

		nadir point of the GTM row in IET (1/1/1958). Represents the time of the nadir point of the GTM row						integer	<table><tr><th></th><th>ue</th></tr><tr><td>NA_INT64_FILL</td><td>-999</td></tr><tr><td>MISS_INT64_FILL</td><td>-998</td></tr><tr><td>ERR_INT64_FILL</td><td>-995</td></tr><tr><td>VDNE_INT64_FILL</td><td>-993</td></tr></table>

8.1.4 Scale Factor and Offset

By examining the VIIRS SDR data XML file shown in Appendix B and the VIIRS EDR geolocation XML file shown in Appendix C, we find that metadata “Offset” and “Scale Factor” exist for all variables. From the two XML files, we find that there are only two different cases regarding the “Scale Factor” and “Offset” values.

8.1.4.1 VIIRS SDR Data XML – Radiance and Reflectance

According to the variable “Radiance” shown in Table 4, the value in the “Scale Factor” column is “Scaled using Radiance Factors”. In the “Description” column in Table 7, the variable RadianceFactors is described as “Scale = first array element; Offset = second array element”. So the value of “Scale Factor” is the first array element of RadianceFactors, and the value of “Offset” is the second array element of RadianceFactors.

Be aware that in the “Offset” column (for the metadata of the variable “Radiance” shown in Table 4) the value is 0. We hope the JPSS team will clarify which value should be used for the Offset: 0 or the second array element of RadianceFactors. It is very possible that the second array element of RadianceFactors is always 0. However, we still need confirmation. ***We also need the JPSS team to clarify if the equation that calculates the final data value by applying “Scale Factor” and “Offset” is***

Final_data_value = “Scale Factor” * Raw_data_value + “Offset”;

If the above equation is the equation that JPSS data uses, the following CF attributes should be provided to the variable “Radiance” in the augmented file:

scale_factor = first array element of RadianceFactors,

add_offset = second array element of RadianceFactors or 0;

The HDF Group

If the above equation is not the equation the JPSS data uses, it is required that the equivalent attribute name "Scale Factor" should not be scale_factor and the equivalent attribute name "Offset" should not be add_offset. In this way, tools will not treat those attributes as CF scale_factor and add_offset attributes and will not miscalculate the final data values by following the wrong equation. The impact of CF attributes "scale_factor" and "add_offset" on using netCDF tools to display netCDF data can be observed from cases 8 and 9 in Appendix A. The way JPSS handles "Scale Factor" and "Offset" for the variable "Reflectance" is exactly the same as "Radiance".

In summary, we hope the JPSS team will clarify whether the equation that is applied to scale and offset for "Radiance" and "Reflectance" is the same as that defined by CF conventions. Based on the team's answer we might be able to create the corresponding scale and offset attributes.

Table 7: Metadata of a VIIRS SDR data field "RadianceFactors"

<table>
<tr><th colspan="3">Fields</th></tr>
<tr><th>Name</th><th>Data Size</th><th>Description</th></tr>
<tr><td>RadianceFactors</td><td>4byte(s)</td><td>
<table>
<tr><th colspan="5">Dimension(s)</th></tr>
<tr><th>Dimension Name</th><th>Granule Bounding</th><th>Dynamic</th><th>Max Index</th><th>Min Index</th></tr>
<tr><td>Factors</td><td>Yes</td><td>No</td><td>2</td><td>2</td></tr>
</table>
<table>
<tr><th colspan="7">Datum</th></tr>
<tr><th>Description</th><th>Offset</th><th>Scale Factor</th><th>Unit of Measure</th><th>Range Min</th><th>Range Max</th><th>Data Type</th></tr>
<tr><td>Scale = first array element; Offset = second array element</td><td>0</td><td>Unscaled</td><td>scale = unitless; offset = W/(m^2 µm sr)</td><td></td><td></td><td>32-bit floating point</td></tr>
</table>
</td></tr>
</table>

8.1.4.2 Other fields in VIIRS SDR Data XML and fields in VIIRS EDR Geolocation XML

For all other fields except "Radiance" and "Reflectance" shown in Appendix B and Appendix C, the value of "Scale Factor" is Unscaled and "Offset" is 0. To make "Scale Factor" and "Offset" follow CF conventions, we can do the following:

1) Method 1: Do not provide the "Scale Factor" and "Offset" attributes at all.

2) Method 2: Provide the equivalent CF attributes of "Scale Factor" and "Offset". Their names are "scale_factor" and "add_offset". Assign the value of scale_factor to be 1 and the value of add_offset to be 0.

8.2 Other discussions related to CF

8.2.1 VIIRS SDR Fields

According to Chapter 5 of CF conventions version 1.5[1], "all of a variable's spatiotemporal dimensions that are not latitude, longitude, vertical, or time dimensions are required to be associated with the relevant latitude, longitude, vertical, or time coordinates via the new 'coordinates' attribute of the variable. The value of the 'coordinates' attribute is a blank separated list of the names of auxiliary coordinate variables. There is no restriction on the order in which the auxiliary coordinate variables appear in the coordinates attribute string. The dimensions of an auxiliary coordinate variable must be a subset of the dimensions of the variable with which the coordinate is associated…. We recommend that the name of a multidimensional coordinate variable should not match the name of any of its dimensions because that precludes supplying a coordinate variable for the dimension."

As discussed briefly in section 6 and shown in Appendix B and C, the JPSS dimension names for both the variable "Radiance" and the variable "Reflectance" are "AlongTrack" and "CrossTrack". The JPSS coordinate variables for both the variable "Radiance" and the variable "Reflectance" are "Latitude" and "Longitude". So the variables "Radiance" and "Reflectance" should have the "coordinates" attribute. The value of the "coordinates" attribute should be "Latitude Longitude". Although we need more JPSS XML and data files to verify, we believe that the 'coordinates' attribute should be provided to every variable of the JPSS data that has spatiotemporal dimensions. It is beyond the scope of this document to discuss the possible values of every "coordinates" attribute. When more JPSS data is available, a separate note can be dedicated to this topic.

8.2.2 VIIRS EDR Geolocation Fields

8.2.2.1 Latitude and Longitude

As discussed in section 8.1.1.1, the units of latitude should be "degrees_north" and the units of longitude should be "degrees_east", as required by CF conventions. Although IDV and Panoply can figure out the latitude and longitude variables by only checking their units, other tools may also need to check the value of another CF attribute, standard_name, to identify whether the variable stores latitude or longitude values. The value of standard_name must be "latitude" for the variable that stores the latitude regardless of the name of the variable. The value of standard_name must be "longitude" for the variable that stores the longitude regardless of the name of the variable. Both units and standard_name should be provided in the augmented file:

Latitude: units = "degrees_north"; standard_name = "latitude;

Longitude: units = "degrees_east"; standard_name = "longitude";

8.2.2.2 Time

The CF conventions also list “time” as a coordinate variable. The unit of “time” is described in section 4.4 of Chapter 4 of the CF conventions version 1.5[1]. The usage of “time” as a coordinate can be found at Example 5.1 of the CF conventions version 1.5. In that example, a physical four-dimensional array variable xwind(time, pres, lat,lon) describes the xwind with latitude, longitude, vertical and time. The coordinate variable time is orthogonal to the coordinate variable latitude and longitude, which means that the dimension of “time” is different than the dimension of “latitude” and “longitude”. This case is the most common one that netCDF Java visualization tools can currently handle. However, according to Appendix C, the dimension name of the JPSS “Time” variable is “AlongTrack”, which is the same as one of the dimensions of “latitude” and “longitude”. ***This implies that perhaps the JPSS geolocation variable “Time” may not be the same “time” listed at CF conventions. According to the description of Time, the record is the Time of the nadir point of the GTM row in IET (1/1/1958), and it represents the time of the nadir point of the GTM row. It seems that it records the time at a certain point when the satellite moves along the satellite track. Therefore, although this variable is very important for researchers to identify the exact time the physical variables are measured, we believe that it is different than the coordinate variable “Time” listed at the CF conventions. We suggest considering the JPSS Time variable as a general variable. We need the JPSS team to verify or to correct our understandings.***

8.2.2.3 Height

Like the “Time” variable, the “Height” variable is also listed as a coordinate variable by the CF conventions. Like the JPSS “Time” variable, the JPSS “Height” variable should be considered as a general variable rather than a coordinate variable, as defined by CF Conventions. The reason is as follows: according to Appendix C, “Height” is a two-dimensional array with the dimension names “AlongTrack” and “CrossTrack”. “AlongTrack” and “CrossTrack” are also dimension names of Latitude and Longitude. According to CF conventions version 1.5, the two-dimensional array “Height” coordinate variable is not supported. The “Height” coordinate variable in CF conventions still follows COARDS conventions, which is a one-dimensional array. Therefore, we suggest considering the “Height” variable just as a general variable regarding CF conventions.

8.2.2.4 Others

We suggest considering all other geolocation fields as general variables in terms of CF conventions since we cannot find the equivalent coordinate variables defined by CF conventions. We need the JPSS team to confirm this.

8.2.3 Special Characters inside Field names, Dimension names and Attribute names

According to CF-naming conventions, variable, dimension and attribute names should begin with a letter and be composed of letters, digits, and underscores. Characters that are not letters, digits and underscores are special characters. NetCDF files may not even be generated if variable, attribute or dimension names contain special characters (Case 10 in Appendix A). ***It is very possible that all variable, dimension and attribute names in JPSS files follow CF-naming conventions.*** Here we only demonstrate one possible way to make JPSS data follow CF-naming conventions in case some JPSS products violate the CF-naming conventions. The real implementation can be different.

We suggest changing any special character to an underscore for the variable, dimension and attribute names containing special characters.

To preserve the original variable names, add a new attribute "original_name" and save the original name value to this attribute***. We hope to get feedback from the JPSS team on whether to preserve the original dimension names and attribute names if some dimension or attribute names have special characters.***

If the decision is made to preserve the original dimension names and attribute names, there are many ways to do so. In this document, we list one possible solution. The real implementation can be different.

To handle the dimension names, for example, if the original dimension name is "Fake Dim", follow CF-naming conventions by changing it to "Fake_Dim". An attribute called "Original_dim_name_Fake_Dim" can be added and its value should be "Fake Dim". To handle the attribute names, if the original attribute name is "Scale Factor", follow CF-naming conventions by changing it to "Scale_Factor". An attribute called "Original_attr_name_Scale_Factor" should be added and its value should be "Scale Factor".

8.2.4 Dimensions

There are no CF attributes related to the variable dimension. However, netCDF requires that each dimension provide the dimension name and the dimension length. If users do not provide a dimension name, the netCDF library will assign a fake dimension name. Furthermore, for JPSS files, dimension name and length pair may be used to connect the coordinate variables with the physical variables. Therefore, we discuss JPSS variable dimensions in this subsection.

We use the VIIRS SDR data XML file shown in Appendix B and the VIIRS EDR GTM Geolocation XML file shown in Appendix C to demonstrate why dimension name and length are important. Since we cannot find the corresponding geolocation XML profile of the SDR data, the dimension name and length of the dimension are not consistent. When examining the dimensionality of JPSS data and the corresponding geolocation files, the dimension length of the data variable and the geolocation variable are consistent. ***So we can fairly assume that the dimension names, lengths of the data XML file and the corresponding geolocation XML file of the same JPSS product are also consistent.***

Under these assumptions, let us evaluate the dimensions of the JPSS data XML file and the JPSS geolocation XML file. Shown in both Appendix B and C, under the table "Table of Granule Dimensions", we find dimensions that have the names "AlongTrack" and "CrossTrack". From the table "VIIRS I-Band SDR Data Product Profile" at the SDR data XML profile file shown in Appendix B, the dimension names of the variable "Radiance" are "AlongTrack" and "CrossTrack". From the table "VIIRS NCC-Band Imagery GTM Geolocation Product Profile" shown in Appendix C, we also find that the dimension names of coordinate variable "Latitude" and "Longitude" are "AlongTrack" and "CrossTrack". The dimension lengths are not consistent but, as we described, we can fairly assume that the dimension length of the same dimension name of the JPSS data XML file and the JPSS geolocation XML file of the same JPSS data product is consistent. Then by checking the dimension name and length of the physical variable and the corresponding coordinate variables, we can connect the variable "Radiance" with coordinate variables "Latitude" and "Longitude".

We still need the JPSS team to clarify whether "Granule Boundary", "Dynamic" and "Max Index" have any relations with CF conventions. .

8.3 Summary of the mapping of JPSS XML metadata to CF

Here we summarize the possible mapping of JPSS XML metadata to CF, as we discussed in previous sections. Table 8 summarizes the mapping of CF-equivalent metadata to CF attributes. Table 9 summarizes the handling of other CF-related issues.

Table 8: Mapping XML CF-equivalent metadata to CF attributes

CF-equivalent metadata	JPSS variables	Descriptions	Issues	Possible CF attributes
Unit of Measure	VIIRS EDR Geolocation XML – Latitude and Longitude	units = "degrees"		Latitude: units ="degrees_north" Longitude: units ="degrees_east"
	Other variables	All contain values under "Unit of Measure".		units = value of "Unit of Measure"
Description	All variables	CF attribute long_name is a descriptive name that indicates a variable's contents.		long_name = value of "Description"
Fill Values	VIIRS SDR Data XML – Radiance and Reflectance	Have multiple fill values The fill values are 65535, 65534, 65533, 65532, 65531, 65529, 65528. The data type is unsigned 16-bit integer.	Need to know if any data value can be 65530.	If the measured data value cannot be 65530, we can use the following CF attributes to make the netCDF-CF tools not display the multiple fill values. valid_min = 0 valid_max = 65527
	VIIRS SDR Data XML – other variables –	Have multiple fill values. The Fill values are 253,252 and 250. The	It may not be important to make these variables follow CF	If the measured data value cannot be 253,252 and 250, we can use the following

	ModeGran, ModeScan	datatype is unsigned 8-bit char.	conventions. If yes, need to know if any data value of ModeGran and ModeScan can be 253,252 and 250.	CF attributes to make the netCDF-CF tools not display the multiple fill values. valid_min = 0 valid_max = 248
	VIIRS SDR Data XML – other variables – NumberOfMissingPkts, NumberOfDiscardedPkts, NumberOfBadChecksums	Have multiple fill values. The fill values are -998 and -993. The datatype is 32-bit integer.	It seems that the values of these variables may always be no less than 0.	No CF Fill Values attributes are necessary.
	VIIRS EDR Geolocation XML – Latitude and Longitude	May have multiple fill values. The fill values are -999.9, -999.8, -999.5, -999.4 and -999.3. The datatype is 32-bit floating point. The valid range of Latitude is from -90 to 90. The valid range of Longitude is from -180 to 180.	It is strongly recommended that no fill values are present in Latitude and Longitude.	Latitude: valid_min = -90.0 valid_max = 90.0 Longitude: valid_min = -180.0 valid_max = 180.0
	VIIRS EDR Geolocation XML - Time	Have multiple fill values. It has two other attributes: “Range Min” and “Range Max”. Fill values are outside the valid range.		valid_min =1483228832000000 valid_max =2272147232000000
Scale **Factor** and Offset	VIIRS SDR Data XML – Radiance and Reflectance	Scale Factor = the first array element of the variable RadianceFactors/ReflectanceFactors Offset = the second	Need to clarify 1. The Offset value. 2. Whether the equation that calculates the final data value is	If the equation follows CF conventions: scale_factor= first array element of RadianceFactors/Ref

		array element of the variable RadianceFactors/ReflectanceFactors or 0	following CF conventions: Final_data_value = "Scale Factor" * Raw_data_value + "Offset"	electanceFactors add_offset = second array element of RadianceFactors/ReflectanceFactors or 0
	Other variables in VIIRS SDR Data XML and VIIRS EDR Geolocation XML files	The "Scale Factor" is Unscaled and the "Offset" is 0.		Two ways: 1. Don't provide "scale_factor" and "add_offset" at all. 2. scale_factor = 1 add_offset = 0

Table 9: The handling of Other CF-related Issues

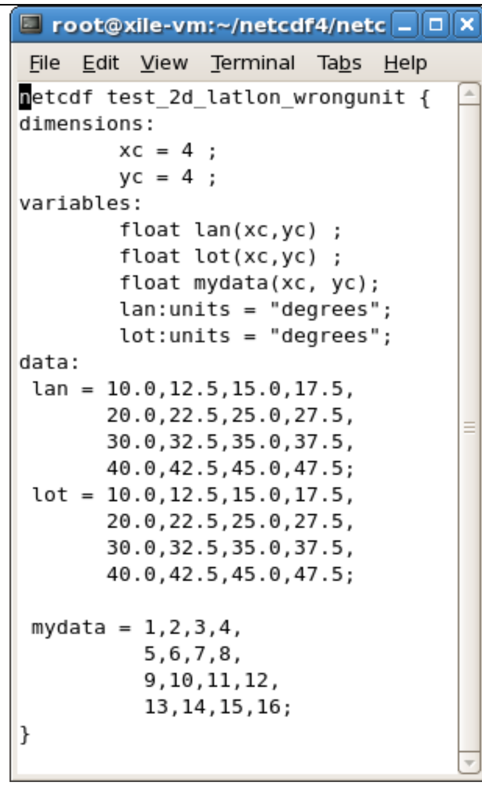
Items	Descriptions	Issues	Possible Solutions related to CF
VIIRS SDR Fields	The JPSS dimension names are different than the corresponding coordinate variable names. For example,The dimension names are "AlongTrack" and "CrossTrack". The coordinate variables are "Latitude" and "Longitude".		A "coordinates" attribute that lists the name of the coordinate variables for each variable.
VIIRS EDR Geolocation Fields	Latitude and Longitude		A CF attribute "standard_name" should be provided to make more tools identify if the variable stores latitude or longitude values. Latitude: standard_name="latitude" Standard_name ="longitude"

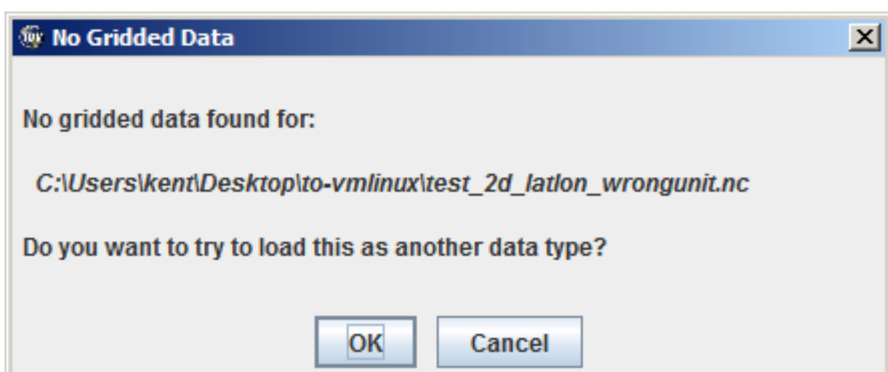
	Other Geolocation Fields (including Time and Height)		No equivalent CF coordinates Suggest mapping geolocation fields to general netCDF variables.
Special characters	According to CF-naming conventions, variable, dimension and attribute names should begin with a letter and be composed of letters, digits, and underscores.	We are not certain if all variable, dimension and attribute names in JPSS files are composed of letters, digits and underscores.	Suggest changing any special character to underscore for the variable, dimension and attribute names containing special characters. To preserve the original variable names, add a new attribute "original_name" and save the original name value to this attribute.
Dimensions	The dimension name and length pair can be used to connect the coordinate variables with the physical variables.	***Whether "Granule Boundary", "Dynamic" and "Max Index" have any relation with CF conventions.***	

Appendix A: Demonstrations of CF requirements for JPSS VIIRS-like data with netCDF-4 classic files

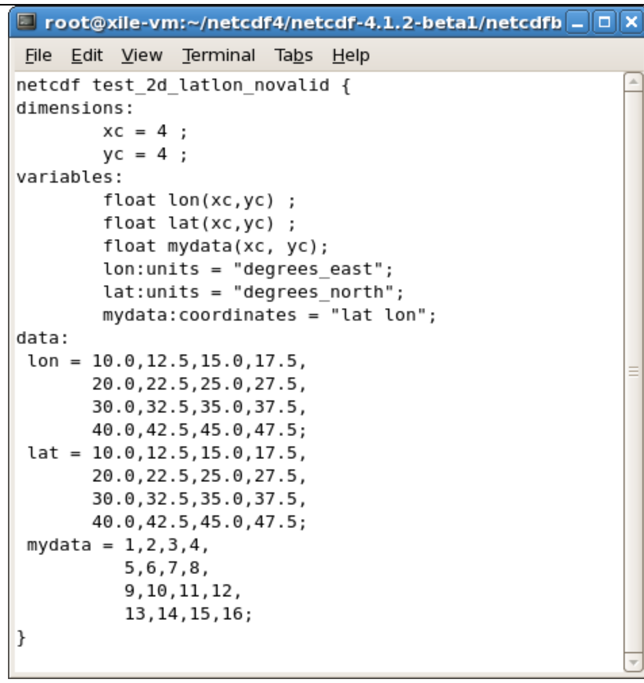
In this appendix, we use IDV to demonstrate CF requirements for JPSS VIIRS-like data with a few netCDF-4 classic files. The Common Data Language (CDL) file is the output file of a netCDF file generated by a netCDF dumper tool (ncdump). A CDL file can also be an input file for ncgen, a netCDF file generation tool, to generate a netCDF3 file or a netCDF4 file. The netCDF-4 classic files are also HDF5 files. One can use HDF5 tools to view the contents. To clearly demonstrate various CF requirements, the files we created are fake files and are plotted with the "value plots" feature of IDV to show the exact data values of the netCDF-4 classic variables.

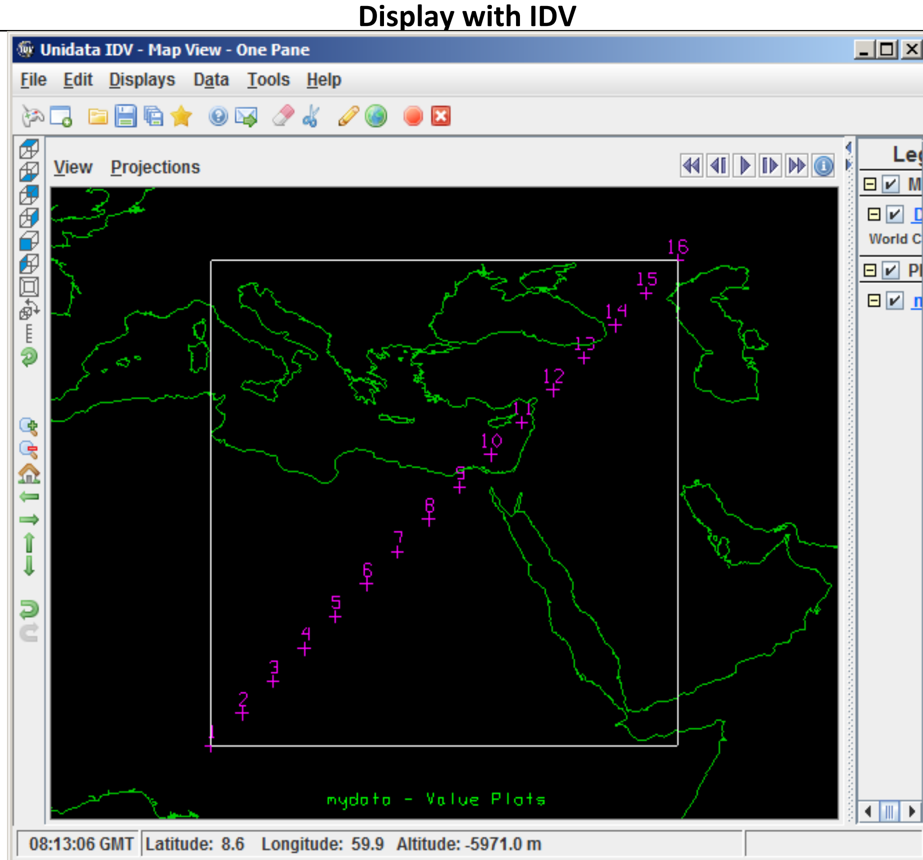
Case 1: "units" of a coordinate variable that doesn't follow CF conventions

<table>
<tr><th>CF description</th><th>CDL file</th></tr>
<tr><td rowspan="3">"units" of both latitude and longitude are "degrees", which don't follow CF conventions. IDV cannot open this netCDF file.</td><td>
root@xile-vm:~/netcdf4/netc
File Edit View Terminal Tabs Help
netcdf test_2d_latlon_wrongunit {
dimensions:
xc = 4 ;
yc = 4 ;
variables:
float lan(xc,yc) ;
float lot(xc,yc) ;
float mydata(xc, yc);
lan:units = "degrees";
lot:units = "degrees";
data:
lan = 10.0,12.5,15.0,17.5,
20.0,22.5,25.0,27.5,
30.0,32.5,35.0,37.5,
40.0,42.5,45.0,47.5;
lot = 10.0,12.5,15.0,17.5,
20.0,22.5,25.0,27.5,
30.0,32.5,35.0,37.5,
40.0,42.5,45.0,47.5;

mydata = 1,2,3,4,
5,6,7,8,
9,10,11,12,
13,14,15,16;
}</td></tr>
<tr><th>Display with IDV</th></tr>
<tr><td>
</td></tr>
</table>

Case 2: "units" of a two-dimensional coordinate variable that follows CF conventions

<table>
<tr><th>CF description</th><th>CDL file</th></tr>
<tr><td rowspan="3">Latitude:
Units = degrees_north
Longitude:
Units = degrees_east
Follow CF conventions,
IDV can open the file and correctly display the data. Please note that even removing the "coordinates" attribute, IDV can still display the data.</td><td>

```
netcdf test_2d_latlon_novalid {
dimensions:
        xc = 4 ;
        yc = 4 ;
variables:
        float lon(xc,yc) ;
        float lat(xc,yc) ;
        float mydata(xc, yc);
        lon:units = "degrees_east";
        lat:units = "degrees_north";
        mydata:coordinates = "lat lon";
data:
 lon = 10.0,12.5,15.0,17.5,
       20.0,22.5,25.0,27.5,
       30.0,32.5,35.0,37.5,
       40.0,42.5,45.0,47.5;
 lat = 10.0,12.5,15.0,17.5,
       20.0,22.5,25.0,27.5,
       30.0,32.5,35.0,37.5,
       40.0,42.5,45.0,47.5;
 mydata = 1,2,3,4,
          5,6,7,8,
          9,10,11,12,
          13,14,15,16;
}
```
</td></tr>
<tr><td>Display with IDV</td></tr>
<tr><td>
</td></tr>
</table>

Case 3: "units" of a three-dimensional coordinate variable that follows CF conventions

CF description

Latitude:

units = degrees_north

Longitude:

units = degrees_east

Pressure:

units = hPa

mydata: coordinates

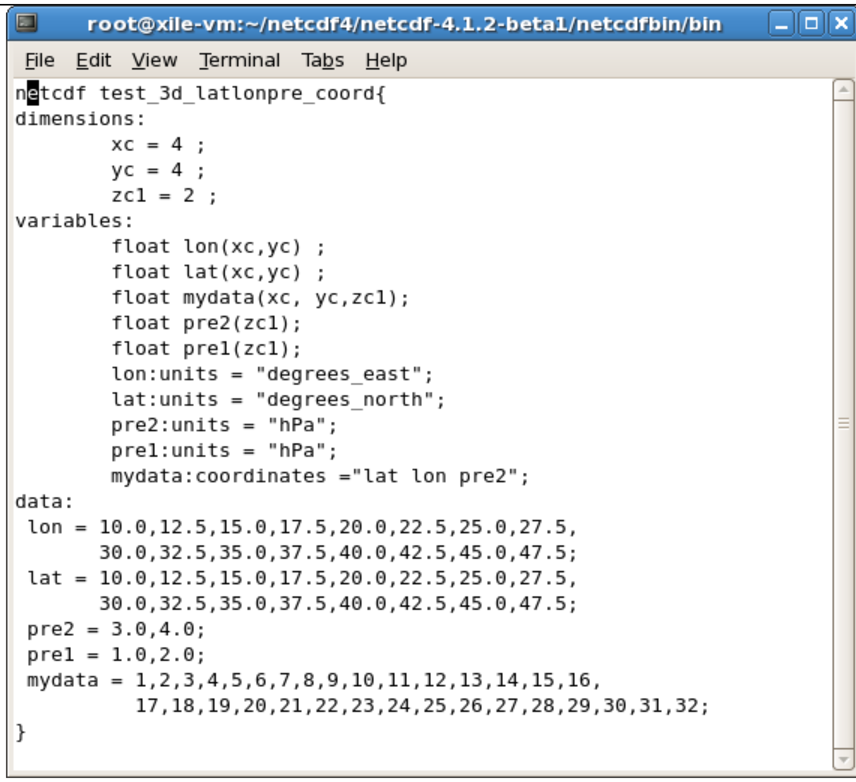
follow CF conventions.

Please note that mydata: coordinates is necessary to tell IDV that the vertical dimension is pre2 instead of pre1. mydata is a 3-D array float mydata[4][4][2].

The values of pre2 are 3.0 and 4.0. So the subset values of mydata at mydata[4][4][0:1]

(level 3 hPa) are 1,3,5,7,9,.....,29,31; which are correctly displayed by IDV.

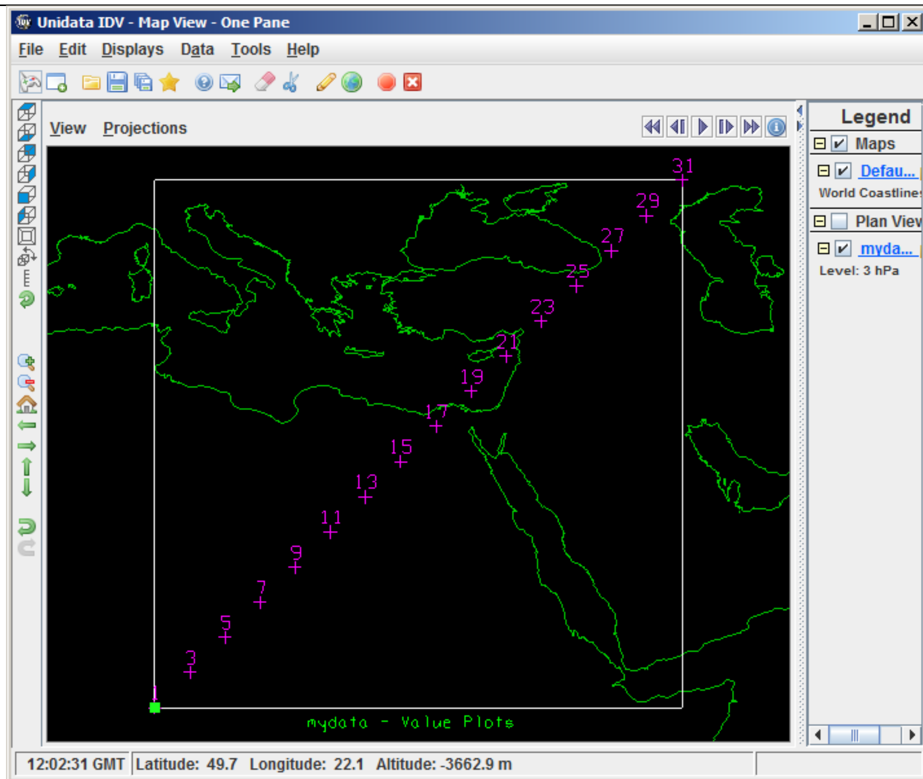
CDL file


root@xile-vm:~/netcdf4/netcdf-4.1.2-beta1/netcdfbin/bin

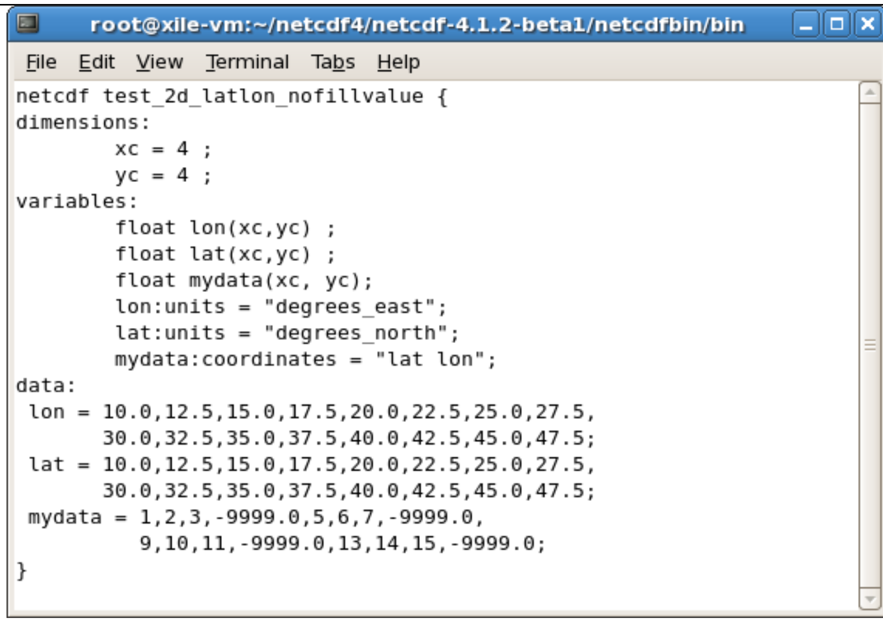
File Edit View Terminal Tabs Help

```
netcdf test_3d_latlonpre_coord{
dimensions:
        xc = 4 ;
        yc = 4 ;
        zc1 = 2 ;
variables:
        float lon(xc,yc) ;
        float lat(xc,yc) ;
        float mydata(xc, yc,zc1);
        float pre2(zc1);
        float pre1(zc1);
        lon:units = "degrees_east";
        lat:units = "degrees_north";
        pre2:units = "hPa";
        pre1:units = "hPa";
        mydata:coordinates ="lat lon pre2";
data:
 lon = 10.0,12.5,15.0,17.5,20.0,22.5,25.0,27.5,
       30.0,32.5,35.0,37.5,40.0,42.5,45.0,47.5;
 lat = 10.0,12.5,15.0,17.5,20.0,22.5,25.0,27.5,
       30.0,32.5,35.0,37.5,40.0,42.5,45.0,47.5;
 pre2 = 3.0,4.0;
 pre1 = 1.0,2.0;
 mydata = 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,
          17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32;
}
```

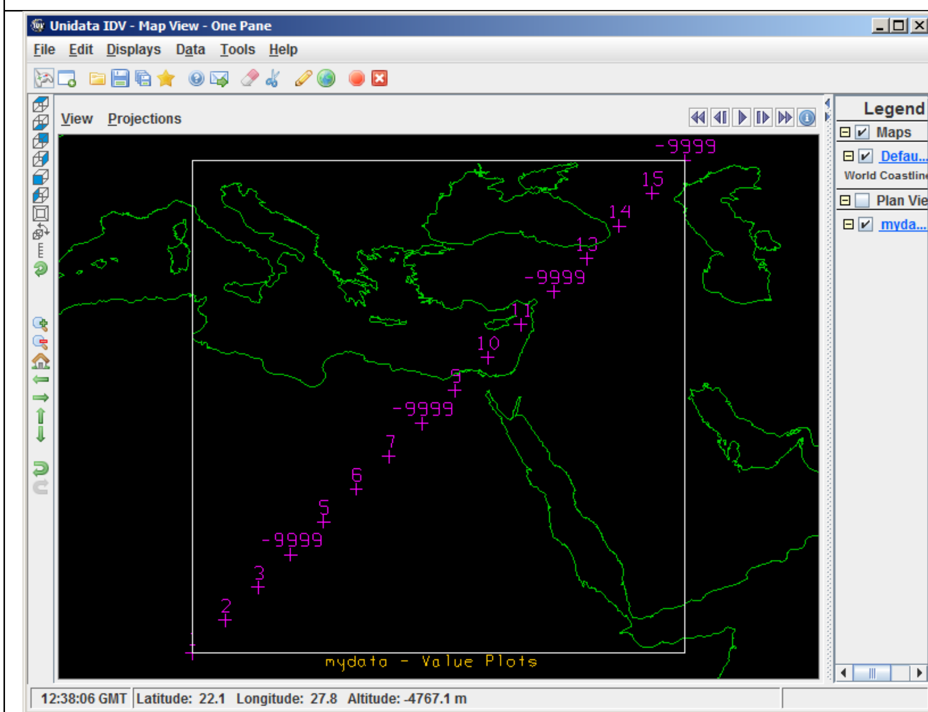
Display with IDV



Case 4: Have missing data but without using _FillValue attribute

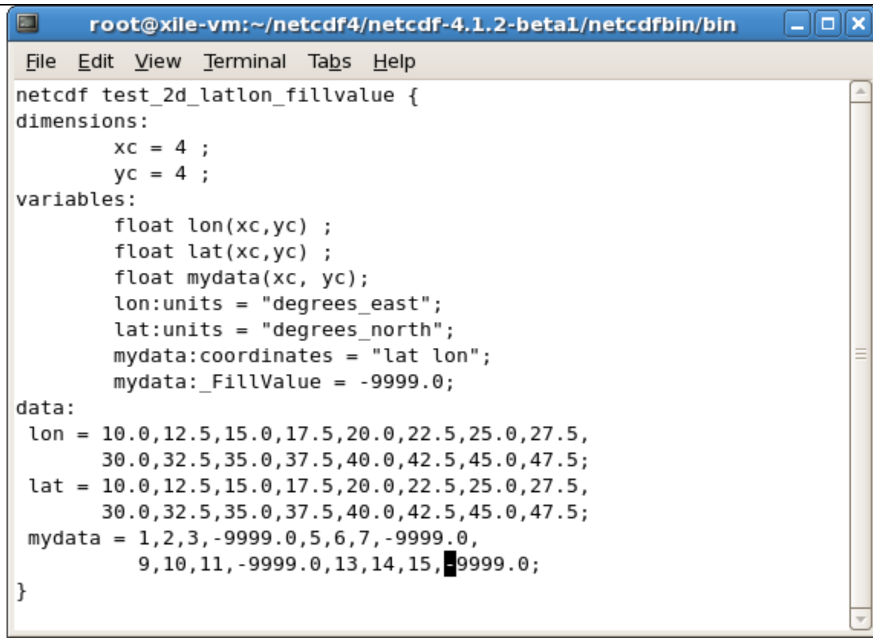
CF description	CDL file
Coordinate variables follow CF conventions. The real data variable mydata contains a fill value -9999.0. However, CF attribute _Fillvalue is not specified. IDV will treat -9999.0 as real value.	

Display with IDV



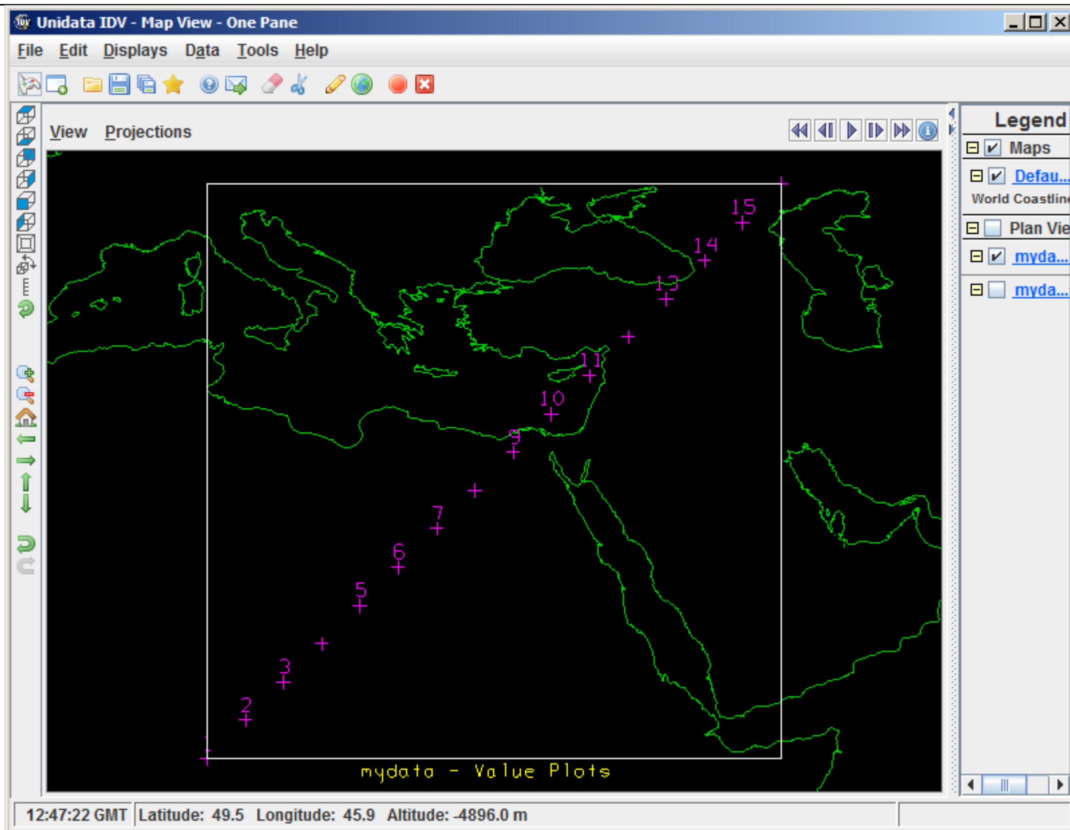
Case 5: Have missing data and have _FillValue attribute

CF description	CDL file
CF _FillValue attribute is given. IDV will correctly omit fill values. Note that no number is displayed for the data location that have fill values.	(see terminal screenshot below)



```
netcdf test_2d_latlon_fillvalue {
dimensions:
        xc = 4 ;
        yc = 4 ;
variables:
        float lon(xc,yc) ;
        float lat(xc,yc) ;
        float mydata(xc, yc);
        lon:units = "degrees_east";
        lat:units = "degrees_north";
        mydata:coordinates = "lat lon";
        mydata:_FillValue = -9999.0;
data:
 lon = 10.0,12.5,15.0,17.5,20.0,22.5,25.0,27.5,
       30.0,32.5,35.0,37.5,40.0,42.5,45.0,47.5;
 lat = 10.0,12.5,15.0,17.5,20.0,22.5,25.0,27.5,
       30.0,32.5,35.0,37.5,40.0,42.5,45.0,47.5;
 mydata = 1,2,3,-9999.0,5,6,7,-9999.0,
          9,10,11,-9999.0,13,14,15,-9999.0;
}
```

Display with IDV



Case 6: Have multiple special values but the file doesn't follow CF conventions

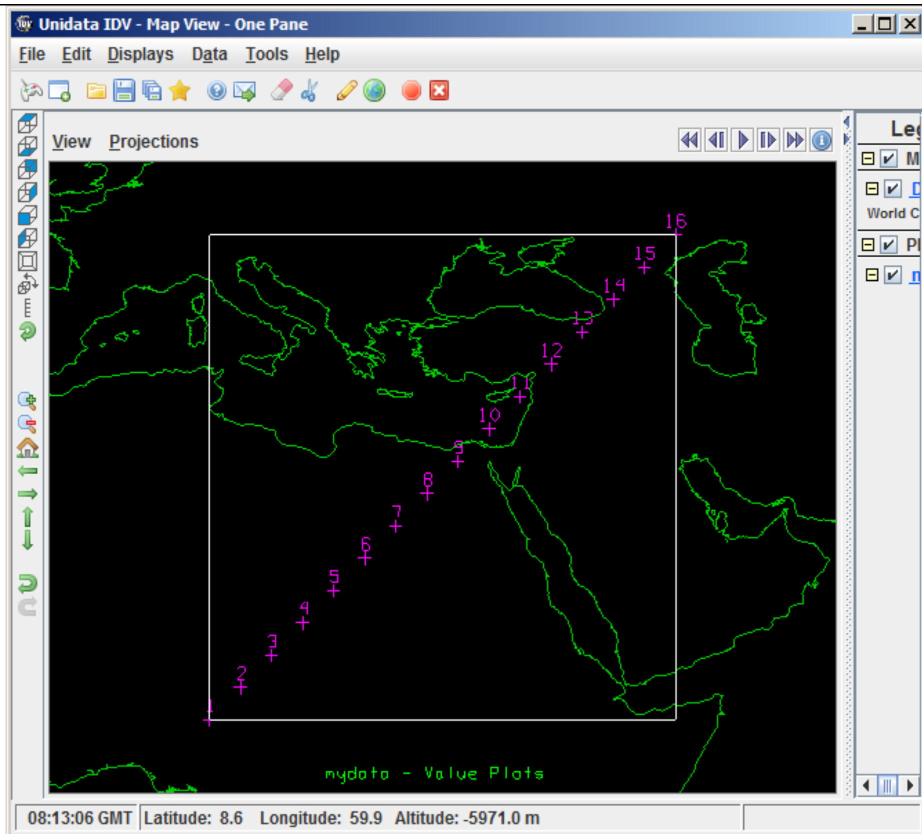
CF description

In this case, we assume that the data values greater than 10 are special values that should not be treated as real data. However, no CF attributes valid_min and valid_max are specified. All data values are simply displayed.

CDL file

```
netcdf test_2d_latlon_novalid {
dimensions:
        xc = 4 ;
        yc = 4 ;
variables:
        float lon(xc,yc) ;
        float lat(xc,yc) ;
        float mydata(xc, yc);
        lon:units = "degrees_east";
        lat:units = "degrees_north";
        mydata:coordinates = "lat lon";
data:
 lon = 10.0,12.5,15.0,17.5,
       20.0,22.5,25.0,27.5,
       30.0,32.5,35.0,37.5,
       40.0,42.5,45.0,47.5;
 lat = 10.0,12.5,15.0,17.5,
       20.0,22.5,25.0,27.5,
       30.0,32.5,35.0,37.5,
       40.0,42.5,45.0,47.5;
 mydata = 1,2,3,4,
          5,6,7,8,
          9,10,11,12,
          13,14,15,16;
}
```

Display with IDV

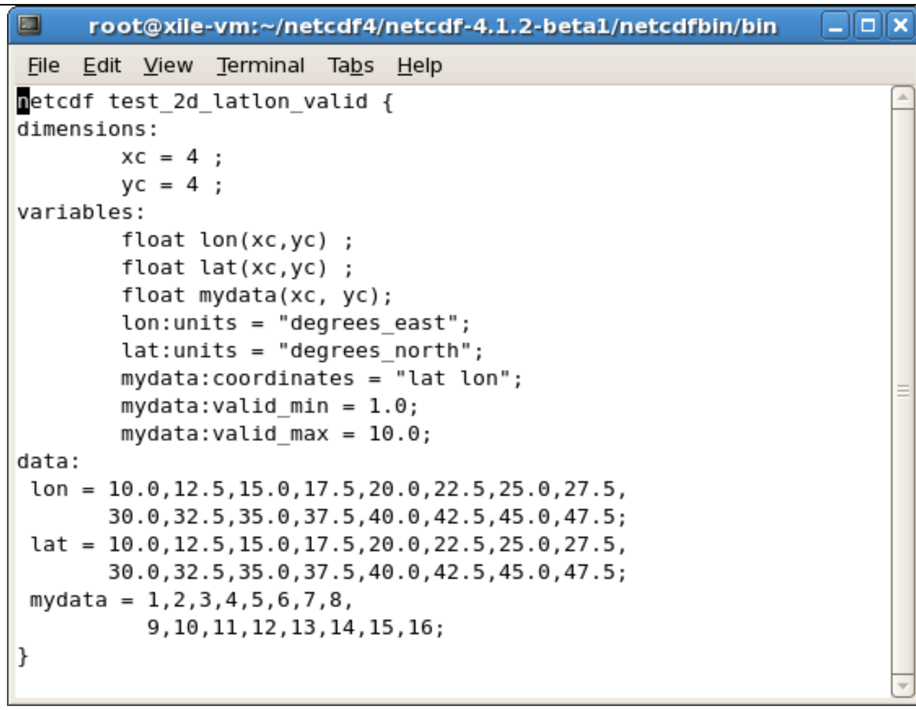


Case 7: Have multiple special values but follows CF conventions

CF description

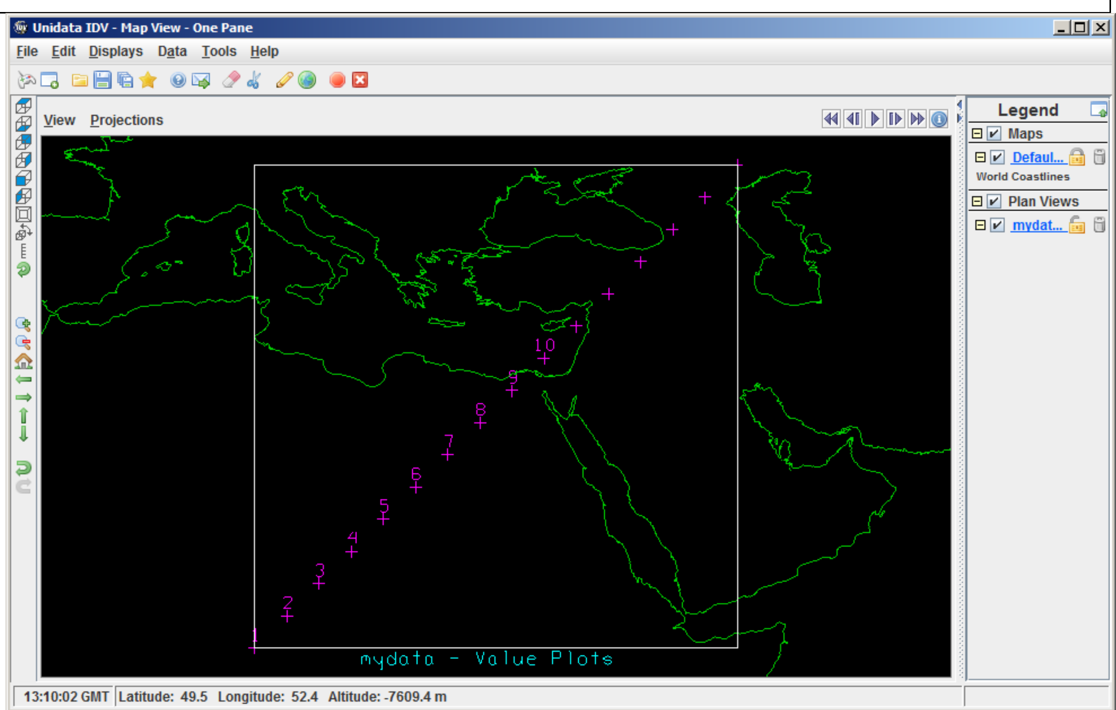
In this case, we assume that the data values greater than 10 are special values that should not be treated as real data. CF attributes valid_min and valid_max are specified. IDV correctly omit data values greater than 10.

CDL file

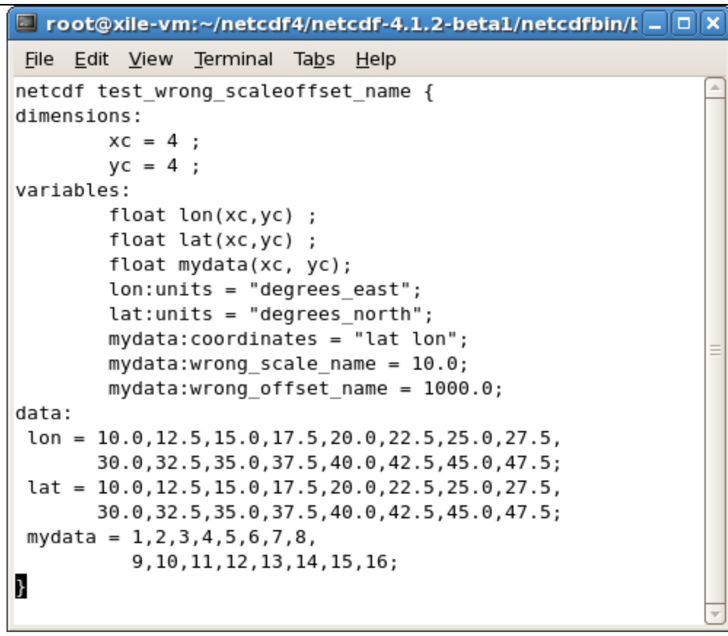
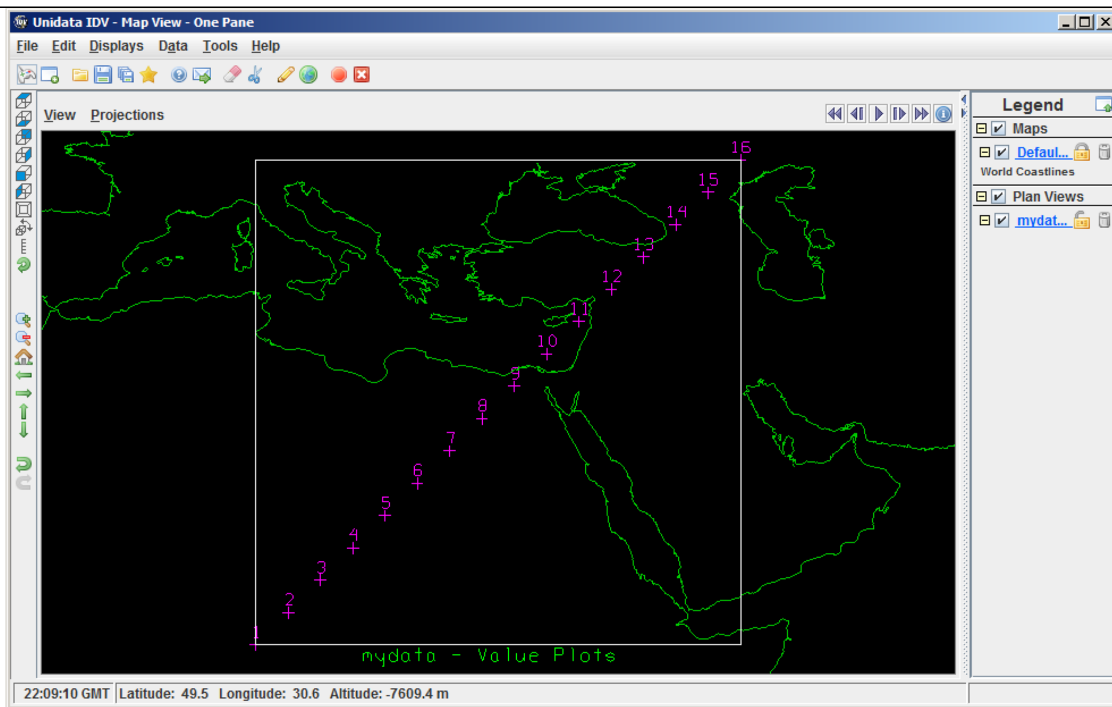



```
netcdf test_2d_latlon_valid {
dimensions:
        xc = 4 ;
        yc = 4 ;
variables:
        float lon(xc,yc) ;
        float lat(xc,yc) ;
        float mydata(xc, yc);
        lon:units = "degrees_east";
        lat:units = "degrees_north";
        mydata:coordinates = "lat lon";
        mydata:valid_min = 1.0;
        mydata:valid_max = 10.0;
data:
 lon = 10.0,12.5,15.0,17.5,20.0,22.5,25.0,27.5,
       30.0,32.5,35.0,37.5,40.0,42.5,45.0,47.5;
 lat = 10.0,12.5,15.0,17.5,20.0,22.5,25.0,27.5,
       30.0,32.5,35.0,37.5,40.0,42.5,45.0,47.5;
 mydata = 1,2,3,4,5,6,7,8,
          9,10,11,12,13,14,15,16;
}
```

Display with IDV



Case 8: Have scale offset factor but doesn’t follow CF conventions

<table>
<tr><td>CF description</td><td>CDL file</td></tr>
<tr><td rowspan="3">This file has scale, offset attributes. However, the attribute names don’t follow CF conventions. IDV simply display the data values without applying scale and offset factors.</td><td>
root@xile-vm:~/netcdf4/netcdf-4.1.2-beta1/netcdfbin/t
File Edit View Terminal Tabs Help
netcdf test_wrong_scaleoffset_name {
dimensions:
xc = 4 ;
yc = 4 ;
variables:
float lon(xc,yc) ;
float lat(xc,yc) ;
float mydata(xc, yc);
lon:units = "degrees_east";
lat:units = "degrees_north";
mydata:coordinates = "lat lon";
mydata:wrong_scale_name = 10.0;
mydata:wrong_offset_name = 1000.0;
data:
lon = 10.0,12.5,15.0,17.5,20.0,22.5,25.0,27.5,
30.0,32.5,35.0,37.5,40.0,42.5,45.0,47.5;
lat = 10.0,12.5,15.0,17.5,20.0,22.5,25.0,27.5,
30.0,32.5,35.0,37.5,40.0,42.5,45.0,47.5;
mydata = 1,2,3,4,5,6,7,8,
9,10,11,12,13,14,15,16;
}</td></tr>
<tr><td>Display with IDV</td></tr>
<tr><td></td></tr>
</table>

Case 9: Have CF scale_factor and add_offset attributes

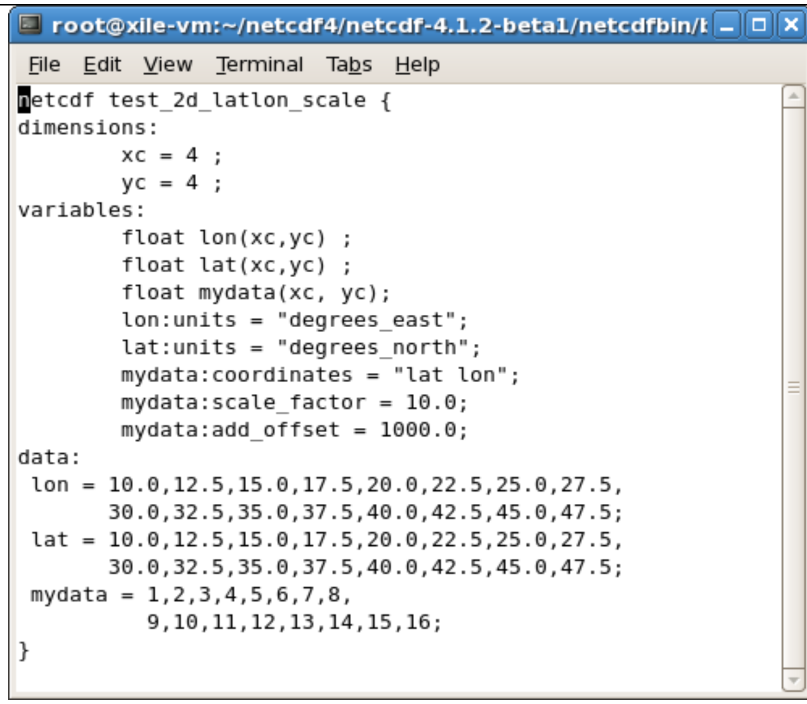
CF description

CF scale_factor and add_offset attributes are provided to the variable mydata. IDV will apply the scale_factor and add_offset to every data value by following the linear equation:

Final data = scale_factor *given data + add_offset

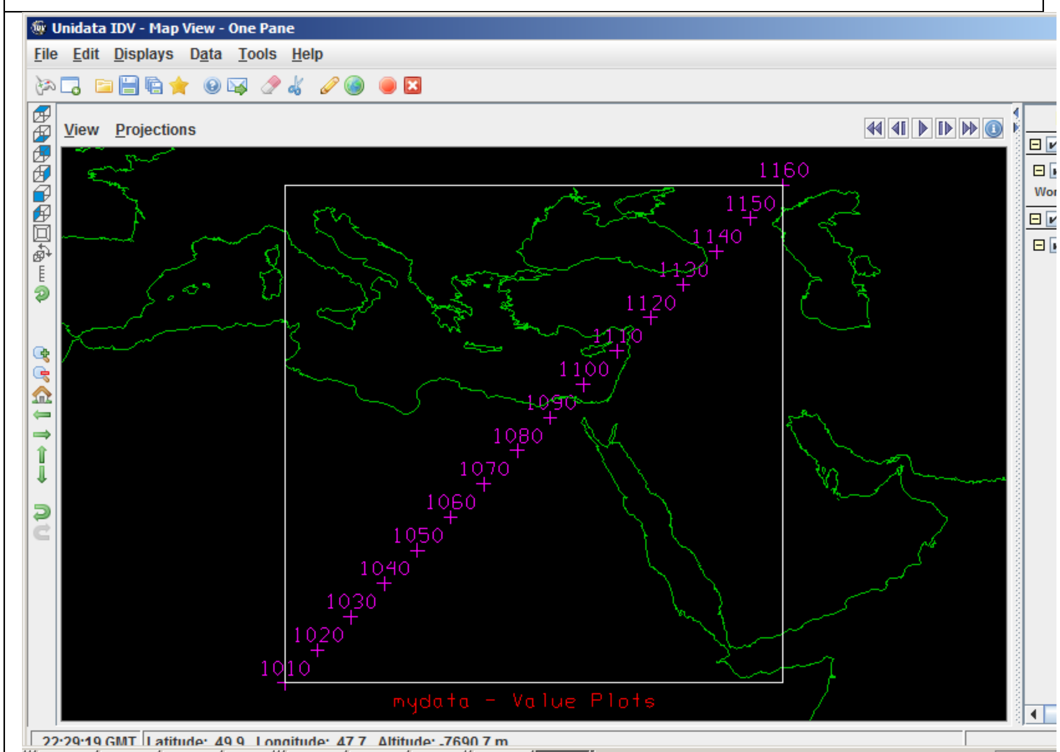
For example, the given value at the first location is 1; the final value at this location = 1*10+1000=1010.

CDL file



```
netcdf test_2d_latlon_scale {
dimensions:
        xc = 4 ;
        yc = 4 ;
variables:
        float lon(xc,yc) ;
        float lat(xc,yc) ;
        float mydata(xc, yc);
        lon:units = "degrees_east";
        lat:units = "degrees_north";
        mydata:coordinates = "lat lon";
        mydata:scale_factor = 10.0;
        mydata:add_offset = 1000.0;
data:
 lon = 10.0,12.5,15.0,17.5,20.0,22.5,25.0,27.5,
       30.0,32.5,35.0,37.5,40.0,42.5,45.0,47.5;
 lat = 10.0,12.5,15.0,17.5,20.0,22.5,25.0,27.5,
       30.0,32.5,35.0,37.5,40.0,42.5,45.0,47.5;
 mydata = 1,2,3,4,5,6,7,8,
          9,10,11,12,13,14,15,16;
}
```

Display with IDV



Case 10: Have a special character inside the CDL file

<table>
<tr><th>CF description</th><th>CDL file</th></tr>
<tr><td rowspan="3">In this case, the variable name is “my*data”. It includes a special character “*”. So even the netCDF file cannot be generated by ncgen.</td><td>

root@xile-vm:~/netcdf4/netcdf-4.1.2-beta1/netcdfbin/bin

File Edit View Terminal Tabs Help

```
netcdf test_special_character {
dimensions:
        xc = 4 ;
        yc = 4 ;
variables:
        float lon(xc,yc) ;
        float lat(xc,yc) ;
        float mydata(xc, yc);
        lon:units = "degrees_east";
        lat:units = "degrees_north";
        my*data:coordinates = "lat lon";
data:
 lon = 10.0,12.5,15.0,17.5,20.0,22.5,25.0,27.5,
       30.0,32.5,35.0,37.5,40.0,42.5,45.0,47.5;
 lat = 10.0,12.5,15.0,17.5,20.0,22.5,25.0,27.5,
       30.0,32.5,35.0,37.5,40.0,42.5,45.0,47.5;
 my*data = 1,2,3,4,5,6,7,8,
          9,10,11,12,13,14,15,16;
}
```

</td></tr>
<tr><td>Display with IDV</td></tr>
<tr><td>Cannot even generate a valid netCDF file. The error message is:

“Undefined name: my”</td></tr>
</table>

Appendix B: The Dump of a JPSS VIIRS SDR data XML file

NPOESS Product Definition

SVI01: VIIRS Imagery Band 01 SDR

Table of Granule Dimensions

Name	Granule Boundary	Dynamic	Max Index
AlongTrack	Yes	No	1536
CrossTrack	No	No	6400
Detector	Yes	No	32
Factors	Yes	No	2
Granule	Yes	No	1
Granule	Yes	No	3
Scan	Yes	No	48

Data Content Summary

Field Name	Description	Data Type	Dimensions	Units
ModeGran	The VIIRS operational mode, reported at the granule level	unsigned 8-bit char	[N* 1]	unitless
ModeScan	The VIIRS operational mode, reported at the scan level	unsigned 8-bit char	[N* 48]	unitless
NumberOfBadChecksums	Number of packets with bad checksum in scan	32-bit integer	[N* 48]	unitless
NumberOfDiscardedPkts	Number of discarded packets in scan	32-bit integer	[N* 48]	unitless
NumberOfMissingPkts	Number of missing packets in scan	32-bit integer	[N* 48]	unitless
NumberOfScans	Actual number of VIIRS scans that were used to create this granule	32-bit integer	[N* 1]	unitless
PadByte1	Pad byte	unsigned 8-bit char	[N* 3]	unitless
QF1_VIIRSIBANDSDR	SDR Quality - Indicates calibration	2 bit(s)	[N* 1536 ,	unitless

	quality due to bad space view offsets, OBC view offsets, etc or use of a previous calibration view Saturated Pixel - Indicates the level of pixel saturation Missing Data - Data required for calibration processing is not available for processing Out of Range - Calibrated pixel value outside of LUT threshold limits	2 bit(s) 2 bit(s) 2 bit(s)	6400]	unitless unitless unitless
QF2_SCAN_SDR	Half Angle Mirror Side The Moon has corrupted the space view Spare	1 bit(s) 1 bit(s) 6 bit(s)	[N* 48]	unitless unitless unitless
QF3_SCAN_RDR	Checksum failed for zone 1 Checksum failed for zone 2 Checksum failed for zone 3 Checksum failed for zone 4 Checksum failed for zone 5 Checksum failed for zone 6 Scan data is not Present (No valid data) Spare	1 bit(s) 1 bit(s) 1 bit(s) 1 bit(s) 1 bit(s) 1 bit(s) 1 bit(s) 1 bit(s)	[N* 48]	unitless unitless unitless unitless unitless unitless unitless unitless
QF4_SCAN_SDR	Quality for this scan-line is reduced. The value is determined by the combined number of steps required to find a replacement for thermistor or calibration source data	unsigned 8-bit char	[N* 1536]	unitless
QF5_GRAN_BADDETECTOR	Bad Detector Spare	1 bit(s) 7 bit(s)	[N* 32]	unitless unitless
Radiance	Calibrated Top of Atmosphere (TOA) Radiance for each VIIRS pixel	unsigned 16-bit integer	[N* 1536 , 6400]	W/(m^2 sr µm)
RadianceFactors	Scale = first array element; Offset = second array element	32-bit floating point	[N* 2]	scale = unitless; offset = W/(m^2 µm sr)

Reflectance	Calibrated Top of Atmosphere (TOA) Reflectance for each VIIRS pixel	unsigned 16-bit integer	[N* 1536 , 6400]	unitless
ReflectanceFactors	Scale = first array element; Offset = second array element	32-bit floating point	[N* 2]	unitless

VIIRS I-Band SDR Data Product Profile

Fields

Name	**Data Size**	**Description**
Radiance	2byte(s)	(see Dimension(s) and Datum below)

Dimension(s)

Dimension Name	**Granule Bounding**	**Dynamic**	**Max Index**	**Min Index**
AlongTrack	Yes	No	1536	1536
CrossTrack	No	No	6400	6400

Datum

Description	**Offset**	**Scale Factor**	**Unit of Measure**	**Range Min**	**Range Max**	**Data Type**	**Fill Values**
Calibrated Top of Atmosphere (TOA) Radiance for each VIIRS pixel	0	Scaled using RadianceFactors	W/(m^2 sr µm)			unsigned 16-bit integer	(see Fill Values below)

Name	**Value**
NA_UINT16_FILL	65535
MISS_UINT16_FILL	65534
ONBOARD_PT_UINT16_FILL	65533
ONGROUND_PT_UINT16_FILL	65532
ERR_UINT16_FILL	655

Name	Value
	31
VDNE_UINT16_FILL	65529
SOUB_UINT16_FILL	65528

Reflectance — 2byte(s)

Dimension(s)

Dimension Name	Granule Bounding	Dynamic	Max Index	Min Index
AlongTrack	Yes	No	1536	1536
CrossTrack	No	No	6400	6400

Datum

Description	Offset	Scale Factor	Unit of Measure	Range Min	Range Max	Data Type	Fill Values
Calibrated Top of Atmosphere (TOA) Reflectance for each VIIRS pixel	0	Scaled using Reflectance Factors	unitless			unsigned 16-bit integer	see Fill Values table below

Fill Values

Name	Value
NA_UINT16_FILL	65535
MISS_UINT16_FILL	65534
ONBOARD_PT_UINT16_FILL	65533
ONGROUND_PT_UINT16_FILL	65532
ERR_UINT16_FILL	65531
VDNE_UINT16_FILL	65529
SOUB_UINT16_FILL	65528

ModeScan	1byte(s)	See Dimension(s) and Datum below

Dimension(s)

Dimension Name	Granule Bounding	Dynamic	Max Index	Min Index
Scan	Yes	No	48	48

Datum

Description	Offset	Scale Factor	Unit of Measure	Range Min	Range Max	Data Type	Fill Values	Legend Entries
The VIIRS operational mode, reported at the scan level	0	**Unscaled**	unitless			unsigned 8-bit char	(see Fill Values below)	(see Legend Entries below)

Fill Values:

Name	Value
MISS_UINT8_FILL	254
ERR_UINT8_FILL	251
VDNE_UINT8_FILL	249

Legend Entries:

Value	Meaning
0	Night
1	Day

ModeGran	1byte(s)	See Dimension(s) and Datum below

Dimension(s)

Dimension Name	Granule Bounding	Dynamic	Max Index	Min Index
Granule	Yes	No	1	1

Datum

Description	Offset	Scale Factor	Unit of Measure	Range Min	Range Max	Data Type	Fill Values	Legend Entries
The VIIRS operational mode, reported at the granule	0	**Unscaled**	unitless			unsigned 8-bit char	(see Fill Values below)	(see Legend Entries below)

Fill Values:

Name	Value
MISS_UINT8_FILL	254
ERR_UINT8_FILL	251

Legend Entries:

Value	Meaning
0	Night
1	Day
2	Mixed

level							VDNE_UINT8_FILL	249	

PadByte1 — 1byte(s)

Dimension(s)				
Dimension Name	**Granule Bounding**	**Dynamic**	**Max Index**	**Min Index**
Granule	Yes	No	3	3

Datum						
Description	**Offset**	**Scale Factor**	**Unit of Measure**	**Range Min**	**Range Max**	**Data Type**
Pad byte	0	**Unscaled**	unitless			unsigned 8-bit char

NumberOfScans — 4byte(s)

Dimension(s)				
Dimension Name	**Granule Bounding**	**Dynamic**	**Max Index**	**Min Index**
Granule	Yes	No	1	1

Datum						
Description	**Offset**	**Scale Factor**	**Unit of Measure**	**Range Min**	**Range Max**	**Data Type**
Actual number of VIIRS scans that were used to create this granule	0	**Unscaled**	unitless			32-bit integer

NumberOfMissingPkts — 4byte(s)

Dimension(s)				
Dimension Name	**Granule Bounding**	**Dynamic**	**Max Index**	**Min Index**
Scan	Yes	No	48	48

Datum								
Description	**Offset**	**Scale Factor**	**Unit of Measure**	**Range Min**	**Range Max**	**Data Type**	**Fill Values**	
Number of	0	**Unscaled**	unitless			32-bit integ	**Name**	**Valu**

missing packets in scan						er		e
							MISS_INT32_FILL	-998
							VDNE_INT32_FILL	-993

NumberOfBadChecksums — 4byte(s)

Dimension(s)

Dimension Name	Granule Bounding	Dynamic	Max Index	Min Index
Scan	Yes	No	48	48

Datum

Description	Offset	Scale Factor	Unit of Measure	Range Min	Range Max	Data Type	Fill Values	
Number of packets with bad checksum in scan	0	**Unscaled**	unitless			32-bit integer	**Name**	**Value**
							MISS_INT32_FILL	-998
							VDNE_INT32_FILL	-993

NumberOfDiscardedPkts — 4byte(s)

Dimension(s)

Dimension Name	Granule Bounding	Dynamic	Max Index	Min Index
Scan	Yes	No	48	48

Datum

Description	Offset	Scale Factor	Unit of Measure	Range Min	Range Max	Data Type	Fill Values	
Number of discarded packets in scan	0	**Unscaled**	unitless			32-bit integer	**Name**	**Value**
							MISS_INT32_FILL	-998

								VDNE_INT32_FILL	-993

QF1_VIIRSIBANDSDR — 1byte(s)

Dimension(s)

Dimension Name	**Granule Bounding**	**Dynamic**	**Max Index**	**Min Index**
AlongTrack	Yes	No	1536	1536
CrossTrack	No	No	6400	6400

Datum

Description	**Offset**	**Scale Factor**	**Unit of Measure**	**Range Min**	**Range Max**	**Data Type**	**Legend Entries**
SDR Quality - Indicates calibration quality due to bad space view offsets, OBC view offsets, etc or use of a previous calibration view	0	**Unscaled**	unitless			2 bit(s)	**Value** / **Meaning**: 0 Good; 1 Poor; 2 No Calibration
Saturated Pixel - Indicates the level of pixel saturation	2	**Unscaled**	unitless			2 bit(s)	**Value** / **Meaning**: 0 None Saturated; 1 Some Saturated; 2 All Saturated

		Missing Data - Data required for calibration processing is not available for processing	4	**Unscaled**	unitless			2 bit(s)	**Value** / **Meaning** 0 – All data present 1 – EV RDR data missing 2 – Cal data (SV, CV, SD, etc.) missing 3 – Thermistor data missing
		Out of Range - Calibrated pixel value outside of LUT threshold limits	6	**Unscaled**	unitless			2 bit(s)	**Value** / **Meaning** 0 – All data within range 1 – Radiance out of range 2 – Reflectance or EBBT out of range 3 – Both Radiance and Reflectance or EBBT out of range
QF2_SCAN_SDR	1byte(s)	**Dimension(s)**							

Dimension Name	Granule Bounding	Dynamic	Max Index	Min Index
Scan	Yes	No	48	48

Datum

Description	Offset	Scale Factor	Unit of Measure	Range Min	Range Max	Data Type	Legend Entries
Half Angle Mirror Side	0	**Unscaled**	unitless			1 bit(s)	**Value** / **Meaning** 0 / A-Side 1 / B-Side
The Moon has corrupted the space view	1	**Unscaled**	unitless			1 bit(s)	**Value** / **Meaning** 0 / False 1 / True
Spare	2	**Unscaled**	unitless			6 bit(s)	

QF3_SCAN_RDR	1byte(s)	see below

Dimension(s)

Dimension Name	Granule Bounding	Dynamic	Max Index	Min Index
Scan	Yes	No	48	48

Datum

Description	Offset	Scale Factor	Unit of Measure	Range Min	Range Max	Data Type	Legend Entries
Checksum failed for zone 1	0	**Unscaled**	unitless			1 bit(s)	**Value** / **Meaning** 0 / False 1 / True

		Checksum failed for zone 2	1	**Unscaled**	unitless			1 bit(s)	**Value** | **Meaning** 0 | False 1 | True
		Checksum failed for zone 3	2	**Unscaled**	unitless			1 bit(s)	**Value** | **Meaning** 0 | False 1 | True
		Checksum failed for zone 4	3	**Unscaled**	unitless			1 bit(s)	**Value** | **Meaning** 0 | False 1 | True
		Checksum failed for zone 5	4	**Unscaled**	unitless			1 bit(s)	**Value** | **Meaning** 0 | False 1 | True
		Checksum failed for zone 6	5	**Unscaled**	unitless			1 bit(s)	**Value** | **Meaning** 0 | False 1 | True
		Scan data is not Present (No valid data)	6	**Unscaled**	unitless			1 bit(s)	**Value** | **Meaning** 0 | False 1 | True
		Spare	7	**Unscaled**	unitless			1 bit(s)	

<table>
<tr>
<td>QF4_SCAN_SDR</td>
<td>1byte(s)</td>
<td>
<table>
<tr><th colspan="5">Dimension(s)</th></tr>
<tr><th>Dimension Name</th><th>Granule Bounding</th><th>Dynamic</th><th>Max Index</th><th>Min Index</th></tr>
<tr><td>AlongTrack</td><td>Yes</td><td>No</td><td>1536</td><td>1536</td></tr>
</table>
<table>
<tr><th colspan="8">Datum</th></tr>
<tr><th>Description</th><th>Offset</th><th>Scale Factor</th><th>Unit of Measure</th><th>Range Min</th><th>Range Max</th><th>Data Type</th><th>Legend Entries</th></tr>
<tr><td>Quality for this scan-line is reduced. The value is determined by the combined number of steps required to find a replacement for thermistor or calibration source data</td><td>0</td><td>Unscaled</td><td>unitless</td><td></td><td></td><td>unsigned 8-bit char</td><td>
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>0</td><td>False</td></tr>
<tr><td>>1</td><td>True</td></tr>
</table>
</td></tr>
</table>
</td>
</tr>
<tr>
<td>QF5_GRAN_BAD DETECTOR</td>
<td>1byte(s)</td>
<td>
<table>
<tr><th colspan="5">Dimension(s)</th></tr>
<tr><th>Dimension Name</th><th>Granule Bounding</th><th>Dynamic</th><th>Max Index</th><th>Min Index</th></tr>
<tr><td>Detector</td><td>Yes</td><td>No</td><td>32</td><td>32</td></tr>
</table>
<table>
<tr><th colspan="8">Datum</th></tr>
<tr><th>Description</th><th>Offset</th><th>Scale Factor</th><th>Unit of Measure</th><th>Range Min</th><th>Range Max</th><th>Data Type</th><th>Legend Entries</th></tr>
<tr><td>Bad</td><td>0</td><td>Unscale</td><td>unitless</td><td></td><td></td><td>1</td><td>
<table>
<tr><th>Valu</th><th>Meanin</th></tr>
</table>
</td></tr>
</table>
</td>
</tr>
</table>

<table>
<tr><td></td><td></td><td>
<table>
<tr><td>Detector</td><td></td><td>d</td><td></td><td></td><td></td><td>bit(s
)</td><td>
<table>
<tr><td>e</td><td>g</td></tr>
<tr><td>0</td><td>False</td></tr>
<tr><td>1</td><td>True</td></tr>
</table>
</td></tr>
<tr><td>Spare</td><td>1</td><td>Unscale
d</td><td>unitless</td><td></td><td></td><td>7
bit(s
)</td><td></td></tr>
</table>
</td></tr>
<tr><td>RadianceFactors</td><td>4byt
e(s)</td><td>
<table>
<tr><td colspan="5">Dimension(s)</td></tr>
<tr><td>Dimension Name</td><td>Granule Bounding</td><td>Dynamic</td><td>Max Index</td><td>Min Index</td></tr>
<tr><td>Factors</td><td>Yes</td><td>No</td><td>2</td><td>2</td></tr>
</table>
<table>
<tr><td colspan="7">Datum</td></tr>
<tr><td>Description</td><td>Offset</td><td>Scale Factor</td><td>Unit of Measure</td><td>Range Min</td><td>Range Max</td><td>Data Type</td></tr>
<tr><td>Scale = first array element; Offset = second array element</td><td>0</td><td>Unscaled</td><td>scale = unitless; offset = W/(m^2 µm sr)</td><td></td><td></td><td>32-bit floating point</td></tr>
</table>
</td></tr>
<tr><td>ReflectanceFact
ors</td><td>4byt
e(s)</td><td>
<table>
<tr><td colspan="5">Dimension(s)</td></tr>
<tr><td>Dimension Name</td><td>Granule Bounding</td><td>Dynamic</td><td>Max Index</td><td>Min Index</td></tr>
<tr><td>Factors</td><td>Yes</td><td>No</td><td>2</td><td>2</td></tr>
</table>
<table>
<tr><td colspan="7">Datum</td></tr>
<tr><td>Description</td><td>Offset</td><td>Scale Factor</td><td>Unit of Measure</td><td>Range Min</td><td>Range Max</td><td>Data Type</td></tr>
<tr><td>Scale = first array element; Offset = second array element</td><td>0</td><td>Unscaled</td><td>unitless</td><td></td><td></td><td>32-bit floating point</td></tr>
</table>
</td></tr>
</table>

Appendix C: The Dump of a JPSS VIIRS EDR geolocation XML file

NPOESS Product Definition

GNCCO: VIIRS NCC Band Imagery EDR GTM Geolocation Data

Table of Granule Dimensions

Name	Granule Boundary	Dynamic	Max Index
AlongTrack	Yes	No	771
AlongTrack	Yes	No	1
CrossTrack	No	No	4121
Granule	Yes	No	1
Granule	Yes	No	4

Data Content Summary

Field Name	Description	Data Type	Dimensions	Units
Height	Ellipsoid-Geoid separation	32-bit floating point	[N* 771 , 4121]	meter
Latitude	Latitude of each pixel (positive North)	32-bit floating point	[N* 771 , 4121]	degree
Longitude	Longitude of each pixel (positive East)	32-bit floating point	[N* 771 , 4121]	degree
LunarAzimuthAngle	Azimuth angle of moon (measured clockwise positive from North) at each pixel position	32-bit floating point	[N* 771 , 4121]	unitless
LunarZenithAngle	Zenith angle of moon at each pixel position	32-bit floating point	[N* 771 , 4121]	degree
MoonIllumFraction	Fraction of the moon illuminated (expressed as percent)	32-bit floating point	[N* 1]	percent

PadByte1	Pad byte	unsigned 8-bit char	[N* 4]	unitless
PixelColSDR	Day-Night Band SDR pixel column index number that was remapped to this GTM pixel (column numbering begins with zero)	unsigned 16-bit integer	[N* 771 , 4121]	unitless
PixelRowSDR	Day-Night Band SDR pixel row index number that was remapped to this GTM pixel (row numbering begins with zero)	unsigned 16-bit integer	[N* 771 , 4121]	unitless
QF1_VIIRSGTMGEO	DNB Pixel Mapping (GTM to SDR DNB). Indicates whether this pixel originated from the previous, current, or next granule in the SDR DNB geolocation. Spare	2 bit(s) 6 bit(s)	[N* 771 , 4121]	unitless unitless
QF2_VIIRSGTMGEO	Solar Eclipse Lunar Eclipse Spare	1 bit(s) 1 bit(s) 6 bit(s)	[N* 1]	unitless unitless unitless
SatelliteAzimuthAngle	Azimuth angle (measured clockwise positive from North) to Satellite at each pixel position	32-bit floating point	[N* 771 , 4121]	degree
SatelliteRange	Line of sight distance from the ellipsoid intersection to the satellite	32-bit floating point	[N* 771 , 4121]	meter
SatelliteZenithAngle	Zenith angle to Satellite at each pixel position	32-bit floating point	[N* 771 , 4121]	degree
SolarAzimuthAngle	Azimuth angle of sun (measured clockwise positive from North) at each pixel position	32-bit floating point	[N* 771 , 4121]	degree
SolarZenithAngle	Zenith angle of sun at each pixel position	32-bit floating point	[N* 771 , 4121]	degree
Time	Time of the nadir point of the GTM row in IET (1/1/1958). Represents the time of the nadir point of the GTM	64-bit integer	[N* 771]	microsecond

	row			

VIIRS NCC-Band Imagery GTM Geolocation Product Profile

Fields		
Name	**Data Size**	**Description**
Time	8byte(s)	**Dimension(s)**

Dimension(s)

Dimension Name	**Granule Bounding**	**Dynamic**	**Max Index**	**Min Index**
AlongTrack	Yes	No	771	771

Datum

Description	**Offset**	**Scale Factor**	**Unit of Measure**	**Range Min**	**Range Max**	**Data Type**	**Fill Values**
Time of the nadir point of the GTM row in IET (1/1/1958). Represents the time of the nadir point of the GTM row	0	**Unscaled**	microsecond	1483228832000000	2272147232000000	64-bit integer	see below

Fill Values

Name	**Value**
NA_INT64_FILL	-999
MISS_INT64_FILL	-998
ERR_INT64_FILL	-995
VDNE_INT64_FILL	-993

Name	**Data Size**	**Description**
Latitude	4byt	**Dimension(s)**

	e(s)	Dimension Name | Granule Bounding | Dynamic | Max Index | Min Index AlongTrack | Yes | No | 771 | 771 CrossTrack | No | No | 4121 | 4121 **Datum** Description | Offset | Scale Factor | Unit of Measure | Range Min | Range Max | Data Type | Fill Values Latitude of each pixel (positive North) | 0 | **Unscaled** | degree | | | 32-bit floating point | Name: Value — NA_FLOAT32_FILL: -999.9; MISS_FLOAT32_FILL: -999.8; ERR_FLOAT32_FILL: -999.5; ELINT_FLOAT32_FILL: -999.4; VDNE_FLOAT32_FILL: -999.3
Longitude	4byte(s)	**Dimension(s)** Dimension Name | Granule Bounding | Dynamic | Max Index | Min Index AlongTrack | Yes | No | 771 | 771 CrossTrack | No | No | 4121 | 4121 **Datum** Description | Offset | Scale Factor | Unit of Measu | Range | Range | Data Type | Fill Values

Description	Offset	Scale Factor	Unit of Measure	Range Min	Range Max	Data Type	Fill Values
Longitude of each pixel (positive East)	0	**Unscaled**	degree			32-bit floating point	see below

Name	Value
NA_FLOAT32_FILL	-999.9
MISS_FLOAT32_FILL	-999.8
ERR_FLOAT32_FILL	-999.5
ELINT_FLOAT32_FILL	-999.4
VDNE_FLOAT32_FILL	-999.3

SolarZenithAngle | 4byte(s)

Dimension(s)

Dimension Name	Granule Bounding	Dynamic	Max Index	Min Index
AlongTrack	Yes	No	771	771
CrossTrack	No	No	4121	4121

Datum

Description	Offset	Scale Factor	Unit of Measure	Range Min	Range Max	Data Type	Fill Values
Zenith angle of sun at each pixel position	0	**Unscaled**	degree			32-bit floating point	see below

Name	Value
NA_FLOAT32_FILL	-999.9
MISS_FLOAT32_F	-

								ILL	999.8
								ERR_FLOAT32_FILL	-999.5
								ELINT_FLOAT32_FILL	-999.4
								VDNE_FLOAT32_FILL	-999.3

SolarAzimuthAngle | 4byte(s)

Dimension(s)

Dimension Name	**Granule Bounding**	**Dynamic**	**Max Index**	**Min Index**
AlongTrack	Yes	No	771	771
CrossTrack	No	No	4121	4121

Datum

Description	**Offset**	**Scale Factor**	**Unit of Measure**	**Range Min**	**Range Max**	**Data Type**	**Fill Values** Name	Value
Azimuth angle of sun (measured clockwise positive from North) at each pixel position	0	**Unscaled**	degree			32-bit floating point	**Name**	**Value**
							NA_FLOAT32_FILL	-999.9
							MISS_FLOAT32_FILL	-999.8
							ERR_FLOAT32_FILL	-999.5
							ELINT_FLOAT32_FILL	-999.

										4
									VDNE_FLOAT32_FILL	-999.3

SatelliteZenithAngle — 4byte(s)

Dimension(s)

Dimension Name	Granule Bounding	Dynamic	Max Index	Min Index
AlongTrack	Yes	No	771	771
CrossTrack	No	No	4121	4121

Datum

Description	Offset	Scale Factor	Unit of Measure	Range Min	Range Max	Data Type	Fill Values (Name)	Fill Values (Value)
Zenith angle to Satellite at each pixel position	0	**Unscaled**	degree			32-bit floating point	NA_FLOAT32_FILL	-999.9
							MISS_FLOAT32_FILL	-999.8
							ERR_FLOAT32_FILL	-999.5
							ELINT_FLOAT32_FILL	-999.4
							VDNE_FLOAT32_FILL	-999.3

SatelliteAzimuthAngle — 4byte(s)

Dimension(s)

Dimension Name	Granule Bounding	Dynamic	Max Index	Min Index

Dimension Name	Granule Bounding	Dynamic	Max Index	Min Index
AlongTrack	Yes	No	771	771
CrossTrack	No	No	4121	4121

Datum

Description	**Offset**	**Scale Factor**	**Unit of Measure**	**Range Min**	**Range Max**	**Data Type**	**Fill Values**
Azimuth angle (measured clockwise positive from North) to Satellite at each pixel position	0	**Unscaled**	degree			32-bit floating point	see below

Name	**Value**
NA_FLOAT32_FILL	-999.9
MISS_FLOAT32_FILL	-999.8
ERR_FLOAT32_FILL	-999.5
ELINT_FLOAT32_FILL	-999.4
VDNE_FLOAT32_FILL	-999.3

Height	4byte(s)	**Dimension(s)**

Dimension Name	**Granule Bounding**	**Dynamic**	**Max Index**	**Min Index**
AlongTrack	Yes	No	771	771
CrossTrack	No	No	4121	4121

Datum

Description	**Offset**	**Scale Factor**	**Unit of Measure**	**Range Min**	**Range Max**	**Data Type**	**Fill Values**

Description	Offset	Scale Factor	Unit of Measure	Range Min	Range Max	Data Type	Fill Values
Ellipsoid-Geoid separation	0	**Unscaled**	meter			32-bit floating point	see below

Name	Value
NA_INT16_FILL	-999.9
MISS_INT16_FILL	-999.8
ERR_INT16_FILL	-999.5
ELINT_INT16_FILL	-999.4
VDNE_INT16_FILL	-999.3

SatelliteRange — 4byte(s)

Dimension(s)

Dimension Name	Granule Bounding	Dynamic	Max Index	Min Index
AlongTrack	Yes	No	771	771
CrossTrack	No	No	4121	4121

Datum

Description	Offset	Scale Factor	Unit of Measure	Range Min	Range Max	Data Type	Fill Values
Line of sight distance from the ellipsoid intersection to the satellite	0	**Unscaled**	meter			32-bit floating point	see below

Name	Value
NA_FLOAT32_FILL	-999.9
MISS_FLOAT32_FILL	-999.

	8
ERR_FLOAT32_FILL	-999.5
ELINT_FLOAT32_FILL	-999.4
VDNE_FLOAT32_FILL	-999.3

QF1_VIIRSGTMGEO — 1byte(s)

Dimension(s)

Dimension Name	Granule Bounding	Dynamic	Max Index	Min Index
AlongTrack	Yes	No	771	771
CrossTrack	No	No	4121	4121

Datum

Description	Offset	Scale Factor	Unit of Measure	Range Min	Range Max	Data Type	Legend Entries
DNB Pixel Mapping (GTM to SDR DNB). Indicates whether this pixel originated from the previous, current, or next granule in the SDR DNB geolocation.	0	**Unscaled**	unitless			2 bit(s)	(see legend below)
Spare	2	**Unscaled**	unitless			6 bit(s)	

Legend Entries (DNB Pixel Mapping):

Value	Meaning
0	Error
1	Previous Granule
2	Current Granule
3	Next Granule

<table>
<tr><td>QF2_VIIRSGTMGEO</td><td>1byte(s)</td><td>
Dimension(s)
<table>
<tr><th>Dimension Name</th><th>Granule Bounding</th><th>Dynamic</th><th>Max Index</th><th>Min Index</th></tr>
<tr><td>Granule</td><td>Yes</td><td>No</td><td>1</td><td>1</td></tr>
</table>
Datum
<table>
<tr><th>Description</th><th>Offset</th><th>Scale Factor</th><th>Unit of Measure</th><th>Range Min</th><th>Range Max</th><th>Data Type</th><th>Legend Entries</th></tr>
<tr><td>Solar Eclipse</td><td>0</td><td>Unscaled</td><td>unitless</td><td></td><td></td><td>1 bit(s)</td><td><table><tr><th>Value</th><th>Meaning</th></tr><tr><td>0</td><td>False</td></tr><tr><td>1</td><td>True</td></tr></table></td></tr>
<tr><td>Lunar Eclipse</td><td>1</td><td>Unscaled</td><td>unitless</td><td></td><td></td><td>1 bit(s)</td><td><table><tr><th>Value</th><th>Meaning</th></tr><tr><td>0</td><td>False</td></tr><tr><td>1</td><td>True</td></tr></table></td></tr>
<tr><td>Spare</td><td>2</td><td>Unscaled</td><td>unitless</td><td></td><td></td><td>6 bit(s)</td><td></td></tr>
</table>
</td></tr>
<tr><td>PixelRowSDR</td><td>2byte(s)</td><td>
Dimension(s)
<table>
<tr><th>Dimension Name</th><th>Granule Bounding</th><th>Dynamic</th><th>Max Index</th><th>Min Index</th></tr>
<tr><td>AlongTrack</td><td>Yes</td><td>No</td><td>771</td><td>771</td></tr>
<tr><td>CrossTrack</td><td>No</td><td>No</td><td>4121</td><td>4121</td></tr>
</table>
Datum
<table>
<tr><th>Description</th><th>Offset</th><th>Scale Factor</th><th>Unit of Measure</th><th>Range Min</th><th>Range Max</th><th>Data Type</th><th>Fill Values</th></tr>
<tr><td>Day-Night Band SDR pixel row index</td><td>0</td><td>Unscaled</td><td>unitless</td><td></td><td></td><td>unsigned 16-bit integer</td><td><table><tr><th>Name</th><th>Value</th></tr><tr><td>NA_UINT16_FILL</td><td>65535</td></tr></table></td></tr>
</table>
</td></tr>
</table>

		number that was remapped to this GTM pixel (row numbering begins with zero)						

MISS_UINT16_FILL	65534
ONBOARD_PT_UINT16_FILL	65533
ONGROUND_PT_UINT16_FILL	65532
ERR_UINT16_FILL	65531
ELINT_UINT16_FILL	65530
VDNE_UINT16_FILL	65529

PixelColSDR — 2byte(s)

Dimension(s)

Dimension Name	**Granule Bounding**	**Dynamic**	**Max Index**	**Min Index**
AlongTrack	Yes	No	771	771
CrossTrack	No	No	4121	4121

Datum

Description	**Offset**	**Scale Factor**	**Unit of Measure**	**Range Min**	**Range Max**	**Data Type**	**Fill Values**
Day-Night Band SDR pixel column index number that was remapped to this GTM	0	**Unscaled**	unitless			unsigned 16-bit integer	see below

Name	**Value**
NA_UINT16_FILL	65535
MISS_UINT16_FILL	65534
ONBOARD_PT_UINT16_FILL	65533
ONGROUND_PT_UINT16_FILL	65532

<table>
<tr><td></td><td></td><td>
<table>
<tr><td>pixel (column numbering begins with zero)</td><td></td><td></td><td></td><td></td><td></td><td></td><td>
<table>
<tr><td>ERR_UINT16_FILL</td><td>65531</td></tr>
<tr><td>ELINT_UINT16_FILL</td><td>65530</td></tr>
<tr><td>VDNE_UINT16_FILL</td><td>65529</td></tr>
<tr><td>SOUB_UINT16_FILL</td><td>65528</td></tr>
</table>
</td></tr>
</table>
</td></tr>
<tr><td>MoonIllumFraction</td><td>4byte(s)</td><td>
<table>
<tr><th colspan="5">Dimension(s)</th></tr>
<tr><th>Dimension Name</th><th>Granule Bounding</th><th>Dynamic</th><th>Max Index</th><th>Min Index</th></tr>
<tr><td>AlongTrack</td><td>Yes</td><td>No</td><td>1</td><td>1</td></tr>
</table>
<table>
<tr><th colspan="8">Datum</th></tr>
<tr><th>Description</th><th>Offset</th><th>Scale Factor</th><th>Unit of Measure</th><th>Range Min</th><th>Range Max</th><th>Data Type</th><th>Fill Values</th></tr>
<tr><td>Fraction of the moon illuminated (expressed as percent)</td><td>0</td><td>Unscaled</td><td>percent</td><td></td><td></td><td>32-bit floating point</td><td>
<table>
<tr><th>Name</th><th>Value</th></tr>
<tr><td>NA_FLOAT32_FILL</td><td>-999.9</td></tr>
<tr><td>MISS_FLOAT32_FILL</td><td>-999.8</td></tr>
<tr><td>ERR_FLOAT32_FILL</td><td>-999.5</td></tr>
<tr><td>VDNE_FLOAT32_FILL</td><td>-999.3</td></tr>
</table>
</td></tr>
</table>
</td></tr>
<tr><td>LunarZenithAngle</td><td>4byte(s)</td><td>
<table>
<tr><th colspan="5">Dimension(s)</th></tr>
<tr><th>Dimension Name</th><th>Granule Bounding</th><th>Dynamic</th><th>Max Index</th><th>Min Index</th></tr>
</table>
</td></tr>
</table>

Dimension Name	Granule Bounding	Dynamic	Max Index	Min Index
AlongTrack	Yes	No	771	771
CrossTrack	No	No	4121	4121

Datum

Description	Offset	Scale Factor	Unit of Measure	Range Min	Range Max	Data Type	Fill Values
Zenith angle of moon at each pixel position	0	**Unscaled**	degree			32-bit floating point	see below

Fill Values:

Name	Value
NA_FLOAT32_FILL	-999.9
MISS_FLOAT32_FILL	-999.8
ERR_FLOAT32_FILL	-999.5
ELINT_FLOAT32_FILL	-999.4
VDNE_FLOAT32_FILL	-999.3

LunarAzimuthAngle	4byte(s)	Dimension(s)

Dimension(s)

Dimension Name	Granule Bounding	Dynamic	Max Index	Min Index
AlongTrack	Yes	No	771	771
CrossTrack	No	No	4121	4121

Datum

Description	Offset	Scale Factor	Unit of Measure	Range Min	Range Max	Data Type	Fill Values

<table>
<tr>
<td></td>
<td></td>
<td>
<table>
<tr>
<td>Azimuth angle of moon (measured clockwise positive from North) at each pixel position</td>
<td>0</td>
<td>Unscaled</td>
<td>unitless</td>
<td></td>
<td></td>
<td>32-bit floating point</td>
<td>
<table>
<tr><th>Name</th><th>Value</th></tr>
<tr><td>NA_FLOAT32_FILL</td><td>-999.9</td></tr>
<tr><td>MISS_FLOAT32_FILL</td><td>-999.8</td></tr>
<tr><td>ERR_FLOAT32_FILL</td><td>-999.5</td></tr>
<tr><td>ELINT_FLOAT32_FILL</td><td>-999.4</td></tr>
<tr><td>VDNE_FLOAT32_FILL</td><td>-999.3</td></tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
<tr>
<td>PadByte1</td>
<td>1byte(s)</td>
<td>
<table>
<tr><th colspan="5">Dimension(s)</th></tr>
<tr><th>Dimension Name</th><th>Granule Bounding</th><th>Dynamic</th><th>Max Index</th><th>Min Index</th></tr>
<tr><td>Granule</td><td>Yes</td><td>No</td><td>4</td><td>4</td></tr>
</table>
<table>
<tr><th colspan="7">Datum</th></tr>
<tr><th>Description</th><th>Offset</th><th>Scale Factor</th><th>Unit of Measure</th><th>Range Min</th><th>Range Max</th><th>Data Type</th></tr>
<tr><td>Pad byte</td><td>0</td><td>Unscaled</td><td>unitless</td><td></td><td></td><td>unsigned 8-bit char</td></tr>
</table>
</td>
</tr>
</table>

Appendix D: Summary of questions and issues for the JPSS team to clarify or correct

1. General questions about JPSS data products

   - ***It seems that different JPSS data products may share similar XML profiles, even for different sensors.***

     Is this statement right?

   - ***We believe that the "aqu" inside the following file name strings represents the AQUA satellite that generates MODIS products. Since the VIIRS data products aim to replace MODIS data products, we assume that the following JPSS geolocation HDF5 file and the JPSS data HDF5 file obtained from NOAA's GRAVITE system are VIIRS products.***

     - ***The JPSS geolocation HDF5 example file name we used for discussion is GMODO_aqu_d20101115_t1829460_e1830558_b45401_c20101115210512589018_grav_dev.he5.***

     - ***The JPSS data HDF5 example file name is SVM03_aqu_d20101119_t1223029_e1224128_b45455_c20101119155937553093_grav_dev.he5.***

     Is our understanding right?

   - ***Are the latitude and longitude fields of VIIRS SDR data like typical MODIS swath, which are 2-D dimensional array data? The dimension names are something like "AlongTrack" and "CrossTrack".***

2. Fill Values

   JPSS XML files show that there are multiple fill values for some JPSS variables. Because we need to know if CF attributes valid_min and valid_max can be applied to these variables***, we need the JPSS team to help us to understand how fill values presented at JPSS XML files are used in the JPSS data.***

   We especially need clarification from the JPSS team about the following:

   - ***VIIRS SDR Data fields – Radiance and Reflectance***

     We need the JPSS team to clarify if the number 65530 can be used as a valid data value or if it is reserved for variables Radiance and Reflectance. Please refer to section 8.1.3.1 for details.

   - ***VIIRS SDR Data fields – ModeGran and ModeScan***

     We need the JPSS team to clarify whether 253,252 and 250 can be valid data values.

   - ***VIIRS EDR Geolocation XML – Time***

     We need the JPSS team to clarify the definition of "Range Min" and "Range Max".

3. Scale Factor and Offset

- ***VIIRS SDR Data fields – Radiance and Reflectance***

  We need the JPSS team to clarify if the equation that calculates the final data value by applying "Scale Factor" and "Offset" is

  Final_data_value = "Scale Factor" * Raw_data_value + "Offset";

4. Geolocation Field – Time

   - ***According to the description of Time, the record is the Time of the nadir point of the GTM row in IET (1/1/1958), and it represents the time of the nadir point of the GTM row. It seems that it records the time at a certain point when the satellite moves along the satellite track.***

     Is our understanding correct?

5. Geolocation Fields – Other than Latitude, Longitude, Time and Height

   - ***We suggest considering all other geolocation fields as general variables in terms of CF conventions since we cannot find the equivalent coordinate variables defined by CF conventions. We need the JPSS team to confirm this.***

6. **Dimensions**

   - ***We would like to know the meaning of some dimensions such as "Granule Boundary", "Dynamic" and "Max Index". We would especially like to understand if these dimensions have any relations with CF conventions.***

7. ***Special Characters inside Field names, Dimension names, Attribute names***

   - ***According to CF-naming conventions, variable, dimension and attribute names should begin with a letter and be composed of letters, digits, and underscores. Characters that are not letters, digits and underscores are special characters. We would like to know if all variable, dimension and attribute names in JPSS files follow CF-naming conventions.***
   - ***If some dimension or attribute names have special characters, We hope to get feedback from the JPSS team on whether to preserve the original dimension names and attribute names.***

References

1. Climate and Forecast Conventions: http://cf-pcmdi.llnl.gov/documents/cf-conventions/1.5
2. ITT. " The IDL website," http://www.ittvis.com/language/en-us/productsservices/idl.aspx
3. MathWorks. "The MATLAB website," http://www.mathworks.com/products/matlab/
4. NCL. "The NCL website," http://www.ncl.ucar.edu/
5. Unidata. "The Integrated Data Viewer(IDV) website," http://www.unidata.ucar.edu/software/idv/
6. NASA Goddard Institute for Space Studies(GISS). "Panoply website," http://www.giss.nasa.gov/tools/panoply/
7. Space Science Engineering Center (SSEC). " McIDAS-V website," http://www.ssec.wisc.edu/mcidas/software/v/
8. Unidata, "netCDF-Java website," http://www.unidata.ucar.edu/software/netcdf-java
9. Space Science Engineering Center (SSEC). "VisAD website," http://www.ssec.wisc.edu/~billh/visad.html
10. Center for Ocean-Land-Atmosphere Studies(COLA). "Grid Analysis and Display System (GrADS) website," http://www.iges.org/grads/
11. Pacific Marine Environmental Laboratory (PMEL). "Ferret," http://ferret.pmel.noaa.gov/Ferret/
12. OPeNDAP, "OPeNDAP website," http://opendap.org