Global Hydrology Resource Center (GHRC)
at the
University of Alabama in Huntsville

SSM/I and SSMIS Data in NetCDF

User's Guide

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

The Remote Sensing Systems (RSS) Special Sensor Microwave Imager (SSM/I and SSMIS) binary format data have been reformatted to network Common Data Form (netCDF) by the Global Hydrology Resource Center (GHRC), a NASA Earth science data center managed by the University of Alabama in Huntsville.

Some of the advantages of using netCDF are:

1. The netCDF 4 format uses HDF5 as its underlying format, so all of the advantages of the previous format are retained, namely

    a. The format allows us to package the navigation and science data into a single file, making the files easier to distribute.

    b. netCDF 4 arrays can be *internally* compressed. This saves considerable disk space. This compression is *invisible* to the data user, as netCDF 4 takes care of the decompression on the fly when the data are read.

2. A netCDF file can be self-describing, including information about the data contained within via global attributes and variable-specific attributes (such as valid value ranges, scales and offsets to be applied to the values read, flag values, etc.).

3. A netCDF file is portable, being readable on any machine for which the API and tools are available, even if the machines have different ways of storing integers, characters, and floating-point numbers. At the time of this writing, netCDF is supported on Linux, Windows, Mac OS X, IRIX/IRIX64, Solaris, AIX, and HPUX.

4. The netCDF API is available for several commonly-used programming languages, including C/C++, Java, FORTRAN, and others.

Chapter 2 briefly describes the netCDF data format.

Chapter 3 describes the SSMI/SSMIS data in netCDF format. You will need to know this information in order to properly and efficiently access the data in the file.

Chapter 4 lists the additional software requirements for using netCDF formatted files. It also describes the compilation and linking options needed to build your application program.

Chapter 5 is an example program in C which demonstrates how to open and discover the information present in an SSM/I or SSMIS netCDF file.

Chapter 6 contains links to software packages that can be used to manipulate netCDF files.

If you require assistance, please contact the GHRC User Services Office:

GHRC User Services
320 Sparkman Drive
Huntsville, AL 35806
Phone: (256) 961-7932
E-mail: ghrcdaac@itsc.uah.edu

CHAPTER 2

NETCDF FORMAT

WHAT IS NETCDF?

netCDF files can contain objects such as scientific data sets, annotations, and raster images, all of which are handled in a system- and hardware-independent manner. This relieves the data provider and the data user of the burden of reformatting binary data for different hardware types. For example, a 32-bit integer is stored on SGI or Sun hardware in "Big Endian" format; that is, the bytes are ordered with the most-significant byte on the left and the least-significant byte on the right. The same 32-bit integer is stored on a PC in "Little Endian" format with the most-significant byte on the right and the least-significant byte on the left. If a file were written on the Big Endian machine and subsequently read by a Little Endian machine, the data bytes would be "backwards", which would require the Little Endian machine to perform byte-swapping. netCDF takes care of this for the end-user.

netCDF provides an additional "layer of abstraction" providing high-level objects called variables and attributes. Variables correspond to the most common array structures used by Earth Observing Systems data. A single file can contain any number or combination of variables of varying dimensions, but variables that are arrays of dimensions 1, 2, and 3 are the only types used for SSM/I and SSMIS datasets. Attributes can be global or associated with a specific variable. Global attributes provide information relevant to the file as a whole while variable attributes provide information relevant to one specific variable.

Variables holding the grid objects consist of rectangular grids of data points. The objects are either two or three dimensions. Grids that are averages of three-day, weekly, or monthly time periods are two dimensions while grids of data for one day are three dimensions, with the third dimension serving to separate the grids for ascending and descending passes. The two-dimensional grids (and the two slices of the three-dimensional grids) represent maps in a simple equirectangular latitude/longitude projection.

Data from a variable can be geolocated and, when applicable, temporally located. This is done by providing 2 additional one-dimensional variables containing the latitude and longitude values for the grid, as well as an additional 2-dimensional variable, when applicable, with the time values, which temporally locate each grid point.

OBJECT HIERARCHY

In netCDF all elements are part of a hierarchy as follows: A *file* can consist of one or more *groups*, each of which can consist of variables and attributes as well as other groups. Variables and attributes are only visible within the group in which they are defined. However, the SSM/I and SSMIS datasets are relatively simple and only use one group (called the "root" group) to contain variables. Attributes not associated with a variable and placed in a group are global to that group. Attributes global to the root group are used for "global attributes" containing metadata specific for the entire file. Variables can also have attributes, such as a description, the valid value range, and the scale/offset to be applied to the data values read to get the real values.

ACCESSING OBJECTS AND FIELDS

In netCDF, when a file is opened, the ID for the root group of that file is returned, providing a starting point for navigating the file. The netCDF calls take an ID, which determines where the call will take its action. With a handle to a group, the attributes, dimensions, and variables can be accessed. There are two ways to proceed: The more general way is to query the root group to discover the contents of the group, and act accordingly, depending what is discovered. However, it is often the case (as it is with the SSM/I and SSMIS data) that fixed names have been chosen for the various elements, so it is possible to directly access elements by their predefined names. However, if there is no element with a given name, a call to read it in will result in an error.

DIMENSIONS

The SSM/I and SSMIS data contain either 2 or 3 fixed dimensions: Daily files have 3 (for time, latitude, and longitude) and the files for 3-day, weekly, and monthly averages have 2 (for latitude and longitude).

netCDF uses *named dimensions*. For example, when creating a variable, the code must have already defined the names and values of the dimensions. The three dimensions used in SSM/I and SSMIS files are "Latitude", "Longitude", and "Time". Since all data grids (or slices) in these datasets are equirectangular projections with half-degree resolution, all such grids are 720 by 1440. The "Latitude" dimension has value 720 and the "Longitude" dimension has value 1440. For daily files having separate grids for both ascending and descending passes, the "Time" dimension of size 2 is used for the third dimension. Monthly, weekly, and 3-day average files do not have this third dimension.

DATA TYPES

As in HDF-EOS, netCDF uses *named data types* for all field data. The underlying netCDF library takes care of making all of the data types portable, so any data type written on one system can be read by another. SSM/I and SSMIS netCDF files use only 3 of the many types: NC_CHAR (character string), NC_FLOAT (32-bit floating point), and NC_SHORT (signed 16-bit integer).

CHAPTER 3

SSM/I DATA IN NETCDF

The SSM/I and SSMIS RSS version 7 data have been reformatted to netCDF Version 4 format (which is based on HDF Version 5).

FILE NAMING CONVENTIONS

All data is in grid format. There are daily files with 2 grids: one for the ascending and one for the descending passes. There are three files with grids for 3-day, weekly, and monthly averages. Note that the file name extension is now ".nc" in all cases to denote a netCDF formatted file. All files contain grids for 10-meter surface wind speed, columnar water vapor, columnar cloud liquid water, and rain rate.

In all filenames below, the following strings are interpreted as follows:

nn – Is the satellite number (08-18)
yyyy – Is the 4-digit year
mm – Is the 2-digit month
dd – Is the 2-digit day of month

Daily files consist of a single file per day.

f*nn_yyyymmddv7.nc*

3-Day average files consist of a single file with the date being the ending day of the 3-day period.

f*nn_yyyymmddv7_d3d.nc*

Weekly average files consist of a single file with the date being the ending day of the one-week period.

f*nn_yyyymmddv7_wk.nc*

Monthly average files consist of a single file with the date being the month (note that the day is not present).

f*nn_yyyymmv7_d3d.nc*

DATA

SSM/I and SSMIS data are stored as grids with 4 products per file, one file per day.

Within the netCDF file are dimensions; geo, time, and data variables; and attributes.

Dimensions

Dimensions are named and will always consist of the following set:

"Latitude" is the number of horizontal lines in the data grids (always 720)
"Longitude" is the number of vertical lines in the data grids (always 1440)
"Time" is the number of passes (daily files only, always 2 if present)

Geo and Time Variables

Variables with the fixed names “Latitude” and “Longitude” are defined in all SSM/I netCDF data files. The variable “SST_DTime” is defined only in daily files.

“Latitude” and “Longitude” are stored as one-dimensional arrays of 32-bit floating-point values with dimensions named “latitude” and “longitude”. Valid latitude values range from –89.875 to +89.875 (South pole to North pole “pixel centers”), and valid longitude values range from +0.125 (just east of Dateline) eastward to 359.875 (just west of Dateline).

“SST_DTime” is stored as a two-dimensional array (with dimensions “latitude” by “longitude”) of 16-bit signed integers. The scale value of 0.1 is to be applied to the values to produce a time value representing the number of hours since the beginning of the day, GMT, that the daily file represents. The valid range is 0.0 to 24.0 m/s.

Data Fields

The following data fields are defined.

“10-Meter Surface Wind Speed” – A two-dimensional array (“latitude” by “longitude”) of 16-bit signed integers. A scale value of 0.2 is to be applied to produce the real values. The valid range of real values is 0.0 to 50.0.

“Columnar Water Vapor” – A two-dimensional array (“latitude” by “longitude”) of 16-bit signed integers. A scale value of 0.3 is to be applied to produce the real values. The valid range of real values is 0.0 to 75.0 kg/m^2.

“Columnar Cloud Liquid Water” – A two-dimensional array (“latitude” by “longitude”) of 16-bit signed integers. A scale value of 0.01, then an offset of -0.05 is to be applied to produce the real values. The valid range of real values is -0.05 to 2.45 kg/m^2.

“Rain Rate” – A two dimensional array (“latitude” by “longitude”) of 16-bit signed integers containing the data. A scale value of 0.1 is to be applied to produce the real values. The valid range of real values is 0.0 to 25.0 mm/hr.

We have retained the data values, scaling, and offsets from the RSS binary files in the translation to netCDF so that a direct copy of the values stored in the RSS binary files for the 4 data fields could be made without changing any values.

netCDF 4 supports data compression, but requires that the data be tiled into “chunks”. All two-dimensional arrays are chunked using 2x90x90 chunk sizes for daily files or 90x90 for the 3-day, weekly, and monthly average files, and compressed. Chunking and compression in netCDF are invisible to the end-user since fields are unchunked and uncompressed as needed as they are read. Tests were performed to determine a reasonable chunk size by doing experiments with square chunk sizes with dimensions that divide evenly into 720 (the size of the “latitude” dimension). Chunk sizes that are too small hurt the compression ratio and make the data slower to read while chunk sizes that are too large could make subsetting inefficient. The 90x90 choice was at the “sweet spot” where going larger had a negligible effect on the compression and the size gives 128 “tiles” of data (8x16) to possibly help with subsetting.

Global Attributes

netCDF global attributes contain metadata about the file. The following attributes (all character strings) are defined in the SSM/I and SSMIS data files:

“Title” – The name of the dataset.

“Institution” – The institution(s) involved in producing the data files.

“Source” – The source of the data in the files.

"History" – History information, including provenance (providing documentation of where and when the files were produced).

"References" – A URL to documentation containing a description of the data.

"Comment" – Notes about the data, containing a more lengthy description of the history, source, and data file contents.

"Scale" – A summary of the scale values that are to be applied to each variable within the file to produce the real values.

"Value" – A summary of special values used that are to be interpreted in a special way. For SSM/I and SSMIS, such values are outside the valid range and are flag values.

"SatID" – The satellite ID, of the form "DMSP-Fnn", where nn is the satellite number.

"SensorID" – The sensor ID, which is "SSM/I" or "SSMIS".

"identifier_product_DOI" – the Digital Object Identifier (DOI) for the data file.

"PassDirection" – summary of the values used for pass direction (1 for ascending, 2 for descending).

"NumberOfPasses" – the number of passes stored in the data file (2 for daily files, 1 for others).

"ChunkSize" – the dimension of the square tile size used. 90 was chosen, resulting in a chunk size of 2x90x90 for daily files with 2 passes, and 90x90 for other files.

"Conventions" – the CF (Climate and Forecast) convention version number for the CF conventions used for the data file.

CHAPTER 4

BUILDING APPLICATIONS

SOFTWARE REQUIRED

netCDF

The netCDF library is used to read or write netCDF files. It is available for several languages, including Java, C++, C, FORTRAN, and others. At the time of this writing, the netCDF-Java library is at version 4.2 and the C/C++ libraries are also at version 4.2. Version 4.2 or later is recommended, as earlier versions may not fully support the compression and chunking options used in the SSM/I and SSMIS files. The netCDF and netCDF-Java libraries can be downloaded free of charge from NCSA at http://www.unidata.ucar.edu/downloads/netcdf/. Java JAR files are available that have the dependencies in-place, making a project setup much easier. For other languages, other libraries must be obtained in binary form and installed, or compiled from source code.

HDF 5

Since netCDF is based on HDF 5, the HDF version 5 library is required. At the time of this writing, HDF5-1.8.9 is the latest version. This version or later is recommended. HDF can be downloaded free of charge from NCSA at http://www.hdfgroup.org/HDF5/release/obtain5.html. Note that NSCA provides pre-compiled binaries for many platforms or source code if you wish to go through the trouble of customizing a library for your system.

SZIP

HDF 5 requires the SZIP library to perform compression. SZIP can be downloaded free of charge from the NCSA site at http://www.hdfgroup.org/HDF5/release/obtain5.html. NCSA provides pre-compiled binaries or source code for this package.

ZLIB

HDF 5 requires the ZLIB library to perform compression. ZLIB can be downloaded free of charge from the NCSA site at http://www.hdfgroup.org/HDF5/release/obtain5.html. NCSA provides pre-compiled binaries or source code for this package.

JPEG

HDF 5 requires the JPEG library to perform compression. JPEG can be downloaded free of charge from the NCSA site at http://www.hdfgroup.org/HDF5/release/obtain5.html. NCSA provides pre-compiled binaries or source code for this package.

COMPILING PROGRAMS

The native C compiler is recommended for use on all systems. On many systems, this is the gcc compiler or possibly the llvm compiler. On SGI/Irix equipment or other older UNIX machines, each vendor supplied their own compiler, which is usually called “cc”. On many machines having their own C compiler, the gcc or llvm compilers may be available and may be more up-to-date, especially if C++ functionality is needed. At the time of this writing, 32-bit vs. 64-bit is an issue and some platforms are in a transitional state. If you are not compiling from the source, please be sure to obtain the correct versions of libraries and binaries for your platform. In general, an executable must be compiled 32-bit or 64-bit and all dependencies must be one or the other and cannot be mixed on most platforms. If compiling from source, the configure script should detect your platform and adjust accordingly. If any dependencies are 32-bit only, the source code must be compiled with the 32-bit option (“-m32” on gcc, “-n32” on some other compilers, so it is important to check your compiler’s documentation).

C or C++ users must include the following statement to make the symbols and functions of the HDF and HDF-EOS library available:

```
#include <netcdf.h>
```

This file must be in the include path on the compilation line. This is done by including a "-I" (eye) option to define the paths to each library. For example,

```
gcc –I path-to-netcdf include –I path-to-hdf5/include myprog.c –c –o myprog.o
```

Unless the application program directly accesses the routines in the HDF5, SZIP, ZLIB, and/or JPEG libraries, it is not necessary to define the path to their include files.

LINKING PROGRAMS

Any application that uses HDF-EOS must link with *all* of the aforementioned library packages. The paths to the libraries must be defined using a –L option for each library. Since HDF-EOS requires HDF, and HDF requires the SZIP, ZLIB, and JPEG libraries, this can result in a fairly lengthy link command. For example,

```
ld –Lpath-to-netcdf/lib –L path-to-hdf5/lib –L path-to-szip/lib –L path-to-zlib/lib –L path-to-jpeg/lib
        myprog.o -lnetcdf –lhdfeos–lmfhdf –ldf –ljpeg –lsz –lz –lm
        -o myprog
```

If your netCDF library is built with hdf5 included, the compile line can be shorter. For example,

```
ld –Lpath-to-netcdf/lib myprog.o –lnetcdf
```

Note that there are many ways that the libraries can be setup on a machine. One way is to have them in a single place (/usr/local/include and /usr/local/lib, for example) and that can make things simple as well:

```
gcc -I/usr/local/include -L/usr/local/lib -o ReadNetCDF ReadNetCDF.c –lnetcdf
```

How the libraries are setup is determined by the system administrator, but will affect the necessary directories specified on the command-line when compiling and linking.

RUNNING PROGRAMS

On most UNIX style environments, a program is run by typing the name on the command-line. In recent years, it is common not to have the current directory in the path, so "./" must be placed in front of the name when running in the current directory. A path to the executable must be supplied when running from another directory.

```
./myprog
```

CHAPTER 5

SAMPLE APPLICATIONS

READER

A sample C read program is available at ftp://ghrc.nsstc.nasa.gov/pub/doc/ssmi_netcdf/ReadNetCDF.c .

The ReadnetCDF.c code requires that the netCDF.h file be in the users include path. Under UNIX-type environments, this is done by providing the -Ipath_to_netcdf_include option to the C compiler if your sysadmin hasn't installed it on the system in a "standard place".

The purpose of the sample C program is to read the contents of a NetCDF file's "root group" and report the contents.

It prints the following information:

1. Global Attributes

   Each is displayed on a separate line in the form

   AttributeName : "AttributeValue"

2. Dimensions

   Each is displayed on a separate line in the form

   DimensionName : IntegerSize

3. Variables

   A section is displayed for each variable, including its name, type code, number of dimensions, and number of attributes. Variables can have attributes, so each attrribute is displayed in the same manner as the global attributes above. The dimensions are then listed. Finally, if the variable is signed short int or 32-bit float (the only variable types in SSMIS and SSM/I files), a value-summary is printed, consisting of the first few and last few array values. The variable information section looks like the following:

   VariableName : type = TypeCode, dimensions = NumDimensions, number of attributes = NumAttributes

   variable attributes :

   attribute_name_0 : "attribute_value_0"

   attribute_name_1 : "attribute_value_1"

   attribute_name_2 : "attribute_value_2"

   ...

   attribute_name_N-1 : "attribute_value_N-1"

dimensions : dim_size_0 x dim_size_1 x ... x dim_size_N-1 (NNN elements total)

value_summary : [if applicable]

raw TYPE values (value0 value1 value2 value3 value4 ... valueN-3 valueN-2 valueN-1)

values scaled/offsetted [if applicable] (SValue0 SValue1 SValue2 SValue3 SValue4 ... SValueN-3 SValueN-2 SValueN-1)

This program can be easily transformed into a program that makes use of the variable arrays by modifying the print_variables function and using the arrays as needed rather than simply printing a summary of the values.

To use this program with a wider range of netCDF files containing values of other types, simply add if blocks in the print_variables to check for more type ids than NC_SHORT and NC_FLOAT and handle them appropriately.

CHAPTER 6

APPLICATION PACKAGES

There are a number of freeware packages that can be downloaded to examine and manipulate netCDF files. Many of these can be found at the HDF-EOS web site, http://hdfeos.org/software.

Panoply is a cross-platform Java application which plots geo-gridded arrays from netCDF datasets. There are versions specific for Mac OS X and Windows, as well as generic versions for other platforms that support Java 6. Panoply is available at http://www.giss.nasa.gov/tools/panoply/ .

The Integrated Data Viewer (IDV) is a Java-based software framework for analyzing and visualizing geoscience data. The IDV is developed at the Unidata Program Center (UPC), part of the University Corporation for Atmospheric Research (UCAR), Boulder, Colorado, which is funded by the National Science Foundation. The software is freely available under the terms of the GNU Lesser General Public License, and is available at http://www.unidata.ucar.edu/software/idv/.