GSE 760–S01

Advanced Methods in Geospatial Modeling:

Computation for Remote Sensing Analysis and Product Generation

xiaoyang Zhang, 3/19/2021

Class Schedule

Date	Lecture (Friday)	Date	Exercises (Friday)
Jan 15	**Lecture 1**: Course overview and introduction to remote sensing processing on Linux system	Jan 15	Lab assignment 1- Linux system setup
Jan 22	**Lecture 2**: Getting start with Linux system	Jan 25	Lab assignment 2 - Command line syntax
Jan 29	**Lecture 3:** Linux files and file utilities	Jan 29	Lab assignment 3 – File utilities
Feb 5	**Lecture 4:** File system and processes	Feb 5	Lab assignment 4 – File system and processes
Feb 12	**Lecture 5:** Shell scripting	Feb 12	Lab assignment 5- Shell scripting
Feb 19	**Lecture 6:** Perl scripting (1)	Feb 19	Lab assignment 6- Perl scripting
Feb 26	**Lecture 7:** Perl scripting (2)	Feb 26	Lab assignment 7- Perl scripting
March 5	**Lecture 8:** Python scripting	March 5	Midterm Exam
March 12	**Spring Break**		**Spring Break**
March 19	**Lecture 9:** Satellite data and file format	March 19	Lab assignment 8- Python scripting
March 26	**Lecture 10:** Satellite data processing	March 26	Lab assignment 9- Satellite data processing
April 2	**No class/Easter Recess**	April 2	**No class/Easter Recess**
April 9	**Lecture 11:** Operational product generation	April 9	Lab assignment 10- Programing for product generation
April 16	**Lecture 12:** Software and product documentation	April 16	Project work overview
April 23	Work on projects	April 23	Work on projects
April 30	Work on projects	April 30	Work on projects
May 7	**Lecture 13:** Final presentation		

Final Exam: We will schedule final presentations during the final exam period.

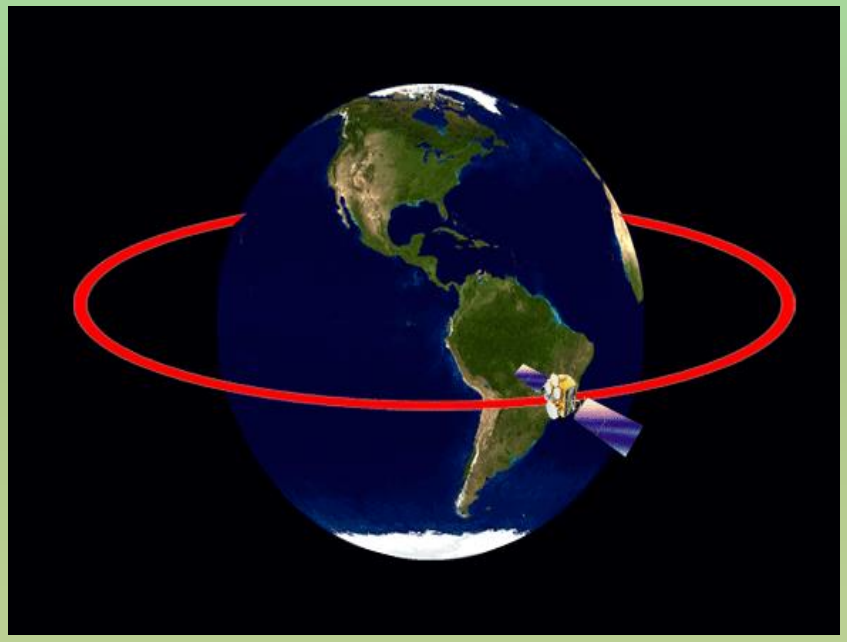
Note: Recommended to readings to accompany each chapter will be assigned on the class D2L site.

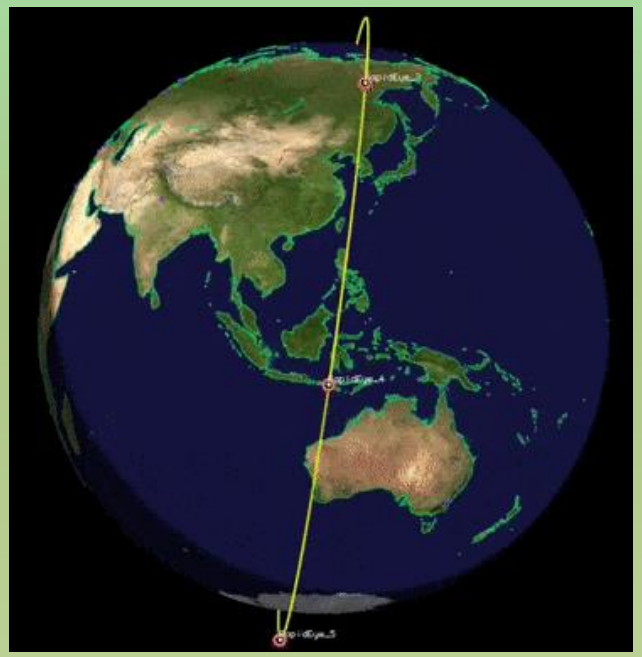
Earth Observation Satellite Data and Format

Earth Observation Satellites

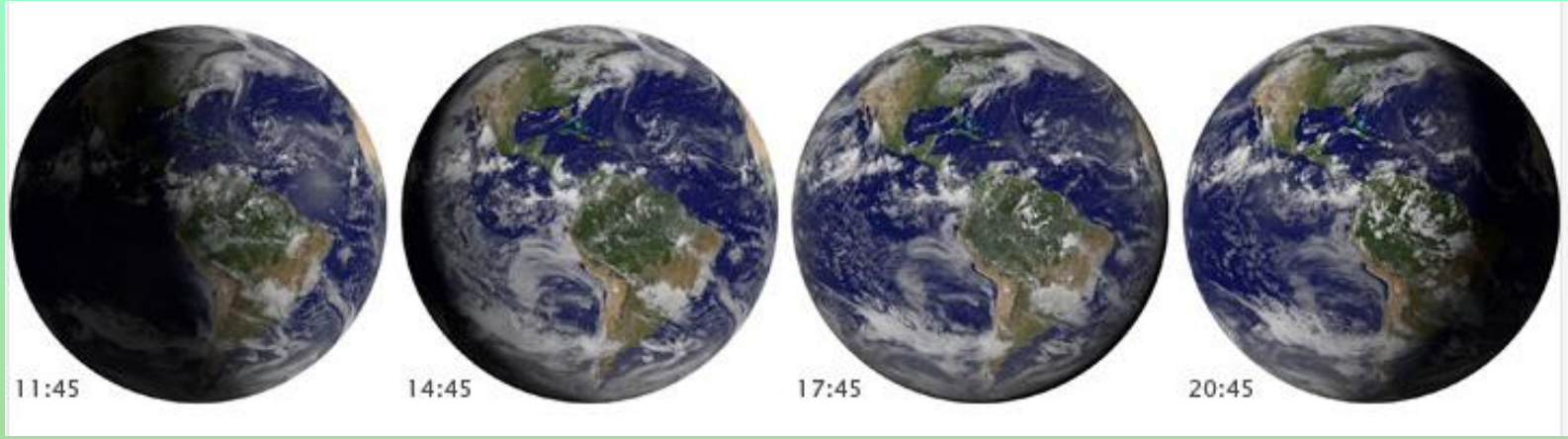
Earth observation satellites are satellites specifically designed for Earth observation from orbit, such as environmental monitoring, meteorology, map making etc.

Geostationary & Polar Orbiting satellites

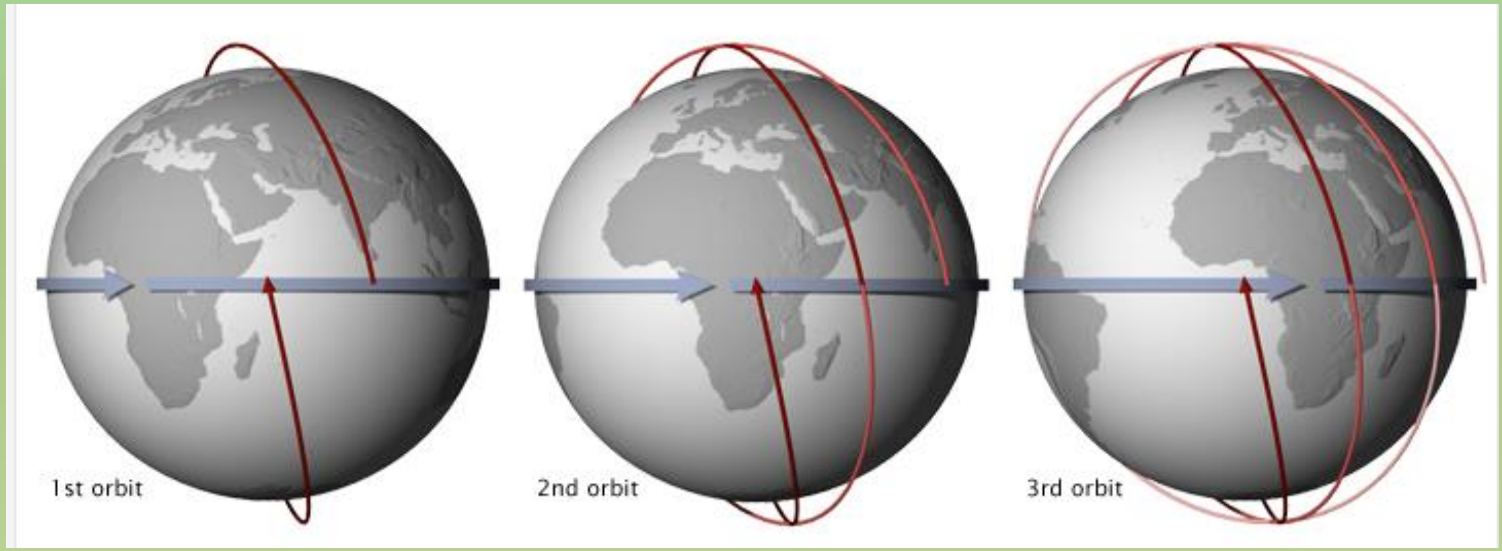




Observations from Leo and Geo Satellites



GEO (geostationary) satellites rotate with the Earth directly above the equator, continuously staying above the same spot.



LEO satellites, a Sun-synchronous orbit crosses over the equator at approximately the same local time each day (and night). This orbit allows consistent scientific observations with the angle between the Sun and the Earth's surface remaining relatively constant.

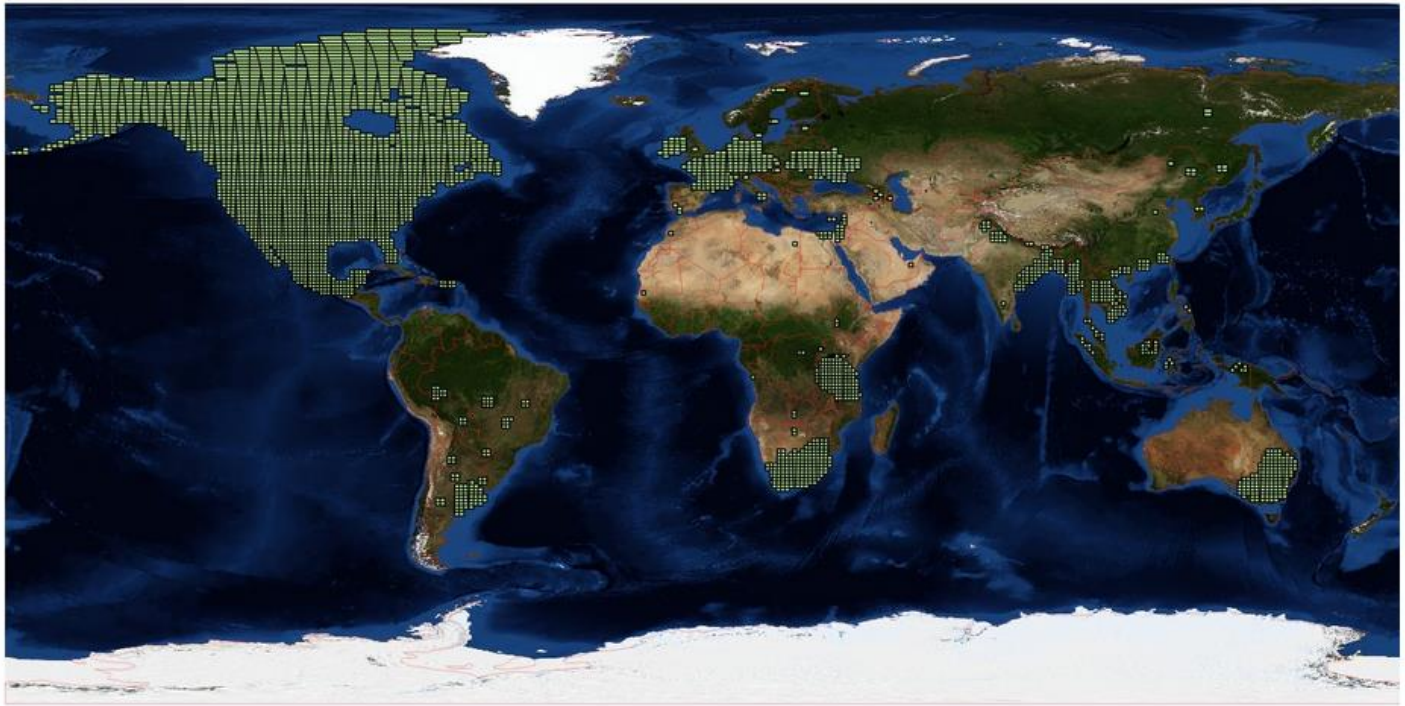
Harmonized Landsat Sentinel-2

Home | Algorithms | Products Description | Test Sites | Data | QA | Documents

Test Sites » Map

- Table
- Map
- Data

The locations of all tiles can be viewed in this

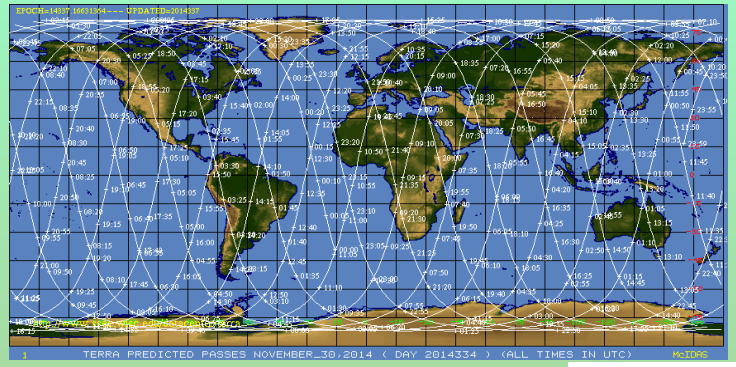


NASA Official: Jeffrey Masek

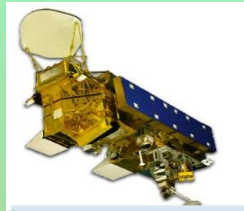
Last Updated: September 19, 2018

Global Observations from Polar Orbiting Satellites

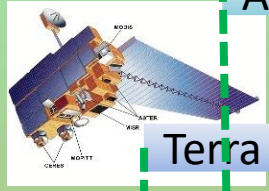
Moderate resolution (250-1000m)



MODIS - MODerate Resolution Imaging Spectroradiometer



Aqua MODIS



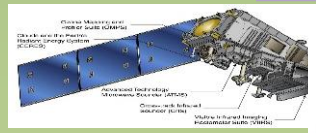
Terra MODIS

Advanced Very High Resolution Radiometer (*AVHRR*)

NOAA AVHRR

1980 1985 1990 1995 2000 2005 2010 2015 2020 2025 2030

Suomi-NPP VIIRS



NOAA20 VIIRS

NOAA 21

NOAA 22

NOAA 23

VIIRS - Visible Infrared Imaging Radiometer Suite

Suomi-NPP: Suomi-National Polar-orbiting Partnership

JPSS: Joint Polar Satellite System (NOAA 20 ---)

High Resolution Commercial Satellites

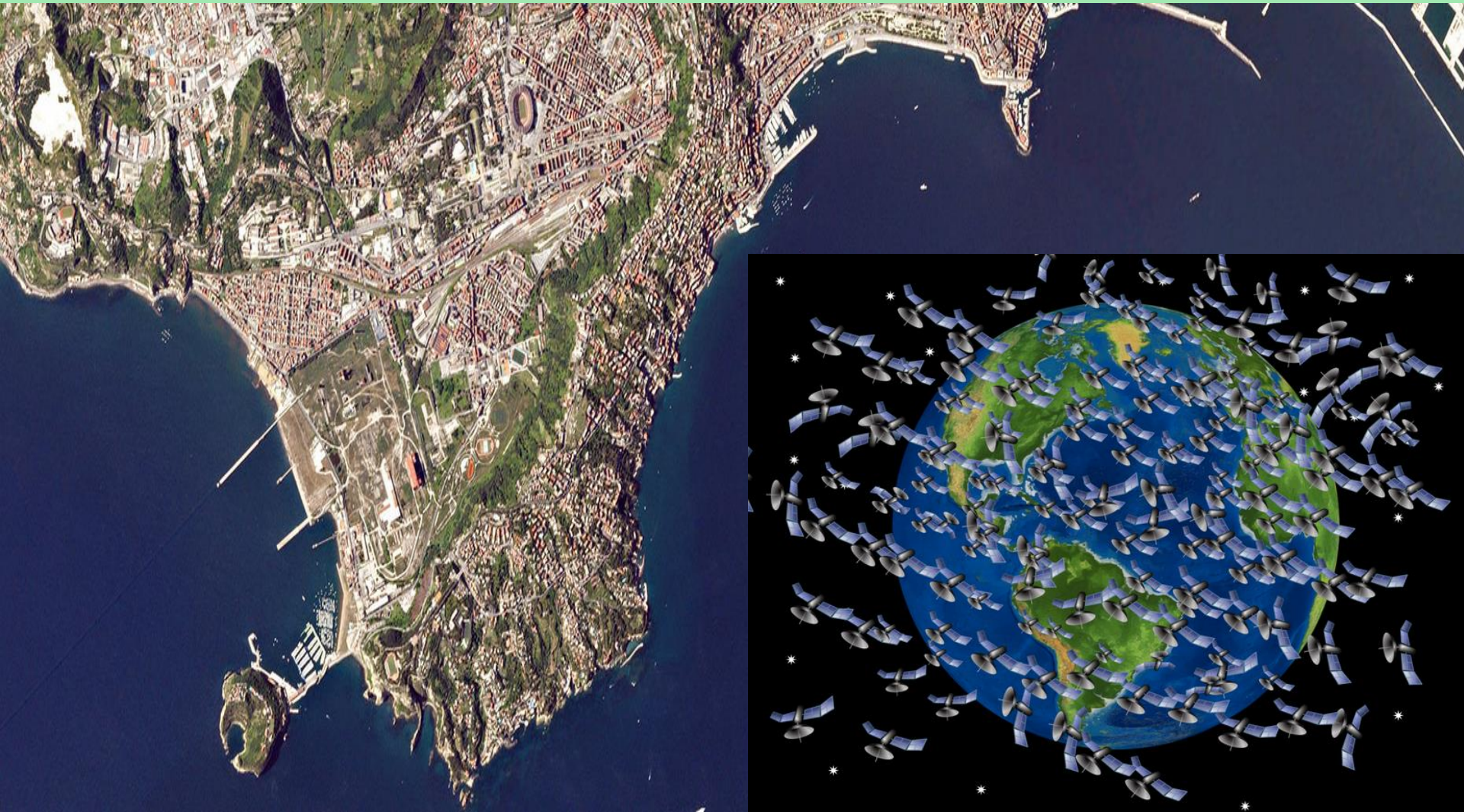
Disaster Monitoring Constellation
(9 different satellites as of 2013, 5 operating)
PIeiades(2 operating satellites)
EROS A & B
FORMOSAT-2
IKONOS
QuickBird
Rapideye
(5 identical satellites as of 2012)
SPOT (6 satellites, 2 operating)
COSMO-SkyMed
(4 identical satellites as of 2012)
WorldView-1
WorldView-2
GeoEye-1

http://en.wikipedia.org/wiki/

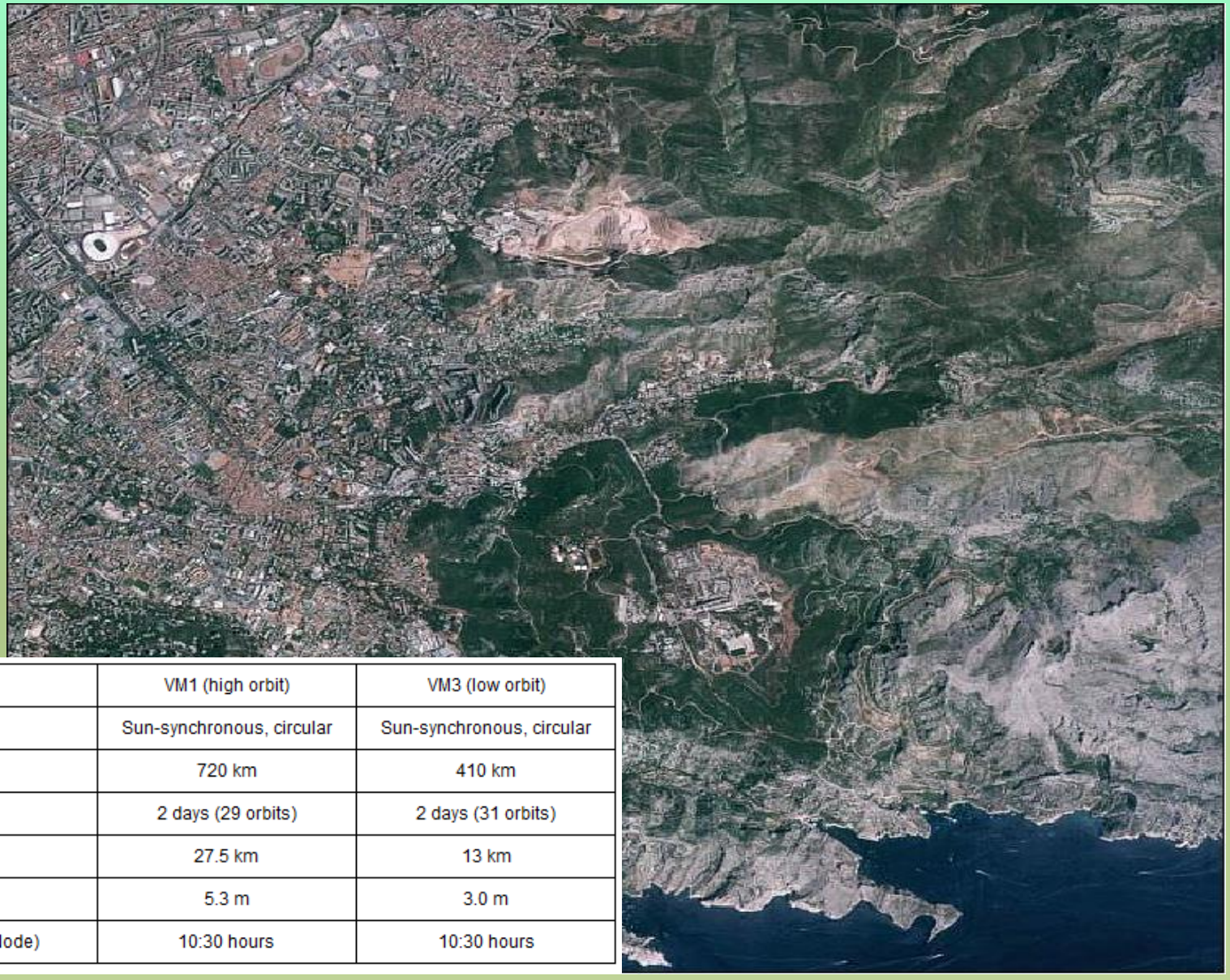
IKONOS image

Planet Monitoring

With 130+ satellites in orbit, Planet is able image anywhere on Earth daily at 3 meter and 72 centimeter resolution. Monitor your areas of interest, discover patterns, and deliver timely insights.



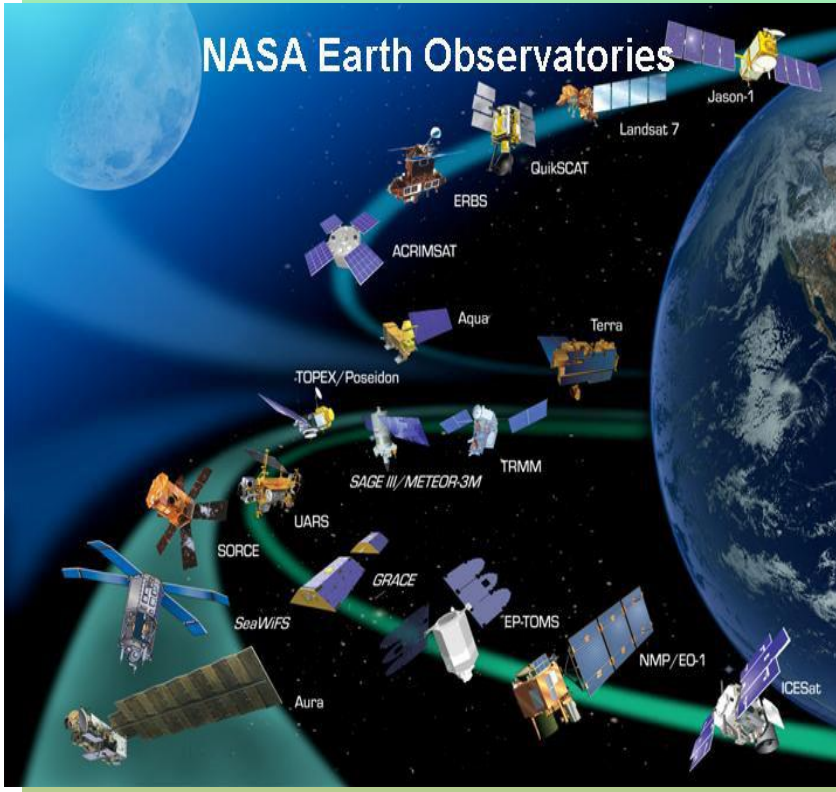
VENµS (Vegetation and Environment monitoring on a New MicroSatellite)



Parameter / VENµS mission phase	VM1 (high orbit)	VM3 (low orbit)
Orbit type	Sun-synchronous, circular	Sun-synchronous, circular
Altitude	720 km	410 km
Revisit time	2 days (29 orbits)	2 days (31 orbits)
Swath width	27.5 km	13 km
Imaging resolution	5.3 m	3.0 m
LTDN (Local Time on Descending Node)	10:30 hours	10:30 hours

NASA Earth Observing System

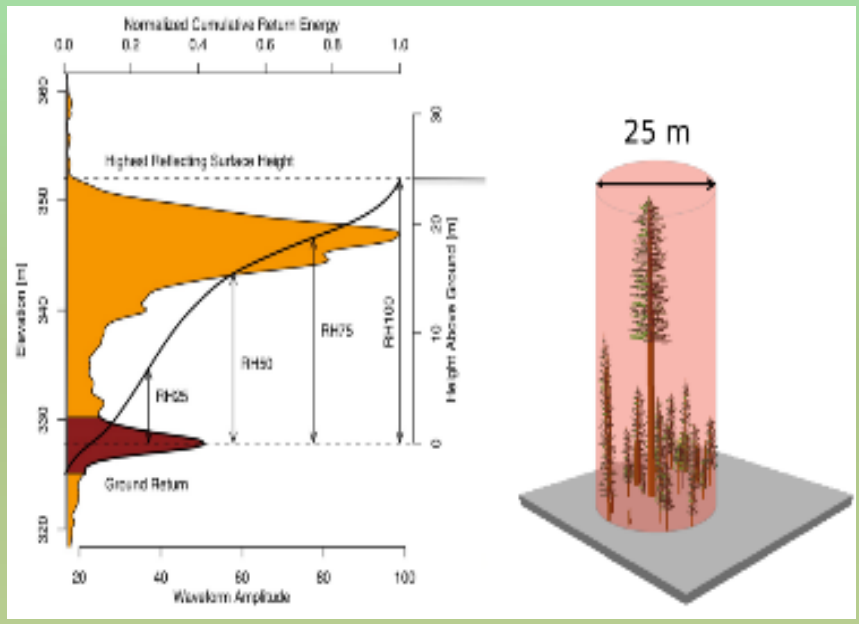
Satellite	Launch Date	Launch Site	Agency
SeaWiFS	1 August 1997		
TRMM	27 November 1997	Tanegashima	NASA / JAXA
Landsat 7	15 April 1999	Vandenberg	NASA
QuikSCAT	19 June 1999	Vandenberg	NASA / JPL
Terra	18 December 1999	Vandenberg	multiple
ACRIMSAT	20 December 1999	Vandenberg	NASA / JPL
NMP/EO-1	21 November 2000		NASA / GSFC
Jason 1	7 December 2001		NASA / CNES
Meteor 3M-1/Sage III	10 December 2001	Baikonur	
GRACE	17 March 2002	Plesetsk Cosmodrome	NASA / DLR
Aqua	4 May 2002	Vandenberg	multiple
ADEOS II (Midori II)	12 December 2002	Tanegashima	
ICESat	12 January 2003	Vandenberg	NASA
SORCE	25 January 2003	Cape Canaveral	NASA
Aura	16 July 2004	Vandenberg	multiple
CloudSat	28 April 2006	Vandenberg	NASA
CALIPSO			
Hydros	June 2006		
NPOESS	TBD		NASA / NOAA
OCO	23 February 2009	Failed to reach orbit	NASA / NOAA
Aquarius	10 June 2011	Vandenberg	NASA / CONAE
NMP/EO-3			
Landsat 8	11 February 2013	Vandenberg	NASA / USGS



xiaoyang Zhang, 3/19/2021

Global Ecosystem Dynamics Investigation (GEDI)

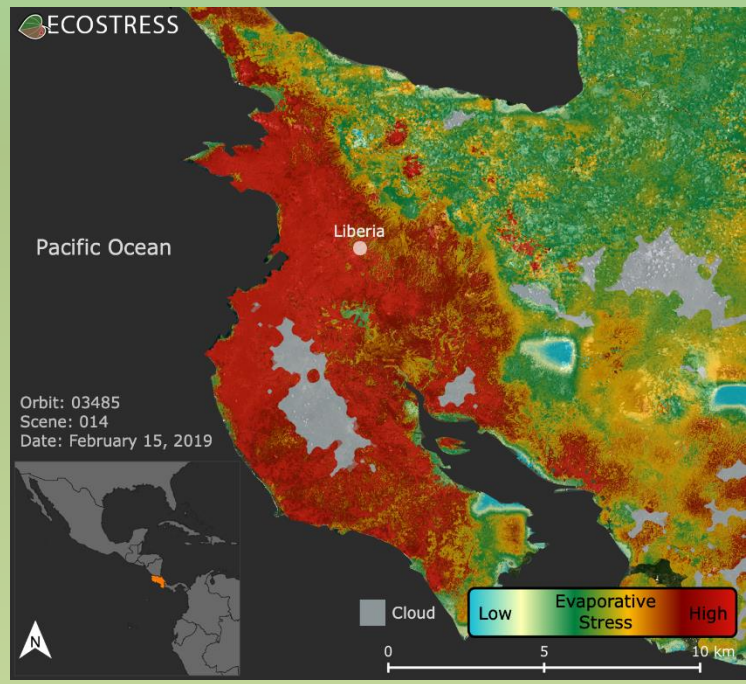
- GEDI is a full-waveform lidar instrument that makes detailed measurements of the 3D structure of the Earth's surface.
- Lidar is an active remote sensing technology (the laser version of radar) which uses pulses of laser light to measure 3D structure.
- The light is reflected by the ground, vegetation and any clouds and is then collected by GEDI's telescope.
- The photons are then directed towards detectors, converting the brightness of the light to an electronic voltage which is then recorded as a function of time in 1 ns (15 cm) intervals.
- Time is converted to range (a distance) by multiplying by the speed of light. The recorded voltage as a function of range is the full-waveform.



GEDI was deployed on the International Space Station (ISS) in December 2018 for a two-year mission

ECOsystem Spaceborne Thermal Radiometer Experiment on Space Station (ECOSTRESS)

- Prototype HyspIRI Thermal Infrared Radiometer (PHyTIR) on the International Space Station (ISS) and using it to gather the measurements needed to address the science goals and objectives.
- From the ISS, PHyTIR provides data with 38-m in-track by 69-m cross-track spatial resolution (science requirement is 100 m) and predicted temperature sensitivity of ≤0.1 K (science requirement is 0.3 K).
- The ISS orbit allows excellent coverage of the selected targets including diurnal coverage.
- Launched on June 29, 2018
- CONUS, twelve 1,000 x 1,000 km key climate zones and twenty-five Fluxnet sites for all opportunities.
- On average 1 hour of science data per day



EcostressBrightnessTemperatureV01.h5

xiaoyang Zhang, 3/19/2021

Satellite dataset Metadata

Grid information:
Position of origin
Grid reference (origin corner, origin coordinate)
Grid framework (grid dimensions, grid spacing, units, offset structure)

Data Structure:
Content values
Content structure

General location:
Bounding Box (min point and max point)

Projection
Central meridian
False easting and false northing

Time Object Duration:
Start time and length of time)

Satellite Data file = metadata+ digital satellite images

Table 1: AVHRR Metadata

Grid	
No. lines	2889
No. samples	4587
No. bands	10
bands	AVHRR channels 1-5, NDVI, satellite zenith, solar zenith, relative azimuth, pixel source date
units of measure	meters
pixel size	1000 (meters)

Spatial Registration		
map projection	Lambert Azimuthal	
lat/lon of center	100 West	45 North
false easting/northing	0	0
nw corner	-2050000	752000
Bounding box:		
lower left lat/lon	-119.9722899	23.5837576
upper left lat/lon	-128.5300591	48.4030555
upper right lat/lon	-65.3946489	46.7048989
lower right lat/lon	-75.4163527	22.4793919

Temporal Information	
Begin date	4-Jan-1991
End date	17-Jan-1991

Satellite Data Formats and Standards

HRIT/LRIT For NOAA AVHRR and old
HRPT/LRPT geostationary data

Binary data -- without metadata

WMO binary data exchange formats - BUFR, GRIB

McIDAS

GeoTIFF

NetCDF

HDF

http://www.wmo.int/pages/prog/sat/formatsandstandards_en.php

Binary Data with a Header file

ImageryData

ImageryData.hdr

ENVI
description = {HDF Imported into ENVI.}
samples = 3660
lines = 3660
bands = 1
header offset = 0
file type = HDF Scientific Data
data type = 2
interleave = bsq
byte order = 0
map info = {UTM, 1.0, 1.0, 199980.000000, 4700040.000000, 30.000000, 30.000000, 15, North, WGS-84, units=Meters}
coordinate system string = {PROJCS["UTM_Zone_15N",GEOGCS["GCS_WGS_1984",DATUM["D_WGS_1984",SPHEROID["WGS_1984",6378137.0,298.257223563]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]],PROJECTION["Transverse_Mercator"],PARAMETER["False_Easting",500000.0],PARAMETER["False_Northing",0.000000],PARAMETER["Central_Meridian",93.000000],PARAMETER["Scale_Factor",0.9996],PARAMETER["Latitude_Of_Origin",0.0],UNIT["Meter",1.0]]}

McIDAS

- The **M**an **c**omputer **I**nteractive **D**ata **A**ccess **S**ystem (**McIDAS**) is not simply a satellite data format. It is rather a suite of applications for analyzing and displaying meteorological data for research and education.

- McIDAS has been in use and under continual development by the University of Wisconsin-Madison Space Science and Engineering Center (SSEC) since 1972.

- The Unidata McIDAS software (a superset of SSEC McIDAS) has been under development since 1985 and in distribution since 1988. The software can be used with conventional observational, satellite, and grid-point data.

  - Each file consists of three blocks, the Area Block, the Navigation Block, and the Data Block.

McIDAS--Example

Area Block

The Area Block consists of 256 bytes of intermixed 4-byte integers (words) and ASCII characters.

Word	Contents	In the AMSU files, expect this
1	contains zeros if the record is valid	0
2	area format: always 4	4
3	sensor source number	NOAA satellite number + 50 (65 indicates NOAA 15)
4	start date; YYYDDD	Year = 1900 + YYY
5	start time; HHMMSS UTC	
6	starting line number	1
7	starting element number	1
8	not used	0
9	number of lines in the area	
10	number of elements in each line	32 or 92
11	number of bytes per element (1, 2 or 4)	2
12	line resolution; number of image lines between consecutive area lines	1
13	element resolution; number of image elements between consecutive area elements	1
14	maximum number of bands per line of the area	1

McIDAS--Example

Navigation Block

The Navigation Block also consists of 512 bytes of intermixed 4-byte integers (words) and ASCII characters.

Word	Parameter	In the AMSU files....
1	navigation type; 4 ASCII characters	TIRO
2	sensor source, year, day of year (SSSYYDDD)	sensor source is given in Area Block Word 3
3	start time; HHMMSS UTC	Should be same as Word 5 in the Area Block, but in the past has been mistakenly set to zero. See Word 48, which is what McIDAS uses.
4	orbit type; always 1	1
5	epoch date; YYMMDD	
6	epoch time; HHMMSS UTC	Note: Word 15 has the fraction of a second in epoch time
7	semimajor axis; km*100	
8	orbital eccentricity; *1000000	
9	orbital inclination; degrees*1000	
10	mean anomaly; degrees*1000	

McIDAS--Example

Data Block

The last thing in the file is the data.

Calibration

-1	Not observed
-2	Not retrieved ("flagged")
...	Other problems

File Extension	Parameter	Unit	Range
C01...C20	antenna temps in channels 1-20	K	70 - 325
RR	AMSU-A rain rate	mm/hr	0 - 30
RRB	AMSU-B rain rate	mm/hr	0 - 30
TPW	total precipitable water	mm	0 - 75
CLW	cloud liquid water	mm	0 - 6
ICE	sea ice	Areal coverage in percent	30 - 100
IC2	sea ice (with edges)	Areal coverage in	30 - 100

Geographic Tagged Image File Format (GeoTIFF)

GeoTIFF offers the capability to embed a wide range of **georeferencing information** (e.g. projection, datums and ellipsoids, coordinate values) as compliant descriptive tags (metadata) and structures within the Tagged Image File Format (TIFF) file.

GeoTIFF files can be read directly in ERDAS Imagine, ENVI, and PCI. However, it is important to note that ERDAS Imagine and PCI do not correctly translate the GeoTIFF projection information.

Geographic Tagged Image File Format (GeoTIFF)

GTiff -- GeoTIFF File Format

GeoTIFF is a public domain metadata standard which allows georeferencing information to be embedded within a TIFF file. The potential additional information includes map projection, coordinate systems, ellipsoids, datums, and everything else necessary to establish the exact spatial reference for the file

Internal nodata masks

TIFF files can contain internal transparency masks.

Overviews

The GeoTIFF driver supports reading, creation and update of internal overviews.

Metadata

GDAL can deal with the following baseline TIFF tags as dataset-level metadata :

TIFFTAG_DOCUMENTNAME

TIFFTAG_IMAGEDESCRIPTION

TIFFTAG_SOFTWARE

TIFFTAG_DATETIME

……

TIFFTAG_RESOLUTIONUNIT

TIFFTAG_MINSAMPLEVALUE (read only)

TIFFTAG_MAXSAMPLEVALUE (read only)

Color Profile Metadata

Nodata value

Hierarchical Data Format (HDF)

Hierarchical Data Format, commonly abbreviated **HDF**, **HDF4**, or **HDF5,** is a library and multi-object file format for the transfer of graphical and numerical data between computers.

HDF is **self-describing**, allowing an application to interpret the structure and contents of a file without any outside information.

HDF is a container for several different datasets, storing Scientific Datasets (SDS).

SDS is a multidimensional array filled by data. One HDF file may contain several different SDS arrays.

SDS may differ in size, number of dimensions and may represent data for different regions.

HDF-EOS format

NASA's Earth Observing System (EOS) maintains its own HDF modification called HDF-EOS.
EOS use HDF4-EOS for data storing data such as from `Terra' and `Aqua' satellites. HDF4-EOS has been switched to HDF5-EOS format in VIIRS.

HDF-EOS data format is standard HDF by adding conventions, data types, and metadata elements as specified for the Core System of the Earth Observing System Data and Information System (**ECS**).
HDF-EOS adds three geolocation data types (point, grid, and swath) which allow the file contents to be queried by earth coordinates and time.

HDF-EOS

- HDF-EOS is an extension to HDF which standardizes storage and access to common Earth Sciences data structures
- By providing a single interface to data structures, common to earth science
- By providing a container for EOS inventory, archive and product specific metadata
- By establishing a consistent relationship between geolocation and science data.

HDF4 - based, storage format for EOS standard products.
Used operationally by MODIS, MISR, ASTER, Landsat, AIRS and other EOS instruments such as HLS (harmonized Landsat and Sentinel-2)
Support for Grid/Point/Swath structures

HFD5

- Based on HDF5, a complete rewrite of HDF-EOS2 with a different user interface.
  - –First released in 2000.
  - –Format used by EOS Aura instruments
  - Format used by VIIRS land product, ECOSTRESS products

- Designed to 'resemble' HDF-EOS 2 to the maximum extent possible.
  - –Supports same data structures
  - –Added prefix 'HE5_' to HDF-EOS 2 functions.

A compound data type example

```
types:
  compound wind_vector_t {
    float eastward ;
    float northward ;
    }
dimensions:
    lat = 18 ;
    lon = 36 ;
    pres = 15 ;
    time = 4 ;
variables:
       wind_vector_t gwind(time, pres, lat, lon) ;
       wind:long_name = "geostrophic wind vector" ;
       wind:standard_name = "geostrophic_wind_vector" ;
data:
    gwind = {1, -2.5}, {-1, 2}, {20, 10}, {1.5, 1.5}, ...;
```

HDF4-EOS

dimensions

variables

global attributes

CoreMetadata

SDS data

Read HDF4-EOS Data

(1)Install HDF4 Tools in Linux

(2)Check the commands

(3) Check the variables using

ncdump

(4) Check data using

Hdp

Hdp dumpsds –h

(5) Dump data:

Hdf dumpsds

Ncdump in HDF4 Tools - Check hdf4 files

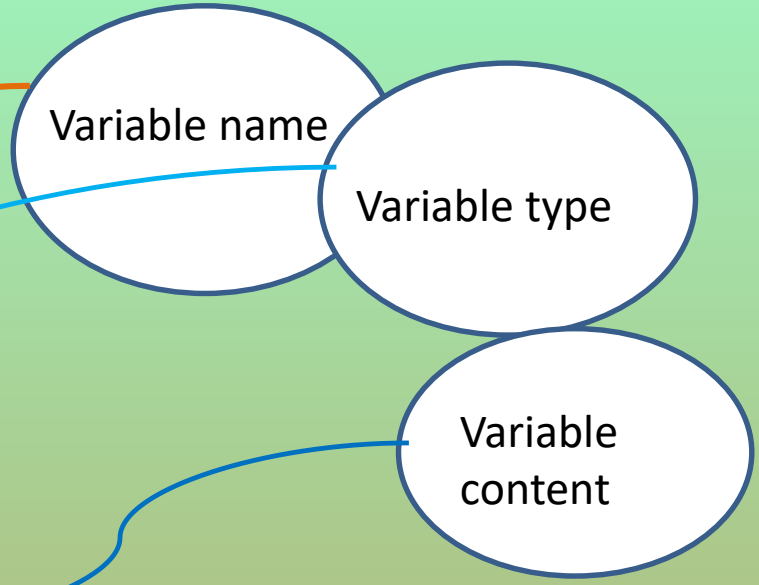
[zhangx@hunter]$ /hunter/data/CODE/HDF/hdf-4.2.6-linux-x86_64/bin/ncdump

./ncdump [-V|-c|-h|-u] [-v ...] [[-b|-f] [c|f]] [-l len] [-n name] [-d n[,n]] file

[-V] Display version of the HDF4 library and exit
[-c] Coordinate variable data and header information
[-h] Header information only, no data
[-u] Replace nonalpha-numerics in names with underscores
[-v var1[,...]] Data for variable(s) <var1>,... only
[-b [c|f]] Brief annotations for C or Fortran indices in data
[-f [c|f]] Full annotations for C or Fortran indices in data
[-l len] Line length maximum in data section (default 80)
[-n name] Name for netCDF (default derived from file name)
[-d n[,n]] Approximate floating-point values with less precision
file File name of input netCDF file

Ncdump -- Example (1)

```
/hunter/data/CODE/HDF/hdf-4.2.6-linux-x86_64/bin/ncdump -h
MCD12C1.A2007001.005.2009288175546.hdf|more
```




```
MCD12C1.A2007001.005.2009288175546 {
dimensions:
        YDim_MOD12C1 = 3600 ;
        XDim_MOD12C1 = 7200 ;
        Num_IGBP_Classes_MOD12C1 = 18 ;
        Num_UMD_Classes_MOD12C1 = 15 ;
        Num_LAI_FPAR_Classes_MOD12C1 = 12 ;

variables:
        byte Majority_Land_Cover_Type_1 (YDim_MOD12C1, XDim_MOD12C1) ;
            Majority_Land_Cover_Type_1:long_name = "Majority_Land_Cover_Type_1" ;
            Majority_Land_Cover_Type_1:units = "class number" ;
            Majority_Land_Cover_Type_1:valid_range = '\0', '\20' ;
            Majority_Land_Cover_Type_1:_FillValue = '\377' ;
            Majority_Land_Cover_Type_1:water = '\0' ;
            Majority_Land_Cover_Type_1:evergreen_needleleaf_forest = '\1' ;
……..
            Majority_Land_Cover_Type_1:unclassified = '\377' ;
        byte Majority_Land_Cover_Type_1_Assessment(YDim_MOD12C1, XDim_MOD12C1) ;
```

Ncdump -- Example (2)

byte Majority_Land_Cover_Type_1_Assessment(YDim_MOD12C1, XDim_MOD12C1) ;

Majority_Land_Cover_Type_1_Assessment:long_name = "Majority_Land_Cover_Type_1_Assessment" ;
Majority_Land_Cover_Type_1_Assessment:units = "confidences" ;
Majority_Land_Cover_Type_1_Assessment:valid_range = '\0', 'd' ;
Majority_Land_Cover_Type_1_Assessment:_FillValue = '\377' ;

byte Land_Cover_Type_1_Percent (YDim_MOD12C1, XDim_MOD12C1, Num_IGBP_Classes_MOD12C1) ;

Land_Cover_Type_1_Percent:long_name = "Land_Cover_Type_1_Percent" ;
Land_Cover_Type_1_Percent:units = "percent in integers" ;
Land_Cover_Type_1_Percent:valid_range = '\0', 'd' ;
Land_Cover_Type_1_Percent:_FillValue = '\377' ;
Land_Cover_Type_1_Percent:Layer_0 = "water" ;
Land_Cover_Type_1_Percent:Layer_1 = "evergreen needleleaf forest" ;

……

Land_Cover_Type_1_Percent:Layer_15 = "snow and ice" ;
Land_Cover_Type_1_Percent:Layer_16 = "barren or sparsely vegetated" ;
Land_Cover_Type_1_Percent:Layer_17 = "unclassified + fill values" ;

Ncdump -- Example (3)

byte Majority_Land_Cover_Type_1_QC(YDim_MOD12C1, XDim_MOD12C1) ;

Majority_Land_Cover_Type_1_QC:long_name = "Majority_Land_Cover_Type_1_QC" ;

Majority_Land_Cover_Type_1_QC:units = "concatenated flags" ;

…….

Majority_Land_Cover_Type_1_QC:not_processed_due_to_other_effects = '\3' ;

byte Majority_Land_Cover_Type_2(YDim_MOD12C1, XDim_MOD12C1) ;

Majority_Land_Cover_Type_2:long_name = "Majority_Land_Cover_Type_2" ;

Majority_Land_Cover_Type_2:units = "class number" ;

Majority_Land_Cover_Type_2:valid_range = '\0', '\20' ;

Majority_Land_Cover_Type_2:_FillValue = '\377' ;

Majority_Land_Cover_Type_2:water = '\0' ;

Majority_Land_Cover_Type_2:evergreen_needleleaf_forest = '\1' ;

…….

Majority_Land_Cover_Type_2:barren_or_sparsely_vegetated = '\20' ;

Majority_Land_Cover_Type_2:unclassfied = '\377' ;

Ncdump -- Example (4)

```
// global attributes:
            :HDFEOSVersion = "HDFEOS_V2.9" ;
            :StructMetadata_0 = "GROUP=SwathStructure\n",
   "END_GROUP=SwathStructure\n",
   "GROUP=GridStructure\n",
   "\tGROUP=GRID_1\n",
   "\t\tGridName=\"MOD12C1\"\n",
   "\t\tXDim=7200\n",
   "\t\tYDim=3600\n",
   "\t\tUpperLeftPointMtrs=(-180000000.000000,90000000.000000)\n",
   "\t\tLowerRightMtrs=(180000000.000000,-90000000.000000)\n",
   "\t\tProjection=GCTP_GEO\n",
   "\t\tGridOrigin=HDFE_GD_UL\n",
   "\t\tGROUP=Dimension\n",
   "\t\t\tOBJECT=Dimension_1\n",
   "\t\t\t\tDimensionName=\"Num_IGBP_Classes\"\n",
   "\t\t\t\tSize=18\n",
   "\t\t\tEND_OBJECT=Dimension_1\n",
   "\t\t\tOBJECT=Dimension_2\n",
……
 "END_GROUP=GridStructure\n",
   "GROUP=PointStructure\n",
   "END_GROUP=PointStructure\n",
   "END\n",
```

Ncdump -- Example (5)

```
:CoreMetadata_0 = "\n",
  "GROUP                = INVENTORYMETADATA\n",
  "  GROUPTYPE          = MASTERGROUP\n",
  "  GROUP              = ECSDATAGRANULE\n",
        "    OBJECT              = REPROCESSINGPLANNED\n",
          "      NUM_VAL           = 1\n",
         "      VALUE             = \"further update is anticipated\"\n",
         "    END_OBJECT          = REPROCESSINGPLANNED\n",
      "    OBJECT              = REPROCESSINGACTUAL\n",
  "      NUM_VAL           = 1\n",
  "      VALUE             = \"reprocessed\"\n",
  "    END_OBJECT          = REPROCESSINGACTUAL\n",
  "    OBJECT              = LOCALGRANULEID\n",
  "      NUM_VAL           = 1\n",
  "      VALUE             = \"MCD12C1.A2007001.005.2009288175546.hdf\"\n",
  "    END_OBJECT          = LOCALGRANULEID\n",
  "    OBJECT              = DAYNIGHTFLAG\n",
  "      NUM_VAL           = 1\n",
  "      VALUE             = \"Day\"\n",
  "    END_OBJECT          = DAYNIGHTFLAG\n",
  "    OBJECT              = PRODUCTIONDATETIME\n",
  "      NUM_VAL           = 1\n",
  "      VALUE             = \"2009-10-15T17:55:46.000Z\"\n",
  "    END_OBJECT          = PRODUCTIONDATETIME\n",
  "    OBJECT              = LOCALVERSIONID\n",
  "      NUM_VAL           = 1\n",
  "      VALUE             = \"3.1.0\"\n",
  "    END_OBJECT          = LOCALVERSIONID\n",
  "  END_GROUP           = ECSDATAGRANULE\n",
```

Ncdump -- Example (6)

```
"  GROUP                = MEASUREDPARAMETER\n",
   "    OBJECT               = MEASUREDPARAMETERCONTAINER\n",
   ……
"    END_OBJECT           = MEASUREDPARAMETERCONTAINER\n",
"  END_GROUP            = MEASUREDPARAMETER\n",
"  GROUP                = COLLECTIONDESCRIPTIONCLASS\n",
"    OBJECT             = SHORTNAME\n",
   ……
"  GROUP                = INPUTGRANULE\n",
"    OBJECT             = INPUTPOINTER\n",
   "      NUM_VAL            = 100\n",
   "      VALUE              = (\"MLCT_1.A2008001.005.hdf\", \"MLCT_1_A.A2008001.005.hdf\",
\"LCT_1_P.A2008001.005.hdf\", \"MLCT_2.A2008001.005.hdf\", \"MLCT_2_A.A20080
01.005.hdf\", \"LCT_2_P.A2008001.005.hdf\", \"MLCT_3.A2008001.005.hdf\",
\"MLCT_3_A.A2008001.005.hdf\", \"LCT_3_P.A2008001.005.hdf\")\n",
   "    END_OBJECT           = INPUTPOINTER\n",
"  END_GROUP            = INPUTGRANULE\n",
```

Ncdump -- Example (7)

```
" GROUP = SPATIALDOMAINCONTAINER\n",
"  GROUP = HORIZONTALSPATIALDOMAINCONTAINER\n",
"   GROUP = BOUNDINGRECTANGLE\n",
"    OBJECT = WESTBOUNDINGCOORDINATE\n",
"     NUM_VAL = 1\n",
"     VALUE = -180.0\n",
"    END_OBJECT = WESTBOUNDINGCOORDINATE\n",
"    OBJECT = NORTHBOUNDINGCOORDINATE\n",
"     NUM_VAL = 1\n",
"     VALUE = 90.0\n",
"    END_OBJECT = NORTHBOUNDINGCOORDINATE\n",
"    OBJECT = EASTBOUNDINGCOORDINATE\n",
"     NUM_VAL = 1\n",
"     VALUE = 180.0\n",
"    END_OBJECT = EASTBOUNDINGCOORDINATE\n",
"    OBJECT = SOUTHBOUNDINGCOORDINATE\n",
"     NUM_VAL = 1\n",
"     VALUE = -90.0\n",
"    END_OBJECT = SOUTHBOUNDINGCOORDINATE\n",
"   END_GROUP = BOUNDINGRECTANGLE\n",
"  END_GROUP = HORIZONTALSPATIALDOMAINCONTAINER\n",
" END_GROUP = SPATIALDOMAINCONTAINER\n",
" GROUP = RANGEDATETIME\n",
"  OBJECT = RANGEBEGINNINGDATE\n",
"   NUM_VAL = 1\n",
"   VALUE = \"2007-01-01\"\n",
"  END_OBJECT = RANGEBEGINNINGDATE\n",
"  OBJECT = RANGEBEGINNINGTIME\n",
"   NUM_VAL = 1\n",
"   VALUE = \"00:00:00.000000\"\n",
"  END_OBJECT = RANGEBEGINNINGTIME\n",
```

```
"  OBJECT = RANGEENDINGDATE\n",
"   NUM_VAL = 1\n",
"   VALUE = \"2007-12-31\"\n",
"  END_OBJECT = RANGEENDINGDATE\n",
"  OBJECT = RANGEENDINGTIME\n",
"   NUM_VAL = 1\n",
"   VALUE = \"23:59:00.000000\"\n",
"  END_OBJECT = RANGEENDINGTIME\n",
" END_GROUP = RANGEDATETIME\n",
" GROUP = PGEVERSIONCLASS\n",
"  OBJECT = PGEVERSION\n",
"   NUM_VAL = 1\n",
"   VALUE = \"5.0.0\"\n",
"  END_OBJECT = PGEVERSION\n",
" END_GROUP = PGEVERSIONCLASS\n",
" GROUP = ASSOCIATEDPLATFORMINSTRUMENTSENSOR\n",
```

Ncdump -- Example (8)

```
"       VALUE           = \"58\"\n",
   "       END_OBJECT          = PARAMETERVALUE\n",
"     END_GROUP           = INFORMATIONCONTENT\n",
"    END_OBJECT          =
ADDITIONALATTRIBUTESCONTAINER\n",
"    OBJECT              = ADDITIONALATTRIBUTESCONTAINER\n",
   "      CLASS           = \"2\"\n",
"     OBJECT              = ADDITIONALATTRIBUTENAME\n",
   "       CLASS           = \"2\"\n",
   "       NUM_VAL           = 1\n",
   "       VALUE           = \"QAPERCENTOTHERQUALITY\"\n",
   "      END_OBJECT          = ADDITIONALATTRIBUTENAME\n",
"     GROUP               = INFORMATIONCONTENT\n",
   "       CLASS           = \"2\"\n",
"      OBJECT              = PARAMETERVALUE\n",
   "        NUM_VAL           = 1\n",
   "        CLASS           = \"2\"\n",
   "        VALUE           = \"115\"\n",
   "       END_OBJECT          = PARAMETERVALUE\n",
"     END_GROUP           = INFORMATIONCONTENT\n",
"    END_OBJECT          =
ADDITIONALATTRIBUTESCONTAINER\n",
"    OBJECT              = ADDITIONALATTRIBUTESCONTAINER\n",
   "      CLASS           = \"3\"\n",
"     OBJECT              = ADDITIONALATTRIBUTENAME\n",
   "       CLASS           = \"3\"\n",
   "       NUM_VAL           = 1\n",
   "       VALUE           =
\"QAPERCENTNOTPRODUCEDCLOUD\"\n",
   "      END_OBJECT          = ADDITIONALATTRIBUTENAME\n",
"     GROUP               = INFORMATIONCONTENT\n",
   "       CLASS           = \"3\"\n",
"      OBJECT              = PARAMETERVALUE\n",
   "        NUM_VAL           = 1\n",
   "        CLASS           = \"3\"\n",
   "        VALUE           = \"0\"\n",
   "       END_OBJECT          = PARAMETERVALUE\n",
"     END_GROUP           = INFORMATIONCONTENT\n",
"    END_OBJECT          =
ADDITIONALATTRIBUTESCONTAINER\n",
"    OBJECT              =
ADDITIONALATTRIBUTESCONTAINER\n",
   "      CLASS           = \"4\"\n",
"     OBJECT              = ADDITIONALATTRIBUTENAME\n",
   "       CLASS           = \"4\"\n",
   "       NUM_VAL           = 1\n",
   "       VALUE           =
\"QAPERCENTNOTPRODUCEDOTHER\"\n",
   "      END_OBJECT          =
ADDITIONALATTRIBUTENAME\n",
"     GROUP               = INFORMATIONCONTENT\n",
   "       CLASS           = \"4\"\n",
"      OBJECT              = PARAMETERVALUE\n",
   "        NUM_VAL           = 1\n",
   "        CLASS           = \"4\"\n", "       VALUE           = \"-
73\"\n",
   "       END_OBJECT          = PARAMETERVALUE\n",
"     END_GROUP           = INFORMATIONCONTENT\n",
"    END_OBJECT          =
ADDITIONALATTRIBUTESCONTAINER\n",
"  END_GROUP           = ADDITIONALATTRIBUTES\n",
"END_GROUP           = INVENTORYMETADATA\n",
"END\n",
   "" ;
```

xiaoyang Zhang, 3/19/2021

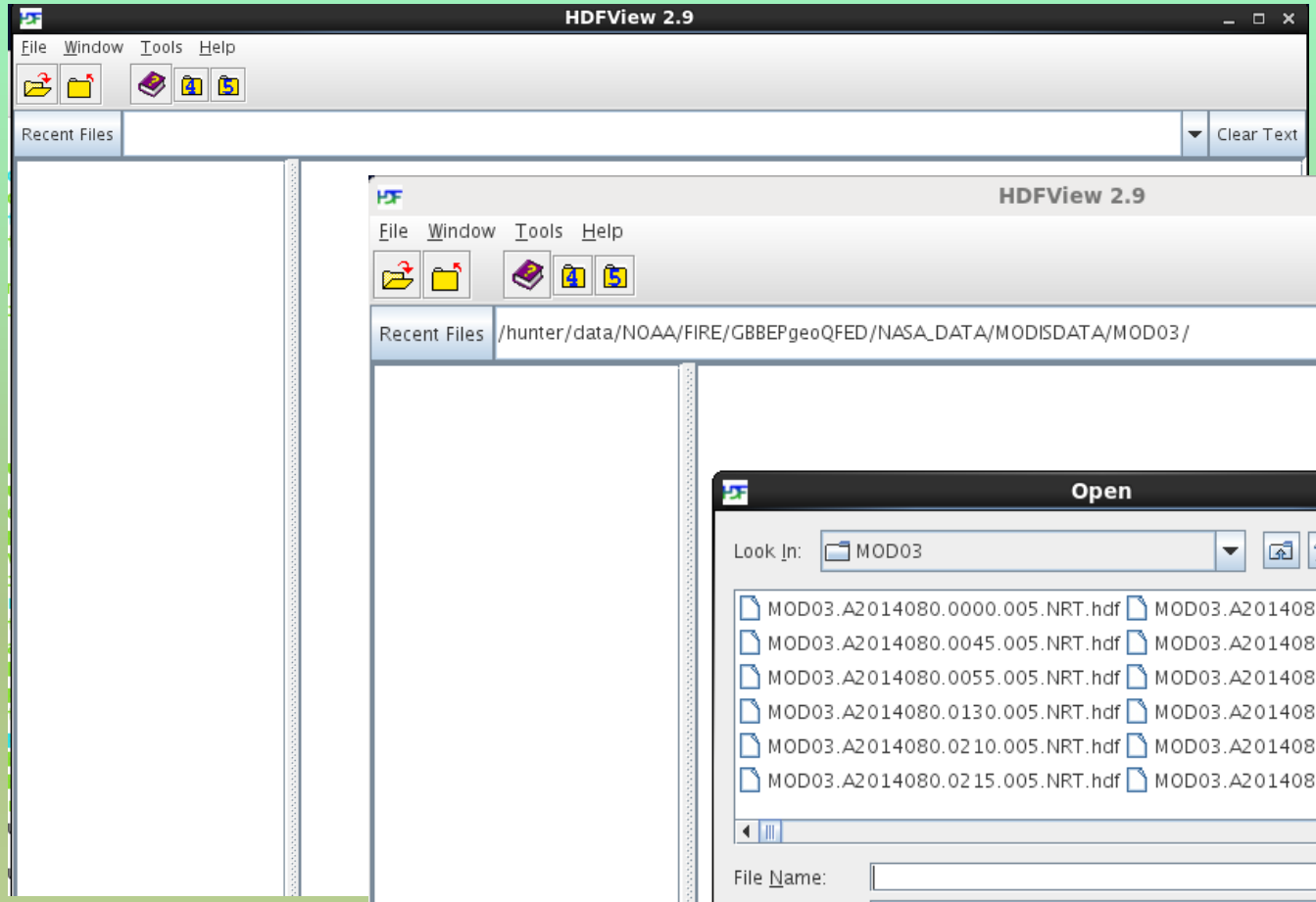
Ncdump -- Example (9)

```
:ArchiveMetadata_0 = "\n",
   "GROUP              = ARCHIVEDMETADATA\n",
   "  GROUPTYPE          = MASTERGROUP\n",
"  OBJECT             = DESCRREVISION\n",
   "    NUM_VAL            = 1\n",
   "    VALUE            = \"5.0\"\n",
   "  END_OBJECT           = DESCRREVISION\n",
"  OBJECT             = LONGNAME\n",
   "    NUM_VAL            = 1\n",
   "    VALUE            = \"MODIS/Terra+Aqua Land Cover Type
Yearly L3 Global 0.05Deg CMG\"\n",
   "  END_OBJECT           = LONGNAME\n",
"  OBJECT             =
ALGORITHMPACKAGEACCEPTANCEDATE\n",
……
"  END_OBJECT           = ALGORITHMPACKAGEVERSION\n",
"  OBJECT             = GEOANYABNORMAL\n",
   "    NUM_VAL            = 1\n",
   "    VALUE            = \"False\"\n",
   "  END_OBJECT           = GEOANYABNORMAL\n",
"  OBJECT             = GEOESTMAXRMSERROR\n",
   "    NUM_VAL            = 1\n",
   "    VALUE            = 500.0\n",
   "  END_OBJECT           = GEOESTMAXRMSERROR\n",
```

```
"  OBJECT             = SPSOPARAMETERS\n",
   "    NUM_VAL            = 1\n",
   "    VALUE            = \"2669\"\n",
   "  END_OBJECT           = SPSOPARAMETERS\n",
"  OBJECT             = PROCESSINGCENTER\n",
   "    NUM_VAL            = 1\n",
   "    VALUE            = \"MODAPS\"\n",
   "  END_OBJECT           = PROCESSINGCENTER\n",
"  OBJECT             = GLOBALGRIDCOLUMNS\n",
   "    NUM_VAL            = 1\n",
   "    VALUE            = 7200\n",
   "  END_OBJECT           = GLOBALGRIDCOLUMNS\n",
"  END_OBJECT           = NUMBEROFGRANULES\n",
"  OBJECT             = COVERAGECALCULATIONMETHOD
\n",
   "    NUM_VAL            = 1\n",
   "    VALUE            = \"area\"\n",
   "  END_OBJECT           =
COVERAGECALCULATIONMETHOD\n",
"  OBJECT             = NADIRDATARESOLUTION\n",
   "    NUM_VAL            = 1\n",
   "    VALUE            = \"0.25 Degree\"\n",
   "  END_OBJECT           = NADIRDATARESOLUTION\n",
   "\n",
   "END_GROUP           = ARCHIVEDMETADATA\n",
   "\n",
   "END\n",
   "" ;
```

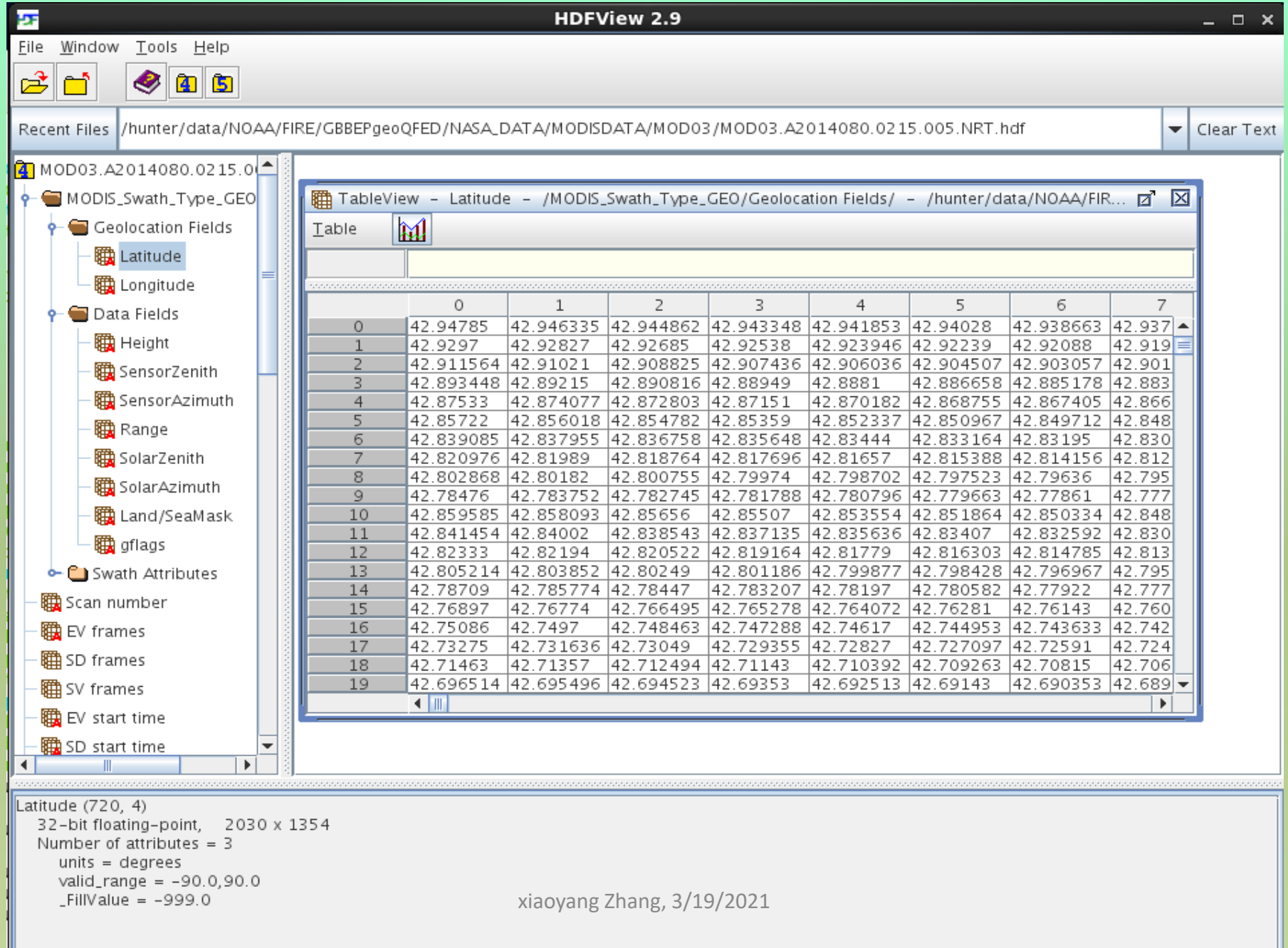
Hdfview (1)

[zhangx@hunter ~]$ hdfview



xiaoyang Zhang, 3/19/2021

Hdfview (2)


HDFView 2.9
File Window Tools Help
Recent Files
/hunter/data/NOAA/FIRE/GBBEPgeoQFED/NASA_DATA/MODISDATA/MOD03/MOD03.A2014080.0215.005.NRT.hdf
Clear Text
MOD03.A2014080.0215.0
MODIS_Swath_Type_GEO
Geolocation Fields
Latitude
Longitude
Data Fields
Height
SensorZenith
SensorAzimuth
Range
SolarZenith
SolarAzimuth
Land/SeaMask
gflags
Swath Attributes
Scan number
EV frames
SD frames
SV frames
EV start time
SD start time
TableView - Latitude - /MODIS_Swath_Type_GEO/Geolocation Fields/ - /hunter/data/NOAA/FIR...
Table
0 1 2 3 4 5 6 7
0 42.94785 42.946335 42.944862 42.943348 42.941853 42.94028 42.938663 42.937
1 42.9297 42.92827 42.92685 42.92538 42.923946 42.92239 42.92088 42.919
2 42.911564 42.91021 42.908825 42.907436 42.906036 42.904507 42.903057 42.901
3 42.893448 42.89215 42.890816 42.88949 42.8881 42.886658 42.885178 42.883
4 42.87533 42.874077 42.872803 42.87151 42.870182 42.868755 42.867405 42.866
5 42.85722 42.856018 42.854782 42.85359 42.852337 42.850967 42.849712 42.848
6 42.839085 42.837955 42.836758 42.835648 42.83444 42.833164 42.83195 42.830
7 42.820976 42.81989 42.818764 42.817696 42.81657 42.815388 42.814156 42.812
8 42.802868 42.80182 42.800755 42.79974 42.798702 42.797523 42.79636 42.795
9 42.78476 42.783752 42.782745 42.781788 42.780796 42.779663 42.77861 42.777
10 42.859585 42.858093 42.85656 42.85507 42.853554 42.851864 42.850334 42.848
11 42.841454 42.84002 42.838543 42.837135 42.835636 42.83407 42.832592 42.830
12 42.82333 42.82194 42.820522 42.819164 42.81779 42.816303 42.814785 42.813
13 42.805214 42.803852 42.80249 42.801186 42.799877 42.798428 42.796967 42.795
14 42.78709 42.785774 42.78447 42.783207 42.78197 42.780582 42.77922 42.777
15 42.76897 42.76774 42.766495 42.765278 42.764072 42.76281 42.76143 42.760
16 42.75086 42.7497 42.748463 42.747288 42.74617 42.744953 42.743633 42.742
17 42.73275 42.731636 42.73049 42.729355 42.72827 42.727097 42.72591 42.724
18 42.71463 42.71357 42.712494 42.71143 42.710392 42.709263 42.70815 42.706
19 42.696514 42.695496 42.694523 42.69353 42.692513 42.69143 42.690353 42.689
Latitude (720, 4)
32-bit floating-point, 2030 x 1354
Number of attributes = 3
units = degrees
valid_range = -90.0,90.0
_FillValue = -999.0

HDP the HDF dumper

SYNOPSIS

hdp [hdp options] hdp command [command options] <filename list>

HDP COMMANDS

hdp currently has two types of commands: list and dump. Other types of commands such as those for editing may be added in the future.

hdp list: lists contents of files

hdp dumpsds: displays data of NDGs and SDGs in the listed files.

hdp dumpvd : displays data of vdatas in the listed files.

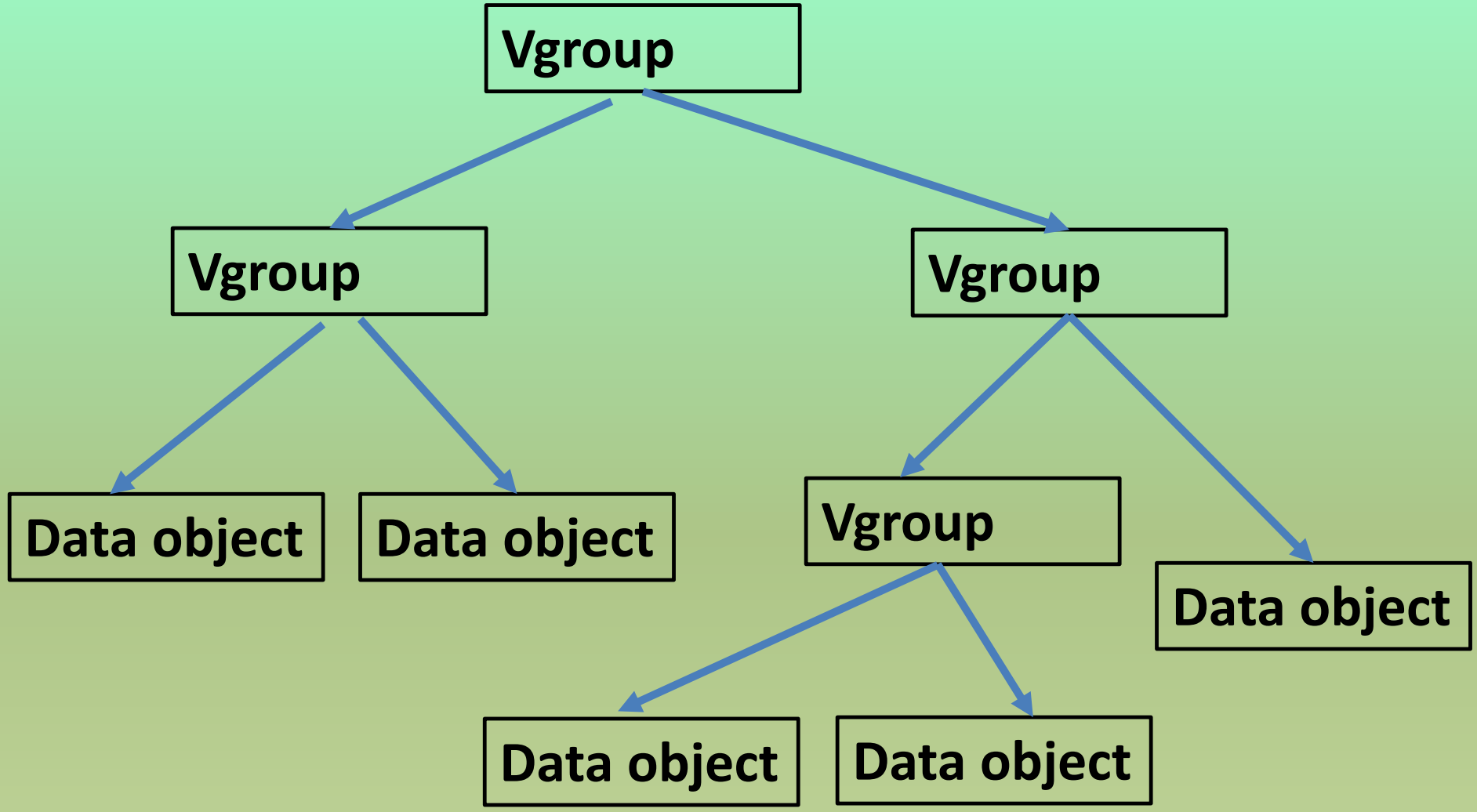
hdp dumpvg: **displays data of objects in vgroups in the listed files.**

hdp dumprig: displays data of RIGs in the listed files.

hdp dumpgr: **displays data of general RIGs in the listed files.**

<filelist> list of hdf file names, separated by spaces.

A *vgroup* is a structure designed to associate related data objects



hdp -H list Usage:

```
[zhangx@hunter TEMP]$  hdp -H list Usage:
hdp list [-acensldg] [-o<f|g|t|n>] [-t tag] <filelist>
        -a          Print annotations of items (sets long output)
        -c          Print classes of items (sets long output)
        -n          Print names or labels of items (sets long output)
        -e          Print special element information of items (sets long output)
        -s          Short output (default)
        -l          Long output
        -d          Debugging output
        -g          Print groups only
        -t <number>         Print items of with a given tag number
        -t <name>           Print items of with a given tag name
        -of         Print items in the order found in the file
        -og         Print items in group order
        -ot         Print items in tag order (default)
        <filelist> List of hdf file names, separated by spaces
```

hdp (2)

[zhangx@hunter MYD03]$ hdp dumpsds
Usage:
hdp dumpsds [-a|-i <indices>|-r <refs>|-n <names>] [-cdhvs] [-o <filename>] [-bx] <filelist>

-a Dump all SDSs in the file (default)
-i <indices> Dump the SDSs at positions listed in <indices>
-r <refs> Dump the SDSs with reference number listed in <refs>
-n <names> Dump the SDSs with name listed in <names>
-d Dump data only, no tag/ref, formatted to input to hp2hdf
-h Dump header only, no annotation for elements nor data
-v Dump everything including all annotations (default)
-c Print space characters as they are, not \digit
-g Do not print data of file (global) attributes
-l Do not print data of local attributes
-s Do not add carriage return to a long line - dump it as a stream
-o <filename> Output to file <filename>
-b Binary format of output
-x Ascii text format of output (default)
<filelist> List of hdf file names, separated by spaces

Hdp (3)

To obtain variable Name, Index, Reference:

[zhangx@hunter MYD03]$ hdp dumpsds -h MYD03.A2014174.2245.005.NRT.hdf|more

But there are large File attributes:

File attributes:

Attr0: Name = HDFEOSVersion
Type = 8-bit signed char
Count= 12
Value = HDFEOS_V2.17
Attr1: Name = StructMetadata.0
Type = 8-bit signed char
Count= 32000
Value = GROUP=SwathStructure\012\011GROUP=SWATH_1

……
\000\000\000\000\000\000\000\000\000\000\000
…….

Variable Name = Latitude
Index = 0
Type= 32-bit floating point
Ref. = 4
Compression method = DEFLATE
Rank = 2
Number of attributes = 3
Dim0: Name=nscans*10:MODIS_Swath_Type_GEO
Size = 2030
Scale Type = number-type not set
Number of attributes = 0

hdp dumpsds -n I1_SurfRefl_CMG -c -o tt1 -x NPP_SRFLIP_CMG.A2013359.C1_03110.all.hdf

```
zhangx@hunter ]$ hdp dumpsds -n I1_SurfRefl_CMG -c NPP_SRFLIP_CMG.A2013359.C1_03110.all.hdf|more
File name: NPP_SRFLIP_CMG.A2013359.C1_03110.all.hdf
File attributes:
        Attr0: Name = HDFEOSVersion
               Type = 8-bit signed char
               Count= 12
               Value = HDFEOS_V2.17
        Attr1: Name = StructMetadata.0
               Type = 8-bit signed char
               Count= 32000
               Value =
GROUP=SwathStructure
END_GROUP=SwathStructure
GROUP=GridStructure
        GROUP=GRID_1
                GridName="VIIRSCMG"
                XDim=7200
                YDim=3600
                UpperLeftPointMtrs=(-180000000.000000,90000000.000000)
                LowerRightMtrs=(180000000.000000,-90000000.000000)
                Projection=GCTP_GEO
                GROUP=Dimension
                END_GROUP=Dimension
                GROUP=DataField
```

[z@hunter]$ **hdp dumpsds -n I1_SurfRefl_CMG –g NPP_SRFLIP_CMG.A2013359.C1_03110.all.hdf|more**

File name: NPP_SRFLIP_CMG.A2013359.C1_03110.all.hdf

File attributes:

Attr0: Name = HDFEOSVersion
Type = 8-bit signed char
Count= 12
Attr1: Name = StructMetadata.0
Type = 8-bit signed char
Count= 32000

……

Variable Name = I1_SurfRefl_CMG
Index = 0
Type= 16-bit signed integer
Ref. = 4
Compression method = DEFLATE
Deflate level = 4
Compression ratio (original:compressed) = 1.38:1
Rank = 2
Number of attributes = 9
Dim0: Name=YDim:VIIRSCMG
Size = 3600
Scale Type = number-type not set
Number of attributes = 0
Dim1: Name=XDim:VIIRSCMG
Size = 7200
Scale Type = number-type not set
Number of attributes = 0
Attr0: Name = valid_range
Type = 16-bit signed integer
Count= 2
Value = -100 16000
Attr1: Name = _FillValue
Type = 16-bit signed integer
Count= 1
Value = -28672
Attr2: Name = scale_factor
Type = 64-bit floating point
Count= 1
Value = 10000.000000
Attr3: Name = scale_factor_err
Type = 64-bit floating point
Count= 1
Value = 0.000000
Attr4: Name = add_offset
Type = 64-bit floating point
Count= 1
Value = 0.000000
Attr5: Name = add_offset_err
Type = 64-bit floating point
Count= 1
Value = 0.000000
Attr6: Name = calibrated_nt
Type = 32-bit signed integer
Count= 1
Value = 5
Attr7: Name = units
Type = 8-bit signed char
Count= 11
Value = reflectance
Attr8: Name = long_name
Type = 8-bit signed char
Count= 15
Value = I1_SurfRefl_CMG
Data :
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

[zhangx@hunter TEMP]$ hdp dumpvd

Usage:

hdp dumpvd [-a|-i <indices>|-r <refs>|-n <names>|-c <classes>|-f <f1, f2,..>] [-dhv] [-o <filename>] [-bx] <filelist>

-a Dump all VDs in the file (default)

-i <indices> Dump the VDs at positions listed in <indices>

-r <refs> Dump the VDs with reference number listed in <refs>

-n <names> Dump the VDs with name listed in <names>

-c <classes> Dump the VDs with class listed in <classes>

-f <f1, f2,..> Dump based on fields in vdata header

-d Dump data only, no tag/ref, formatted to input to hp2hdf

-h Dump header only, no annotation for elements nor data

-v Dump everything including all annotations (default)

-o <filename> Output to file <filename>

-b Binary format of output

-x Ascii text format of output (default)

<filelist> List of hdf file names, separated by spaces

```
[zhangx@hunter TEMP]$ hdp dumpvd   NPP_SRFLIP_CMG.A2013359.C1_03110.all.hdf|more
File name: NPP_SRFLIP_CMG.A2013359.C1_03110.all.hdf

Vdata: 0
   tag = 1962; reference = 74;
   number of records = 1; interlace = FULL_INTERLACE (0);
   fields = [Values];
   record size (in bytes) = 4;
   name = YDim:VIIRSCMG; class = DimVal0.1;
   number of attributes = 0
- field index 0: [Values], type=24, order=1
   number of attributes = 0
Loc.     Data
0        3600  ;

Vdata: 1
   tag = 1962; reference = 76;
   number of records = 1; interlace = FULL_INTERLACE (0);
   fields = [Values];
   record size (in bytes) = 4;
   name = XDim:VIIRSCMG; class = DimVal0.1;
   number of attributes = 0
- field index 0: [Values], type=24, order=1
   number of attributes = 0
Loc.     Data
0        7200  ;
```

```
[zhangx@hunter TEMP]$ hdp dumpvg   NPP_SRFLIP_CMG.A2013359.C1_03110.all.hdf|more
File name: NPP_SRFLIP_CMG.A2013359.C1_03110.all.hdf
……
Vgroup:0
    tag = 1965; reference = 2;
    name = VIIRSCMG; class = GRID;
    number of entries = 2;
  number of attributes = 0
Entries:-
    #0 (Vgroup)
                tag = 1965;reference = 3;
                number of entries = 34;
                name = Data Fields; class = GRID Vgroup
  number of attributes = 0

Vgroup:6
    tag = 1965; reference = 89;
    name = I1_SurfRefl_CMG; class = Var0.0;
    number of entries = 16;
  number of attributes = 0
Entries:-
    #0 (Vgroup)
                tag = 1965;reference = 75;
                number of entries = 1;
                name = YDim:VIIRSCMG; class = Dim0.0
  number of attributes = 0
    #1 (Vgroup)
                tag = 1965;reference = 77;
                number of entries = 1;
                name = XDim:VIIRSCMG; class = Dim0.0
  number of attributes = 0
```

Hdp dumpsds (4)

[x@hunter]$ hdp dumpsds -n Latitude -d -o test -b MYD03.A2014174.2245.005.NRT.hdf

[zhangx@hunter]$ hdp dumpsds -n "EV frames" -d -o test21 -x MYD03.A2014174.2245.005.NRT.hdf

[zhangx@hunter]$ hdp dumpsds -i 0 -d -o test23 -x MYD03.A2014174.2245.005.NRT.hdf

[zhangx@hunter]$ hdp dumpsds -r 4 -d -o test24 -b MYD03.A2014174.2245.005.NRT.hdf

Note that there is a space for variable name of EV frames, so that a quote is needed.

Extract SDS data Using Perl Scripts

Examples to extract SDS (Latitude, and sun zenith angle) from a group of hdf file using a Perl scrip:

MYD03.A2014174.2220.005.NRT.hdf

……

MYD03.A2014174.2250.005.NRT.hdf

```
#!/usr/bin/perl
@files=("MYD03.A2014174.2220.005.NRT.hdf",  "MYD03.A2014174.2235.005.NRT.hdf",
"MYD03.A2014174.2245.005.NRT.hdf", "MYD03.A2014174.2255.005.NRT.hdf",
"MYD03.A2014174.2230.005.NRT.hdf",  "MYD03.A2014174.2240.005.NRT.hdf",
"MYD03.A2014174.2250.005.NRT.hdf");
$ENV{HDP}="/user/bin";
Foreach $file(@files){
chomp($file);
$stem=(split('NRT', $file))[0];   ### it removes "NRT.hdf"
$file_lat=join('',$stem, "lat.bin");
$file_sun_zenith=join('', $stem, "sun_zenith.txt");
system("$ENV{HDP}/hdp dumpsds -n Latitude -d -o $file_lat -b $file");
system("$ENV{HDP}/hdp dumpsds -n  "SD Sun zenith" -d -o $file_sun_zenith -x
$file");
}
```

Extract SDS data Using Perl Scripts

MYD03data.txt:
MYD03.A2014174.2220.005.NRT.hdf
……
MYD03.A2014174.2250.005.NRT.hdf

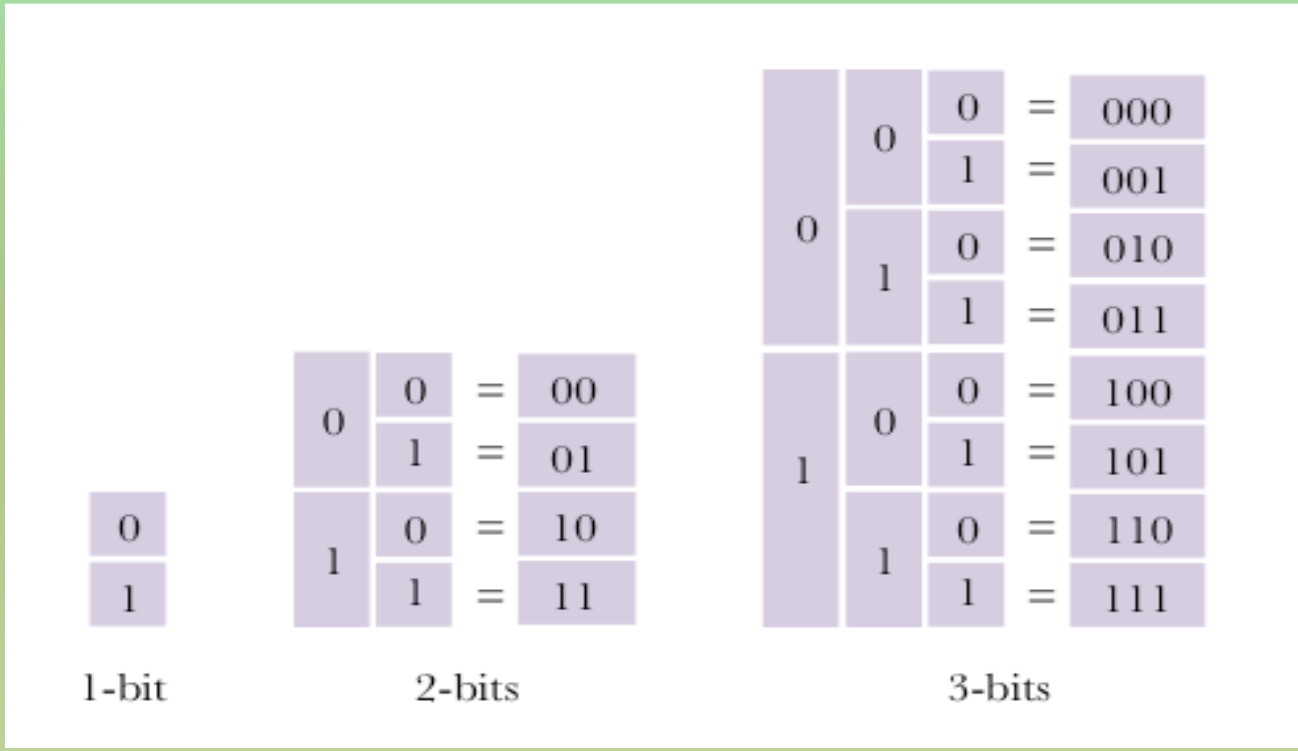
```
#!/usr/bin/perl
$infile="MYD03data.txt";
Open(INPUT, ">$infile")
@file = <INPUT>;        # Import all from a file into an array
close(INPUT);

$ENV{HDP}="/user/bin";
Foreach $file(@files){
chomp($file);
$stem=(split('NRT', $file))[0];   ### it removes "NRT.hdf"
$file_lat=join('',$stem, "lat.bin");
$file_sun_zenith=join('', $stem, "sun_zenith.txt");
system("$ENV{HDP}/hdp dumpsds -n Latitude -d -o $file_lat -b $file");
system("$ENV{HDP}/hdp dumpsds -n  "SD Sun zenith" -d -o $file_sun_zenith -x
$file");
}
```

Quality Controls in Binary Bits and bit Fields

- Interpreting the binary encoded bitmap

❖A single bit represents two values (0, 1)
❖Two bits represent four values, and
❖Three bits represent eight values



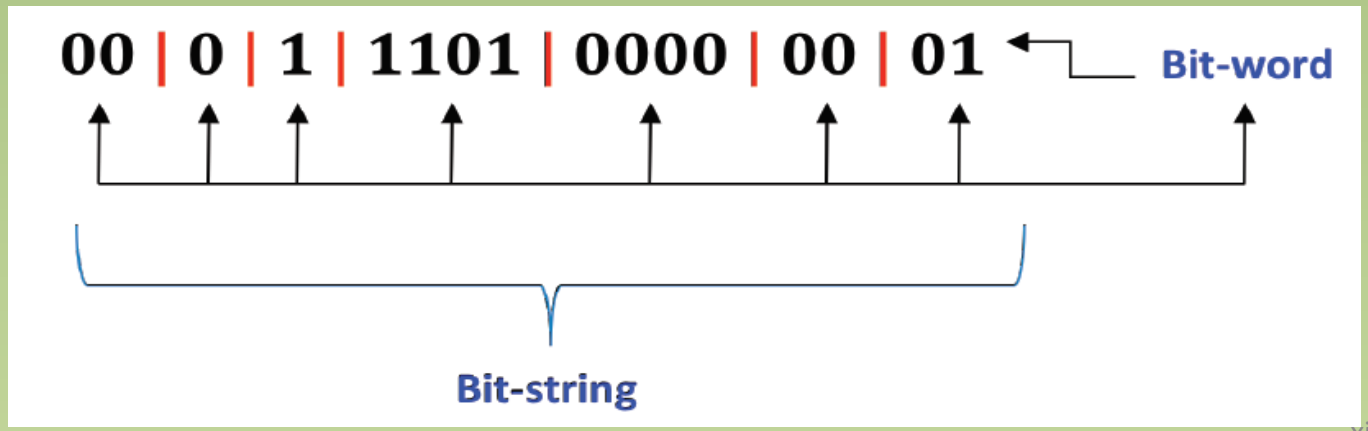
https://lpdaac.usgs.gov/sites/default/files/public/modis/docs/MODIS_LP_QA_Tutorial-1b.pdf

Binary Bits and Bit-Fields

Number of Bits	Number of Values	Formula
1	2	2^1
2	4	2^2
3	8	2^3
4	16	2^4
5	32	2^5
6	64	2^6
7	128	2^7
8	256	2^8

7425 decimal value

a binary value of 1110100000001



Example of QC in VIIRS Phenology Products

Table 2. Values of VIIRS GLSP_QC (8-bit)

Bit No.	Parameter Name	Bit Comb	GLSP_QC
0-2	GLSP_QC	000	0=processed, good quality
		001	1= processed, other quality
		010	2=processed, backup algorithm
		011	3= not processed, bad quality
		100	4=not processed, other
3-4	TBD	TBD	TBD
5-7	Land/water mask	000	0 = Shallow ocean
		001	1 = Land (Nothing else but land)
		010	2 = Ocean coastlines and lake shorelines
		011	3 = Shallow inland water
		100	4 = Ephemeral water
		101	5 = Deep inland water
		110	6 = Moderate or continental ocean
		111	7 = Deep ocean

DecBin.pl

```
#!/usr/bin/perl

my $binum='100101';   ##binary number
my $decimal=unpack("N", pack("B32",substr("0"x32 . $binum,-32)));
print "Binary $binum = Decimal $decimal \n";

my $decimal=38;
my $binum=unpack("B32", pack("N", $decimal));
print "Decimal $decimal = Binary $binum  \n";

$bin_num=11000011;
$decimal = oct("0b", $bin_num);  ##Octal
print "Decimal= $decimal\n";

my $decimal=100;
$bin_num=sprintf("%b",$decimal);
print "bin_num = $bin_num \n",
```

```
[zhangx@hunter TEMP]$./DecBin.pl
Binary 100101 = Decimal 37
Decimal 38 = Binary
00000000000000000000000000100110
Decimal= 99
bin_num = 1100100
[zhangx@hunter TEMP]$
```

HDF-EOS to GeoTIFF Converter (HEG)

HEG: Utility that converts EOSDIS data from HDF-EOS format to common Geographical Information System (GIS)-compatible formats.

- **Two versions available**:
  - A downloadable desktop version
  - Access through NASA archive online storage (Data Pools)

  - Implemented also in some prototypes:
    - Stand-alone Data Pool
    - OGC (Open Geospatial Consortium) for Web Services Chaining
    - DOWS (Deploy OGC Web Services) on the DAACs

HEG Functionality

- Conversion:
  - HDF-EOS (Swath or Grid) to a single-band or multi-band GeoTIFF's
  - HDF-EOS Swath to HDF-EOS Grid
  - HDF-EOS Swath or Grid to generic Binary
  - Metadata preservation/creation.
  - Allow file selection from users local storage

HEG Functionality

- Subsetting (spatial, field, band)
- Stitching (mosaicing) + subsetting + Reprojecting
- Reprojection (to UTM, PS, TM, STP, LCC, LAMAZ, GEO, SIN, Albers)
- Subsampling (Subsample stacks)

- Running from Command Line (batch jobs) or with Java-based GUI

HEG Functionality

- Currently supports MODIS, MISR, ASTER, AIRS and AMSR-E products on TERRA and AQUA (>110 products),
- Operable on Sun, SGI, Win, Linux, MAC

- Integrated into ECS Data Pool
  - Reduces the transfer time of HDF-EOS data sets (if subset is requested)
  - Provides the end-user with the exact file required by their application

- User Interface
  - Portable, written in Java.
  - Not dependent on COTS (eg. IDL).

TOOLKIT:
http://newsroom.gsfc.nasa.gov/sdptoolkit/toolkit.html

HDFView:
http://newsroom.gsfc.nasa.gov/sdptoolkit/HDFView/

HDFView_hdfeos_plugin.html

HEG:
http://newsroom.gsfc.nasa.gov/sdptoolkit/HEG/HEGHome.html

NetCDF

NetCDF (**Network Common Data Form**)

- a machine-independent, self-describing, binary data format standard for exchanging scientific data

The project homepage is hosted by the Unidata program at the University Corporation for Atmospheric Research (UCAR).

The data format is "self-describing“: a header describes the layout of the rest of the file, in particular the data arrays, as well as arbitrary file metadata in the form of name/value attributes.

The format is platform independent, with issues such as endianness being addressed in the software libraries. The data arrays are rectangular, not ragged, and stored in a simple and regular fashion that allows efficient subsetting.

endianness refer to the convention used to interpret the bytes making up a data word when those bytes are stored in computer memory.

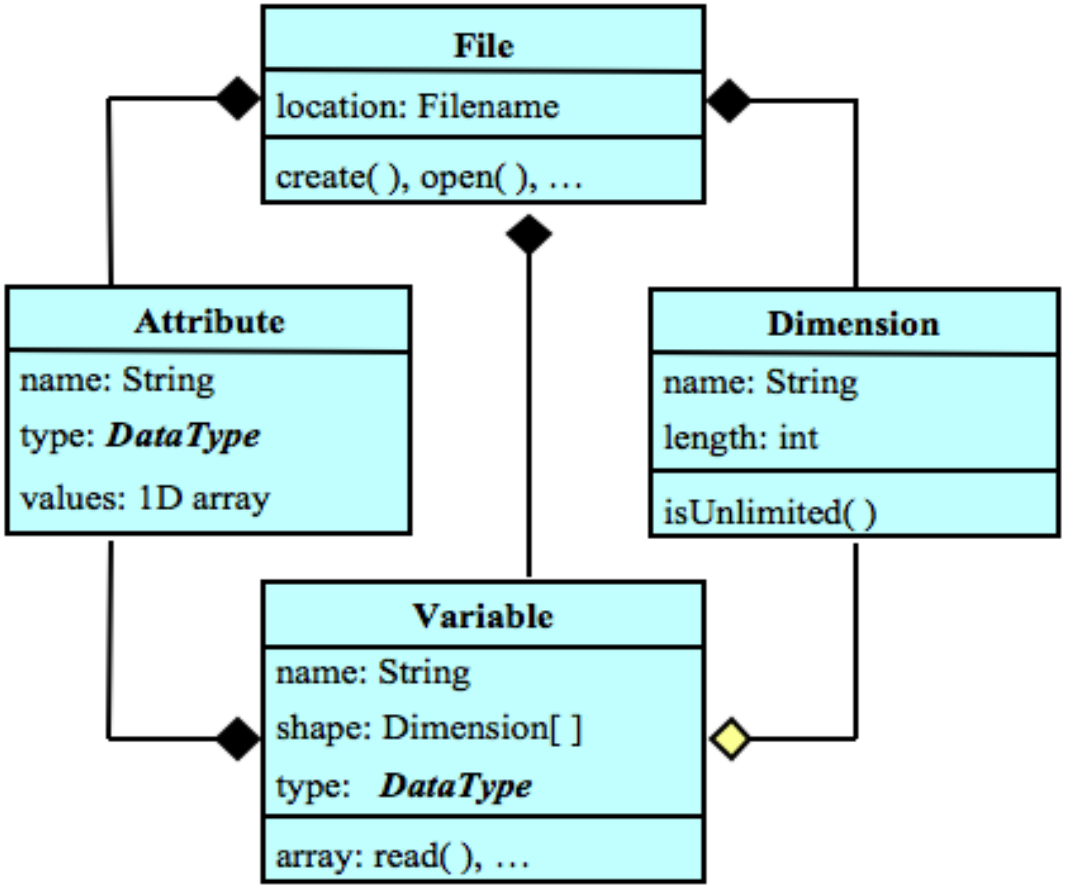
xiaoyang Zhang, 3/19/2021

NetCDF File

Components of a NetCDF file

Netcdf name {
Dimension:
Variables:
Attributes:
Data:
}

NetCDF Classic Model



Variables and attributes have one of six primitive data types.

DataType
char
byte
short
int
float
double

A file has named variables, dimensions, and attributes. Variables also have attributes. Variables may share dimensions, indicating a common grid. One dimension may be of unlimited length.

NetCdf--ncdump

zhangx@hunter TEMP]$ ncdump -h

ncdump [-c|-h] [-v ...] [[-b|-f] [c|f]] [-l len] [-n name] [-p n[,n]] [-k] [-x] [-s] [-t] [-w] file

[-c] Coordinate variable data and header information
[-h] Header information only, no data
[-v var1[,...]] Data for variable(s) <var1>,... only
[-b [c|f]] Brief annotations for C or Fortran indices in data
[-f [c|f]] Full annotations for C or Fortran indices in data
[-l len] Line length maximum in data section (default 80)
[-n name] Name for netCDF (default derived from file name)
[-p n[,n]] Display floating-point values with less precision
[-k] Output kind of netCDF file
[-x] Output XML (NcML) instead of CDL
[-s] Output special (virtual) attributes
[-t] Output time data as date-time strings
[-w] Without client-side caching of variables for DAP URLs
file Name of netCDF file

NetCDF File

To get metadata:

- **ncdump**
- **hdfview**

```
[zhangx@hunter 2013121]$ ncdump -h VGVI.G500m.C01.npp.P2013121_r02c03.nc|more
netcdf VGVI.G500m.C01.npp.P2013121_r02c03 {
dimensions:
        HEIGHT = 3616 ;
        WIDTH = 10000 ;
variables:
        short reflectance_I1(HEIGHT, WIDTH) ;
                reflectance_I1:long_name = "reflectance_I1" ;
                reflectance_I1:coordsys = "cartesian" ;
                reflectance_I1:units = "NONE" ;
                reflectance_I1:range = 0.f, 3.f ;
                reflectance_I1:_FillValue = -1s ;
                reflectance_I1:scale_factor = 0.0001f ;
                reflectance_I1:add_offset = 0.f ;
                reflectance_I1:Remark = "Value= scale_factor * (ScaledInteger - add_offset)" ;
        short reflectance_I2(HEIGHT, WIDTH) ;
                reflectance_I2:long_name = "reflectance_I2" ;
                reflectance_I2:coordsys = "cartesian" ;
                reflectance_I2:units = "NONE" ;
                reflectance_I2:range = 0.f, 3.f ;
                reflectance_I2:_FillValue = -1s ;
                reflectance_I2:scale_factor = 0.0001f ;
                reflectance_I2:add_offset = 0.f ;
                reflectance_I2:Remark = "Value= scale_factor * (ScaledInteger - add_offset)" ;
        short temperature_I5(HEIGHT, WIDTH) ;
        ……
```

NetCDF4

// global attributes:

:Conventions = "CF-1.5" ;
:Metadata_Conventions = "CF-1.5, Unidata Dataset Discovery v1.0" ;
:standard_name_vocabulary = "CF Standard Name Table (version 17, 24 March 2011)" ;
:project = "S-NPP Data Exploitation" ;
:institution = "DOC/NOAA/NESDIS/NDE > S-NPP Data Exploitation, NESDIS, NOAA, U.S. Department of Commerce" ;
:naming_authority = "gov.noaa.nesdis.nde" ;
:satellite_name = "NPP" ;
:instrument_name = "VIIRS" ;
:title = "VHP" ;
:summary = "Vegetation Health Product" ;
:history = "Version 1" ;
:processing_level = "NOAA Level 3" ;
:source = "VIIRS-I1-SDR, VIIRS-I2-SDR, VIIRS-I5-SDR, ICCMO" ;
:references = "TBD" ;
:cdm_data_type = "grid" ;
:geospatial_lat_units = "degrees_north" ;
:geospatial_lon_units = "degrees_east" ;
:creator_name = "DOC/NOAA/NESDIS/STAR > VHP Team, Center for Satellite Applications and Research, NESDIS, NOAA, Department of Commerce" ;
:creator_email = "Felix.kogan@noaa.gov" ;
:creator_url = "http://www.star.nesdis.noaa.gov/smcd/emb/vci/VH/" ;
:publisher_name = "DOC/NOAA/NESDIS/NDE > S-NPP Data Exploitation, NESDIS, NOAA, U.S. Department of Commerce" ;
:publisher_email = "espcoperations@noaa.gov" ;
:publisher_url = "http://projects.osd.noaa.gov/NDE" ;
:version = "v1r1" ;
:PRODUCT_NAME = "VIIRS Daily Map " ;
:PROJECTION = "Plate_Carree" ;
:DATE_BEGIN = "121" ;
:DATE_END = "121" ;
:TIME_BEGIN = "00:00 UTC (use day time data only)" ;
:TIME_END = "23:59 UTC (use day time data only)" ;

}

xiaoyang Zhang, 3/19/2021

NetCDF4

```
// global attributes:
                        :ANCILLARY_FILES =
"File_Configure:viirsgvi.config.500m_v20130718\nFile_Landseamask11:../ancillary/landseamaskll.bit.hdf\nFile_M
etadata_Regions:../ancillary/regi
ons_for_metadata.txt\nFile_Static_Metadata:../ancillary/vgvi_static_metadata.txt\n\n" ;
                        :CONFIGURE_FILE_CONTENT = "# The parameters for running viirs_gvi.exe
\nMachine_Type=          LINUX\nResolution=          0.5 KM\nMax_Solar_Zenith=
BYTE,    NONE, -128,127,0,0, 1\n[END]\n" ;
                        :YEAR = 2013 ;
                        :PERIOD_OF_YEAR = 121 ;
                        :DAYS_PER_PERIOD = 1 ;
                        :geospatial_lat_min = 42.48f ;
                        :geospatial_lon_min = -90.00001f ;
                        :geospatial_lat_max = 58.752f ;
                        :geospatial_lon_max = -45.00002f ;
                        :INPUT_FILES = 1013 ;
                        :FILENAME = "VGVI.G500m.C01.npp.P2013121_r02c03.nc" ;
                        :Max_Solar_Zenith = 80.f ;
                        :Max_Sensor_Zenith = 70.f ;
}
```

ncdump -v Latitude GBBEPx.emis_pm25.001.20140714.nc4|more

```
netcdf GBBEPx.emis_pm25.001.20140714 {
dimensions:
        Time = 1 ;
        Latitude = 721 ;
        Longitude = 1152 ;
variables:
        float Time(Time) ;
                Time:units = "days since 2014-07-14 12:00:00" ;
                Time:time_increment = "240000" ;
                Time:begin_date = "20140714" ;
                Time:begin_time = "120000" ;
        float Latitude(Latitude) ;
                Latitude:long_name = "Latitude" ;
                Latitude:units = "degrees_north" ;
                Latitude:valid_range = -90.f, 90.f ;
                Latitude:scale_factor = 1.f ;
                Latitude:add_offset = 0.f ;
                Latitude:_FillValue = -9.f ;
……..
        // global attributes:
                :PRODUCT_ALGORITHM_VERSION = 1.f ;
                :TIME_RANGE = "day" ;
                :RangeBeginningDate\(YYYY-DOY\) = "2014-14" ;
                :RangeBeginningTime\(UTC-hour\) = "0" ;
                :RangeEndingDate\(YYYY-DOY\) = "2014-14" ;
                :RangeEndingTime\(UTC-hour\) = "23" ;
                :WestBoundingCoordinate\(degree\) = -180.f ;
                :EastBoundingCoordinate\(degree\) = 180.f ;
                :NorthBoundingCoordinate\(degree\) = 90.f ;
                :SouthBoundingCoordinate\(degree\) = -90.f ;
data:
 Latitude = -90, -89.8, -89.5, -89.2, -89, -88.8, -88.5, -88.2, -88, -87.8,
   -87.5, -87.2, -87, -86.8, -86.5, -86.2, -86, -85.8, -85.5, -85.2, -85,
```

Read NetCDF4 Files

Display and analysis tools include: FERRET, GrADS, NCL, MATLAB, and IDL.

```
[zhangx@hunter NETCDF]$ ./XYZ_NetCDF4_Read_2D.exe

USAGE: XYZ_NetCDF4_Read.exe NetCdf4(input_file_name) Variable_name  Data_type
Number_Col  Number_Row  Output_file_name(binary)

Data_type: 1--char (1 byte), 2--short integer (2byte), 4--float (4byte)
```

```
[zhangx@hunter NETCDF]$ ..exe  VGVI.G500m.C01.npp.P2013121_r02c03.nc
reflectance_I1 2 10000 3616 reflectance_I1.bin

NetCdf4(input_file_name) -- VGVI.G500m.C01.npp.P2013121_r02c03.nc
Variable_name -- reflectance_I1
Data_type – 2
Number_Col – 10000
Number_Row -- 3616
Output_file_name(binary) -- reflectance_I1.bin
```

NCDUMP Read HDF5 Files

```
[zhangx@hunter 20200325]$ ncdump  VIIRS_phenology_500m_2014_h17v07_20200325_attribute.h5|more
netcdf VIIRS_phenology_500m_2014_h17v07_20200325_attribute {
group: HDFEOS {
  group: ADDITIONAL {
    group: FILE_ATTRIBUTES {
      } // group FILE_ATTRIBUTES
    } // group ADDITIONAL
  group: GRIDS {
    group: Cycle\ 1 {
      group: Data\ Fields {
        dimensions:
                phony_dim_0 = 2400 ;
        variables:
                ushort Date_Mid_Greenup_Phase_1(phony_dim_0, phony_dim_0) ;
                        Date_Mid_Greenup_Phase_1:units = "day of year" ;
                        Date_Mid_Greenup_Phase_1:scale_factor = 1 ;
                        Date_Mid_Greenup_Phase_1:add_offset = "-366(given year -2000)" ;
                        Date_Mid_Greenup_Phase_1:long_name = "Date at a mid-greenup phase" ;
                        Date_Mid_Greenup_Phase_1:_FillValue = 32767 ;
                        Date_Mid_Greenup_Phase_1:valid_range = 1, 32766 ;
                ushort Date_Mid_Senescence_Phase_1(phony_dim_0, phony_dim_0) ;
                        Date_Mid_Senescence_Phase_1:units = "day of year" ;
                        Date_Mid_Senescence_Phase_1:_FillValue = 32767 ;
                        Date_Mid_Senescence_Phase_1:valid_range = 1, 32766 ;
                        Date_Mid_Senescence_Phase_1:scale_factor = 1 ;
                        Date_Mid_Senescence_Phase_1:add_offset = "-366(given year -2000)" ;
                        Date_Mid_Senescence_Phase_1:long_name = "Date at a mid-senescence phase" ;
```

Hdp dumpsds

```
[x@hunter]$ hdp dumpsds -n Latitude -d -o test.txt -x
gbbgpeo_qfed2.emis_pm25.001.20150503.nc

[x@hunter]$ hdp dumpsds -n Latitude -d -o test1.bin -b
gbbgpeo_qfed2.emis_pm25.001.20150503.nc
```

END