

MODIS Land Data Operational Product Evaluation (LDOPE) Tools

Release 1.4

April 30, 2004

*Land Processes Distributed Active Archive Center (LP DAAC)
USGS EROS Data Center*

Table of Contents

Table of Contents	2
Introduction	3
MODIS LDOPE Tools Capabilities.....	5
comp_sds_hist	7
comp_sds_range	9
comp_sds_stat	11
comp_sds_values	13
convert_l1b_data.....	15
cp_proj_param.....	17
create_mask.....	19
create_sds_ts_stat	22
enlarge_sds	25
geolocation	27
mask_sds	28
math_sds.....	33
mosaic_sds	36
read_l2g.....	39
read_meta	43
read_pixvals.....	45
read_proj_param	47
read_sds_attributes	49
reduce_sds	50
reduce_sds_rank.....	54
sds2bin.....	57
subset_sds.....	59
tile_id.....	61
transpose_sds	62
unpack_sds_bits	64
Appendix A: IRIX Installation	68
Appendix B: Linux Installation.....	71
Appendix C: Solaris Installation	74
Appendix D: Windows Installation.....	76
Appendix E: README for HDF/netCDF 4.1 Distribution	80
Appendix F: LDOPE Tools Command-line Syntax	82
Appendix G: QA Tool testing	83
Appendix H: Understanding unpack_sds_bits.....	100

Introduction

To continue a strong commitment to the US space program, the National Aeronautics and Space Administration (NASA) has undertaken a program of long-term observation, research, and analysis of the Earth's land, oceans, atmosphere and their interactions, including measurements from the Earth Observing System (EOS). The EOS is funded by the NASA Earth Science Enterprise (ESE) Program and has three main components, (i) a coordinated series of Earth-observing satellites, (ii) an advanced data system designed to support the production, archival, and dissemination of satellite derived data products, (iii) teams of scientists who are developing the science algorithms to make the data products. The Moderate Resolution Imaging Spectroradiometer (MODIS) is a key instrument onboard the EOS Terra satellite, successfully launched in December 1999 and again on the Aqua satellite in 2002.

The MODIS Land (MODLAND) Science Team (ST) is funded by NASA to develop the science algorithms and processing software used to generate the MODLAND products. For more information, see the MODLAND special edition of *Remote Sensing of Environment*, November 2002, Vol. 83, Issues 1-2. The standard MODLAND products are sent to Distributed Active Archive Centers (DAACs) for archive and distribution to the user community. The MODLAND cryospheric products are sent to the National Snow and Ice Data Center DAAC and the non-cryospheric land products are sent to the Earth Resources Observation Systems (EROS) Data Center (EDC) DAAC.

The MODIS Land Data Operational Product Evaluation (LDOPE) facility (http://landweb.nascom.nasa.gov/QA_WWW/) was formed to support the MODLAND ST and to provide a coordination mechanism for MODLAND's quality assessment (QA) activities (Roy, D.P., Borak, J.S., Devadiga, S., Wolfe, R.E., Zheng, M., Descloitres, J., 2002, The MODIS Land Product Quality Assessment Approach, *Remote Sensing of Environment*, 83, 62-76). As part of their activities the LDOPE personnel develop and maintain software tools for the manipulation, visualization and analysis of the MODLAND products. The tools have been developed with feedback from the MODLAND ST and incorporate the scientific knowledge, experience and insights gained during the substantial MODLAND product development period. They have been distributed in versioned drops within the ST in response to ST bug fix requests and requests for new functionality.

A subset of the LDOPE QA tools is available to the MODLAND product user community. Several of the tools may also be used to manipulate non-MODLAND HDF-EOS products. The tools are provided as source code and command line executables running on a small number of operating systems and are distributed via the Internet from the EDC DAAC with full documentation and installation instructions. The tools are written in C and may be run at the command line or called from scripts and other packages. They are invoked using an UNIX-like command and argument syntax.

This activity was funded by an unsolicited proposal to NASA, “Data Analysis Tools for the MODIS Land User Community”, 2002, D. P. Roy, S. Devadiga, J. L. Dwyer.

Reference for acknowledgement

Continued tool development is contingent on support from the user community.

Please acknowledge the use of these tools, including use of significant code fragments taken from the source code, with a sentence such as “Software tools provided by the MODIS Land quality assessment group (Roy *et al.* 2002)”.

Roy, D.P., Borak, J.S., Devadiga, S., Wolfe, R.E., Zheng, M., Descloitres, J., 2002, The MODIS Land Product Quality Assessment Approach, *Remote Sensing of Environment*, 83, 62-76.

MODIS LDOPE Tools Capabilities

The tools were released to the public in separate deliveries (December 6, 2002; March 7, 2003; July 30 2003; April 30 2004) and this document was updated with each delivery. The tools are supported for Linux, Irix, Solaris and Windows operating systems. The following table summarizes the tool capabilities.

Tool Name	Description
comp_sds_hist	Print the histogram of SDS values (frequency and values), excluding no-data and missing values, of specified SDSs in any MODIS Land HDF-EOS data product.
comp_sds_range	Print the observed range, excluding no-data and missing values, of specified SDSs in a MODIS Land HDF-EOS data product.
comp_sds_stat	Print summary statistics (mean, standard deviation, minimum, maximum and number of observations) of any SDS, excluding no-data and missing values, of a MODIS Land HDF-EOS data product.
comp_sds_values	Print the unique values found in specified SDSs of a MODIS Land HDF-EOS data product. This tool is most useful for summarizing the data distribution of SDSs that have only a small number of values (e.g., the MODIS Land HDF-EOS snow and fire products).
convert_11b_data	Convert MODIS L1B data to Top of Atmosphere (TOA) reflectance for the MODIS reflective bands and TOA radiance for the MODIS emissive bands and write these to 2D HDF SDS(s) that can be read by commercial of the shelf (COTS) software. The conversion is performed using the scale and off-sets defined in the MODIS L1B product metadata.
cp_proj_param	Copy projection metadata into an HDF file that is defined in the MODIS Land Integerized Sinusoidal projection or Sinusoidal projection. The HDF file may then be reprojected using the EDC reprojection tool. This allows reprojection of MODIS Land L2G/L3/L4 HDF-EOS data product SDS(s) filtered or masked by LDOPE QA tools or other software.
create_mask	Apply relational and logical operators to one or more SDS in one or more MODIS Land HDF-EOS data products to create an output 2D HDF SDS that can be read by conventional COTS. For example, create a binary SDS that shows the pixel locations where only good quality, non-cloudy, 16-day vegetation index values with a land cover type = 3 are present.
create_sds_ts_stat	Create a summary statistic HDF file containing one or more output 2D SDS that describe the mean, standard deviation, minimum, maximum, sum, and number of observations, computed on pixel wise basis from a time series of input MODIS Land HDF-EOS data products.
enlarge_sds	The inverse of companion tool <i>reduce_sds</i> . Simulate finer resolution data by pixel replication.
geolocation	Compute the geographic latitude and longitude of a MODIS Land L2G/L3/L4 pixel coordinate.
mask_sds	Mask one of more SDS of a MODIS Land HDF-EOS data product file and output the SDS values at pixels where the mask criteria are met and output fill values elsewhere.
math_sds	Perform simple arithmetic on two input SDSs of the same or different MODIS Land HDF-EOS data products and output the results to a 2D SDS.
mosaic_sds	Create a spatial mosaic of SDSs from different L3/L4 MODIS Land HDF-EOS data products. Specified SDSs are spatially arranged based either on their geolocation or in a user specified manner.

read_l2g	The MODIS Land L2G HDF-EOS data products store one or more L2 observations for each L2G pixel in a series of layers (that reflect the MODIS orbit overpass and swath sensing geometry) in a compressed run length encoded format. This tool reads the L2G format and writes user specified layers to output 2D HDF Science Data Sets (SDSs) that can be read by conventional COTS.
read_meta	Print the ECS core and archive metadata and SDS attributes of any MODIS Land HDF-EOS data product.
read_pixvals	Read MODIS Land HDF-EOS data product values at specified pixel locations.
read_proj_param	Read the projection parameter information of a L2G/L3/L4 MODIS Land HDF-EOS data product. This information is needed to project non-MODLAND data into registration with a geolocated MODIS Land HDF-EOS data product.
read_sds_attributes	Print the attributes of one or more SDS of MODIS Land HDF-EOS data products.
reduce_sds	Generate reduced spatial resolution MODIS Land HDF-EOS data product SDSs by sub-sampling or averaging. Handle the MODLAND product no-data and missing values. This may be used to reduce data volumes, and to quickly enable analysis of the different MODIS Land HDF-EOS data product spatial resolutions (250m, 500m, 1km), or to enable comparison with other coarser spatial resolution data sets.
reduce_sds_rank	Several MODIS Land HDF-EOS data products (e.g., MOD43) and related MODIS products (e.g., MOD35) contain multidimensional SDSs. This tool converts multidimensional (3D or 4D) SDS to a series of 2D HDF SDSs that can be read by conventional COTS.
sds2bin	Convert an SDS of any MODIS Land HDF-EOS data product to a flat binary image format.
subset_sds	Create spatial subset SDS(s) from one or more SDS of a MODIS Land HDF-EOS data product.
tile_id	Compute the MODIS Land L2G/L3/L4 tile id for a given latitude and longitude. This tool identifies the MODIS Land tile that corresponds to a known geographic location.
transpose_sds	Transpose one or more SDS in a MODIS Land HDF-EOS data product by rotating the SDS 180 degree in clockwise direction This tool enables qualitative comparison of MODIS Aqua and Terra Level 2 or Level 1 granules.
unpack_sds_bits	The MODIS Land HDF-EOS data product per-pixel QA information and other information such as the land-sea mask, logical criteria used by the algorithm, and cloud state are stored in an efficient bit encoded manner. This tool decodes requested bit fields and writes them to 2D HDF SDSs that can be read by conventional COTS.

comp_sds_hist

NAME

comp_sds_hist – Print histogram of data values in one or more SDS of a MODIS Land HDF-EOS data product.

SYNOPSIS

comp_sds_hist [-help] [filename]

comp_sds_hist [-sds=<SDS_name1>[,<SDS_name2>. . .]] [-layer]
[-range=<min,max>] filename

DESCRIPTION

Compute histogram of data values in one or more SDS of a MODIS Land HDF-EOS data product. The histogram may be computed for a user specified range of SDS values. The output to stdout includes the SDS name, dimension size, fill value and a list of SDS values.

If an SDS is 3D or 4D, then the tool can optionally output the histogram for each layer/slice of the 3D/4D SDS.

The tool command arguments can be specified in any order.

ARGUMENTS

- help Display this help message. If the input filename is specified with this option, then the names of all the SDS in the file are displayed.
- sds=<SDS_list> List of SDSs to read. SDS names are separated by commas with no space. By default sds values are printed for all SDSs in the input file.
- layer Compute histogram for every layer/slice separately for a 3D/4D SDS.
- range=<min,max> Histogram range (minimum and maximum values). Default is set to valid range of the SDS. Fill value is counted separately. If valid range attribute is not available the range of the SDS data type is used as the limit. The range value for various data type is shown below
 - INT8: (-128, 127) UINT8: (0, 255)
 - INT16: (-32768, 32767) UINT16: (0, 65535)
 - INT32: (-2147483648, 2147483647) UINT32: (0, 4294967295)
 - FLOAT32: UNDEFINED.

For float data type the histogram is computed after converting the float values to their closest integer.

Filename input filenames

EXAMPLE

```
comp_sds_hist -sds=sur_refl_b01
MOD09A1.A2001033.h08v05.001.2001166175830.hdf
```

```
comp_sds_hist -layer -sds=Surface_Refl -range=0,10000
MODAGAGG.A2000065.h13v02.002.2000075160322.hdf
```

AUTHOR

Code: S. Devadiga and Yi Zhang
Documentation: S. Devadiga and D. Roy

Version 1.0, 04/05/2004

comp_sds_range

NAME

comp_sds_range - Print the range of data values in one or more SDS of a MODIS Land HDF-EOS data product.

SYNOPSIS

```
comp_sds_range -help [filename]
```

```
comp_sds_range [-sds=<SDS_name1>[,<SDS_name2>. .]] [-layer] [-scale] filename
```

DESCRIPTION

Read the minimum and maximum value that is not a fill value in one or more user specified SDS(s) of a MODIS Land HDF-EOS data product and write the output to stdout. If an SDS is 3D or 4D, then the tool can optionally output the range information for each layer/slice of the 3D/4D SDS.

Output contains SDS name, layer id, minimum, maximum, fill value and out of range indicator for each input SDS. Layer id is provided only for 3D and 4D SDS(s). If an SDS contains only fill values, then the minimum and maximum field will contain a string of *. SDS attributes scale_factor and add_offset are printed when the -scale option is used.

The tool command arguments can be specified in any order.

ARGUMENTS

- | | |
|------------------|---|
| -help [filename] | Display this help message. If the input filename is specified with this option, then the names of all the SDS in the file are displayed. |
| -sds=<SDS_list> | List of SDSs to read. SDS names are separated by commas with no space. By default range information are printed for all SDSs in the input file. |
| -layer | Output range information for each layer separately if the SDS is a 3D/4D SDS. |
| -scale | Output the SDS attributes scale and offset. |
| filename | Input filename. |

EXAMPLES

```
comp_sds_range -sds=sur_refl_b01,sur_refl_b03,sur_refl_b04 -scale MOD09A1.A1999049.h13v13.001.1999268235744.hdf
```

comp_sds_range -layer
MOD43B1.A1999064.h10v04.001.1999356235305.hdf

AUTHOR

Code: S. Devadiga and Yi Zhang
Documentation: S. Devadiga and D. Roy

Version 1.0, 04/05/2004

comp_sds_stat

NAME

comp_sds_stat – Print summary statistics of data values in one or more SDS of a MODIS Land HDF-EOS data product.

SYNOPSIS

```
comp_sds_stat -help [filename]
```

```
comp_sds_stat [-sds=<SDS_name1>[,<SDS_name2>. . .]]  
               [-class=<classification filename>,<classification SDSname>]  
               [-reg=<r_min,r_max,c_min,c_max>] filename
```

DESCRIPTION

Print summary statistics of data values in one or more SDS of a MODIS Land HDF-EOS data product. Statistics include number of pixels, mean, standard deviation, minimum and maximum values of pixels in the SDS.

Statistics can be computed for all pixels in the input SDS(s) or a spatial subset of the SDS(s) specified by the pixel row-column range. Statistics for each SDS are output in a separate row.

A classification file containing a class label SDS can be used to compute the statistics of input SDS pixels for each class. Statistics of pixels belonging to each class are output on separate rows. The classification SDS must have the same dimensions as the input SDS(s).

This tool supports 2D/3D/4D SDSs.

The tool command arguments can be specified in any order.

ARGUMENTS

-help Display this help message. If the input filename is specified with this option, then the names of all the SDS in the file are displayed.

-sds=<SDS_list> List of SDSs. The SDS names are separated by commas with no space. By default all SDSs are processed maintaining the input SDS interleaving.

To process a specific layer of a 3D SDS, specify the element number of the third dimension as a dot extension of the SDS name: sds_name.n (e.g., sur_refl_b02.1 = the layer defined by the 1st element of the 3rd dimension of the 3D SDS sur_refl_b02).

To process a specific layer of a 4D SDS, specify the higher dimension element number(s) as a dot extension of the SDS name: sds_name.n.m (e.g., Surface_Refl.1.2 = the layer defined by the 1st element of the 3rd dimension and the 2nd element of the 4th dimension of the 4D SDS Surface_Refl).

Note that wildcards and ranges of element values may be specified as sds_name.* and as sds_name.n1-n2.m respectively.

`-class=<classification file,SDSname>`
Input classification file and SDS such as land cover, biome file etc. The classification SDS should have pixel values in the range 0-255 (i.e. be an UNSIGNED INT data type).

`-reg=<r_min,r_max,c_min,c_max>`
Pixel row and column range of spatial subset.

filename Input filename

EXAMPLES

```
comp_sds_stat
-class=MOD12Q1.A1990001.h09v05.002.1999325030448.hdf,
Land_Cover_Type_2
MOD11A2.A2003281.h09v05.004.2003294184557.hdf
```

```
comp_sds_stat -sds=BRDF_Albedo_Parameters.1.2
-class=MOD12Q1_5KM.A1990001.h26v04.002.1999325060200.hdf,
Land_Cover_Type_2
MOD43B1.A2001193.h26v04.004.2003134105311.hdf
```

```
comp_sds_stat -reg=0,100,50,150
-class=LCS.A2001001.h20v05.hdf,Land_Cover_Type_1_Secondary
MOD09A1.A2001201.h20v05.003.2002281080105.hdf
```

AUTHOR

Code: S. Devadiga and Yi Zhang
Documentation: S. Devadiga and D. Roy

Version 1.0, 04/05/2004

comp_sds_values

NAME

comp_sds_values - Print a list of the unique data values present in one or more SDS of a MODIS Land HDF-EOS data product and optionally produce an ENVI compatible density slice color table.

SYNOPSIS

```
comp_sds_values -help [filename]
```

```
comp_sds_values [-sds=<SDS_name1>[,<SDS_name2>. . ]] [-layer]  
[-clr=<nc,f1,f2>] filename
```

DESCRIPTION

Read the unique values present in one or more SDS of a MODIS Land HDF-EOS data product and write the output to stdout. Output contains SDS name, dimension size, fill value and a list of SDS values. If an SDS is 3D or 4D, then the tool can optionally output the list of values for each layer/slice of the 3D/4D SDS.

Tool can optionally produce an ENVI compatible density slice color table for the range of values found in the SDS. The output color table is an ASCII file containing requested number of colors at equal intervals selected from the input color table covering the full range of SDS values. If the number of unique SDS values are less than or equal to specified number of output colors then distinct colors are assigned to distinct SDS values, otherwise the SDS value range is divided into input number of colors.

Warning: This tool is mainly for use when an SDS contains a small number of different values such as QA SDS, land cover SDS etc. The maximum number of output values is 16000.

The tool command arguments can be specified in any order.

ARGUMENTS

- | | |
|-----------------|--|
| -help | Display this help message. If the input filename is specified with this option, then the names of all the SDS in the file are displayed. |
| -sds=<SDS_list> | List of SDSs to read. SDS names are separated by commas with no space. By default sds values are printed for all SDSs in the input file. |
| -layer | Output sds values for each layer separately if the SDS is a 3D/4D SDS. |

-clr=<nc,f1,f2> Number of output colors, input file f1 containing full range color table, and output file f2 containing ENVI compatible density slice color table.

filename Input filename.

EXAMPLES

```
comp_sds_values -sds="500m Reflectance Band Quality"  
MOD09.A2000064.1500.002.2000069220633.hdf
```

```
comp_sds_values -sds=Band_QC -layer  
MODAGAGG.A2000065.h13v02.002.2000075160322.hdf
```

AUTHOR

Code: S. Devadiga and Yi Zhang
Documentation: S. Devadiga and D. Roy

Version 1.0, 04/05/2004

convert_l1b_data

NAME

convert_l1bdata - Convert the L1B data values of one or more reflective band SDS to top of atmosphere (TOA) reflectance and one or more emissive band SDS to radiance using the scale and offsets defined in the MODIS L1B data product metadata.

SYNOPSIS

```
convert_l1b_data -help [filename]
```

```
convert_l1b_data -of=output_file [-bn=1[,2.. .] ] [-meta] filename
```

DESCRIPTION

Convert the MODIS L1B data values of one or more reflectance bands to TOA reflectance and one or more emissive bands to TOA radiance. Output is an HDF file containing one SDS for each input L1B band. All requested bands are output as 2D SDS with a FLOAT data type.

The TOA reflectance and radiance are computed by applying the original scale and offset defined in the input L1B metadata. The output values are rescaled with a new scale value and zero offset value (this is reflected in the output SDS attribute).

ARGUMENTS

- | | |
|----------------|---|
| -help | Display this help message. If the input filename is specified with this option, then the names of all the SDS in the file are displayed. |
| -of=<filename> | Output filename. |
| -bn=<1,2.. .N> | L1B band numbers separated by comma. By default all bands in the input L1B granule are processed. No bands are output if none of the band numbers input match the band numbers in the input L1B granule.

Note that MOD02QKM contains only bands 1 and 2, MOD02HKM contains bands 1-7, and MOD021KM contains all MODIS L1B bands. |
| -meta | Copy metadata from the input file to the output file. |
| filename | Input filename. |

EXAMPLES

```
conv_l1b_data -bn=1,2
```

-of=TOAREFL_b1-2.A2000366.0525.003.2001296055758.hdf
MOD02QKM.A2000366.0525.003.2001296055758.hdf

conv_l1b_data -bn=3,4,5,6,7
-of=TOAREFL_b3-7.A2000366.0525.003.2001296055758.hdf
MOD02HKM.A2000366.0525.003.2001296055758.hdf

AUTHOR

Code: S. Devadiga and Yi Zhang
Documentation: S. Devadiga and D. Roy

Version 1.0, 04/05/2004

cp_proj_param

NAME

cp_proj_param - copy projection parameter metadata to an HDF file.

SYNOPSIS

```
cp_proj_param -help
cp_proj_param -of=<output filename> [-ref=<reference>] [-tile=<tile_id(s)>]
[-proj=<projection type>] filename
```

DESCRIPTION

Certain HDF files, created for example using the LDOPE QA tools or other software, may not contain the HDF-EOS metadata that identify the projection type and the geographic extent of the data. This tool creates an output HDF-EOS file by copying all the SDS(s) from the input HDF file and adding the projection parameters and bounding rectangular coordinates to the output file. The output file may then be reprojected using the MODIS reprojection tool available from EDC-DAAC.

This tool is applicable only to geolocated L2G/L3/L4 MODIS Land data products or mosaics of these products. There are two ways to use this tool:

1) The projection metadata may be copied from a reference L2G/L3/L4 MODIS Land HDF-EOS data product. The reference product should have the same geographical extent as the input file. This method only works for single files (not mosaics).

2) The projection metadata may be copied by specifying the MODIS Land HDF-EOS data product tile id and the projection type. Multiple tile ids may be specified if the input file is a mosaic of two or more files.

The tool command arguments can be specified in any order.

Note: cp_proj_param only works on tile mosaics (see *mosaic_sds*) that have the same number of horizontal and vertical tiles (e.g., produced by *mosaic_sds -gr=5,5*).

Note: This version of cp_proj_param now works for all versions of the MODIS reprojection tool. The bounding rectangular coordinates are copied to the output file archive metadata without using the SDP toolkit.

Note: Use of the -tile option may introduce a geolocation imprecision of no more than approximately 50m (i.e., within the MODIS 1σ geolocation specification).

For more information in the MODIS Land tile grid see http://landweb.nascom.nasa.gov/cgi-bin/QA_WWW/newPage.cgi?fileName=prod_format

ARGUMENTS

-help	Display this help message.
-of=<filename>	Output filename.
-ref=<reference file>	Reference L2G/L3/L4 MODIS Land data product filename. If this is specified then do not use the -tile and -proj options.
-tile=<tile id(s)>	One or more tile id(s) separated by commas. The tile id is of the form hxxvyy where xx and yy are the two digit horizontal and vertical MODIS Land data product tile numbers.
-proj=<projection type>	One of the following projection types: ISIN (Integerized Sinusoidal) or SIN (Sinusoidal)
filename	Input filename

EXAMPLES

```
cp_proj_param  
-ref=MOD09A1.A2001145.h20v10.003.2001214125825.hdf  
-of=zoo_coords.hdf zoo.hdf
```

```
cp_proj_param -tile=h20v10 -proj=ISIN  
-of= zoo_coords.hdf zoo.hdf
```

```
cp_proj_param -tile=h20v10,h21v10,h20v11,h21v11 -proj=ISIN  
-of=Coords_4_Mosaiced_tiles.hdf Four_Mosaiced_tiles.hdf
```

```
resample -p your.prm -i Coords_4_Mosaiced_tiles.hdf -o  
projected_4_Mosaiced_tiles.tif
```

AUTHOR

Code: S. Devadiga and Yi Zhang
Documentation: S. Devadiga and D. Roy

Version 1.1, 04/05/2004

create_mask

NAME

create_mask - Create an output mask SDS containing two values, an user defined 'ON' value at pixels where the masking criteria are satisfied and an 'OFF' value elsewhere. The mask criteria are specified using relational and logical operators applied to the SDS of the same or different L2/L3/L4 MODIS Land HDF-EOS data products.

SYNOPSIS

```
create_mask -help [filename]
```

```
create_mask -of=<output filename> -fill=<mask fill value>  
            -mask=<mask1>[,AND|OR,<mask2>][,...] [-on=<output ON value>]  
            [-off=<output OFF value>]
```

where mask_n = <filename>,<SDSname>,<bit_numbers operator
bit_values>

DESCRIPTION

Create an output masking SDS containing two values: a user defined 'ON' value at pixels where the masking criteria are satisfied and an 'OFF' value elsewhere.

The mask criteria are specified using relational and logical operators applied to the SDS of the same or different L2/L3/L4 MODIS Land HDF-EOS data products. SDS(s) used to define the masking criteria must have the same or lower resolution as the input file SDS(s) to be masked.

The mask criteria are defined by a combination of one or more individual masks. Each mask is defined by testing SDS bits against bit values using a relational operator. Testing using a decimal value is also supported. Different masks are combined using the logical "AND" or "OR" operators

Masking criteria cannot be applied at pixels where one or more of the mask SDS(s) have fill values. A mask fill value will be output at these pixels. The mask fill value may be optionally specified or will be set to 255 by default.

This tool supports 2D/3D/4D SDSs. Note, only a two dimensional (2D) SDS or a 2D layer of a 3D/4D SDS can be used to make a mask.

The tool command arguments can be specified in any order.

ARGUMENTS

-help [filename] Display this help message. If a filename is specified with this option, then the names of all the SDSs in the file are displayed.

-of=<filename> Output filename.

-fill=<mask fill value>

User specified mask fill value. The fill value is stored as the SDS attribute "Mask_FillValue" in the output file and is printed to stdout when the tool runs. If a mask fill value is not specified then an arbitrary value not equal to the input file SDS fill value and not equal to a valid input file SDS value is assigned.

-meta Copy metadata from the input file to the output file.

-mask=<mask1>[,AND|OR,<mask2>[,...]]

where mask_n=< filename>,<SDSname>,<bit_numbers operator bit_values>

Define a mask from one or more individual masks combined using the logical operators "AND" or "OR".

Each individual mask consists of:

-filename= MODIS Land product file

-SDSname= name of an SDS in the file

-bit_numbers= a list or range of SDS bits

-operator= relational operator (>, <, <=, >=, ==, !=)

-bit_values= bit values that are tested against

The bits in bit_numbers are specified by the lower bit followed by the higher bit and the bit_values are specified in the reverse order. For example, 0-2,4==0101 signifies bits 4,2,1,0==0101.

If the bit_numbers are omitted, then the bit_values are parsed as a decimal value. This provides a convenient way to refer to a specific value, instead of a list of bits. For example, -mask=file,SDS,>=200 makes a mask where only the SDS values in the file greater than or equal to 200 are considered.

If several masks are combined together then "*" may be used in place of the filename and/or SDS name to specify the same filename and/or SDS name used in the previous mask. For example,

`-mask=file1,SDS1,0-2,4==0101,AND,*,*,4-5==10`

To specify a 3D SDS layer write the element number of the third dimension as a dot extension of the SDS name: `sds_name.n` (e.g. `,sur_refl_b02.1` = the layer defined by the 1st element of the 3rd dimension of the 3D SDS `sur_refl_b02`).

To specify a 4D SDS layer write the higher dimension element number(s) as a dot extension of the SDS name: `sds_name.n.m` (e.g., `Surface_Refl.1.2` = the layer defined by the 1st element of the 3rd dimension and 2nd element of the 4th dimension of the 4D SDS `Surface_Refl`).

`-on=<ON value>` User defined output ON value.

`-off=<OFF value>` User defined output OFF value.

EXAMPLES

```
create_mask -of=land_mask.hdf -on=255 -off=0
-mask="MOD09A1.A1996214.h12v04.002.hdf,sur_refl_state_500m,
3-5==001"
```

```
create_mask -of=clear_land.hdf -on=100 -off=0
-mask="MOD09A1.A1996214.h12v04.002.hdf,sur_refl_state_500m,
3-5==001,AND,*,*,01==00"
```

```
create_mask -of=mod35_cloudy_land.hdf -on=1 -off=0
-mask="MOD35_L2.A1996213.1024.002.hdf,Cloud_Mask.1,
1-2==00,AND,*,*,6-7==11"
```

```
create_mask -of=agg_b01_obs1_land_qc.hdf -on=1 -off=0
-mask="MODAGAGG.A1996214.h12v04.001.hdf,Band_QC.1.1,2-
5==1100,AND,*, Aggregate_QC.1.1,3-5==001"
```

AUTHOR

Code: S. Devadiga and Yi Zhang

Documentation: S. Devadiga and D. Roy

Version 1.0, 04/05/2004

create_sds_ts_stat

NAME

create_sds_ts_stat - Create a summary statistic HDF file containing one or more output 2D SDS that describe the mean, standard deviation, minimum, maximum, sum, and number of observations, computed on pixel wise basis from a time series of input MODIS Land HDF-EOS data products.

SYNOPSIS

```
create_sds_ts_stat [-help] [filename]
```

```
create_sds_ts_stat -  
sds=<SDS_name,sds_minval,sds_maxval,f_nop_in,f_nop_out,dt>  
-of=<filename> -param=[avg][,std][,min][,max][,npix][,sum] f1 [ f2 .. . fn]
```

DESCRIPTION

Compute statistics of SDS values at each pixel from a set of input MODIS Land HDF-EOS data products. For example, the tool can be used to compute time series statistics of SDS values when the input files are a time series sequence of files for the same geographic area i.e., files that have the same tile id.

The output HDF file contains SDS of the following statistics: mean, standard deviation, minimum, maximum, number of pixels and sum of SDS values. By default all parameters are output.

The user can specify the range of the SDS values to be considered for computing statistics and a fill value to be excluded from consideration. Two or more SDSs are processed by repeating the option `-sds` option with the SDS names. An input file is ignored if the user specified SDS is missing in that file.

ARGUMENTS

`-help` Print this help message, If the input filename is specified with this option, then the names of all the SDS in the file are displayed.

```
-sds=<SDS_name,sds_minval,sds_maxval,f_nop_in,f_nop_out,dt>
```

Input SDS name, minimum and maximum of SDS values to be considered, no operation fill value in the input SDS, no operation fill value in the output SDS, output data type.

Input SDS pixel values equal to 'f_nop_in' are not considered. If no values are available for computing the statistics at a particular pixel then 'f_nop_out' is written to the pixel in the output SDS.

Each of the parameters except the SDS name can be replaced by * to assign default values defined as:

sds_min,sds_max: valid range of input SDS.
f_nop_in: input SDS fill value
f_nop_out: f_nop_in
dt: input SDS data type.

Valid values for output data type 'dt' are INT8, UINT8, INT16, UINT16, INT32, UINT32, FLOAT32

The output data type of SDS number of pixels is always INT16.

To process a specific layer of a 3D SDS specify the element number of the third dimension as a dot extension of the SDS name: sds_name.n (e.g., sur_refl_b02.1 = the layer defined by the 1st element of the 3rd dimension of the 3D SDS sur_refl_b02).

To process a specific layer of a 4D SDS, specify the higher dimension element number(s) as a dot extension of the SDS name: sds_name.n.m (e.g., Surface_Refl.1.2 = the layer defined by the 1st element of the 3rd dimension and the 2nd element of the 4th dimension of the 4D SDS Surface_Refl).

Note that wildcards and ranges of element values may be specified as sds_name.* and as sds_name.n1-n2.m respectively.

-param=avg,std,min,max,npix,sum

Output statistical parameters include average, standard deviation, minimum, maximum, number of pixels, and sum of SDS values at a pixel. One or more of these parameter may be specified in any order. If this argument is unspecified then all these parameters are output.

-of=<filename> Output filename

EXAMPLES

```
create_sds_ts_stat -sds="LST_Day_1km,*,*,*,*"
-param=min,max,avg
-of=ts_stat_2days_MOD11A1.A2002194.h30v11.003.hdf
MOD11A1.A2002194.h30v11.003.2002199095914.hdf
MOD11A1.A2002195.h30v11.003.20021200015759.hdf
```

```
create_sds_ts_stat -sds=" LST_Day_1km,*,*,*,*"
-sds=" LST_Night_1km,*,*,*,*" -param=max,min,avg,std
-of=ts_stat_2days_MOD11A1.A2002194.h30v11.003.hdf
MOD11A1.A2002194.h30v11.003.2002199095914.hdf
MOD11A1.A2002194.h30v11.003.2002200015759.hdf
```

AUTHOR

Code: S. Devadiga and Yi Zhang
Documentation: S. Devadiga and D. Roy

Version 1.0, 04/05/04

enlarge_sds

NAME

enlarge_sds – enlarge the spatial dimensions of one or more SDS from a MODIS Land HDF-EOS data product by pixel replication.

SYNOPSIS

enlarge_sds -help [filename]

enlarge_sds -of=<output filename> -sf=<scale factor>
[-sds=<SDSname1>[<,SDSname2>...]] [-meta] filename

DESCRIPTION

Each pixel value at (i, j) in the input SDS is duplicated from (sf*i, sf*j) to (sf*i + sf - 1, sf*j + sf - 1) pixels in the output SDS. The enlargement scale factor (sf) must be a non-zero positive integer. The output is an HDF-EOS file. One or more SDSs may be enlarged.

This tool can be used to create fine resolution data set from a coarse resolution data set, e.g., a 500m resolution data set can be created from an input 1km resolution data set using a scale factor of 2. The output dataset will have blocky structure due to pixel replication.

This tool is complemented by the tool *reduce_sds*.

This tool supports 2D/3D/4D SDS(s).

The tool command arguments can be specified in any order.

ARGUMENTS

- help [filename] Display this help message. If the input filename is specified with this option, then the names of all the SDS in the file are displayed.
- of=<filename> Output filename.
- sf=<factor> Scale factor (a non-zero positive integer).
- sds=<SDS list> List of SDS to enlarge. SDS names are separated by commas with no space. By default all SDSs are processed maintaining the input SDS interleaving.

To process a specific layer of a 3D SDS specify the element number of the third dimension as a dot extension of the SDS name: sds_name.n (e.g., sur_refl_b02.1 = the layer defined by the 1st element of the 3rd dimension of the 3D SDS sur_refl_b02).

To process a specific layer of a 4D SDS, specify the higher dimension element number(s) as a dot extension of the SDS name: sds_name.n.m (e.g., Surface_Refl.1.2 = the layer defined by the 1st element of the 3rd dimension and the 2nd element of the 4th dimension of the 4D SDS Surface_Refl).

Note that wildcards and ranges of element values may be specified as sds_name.* and as sds_name.n1-n2.m respectively.

-meta Copy metadata from input to output file.
filename Input filename.

EXAMPLES

```
enlarge_sds -help MYD021KM.A2002189.0350.003.2002191000652.hdf
```

```
enlarge_sds -sds=EV_500_Aggr1km_RefSB.2 -sf=2 -of=refsb_x2.hdf  
MOD021KM.A1996213.1024.002.hdf
```

```
enlarge_sds -sds="EV_500_Aggr1km_RefSB.2-4" -sf=4 -of=refsb_x4.hdf  
MOD021KM.A002189.0350.003.2002191000652.hdf
```

```
enlarge_sds -sf=4 -sds="Cloud_Mask.*,Quality_Assurance" -  
of=myd35_x4.hdf -meta  
MYD35_L2.A2002189.2040.003.2002191125354.hdf
```

```
enlarge_sds -sf=4 -of=refsb_1kmtto250m.hdf -sds="EV_1KM_RefSB.4-6"  
MYD021KM.A2002189.0350.003.2002191000652.hdf
```

```
enlarge_sds -sds="BRDF_Albedo_Parameters.*.2" -sf=2 -meta -  
of=brdf_x2.hdf  
MYD43B1.A2002177.h11v11.003.2002210233848.hdf
```

{Note: This example enlarges selected layers of a 4D SDS. All the layers defined by all the elements of the 3rd dimension and the 2nd element of the 4th dimension of the SDS BRDF_Albedo_parameters are enlarged using a scale factor of 2. The metadata is copied from the input file to the output file. }

AUTHOR

Code: S. Devadiga and Yi Zhang
Documentation: S. Devadiga and D. Roy

Version 1.0, 08/08/2002

geolocation

NAME

geolocation - Compute the geolocation of a pixel in a L3 MODIS Land HDF-EOS data product.

SYNOPSIS

geolocation -help

geolocation -proj=<SIN|ISIN> -xy=<column,row> [-res=qkm|hkm|1km]
[-hv=<horizontal,vertical>] [filename|local granule id]

DESCRIPTION

Compute the geolocation of a pixel in a L2G/L3/L4 MODIS Land data product. Pixel location is specified as row column number at a specified resolution and the tile id is either explicitly input or is implied by the local granule id of the L2G/L3/L4 MODIS Land data product.

The tool command arguments can be specified in any order.

ARGUMENTS

-help	Display this help message
-res=qkm hkm 1km	Reference resolution (default = 1km)
-xy=<column,row>	Row and column location of pixel. (valid: 0-1199 for 1km, 0-2399 for hkm and 0-4799 for qkm)
-hv=<horizontal,vertical>	Horizontal and vertical tile number. Valid for horizontal: 0-35, vertical: 0-17
filename	MODIS Land data product filename/local granule id of a L2G L3 L4 product.

EXAMPLES

geolocation -xy=750,500 -hv=12,4

geolocation -xy=750,500
MOD09GHK.A1997223.h17v03.001.1999153204552.hdf

AUTHOR

Code: S. Devadiga and Yi Zhang
Documentation: S. Devadiga and D. Roy

Version 1.0, 04/05/2004

mask_sds

NAME

mask_sds – Mask one or more SDS of a MODIS Land HDF-EOS data product and output the SDS values at pixels where the mask criteria are met and output fill values elsewhere. The mask criteria are specified using relational and logical operators applied to the SDS of the same or different L2/L3/L4 MODIS Land HDF-EOS data products.

SYNOPSIS

```
mask_sds -help [filename]
mask_sds -of=<output filename>
           -sds=<SDSname1>[,<SDSname2>[,...]]>
           [-fill=<mask fill value>]
           -mask=<mask1>[,<AND|OR,<mask2>[,...]] [-meta] filename
where maskn=< filename>,<SDSname>,<bit_numbers operator bit_values>
```

DESCRIPTION

Mask one or more SDS of a MODIS Land HDF-EOS data product and output the SDS values at pixels where the mask criteria are met and output mask fill values elsewhere.

The mask criteria are specified using relational and logical operators applied to the SDS of the same or different L2/L3/L4 MODIS Land data products. The SDS(s) used to define the masking criteria must have the same or lower spatial resolution as the input file SDS(s) to be masked.

The mask criteria are defined by a combination of one or more individual masks. Each mask is defined by testing SDS bits against bit values using a relational operator. Testing using a decimal value is also supported. Different masks are combined using the logical “AND” or “OR” operators

Pixel values that do not meet the masking criteria are assigned a mask fill value. The mask fill value may be optionally specified or will be set automatically. If the specified mask fill value is set equal to the input SDS fill value or to a valid input file SDS value then the tool will issue a warning message and request another mask fill value.

If pixels in the input file SDS(s) have fill values they cannot be masked. The input file SDS fill value will be output at these pixels.

If pixels in the SDS(s) used to define the masking criteria have fill values then the masking cannot be performed. The mask fill value will be output at these pixels.

This tool supports 2D/3D/4D SDSs. Note, only a two dimensional (2D) SDS or a 2D layer of a 3D/4D SDS can be used to make a mask.

The tool command arguments can be specified in any order.

ARGUMENTS

-help [filename] Display this help message. If the input filename is specified with this option, then the names of all the SDSs in the file are displayed.

-of=<filename> Output filename.

-sds=<SDS list> List of SDSs present in the input file to be masked and written to the output file. SDS names must be specified separated by commas with no space.

To mask a specific layer of a 3D SDS specify the element number of the third dimension as a dot extension of the SDS name: sds_name.n (e.g., sur_refl_b02.1 = the layer defined by the 1st element of the 3rd dimension of the 3D SDS sur_refl_b02). A wildcard may be used as sds_name.*

To mask a specific layer of a 4D SDS, specify the higher dimension element number(s) as a dot extension of the SDS name: sds_name.n.m (e.g., Surface_Refl.1.2 = the layer defined by the 1st element of the 3rd dimension and 2nd element of the 4th dimension of the 4D SDS Surface_Refl). A range of element values may be specified as sds_name.n1-n2.m

-fill=<mask fill value> User specified mask fill value. The fill value is stored as the SDS attribute "Mask_FillValue" in the output file and is printed to stdout when the tool runs. If a mask fill value is not specified then an arbitrary value not equal to the input file SDS fill value and not equal to a valid input file SDS value is assigned.

-meta Copy metadata from the input file to the output file.

- mask=<mask1>[,AND|OR,<mask2>[,...]]
where mask_n=< filename>,<SDSname>,<bit_numbers operator bit_values>

Define a mask from one or more individual masks combined using the logical operators “AND” or “OR”.

Each individual mask consists of:

- filename= MODIS Land product file
- SDSname= name of an SDS in the file
- bit_numbers= a list or range of SDS bits
- operator= relational operator (>, <, <=, >=, ==, !=)
- bit_values= bit values that are tested against

The bits in bit_numbers are specified by the lower bit followed by the higher bit and the bit_values are specified in the reverse order. For example, 0-2,4==0101 signifies bits 4,2,1,0==0101.

If the bit_numbers are omitted, then the bit_values are parsed as a decimal value. This provides a convenient way to refer to a specific value, instead of a list of bits. For example, -mask=file,SDS,>=200 makes a mask where only the SDS values in the file greater than or equal to 200 are considered.

If several masks are combined together then “*” may be used in place of the filename and/or SDS name to specify the same filename and/or SDS name used in the previous mask. For example,
-mask=file1,SDS1,0-2,4==0101,AND,*,*,4-5==10

To specify a 3D SDS layer write the element number of the third dimension as a dot extension of the SDS name: sds_name.n (e.g. ,sur_refl_b02.1 = the layer defined by the 1st element of the 3rd dimension of the 3D SDS sur_refl_b02).

To specify a 4D SDS layer write the higher dimension element number(s) as a dot extension of the SDS name: sds_name.n.m (e.g., Surface_Refl.1.2 = the layer defined by the 1st element of the 3rd dimension and 2nd element of the 4th dimension of the 4D SDS Surface_Refl).

filename

Input filename

EXAMPLES

```
mask_sds -sds=Snow_Cover -of=mod10_mask.hdf
-mask="MOD10A1.A2002122.h30v11.004.2002101010341.hdf,
Snow_Spatial_QA,1-3==000"
MOD10A1.A2002122.h30v11.004.2002101010341.hdf
```

The above example outputs to mod10_mask.hdf the Snow_Cover values stored in MOD10A1.A2003122.h30v11.004.2003134090541.hdf where Snow_Spatial_QA bits 1-3 equal 000. Pixels where Snow_Spatial_QA bits 1-3 do not equal 000 are assigned an automatically generated mask fill value.

```
mask_sds -sds=Snow_Cover -of=mod10_mask.hdf
-mask="MOD10A1.A2002122.h30v11.004.2002101010341.hdf,
Snow_Spatial_QA,1-3==000" -fill=251
MOD10A1.A2002122.h30v11.004.2002101010341.hdf
```

The above example is the same as the first but pixels where Snow_Spatial_QA bits 1-3 do not equal 000 are assigned a mask fill value of 251.

```
mask_sds -sds=sur_refl_b01, sur_refl_b04, sur_refl_b03
-of=mod09_RGB_mask.hdf
-mask="MOD09A1.A2003113.h24v04.003.2003134160954.hdf,
sur_refl_b01,>1500"
MOD09A1.A2003113.h24v04.003.2003134160954.hdf
```

The above example outputs to mod09_RGB_mask.hdf the SDSs sur_refl_b01, sur_refl_b04, sur_refl_b03 in MOD09A1.A2003113.h24v04.003.2003134160954.hdf where sur_refl_b01 is greater than 1500. Pixels where sur_refl_b01 <= 1500 are assigned an automatically generated mask fill value.

```
mask_sds -sds=Snow_Cover -of=howzat.hdf
-mask="MOD09A1.A2003113.h24v04.003.2003134160954.hdf,
sur_refl_b01,>1500"
MOD10A1.A2003113.h24v04.004.2003101010541.hdf
```

The above example outputs to howzat.hdf the Snow_Cover values stored in MOD10A1.A2003113.h24v04.004.2003101010541.hdf at those pixels in MOD09A1.A2003113.h24v04.003.2003134160954.hdf where sur_refl_b01 is greater than 1500. Pixels where sur_refl_b01 <= 1500 are assigned an automatically generated mask fill value.

```
mask_sds -sds=sur_refl_b03 -of=mod09_qa_mask.hdf
```

```
-mask="MOD09A1.A2003113.h24v04.003.2003134160954.hdf,  
sur_refl_qc_500m,10-13==1100,AND,*,sur_refl_state_500m,3-5==001"  
MOD09A1.A2003113.h24v04.003.2003134160954.hdf
```

The above example outputs to mod09_qa_mask.hdf the SDS sur_refl_b03 at those pixels in MOD09A1.A2003113.h24v04.003.2003134160954.hdf where the sur_refl_qc_500m bits 10-13 equal 1100 AND sur_refl_state_500m bits 3-5 equal 001. Pixels where sur_refl_qc_500m bits 10-13 do not equal 1100 or sur_refl_state_500m bits 3-5 do not equal 001 are assigned an automatically generated mask fill value.

AUTHOR

Code: S. Devadiga and Yi Zhang
Documentation: S. Devadiga and D. Roy

Version 1.0, 06/15/2003

math_sds

NAME

math_sds - Perform a simple arithmetic on two input SDSs of the same or different MODIS Land HDF-EOS data products.

SYNOPSIS

```
math_sds -help [filename]
```

```
math_sds -of=filename
```

```
-math=<arithmetic expression>,dt,f_nop1,f_nop2,f_nop3,f_ovf
```

```
where arithmetic_expression = <SDS_name1,f1>,<op>,<SDS_name2,f2>
```

DESCRIPTION

Perform simple arithmetic on two input SDSs of the same or different MODIS Land HDF-EOS data products. Output is an HDF file containing the result of the pixel wise arithmetic operation.

The math option may be repeated with different arithmetic expressions to perform additional arithmetic on the same or different SDSs. Resulting SDSs of all arithmetic expressions are output to the same output HDF file.

If the input SDSs in an arithmetic expression are different resolutions then the resolution of one SDS must be an integral multiple of the other and the output SDS will be the higher of the two input resolutions.

This tool supports 2D/3D/4D SDSs.

The tool command arguments can be specified in any order.

ARGUMENTS

-help Print this help message, If the input filename is specified with this option, then the names of all the SDS in the file are displayed.

```
-math=<arithmetic expression>,dt,f_nop1,f_nop2,f_nop3,f_ovf
```

The output data type 'dt' can be one of INT8, UINT8, INT16, UINT16, INT32, UINT32, or FLOAT32.

f_nop1 and f_nop2 are no operation fill values for the two input SDS. The arithmetic operation is not performed at a pixel if the SDS value at the pixel in the input SDS1 is f_nop1 or SDS2 is f_nop2. If these arguments are unspecified then the input SDS fill values are used.

If the math operation cannot be performed at a pixel then the fill value `f_nop3` is written to the output SDS pixel. If this argument is unspecified then the SDS 1 fill value is used.

The overflow fill value `f_ovf` is written to the pixel in the output SDS if the arithmetic operation at the pixel results in an overflow. If this argument is unspecified then the SDS1 fill value is used.

To set any fill values to default the `*` symbol may be specified in place of the actual value.

where arithmetic expression=`<SDS_name1,f1>,<op>,<SDS_name2,f2>`

The arithmetic expression contains two operands separated by an operator. Each operand contains an SDS and file containing the SDS separated by a comma. The mathematical operator can be one of (+, -, *, /). Expressions are evaluated from left to right.

To process a specific layer of a 3D SDS specify the element number of the third dimension as a dot extension of the SDS name: `sds_name.n` (e.g., `sur_refl_b02.1` = the layer defined by the 1st element of the 3rd dimension of the 3D SDS `sur_refl_b02`).

To process a specific layer of a 4D SDS, specify the higher dimension element number(s) as a dot extension of the SDS name: `sds_name.n.m` (e.g., `Surface_Refl.1.2` = the layer defined by the 1st element of the 3rd dimension and the 2nd element of the 4th dimension of the 4D SDS `Surface_Refl`).

Note that wildcards and ranges of element values may be specified as `sds_name.*` and as `sds_name.n1-n2.m` respectively.

`-of=<filename>` Output filename

EXAMPLES

`math_sds -of=diff_temp.hdf`

-
math=LST_Day_1km,MOD11A1.A2001044.h09v05.002.2001154214604.
hdf,-
,LST_Night_1km,MOD11A1.A2001044.h09v05.002.2001154214604.hdf,I
NT16,0,0,-1,-2

math_sds -
math=sur_refl_b02,MOD09A1.A1999049.h12v03.001.1999265195217.hdf
./,sur_refl_b03,MOD09A1.A1999053.h12v03.001.1999250150605.hdf,INT
16,*,**,*,-of=MOD09A1.h12v03.b03.49-53.hdf

AUTHOR

Code: S. Devadiga and Yi Zhang
Documentation: S. Devadiga and D. Roy

Version 1.0, 04/05/2004

mosaic_sds

NAME

mosaic_sds - prepare a mosaic of SDSs from L3/L4 MODIS Land HDF-EOS data products.

SYNOPSIS

mosaic_sds -help [filename]

mosaic_sds -of=<output filename> [-sds=<SDSname1>[<,SDSname2>...]]
[-fill=<fill value>][-gr=<nh,nv>] [-gr=<hb, vb>]
f1 [-row=<min,range>] [-col=<min,range>] f2 [-row=<min,range>] [-
col=<min,range>]
f3 [..[f4...]]

DESCRIPTION

Create a spatial mosaic from different L3/L4 MODIS Land HDF-EOS data products. Specified SDSs are spatially arranged based either on their geolocation (defined by their tile id) or in a user specified manner. The horizontal and vertical tile numbers are identified from the filename or from the VERTICALTILENUMBER and HORIZONTALTILENUMBER metadata. Any missing tiles (i.e. gaps in the output mosaic) are replaced with a user specified fill value. Multiple SDSs are written to the mosaic if two or more SDSs are listed in the input argument

If the tile id is not recognized in one or more of the input files, then the tiles may be mosaiced with a user specified number of output horizontal and vertical tiles using the -gr option. If the tile id is not recognized and the -gr option is not used, then the tiles are mosaiced in an optimal "least-space" row column order. In either case the first file in the filenames list is placed in the north-west corner and the last file in the south-east corner.

This tool supports 2D/3D/4D SDSs.

The tool command arguments can be specified in any order.

ARGUMENTS

-help [filename]	Display this help message. If the input filename is specified with this option, then the names of all the SDS in the file are displayed.
-of=<filename>	Output filename.
-sds=<SDS list>	List of SDSs to mosaic. The SDS names are separated by commas with no space. By default

all SDSs are processed maintaining the input SDS interleaving.

To process a specific layer of a 3D SDS, specify the element number of the third dimension as a dot extension of the SDS name: sds_name.n (e.g., sur_refl_b02.1 = the layer defined by the 1st element of the 3rd dimension of the 3D SDS sur_refl_b02).

To process a specific layer of a 4D SDS, specify the higher dimension element number(s) as a dot extension of the SDS name: sds_name.n.m (e.g., Surface_Refl.1.2 = the layer defined by the 1st element of the 3rd dimension and the 2nd element of the 4th dimension of the 4D SDS Surface_Refl).

Note that wildcards and ranges of element values may be specified as sds_name.* and as sds_name.n1-n2.m respectively.

- row=<min,max> Range of SDS rows to be included in the mosaic. (default: all rows). Input for each file separately following the input filename.
- col=<min,max> Range of SDS columns to be included in the mosaic. (default: all columns). Input for each file separately following the input filename.
- fill=<fill value> Output fill value for missing “gap” tiles. The fill value must be within the valid range of the input SDS data type, e.g., if the input SDS is of type INT8 then valid range for the fill value is {-128 :127}. If no fill value is specified a fill value equal to the input SDS fill value is used.
- gr=<nh,nv > Number of horizontal and vertical tiles in the output mosaic where each tile is occupied by one of the input tile. If -gr option is not specified and tile ids are not recognized, then the tiles are arranged in a “least-space” manner.
- f1 f2 f3 Input filenames (followed by -row and -col options if subset is required).

EXAMPLES

```
mosaic_sds -sds=fire_mask_1 -of=fire_mask.2002186.003.mosaic.hdf  
MYD14GD.A2002186.h20v09.003.2002190153045.hdf
```

MYD14GD.A2002186.h20v10.003.2002190153230.hdf
MYD14GD.A2002186.h21v09.003.2002190153400.hdf
MYD14GD.A2002186.h21v10.003.2002190153645.hdf

{Note: This example creates a mosaic of SDS fire_mask_1 from the four specified MYD14GD files and outputs the mosaic to a file called fire_mask.2002186.003.mosaic.hdf }

```
mosaic_sds -sds=fire_mask_1,num_observations -of=myd14gd_mosaic.hdf  
MYD14GD.A2002189.*.003.*.hdf
```

{Note: This example creates a mosaic containing two SDSs (fire_mask_1 and num_observations) from all the MYD14GD files of day 2002189 under the current working directory which satisfy the wild card.}

```
mosaic_sds -  
sds=sur_refl_b01_1,sur_refl_b02_1,sur_refl_b03_1,sur_refl_b04_1  
-of=myd09ghk_mosaic.hdf -fill=-1000 MYD09GHK.A2002186.h2*hdf
```

```
mosaic_sds -sds=sur_refl_b01,sur_refl_b02  
-of= tst12_mosaic_MOD09A1.A2001201.h20-21.v05-06.hdf  
MOD09A1.A2001201.h20v05.003.2002281080105.hdf -  
row=1500,2399 -col=1000,2399  
MOD09A1.A2001201.h20v06.003.2002281080112.hdf  
-row=0,1200 -col=1000,2399  
MOD09A1.A2001201.h21v05.003.2002281080255.hdf  
-row=1500,2399 -col=0,1200  
MOD09A1.A2001201.h21v06.003.2002281080110.hdf  
-row=0,1200 -col=0,1200
```

```
mosaic_sds -sds=fire_mask_1 -row=0,599 -col =0,599 -  
of=fire_mask_mosaic.hdf MYD14GD.A2002186.*.hdf
```

```
mosaic_sds -sds=EV_Band26 -fill=-100 -gr=3,3 -of =myd021km_mosaic.hdf  
MYD021KM.A2002189.07*.hdf
```

```
mosaic_sds -sds ="BRDF_Albedo_Parameters.1.2" -of=brdf_mosaic.hdf  
MYD43B1.A2002177.h*.hdf
```

AUTHOR

Code: S. Devadiga and Yi Zhang
Documentation: S. Devadiga and D. Roy

Version 1.0 08/08/2002

read_l2g

NAME

read_l2g - read specified layers or granules from a MODIS Land L2G HDF-EOS data product and write out to 2D SDSs.

SYNOPSIS

```
read_l2g -help [filename]
```

```
read_l2g -of=<output filename> -layer[=layer1[,layer2...]] -  
glist[=orbit,gran1[,gran2...]] -  
-gpidx[=index1[,index2...]] -gid=[granule1[,granule2..]] [-meta]  
[-sds=<SDSname1>[<,SDSname2>...]] [-ptr=<pointer file>]  
filename
```

DESCRIPTION

The different MODIS Land L2G HDF data products store one or more L2 observations for each L2G pixel in a series of layers (that reflect the MODIS orbit overpass and swath sensing geometry) in a compressed run length encoded format. This tool extracts SDSs from a MODIS Land L2G file, selecting observations according to the layer number, granule number, granule pointer index, or granule id and writes them to an output HDF file as separate 2D SDSs.

The output file SDS names reflect the input SDS name and the layer number, granule number, granule pointer index, or granule id. For example:

sur_refl_b01_layer2 (layer 2 of input SDS 'sur_refl_b01'),

fire_mask_gran166_orb8318 (granule 166 from orbit 8318 of input SDS 'fire_mask'),

SolarZenith_gpnt2 (granule with granule pointer index 2 of input SDS 'SolarZenith'),

Snow_Cover_A2001192.1340 (granule with granule id A2001192.1340 of input SDS 'Snow_Cover').

Note that the associated L2G pointer file must be defined in the argument list if observations are selected by granule number, granule pointer index or granule id.

This tool supports L2G only.

The tool command arguments can be specified in any order.

ARGUMENTS

-help [filename]	Display this help message. If the input filename is specified with this option, then the valid values for options <code>-layer</code> , <code>-glist</code> , <code>-gpidx</code> , <code>-gid</code> , <code>-ptr</code> , and <code>-sds</code> are displayed.
-of=<filename>	Output filename.
-layer=layer1[,layer2...]	Select by layer number (1-based). If this option is specified without any value, then all layers are output.
-glist=orbit,gran1[,gran2...]	Select by granule number using orbit and granule number as the unique granule identification. To select granules from an orbit enter the orbit number followed by the granule numbers. Repeat the option to select granules from additional orbits. If this option is specified without any value, then all granules are output.
-gpidx=index1[,index2...]	Select by granule pointer index (0 based). If this option is specified without any value, then all granules are output.
-gid=granule1[,granule2...]	Select by granule ID (e.g., Ayyyyddd.hhmm). If this option is specified without any value, then all granules are output.
-sds=<SDS list>	List of SDS to read. SDS names are separated by commas with no space. By default all SDS in the file are read. SDS names are input without the <code>_1</code> and <code>_c</code> extensions used to indicate the first layer and the compact layer SDS, e.g., although the MOD09GHK files contain SDS <code>sur_refl_b01_1</code> and <code>sur_refl_b01_c</code> , the SDS name should be specified as <code>sur_refl_b01</code> .
-ptr=pointer filename	Pointer filename. Required if reading an SDS by the identity of the L2 granule, using <code>-glist</code> , <code>-gpidx</code> or <code>-gid</code> option.
-meta	Copy metadata from input file to output file.
filename	Input filename.

EXAMPLES

```
read_l2g -layer=1-2 -sds=Snow_Cover -of=myd10l2g_layers1_2.hdf
MYD10L2G.A2002177.h09v05.003.2002180140700.hdf
```

{Note: This example extracts the first and second layers of SDS Snow_Cover from the input MYD10L2G file.}

```
read_l2g -sds=fire_mask -glist=13928,230,231 -
of=fire_mask_orbit13928.hdf -meta
-ptr=MODPT1KD.A2002212.h09v05.003.2002218193538.hdf
MOD14GD.A2002212.h09v05.003.2002218194510.hdf
```

{Note: This example extracts granules 230 and 231 of orbit 13928 of SDS fire_mask from the input MOD14GD file. Two SDSs are written to the output file. The associated MODPT1KD.2002212.h09v05.003.2002218193538.hdf pointer file has to be specified. The metadata is copied from the input file to the output file.}

```
read_l2g -sds=sur_refl_b01,sur_refl_b03,sur_refl_b04 -layer=1-3
-glist=8319,186 -of=mod09gst_grans8319_layers1_3.hdf
-ptr=MODPTHKM.A2001192.h12v09.003.2001309125954.hdf
MOD09GHK.A2001192.h12v09.003.2001309142732.hdf
```

{Note: This example extracts granule number 186 of orbit 8319 and layers 1 to 3 of the SDSs sur_refl_b01, sur_refl_b03, sur_refl_b04 from the input MOD09GHK file. Twelve SDSs are written to the output file. The associated pointer file MODPTHKM.A2001192.h12v09.003.2001309125954.hdf has to be specified.}

```
read_l2g -sds=SolarZenith,SolarAzimuth,SensorZenith,SensorAzimuth
-gpidx=0-1 -of=modmggad_granid0_1.hdf
-ptr=MODPT1KD.A2002212.h09v05.003.2002218193538.hdf
MODMGGAD.A2002212.h09v05.003.2002218193604.hdf
```

{Note: This example extracts the first two granules (defined in the associated MODPT1KD file with granule pointer index 0 to 1) from the input MODMGGAD file of SDSs SolarZenith, SolarAzimuth, SensorZenith and SensorAzimuth. Eight SDSs are written to the output file. The associated pointer file MODPT1KD.2002212.h09v05.003.2002218193538.hdf has to be specified.}

```
read_l2g -sds=state_1km -gid=A2002212.1900,A2002212.1905 -meta
-of=mod09gst_gran_a2002212.003.hdf
-ptr=MODPT1KD.A2002212.h09v05.003.2002218193538.hdf
```

MOD09GST.A2002212.h09v05.003.2002218193744.hdf

{Note: This example extracts granules with granule id A2002212.1900 and A2002212.1905 of SDS state_1km from the input MOD09GST file. Two SDSs are written to the output file. The pointer file MODPT1KD.2002212.h09v05.003.2002218193538.hdf associated with MOD09GST.A2002212.h09v05.003.2002218193744.hdf has to be specified. The metadata is copied from the input file to the output file.}

AUTHOR

Code: S. Devadiga and Yi Zhang
Documentation: S. Devadiga and D. Roy

Version 1.0, 08/08/2002

read_meta

NAME

read_meta - read metadata from any MODIS Land HDF-EOS data product.

SYNOPSIS

read_meta -help

read_meta [-core|arch|qa] [-meta=<metadata1>[<,metadata2>...]] [-case] filename(s)

DESCRIPTION

Read a set of specified metadata from MODIS Land HDF-EOS data product(s) and write the output to stdout. If no options are specified, all metadata are read. Option -case allows the metadata names to be specified case insensitive.

The tool command arguments can be specified in any order.

ARGUMENTS

-help	Display this help message.
-core	Print core metadata.
-arch	Print archive metadata.
-qa	Print only QA metadata.
-meta=<metadata list>	Print the specified metadata. This option is ignored if one of the options above is used. Metadata names are separated by commas with no space.
-case	Make specified metadata name(s) case insensitive. This option can be used only with the -meta option.
filename(s)	One or more input files separated by space. Files are processed in the order they are listed in the command line.

EXAMPLES

```
read_meta -meta=AUTOMATICQUALITYFLAG MOD09GHK.*.hdf
read_meta MYD021KM.A2002189.0350.003.2002191000652.hdf
read_meta -arch MOD021KM.A2002189.0350.003.2002191000652.hdf
read_meta -core MYD09GHK.A2002177.h09v05.003.2002180140102.hdf
read_meta -qa MYDPTHKM.A2002177.h09v05.003.2002180134421.hdf
read_meta -meta=EQUATORCROSSINGDATE -case
    MYD09GHK.A2002177.h09v05.003.2002180140102.hdf
read_meta -meta= DAYNIGHTFLAG, GRINGPOINTLONGITUDE,
    GRINGPOINTLATITUDE MYD03.A2002*.hdf
```

AUTHOR

Code: S. Devadiga and Yi Zhang
Documentation: S. Devadiga and D. Roy

Version 1.0, 08/08/2002

read_pixvals

NAME

read_pixvals – read MODIS Land HDF-EOS data product values at specified pixel locations

SYNOPSIS

```
read_pixvals [-help] [filename]
read_pixvals -xy=col[.cs[.cs]],row[.rs[.rs]]<coordinates filename>
[-res=qkm|hkm|1km] filename(s)
```

DESCRIPTION

Read the pixel values at specified locations in one or more input MODIS Land HDF-EOS data products and output to stdout. The pixel values for each SDS in each file are output as separate lines.

If more than one input MODIS Land HDF-EOS data product is specified then they must be all L2 products *or* all L2G/L3/L4 MODIS Land HDF-EOS data products.

The MODIS Land HDF–EOS data product may contain SDSs with different spatial dimensions corresponding to the 250m, 500m and 1km MODLAND pixel resolutions. In this case, the `–res` option is used to specify which of the 1km, 500m or 250m pixel resolutions is referenced. If `–res` is not specified then the `–xy` location is assumed to reference the coarsest spatial resolution of the different SDSs.

Sub pixel locations may be output by specifying a sub pixel offset (0 or 1 in the x and/or y axes). If not specified a 0 pixel offset is assumed. See examples below.

This tool supports 2D/3D/4D and L2G compact and full format SDSs. Pixel values for each layer of the 3D/4D and L2G SDSs are output.

ARGUMENTS

- | | |
|-------------------------------|---|
| <code>–help [filename]</code> | Display this help message. If the input filename is specified with this option, then the names of all the SDSs in the file are displayed. |
| <code>–res=qkm hkm 1km</code> | Reference SDS resolution (qkm=250m, hkm=500m, 1km=1000m) of the pixel location specified in the <code>–xy</code> argument. If unspecified the coarsest resolution of all the SDSs in the input file list is used. |

`-xy=col[.cs[.cs]],row[.rs[.rs]]|<coordinates filename>`

Column and row pixel locations (0-based) or name of an ASCII coordinates file containing the column and row pixel locations. Multiple locations may be specified by repeating the `-xy` option or by specifying the `x` and `y` coordinates on different lines in the ASCII coordinates file.

Sub pixel offsets for higher spatial resolution SDS may be specified as `col.cs row.rs`. The offsets refer to the top left corner pixel. For example:

`-res=hkm -xy=100.0, 200.0`
(read values at pixel 100,200 from the 500m SDS, and at pixel 200,400 from the 250m SDS)

`-res=hkm -xy=100.1, 200.1`
(read values at pixel 100,200 from the 500m SDS, and at pixel 201,401 from the 250m SDS)

`-res=1km -xy=100.0.1,200.0.1`
(read values at pixel 100,200 from the 1km SDS, at pixel 200,400 from the 500m SDS, and at pixel 401,801 from the 250m SDS)

`filename(s)` One or more input files separated by space.

EXAMPLES

```
read_pixvals -xy=1000,200 -xy=0,0
MOD09A1.A2003057.h29v11.004.2003069051044.hdf
```

```
read_pixvals -xy=10,10 -res=1km
MOD09GHK.A2002225.h19v09.003.2002227235523.hdf
MOD09GQK.A2002225.h19v09.003.2002227235424.hdf
```

AUTHOR

Code: S. Devadiga and Yi Zhang
Documentation: S. Devadiga and D. Roy

Version 1.0, 06/15/2003

read_proj_param

NAME

read_proj_param - read the projection parameter information of a L2G/L3/L4 MODIS Land HDF-EOS data product.

SYNOPSIS

```
read_proj_param [-help]
read_proj_param filename
```

DESCRIPTION

Read the projection parameter information stored in the metadata of a L2G/L3/L4 MODIS Land HDF-EOS data product (i.e., one that is geolocated) and output to stdout. This information is needed to project non-MODLAND data into registration with a geolocated MODIS Land HDF-EOS data product.

An example of the output is:

```
Tile ID: h07v10
UpperLeftPointMtrs: (-2383921.627500, -476784.325500)
LowerRightMtrs: (-1430352.976500, -1430352.976500)
Projection Code: (11, GCTP_LAMAZ)
Projection parameters: (6371228.000000, 0.000000, 0.000000,
0.000000, 0.000000, 90000000.000000, 0.000000, 0.000000,
0.000000, 0.000000, 0.000000, 0.000000, 0.000000)
```

- The tile id is of the form hxxvyy, where xx and yy are the two digit horizontal and vertical MODLAND product tile numbers.
- The UpperLeftPointMtrs and LowerRightMtrs values are the coordinates of the top left and bottom right pixel centers.
- The Projection Code and Projection parameter values are defined by the General Cartographic Transformation Package (GCTP). For further GCTP information please see the MODIS Reprojection Tool documentation or <http://mapping.usgs.gov/www/html/cartsoft.html>.

ARGUMENTS

-help	Display this help message
filename	Input filename

EXAMPLES

```
read_proj_param MOD09Q1.A2001193.h09v05.004.2002200065231.hdf
read_proj_param MYD43B4.A2003081.h26v06.003.2003103182903.hdf >
parameters.txt
```

AUTHOR

Code: S. Devadiga and Yi Zhang
Documentation: S. Devadiga and D. Roy

Version 1.0, 06/15/2003

read_sds_attributes

NAME

read_sds_attributes - read attributes of one or more SDS of a MODIS Land HDF-EOS data product.

SYNOPSIS

```
read_sds_attributes -help [filename]
```

```
read_sds_attributes [-sds=<SDS_name1>[,<SDS_name>. . ]] filename
```

DESCRIPTION

Read attributes of one or more SDSs of an input MODIS Land HDF-EOS data product and output the result to stdout. The SDS attributes include: fill values, units, scaling and offset values, SDS long name, etc.

ARGUMENTS

-help	Display this help message, If the input filename is specified with this option, then the names of all the SDSs in the file are displayed.
-sds=<SDS_list>	List of SDS to process. SDS names are separated by commas with no space. By default attributes for all SDSs in the input file are output.
filename	Input filename.

EXAMPLE

```
read_sds_attributes -sds='500m Surface Reflectance Band 1,500m  
Surface Reflectance Band 3,500m Surface Reflectance Band 4'  
MOD09.A2002123.0040.003.2002125174437.hdf
```

```
read_sds_attributes MOD35_L2.A2002123.0040.003.2002124023706.hdf
```

AUTHOR

Code: S. Devadiga and Yi Zhang
Documentation: S. Devadiga and D. Roy

Version 1.0, 04/05/2004

reduce_sds

NAME

reduce_sds - reduce the spatial dimensions of one or more SDSs of a MODIS Land HDF-EOS data product.

SYNOPSIS

reduce_sds -help [filename]

reduce_sds -of=<output_filename> -rf=<reduction factor> -sub|avg|cnt|cl
[-sds=<SDSname1>[<,SDSname2>...]]
[-bit=[<bit range>]<opr><value>[, [<bit range>]<opr><value>]...]
[-min] [-max] [-std] [-num] [-meta] [-float] filename

DESCRIPTION

The spatial dimensions of input SDS(s) may be reduced using one of four different methods. The reduction factor (rf) must be a non-zero positive integer. The output SDS x dimension will be $((x \text{ div } rf) + (x \text{ mod } rf))$ and similarly the y dimension will be $((y \text{ div } rf) + (y \text{ mod } rf))$. If the input SDS list contains SDSs with different spatial dimensions the reduction factor will be applied to the SDS with the smallest spatial dimension and the other SDS(s) will be reduced to have the same output dimension.

All SDS fill values are ignored.

This tool may be used to reduce data volumes, and to enable analysis of the different MODLAND product spatial resolutions (250m, 500m, 1km), or to enable quick comparison with other coarser spatial resolution data sets.

This tool compliments the tool *enlarge_sds*.

This tool supports 2D/3D/4D SDS(s).

The tool command arguments can be specified in any order.

ARGUMENTS

-help [filename]	Display this help message. If the input filename is specified with this option, then the names of all the SDSs in the file are displayed.
-of=<filename>	Output filename
-rf=<reduction factor>	Reduction factor (a non-zero positive integer)

- sub Reduce by sub-sampling. Pixel value at (i, j) in the output SDS is copied from the pixel at (rf*i + rf/2, rf*j + rf/2) in the input SDS.

- avg Reduce by averaging. Pixel value at (i, j) in the output SDS is the average of pixel values in a sub window defined from (rf*i, rf*j) to (rf*i + rf - 1, rf*j + rf - 1) in the input SDS. Optional minimum, maximum, standard deviation, and number of averaged pixels may be output as separate SDS.

- cnt Reduce by pixel counting. Pixel value at (i, j) in the output SDS is the number of pixels in a sub window defined from (rf*i, rf*j) to (rf*i + rf - 1, rf*j + rf - 1) in the input SDS with bit value equal to the user specified value. Relational operators may be used. The output SDS value is in the range {0 : rf x rf}.

- cl Reduce by majority class value. Pixel value at (i, j) in the output SDS is set to the majority value of the pixels in a sub window defined from (rf*i, rf*j) to (rf*i + rf - 1, rf*j + rf - 1). This option is best used for data with a small number of nominal pixel values.

- sds=<SDS list> List of SDSs to reduce. SDS names are separated by commas with no space. By default all SDSs are processed maintaining the input SDS interleaving.

- To process a specific layer of a 3D SDS specify the element number of the third dimension as a dot extension of the SDS name: sds_name.n (e.g., sur_refl_b02.1 = the layer defined by the 1st element of the 3rd dimension of the 3D SDS sur_refl_b02).

- To process a specific layer of a 4D SDS, specify the higher dimension element number(s) as a dot extension of the SDS name: sds_name.n.m (e.g., Surface_Refl.1.2 = the layer defined by the 1st element of the 3rd dimension and the 2nd element of the 4th dimension of the 4D SDS Surface_Refl).

Note that wildcards and ranges of element values may be specified as `sds_name.*` and as `sds_name.n1-n2.m` respectively.

- `-bit=[<bit range>]<operator>< value>,[<bit range>]<operator><value>, . . .`
This option is applicable only with the `-cnt` option. The SDS bit range and corresponding value are specified in decimal separated by a relational operator. If the bit range is not specified the tool considers all the bits in the SDS. Multiple range, operator and value combinations separated by commas will result in separate output SDS for each of such combination. Valid relational operators are: `==, <, >, <=, >=, !=`
- `-std` Compute the standard deviation in each sub window (for `-avg` option only).
- `-min` Compute the minimum value in each sub window (for `-avg` option only).
- `-max` Compute the maximum value in each sub window (for `-avg` option only).
- `-num` Compute the number of averaged pixels in each sub window (for `-avg` option only).
- `-meta` Copy metadata from input file to output file.
- `-float` Output average value SDS in float data type, default is the input data type (for `-avg` option only).
- `filename` Input filename.

EXAMPLES

```
reduce_sds -sds=Fpar_1km -sub -rf=10 -of=sub_fpar.hdf
MOD15A1.A2001193.h09v05.004.2002198025239.hdf
```

{Note: This example reduces the spatial dimensions of SDS `Fpar_1km` by 10 using the sub-sampling method. The output file will have spatial dimensions approximately 10 times smaller.}

```
reduce_sds -sds=Lai_1km -avg -min -max -std -num -rf=10 -of=avg_lai.hdf  
MOD15A1.A2001193.h09v05.004.2002198025239.hdf
```

```
reduce_sds -sds=Cloud_Mask -sub -rf=5 -of=cloud_sub.hdf  
MYD35_L2.A2002189.2040.003.2002191125354.hdf
```

```
reduce_sds -sds=Cloud_Mask.1-2 -avg -rf=5 -of=cloud_avg.hdf  
MYD35_L2.A2002189.2040.003.2002191125354.hdf
```

```
reduce_sds -cnt -bit="0-3<=2,0-3==3,0-3==4,0-3==5,0-3==6,0-3==7" -  
meta -sds="most confident detected fire" -rf=5 -of=fire_class.hdf  
MOD14A1.A2002185.h30v11.003.2002204204451.hdf
```

```
reduce_sds -rf=2 -cl -of=land_cover_class.hdf  
MOD12Q1.A2000289.h01v11.003.2002171021653.hdf
```

AUTHOR

Code: S. Devadiga and Yi Zhang
Documentation: S. Devadiga and D. Roy

Version 1.0, 08/08/2002

reduce_sds_rank

NAME

reduce_sds_rank – convert one or more 3D/4D SDS of a MODIS Land HDF-EOS data product to many 2D SDSs.

SYNOPSIS

```
reduce_sds_rank -help [filename]
```

```
reduce_sds_rank -of=<output filename>  
{[-sds=<SDSname> [-dim=<dimstr> [-dim=<dimstr>. . ]]]} [-all] [-meta]  
filename
```

DESCRIPTION

Several MODIS Land HDF-EOS data products (e.g., MOD43, MYD43) and related MODIS products (e.g., MOD35) contain multidimensional SDSs. This tool converts one or more multidimensional (3D or 4D) SDSs to a series of 2D HDF SDSs. Specific SDS layers may be selected using the `-sds` and `-dim` options for each input 3D or 4D SDS.

The output file SDS names reflect the input SDS name, the dimension name and the dimension element numbers. For example:

BRDF_Albedo_Parameters.Num_Land_Bands_Plus3_3.Num_Parameters_1 (parameter 1 for land band 3 of SDS BRDF_Albedo_Parameters in MOD43B1. Note that BRDF_Albedo_Parameters is the input SDS name, Num_Land_Bands_Plus3 and Num_Parameters are the 3rd and the 4th dimension names)

Surface_Refl.Num_Obs_Max_1.Num_Land_Bands_2 (1st observation of land band 2 of SDS Surface_Refl in MODAGAGG. Note that Surface_Refl is the input SDS name, Num_Obs_Max and Num_Land_Bands are the 3rd and the 4th dimension names)

Angles.Num_Obs_Max_2.Num_Angles_3 (2nd observation of the 3rd angle component of SDS Angles in MODAGAGG. Note that Angles is the input SDS name, Num_Obs_Max and Num_Angles are the 3rd and the 4th dimension names)

This tool supports 3D and 4D SDS(s).

The tool command arguments can be specified in any order.

ARGUMENTS

-help [filename]	Display this help message. If the input filename is provided with this option, then the names of all the SDS in the file with the SDS dimension name and size are displayed.
-of=<filename>	Output filename.
-sds=<SDS name>	Name of SDS to convert. By default all 3D/4D SDS in the input file are converted to 2D SDSs.
-dim=<dimension name, dimension element number(s)>	Name of the 3rd or higher SDS dimension and the dimension element number(s) (1-based). The dimension element numbers can be separated by comma or by '-'. The -dim option must be repeated for each of the defined SDS dimension names, e.g., -dim=Band_id,1-7 -dim=obs_id,0,1. This option should follow the -sds option for each input SDS. By default, the tool reduces all of the 3 rd and 4 th dimensions of a specified SDS.
-all	Create an additional output SDS containing all the 2D SDSs mosaiced in a single SDS in row-column order.
-meta	Copy metadata from input file to output file.
filename	Input filename.

EXAMPLES

```
reduce_sds_rank -sds=BRDF_Albedo_Parameters -  
  dim=Num_Land_Bands_Plus3,1-7  
  -dim=Num_Parameters,1-3 -of=brdf_albedo_2dsds.hdf  
  MYD43B1.A2002177.h11v11.003.2002210233848.hdf
```

{Note: This example extracts twenty-one 2D SDSs from the 4D SDS BRDF_Albedo_Parameters stored in the input MYD43B1file. In this example, the values of the BRDF_Albedo_Parameters 1, 2 and 3 are written as separate 2D SDSs for each of the first 7 land bands. }

```
reduce_sds_rank -sds=Angles -dim=Num_Obs_Max,1-4 -  
dim=Num_Angles,1-4 -all  
-of=agg_angles.hdf  
MODAGAGG.A2002017.h20v11.004.2002203181248.hdf
```

{Note: This example extracts sixteen 2D SDSs from the 4D SDS Angles stored in the input MODAGAGG file. In this example, the values of the four angles (View Zenith, Solar Zenith, Solar Azimuth, View Azimuth) are written as separate 2D SDSs for each of the four observations). An additional output SDS containing all the 2D SDSs in a single SDS in row-column order is also output. }

```
reduce_sds_rank -sds=Surface_Refl -dim=Num_Obs_Max,1,2  
-dim=Num_Land_Bands,1-7 -sds=Band_QC -  
dim=Num_Obs_Max,1,4  
-dim=Num_Band_QC,1-7 -of=agg_angles_bandqc.hdf  
MODAGAGG.A2002017.h20v11.004.2002203181248.hdf
```

AUTHOR

Code: S. Devadiga and Yi Zhang
Documentation: S. Devadiga and D. Roy

Version 1.0, 08/08/2002

sds2bin

NAME

sds2bin - Convert an SDS of a MODIS Land HDF-EOS data product to binary format.

SYNOPSIS

sds2bin -help [filename]

sds2bin -of=<output filename> -sds=<SDSname> filename

DESCRIPTION

Convert a user specified SDS from a MODIS Land HDF-EOS data product to an output binary format file.

The tool supports 2D/3D/4D SDSs.

The tool command arguments can be specified in any order.

ARGUMENTS

-help
[filename] Display this help message. If the input filename is specified with this option, then the names of all the SDS in the file are displayed.

-of=<filename> Output filename.

-sds=<SDS name> SDS to be converted

To process a specific layer of a 3D SDS, specify the element number of the third dimension as a dot extension of the SDS name: sds_name.n (e.g., sur_refl_b02.1 = the layer defined by the 1st element of the 3rd dimension of the 3D SDS sur_refl_b02).

To process a specific layer of a 4D SDS, specify the higher dimension element number(s) as a dot extension of the SDS name: sds_name.n.m (e.g., Surface_Refl.1.2 = the layer defined by the 1st element of the 3rd dimension and the 2nd element of the 4th dimension of the 4D SDS Surface_Refl).

Note that wildcards and ranges of element values may be specified as sds_name.* and as sds_name.n1-n2.m respectively.

filename Input filename.

EXAMPLES

```
sds2bin -sds=sur_refl_b01 -of=sur_refl_b01.img  
MOD09A1.A2001145.h20v10.003.2001214125825.hdf
```

```
sds2bin -sds=EV_1KM_Emissive -of=ev_1km_emissive.img  
MYD021KM.A2002189.2040.003.2002191123800.hdf
```

```
sds2bin -sds="BRDF_Albedo_Parameters.1.2" -of=brdf_albedo.img  
MYD43B1.A2002177.h11v11.003.2002210233848.hdf
```

{Note: This example converts the layer defined by the 1st element of the 3rd dimension and the 2nd element of the 4th dimension of the SDS BRDF_Albedo_parameters to the binary format file brdf_albedo.img. The output is a 2D binary image}

```
sds2bin -sds="BRDF_Albedo_Parameters.1-7.1" -of=brdf_albedo.img  
MYD43B1.A2002177.h11v11.003.2002210233848.hdf
```

{Note: This examples converts the 7 layers defined by the 1st seven elements of the 3rd dimension and the 1st element of the 4th dimension of the SDS BRDF_Albedo_parameters to the binary format file brdf_albedo.img. The 7 layers are output as a single 3D binary image where the number of elements in the 3rd dimension is 7}

AUTHOR

Code: S. Devadiga and Yi Zhang
Documentation: S. Devadiga and D. Roy

Version 1.0, 08/08/2002

subset_sds

NAME

subset_sds - Create spatial subset SDSs of one or more SDS of a MODIS Land HDF-EOS data product.

SYNOPSIS

subset_sds -help [filename]

subset_sds -of=<output filename> -row=<start,end> -col=<start,end>
[-sds=<SDS_name1>[,<SDS_name2>,. . .]] filename

DESCRIPTION

Create spatial subset SDS(s) of one or more SDS of an input MODIS Land HDF-EOS data product. The spatial subset is specified by the pixel row-column range in the input product.

This tool supports 2D/3D/4D SDS(s).

The tool command arguments can be specified in any order.

ARGUMENTS

-help

Display this help message, **If the input filename is specified with this option, then the names of all the SDS in the file are displayed.**

-sds=<SDS_list>

List of SDS to subset. SDS names are separated by commas with no space. By default all SDSs are subsetted maintaining the input SDS interleaving.

To process a specific layer of a 3D SDS specify the element number of the third dimension as a dot extension of the SDS name: sds_name.n (e.g., sur_refl_b02.1 = the layer defined by the 1st element of the 3rd dimension of the 3D SDS sur_refl_b02).

To process a specific layer of a 4D SDS, specify the higher dimension element number(s) as a dot extension of the SDS name: sds_name.n.m (e.g., Surface_Refl.1.2 = the layer defined by the 1st element of the 3rd dimension and the 2nd element of the 4th dimension of the 4D SDS Surface_Refl).

Note that wildcards and ranges of element values may be specified as sds_name.* and as sds_name.n1-n2.m respectively.

-row=<start,end> Subset row range (start and end are inclusive)
-col=<start,end> Subset column range (start and end are inclusive)
-of=<out filename> Output filename
filename Input filename

EXAMPLES

```
subset_sds -sds="most confident detected fire" -row=0,10  
-col=1130,1140 MYD14A1.A2003281.h09v05.003.2003296025915.hdf  
-of=Subset_MYD14A1.A2003281.h09v05.003.2003296025915.hdf
```

```
subset_sds -sds=Surface_Refl -row=10,100 -col=100,200  
-of=Subset_MODAGAGG.A2003077.h19v07.004.2003079210733.hdf  
MODAGAGG.A2003077.h19v07.004.2003079210733.hdf
```

```
subset_sds -sds=EV_Band26 -row=300,400 -col=1000,1100  
-of=Subset_MOD021KM.A2004007.0155.004.2004007095233.hdf  
MOD021KM.A2004007.0155.004.2004007095233.hdf
```

AUTHOR

Code: S. Devadiga and Yi Zhang
Documentation: S. Devadiga and D. Roy

Version 1.0, 04/05/2004

tile_id

NAME

tile_id - Compute the L2G/L3/L4 MODIS Land tile id (horizontal and vertical tile number) that contains the user specified geographic longitude and latitude.

SYNOPSIS

tile_id -help

tile_id -proj=SIN|ISIN -lon=<longitude> -lat=<latitude>

DESCRIPTION

Compute the horizontal and vertical tile number of the L3 MODIS Land tile id that contains the user specified longitude and latitude. For more information in the MODIS Land tile grid see http://landweb.nascom.nasa.gov/cgi-bin/QA_WWW/newPage.cgi?fileName=prod_format

The tool command arguments can be specified in any order.

ARGUMENTS

-help	Display this help message
-proj=SIN ISIN	Projection type (SIN for Sinusoidal, ISIN for Intergerized Sinusoidal)
-lon=<longitude>	Longitude of the point in degrees (-180.0, 180.0)
-lat=<latitude>	Latitude of the point in degrees (-90.0 , 90.0)

EXAMPLE

tile_id -proj=SIN -lon=45.0 -lat=45.0

AUTHOR

Code: S. Devadiga and Yi Zhang
Documentation: S. Devadiga and D. Roy

Version 1.0, 04/05/2004

transpose_sds

NAME

transpose_sds - Transpose one or more SDS of a MODIS Land HDF-EOS data product by rotating the SDS 180 degrees in a clockwise direction.

SYNOPSIS

```
transpose_sds help [filename]
```

```
transpose [-sds=<SDS_name>1[,<SDS_name2>. . ]] -of=<output_file>  
[-meta] filename
```

DESCRIPTION

Transpose one or more SDS of a MODIS Land HDF-EOS data product by rotating the SDS 180 degrees in a clockwise direction. This tool enables qualitative comparison of MODIS Aqua and Terra Level 2 or Level 1 granules.

This tool supports 2D/3D/4D SDSs.

The tool command arguments can be specified in any order.

ARGUMENTS

- help Display this help message, If the input filename is specified with this option, then the names of all the SDS in the file are displayed.
- sds=<SDS_list> List of SDS to enlarge. SDS names are separated by commas with no space. By default all SDSs are processed maintaining the input SDS interleaving.
- To process a specific layer of a 3D SDS specify the element number of the third dimension as a dot extension of the SDS name: sds_name.n (e.g., sur_refl_b02.1 = the layer defined by the 1st element of the 3rd dimension of the 3D SDS sur_refl_b02).
- To process a specific layer of a 4D SDS, specify the higher dimension element number(s) as a dot extension of the SDS name: sds_name.n.m (e.g., Surface_Refl.1.2 = the layer defined by the 1st element of the 3rd dimension and the 2nd element of the 4th dimension of the 4D SDS Surface_Refl).

Note that wildcards and ranges of element values may be specified as sds_name.* and as sds_name.n1-n2.m respectively.

-of=filename	Output filename
-meta	Copy metadata from the input file to the output
filename	Input filename

EXAMPLES

```
transpose_sds -sds='500m Surface Reflectance Band 1,500m Surface  
Reflectance Band 3,500m Surface Reflectance Band 4'  
MOD09.A2002123.0040.003.2002125174437.hdf -of=test.hdf
```

```
transpose_sds -sds=Cloud_Mask,Quality_Assurance -of=test.hdf  
MOD35_L2.A2002123.0040.003.2002124023706.hdf -meta
```

AUTHOR

Code: S. Devadiga and Yi Zhang
Documentation: S. Devadiga and D. Roy

Version 1.0, 04/05/2004

unpack_sds_bits

NAME

unpack_sds_bits - unpack specified bits from one or more SDS of a MODIS Land HDF-EOS data product.

SYNOPSIS

```
unpack_sds_bits -help [filename]
```

```
unpack_sds_bits -of=<output filename> [-  
sds=<SDSname1>[<,SDSname2>...]  
-bit=<Bitnumbers> -meta filename
```

DESCRIPTION

The MODIS Land per-pixel QA information and other information, such as the land-sea mask, logical criteria used by the algorithm, and cloud state, are stored in an efficient bit encoded manner. This tool decodes requested bit fields and writes them to an output HDF file. The output SDS data type is uint8, uint16 or uint32 depending on the number of unpacked bits.

Note that the unpacked bits are stored in the least significant bits of the output SDS. Refer to the MODIS product file specifications for information on which bits to select for unpacking.

This tool supports 2D/3D/4D SDSs.

The tool command arguments can be specified in any order.

ARGUMENTS

-help [filename] Display this help message. If the input filename is specified with this option, then the names of all the SDS in the file are displayed.

-of=<filename> Output filename.

-sds=<SDS list> List of SDSs to process. SDS names are separated by commas with no space. By default all SDSs are processed maintaining the input SDS interleaving.

To process a specific layer of a 3D SDSs specify the element number of the third dimension as a dot extension of the SDS name: sds_name.n (e.g., sur_refl_b02.1 = the layer defined by the 1st element of the 3rd dimension of the 3D SDS sur_refl_b02).

To process a specific layer of a 4D SDS, specify the higher dimension element number(s) as a dot extension of the SDS name: sds_name.n.m (e.g., Surface_Refl.1.2 = the layer defined by the 1st element of the 3rd dimension and the 2nd element of the 4th dimension of the 4D SDS Surface_Refl).

Note that wildcards and ranges of element values may be specified as sds_name.* and as sds_name.n1-n2.m respectively.

-bit=<Bitnumbers> List of bit numbers separated by commas. A range of bit numbers may be specified using '-' between the starting and ending bit numbers, e.g., -bit=4-8.

Note the bit numbers are zero-based and bits are numbered from right (least significant bit) to left (most significant bit), e.g., in the 8 bit number 10101110 the value of bit 2-4 when unpacked is 011.

-meta Copy metadata from input file to output file.

filename Input filename.

EXAMPLES

```
unpack_sds_bits -sds=Cloud_Mask.1 -bit=1-2 -of=cloud_bits.hdf
MYD35_L2.A2002189.2040.003.2002191125354.hdf
```

{Note: This example unpacks bits 1 and 2 of the layer defined by the 1st element of the 3rd dimension of the SDS Cloud_Mask.}

```
unpack_sds_bits -sds=sur_refl_qc_500m -bit=10-13 -of=srefl_qc_bits.hdf
MOD09A1.A2001193.h09v05.004.2002200065231.hdf
```

```
unpack_sds_bits -sds=sur_refl_qc_500m -bit=10-13,14-17,18-21
-of=srefl_qc_bits.hdf
MOD09A1.A2001193.h09v05.004.2002200065231.hdf
```

```
unpack_sds_bits -sds="most confident detected fire" -bit=0-3
-of=fire_bits.hdf
MOD14A1.A2002185.h30v11.003.2002204204451.hdf
```

AUTHOR

Code: S. Devadiga and Yi Zhang
Documentation: S. Devadiga and D. Roy

Version 1.0, 08/08/2002

Contacts

The EROS Data Center is the main support facility for the LDOPE Tools. Please do not contact the developers directly.

SOFTWARE REQUESTS, BUG REPORTS, AND OTHER COMMENTS SHOULD BE SENT TO:

Web: <http://lpdaac.usgs.gov/tools/ldope/register.html>

Mail: **JOHN DWYER**
LP DAAC Project Scientist
SAIC/EROS Data Center
47914 252nd Street
Sioux Falls, SD 57198
Voice: (605) 594-6060
Fax: (605) 594-6567
Email: dwyer@usgs.gov

DOWNLOAD AND INSTALLATION ASSISTANCE should be sent to:

Mail: **CAROLYN GACKE**
Technical Specialist
SAIC/EROS Data Center
47914 252nd Street
Sioux Falls, SD 57198
Voice: (605) 594-6822
Fax: (605) 594-6963
Email: cgacke@usgs.gov

Appendix A: IRIX Installation

Requirements

These tools work on SGI/IRIX 6.5 Operating Systems.

These tools are available as source code that will need to be compiled or as ready-to-run (run-time) binary executables.

If you wish to compile the source code, you will need the HDF4.1r1 and HDF-EOS2.8 library included with this delivery and a 64-bit C compiler.

If you already have HDF4.1r1 installed on your machine you may not need to install it again, otherwise please follow the instructions for HDF installation provided below. For additional information on the HDF library refer to Appendix E.

It is recommended that a 64-bit C compiler is used. A 32-bit compiler will work but it may generate warning messages during the LDOPE tool installation (e.g., >read_l2g_comp.c warning: decimal constant is so large that it is unsigned).

Instructions for Run-time

1. Save the LDOPE_Irix_bin.tar file to a directory where you want to install the software. Untar the LDOPE_Irix_bin.tar file.

```
tar -xvf LDOPE_Irix_bin.tar
```

Directory LDOPE_Irix_bin should show the two subdirectories: bin and ANCILLARY. Directory 'bin' contains the executables. These were compiled under the Irix 6.5 Operating System using a 64-bit compiler. Directory 'ANCILLIARY' contains the projection parameter files containing the projection parameters and bounding coordinates for all land tiles in Sinusoidal and Integerized Sinusoidal projection.

For convenience you may want to add the binary directory path to the \$PATH environment variable.

2. Set up the environment variable \$ANCPATH to point to the ANCILLARY directory

```
>setenv ANCPATH 'install_directory'/LDOPE_Irix_bin/ANCILLARY
```

where '*install_directory*' is the directory where this software is installed.
3. You may wish to test the correct functioning of the tools using the test data sets described in Appendix G.

Instructions for Source Code

1. Save the LDOPE_irix.tar file to a directory where you want to install the software. Untar the LDOPE_irix.tar file as:

```
>tar -xvf LDOPE_irix.tar
```

The directory LDOPE_Irix should show the following files:

README_Irix_install.pdf (this file in pdf)
README_Irix_install.txt (this file in text)
HDF4.1r1.tar.gz (HDF library install)
MODAT.tar (LDOPE tools install)

2. Untar the MODAT.tar files as:

```
>tar -xvf MODAT.tar
```

The MODAT directory should contain the following files and directories:

LDOPE_IRIX_install.sh (software installation script)
ANCILLARY (directory containing projection parameters)
include (directory containing HDFEOS includes)
lib (directory containing HDFEOS library)
bin (directory containing executables)
obj (directory containing object files)
src (directory containing source code)

3. To compile the source code, the HDF libraries must be installed first. If you already have HDF libraries installed on your machine please go to Step 6.

Gunzip and untar the HDF4.1r1.tar.gz file as:

```
>gunzip HDF4.1r1.tar.gz  
>tar -xvf HDF4.1r1.tar
```

The directory HDF4.1r1 will contain the following files and directories:

README
INSTALL
COPYING (copyrights)
config.guess
config.sub
configure
configure.in
install-sh
Makefile.in
mkinstalldirs
move-if-change
win32mak.zip.uu
config (directory)
hdf (directory)
man (directory)
mfhdf (directory)
release_notes (directory)

4. Read the README file for the HDF directory explanations and instructions. Read the INSTALL file for specific installation requirements and supported operating systems.
5. The basic INSTALL functions will be 'configure' and 'make', taking into account any caveats from Step 3.
6. Ensure that the file privilege of the shell file LDOPE_Irix_install.sh is set to executable as:
 >chmod +x LDOPE_Irix_install.sh

Invoke the installation shell script as:

```
> ./LDOPE_Irix_install.sh
```

This script will prompt you to provide the correct HDF library paths in your system.

Compiled executables will be written to the MODAT/bin directory. For convenience you may want to add this directory path to \$PATH environment variable.

7. Setup the environment variable \$ANCPATH to point to the ANCILLARY directory
 >setenv ANCPATH '*install_directory*'/LDOPE_Irix/ANCILLARY
 where '*install_directory*' is the directory where this software is installed as mentioned above in the ready-to-run section.
8. You may wish to test the correct functioning of the installed tools using the test data sets described in Appendix G.

Contact

LP DAAC Customer Support Center:
http://lpdaac.usgs.gov/cleopatra/user/help_home.asp
LP DAAC Home Page:
<http://lpdaac.usgs.gov>

Appendix B: Linux Installation

Requirements

These tools work on Linux (Redhat 7.3) Operating Systems.

These tools are available as source code that will need to be compiled or as ready-to-run (run-time) binary executables.

If you wish to compile the source code, you will need the HDF4.1r1 and HDF-EOS2.8 library included with this delivery and a 64-bit C compiler.

If you already have HDF4.1r1 installed on your machine you may not need to install it again, otherwise please follow the instructions for HDF installation provided below. For additional information on the HDF library refer to Appendix E.

It is recommended that a 64-bit C compiler is used. A 32-bit compiler will work but it may generate warning messages during the LDOPE tool installation (e.g., >read_l2g_comp.c warning: decimal constant is so large that it is unsigned).

Instructions for Run-time

1. Save the LDOPE_Linux_bin.tar file to a directory where you want to install the software. Untar the LDOPE_Linux_bin.tar file.

```
tar -xvf LDOPE_Linux_bin.tar
```

Directory LDOPE_Linux_bin should show the two subdirectories: bin and ANCILLARY. Directory 'bin' contains the executables. These were compiled under the Redhat 7.3 Operating System using a 64-bit compiler. Directory 'ANCILLIARY' contains the projection parameter files containing the projection parameters and bounding coordinates for all land tiles in Sinusoidal and Integerized Sinusoidal projection.

For convenience you may want to add the binary directory path to the \$PATH environment variable.

2. Set up the environment variable \$ANCPATH to point to the ANCILLARY directory

```
>setenv ANCPATH 'install_directory'/LDOPE_Linux_bin/ANCILLARY
```

where 'install_directory' is the directory where this software is installed.
3. You may wish to test the correct functioning of the tools using the test data sets described in Appendix G.

Instructions for Source Code

1. Save the LDOPE_Linux.tar file to a directory where you want to install the software. Untar the LDOPE_Linux.tar file as:

```
>tar -xvf LDOPE_Linux.tar
```

The directory LDOPE_Linux should show the following files:

README_Linux_install.pdf (this file in pdf)
README_Linux_install.txt (this file in text)
HDF4.1r1.tar.gz (HDF library install)
MODAT.tar (LDOPE tools install)

2. Untar the MODAT.tar files as:

```
>tar -xvf MODAT.tar
```

The MODAT directory should contain the following files and directories:

LDOPE_Linux_install.sh (software installation script)
ANCILLARY (directory containing projection parameters)
include (directory containing HDFEOS includes)
lib (directory containing HDFEOS library)
bin (directory containing executables)
obj (directory containing object files)
src (directory containing source code)

3. To compile the source code, the HDF libraries must be installed first. If you already have HDF libraries installed on your machine please go to Step 6.

Gunzip and untar the HDF4.1r1.tar.gz file as:

```
>gunzip HDF4.1r1.tar.gz  
>tar -xvf HDF4.1r1.tar
```

The directory HDF4.1r1 will contain the following files and directories:

README
INSTALL
COPYING (copyrights)
config.guess
config.sub
configure
configure.in
install-sh
Makefile.in
mkinstalldirs
move-if-change
win32mak.zip.uu
config (directory)
hdf (directory)
man (directory)
mfhdf (directory)
release_notes (directory)

4. Read the README file for the HDF directory explanations and instructions. Read the INSTALL file for specific installation requirements and supported operating systems.
5. The basic INSTALL functions will be 'configure' and 'make', taking into account any caveats from Step 3.
6. Ensure that the file privilege of the shell file LDOPE_Linux_install.sh is set to executable as:
 >chmod +x LDOPE_Linux_install.sh
Invoke the installation shell script as:
 > ./LDOPE_Linux_install.sh

This script will prompt you to provide the correct HDF library paths in your system.

Compiled executables will be written to the MODAT/bin directory. For convenience you may want to add this directory path to \$PATH environment variable.

7. Setup the environment variable \$ANCPATH to point to the ANCILLARY directory
 >setenv ANCPATH '*install_directory*'/LDOPE_Linux/ANCILLARY
where '*install_directory*' is the directory where this software is installed as mentioned above in the ready-to-run section.
8. You may wish to test the correct functioning of the installed tools using the test data sets described in Appendix G.

Contact

LP DAAC Customer Support Center:
http://lpdaac.usgs.gov/cleopatra/user/help_home.asp
LP DAAC Home Page:
<http://lpdaac.usgs.gov>

Appendix C: Solaris Installation

Requirements

These tools work on Solaris 2.8 Operating Systems.

These tools are available as source code that will need to be compiled or as ready-to-run (run-time) binary executables.

If you wish to compile the source code, you will need the HDF4.1r1 and HDF-EOS2.8 library included with this delivery and a C compiler.

The HDF4.1r1 and HDF-EOS2.8 library is included in this delivery and hence installation of the HDF library is not required. Please note that installation of the HDF4.1r1 and HDF-EOS2.8 library will not modify/delete any HDF or HDF-EOS library that you may have already installed on your machine

Instructions for Run-time

1. Save the LDOPE_Solaris_bin.tar file to a directory where you want to install the software. Untar the LDOPE_Solaris_bin.tar file.

```
tar -xvf LDOPE_Solaris_bin.tar
```

Directory LDOPE_Solaris_bin should show the two subdirectories: bin and ANCILLARY. Directory 'bin' contains the executables. These were compiled under the Redhat 7.3 Operating System using a 64-bit compiler. Directory 'ANCILLIARY' contains the projection parameter files containing the projection parameters and bounding coordinates for all land tiles in Sinusoidal and Integerized Sinusoidal projection.

For convenience you may want to add the binary directory path to the \$PATH environment variable.

2. Set up the environment variable \$ANCPATH to point to the ANCILLARY directory

```
>setenv ANCPATH 'install_directory'/LDOPE_Linux_bin/ANCILLARY
```

where 'install_directory' is the directory where this software is installed.
3. You may wish to test the correct functioning of the tools using the test data sets described in Appendix G.

Instructions for Source Code

1. Save the LDOPE_Solaris.tar file to a directory where you want to install the software.

Untar the LDOPE_Solaris.tar file as:

```
>tar -xvf LDOPE_Solaris.tar
```

The directory LDOPE_Solaris should show the following files and directory:
README_Solaris_install.txt (this file in text)

LDOPE_Solaris_install.sh (software installation script)
ANCILLARY (directory containing projection parameters)
include (directory containing HDFEOS includes)
lib (directory containing HDFEOS library)
bin (directory containing executables)
obj (directory containing object files)
src (directory containing source code)

2. To compile the source code :

Ensure that the file privilege of the shell file LDOPE_Solaris_install.sh is set to executable as:

```
>chmod +x LDOPE_Solaris_install.sh
```

Invoke the installation shell script as:

```
> ./LDOPE_Solaris_install.sh
```

This script will prompt you to provide the command of the C compiler on your system (e.g., gcc).

The compiled executables will be written to the LDOPE_Solaris/bin directory. For convenience you may want to add this directory path to \$PATH environment variable.

3. Setup the environment variable \$ANCPATH to point to the ANCILLARY directory

```
>setenv ANCPATH 'install_directory'/LDOPE_Linux/ANCILLARY
```

where 'install_directory' is the directory where this software is installed as mentioned above in the ready-to-run section.

4. You may wish to test the correct functioning of the installed tools using the test data sets described in Appendix G.

Contact

LP DAAC Customer Support Center:
http://lpdaac.usgs.gov/cleopatra/user/help_home.asp
LP DAAC Home Page:
<http://lpdaac.usgs.gov>

Appendix D: Windows Installation

Requirements

These tools work on the Windows Operating System (Windows 2000/XP).

These tools are available as source code that will need to be compiled or as ready-to-run (run-time) binary executables. The Cygwin1.dll dynamic link library is included in this software and is required to support the ready-to-run version executables.

If you wish to compile the source code, you will need The HDF4.1r1 and HDF-EOS2.8 library (included with this delivery), Red Hat Cygwin (not included with this delivery), and a 32-bit or 64-bit C compiler.

The HDF4.1r1 and HDF-EOS2.8 library are included in this software and hence installation of the HDF and HDF-EOS2.8 library is not required. Please note that installation of the HDF4.1r1 and HDF-EOS2.8 library will not modify/delete any HDF or HDF-EOS library that you may have already installed on your machine.

If you wish to compile the source code you will need to install Red Hat Cygwin. If you already have Cygwin installed on your machine you do not need to reinstall it. You may download Red Hat Cygwin from the same webpage you downloaded this software and follow the instruction for Cygwin installation provided below. You may also install the Cygwin software directly (from <http://sources.redhat.com/cygwin/>).

It is recommended that a 64-bit C compiler is used. A 32-bit compiler will work but it may generate warning messages during software compilation (e.g., >read_l2g_comp.c warning: decimal constant is so large that it is unsigned).

Instructions for Run-time

1. Save the LDOPE_Win_bin.zip file to a directory where you want to install the software. Unzip the LDOPE_Win_bin.zip file.
>unzip LDOPE_Win_bin.zip
or use a Windows compression utility like Winzip to unzip the LDOPE_Win_bin.zip file.

Directory LDOPE_Win_bin should show the two subdirectories: bin and ANCILLARY. Directory 'bin' contains the executables. These were compiled under the Windows 2000/XP Operating System using a 64-bit compiler and HDF4.1r1 library. Directory 'ANCILLIARY' contains the projection parameter files containing the projection parameters and bounding coordinates for all land tiles in Sinusoidal and Integerized Sinusoidal projectionTwo subdirectory bin and ANCILLARY will be created.

2. Setup the environment variable \$ANCPATH to point to the ANCILLARY directory '*install_directory*'/LDOPE_Win_bin/ANCILLARY where

'install_directory' is the directory where this software is installed as mentioned above in the ready-to-run section

Under Windows 2000/XP, do the following to set up the environment

\$ANCPATH :

- Click the Windows 'start' button,
- Click 'Control Panel'
- Click the 'System' icon in the Control Panel window, select the 'Advanced' tab, select the button labeled 'Environment Variables'.
- In the Panel labeled 'User variable for username', click on the 'New' button.
- Enter the following in the two small boxes:
 - For 'Variable name', enter ANCPATH.
 - For 'Variable value', enter *'install_directory'*/LDOPE_Win_bin/ANCILLARY
- Click on 'OK's to apply the changes.

For convenience you may want to add the binary directory path to the \$path environment variable.

3. You may wish to test the correct functioning of the tools using the test data sets described in Appendix G.

Instructions for Source Code

1. Save the LDOPE_Win.zip file to a directory where you want to install the software. Unzip the LDOPE_Win.zip file under Cygwin as:
 - >unzip LDOPE_Win.zipor use a Windows compression utility like Winzip to unzip the LDOPE_Win.zip file.

The LDOPE_Win directory should contain the following files and directories:

- README_Win.txt (this file in txt)
- LDOPE_Win_install.sh (software installation script)
- ANCILLARY (directory containing projection parameters)
- bin (directory containing executables & Cygwin1.dll)
- include (directory containing HDF and HDF-EOS includes)
- lib (directory containing HDF and HDF-EOS library)
- obj (directory containing object files)
- src (directory containing source code)
- Cygwin (directory for Cygwin licensing doc)

2. To compile the source code, follow Step 3 (only if you do not have Red Hat Cygwin installed) and then Step 4.
3. If Red Hat Cygwin is not installed:
 - Download the Cygwin_src.zip from the LDOPE Tools download website.

Create a directory called Cygwin under your C drive.
Use a Windows compression utility like Winzip to unzip the cygwin_src.zip under the C:\Cygwin directory. Now the C:\Cygwin directory should contain the following files and directories:

- cygwin.bat (cygwin startup script)
- cygwin.ico (cygwin icon)
- bin (directory)
- etc (directory)
- home (directory)
- lib (directory)
- sbin (directory)
- tmp (directory)
- usr (directory)
- var (directory)

Note, alternatively, you may install Red Hat Cygwin directly from <http://sources.redhat.com/cygwin/>

Now follow step 4 to install the tool software.

4. Ensure that the file privilege of the shell file LDOPE_Win_install.sh is set to executable as
 >chmod +x LDOPE_Win_install.sh

Invoke the installation shell script as:
 > ./LDOPE_Win_install.sh

This script will prompt you to provide the command of the C compiler under your system (e.g., gcc).

The compiled executables will be written to the LDOPE_Win/bin directory. For convenience you may want to add this directory path to \$path environment variable.

5. Setup the environment variable \$ANCPATH to point to the ANCILLARY directory '*install_directory*'\LDOPE_Win_bin\ANCILLARY where '*install_directory*' is the directory where this software is installed as mentioned above in the ready-to-run section

Under Windows 2000/XP, do the following to set up the environment \$ANCPATH :

- Click the Windows 'start' button,
- Click 'Control Panel'
- Click the 'System' icon in the Control Panel window, select the 'Advanced' tab, select the button labeled 'Environment Variables'.
- In the Panel labeled 'User variable for username', click on the 'New' button.
- Enter the following in the two small boxes:

For 'Variable name', enter ANCPATH.
For 'Variable value', enter
'install_directory'/LDOPE_Win_bin/ANCILLARY
Click on 'OK's to apply the changes.

6. You may wish to test the correct functioning of the installed tools using the test data sets described in Appendix G.

Contact

LP DAAC Customer Support Center:
http://lpdaac.usgs.gov/cleopatra/user/help_home.asp
LP DAAC Home Page:
<http://lpdaac.usgs.gov>

Appendix E: README for HDF/netCDF 4.1 Distribution

Obtaining the latest version

The most recent version of the distribution can be obtained from the NCSA ftp archive site at:

`ftp://ftp.ncsa.uiuc.edu/HDF/HDF_Current`

NOTE: The MODIS LDOPE Tools require the use of version HDF4.1r1 of the HDF libraries. More current versions of the HDF libraries are not compatible with the LDOPE software.

Distribution Layout

There are five subdirectories in this directory:

`config` -- contains machine dependent makefile fragments(mh-<os>).

`hdf` -- contains the source code for the HDF 'base library', the multi_file annotation interface, the multi_file raster image interface, HDF command line utilities (hdp, the hdf dumper is in mfhdf/), Pablo instrumentation support, and a test suite. It also contains the software distributions for IJPEG(hdf/jpeg) and ZLIB(hdf/zlib) which are required by the HDF library.
Please see the README in each directory for further info on each package.

`mfhdf` -- contains the netCDF(mfhdf) part of the HDF/netCDF distribution and HDF dumper utility(hdp).

`man` -- incomplete set of man page(s) for HDF.

`release_notes` -- descriptions on new features and bug fixes in this release. Files in this sub-directory can be used as supplemental documentation for HDF4.1. These files are also available on the NCSA ftp server, in /HDF/HDF_Current/release_notes.

Third Party Software Requirements:

1. IJPEG distribution release 6b(libjpeg.a). The "official" site for this is `ftp://ftp.uu.net/graphics/jpeg/jpegsrc.v6b.tar.gz`
2. ZLIB 1.0.4(libz.a) distribution.

Both of these distributions are included with this distribution in 'hdf/jpeg' and 'hdf/zlib'. The HDF/netCDF base distribution is known to work with these versions only.

System Requirements

To build both HDF and netCDF library from source, you need:

- an ANSI C compiler. The native ANSI compilers on the above platforms are supported. On platforms where no ANSI compiler was available the free GNU ANSI compiler GCC was used.
- a Fortran 77 compiler if you want Fortran support. See above table for platforms where Fortran is supported. You can compile both libraries without Fortran support by setting the Fortran compiler variable 'FC = NONE' in the respective makefile fragment(mh-<os>) found in the top-level 'config' directory. See the INSTALL file for further details on configuration and installation.

NOTE: Since the MODIS LDOPE Tools are written in C, they will not be compatible with the FORTRAN version of the HDF libraries.

Configuring/Testing/Installing

See the INSTALL file for instructions on configuring, testing and installing for Unix and non-UNIX systems.

DOCUMENTATION/FAQ/HELP

The HDF 4.1r1 documentation can be found on the NCSA ftp server in the directory /HDF/Documentation/HDF4.1r1. The HDF home page is at:

<http://hdf.ncsa.uiuc.edu/>

An FAQ is available on our ftp server, as well as under "Information about HDF" in the home page.

If you have any questions or comments, or would like to be added to or removed from our hdfnews email list, contact us at: hdfhelp@ncsa.uiuc.edu

Appendix F: LDOPE Tools Command-line Syntax

A LDOPE tool command consists of a command name followed by one or more arguments. There are two types of arguments: arguments indicating certain processing parameters and arguments that are input filenames.

```
command_name    -argument_name=argument_value  filename(s)
```

Arguments that indicate processing options or parameters start with a – sign. Such arguments have argument name and one or more values separated by comma with no space. Argument values may not be allowed if they are implied by the argument name. In cases where argument value is required but is not input, default value is used. Arguments can be input in any order following the command name.

- ***Use of various brackets in the command syntax***
 - <argx or xval> argument ‘argx’ or argument value ‘xval’ is required.
 - [argy or yval] argument ‘argy’ or argument value ‘yval’ is optional.
 - {argz or zval} argument ‘argz’ or argument value ‘zval’ is repeatable.
- ***Command line arguments and values***
 - argx argument ‘argx’ is used without any value.
 - argy=<yval> argument ‘argy’ is used with a value ‘yval’. Argument value is required. ‘yval’ can be string or numeric.
- ***Common arguments and meaning***
 - help Print help message for the command.
 - help filename Print valid values for the various command line arguments of this command.
 - sds Input SDS names.
 - of Output filename.
 - meta Copy metadata from the input file to the output file.
 - bit Bit numbers.
 - xy Pixel location in sample and line number.
 - bn band numbers.
- **MODISLand data product file specification**

Specification for MODISLand data product is available on the web: ftp://mo1.nascom.nasa.gov:9000/pub/stig_temp/mlinda/www/LatestFilespecs/

Appendix G: QA Tool testing

This document serves as a guide to verification testing of the MODIS LDOPE Tools, i.e. to verify that the software has been installed properly and that it executes correctly. The scenarios included in this document will also familiarize the user with various MODIS product characteristics and data structures. The following scenarios will test how the tools work on different Science Data Sets using different combinations of command options.

There are four files of input data that are made available for testing (Testing_Input1.tar.gz, Testing_Input2.tar.gz, Testing_Input3.tar.gz, Testing_Input4.tar.gz), the results from which can be compared to known results that are provided in five files of output data (Testing_Output1.tar.gz, Testing_Output2.tar.gz, Testing_Output3.tar.gz, Testing_Output4.tar.gz, Testing_Output5.tar.gz). All input and output files are in HDF-EOS format.

Testing Input1.tar.gz

MOD09.A2001201.1335.003.2001318080318.hdf
MOD09A1.A2002017.h09v05.004.2002208084151.hdf
MODAGAGG.A2001193.h26v04.003.2001308172749.hdf
MODPTHKM.A2001193.h26v04.004.2002198054041.hdf
MYD021KM.A2002189.0350.003.2002191000652.hdf
MYD021KM.A2002208.1925.003.2002209191726.hdf
MYD14GD.A2002225.h21v08.003.2002227234523.hdf
MYD35_L2.A2002189.2040.003.2002191125354.hdf
MYD35_L2.A2002208.1925.003.2002209192643.hdf
MYD35_L2.A2002208.2110.003.2002209195539.hdf
MYDAGAGG.A2002225.h21v08.003.2002227194922.hdf

Testing Input2.tar.gz

MOD09GST.A2001208.h10v11.004.2002276233729.hdf
MOD10L2G.A2001208.h10v11.004.2002276233845.hdf
MOD29PGD.A2001208.h10v11.003.2002276065837.hdf
MODMGGAN.A2001208.h10v11.004.2002276233721.hdf
MODPT1KN.A2001208.h10v11.004.2002276233717.hdf
MODPTHKM.A2001208.h10v11.004.2002276233708.hdf
MODPTPGD.A2001208.h10v11.003.2002276065455.hdf
MODPTPGN.A2001208.h10v11.003.2002276064938.hdf
MYD021KM.A2002208.1930.003.2002209191241.hdf
MYD021KM.A2002208.2105.003.2002209195055.hdf
MYD021KM.A2002208.2110.003.2002209194936.hdf
MYD09GHK.A2002225.h19v10.003.2002228000126.hdf
MYD09GHK.A2002225.h19v11.003.2002228000539.hdf
MYD09GHK.A2002225.h19v12.003.2002227235146.hdf
MYD14GD.A2002225.h21v05.003.2002228002827.hdf
MYD14GD.A2002225.h21v06.003.2002228002759.hdf
MYD14GD.A2002225.h21v07.003.2002227234318.hdf
MYD43B1.A2002177.h03v08.003.2002210144505.hdf

MYD43B1.A2002177.h11v11.003.2002210233848.hdf

Testing Input3.tar.gz

MOD09A1.A2001201.h20v05.003.2002281080105.hdf
MOD09A1.A2001201.h20v06.003.2002281080112.hdf
MOD09A1.A2001201.h21v05.003.2002281080255.hdf
MOD09A1.A2001201.h21v06.003.2002281080110.hdf
MOD09GHK.A2001208.h10v11.004.2002276233751.hdf
MOD12Q1.A2000289.h21v10.003.2002170195353.hdf
MOD14A1.A2002225.h24v04.003.2002245105445.hdf
MOD15A1.A2001193.h09v05.004.2002198025239.hdf
MYD35_L2.A2002208.1930.003.2002209192055.hdf
MYDAGAGG.A2002225.h20v04.003.2002227203243.hdf
MYDAGAGG.A2002225.h20v05.003.2002227202827.hdf
MYDAGAGG.A2002225.h20v06.003.2002227202457.hdf
MYDAGAGG.A2002225.h21v05.003.2002227203558.hdf
MYDAGAGG.A2002225.h21v06.003.2002227203447.hdf
MYDAGAGG.A2002225.h21v07.003.2002227194908.hdf

Testing Input4.tar.gz

MOD09A1.A2003057.h29v11.004.2003069051044.hdf
MOD09Q1.A2001193.h09v05.004.2002200065231.hdf
MOD43B1.A2001193.h26v04.004.2003134105311.hdf
MOD43B4.A2001193.h30v11.004.2003134233220.hdf
pixels.txt

Testing Input5.tar.gz

MODAGAGG.A2003122.h30v11.004.2003134044435.hdf
MYD09A1.A2003113.h24v04.003.2003134160954.hdf
MYD09GHK.A2002225.h19v09.003.2002227235523.hdf
MYD43B4.A2003081.h26v06.003.2003103182903.hdf

Testing Input6.tar.gz

MOD10A1.A2003122.h30v11.004.2003134090541.hdf
MYD09GQK.A2002225.h19v09.003.2002227235424.hdf
tst_2_mask_l3_sds_MOD43B4.A2001193.h30v11.004.2003134233220.hdf
tst_2_unpack_MOD09A1.A2002017.h09v05.004.2002208084151.hdf

Testing Input7.tar.gz

ERODED_MOD12Q1.A1990001.h09v05.002.1999325030448.hdf
ERODED_MOD12Q1_5KM.A1990001.h26v04.002.1999325060200.hdf
LCS.A2001001.h20v05.hdf
MOD021KM.A2004007.0155.004.2004007095233.hdf
MOD09A1.A2001201.h21v05.003.2002281080255.hdf
MOD09A1.A2001201.h21v06.003.2002281080110.hdf
MOD09A1.A2003057.h29v11.004.2003069051044.hdf
MOD10A1.A2002194.h30v11.003.2002199095914.hdf
MOD11A2.A2003281.h09v05.004.2003294184557.hdf

MOD43B1.A2001193.h26v04.004.2003134105311.hdf
MODAGAGG.A2003077.h19v07.004.2003079210733.hdf
MODAGAGG.A2004047.h11v03.004.2004050014547.hdf
MYD10A1.A2002194.h30v11.003.2002199015759.hdf
MYD14A1.A2003281.h09v05.003.2003296025915.hdf
MYDAGAGG.A2004047.h11v03.004.2004050023819.hdf

Testing Output1.tar.gz

tst_1_enlarge_sds_MOD09.A2001201.1335.003.2001318080318.hdf
tst_2_enlarge_sds_MYD35_L2.A2002189.2040.003.2002191125354.hdf
tst_4_enlarge_sds_MYD43B1.A2002177.h11v11.003.2002210233848.hdf
tst_5_enlarge_sds_MODAGAGG.A2001193.h26v04.003.2001308172749.hdf

Testing Output2.tar.gz

tst_1_sds2bin_MYDAGAGG.A2002225.h21v08.003.2002227194922.img
tst_2_sds2bin_MYD35_L2.A2002208.2110.003.2002209195539.img
tst_3_enlarge_sds_MYD021KM.A2002189.0350.003.2002191000652.hdf
tst_3_sds2bin_MYD021KM.A2002208.1925.003.2002209191726.img
tst_4_sds2bin_MYD021KM.A2002208.1925.003.2002209191726.img
tst_5_sds2bin_MYD43B1.A2002177.h03v08.003.2002210144505.img

Testing Output3.tar.gz

tst_6_enlarge_sds_MODAGAGG.A2001193.h26v04.003.2001308172749.hdf

Testing Output4.tar.gz

tst10_mosaic_MYDAGAGG.A2002225.h20-21v4-8.hdf
tst12_mosaic_MOD09A1.A2001201.h20-21.v05-06.hdf
tst1_mosaic_MYD14GD.A2002225.h21v05-v08.hdf
tst3_mosaic_MYD09GHK.A2002225.h19v10-v12.hdf
tst6_mosaic_MYD021KM.A2002208.1925-2110.hdf
tst_1_L2G_MOD09GST.A2001208.h10v11.004.2002276233729.hdf
tst_1_read_meta_MYD021KM.A2002208.1925.003.2002209191726.txt
tst_1_reduce_rank_MYDAGAGG.A2002225.h21v08.003.2002227194922.txt
tst_1_reduce_sds_MOD15A1.A2001193.h09v05.004.2002198025239.hdf
tst_1_unpack_MOD09A1.A2002017.h09v05.004.2002208084151.hdf
tst_2_L2G_MOD09GHK.A2001208.h10v11.004.2002276233751.hdf
tst_2_read_meta_MYD35_L2.A2002208.2110.003.2002209195539.txt
tst_2_reduce_rank_MYDAGAGG.A2002225.h21v08.003.2002227194922.hdf
tst_2_reduce_sds_MOD14A1.A2002225.h24v04.003.2002245105445.hdf
tst_2_unpack_MOD09A1.A2002017.h09v05.004.2002208084151.hdf
tst_3_L2G_MOD10L2G.A2001208.h10v11.004.2002276233845.hdf
tst_3_read_meta_MODPTHKM.A2001193.h26v04.004.2002198054041.txt
tst_3_reduce_rank_MYDAGAGG.A2002225.h21v08.003.2002227194922.hdf
tst_3_reduce_sds_MOD12Q1.A2000289.h21v10.003.2002170195353.hdf
tst_3_unpack_MYD35_L2.A2002208.1925.003.2002209192643.hdf
tst_4_L2G_MOD29PGD.A2001208.h10v11.003.2002276065837.hdf
tst_4_read_meta_MYDAGAGG.A2002225.h21v08.003.2002227194922.txt
tst_4_reduce_rank_MYDAGAGG.A2002225.h21v08.003.2002227194922.hdf

tst_4_reduce_sds_MYD35_L2.A2002208.1930.003.2002209192055.hdf
tst_4_unpack_MYDAGAGG.A2002225.h21v08.003.2002227194922.hdf
tst_5_L2G_MODMGGAN.A2001208.h10v11.004.2002276233721.hdf
tst_5_read_meta_MYD43B1.A2002177.h11v11.003.2002210233848.txt
tst_5_reduce_rank_MYD021KM.A2002208.1925.003.2002209191726.hdf
tst_6_L2G_MODPTPGN.A2001208.h10v11.003.2002276064938.hdf
tst_6_read_meta_MYD14GD.A2002225.h21v08.003.2002227234523.txt
tst_6_reduce_rank_MYD35_L2.A2002208.1930.003.2002209192055.hdf

Testing Output5.tar.gz

new.tst 2 mask l3 sds MOD43B4.A2001193.h30v11.004.2003134233220.hdf
new.tst 2 unpack MOD09A1.A2002017.h09v05.004.2002208084151.hdf
tst 1 mask sds MOD10A1.A2003122.h30v11.004.2003134090541.hdf
tst 1 MOD09Q1.A2001193.h09v05.004.2002200065231.txt
tst 1 read pixvals MOD09A1.A2003057.h29v11.004.2003069051044.txt
tst 2 mask sds MOD43B4.A2001193.h30v11.004.2003134233220.hdf
tst 2 MYD43B4.A2003081.h26v06.003.2003103182903.txt
tst 2 read pixvals MOD09A1.A2003057.h29v11.004.2003069051044.txt
tst 3 mask sds MOD43B1.A2001193.h26v04.004.2003134105311.hdf
tst 3 read pixvals multiple input file.txt
tst 4 mask sds MYD09A1.A2003113.h24v04.003.2003134160954.hdf
tst 4 read pixvals subpixels input.txt

Testing Output6.tar.gz

tst_1_comp_sds_stat_MOD11A2.A2003281.h09v05.004.2003294184557.txt
tst_1_convert_MOD021KM.A2004007.0155.004.2004007095233.hdf
tst_1_create_mask_MOD09A1.A2001201.h20v05.003.2002281080105.hdf
tst_1_geolocation.txt
tst_1_math_MOD09A1.A2001201.h21v05.003.2002281080255.hdf
tst_1_sds_attr_MYD35_L2.A2002208.2110.003.2002209195539.txt
tst_1_sds_hist.txt
tst_1_sds_range.txt
tst_1_sds_values.txt
tst_1_subset_MYD14A1.A2003281.h09v05.003.2003296025915.hdf
tst_1_tileid.txt
tst_1_transpose_MYD14A1.A2003281.h09v05.003.2003296025915.hdf
tst_1_ts_stat_MOD10A1.A2002194.h30v11.003.2002199095914.hdf
tst_2_comp_sds_stat_MOD43B1.A2001193.h26v04.004.2003134105311.txt
tst_2_convert_MYD021KM.A2002208.1925.003.2002209191726.hdf
tst_2_create_mask_mask_MOD09A1.A2001201.h20v05.003.2002281080105.hdf
tst_2_geolocation.txt
tst_2_math_MYD_MODAGAGG.A2004047.h11v03.hdf
tst_2_sds_attr_MOD09.A2001201.1335.003.2001318080318.txt
tst_2_sds_hist.txt
tst_2_sds_range.txt
tst_2_sds_values.txt
tst_2_subset_MODAGAGG.A2003077.h19v07.004.2003079210733.hdf
tst_2_tileid.txt
tst_2_tranpose_MOD09.A2001201.1335.003.2001318080318.hdf
tst_2_ts_stat_MOD10A1.A2002194.h30v11.003.2002199095914.hdf
tst_3_comp_sds_stat_MOD09A1.A2001201.h20v05.003.2002281080105.txt
tst_3_convert_MYD021KM.A2002189.0350.003.2002191000652.hdf
tst_3_create_mask_MYD35_L2.A2002208.2110.003.2002209195539.hdf

tst_3_subset_MOD021KM.A2004007.0155.004.2004007095233.hdf
tst_4_create_mask_MYDAGAGG.A2002225.h21v08.003.2002227194922.hdf

Enlarge sds

Case 1: enlarge a 2D SDS.

```
enlarge_sds -sds="500m Surface Reflectance Band 1" -sf=4 -  
tst_1_enlarge_sds_MOD09.A2001201.1335.003.2001318080318.hdf -meta  
MOD09.A2001201.1335.003.2001318080318.hdf
```

Case 2: enlarge all layers in 3D SDSs.

```
enlarge_sds -sf=3  
-of=tst_2_enlarge_sds_MYD35_L2.A2002189.2040.003.2002191125354.hdf  
-meta -sds="Cloud_Mask.*,Quality_Assurance.*"  
MYD35_L2.A2002189.2040.003.2002191125354.hdf
```

Case 3: enlarge a subset of layers in a 3D SDS.

```
enlarge_sds -sf=10  
-of=tst_3_enlarge_sds_MYD021KM.A2002189.0350.003.2002191000652.hdf  
-sds="EV_1KM_RefSB.4-6" MYD021KM.A2002189.0350.003.2002191000652.hdf
```

Case 4: enlarge all 3rd dimension layers of a specified 4th dimension of a 4D SDS.

```
enlarge_sds -sds="BRDF_Albedo_Parameters*.2" -sf=2  
-of=tst_4_enlarge_sds_MYD43B1.A2002177.h11v11.hdf  
MYD43B1.A2002177.h11v11.003.2002210233848.hdf
```

Case 5: enlarge all 3rd dimension layers of specified or all 4th dimension in 4D SDSs.

```
enlarge_sds -sf=2  
-of=tst_5_enlarge_sds_MODAGAGG.A2001193.h26v04.003.2001308172749.hdf  
-meta -sds="Angles*.1,Surface_Refl.*.*"  
MODAGAGG.A2001193.h26v04.003.2001308172749.hdf
```

Case 6: Test default option of `enlarge_sds`. It will enlarge all SDS, all layers, and retain original interleaving.

```
enlarge_sds -sf=3 -meta -  
of=tst_6_enlarge_sds_MODAGAGG.A2001193.h26v04.003.2001308172749.hdf  
MODAGAGG.A2001193.h26v04.003.2001308172749.hdf
```

read_meta:

Case 1: Test `read_meta` default option

```
read_meta MYD021KM.A2002208.1925.003.2002209191726.hdf
```

Result: Print to screen. Saved as
tst_1_read_meta_MYD021KM.A2002208.1925.003.2002209191726.txt for
comparison purpose.

Case 2: read only Core metadata.

```
read_meta -core MYD35_L2.A2002208.2110.003.2002209195539.hdf
```

Result: Print to screen. Saved as
tst_2_read_meta_MYD35_L2.A2002208.2110.003.2002209195539.txt for
comparison purpose.

Case 3: read only QA metadata.

```
read_meta -qa MODPTHKM.A2001193.h26v04.004.2002198054041.hdf
```

Result: Print to screen. Saved as
tst_3_read_meta_MODPTHKM.A2001193.h26v04.004.2002198054041.txt for
comparison purpose.

Case 4: read only Archive metadata.

```
read_meta -arch MYDAGAGG.A2002225.h21v08.003.2002227194922.hdf
```

Result: Print to screen. Saved as
tst_4_read_meta_MYDAGAGG.A2002225.h21v08.003.2002227194922.txt for
comparison purpose.

Case 5: read selected metadata.

```
read_meta -meta=INPUTPOINTER  
MYD43B1.A2002177.h11v11.003.2002210233848.hdf
```

Result: Print to screen. Saved as
tst_5_read_meta_MYD43B1.A2002177.h11v11.003.2002210233848.txt for
comparison purpose.

Case 6: read selected metadata, case insensitive.

```
read_meta -meta=DAYNIGHTFLAG,GRINGPOINTLONGITUDE  
MYD14GD.A2002225.h21v08.003.2002227234523.hdf
```

Result: Print to screen. Saved as
tst_6_read_meta_MYD14GD.A2002225.h21v08.003.2002227234523.txt for
comparison purpose.

Unpack_sds_bits

Case 1: unpack a single SDS.

```
unpack_sds_bits -sds=sur_refl_gc_500m -bit=10-13  
MOD09A1.A2002017.h09v05.004.2002208084151.hdf -of=  
tst_1_unpack_MOD09A1.A2002017.h09v05.004.2002208084151.hdf
```


Case 2: unpack bits of different range of a single SDS.

```
unpack_sds_bits -sds=sur_refl_qc_500m -bit=10-13,14-17,18-21
MOD09A1.A2002017.h09v05.004.2002208084151.hdf -of=
tst_2_unpack_MOD09A1.A2002017.h09v05.004.2002208084151.hdf
```

Case 3: unpack bits of a selected 3rd dimension of a 3D SDS.

```
unpack_sds_bits -sds=Cloud_Mask.1 -meta
-of=tst_3_unpack_MYD35_L2.A2002208.1925.003.2002209192643.hdf -bit=1-2
MYD35_L2.A2002208.1925.003.2002209192643.hdf
```

Case 4: unpack bits of different range of a list of SDSs.

```
unpack_sds_bits -sds=Surface_Refl,Band_QC -bit=1-3,4-6
-of=tst_4_unpack_MYDAGAGG.A2002225.h21v08.003.2002227194922.hdf
MYDAGAGG.A2002225.h21v08.003.2002227194922.hdf
```

read_l2g

Case 1: extracts layers from a L2G file. (for this example, the output contains 4 SDSs).

```
read_l2g -layer=1-2
-of= tst_1_L2G_MOD09GST.A2001208.h10v11.004.2002276233729.hdf
MOD09GST.A2001208.h10v11.004.2002276233729.hdf
```

Case 2: extracts a list of SDSs of specified granule numbers and layers of a L2G file. (for this example, the output contains 9 SDSs).

```
read_l2g -sds=sur_refl_b01,sur_refl_b03,sur_refl_b04 -layer=1-2
-glist=8552,187 -ptr=MODPTHKM.A2001208.h10v11.004.2002276233708.hdf
-of=tst_2_L2G_MOD09GHK.A2001208.h10v11.004.2002276233751.hdf
MOD09GHK.A2001208.h10v11.004.2002276233751.hdf
```

Case 3: extract SDS by input layer number and granule pointer index. (for this example, 2 SDSs are outputted.)

```
read_l2g -sds=Snow_Cover -layer=2 -gpidx=0
-ptr=MODPTHKM.A2001208.h10v11.004.2002276233708.hdf
-of=tst_3_L2G_MOD10L2G.A2001208.h10v11.004.2002276233845.hdf
MOD10L2G.A2001208.h10v11.004.2002276233845.hdf
```

Case 4: extract all SDSs of a list of granule numbers from a L2G file. (for this example, 24 SDSs are outputted.)

```
read_l2g -glist=8548,102,103 -glist=8549,122,123
-ptr=MODPTPGD.A2001208.h10v11.003.2002276065455.hdf
-of=tst_4_L2G_MOD29PGD.A2001208.h10v11.003.2002276065837.hdf
MOD29PGD.A2001208.h10v11.003.2002276065837.hdf
```

Case 5: extract all SDSs of a specified granule id from a L2G file. (for this example, 6 SDSs are outputted.)

```
read_l2g -gid=A2001208.0430
-ptr=MODPT1KN.A2001208.h10v11.004.2002276233717.hdf
-of=tst_5_L2G_MODMGGAN.A2001208.h10v11.004.2002276233721.hdf
MODMGGAN.A2001208.h10v11.004.2002276233721.hdf
```

Case 6: extract all SDSs of a list of layers from a L2G pointer file. (for this example, 36 SDSs are outputted.)

```
read_l2g -layer=1-6
-of=tst_6_L2G_MODPTPGN.A2001208.h10v11.003.2002276064938.hdf
MODPTPGN.A2001208.h10v11.003.2002276064938.hdf
```

sds2bin

Case 1: Convert a 4D SDS to binary format.

```
sds2bin -sds=Surface_Ref1
-of=tst_1_sds2bin_MYDAGAGG.A2002225.h21v08.003.2002227194922.img
MYDAGAGG.A2002225.h21v08.003.2002227194922.hdf
```

Case 2: Convert a 3D SDS to binary format.

```
sds2bin -sds=Quality_Assurance.*
-of=tst_2_sds2bin_MYD35_L2.A2002208.2110.003.2002209195539.img
MYD35_L2.A2002208.2110.003.2002209195539.hdf
```

Case 3: Convert selected layers from a 3D SDS to binary format.

```
sds2bin -sds=EV_1KM_Emissive.2-3
-of=tst_3_sds2bin_MYD021KM.A2002208.1925.003.2002209191726.img
MYD021KM.A2002208.1925.003.2002209191726.hdf
```

Case 4: Convert a 2D SDS to binary format.

```
sds2bin -sds=EV_Band26
-of=tst_4_sds2bin_MYD021KM.A2002208.1925.003.2002209191726.img
MYD021KM.A2002208.1925.003.2002209191726.hdf
```

Case 5: Convert selected layers from a 4D SDS to binary format.

```
sds2bin -sds=BRDF_Albedo_Parameters.1-2.2-3
-of=tst_5_sds2bin_MYD43B1.A2002177.h03v08.003.2002210144505.img
MYD43B1.A2002177.h03v08.003.2002210144505.hdf
```

reduce_sds_rank

Case 1: Show how to use the help to get SDS dimension information(name, size) which is useful for processing the data.

```
reduce_sds_rank -help MYDAGAGG.A2002225.h21v08.003.2002227194922.hdf
```

The output is saved as :

```
tst_1_reduce_rank_MYDAGAGG.A2002225.h21v08.003.2002227194922.txt
```

Case 2: Reduce the rank of 4D SDSs.

```
reduce_sds_rank  
-of=tst_2_reduce_rank_MYDAGAGG.A2002225.h21v08.003.2002227194922.hdf  
MYDAGAGG.A2002225.h21v08.003.2002227194922.hdf -all
```

Case 3: Reduce selected SDS by specified dimension name and range from a 4D SDS.

```
reduce_sds_rank -sds=Angles -dim=Num_Obs_Max,1,4 -dim=Num_Angles,1-4  
-of=tst_3_reduce_rank_MYDAGAGG.A2002225.h21v08.003.2002227194922.hdf  
MYDAGAGG.A2002225.h21v08.003.2002227194922.hdf -all
```

Case 4: Similar to case 3, but to demo how to selected more than one SDS.

```
reduce_sds_rank -sds=Surface_Refl -dim=Num_Obs_Max,1,2  
-dim=Num_Land_Bands,1-7 -sds=Band_QC -dim=Num_Obs_Max,1,4  
-dim=Num_Band_QC,1-7  
-of=tst_4_reduce_rank_MYDAGAGG.A2002225.h21v08.003.2002227194922.hdf  
MYDAGAGG.A2002225.h21v08.003.2002227194922.hdf
```

Case 5: Reduce a 3D SDS. (band sequential format)

```
reduce_sds_rank -sds=EV_1KM_RefSB -dim=Band_1KM_RefSB,1-15  
-of=tst_5_reduce_rank_MYD021KM.A2002208.1925.003.2002209191726.hdf  
MYD021KM.A2002208.1925.003.2002209191726.hdf -all
```

Case 6: Reduce a 3D SDS. (Band Interleaved by Pixel format)

```
reduce_sds_rank -sds=Quality_Assurance  
-of=tst_8_reduce_rank_MYD35_L2.A2002208.1930.003.2002209192055.hdf  
MYD35_L2.A2002208.1930.003.2002209192055.hdf -all
```

mosaic_sds

Case 1: Create Mosaic of a 2D SDS.

```
mosaic_sds -sds=fire_mask_1  
-of=tst1_mosaic_MYD14GD.A2002225.h21v05-v08.hdf  
MYD14GD.A2002225.h21v05.003.2002228002827.hdf  
MYD14GD.A2002225.h21v06.003.2002228002759.hdf  
MYD14GD.A2002225.h21v07.003.2002227234318.hdf  
MYD14GD.A2002225.h21v08.003.2002227234523.hdf
```

Case 2: Create Mosaic of multiple 2D SDSs.

```
mosaic_sds
-sds=sur_refl_b01_1,sur_refl_b02_1,sur_refl_b03_1,sur_refl_b04_1
-of=tst3_mosaic_MYD09GHK.A2002225.h19v10-v12.hdf
MYD09GHK.A2002225.h19v10.003.2002228000126.hdf
MYD09GHK.A2002225.h19v11.003.2002228000539.hdf
MYD09GHK.A2002225.h19v12.003.2002227235146.hdf
```

Case 3: Create Mosaic of a specified layer from a 3D SDS.

```
mosaic_sds -sds="EV_1KM_Emissive.2"
-of=tst6_mosaic_MYD021KM.A2002208.1925-2110.hdf
MYD021KM.A2002208.1925.003.2002209191726.hdf
MYD021KM.A2002208.1930.003.2002209191241.hdf
MYD021KM.A2002208.2105.003.2002209195055.hdf
MYD021KM.A2002208.2110.003.2002209194936.hdf
```

Case 4: Create Mosaic of a specified layer from a 4D SDS.

```
mosaic_sds -sds="Surface_Refl.1.1.2"
-of=tst10_mosaic_MYDAGAGG.A2002225.h20-21v4-8.hdf
MYDAGAGG.A2002225.h20v04.003.2002227203243.hdf
MYDAGAGG.A2002225.h20v05.003.2002227202827.hdf
MYDAGAGG.A2002225.h20v06.003.2002227202457.hdf
MYDAGAGG.A2002225.h21v05.003.2002227203558.hdf
MYDAGAGG.A2002225.h21v06.003.2002227203447.hdf
MYDAGAGG.A2002225.h21v07.003.2002227194908.hdf
MYDAGAGG.A2002225.h21v08.003.2002227194922.hdf
```

Case 5: Create mosaic from selected areas of each tile.

```
mosaic_sds -sds=sur_refl_b01,sur_refl_b02
-of= tst12_mosaic_MOD09A1.A2001201.h20-21.v05-06.hdf
MOD09A1.A2001201.h20v05.003.2002281080105.hdf -row=1500,2399
-col=1000,2399 MOD09A1.A2001201.h20v06.003.2002281080112.hdf
-row=0,1200 -col=1000,2399
MOD09A1.A2001201.h21v05.003.2002281080255.hdf -row=1500,2399
-col=0,1200 MOD09A1.A2001201.h21v06.003.2002281080110.hdf
-row=0,1200 -col=0,1200
```

reduce_sds

Case 1: reduce sds using the -sub option.

```
reduce_sds -sds=Fpar_1km -sub -rf=5 -meta
-of=tst_1_reduce_sds_MOD15A1.A2001193.h09v05.004.2002198025239.hdf
MOD15A1.A2001193.h09v05.004.2002198025239.hdf
```

Case 2: reduce sds using the -cnt and -bit option.

```
reduce_sds -sds="most confident detected fire" -rf=5 -cnt
-bit="0-3<=2,0-3==3,0-3==4,0-3==5,0-3==6,0-3==7" -meta
-of=tst_2_reduce_sds_MOD14A1.A2002225.h24v04.003.2002245105445.hdf
MOD14A1.A2002225.h24v04.003.2002245105445.hdf
```

Case 3: reduce sds using the -cl option

```
reduce_sds -rf=4 -cl -meta
-of= tst_3_reduce_sds_MOD12Q1.A2000289.h21v10.003.2002170195353.hdf
MOD12Q1.A2000289.h21v10.003.2002170195353.hdf -meta
```

Case 4: reduce sds using the -avg and -min, -max, -std, -num option.

```
reduce_sds -sds=Cloud_Mask.1-2 -avg -rf=5 -avg -min -max -std -num -of=
tst_4_reduce_sds_MYD35_L2.A2002208.1930.003.2002209192055.hdf
MYD35_L2.A2002208.1930.003.2002209192055.hdf -meta
```

read_pixvals

Case 1: Read pixel value from a MODLand HDF-EOS data product by specify the pixel column number and row number (0-based) in the command line.

```
read_pixvals -xy=2000,281 MOD09A1.A2003057.h29v11.004.2003069051044.hdf
```

Result: Print to screen. Saved as

tst_1_read_pixvals_MOD09A1.A2003057.h29v11.004.2003069051044.txt for comparison purpose.

Case 2: Read multiple pixel values from a MODLand HDF-EOS data product by specify the pixel column numbers and row numbers (0-based) in the a file.

```
read_pixvals -xy=pixels.txt MOD09A1.A2003057.h29v11.004.2003069051044.hdf
```

Pixels.txt contains the coordinates of each pixel. (one point per line)

Result: Print to screen. Saved as

tst_2_read_pixvals_MOD09A1.A2003057.h29v11.004.2003069051044.txt for comparison purpose.

Case 3: Read pixel value from several MODLand HDF-EOS data products by specify the pixel column number and row number (0-based) in the command line.

```
read_pixvals -xy=10,10 MYD09GHK.A2002225.h19v09.003.2002227235523.hdf
MYD09GQK.A2002225.h19v09.003.2002227235424.hdf
```

Result: Print to screen. Saved as tst_3_read_pixvals_multiple_input_file.txt for comparison purpose.

Case 4: Read a subsample pixel value from a MODLand HDF-EOS data product by specified the finer resolution as additional subscript to row and column input.

```
read_pixvals -res=hkm -xy=50.1,50.0
MYD09GHK.A2002225.h19v09.003.2002227235523.hdf
MYD09GQK.A2002225.h19v09.003.2002227235424.hdf
```

Result: Print to screen. Saved as

tst_4_read_pixvals_subpixels_input.txt for comparison purpose.

mask_sds

Case 1: mask 2D SDSs of an MODLand L3 data product.

```
mask_sds -sds=Day_Tile_Snow_Cover -fill=225
-mask=MOD10A1.A2003122.h30v11.004.2003134090541.hdf,Snow_Spatial_QA,0-1==00
-of=tst_1_mask_sds_MOD10A1.A2003122.h30v11.004.2003134090541.hdf
MOD10A1.A2003122.h30v11.004.2003134090541.hdf
```

Case 2: mask a specified layer of a 3D SDS of an MODLand L3 data product.

```
mask_sds -sds=Nadir_Reflectance.2 -fill=-5000 -
mask=MOD43B4.A2001193.h30v11.004.2003134233220.hdf,Nadir_Reflectance.2,<=10000,
AND,* ,*,15==0,AND,* ,Nadir_Reflectance_Quality.1,1==0,AND,* ,Nadir_Reflectance_Qu
ality.2,7==0
-of=tst_2_mask_sds_MOD43B4.A2001193.h30v11.004.2003134233220.hdf
MOD43B4.A2001193.h30v11.004.2003134233220.hdf
```

Case 3: mask a specified layer of a 4D SDS of an MODLand L3 data product.

```
mask_sds -sds=BRDF_Albedo_Parameters.1.1 -fill=-2000 -
mask=MOD43B1.A2001193.h26v04.004.2003134105311.hdf,BRDF_Albedo_Parameters.1.1,<
=1000,AND,* ,*,>=0,AND,* ,BRDF_Albedo_Quality.1,1==0,AND,* ,BRDF_Albedo_Quality.2,
3==0 -of=tst_3_mask_sds_MOD43B1.A2001193.h26v04.004.2003134105311.hdf
MOD43B1.A2001193.h26v04.004.2003134105311.hdf
```

Case 4: mask two 2D SDS of an MODLand L3 data product by define the mask as combination of individual masks separated by the logical operator AND and OR.

```
mask_sds -sds=sur_refl_b01,sur_refl_b02 -fill=-5000 -
mask=MYD09A1.A2003113.h24v04.003.2003134160954.hdf,sur_refl_gc_500m,1==0,AND,* ,
*,2-9==00000000,AND,* ,sur_refl_state_500m,7-10==0000,AND,* ,*,15==0 -
of=tst_4_mask_sds_MYD09A1.A2003113.h24v04.003.2003134160954.hdf
MYD09A1.A2003113.h24v04.003.2003134160954.hdf
```

read_proj_param

Case 1: read the projection parameter metadata in the input HDF-EOS product (collection 4)

```
read_proj_param MOD09Q1.A2001193.h09v05.004.2002200065231.hdf
```

Result: Print to screen. Saved as `tst_1_MOD09Q1.A2001193.h09v05.004.2002200065231.txt` for comparison purpose.

Case 2: example on another HDF-EOS product (collection 3)

```
read_proj_param MYD43B4.A2003081.h26v06.003.2003103182903.hdf
```

Result: Print to screen. Saved as `tst_2_MYD43B4.A2003081.h26v06.003.2003103182903.txt` for comparison purpose.

cp_proj_param

NOTE:

You should setup the environment variable ANCPATH first before running this tool.
e.g.: `setenv ANCPATH /ltp/home/LDOPE_Linux_drop2/ancillary`

Case 1: copy the required projection metadata from an ancillary projection parameter file by specifying the tile id of the input file.

```
cp_proj_param -tile=h09v05 -proj=SIN  
tst_2_unpack_MOD09A1.A2002017.h09v05.004.2002208084151.hdf  
-of=new.tst_2_unpack_MOD09A1.A2002017.h09v05.004.2002208084151.hdf
```

Case 2: copy the projection metadata from another HDF-EOS file containing the projection metadata.

```
cp_proj_param tst_2_mask_l3_sds_MOD43B4.A2001193.h30v11.004.2003134233220.hdf  
-ref=MOD43B4.A2001193.h30v11.004.2003134233220.hdf  
-of=new.tst_2_mask_l3_sds_MOD43B4.A2001193.h30v11.004.2003134233220.hdf
```

create_mask

Case 1: uses a single mask in the `-mask` option and user specified output ON and OFF values.

```
create_mask -of=tst_1_create_mask_MOD09A1.A2001201.h20v05.003.2002281080105.hdf  
-mask="MOD09A1.A2001201.h20v05.003.2002281080105.hdf,sur_refl_state_500m,3-5==001"  
-on=255 -off=0
```

Case 2: uses logical combination of two masks used in the `-mask` option

```
create_mask -of=tst_2_create_mask_MOD09A1.A2001201.h20v05.003.2002281080105.hdf  
-mask="MOD09A1.A2001201.h20v05.003.2002281080105.hdf,sur_refl_state_500m,3-5==001,AND,*,*,01==00" -on=100 -off=0
```

Case 3: uses 2D layer of a 3D SDS in the `-mask` option

```
create_mask -of=tst_3_create_mask_MYD35_L2.A2002208.2110.003.2002209195539.hdf  
-on=1 -off=0 -mask="MYD35_L2.A2002208.2110.003.2002209195539.hdf,Cloud_Mask.1,1-
```

2==00,AND,*,*,6-7==11"

Case 4: uses 2D layer of a 4D SDS in the -mask option

```
create_mask -of=tst_4_create_mask_MYDAGAGG.A2002225.h21v08.003.2002227194922.hdf  
-mask="MYDAGAGG.A2002225.h21v08.003.2002227194922.hdf,Band_QC.1.1,2-  
5==1100,AND,*,Aggregate_QC.1.1,3-==001" -on=1 -off=0
```

tile_id

Case 1: use SIN projection

```
tile_id -proj=SIN -lon=45.0 -lat=45.0
```

Case 2: use ISIN projection

```
tile_id -proj=ISIN -lon=91 -lat=54
```

geolocation

Case 1: example using the -hv, proj, and resolution option

```
geolocation -xy=750,500 -hv=12,4 -proj=SIN -res=qkm
```

Case 2: example using an input file (-hv, proj and resolution are implied)

```
geolocation -xy=750,500 MOD09A1.A2001201.h20v05.003.2002281080105.hdf
```

comp_sds_range

Case 1: output SDS range for each layer of all SDS in the input file

```
comp_sds_range -layer MYD43B1.A2002177.h11v11.003.2002210233848.hdf >  
tst_1_comp_sds_range.txt
```

Case 2: output SDS range and the scale for user specified SDS in the input file

```
comp_sds_range -sds=sur_refl_b01,sur_refl_b03,sur_refl_b04 -scale  
MOD09A1.A2001201.h20v05.003.2002281080105.hdf > tst_2_comp_sds_range.txt
```

comp_sds_values

Case 1: output SDS values in each layer of user specified SDS in the input file.

```
comp_sds_values -sds=Band_QC -layer MYDAGAGG.A2002225.h21v08.003.2002227194922.hdf >  
tst_1_comp_sds_values.txt
```

Case 2: output SDS values in the user specified SDS

```
comp_sds_values -sds=sur_refl_b02 MOD09A1.A2001201.h20v05.003.2002281080105.hdf  
> tst_2_comp_sds_values.txt
```

comp_sds_hist

Case 1: compute histogram of SDS values in the user specified SDS

```
comp_sds_hist -sds=sur_refl_b02 MOD09A1.A2001201.h20v05.003.2002281080105.hdf >
tst_1_comp_sds_hist.txt
```

Case 2: compute histogram of SDS values for each layer of input 4D SDS

```
comp_sds_hist -sds=Surface_Refl -range=1000,3000 -layer
MYDAGAGG.A2002225.h21v08.003.2002227194922.hdf > tst_2_comp_sds_hist.txt
```

comp_sds_stat

Case 1: compute stat SDS for all SDS in the input HDF file.

```
comp_sds_stat .MOD11A2.A2003281.h09v05.004.2003294184557.hdf -
class=ERODED_MOD12Q1.A1990001.h09v05.002.1999325030448.hdf,Erode_Land_Cover_Type_2
```

Case 2: compute stat SDS of user specified 2D layer of a 4D SDS

```
comp_sds_stat MOD43B1.A2001193.h26v04.004.2003134105311.hdf -
class=ERODED_MOD12Q1_5KM.A1990001.h26v04.002.1999325060200.hdf,Erode_Land_Cover_Type
_2 -sds=BRDF_Albedo_Parameters.1.2
```

Case 3: compute stat SDS for a spatial subset of input file

```
comp_sds_stat MOD09A1.A2001201.h20v05.003.2002281080105.hdf -
class=LCS.A2001001.h20v05.hdf,Land_Cover_Type_1_Secondary -reg=0,100,50,150
```

convert_l1b_data

Case 1: convert all L1B bands (reflective and emissive bands) in the input file

```
convert_l1b_data MOD021KM.A2004007.0155.004.2004007095233.hdf -
of=tst_1_convert_MOD021KM.A2004007.0155.004.2004007095233.hdf -meta
```

Case 2: process only bands 3-7 in the input file.

```
convert_l1b_data -bn=3,4,5,6,7
-of=tst_2_convert_MYD021KM.A2002208.1925.003.2002209191726.hdf
MYD021KM.A2002208.1925.003.2002209191726.hdf
```

Case 3: process only band 31 in the input file

```
convert_l1b_data -bn=31 -meta -of=tst_3_convert_MYD021KM.A2002189.0350.003.2002191000652.hdf
MYD021KM.A2002189.0350.003.2002191000652.hdf
```

math_sds

Case 1: pixelwise division of 2D SDS.

```
math_sds -
math="sur_refl_b02,MOD09A1.A2001201.h21v05.003.2002281080255.hdf,/,sur_refl_b03,MOD09A1.A2
001201.h21v06.003.2002281080110.hdf,INT16,*,*,*,**"
-of=tst_1_math_MOD09A1.A2001201.h21v05.003.2002281080255.hdf
```

Case 2: pixelwise subtraction of 2D layer of 4D SDS

```
math_sds
-math="Surface_Refl.1.1,MODAGAGG.A2004047.h11v03.004.2004050014547.hdf,-
,Surface_Refl.1.4,MYDAGAGG.A2004047.h11v03.004.2004050023819.hdf,*,*,*,*"
-of=tst_2_math_MYD_MODAGAGG.A2004047.h11v03.hdf
```

transpose_sds

Case 1: transpose a 3D SDS

```
transpose_sds -sds="most confident detected fire" MYD14A1.A2003281.h09v05.003.2003296025915.hdf
-of=tst_1_transpose_MYD14A1.A2003281.h09v05.003.2003296025915.hdf -meta
```

Case 2: transpose user specified 2D SDSs

```
transpose_sds -sds="500m Surface Reflectance Band 1,500m Surface Reflectance Band 3,500m Surface
Reflectance Band 4" MOD09.A2001201.1335.003.2001318080318.hdf -
of=tst_2_tranpose_MOD09.A2001201.1335.003.2001318080318.hdf
```

read_sds_attributes

Case 1: output attributes of all SDS in the input file

```
read_sds_attributes MYD35_L2.A2002208.2110.003.2002209195539.hdf
```

Case 2: output attributes of user specified SDSs in the input file

```
read_sds_attributes -sds="500m Surface Reflectance Band 1,500m Surface Reflectance Band 3,500m
Surface Reflectance Band 4" MOD09.A2001201.1335.003.2001318080318.hdf
```

create_sds_ts_stat

Case 1: create SDS containing minimum of SDS values in the input files

```
create_sds_ts_stat -sds="Day_Tile_Snow_Cover,*,*,*,*" -param=min
MOD10A1.A2002194.h30v11.003.2002199095914.hdf
MYD10A1.A2002194.h30v11.003.2002199015759.hdf -
of=tst_1_ts_stat_MOD10A1.A2002194.h30v11.003.2002199095914.hdf
```

Case 2: create SDS of maximum values for two SDSs in the input files

```
create_sds_ts_stat -sds="Day_Tile_Snow_Cover,*,*,*,*" -sds="Snow_Spatial_QA,*,*,*,*" -
param=max
MOD10A1.A2002194.h30v11.003.2002199095914.hdf
MYD10A1.A2002194.h30v11.003.2002199015759.hdf -
of=tst_2_ts_stat_MOD10A1.A2002194.h30v11.003.2002199095914.hdf
```

subset_sds

Case 1: extract a spatial subset of a 3D SDS

```
subset_sds -sds="most confident detected fire" -row=0,10 -col=1130,1140
MYD14A1.A2003281.h09v05.003.2003296025915.hdf
```

```
-of=tst_1_subset_MYD14A1.A2003281.h09v05.003.2003296025915.hdf
```

Case 2: extract a spatial subset of a 4D SDS

```
subset_sds -sds=Surface_Refl -row=10,100 -col=100,200  
MODAGAGG.A2003077.h19v07.004.2003079210733.hdf  
-of=tst_2_subset_MODAGAGG.A2003077.h19v07.004.2003079210733.hdf
```

Case 3: extract a spatial subset of a 2D SDS

```
subset_sds -sds=EV_Band26 -row=300,400 -col=1000,1100  
MOD021KM.A2004007.0155.004.2004007095233.hdf -  
of=tst_3_subset_MOD021KM.A2004007.0155.004.2004007095233.hdf
```

Appendix H: Understanding unpack_sds_bits

All of the MODIS Land products store quality assessment and other important information at the bit level within certain Science Data Sets (SDS). Most basic computer text books describe how numbers are stored in bytes which are composed of 8, 16, 32 or 64 bits, with each bit storing a binary (0 or 1) value.

The MODIS Land products store quality assessment and other information, such as the land-sea mask, logical criteria used by the algorithm, and cloud state, in an efficient bit encoded manner for each pixel. The `unpack_sds_bits` tool decodes requested bit fields and writes them to an output HDF file containing a new SDS that can be viewed with software supporting the HDF or HDF-EOS data formats.

Prior to unpacking the bits, it is important to consult the product User Guide or File Specification to determine the SDS names and the range and definition of the bits for unpacking. Each product will vary in its SDS names and its bit coding and content. The User Guides and File Specifications are available on the internet and may be updated periodically. Please follow the links at http://landweb.nascom.nasa.gov/QA_WWW/ to find these information.

Example of unpacking file: MOD13A1.A2001225.h11v05.003.2001342094517.hdf

1. Analyze the NDVI quality for MOD13A1, the 500m 16-day MODIS VI product.
2. Reference the MOD13A1 User Guide (http://tbrs.arizona.edu/projects/modis/UserGuide_doc.htm#QASDS1) or File Specification listing (ftp://mol.gsfc.nasa.gov:9000/pub/stig_temp/mlinda/www/LatestFilespecs/MOD13A1.fs) for SDS names and bit information:
3. To get list of SDS names in the input file use
`unpack_sds_bits -help MOD13A1.A2001225.h11v05.003.2001342094517.hdf`
 - 500m 16 days NDVI (2400 x 2400) INT16
 - 500m 16 days EVI (2400 x 2400) INT16
 - 500m 16 days NDVI Quality (2400 x 2400) UINT16
 - 500m 16 days EVI Quality (2400 x 2400) UINT16
 - 500m 16 days red reflectance (2400 x 2400) INT16
 - 500m 16 days NIR reflectance (2400 x 2400) INT16
 - 500m 16 days blue reflectance (2400 x 2400) INT16
 - 500m 16 days MIR reflectance (2400 x 2400) INT16
 - 500m 16 days average view zenith angle (2400 x 2400) INT16
 - 500m 16 days average sun zenith angle (2400 x 2400) INT16
 - 500m 16 days average relative azimuth angle (2400 x 2400) INT16

4. Referencing the User Guide

(http://tbrs.arizona.edu/projects/modis/UserGuide_doc.htm#QASDS1)

Get explanation for values of bits 0-1 and 2-5 for the SDS name '500m 16 days NDVI Quality'.

Bit Descriptions:

0-1 NDVI Quality

00 NDVI produced, good quality

01 NDVI produced, but check QA

10 (pixel not produced, due to cloud effects)

11 (pixel not produced due to other reasons than cloud)

Unpacking bits 0-1 would create a SDS band with the values 0, 1, 2, 3

Bit value 00=0, bit value 01=1, bit value 10=2, bit value 11=3. The digital value of 0 indicates good quality.

Note: Binary conversion is read from right to left. If the bit value was 00011001, the decimal value would be 25. In another example related to the 500m Surface Reflectance QC bits, the bit value of 00001111 converts to a decimal value of 15.

2-5 VI Usefulness Index

0000 Perfect quality (equal to VI quality = 00: VI produced with good quality)

0001 High quality

0010 Good quality

0011 Acceptable quality

0100 Fair quality

0101 Intermediate quality

0110 Below intermediate quality

0111 Average quality

1000 Below average quality

1001 Questionable quality

1010 Above marginal quality

1011 Marginal quality

1100 Low quality

1101 No atmospheric correction performed

1110 Quality too low to be useful

1111 Not useful for other reasons

(equal to VI quality = 11: VI not produced due to bad quality)

Four-bit range

0= highest quality

3= acceptable quality

13= no atmospheric correction,

14= quality too low to be useful

15= not useful for any other reason

Unpacking bits 2-5 would create a SDS band with the values 0 to 15. This description already defines the decimal representation and not the binary value.

Note: The User Guide bit descriptions will vary in their definitions and can reflect binary values or decimal values.

5. `unpack_sds_bits -sds='500m 16 days NDVI Quality' -bit=0-1,2-5 -of=unpack_MOD13A1.A2001225.h11v05.003.2001342094517.hdf MOD13A1.A2001225.h11v05.003.2001342094517.hdf`

** Important to use correct SDS name and bit ranges.

6. 'unpack_MOD13A1.A2001225.h11v05.003.2001342094517.hdf' is the output HDF file and can be imported into software supporting HDF or HDF-EOS data formats. There will be two separate SDS names:
500m 16 days NDVI Quality_bits_0-1
500m 16 days NDVI Quality_bit_2-5

The SDS bands can be viewed and used for analysis such as computing statistics, masking or overlaying in conjunction with other SDS layers, i.e. the NDVI band.

Example of unpacking file: MOD43B3.A2001225.h11v05.003.2001342111741.hdf (that contains 3-dimensional SDS).

1. Extract the quality bits for the MODIS band 7 of the 1km 16-day Albedo product (MOD43b3).
2. Reference the MOD43B3 User Guide or File Specification listing for SDS names and bit information:
<http://geography.bu.edu/brdf/userguide/albedo.html>
3. Use 'unpack_sds_bits -help MOD43B3.A2001225.h11v05.003.2001342111741.hdf' to get exact list of SDS names.

Albedo_Quality (1200 x 1200 x 2) UINT32

** This is the albedo quality information and is an example of a 3-dimensional SDS. The '2' refers to the size of the 3rd dimension

Albedo (1200 x 1200 x 10 x 2) INT16

** This is an example of a 4-dimensional SDS. The '10' refers to the size of the 3rd

dimension and is the number of bands. The '2' refers to the size of the 4th dimension

and is the number of albedos (black-sky albedo and white-sky albedo for each band.

4. Referencing the User Guide, band quality is stored in the 2nd word or element of the 3rd dimension of SDS name = 'Albedo_Quality'. Within the 2nd word bits 24-27 are used for the band 7 quality information.

Bit Description for Word 2:

00-03 Band 1 Quality

- 0 = RMSE good, WoD(ref) good, WoD(WSA) good
- 1 = RMSE good, WoD(ref) good, WoD(WSA) moderate
- 2 = RMSE good, WoD(ref) moderate, WoD(WSA) good
- 3 = RMSE good, WoD(ref) moderate, WoD(WSA) moderate
- 4 = RMSE moderate, WoD(ref) good, WoD(WSA) good
- 5 = RMSE moderate, WoD(ref) good, WoD(WSA) moderate
- 6 = RMSE moderate, WoD(ref) moderate, WoD(WSA) good
- 7 = RMSE moderate, WoD(ref) moderate, WoD(WSA) moderate
- 8 = magnitude inversion (numobs >=7)
- 9 = magnitude inversion (numobs >3 & <=3)
- 11 = Bus-in DB parameters
- 15 = Fill value

24-27 Band 7 Quality (see explanation above)

5. Unpacking bits 24-27 would create a SDS band with digital values of 0-15.

6. `unpack_sds_bits -sds='Albedo_Quality.2' -bit=24-27`
`-of=unpack_MOD43B3.A2001225.h11v05.003.2001342111741.hdf`
`MOD43B3.A2001225.h11v05.003.2001342111741.hdf`

** The '.2' within -sds refers to the 2nd word or element of the 3rd dimension. The new SDS band within the output HDF file will have the digital values of 0-15.