

MODIS REPROJECTION TOOL USER'S MANUAL

**Release 4.1
April 2011**

*Land Processes DAAC
USGS Earth Resources Observation and Science (EROS) Center*

Table of Contents

Introduction	4
MODIS Reprojection Tool Capabilities.....	5
Platforms	5
Interfaces	5
Data Products	5
File Formats.....	5
Data Types.....	6
Map Projections.....	6
Resampling.....	6
Format Conversion.....	6
Mosaic Tool.....	7
Datum Conversions	7
Spectral subsetting.....	8
Spatial subsetting.....	8
Output Pixel Size.....	9
Parameter Files.....	9
Metadata	9
Background Fill.....	10
Corner Coordinates.....	10
Log File	10
Bounding Tiles	10
Bounding Tile Issue	11
Bounding Tile Solution.....	13
MODIS Reprojection Tool Installation.....	14
Pre-Installation	14
Automatic Installation.....	15
Adding a Shortcut Icon in Windows.....	16
Manual Installation.....	17
Manual Installation Instructions for UNIX Platforms.....	17
Manual Installation Instructions for Windows Platforms	20
Building the MRT	24
Command Line Interface.....	28
Parameter Files.....	28
Resample Tool.....	29
Mosaic Tool.....	33
Batch Processing Options.....	37

Windows Batch Processing.....	37
UNIX Bourne Shell Batch Processing.....	39
Direct Argument Batch Processing.....	39
Automated Batch Processing.....	40
MODIS Reprojection Tool GUI.....	42
Resampling Tool.....	42
Opening an Input File.....	43
Metadata Examination.....	45
Tile Locator.....	46
Spectral Subsetting.....	47
Spatial Subsetting.....	47
Specify Output File.....	48
Output File Type.....	49
Resampling Type.....	50
Output Projection Type.....	50
Output Pixel Size.....	52
Load or Save Parameter File.....	52
Executing resample.....	52
Exiting the GUI.....	53
Format Conversion.....	54
Mosaic Tool.....	54
Credits.....	56
Contacts.....	56
Appendix A: MRT Parameter File Format.....	57
File naming conventions.....	57
Editing parameter files.....	57
Parameter file format.....	57
Appendix B: MRT Raw Binary File Format.....	61
File naming conventions.....	61
Header file format.....	62
Editing header files.....	63
Notes.....	63
Appendix C: Projection Parameters.....	65
Projection Parameters 1-8.....	65
Projection Parameters 9-15.....	66
Notes.....	68

Introduction

The Moderate Resolution Imaging Spectroradiometer (MODIS) was launched into space aboard the National Aeronautics and Spaces Administration (NASA) Earth Observing System (EOS) platform Terra in December 1999. A second MODIS sensor was launched on the Aqua platform in May 2002. The objective of MODIS is to provide a comprehensive series of global observations of Earth's land, oceans, and atmosphere at higher spatial resolutions (250-meter (m), 500-m, 1,000-m) than its predecessor, the Advanced Very High Resolution Radiometer (AVHRR), and more frequently (near-daily overpass) than its orbital neighbor Landsat 7¹. MODIS observations are critical for studies of climate, vegetation, pollution, global change, and many other important economic and environmental issues.

The MODIS Reprojection Tool (MRT) was developed to support higher level MODIS Land products which are distributed as Hierarchical Data Format-Earth Observing System (HDF-EOS)² files projected to a tile-based Sinusoidal grid. MRT software facilitates the use of MODIS Land tiles by providing map projection, format conversion, and spectral and spatial subsetting options and is compiled for use on multiple operating systems.

MRT functionality is based on the `resample` and `mrtmosaic` executables that may be run either from the command line or from a Graphical User Interface (GUI). The GUI is an easy, user-friendly way to input data for manipulation, and the more powerful command line interface serves users with intensive data processing requirements. This User's Manual describes the use of both to run the MRT `resample` and `mrtmosaic` programs.

¹ More information about MODIS is available from <http://modis.gsfc.nasa.gov/>.

² More information about HDF and HDF-EOS is available from <http://www.hdfgroup.org/> and <http://hdfeos.net/> respectively.

MODIS Reprojection Tool Capabilities

Platforms

MRT is highly portable software available for four platforms, and has been tested on the following systems.

- Windows NT+ 32-bit
- Linux 32-bit
- Linux 64-bit
- Macintosh OS X 32-bit

Though they were not tested, MRT is expected to install and run on other systems (such as Windows Vista and Windows 7). Consult the Release Notes for platform-specific differences and caveats.

Interfaces

MRT may be invoked either from a user-friendly GUI or from a powerful command-line interface. The GUI allows novices and users with light processing requirements to reproject, convert, and subset MODIS tiled data. The GUI also allows easy inspection of metadata. The scriptable command-line interface, with its variety of command-line options, is likely to be the method of choice for reprojecting large numbers of files.

Data Products

MRT currently allows reprojection of all gridded (levels-2G, 3, and 4) MODIS Land data products. Support for swath (levels-1B and 2) data is available in the MRTSwath application³.

Most MODIS products are 2-Dimensional (2-D), but there are some 3-D and 4-D data sets (e.g., the MCD43 BRDF-Albedo suite⁴). The MRT supports 3-D and 4-D data products and currently outputs them to 2-D data products for raw binary, georeferenced tagged image file format (GeoTIFF), and HDF-EOS output formats.

File Formats

MRT accepts raw binary or tiled MODIS Land products in HDF-EOS format as input. Output file formats include raw binary, HDF-EOS, and GeoTIFF. The raw binary file format is specified in Appendix B.

³ MRTSwath is a no-cost download from https://lpdaac.usgs.gov/lpdaac/tools/modis_reprojection_tool_swath.

⁴ Summary information on an example of multi-dimensional MODIS data is available at https://lpdaac.usgs.gov/lpdaac/products/modis_products_table/brdf_albedo_model_parameters/16_day_13_global_500m/mcd43a1.

Data Types

MRT supports 8-bit, 16-bit, and 32-bit integer data (both signed and unsigned), as well as 32-bit float data. The output data type is always the same as the input data type.

Map Projections

MRT uses calls to the General Cartographic Transformation Package (GCTP)⁵ and as such allows projection to the following mapping grids.

- Albers Equal Area
- Equirectangular
- Geographic
- Hammer
- Integerized Sinusoidal
- Interrupted Goode Homolosine
- Lambert Azimuthal
- Lambert Conformal Conic
- Mercator
- Molleweide
- Polar Stereographic
- Sinusoidal
- Transverse Mercator
- Universal Transverse Mercator

The GCTP used by MRT applications has been modified to incorporate the Integerized Sinusoidal Projection originally applied to Version 001 MODIS products.

Resampling

Three resampling methods are available in MRT: nearest neighbor (NN), bilinear (BIL), and cubic convolution (CC).

Format Conversion

MRT provides an option to convert an input file to a different format without reprojecting. The possible input and output formats are described in File Formats above. The Format Converter will support spectral and spatial subsetting. When doing format conversion, the resampling process will be skipped. The output projection type and output projection parameters are not needed, and will be ignored if specified. In

⁵ Summary information on the GCTP software is available from <http://gcmd.nasa.gov/records/USGS-GCTP.html>.

format conversion the output projection type is the same as the input projection type, and the output projection parameters are the same as the input projection parameters. The output pixel size (if specified will be ignored) remains the same as the input pixel size, as does the output data type.

NOTE: A simple command-line tool (called `hdf2rb`) is used for format conversion from HDF to raw binary. It does not rely on geographic information and therefore works well with bounding tiles.

Mosaic Tool

MRT can mosaic several tiles together before reprojecting them. Mosaicking is done automatically from the GUI by selecting several filenames for the input filename. The input files are mosaicked first, and then reprojected. Mosaicking can also be done via the command line using the `mrtmosaic` executable.

Datum Conversions

A limited number of input and output datums are supported by the MRT for datum conversions. Supported datums are North American Datums of 1927 (`NAD27`), and 1983 (`NAD83`), and World Geodetic Systems of 1966 (`WGS66`), 1972 (`WGS72`), and 1984 (`WGS84`). User specification of the output datum is supported as part of the output parameters for the MRT in both the GUI and command-line interfaces. The GUI uses a datum drop-down box to allow the datum to be specified. This drop-down box has `NODATUM` set as the default, with options to select available datums. If the command-line version is being used, then the datum should be specified using the `DATUM` parameter in the parameter file, entering either any of the datums listed above or `NODATUM`. If no value is provided for the `DATUM` in the parameter file, `NODATUM` will be used as the default.

A datum is a standard definition of the semi-major and semi-minor axes of a shape. If the `NODATUM` option is selected, MRT will expect the user to enter spheroid information in the first two fields of the projection parameters for all the MRT-supported projections except UTM and Geographic. If the semi-major and semi-minor axes are not specified while the `NODATUM` is selected, MRT will exit with an error. Likewise, if semi-major and semi-minor axes are entered in the projection parameters and a specific datum is selected, the MRT will exit with an error.

Please note that the current GCTP package automatically uses the radius of Sphere 19 (6370997-m) for any sphere-based projections, except for Sinusoidal and Integerized Sinusoidal. Until this is fixed in GCTP, if the user needs to specify a radius other than that used for Sphere 19, the `NODATUM` option must be used. For Integerized Sinusoidal and Sinusoidal projections, users are allowed to specify their own sphere radius values.

Though a data product may be “referenced to” a datum, it is important that users understand sphere-based projections technically have no datum. Any sphere-based output will not contain any datum information. It will instead contain information pertaining to its spheroid. It will be up to the user to keep track of the datum to which the data is referenced. In addition, the GCTP/Geolib software prevents datum conversions when an initial datum is not known. Thus, if a product is output without a datum it can no longer be converted to another datum using the MRT software.

The datum value will be used as output to HDF-EOS, GeoTIFF, and raw binary data files. The datum will be specified in HDF for the HDF-EOS files, since HDF-EOS does not support datums. (HDF-EOS files are assumed to be referenced to WGS84 according to the HDF-EOS documentation.)

Once the MRT knows the input and output datums and has verified that the datum/projection parameter combinations are valid, the reprojection and datum conversion must be handled. The following are the steps that will be followed by the MRT to reproject the input `SIN` data to the specified output projection and datum.

1. Project the input data to the Geographic projection, using GCTP.
2. Convert from the input datum to the output datum, in the Geographic projection.
3. Project from the Geographic projection to the output projection.

Both steps 2 and 3 are handled by the same call to Geolib. If the input data is not in the `SIN` projection, then a single call to Geolib will handle the reprojection and datum conversion.

Spectral subsetting

HDF-EOS input files contain several layers of data, which are called Science Data Sets (SDS). The term “SDS” is used interchangeably in this document with the term “band.” Any subset of the input bands may be selected for reprojection. The default is to reproject all input bands.

Spatial subsetting

A spatial subset is defined by entering two corners (upper left and lower right) of a rectangle. These corners are specified either as input latitude and longitude coordinates, as input line/sample pairs, or as output projection coordinates. The default is to project the entire input image, using the bounding rectangular coordinates from the global attributes metadata.

Output Pixel Size

Because native MODIS spatial resolution is based on arcseconds at the equator, input pixel size is not likely to be exactly as advertised. For example, 250-m products actually contain 231.7-m pixels; 500-m products have 463.3-m pixels; 1,000-m products have 926.6-m pixels. MRT will default its output to the input pixel size unless otherwise specified. Output pixel size specification must be in meters unless projecting to a Geographic mapping grid, in which case decimal degrees must be used. The specified output pixel size will be used to process all selected bands through the GUI, but command line allows distinction between output pixel sizes per band.

Parameter Files

Whether invoked via the GUI or the command line, MRT executables are directed by a parameter file containing all the information necessary to read MODIS data from an input data file, transform the data into a specified output projection, and write the results to an output file. The parameter file contains the file names and file types of the input and output data files, spectral and spatial subsetting information, output projection type, output projection parameters, output UTM zone (if necessary), output resampling type, and output pixel size. Parameter files are generated automatically through the MRT GUI, and can be saved for later use either through the GUI or the command line interface. They are distinguished by a `.prm` extension, and formatted as ASCII text which may be created and edited in any text editor. Starting with a base parameter file from the GUI and modifying as necessary is recommended to avoid accidental processing errors should the user wish to construct a parameter file for command line use. The parameter file format is described fully in Appendix A.

Metadata

MRT extracts useful information from the input files and in GUI mode displays basic file specifications, including the number of bands available, data type, numbers of lines and samples, and the upper left and lower right corners.

Output file metadata is written for HDF-EOS input files only (not for raw binary inputs). The output HDF file contains the output metadata, followed by the original input file metadata. The input structure, core, and archive metadata information is stored under the HDF attributes `OldStructMetadata`, `OldCoreMetadata`, and `OldArchiveMetadata`, respectively.

Background Fill

If the majority of values that lie under the resampling kernel are background fill values, then a background fill value is output. Otherwise, resampling is performed only from non-background fill values, and kernel weights are adjusted accordingly. MRT reads the “_FillValue” for each input band and uses that value for the output background fill. If no _FillValue is specified, then the default value is 0.

NOTE: For some MODIS products, the fill value is very high (i.e. 65535) rather than a lower value or negative value as some users may be accustomed to. For these products, the non-image data in the resampled image will also be background fill. This will result in very bright pixels surrounding the actual image data, instead of the dark pixels that may be expected.

Corner Coordinates

The upper left (UL) corner specified for output in GeoTIFF refers to the center of the corner pixel. All other corners are expressed using the HDF standard, which is the outer extent of the UL and lower right (LR) corners. Both HDF-EOS and raw binary MRT output coordinates thus represent the UL corner of the corner pixel. Any output corner coordinates specified by the GUI, in the status box, or by the command line, in standard output or to the log file, represent the outer extent of the pixel.

Log File

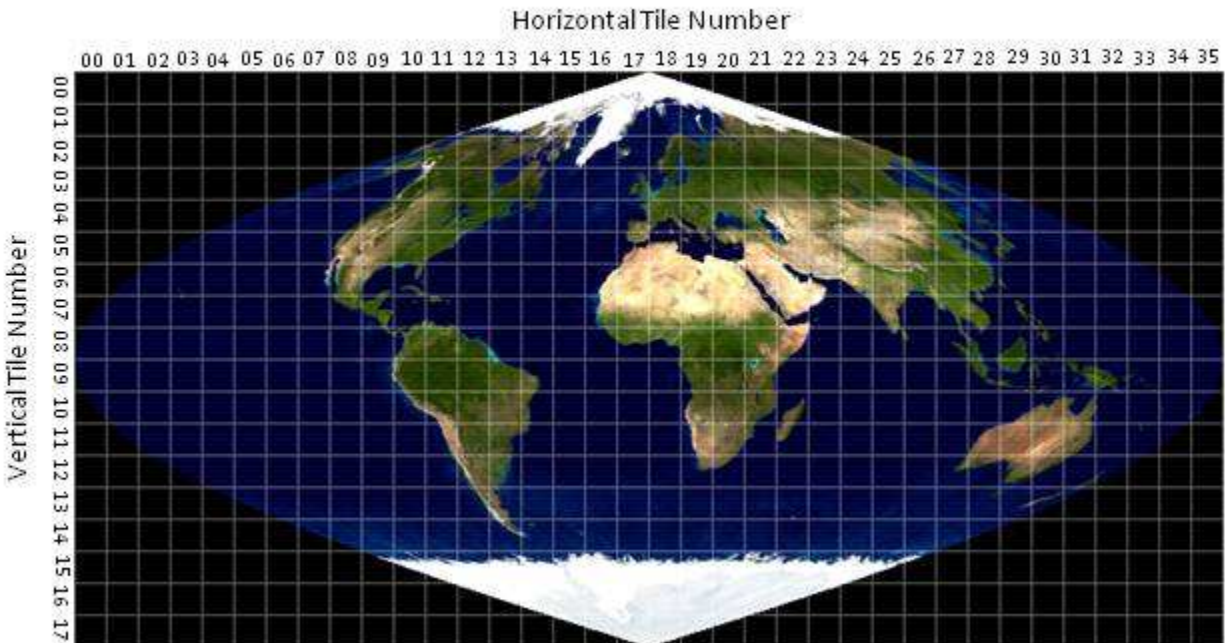
MRT writes logging and status information to a screen display and also to a log file. The log file is named `resample.log`, and is written into the `bin` directory (e.g., `C:\MRT\bin\resample.log`). Details of MRT activity are appended to the log at the completion of every run, so the history of every MRT execution is recorded ad infinitum. The log is an ASCII text file which users may edit or print with standard text file tools.

The command-line version of the resampler will allow the user to specify the path and name of the `resample.log` file, using the `-g` option. The options for the resampler are fully described later in the “Command-Line Interface” section.

Bounding Tiles

Bounding tiles have presented some difficulties for the MRT. These tiles occur on the outer edge of the Sinusoidal globe as seen in the gridded figure below.

MODIS Land Sinusoidal Mapping Grid



The tiles are labeled from top to bottom and from left to right starting at 00. The horizontal tiles range from h00 to h35, and the vertical tiles range from v00 to v17. The MODIS HDF-EOS filename will contain a field specifying the horizontal and vertical location of the tile. For example, a tile covering the state of Florida, USA would be named something like

MOD13Q1.A2011042.h10v06.005.20011060132568.hdf, where “h10v06” indicates its tile location on the Sinusoidal grid. Data over the southern tip of Africa might be named

MO09A1.A2007177.h19v12.005.2009081231246.hdf, where “h19v12” is its tile location.

Bounding Tile Issue

Bounding tiles are unique in that they contain corner points with projection coordinates which do not map to a valid latitude/longitude. For example, tile “h10v02” over Alaska theoretically has corner points along both the far western and far eastern edges of the Sinusoidal globe. Bounding tiles wrap around the edges of the Sinusoidal globe, which places coordinates in discontinuity space (i.e., in the black space in the figure above). This is not unique to MRT or MODIS tiles and impacts any global projection software.

This discussion will be limited to MRT handling of the MODIS tiles. The discontinuity space for the Sinusoidal projection (as with many projections) occurs at the -180.0 / 180.0 longitude line. Along this line, data do not cover all 10 x 10 degrees of a tile, as they do for interior tiles. The blank space is populated with fill data to complete the tile.

The projection coordinates of one to three corner points in a bounding tile fall in the discontinuity space. In this case, the MRT will process the tile as usual, but issue a warning to the user. It will cite that a certain corner point does not have a valid latitude/longitude, and will be adjusted accordingly. When MRT passes bounding tile projection coordinates into the transformation package, GCTP provides the output latitude and longitude values. Because of their proximity to the 180.0/-180.0 line, the returned longitude values wrap around it and fall on the other side of the Earth. When these GCTP coordinates are further passed to the forward mapping algorithm, the output projection coordinates will not match those of the input.

For example, the upper right corner of MODIS bounding tile h27v03 falls in discontinuity space and therefore its longitude will map to the other side of the 180.0/-180.0 line.

The Sinusoidal projection coordinates for this tile are as follows:

UL x,y in meters = 10007554.677000, 6671703.118000

LR x,y in meters = 11119505.196667, 5559752.598333

Upper right and lower left corners are derived by squaring off the tile:

UR x,y in meters = **11119505.196667**, 6671703.118000

LL x,y in meters = 10007554.677000, 5559752.598333

The following are the longitude/latitude corners provided by GCTP inverse mapping:

UL long,lat in degs: 179.999999954516 59.999999994612

UR long,lat in degs: **-160.00000050532** 59.999999994612

LL long,lat in degs: 140.015144391778 49.999999995507

LR long,lat in degs: 155.572382657536 49.999999995507

The UR longitude above (-160) has extended beyond the eastern hemisphere, past the 180 line, and 20 degrees into the western hemisphere. The associated forward mappings from the above latitude/longitude values likewise indicate the UR corner has wrapped around the 180 line into discontinuity space, as shown below.

UL x,y in meters = 10007554.677000002936 6671703.117999996990

UR x,y in meters = **-8895604.162390576676** 6671703.117999996990

LL x,y in meters = 10007554.677000001073 5559752.598332998343

LR x,y in meters = 11119505.196667000651 5559752.598332998343

The forward mapping coordinates above do not match the original “squared-off” upper right corner.

Bounding Tile Solution

The solution for handling the bounding tiles first detects that a latitude/longitude pair has wrapped around the Earth by running forward mapping on the inverse mapping to validate the latitude/longitude values. In the case that the forward mapping projection values are the same⁶ as the original projection corners, then the latitude/longitude values will be used directly. In the case that the forward mapping projection values are not the same, the software will constrain the longitude values at -179.900° (for the upper or lower left corners) or 179.900° (for the upper or lower right corners). Because the longitudinal values returned by the GCTP fall in the discontinuity space (wrap around “backwards”), MRT signs the longitude as opposite of the GCTP value. For example, a GCTP return of -180.000 will be written as +179.900 by MRT.

The longitudinal constraint was not set strictly to 180.0° or -180.0° to avoid the risk of running into further discontinuity issues when going to output space. Enforcing longitudinal bounds at 179.900° and -179.900° on the example over h27v03 used above, the latitude/longitude corners provided by the inverse mapping are changed as shown in the table below. The “Lat (fixed)” and “Lon (fixed)” values are finally converted to output space to define a minimum bounding box in which to project the input data.

Corner	Lat (pre-fix)	Lon (pre-fix)	Lat (fixed)	Lon (fixed)
UL	59.999999994612	179.999999954516	59.999999994612	179.999999954516
UR	59.999999994612	-160.000000050532	59.999999994612	179.900000000000
LL	49.999999995507	140.015144391778	49.999999995507	140.015144391778
LR	49.999999995507	155.572382657536	49.999999995507	155.572382657536

⁶ A tolerance value of 5-m was used to determine if the original projection coordinates and the forward mapping projection coordinates are the same value. This tolerance value should be sufficient for MODIS tiles, but if this software is used in the future for data from other higher resolution images then a more appropriate tolerance value will be needed for processing those products.

MODIS Reprojection Tool Installation

To obtain the MRT software, users are required to create an account on the LP DAAC Tools Web site⁷, a quick process that will enable notifications of software-related information (such as new releases). After successful login, users can download the appropriate platform-specific MRT zip file from the same site.

The MRT software is delivered in `MRT_download_<platform>.zip` that needs to be unzipped before installation. After the MRT package (`MRT_<platform>.zip`) is extracted from the download zip package, the software can be installed with either an automatic or a manual installation process. The automatic installation process is recommended, but instructions for both methods are outlined below.

Pre-Installation

Users no longer need to carefully place software or files in directories with no spaces in their path names, as required by previous MRT versions. As of version 4.1, MRT will install and run in any specified directory. This applies to both the MRT and Java directories. Further related to directory paths, it is also no longer necessary to use forward slashes (/) instead of backslashes (\) during MRT installation to Windows platforms. Backslashes are in fact now recommended when entering directory paths during installation.

Otherwise, there are a couple of pre-install checks that users are encouraged to address to ensure successful installation and execution of MRT software. First note that entry of full pathnames during installation is highly recommended for all directories. Wildcards (?,*) and relative pathnames are accepted, but the MRT may or may not set up correctly.

Second, in order to run the MRT GUI, installation of a current version of Java⁸ is required (at least the Java 2 Runtime Environment version 1.5 or the Java 2 SDK version 1.5 or later). If the only intent is to use the command line interface, Java is not necessary and this requirement can be ignored, but GUI operations will not execute without a current Java version.

⁷ MRT users may register as LP DAAC Tools Users by signing up at https://lpdaac.usgs.gov/lpdaac/tools/modis_reprojection_tool.

⁸ Java software for all platforms may be downloaded from the Oracle Web site at <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

The Java directory path is needed during the installation process, so have this information available during installation. On Windows platforms, Start/Search can be used to locate the Java directories on the system.

On UNIX-based systems such as Linux and Macintosh, type `which java` to determine the path to the existing Java executable. This method will work only if the Java executable is on the current path. Typing `find / -name java` will also work, but this will search all mounted file systems and may take time to complete. If necessary, ask a system administrator where the Java `bin` directory is located.

If the Java executable is on the current path, the version of Java installed on the machine can be determined by typing `java -version`, which will output something similar to this:

```
java version "1.5.0"
```

If the version number is older than 1.5.0, the MRT GUI is not guaranteed to work. In this case, installing a newer Java version is recommended.

Automatic Installation

1. Step one for automatically installing the MRT to either Windows or UNIX systems (Linux or Macintosh) is unzipping the `MRT_download_<platform>.zip` file.
2. Step two is to collect the following information to facilitate successful and efficient installation.
 - The complete pathname of the directory in which MRT will be installed (full pathnames are recommended over wildcards). The default will create a `MRT` subdirectory in the current directory, but users have the option to enter a different directory path if preferred.
 - The pathname of the directory containing the Java executable program (`java` or `java.exe`). This pathname is not necessary on UNIX platforms if the Java executable is on the current path, in which case it will be found automatically. See the information at the start of the Installation section for hints on how to determine the location of the Java `bin` directory.
3. Step three is using a command line interface to navigate to the directory to which `MRT_download_<platform>.zip` was unzipped. On Windows systems, it will contain the four files `mrt_install.bat`, `MRT_<platform>.zip`, `reg_set.exe`, and `unzip.exe`. On UNIX systems only two files, `mrt_install` and `MRT_<platform>.zip`, will be present.

4. Step four is to execute the install scripts while within the directory containing the installation files (see previous step).

- For UNIX (Linux or Macintosh), type `./mrt_install`, such as:

```
[user@server /MRT]$ ./mrt_install
```

- For Windows, type `mrt_install`, from the Command Prompt screen invoked from Start\Programs\Accessories.

```
C:\Program Files\MRT> mrt_install
```

5. Step five is to carefully follow the instructions appearing in response to the install command. The program will prompt the user to enter a preferred location for the install, and the information needed to set up path and environment variables correctly. Providing full pathnames for all directories is advised because although wildcards (`?`,`*`) and relative pathnames will be accepted, they are likely to keep MRT software from setting up correctly. Use copy/paste functions if possible when entering directory paths. Typographical errors will not be recognized during the installation process, and could result in display or execution errors.

6. Step six is a recommended system or session restart to ensure the changes made during MRT installation take effect.

Adding a Shortcut Icon in Windows

MRT has an icon file in the Windows distribution package. It is located in the MRT `bin` directory. A shortcut icon for the MRT can be created and used on the Windows platforms. To create a shortcut icon follow these steps:

1. Click the right mouse button on the desktop
2. Select `New` → `Shortcut`
3. Type in the path and name for the `ModisTool.bat` file (or use the `Browse` button to find the `ModisTool.bat` file, which should be in the MRT `bin` directory, e.g., `C:\Program Files\MRT41\MRT\bin\ModisTool.bat`)
4. Select `Next`
5. Type a name for the shortcut (i.e. `ModisTool`, `MRT`, etc.)
6. Select `Finish`

7. Right click on the shortcut icon that was just created
8. Select Properties
9. Select the Shortcut tab
10. Press Change Icon
11. Click OK to choose an icon from the list or specify a different file
12. Type in the path and name for the ModisTool.ICO file, or use the Browse button to find the ModisTool.ICO file, which should be in the MRT bin directory
13. Press OK, then OK

The MRT icon should now be available on the desktop. When the icon is double-clicked, the MRT GUI should appear.

Manual Installation

Three steps are required for manual installation of the MRT:

- Unzip the software.
- Update the system variables to include PATH, MRT_HOME, and MRT_DATA_DIR information for MRT.
- Edit the GUI script in the MRT bin directory to reflect the system directory structure.

Below are instructions for manually working through these steps. The first section is relevant to UNIX systems and includes specifics for C shell and Bourne shell. The second section is relevant to Windows systems and includes specifics for NT/ME/XP.

Manual Installation Instructions for UNIX Platforms

1. Step one for manual installation of the MRT to a UNIX-based system such as Linux or Macintosh is a two-part extraction process to load the necessary files to the user's system. First, unzip the MRT_download_<platform>.zip file downloaded from the LP DAAC Tools Web site. Use the command unzip or ./unzip from a UNIX terminal to inflate the download package to a chosen location.

```
unzip zipfile -d "MRTdownload_directory"
```

where zipfile is MRT_download_Linux.zip (32-bit compilation), MRT_download_Linux64.zip (64-bit compilation), or MRT_download_Mactel.zip (Macintosh

compilation), and `MRTdownload_directory` is the directory to which the zip contents will be extracted.

If an error is returned saying “`unzip: command not found`”, try using a preceding “`./`” as shown below.

```
./unzip MRT_download_Linux.zip -d "$HOME/MRT"
```

Changing to the `MRTdownload_directory` in the UNIX terminal, the file `mrt_install` should be present along with either `MRT_Linux.zip`, `MRT_Linux64.zip`, or `MRT_Mactel.zip`.

The second part of the extraction process will inflate the MRT software package. Use the `unzip` command to extract the installation software to a chosen location.

```
unzip zipfile -d "MRT_directory"
```

where `zipfile` is `MRT_Linux.zip`, `MRT_Linux64.zip`, or `MRT_Mactel.zip`, and `MRT_directory` is the directory into which MRT will be installed. For example, to install in a “`My MRT`” subdirectory in the home directory on a Linux workstation, type the following.

```
unzip MRT_Linux.zip -d "$HOME/My MRT"
```

2. Step two will add MRT information to the startup files for the current shell (`csh`, `tcsh`, `sh`, `bash`). Typing `echo $0` or `echo $SHELL` at the command prompt will display which shell is being used.

C shell (`csh`, `tcsh`) users will need to add or update the `path`, `MRT_HOME`, and `MRT_DATA_DIR` environment variables to the system `.login` file in the home directory. Assuming installation of MRT in a `My MRT` subdirectory in the home directory, the following lines are appended to the `.login`.

```
set path = ( $path:q "$HOME/My MRT/bin" )
```

```
setenv MRT_DATA_DIR "$HOME/My MRT/data"
```

```
setenv MRT_HOME "$HOME/My MRT"
```

Bourne shell (`sh`, `bash`) users will likewise add or update the `path`, `MRT_HOME`, and `MRT_DATA_DIR` environment variables to the system `.profile` and `.bash_profile` file in the home directory, in addition to setting the path to the Java executable. The following lines are appended to `.profile` or `.bash_profile` files in the home directory.

```
#!/bin/sh

PATH="$PATH:$HOME/My MRT/bin"

MRT_DATA_DIR="$HOME/My MRT/data"

MRT_HOME="$HOME/MRT"

export MRT_HOME PATH MRT_DATA_DIR
```

Note that the final statement starting with "<path_to_java_bin>/java..." is all on one line. Do not break this statement into multiple lines.

These changes will not take effect without restarting a new login session. Also be aware that each new installation will append the environment variable statements to the startup files, so it is advisable to remove these lines prior to a new install.

3. In step 3, the `ModisTool` shell script used to run the MRT GUI is created manually. This script will configure the variables `PATH`, `MRT_HOME`, and `MRT_DATA_DIR`. The `ModisTool` script can be generated in any text editor and should be saved to the `MRT bin` directory. Below is an example that can be copied as a template for creating this file using any text editor. It assumes the directory structure used in the examples above.

Make sure to replace all directory information in the template to reflect the structure of the host machine.

Note that the final statement indicating the Java path is all on one line. Do not break this statement into multiple lines.

Save the file in the `bin` (e.g., `/home/user/My MRT/bin`) as `ModisTool.bat`.

A system restart is advised to ensure the changes made during MRT installation take effect and the tools function as expected.

C shell (csh, tcsh) `ModisTool` template:

```
#!/bin/sh

# *****
# * ModisTool *
# * C Shell script for running the MRT GUI *
# *****
```

```

# Set the MRT_HOME environment variable to the MRT installation
directory.

setenv MRT_HOME "$HOME/My MRT"

# Set the MRT_DATA_DIR environment variable to the data directory.

setenv MRT_DATA_DIR "$HOME/My MRT/data"

# Set the PATH environment variable to include MRT executables.

set path = ( $path:q "$HOME/My MRT/bin" )

# Run the MRT Java GUI.

"<path_to_java_bin>/java" -jar "$HOME/My MRT/bin/ModisTool.jar"

```

Bourne shell (sh, bash) ModisTool template:

```

#!/bin/sh

# *****
# * ModisTool *
# * Bourne Shell script for running the MRT GUI *
# *****

# Set the MRT_HOME environment variable to the MRT installation
directory.

MRT_HOME="$HOME/My MRT"
export MRT_HOME

# Set the MRT_DATA_DIR environment variable to the data directory.

MRT_DATA_DIR="$HOME/My MRT/data"
export MRT_DATA_DIR

# Set the PATH environment variable to include MRT executables.

PATH="$HOME/My MRT/bin:$PATH"
export PATH

# Run the MRT Java GUI.

"<path_to_java_bin>/java" -jar "$HOME/My MRT/bin/ModisTool.jar"

```

Manual Installation Instructions for Windows Platforms

On Windows platforms, note that it is no longer necessary to enter directory pathnames with forward slashes (/) instead of backslashes (\). It is in fact recommended that backslashes are used during Windows installation. Neither is it necessary to avoid directories with blank spaces for the MRT install directory, the Java directory, or the input file directory.

1. Step one for manual Windows installation is unzipping the MRT software, which is a two-part process. First unzip the MRT software package downloaded from the LP DAAC Tools Web site (MRT_download_Win.zip). This can be done automatically common tools like WinZip. For example, right click on the zip, select WinZip then Extract to, and create a directory for the enclosed files (such as C:\Program Files\MRT). The MRT_download_Win.zip will inflate to install.bat, MRT_Win.zip, reg_set.exe, and unzip.exe.

Second, the unzip.exe will be used to extract the files needed for installation from C:\Program Files\MRT\MRT_Win.zip. From Start\Programs\Accessories, open a Command Prompt screen, change to the directory containing the files just extracted (C:\Program Files\MRT), and type the following command.

```
unzip MRT_Win.zip -d "MRT_directory"
```

where MRT_directory is the directory in which the MRT will be finally installed. For example, to install the MRT in C:\Program Files\MRT\Tool, type the command as follows.

```
unzip MRT_Win.zip -d "c:\Program Files\MRT\Tool"
```

2. Step two is to update the system variables to recognize the PATH, MRT_HOME, and MRT_DATA_DIR needed to run MRT.

Windows NT/XP users must edit their user keys to add the MRT PATH, MRT_HOME, and MRT_DATA_DIR to the system variables. Invoke a Command Prompt window from Start\Programs\Accessories and type regedit at the DOS prompt. Administrative privileges may be needed to complete this step.

This will bring up a GUI containing user key information. Double-click on the HKEY_CURRENT_USER key then click on the Environment key. The current user key environment variables and their values will be listed on the right hand side of the GUI. If the keys MRT_DATA_DIR, MRT_HOME, and PATH do not exist, go to the Registry Editor menu bar at the top of the GUI and click Edit. From there, select New then String Value to create a new user key.

The keys for PATH, MRT_HOME, and MRT_DATA_DIR should be listed in the right pane of the Registry Editor. Each can be double-clicked to modify their values. Continuing with the above example, the PATH would be modified to add c:\Program Files\MRT\Tool\bin to the end of the current string, entering a semi-colon to separate directories in the path. The MRT_HOME would be set

to `c:\Program Files\MRT\Tool`, and `MRT_DATA_DIR` would be set to `c:\Program Files\MRT\Tool\data`. It is recommended that any previous values specified for these keys are removed. Close the Registry Editor after the keys have been updated.

3. In step 3, the `ModisTool.bat` script used to run the MRT GUI is created manually. This script will configure the variables `PATH`, `MRT_HOME`, and `MRT_DATA_DIR` for the GUI. The `.bat` can be generated in any text editor (such as Notepad), and should be saved to the MRT `bin` directory. Below is an example that can be copied as a template for creating this file. It assumes the directory structure used in the examples above.

Make sure to replace all directory information in the template to reflect the structure of the host machine.

Note that the final statement starting with "`c:\Program Files\Java...`" is all on one line. Do **not** break this statement into multiple lines.

Save the file in the `bin` (e.g., `c:\Program Files\MRT\Tool\bin`) as `ModisTool.bat`.

A system restart is advised to ensure the changes made during MRT installation take effect and the tools function as expected.

```
@echo off

rem *****
rem * ModisTool.bat *
rem *****

rem Set the MRT_HOME environment variable to the MRT installation
rem directory.

set MRT_HOME=C:\Program Files\MRT\Tool

rem Set the MRT_DATA_DIR environment variable to the data directory.

set MRT_DATA_DIR=C:\Program Files\MRT\Tool\data

rem Set the PATH environment variable to include the MRT executables.

set Path=C:\Program Files\MRT\Tool\bin

rem Run the Java GUI.
rem Change the java.exe path to reflect the directory structure on
rem host.
rem Quotes are only necessary to handle blank spaces in the pathnames.
ECHO is off.
"c:\Program Files\Java\jre1.6.0_05\bin\java.exe" -jar "C:\Program
Files\MRT\Tool\bin\ModisTool.jar"
```

4. To add an icon to the desktop, please see the direction following the Automatic Installation section above.

Building the MRT

The MRT application is buildable across other operating systems, architectures, and different compilers than those mentioned in this manual. Those described in this manual are those available at USGS EROS. Users may wish to attempt compilation on other systems, or make code changes to currently supported platform packages. The descriptions in this section refer to a variable called `MRT_HOME`, which should point to top level installation directory of the MRT application. There is additional information that can be found in the README file.

1. Step 1 (Build third-party libraries): Supported systems are described briefly under the “Platforms” subsection of “MODIS Reprojection Tool Capabilities” above. If the users system is currently supported, Step 1 can be skipped.

If there is a need to build MRT applications on a currently unsupported system, third-party libraries will need to be built first. Please note that the following instruction assumes the libraries will be built statically. Use of shared or dynamic libraries is not addressed here. Also, if some of the libraries needed for MRT are already installed, the user will need to determine whether to use those existing or to build them for MRT application, making sure MRT pointers are directed to the appropriate library. The libraries used by the MRT application are listed below (with the version numbers used for this release) with links that were valid as of this writing.

Projects	Version	Libraries	Link
zlib	1.2.3	libz.a	http://www.zlib.net/
gzip	2.0	libsz.a	http://www.hdfgroup.org/release4/obtain.html
jpeg	6b	libjpeg.a	http://www.hdfgroup.org/release4/obtain.html
tiff	3.8.2	libtiff.a	http://remotesensing.org/libtiff/
Geotiff	1.2.3	libgeotiff.a	http://trac.osgeo.org/geotiff/
hdf4	4.2r1	libdf.a libmfhdf.a	http://www.hdfgroup.org/products/hdf4/ (use the one in the src/patches directory.)
hdf-eos	2.14	Libhdfeos.a	http://www.hdfeos.org/software/

When compiling the `tiff` and `hdf4` libraries, make sure to include their respective dependencies to the `gzip`, `jpeg`, and `zlib` libraries. The `Geotiff` library depends on the `tiff` library and the `hdf-`

`eos` library depends on the `hdf4` library. If compiling on Windows with a native compiler, then the above library names may not have a “.lib” extension instead of a “lib” prefix (e.g. `libz.a` would be called `z.lib`).

Once the third-party libraries are built, copying them into the `$(MRT_HOME)/lib` directory will make modifying the `Makefile` easier. Likewise, copying all the third-party `include` files into the `$(MRT_HOME)/include` directory will ease `Makefile` changes.

2. Step 2 (Modifying the `Makefiles`): This step may be skipped if building on a supported platform. Otherwise, note that MRT actually consists of three libraries and many programs, each having their own `Makefile`. There is a top level `Makefile` as well, that merely iterates through the library and program directories that it then builds using their respective `Makefiles`. Some systems cannot read `Makefiles` and use their own proprietary method of building (Visual Studio for C/C++ on Windows for example), and it would probably be best to set up the project manually through the IDE. The MRT directory structure can be shown as follows:

```
MRT
  \
  |-- shared_src      (library)
  |-- gctp            (library)
  |-- geolib          (library)
  |-- append_meta     (program)
  |-- dumpmeta        (program)
  |-- hdf2rb          (program)
  |-- hdflist         (program)
  |-- mrtmosaic       (program)
  |-- resample        (program)
  |-- sdslist         (program)
  |-- update_tile_meta (program)
```

Examine each `Makefile` to determine what changes to certain flags will be needed to tailor the build for a specific system. All programs are written in C, so some of the flags are centered on what the C compiler, linker, and archive are expecting. Other flags indicate commands to copy, remove, and move files. The following lists the compile flags used by the `Makefiles`.

Flag	Description
CC	The CC flag indicates the compiler to be used. It could be set to gcc, cc, or xlc, for example, depending on system particularities.
CFLAGS	The CFLAGS flag indicates any switches that need to be passed on to the C compiler. For GNU C, “-O3 -Wall -W -Wno-switch” is used for example. Users need to figure out what switches apply to their C compiler.
LDFLAGS	The LDFLAGS flag indicates the libraries and switches to be passed on to the linker. One switch that is used for supported platforms to pass to the GNU linker is the “-s” switch, which strips the executable of any debug information. The libraries needed depends on the application being built.
AR	The AR flag indicates the archive command that needs to be run to create the static or dynamic library. The current setting is set at “ar rcsv”.
CP	The CP flag indicates the command to copy files and the current setting is set to “cp”.
RM	The RM flag indicates the command to remove a file and the current setting is set to “rm -f”.
MV	The MV flag indicates the command to move a file and the current setting is set to “mv”.

3. Step 3 (Building the applications): Once the `Makefiles` are set up, there are a couple ways to build the applications. The easiest way is to go to the top level (`$MRT_HOME`) directory and then issue the following commands.

- `make clean` (optional)
- `make`
- `make install`
- `make clean` (optional)

The `make clean` command removes any object files and other miscellaneous temporary files. The `make` command builds the executables. The `make install` command moves all the executables to the `$MRT_HOME/bin` directory.

Otherwise each of the above commands needs to be issued separately in every subfolder, which includes all the library and program directories. These include the `shared_src`, `gctcp`, `geolib`, `append_meta`, `dumpmeta`, `hdf2rb`, `hdflist`, `mrtmosaic`, `resample`, `sdslist`, and `update_tile_meta`. This may be useful if only one or two programs are being worked on.

4. Step 4 (Build `ModisTool`): The `ModisTool` is written in Java. Under the `JavaGuiSrc` directory is a small script that builds the `ModisTool.jar` file and places it into the `$MRT_HOME/bin` directory.

The script is called `compilejava` and it works under Cygwin or a shell, although it should be fairly easy to modify to run under DOS.

As of version 4.1, `ModisTool` is built with Java version 1.5.0, with scripts targeting the same. The software should build under later releases of Java as well.

Command Line Interface

MRT functionality is based on two primary executables, `resample` and `mrtmosaic`. Both can be run by entering parameter information either through a GUI or a command line interface. The GUI is a simple and user-friendly guide for input information, but is not recommended when processing requirements include large number of files and actions. The MRT command line interface is a useful tool for intensive data conversions, and includes the flexibility of automation. Usage of `resample` and `mrtmosaic` are described separately below, following a description of the parameter file MRT needs as input to any processing.

Parameter Files

Regardless of how it is invoked (via the GUI or the command line) the MRT executables are directed by a parameter file containing all the information necessary to read MODIS data from an input data file, transform the data into a specified output projection, and write the results to an output file. The parameter file contains the file names and file types of the input and output data files, spectral and spatial subsetting information, output projection type, output projection parameters, output UTM zone (if necessary), output resampling type, and output pixel size. Parameter files are generated automatically through the MRT GUI, and can be saved for later use either through the GUI or the command line interface. They are distinguished by a `.prm` extension, and formatted as ASCII text which may be created and edited in any text editor. Starting with a base parameter file from the GUI and modifying as necessary can help avoid accidental processing errors should the user wish to construct a parameter file for command line use. The parameter file format is described fully in Appendix A.

Below is an example of a parameter file. Double quotes may or may not be necessary, depending on how the host system handles spaces in directory paths or file names.

```
INPUT_FILENAME = "D:\My
MRT\Inputs\MYD09GA.A2007113.h24v02.005.2007116092027.hdf"

SPECTRAL_SUBSET = ( 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 )

SPATIAL_SUBSET_TYPE = INPUT_LAT_LONG

SPATIAL_SUBSET_UL_CORNER = ( 65.0 127.0 )
SPATIAL_SUBSET_LR_CORNER = ( 61.0 154.0 )

OUTPUT_FILENAME = "D:\My MRT\Output\MOD09GA.tif"

RESAMPLING_TYPE = BILINEAR
```

```

OUTPUT_PROJECTION_TYPE = GEO

OUTPUT_PROJECTION_PARAMETERS = (
  0.0 0.0 0.0
  0.0 0.0 0.0
  0.0 0.0 0.0
  0.0 0.0 0.0
  0.0 0.0 0.0 )

DATUM = NoDatum

PIXEL_SIZE = 500

```

The sample parameter file shown above specifies that the input data file is MYD09GA.A2007113.h24v02.005.2007116092027.hdf. Bands 11, 13, and 14 (surf_refl_b01, surf_refl_b03, surf_refl_b04) will be projected to Geographic using bilinear interpolation resampling. The output pixel size for all bands is set to 500 to overcome potential inconsistency between the native 463.3-m MODIS pixels and other 500-m data sets. The output will be written to the GeoTIFF file MOD09GA.tif. Since the UL and LR corner points were not specified, the bounding coordinates from the input file's global attributes will be used for the output corner points.

Resample Tool

From the command prompt, typing "resample," "resample -help" or "resample -help=param" in the MRT bin directory results in a brief usage message that lists various command line arguments used to execute MRT. These are the fields typically included in a parameter file, but via command line, any or all of them may be modified. Detailed descriptions of each parameter field follow the usage message below.

```

MODIS Reprojection Tool v4.1 March 2009

Usage: RESAMPLE -p parameter_file [options]

Options that override parameter file specifications:

  -i input_file_name
  -o output_file_name
  -r resampling_type [NN BI CC NONE]
  -t projection_type [AEA ER GEO HAM IGH ISIN LA LCC MERCAT MOL PS SIN TM UTM]
  -j projection_parameter_list ["p1 p2 ... p15"]
  -s spectral_subset ["b1 b2 ... bN"] If using the -s switch, the SDSs
    should be represented as an array of 0s and 1s. A '1' specifies to
    process that SDS; '0' specifies to skip that SDS. Unspecified SDSs will

```

not be processed. If the `-s` switch is not specified, then all SDSs will be processed.

`-a spatial_subset_type [INPUT_LAT_LONG INPUT_LINE_SAMPLE OUTPUT_PROJ_COORDS]`

`-l spatial_subset ["ULlat ULlong LRlat LRlong"]`

`-or- ["ULline ULsample LRline LRsample <0 based>"]`

`-or- ["ULprojx ULprojy LRprojx LRprojy"]`

Line/Sample must be specified for the highest resolution of all SDSs specified to be processed in the product.

`-u UTM_zone`

`-x pixel_size`

`-g filename for the log file`

Usage: RESAMPLE -h file.hdf

creates raw binary header file TmpHdr.hdr

If using an input parameter file (recommended) the only required command line argument is the parameter file⁹ name. To run the MRT `resample` from the command line, type:

```
resample -p <parameter_file>
```

If a parameter file is not used, or if only certain fields require changes, command line arguments will set the parameters directly and override parameter file specifications. MRT is sensitive to the lower case font used for command line arguments.

To override any parameter file option, or to run `resample` without a parameter file, the following arguments may be typed into the MRT command line interface.

`-i input_filename` Input MODIS level-2G, -3, or -4 HDF-EOS file.

`-o output_filename` Output filename, including extension (`.hdr`, `.hdf`, or `.tif`) to set the output file type. Output filenames follow a convention to identify the input product and bands used in the operation.

`-o namegame.tif`

Output Filenames:

`namegame.surf_refl_b01.tif`

⁹ See Appendix A for more information about the parameter file.

namegame.surf_refl_b02.tif

namegame.250m_quality.tif

For multi-dimensional products, 3rd and 4th dimension slices are names as follows:

<SDS name>.<3rd dimension name>_#.<4th dimension name>_#

where # is a two-digit value representing the data slice for the associated dimension. Obviously if the product only has three dimensions, the <4th dimension name>_# will not appear. This naming convention is used for the output filenames, the output HDF-EOS band names, and the band names in the GUI.

The 3-D and 4-D naming convention produces long names when the band name, 3rd dimension name, and 4th dimension name are all of substantial length themselves. Currently the HDF-EOS library for the Windows platform will support only 57 characters for band names, which produces problems with the MRT naming convention on the Windows platforms. For the Windows platforms, if the output band name is going to be longer than 57 characters, the naming convention is then changed as follows to produce smaller output band names.

<SDS name>.3_#.4_#

- r `resampling_type` Resampling kernel type (CUBIC_CONVOLUTION, NEAREST_NEIGHBOR, or BILINEAR). The default is NEAREST_NEIGHBOR.
- t `output_projection_type` Output projection short name. Valid values are AEA (Albers Equal Area), ER (Equirectangular), GEO (Geographic), IGH (Interrupted Goode Homolosine), HAM (Hammer), ISIN (Integerized Sinusoidal), LA (Lambert Azimuthal Equal Area), LCC (Lambert Conformal Conic), MERCAT (Mercator), MOL (Molleweide), PS (Polar Stereographic), SIN (Sinusoidal), TM (Transverse Mercator), and UTM (Universal Transverse Mercator).
- j `projection_parameter_list` Output projection parameters¹⁰. This quoted, floating-point list includes up to 15 projection parameters, with each value separated by white space (“p1 p2 ... p15”). If there are fewer than 15 values specified in the list, the remaining values will be set to zero. Integer values will automatically be converted to floating point.

¹⁰ See Appendix C for more information about projection parameters, or type “resample -help=<projection>” at the command line.

-s `spectral_subset_list` Name of band (Science Data Set, or SDS) to resample. This is a quoted, binary list with each value separated by white space (“1 0 1 0...1 0”). “0” means “do not use this band,” while “1” means “select this band.” If there are fewer values in the list than input bands, the remaining bands will not be selected (i.e., will default to 0). Use the index number (0-based) after the band name to identify a 2D slice of a 3D band.

-a `spatial_subset_type` Type of output spatial subset value. Valid options are `INPUT_LAT_LONG` (latitude and longitude in decimal degrees), `INPUT_LINE_SAMPLE` (input line and sample), and `OUTPUT_PROJ_COORDS` (output projection x and y). The default is `LAT_LONG`, and the value specified for this field must match the type of values specified in the `spatial_subset_list`.

-l `spatial_subset_list` [note: lower case L] Output UL and LR corner coordinates. The entries in this quoted, floating-point list must correspond with the `spatial_subset_type` field. For example, `spatial_subset_type=INPUT_LAT_LONG`, then `spatial_subset_list` should consist of the UL corner latitude, UL corner longitude, LR corner latitude, LR corner longitude. To input coordinates by line and sample, a quoted integer list consisting of the start (UL corner) line, start (UL corner) sample, end (LR corner) line, and end (LR corner) sample may be used. The third option is to specify the output projection coordinates of the UL and LR corners. Thus it may be a quoted floating-point list consisting of the UL projection x, UL projection y, LR projection x, and LR projection y values. Values must appear in the specified order, separated by white space. Latitude/longitude and projection x/y values should represent the outer extent of the UL and LR corners of the subset.

-x `pixel_size` Output pixel size. This may be specified for each band processed, in output projection units (meters for all projections except Geographic which requires decimal degree units). For example, a user who selected five bands for processing and wishes to output one of the bands at a different resolution would enter the following command.

```
-opsz=1000,1200,1000,1000,1000
```

Otherwise, the default is to output to the same resolution as the input for each band, which generally is the same for each band. This field is optional, but given the native pixel resolution of MODIS swath data, it may be used to normalize the output pixels (e.g., change 926.6-m input to 1,000-m output).

If the number of output pixel sizes entered in this field is less than the number of bands processed, then the last pixel size entered will be used as the pixel size all bands. For example, it is not

necessary to write out the output resolution for all bands if the same output resolution is desired for all bands. Simply entering `-opsz=1000` will apply the desired output pixel size to all bands selected for processing.

- `-u UTM_zone` Output zone number, relevant only for UTM projections. Valid values are `-60` to `+60`.
- `-g log_filename` This option allows the user to specify the path and name of the log file. The default log filename is `resample.log`.
- `-h HDF_filename` This creates a raw binary header file from the input HDF-EOS file. The name of the output header file is `TmpHdr.hdr`.
- `-f` This switch causes `resample` to implement format conversion. The input file format is converted to the specified output file format (based on the input and output filename extensions). The format converter will support spectral and spatial subsetting, but it will not execute any resampling. It is necessary to include values for projection information in the parameter file in order for the reformatter to run, but all resampling-related inputs will be ignored during format conversion. The output projection type, output projection parameters, output pixel sizes, and output data types will be the same as the input.

Mosaic Tool

MRT provides a mosaic tool (`mrtmosaic`) for mosaicking tiles together prior to resampling. The mosaic tool requires that all input files are of the same product type and they must contain the same band names, band dimensions (number of lines and samples), band projection types and projection information, band pixel size, etc. If the band characteristics for each input tile do not match, then the mosaic tool will exit with an error.

The mosaic tool requires an input parameter file which lists the full path and filename of each input file to be mosaicked. The input files can be listed in any order and the mosaic tool will determine how they fit together in the mosaic. The mosaic tool also requires an output filename. The file type of the output file must match that of the input files. Thus, if the input files are HDF-EOS then the output file extension must be `.hdf`.

Typing `mrtmosaic` at the command prompt will display the following usage information. The last statement in the usage message will be an alert that no processing parameters were included in the command (`mrtmosaic`).

```
Usage: mrtmosaic -i input_filenames_file -t -h -o output_filename
```

```
-s spectral_subset ["b1 b2 ... bN"]  
-g filename for the log file
```

where `input_filenames_file` is a text file which contains the names of the files to be mosaicked.

If using the `-s` switch, the SDSs should be represented as an array of 0s and 1s. A '1' specifies to process that SDS; '0' specifies to skip that SDS. Unspecified SDSs will not be processed.

If `-t` is specified then the tile locations of the input

filenames are output to `tile.txt` (`-o`, `-s`, and `-h` are not needed).

Raw binary files must specify the tile locations in the filename to be used with the `-t` switch (i.e `mod09ghk_h02v16.hdr`).

If `-h` is specified then the mosaicked header information will be output to `TmpHdr.hdr` (`-o`, `-s`, and `-t` are not needed).

NOTE: Only input Sinusoidal and Integerized Sinusoidal projections are supported for mosaicking.

Example: `mrtmosaic -I TmpMosaic.prm -s "1 1 0 1" -o mosaic.hdf`

This will mosaic the first, second, and fourth SDSs in each of the specified HDF files in `TmpMosaic.prm`

Warning: mosaic : General Processing

: Error processing the arguments for the mosaic tool

To simply mosaic multiple tiles (such as prior to resampling), create a parameter file listing the input filenames. Then type the following command all on one line. Note that the output file must be in `.hdf` format.

```
mrtmosaic -i </directory/input_file_name_list.prm> -o  
</directory/output_mosaic.hdf>
```

To run the `resample` executable on the mosaic (such as to reproject, reformat, or subset), create a standard parameter file, using `output_mosaic.hdf` as the input file.

Spectral subsetting is allowed in `mrtmosaic` by using the `-s` command line switch. The `-h` switch will provide the user with header information for output mosaic. The `-t` switch will output the `h##v##` tile information read from the horizontal and vertical tile information embedded in the HDF-EOS metadata. For raw binary files, the tile information must be specified in the filename itself. Thus if the user wants to use the `-t` switch with mosaic tool, the input filenames must contain “`_h##v##`” somewhere in the filename. The mosaic tool will exit with an error if the tile information is not available.

The status information from the mosaic tool will specify the order of the input files in the output mosaic. In the following example, `file[0]` refers to the first file listed in the parameter file, `file[1]` refers to the second file, and so on. When the input files do not provide a rectangular output mosaic, then sections of the mosaic are filled with a background value. MRT reads the `_FillValue` for each input band and uses that value for the output background fill. If no `_FillValue` is specified, then the default value is 0. In the `Mosaic Array` (see below), fill areas are designated as `file[-9]`.

```

MODIS Mosaic Tool (v4.1 March 2009)
Start Time:  Mon Mar 21 11:18:58 2011
-----
Input filenames (4):
    D:\ModisSave\testdata\MOD12Q1.A2000289.h24v02.002.2001103225932.hdf
    D:\ModisSave\testdata\MOD12Q1.A2000289.h25v02.002.2001103225927.hdf
    D:\ModisSave\testdata\MOD12Q1.A2000289.h26v03.002.2001103225927.hdf
    D:\ModisSave\testdata\MOD12Q1.A2000289.h27v03.002.2001103231505.hdf
Output filename: D:\ModisSave\testdata\TmpMosaic.hdf
Mosaic Array:
    file[ 0]  file[ 1]  file[-9]  file[-9]
    file[-9]  file[-9]  file[ 2]  file[ 3]
Mosaic : processing band Land_Cover_Type_1
% complete (1200 rows): 0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
% complete (1200 rows): 0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
Output mosaic image info
-----
output image corners (lat/lon):
    UL:  70.000000000000 120.006250000000
    UR:  70.000000000000 180.000000000000
    LL:  50.000000000000 120.006250000000
    LR:  50.000000000000 180.000000000000
output image corners (X-Y projection units):
    UL:  6671703.118599000387 7783653.638365999795
    UR:  11119505.197665000334 7783653.638365999795
    LL:  6671703.118599000387 5559752.598833000287
    LR:  11119505.197665000334 5559752.598833000287

```

band	type	lines	smpls	pixsiz	min	max	fill
1) Land_Cover_Type_1	UINT8	2400	4800	926.6254	0	254	255

End Time: Mon Mar 21 11:19:09 2011

Finished mosaicking!

Batch Processing Options

The command line MRT is fully scriptable for batch processing of MODIS data. There are a number of ways to use this functionality, all of which involve the creation of parameter files for each input and directing the MRT to use them for processing.

Users may execute specific jobs with parameter files that are simple to automate using either script/batch files or iterative commands depending on the host operating system. The `MRTBatch.jar` capability is also available to facilitate the set up for running batch processing jobs.

Windows Batch Processing

For example, a user may wish to project a time series of fire data to the same mapping grid. Any text editor can be used to create a parameter file (`.prm`) for each file in the series. The scenario below is based on only 5 files, but applies to any number of inputs.

Input File List:

```
MOD14A2.A2005185.h10v04.005.2005185215618.hdf
MOD14A2.A2006185.h10v04.005.2006185181530.hdf
MOD14A2.A2007185.h10v04.005.2007186022514.hdf
MOD14A2.A2008185.h10v04.005.2008185201335.hdf
MOD14A2.A2009185.h10v04.005.2009185224517.hdf
```

.....

Parameter File `MOD14A2.A2005.prm`, saved from a GUI run or created as recommended in Appendix A:

```
INPUT_FILENAME =
D:\MRT4.1\input\MOD14A2.A2005185.h10v04.005.2008044015307.hdf

SPECTRAL_SUBSET = ( 1 1 )

SPATIAL_SUBSET_TYPE = INPUT_LAT_LONG

SPATIAL_SUBSET_UL_CORNER = ( 46.0 -105.0 )

SPATIAL_SUBSET_LR_CORNER = ( 43.0 -93.0 )

OUTPUT_FILENAME = D:\MRT4.1\output\sd_fire_2005.tif

RESAMPLING_TYPE = NEAREST_NEIGHBOR

OUTPUT_PROJECTION_TYPE = LA

OUTPUT_PROJECTION_PARAMETERS = (
```

```
6370997.0 0.0 0.0
0.0 -100.0 45.0
0.0 0.0 0.0
0.0 0.0 0.0
0.0 0.0 0.0 )
DATUM = NoDatum
OUTPUT_PIXEL_SIZE = 1000
```

The parameter file is edited and saved for each of the input files, changing the input and output filenames for each file. This can be a manually intensive process if large numbers of files are required (see below for description of the automated batch processing functions in `MRTBatch.jar`).

Parameter File List:

```
MOD14A2.A2005.prm
MOD14A2.A2006.prm
MOD14A2.A2007.prm
MOD14A2.A2008.prm
MOD14A2.A2009.prm
.....
```

Then a batch file is written to encapsulate the repetition of resample execution on each file.

Batch File `sd_fire.bat`:

```
resample -p=MOD14A2.A2005.prm
resample -p=MOD14A2.A2006.prm
resample -p=MOD14A2.A2007.prm
resample -p=MOD14A2.A2008.prm
resample -p=MOD14A2.A2009.prm
.....
```

And finally, the batch file is run from the command prompt window.

Executing Batch Command:

```
C:\Program Files\MRT4.1\MRT\bin>sd_fire.bat
```

UNIX Bourne Shell Batch Processing

A non-windows alternative uses iteration such as the following UNIX Bourne shell for loop rather than a `.bat`, but the parameter files need to be generated just the same (see below for description of the automated batch processing functions in `MRTBatch.jar`).

```
for i in *.hdf
do
    resample -p $i.prm
done
```

Direct Argument Batch Processing

MRT processing can be directed by parameter files, as used in the Windows and UNIX batch processing descriptions above, but it also responds to direct argument input. The arguments, detailed in the previous “Resample Tools” section, provide the same functionality as, and may be used instead of or in addition to, the parameter file. These optional arguments provide no additional functionality, but do make it easier to control the `resample` executable because it may greatly reduce the number of parameter files needed.

The optional arguments are designed to override fields in the parameter file. Consider a scenario in which there are a number of MODIS input data files to process by performing essentially the same operations on each data set (transformation to an output projection, spectral subsetting, spatial subsetting, etc.). Rather than creating a different parameter file for each input data product (a tedious and error-prone process at best), a single parameter file serves the purpose with overriding input and output filename changes addressed via command-line arguments.

Consider the following shell iteration:

```
for i in *.hdf
do
    resample -p prmfile.prm -i $i -o `basename $i hdf`tif
done
```

This example reprojects every HDF-EOS file in the current directory using a single parameter file `prmfile.prm`. The input and output filenames are changed in the parameter files for each file. The output is written to GeoTIFF with the same base filename as the input file.

Any of the fields listed in the previous “Resample Tools” section can be modified as direct arguments

Automated Batch Processing

Since the 2008 release of MRT, an automated batch processing package is available that runs MRT from command line with minimal interference from the user. It sits with the other MRT software in the `bin` directory (e.g., `C:\Program Files\MRT\Tool\bin`) and is called `MRTBatch.jar`. This file contains the scripts needed to automate `.prm` and `.bat` builds and to execute the batch processing. Its usage is described in the following steps.

- Gather all input files
- Create parameters files for each input and write a batch script
- Execute the batch script

The sections above on Batch Processing describe the functions wrapped into `MRTBatch.jar` which automates this otherwise very manual process. To use `MRTBatch.jar` follow the steps below.

1. Step one is to simply gather all input files in an exclusive directory (containing nothing but MRT input).
2. Step two will create a base parameter file using the MRT GUI. Open one input file in the GUI and enter the desired parameters (see MRT GUI section below). Instead of clicking “Run” at the end, Click “Save Parameter File” to save the resulting `.prm`. Then exit the MRT GUI.
3. Step 3 invokes the `MRTBatch.jar` to create parameter files (`.prm`) for each input file in the input data directory. It also writes out a batch script that is later used to execute the processing jobs (`MRTBatch.bat`).

From the Command Prompt window, navigate to the MRT `bin` directory. This is where the `.jar` file lives. Type the following command to initiate parameter file and batch script generation (place all fields in one line; i.e., do not break this statement into multiple lines).

```
java -jar MRTBatch.jar -d input_directory -p parameter_directory\  
input_parameter_file -o parameter_file_directory
```

where

`input_file_directory` is the directory in which all input files are placed (e.g., `C:\Project\InputFiles`);

`parameter_file_directory` is the directory where the parameter file created using the GUI (-p) was saved and to which the parameter files built by this application will be written (-o);

`input_parameter_file` is the parameter file was saved from the GUI (e.g., `mrtauto_test.prm`).

Note that the `-o parameter_file_directory` is not necessary. Users can omit it from the command line, which will result in a `prm` directory containing the new parameter files being built in the `input_file_directory` instead.

4. Step four will run the `MRTBatch` created just above. The `MRTBatch.bat` contains a list of commands to run the MRT on every input file according to the parameter files.

Change directories at the Command Prompt to go to the MRT `bin` directory. Tell MRT what to do by typing the directory path and name of the batch file, such as shown below.

Windows:

```
C:\Program Files\MRT\Tool\bin\mrtbatch.bat
```

UNIX:

```
"$HOME\My MRT\bin\mrtbatch"
```

The results of the processing job will echo out in log form on the command prompt display, and when completed, files should be ready in the output directory specified in the original base parameter file.

MODIS Reprojection Tool GUI

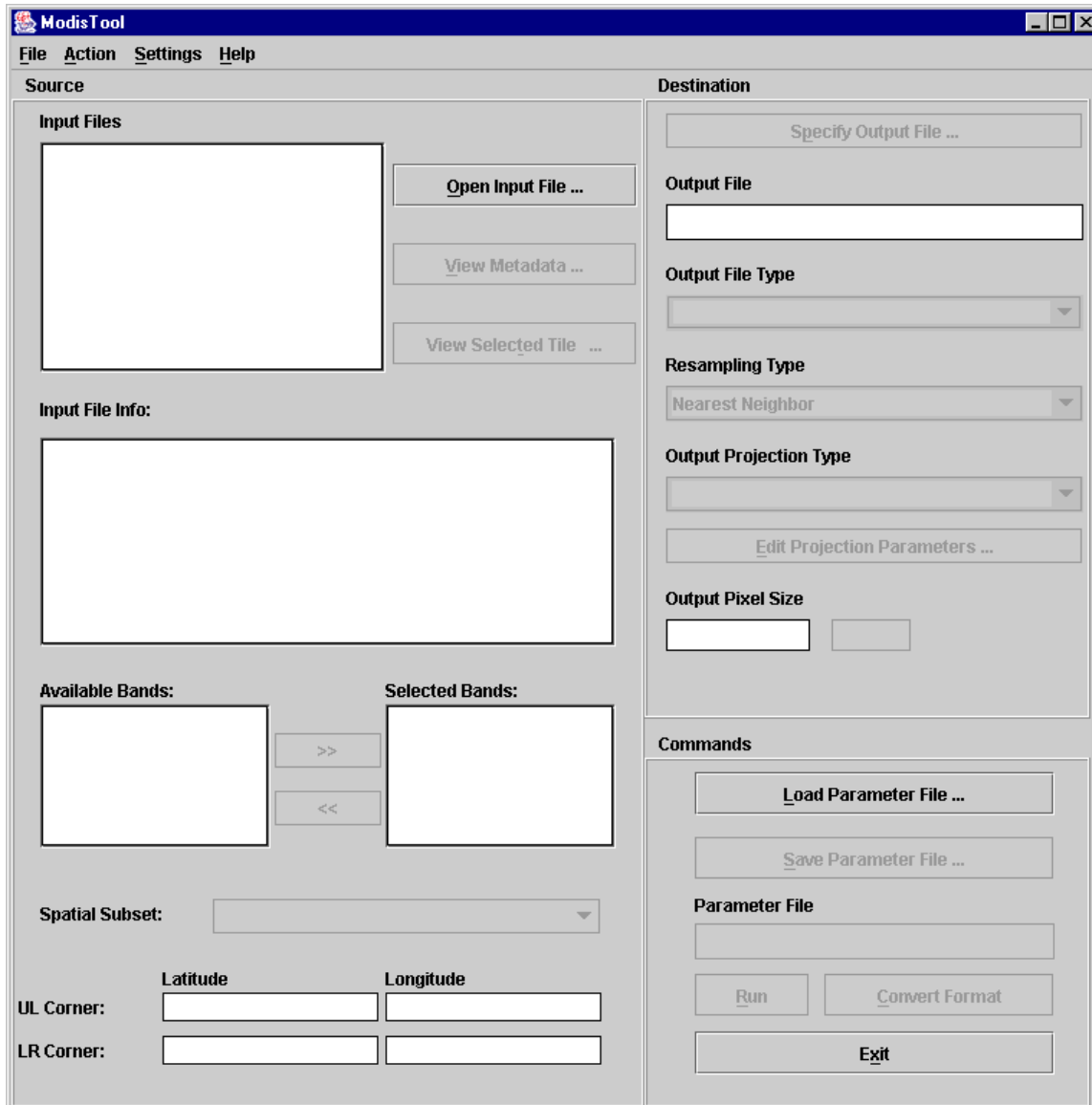
Resampling Tool

This section of the User's Manual describes how to run `resample` from the MRT GUI. This GUI serves two purposes: one, it helps a novice user, or one with light processing needs, to enter all requisite inputs, and process those jobs; two, it helps users with heavier processing requirements to enter all requisite inputs and save them in a parameter file, which they may edit and use to run `resample` from command line. Users may also load saved parameter files into the GUI for inspection and/or editing.

To start the GUI, type `ModisTool` from within the `MRT/bin` directory at the command prompt, doubleclick on the `MRT` icon on the desktop (if one was created), or click on `ModisTool.bat` in the `MRT bin` directory. A `ModisTool` window will pop up on the screen in a few seconds. Failure to display the GUI indicates a configuration error, most likely in the `PATH` to the Java executables on the host machine. See Manual Installation instructions above to determine how to check and resolve any such errors. The `ModisTool.bat` (Windows) or `ModisTool` (UNIX) may need to be edited to identify a valid Java path. Another potential reason is that a session restart was not completed after the software installation.

To run `resample` from the GUI, users need to fill in the various fields and click `Run`. In general, complete fields in the order listed, working down the left (`Source`) side and progressing down the right (`Destination`) side. For example, spectral and spatial subsetting parameters may not be selected until an input file is opened, output projection parameters may not be entered until an output projection type is specified, etc. The contents of any field may be changed at any time prior to running `resample`. Simply click on the field and make the desired changes.

Along the top left banner on the GUI is a menu bar labeled with `File`, `Action`, `Settings`, and `Help`. The `File` menu includes options for opening an input file, opening a parameter file, or exiting the GUI. Other options for specifying an output file and saving a parameter file become active on the `File` menu after `Source` information (the fields on the left side of the pane) and `Destination` information (right side) have been entered respectively. These `File` options mirror the button functions on the GUI.



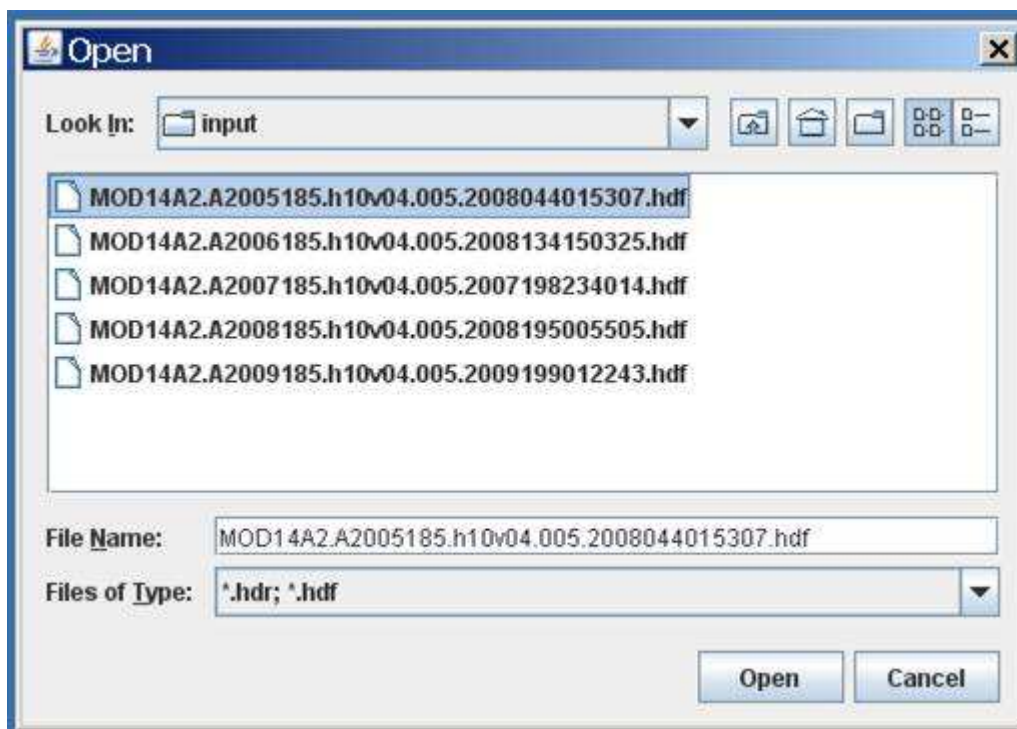
The Action menu includes the Run and Convert functions, and is simply an alternative to clicking the Run and Convert Format buttons on the bottom right side of the GUI. Use the Settings menu to set the default directory locations for input files, output files, and parameter files. This is useful, as otherwise MRT will default to its bin directory for everything. The Help menu contains “About” information, and will open a popup displaying the version and release date for the installed MRT.

Opening an Input File

Start on the left pane of the GUI, where `Source` information is entered. Click `Open Input File` to the right of the `Input File` field. This will pop up an input file selection dialog box, allowing selection of the desired input file by pointing and left-clicking with the mouse. The default directory is set to the

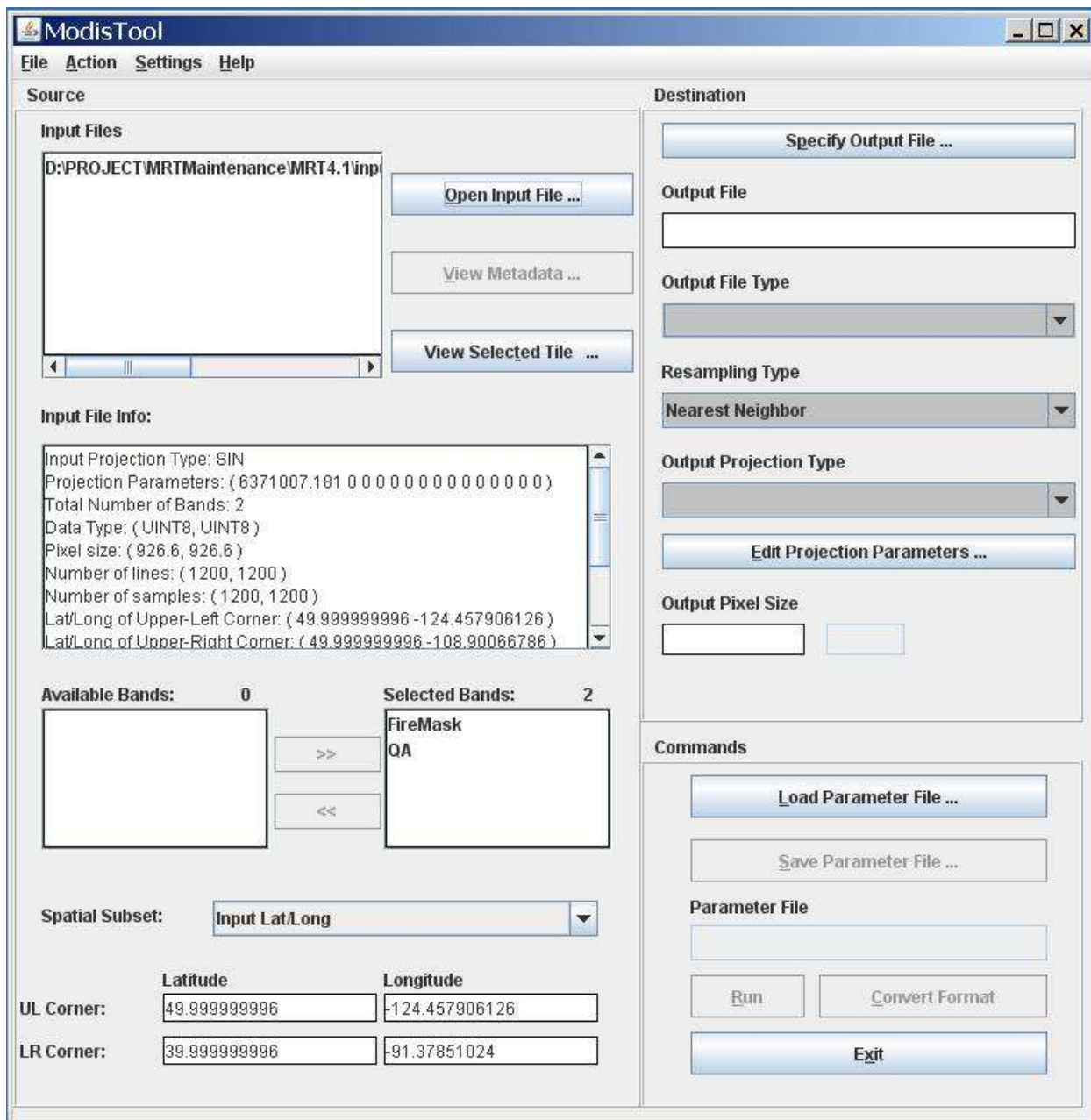
MRT bin, so users need to either set a preferred directory using the `Settings` menu on the GUI's top banner, or navigate to the directory containing the input files using the browse tools in the dialog box. These include a drop-down directory tree search, a button to move up one directory, a button to return to the user's home directory, a button to create a new directory, and options to list or show details of the files in the open directory.

If several adjacent tiles are desired for processing, MRT will allow selection of multiple input filenames. This is discussed later in the "Mosaic Tool" section below.



An input file is selected either by typing its name into the `File Name` field at the bottom of the dialog box, by double-clicking on the file name in the directory list, or by highlighting the file name with a single click and pressing `Open`.

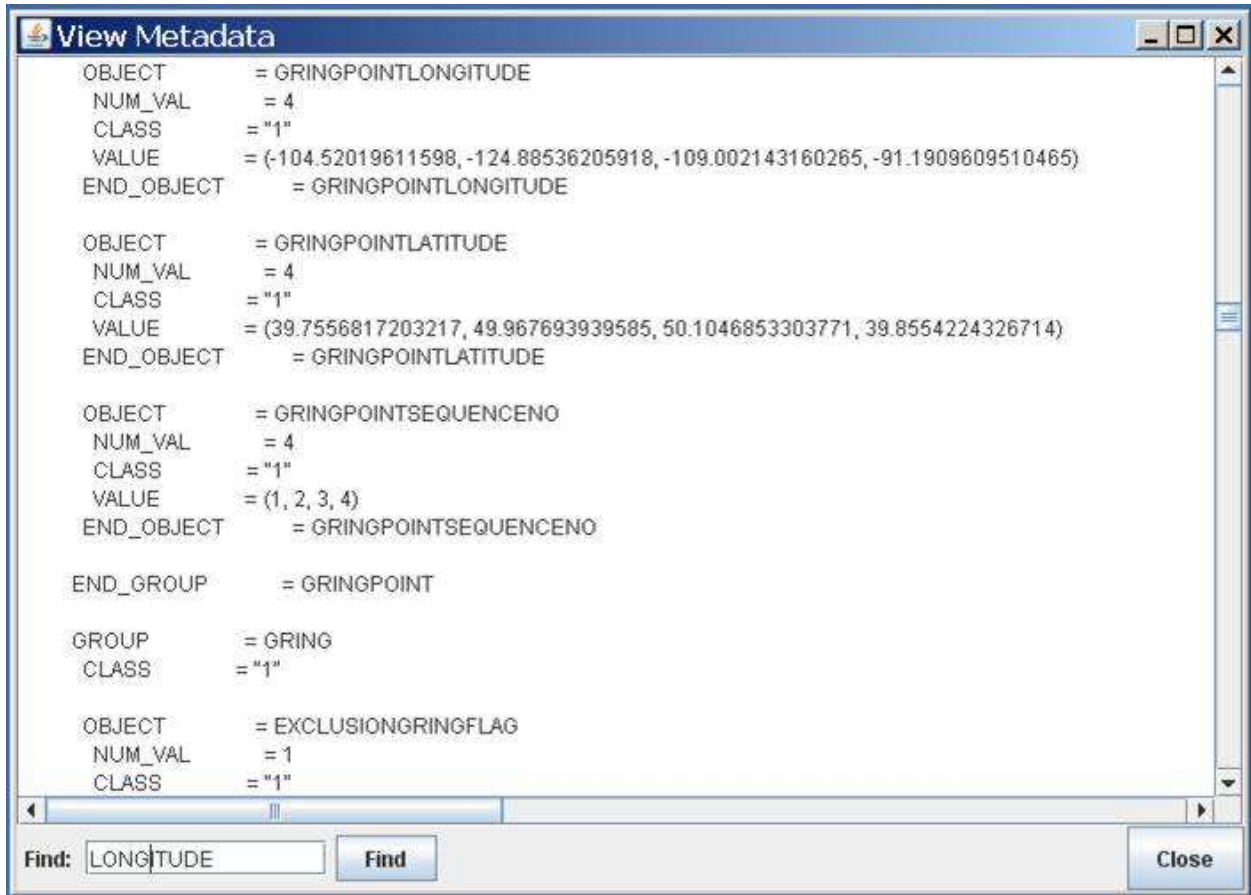
When an input file is opened, `Source` information is loaded into the GUI. The file name will appear in the `Input File` field, basic metadata will be displayed in the `Input File Info` window, the `Selected Bands` window will list the science data sets (bands) available in the input file, and the corner coordinates are shown in the `Latitude` and `Longitude` fields.



Metadata Examination

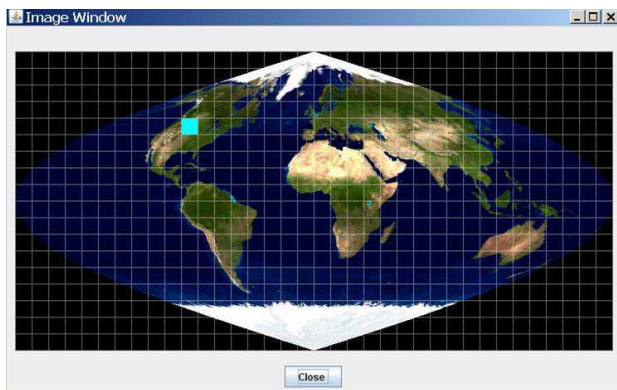
The basic metadata displayed in the `Input File Info` window includes the input projection type (Sinusoidal), the input projection parameters, the total number of bands, the data type of each band, the pixel size of each band, the number of lines and samples in each band, and the corner coordinates of the tile. Users may examine detailed metadata by highlighting the filename in the `Input Files` box, then clicking on `View Metadata` below the `Open Input File` button. This will pop up the internal metadata from the input HDF-EOS tile, including its inventory metadata, archive metadata, and its grid structure.

The View Metadata window offers a Find service in the lower left corner, into which a case sensitive text string can be typed to parse the internal metadata for terms of interest.



Tile Locator

Just beneath the View Metadata window is another button labeled View Selected Tile. This feature will bring up a global map on the Sinusoidal grid, and highlight the location of the tiles in the Input Files box in light blue.



The UL Corner and LR Corner are populated with the bounding coordinates by default. These fields can be edited to specify an area of interest by simply clicking in the box and entering the bounds of the desired subset. If the `Spatial Subset` type is set to `Input Lat/Long`, enter corner points in decimal degrees. To subset using `Input Line/Sample`, specify line/sample pairs using a zero-based coordinate system assuming the upper left corner is (0,0).

MRT will use `Input Lat/Long` or `Input Line/Sample` to automatically compute the other two rectangle corners (upper right and lower left) in input space. All four corner points will be projected into output space, using the map projection specified later on the `Destination` side of the GUI. A minimum-bounding rectangle is computed in output space that contains the four projected points. All points inside this rectangle in output space are mapped back into input space for projection.

When creating a subset based on `Output Projection X/Y`, these coordinates must be specified in the same units used for the projection (i.e., decimal degrees for Geographic and meters for all other projections). The upper right and lower left corners will be computed from the specified `UL Corner` and `LR Corner` to create a rectangle in output space defined by the map projection specified later. These output corners are mapped back to input space to determine their location in input space.

The final product after processing should have the same image corner coordinates as specified by the user, but note that MRT will adjust the lower right corner based on output pixel size if its location does not fall in an integral number of lines and samples.

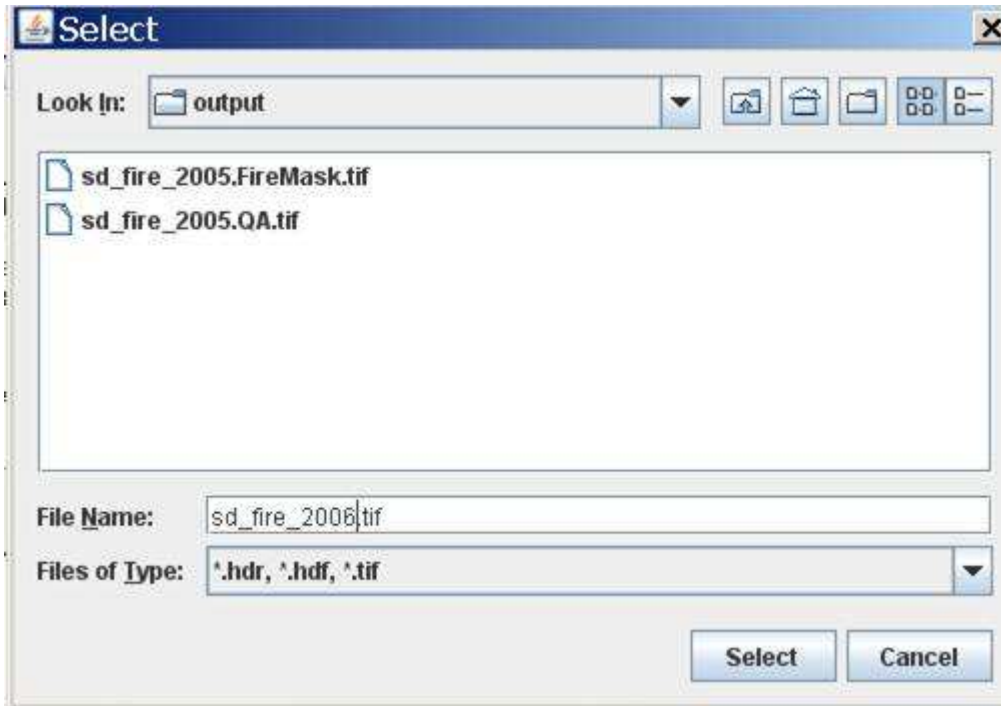
The MRT expects that the upper left and lower right corner point values of any input or output HDF-EOS file, whether appearing in the `Input File Info` or entered in `Spatial Subset`, will reflect the outer extents of the image (i.e. upper left of the upper left and lower right of the lower right) not to the center of the upper left and lower right pixels. Be aware however, that GeoTIFF output will be tagged with coordinates from the center of pixel.

Specify Output File

After entering all the necessary `Source` information, the next step is to define the `Destination` parameters. The first is the name of the output file, which is user-defined through a dialog box invoked by clicking the `Specify Output File` button on the top of the right pane on the GUI. Similar to the dialog for opening files, output file naming is done by navigating to a preferred directory, then selecting an existing file, or by typing in a new output file name, as shown below. Recall that a preferred default directory for all output files may be configured using the `Settings` menu on the top banner of the GUI.

MRT will expect a file extension appended to the file name, and will not accept a name without it. The file extension indicates the file format of the output image. Adding `.hdf` will tell MRT to output to HDF-EOS, `.tif` to output to GeoTIFF, and `.hdr` to output to raw binary.

If an existing file is selected without modifying the file name, an overwrite warning will pop up.



The output file format also determines how many files will be delivered at the end of processing. HDF-EOS output consists of one file, and however many bands were processed are packed as science data sets within the file. GeoTIFF and raw binary formats store one band per file, so each band selected for processing results in an output file. The user does not need to specify a new output GeoTIFF name for each selected band, as the band name is automatically appended to the base output file name.

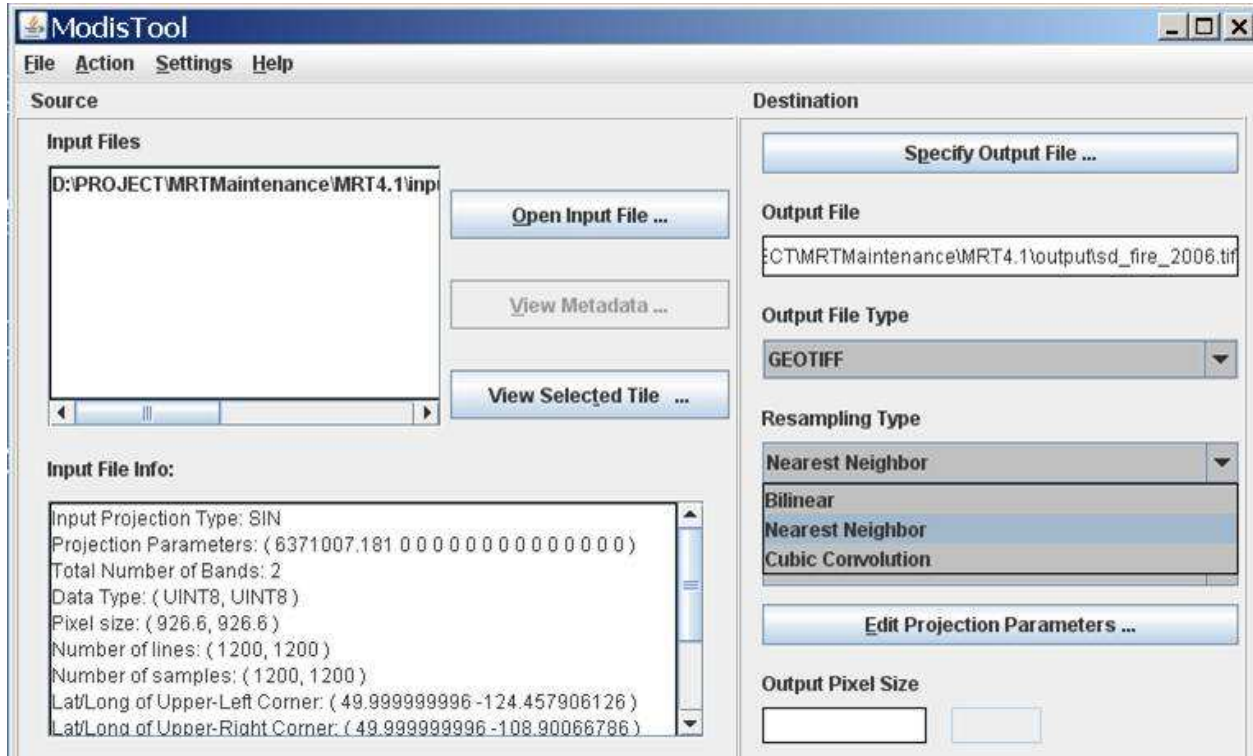
For example, if both bands in a MOD14A2 tile were processed (QA and FireMask), HDF-EOS output would consist of a single `sd_fire_2006.hdf`, while in GeoTIFF the same output would be delivered as `sd_fire_2006.QA.tif` and `sd_fire_2006.FireMask.tif`.

Output File Type

This option is positioned under the `Output File` window, and can be used after a file name is selected to change its format without having to relocate and rename it. Select `GeoTIFF`, `HDFEOS`, or `RAWBINARY` from the `Output File Type` drop-down list to automatically change the extension of the file name entered in the `Output File` field.

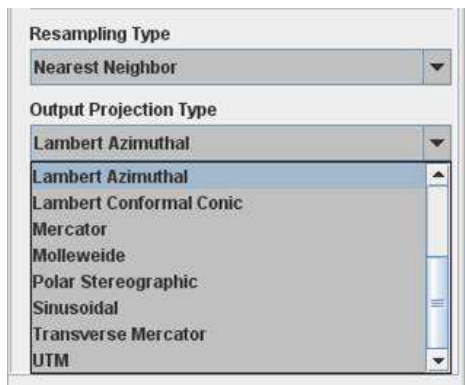
Resampling Type

MRT offers three resampling methods from the *Resampling Type* drop-down list (Nearest Neighbor, Bilinear, or Cubic Convolution). Note that when projecting thematic data sets, such as a fire mask or any QC band, nearest neighbor is required to avoid mixing categorical values. For example, if 1 = snow, 2 = water, and 3 = fire, resampling methods using averaging will potentially mix 1 and 3 and output a 2.



Output Projection Type

The *Output Projection Type* (Geographic, UTM, etc.) is selected from a drop-down list, and its parameters are entered using the *Edit Projection Parameters* button.



Clicking the `Edit Projection Parameters` button will pop up a dialog box in which users may enter or edit up to 15 gridding parameters. Note that the default values appearing in the pop up box are not automatically overwritten when their fields are clicked. Users must delete or highlight them for replacement. Integer values are automatically converted to floating point. Some parameter fields will be grayed out (uneditable) when they are not used for a particular output projection. These fields are based on the table of projection parameters in Appendix C.



When editing the projection parameters, an output datum may be specified. For all projections, the default will be `NoDatum`. The “Datum Conversions” section of this manual discusses more information on this field, but it is important to iterate that either a `Datum` or the first two projection parameters may be specified, but not both. If both are specified, then the MRT will exit with an error.

A special note for UTM projections: MRT will automatically set the `UTM Zone` when UTM is set as the `Projection Type`. Although it is an option, it is not necessary to enter a `UTM Zone` value in the `Edit Projection Parameters` box.

A special note for Lambert Conformal Conic projections: Although the standard parallels and latitude of origin fields are both open for editing, enter a value into only one or the other. Setting standard parallels effectively does the job of the latitude of origin in LCC projection, and having the `OriginLat` specified as well as the `STDP1` and `STDP2` will place the projected output at inaccurate latitude/longitude coordinates.

Output Pixel Size

Right under `Edit Projection Parameters` is an `Output Pixel Size` box. This can be used to specify a different pixel size for output files than was input to MRT. A new pixel size can be entered using decimal degrees for the Geographic projection and meters for all other projections. For reference, 250-m is 0.00225 degrees, 500-m is 0.0045 degrees, and 1,000-m is 0.009 degrees. If left blank, the output pixel size remains the same as the corresponding input pixel size for each selected band. If specified on the GUI, the output pixel size will be used for all selected bands. This helps normalize MODIS pixels which, due to their basis in arcseconds, are actually 926.6-m pixels instead of 1,000-m.

The MRT `Output Pixel Size` function is not recommended as a tool to rescale 1,000-m data to higher resolutions.

Load or Save Parameter File

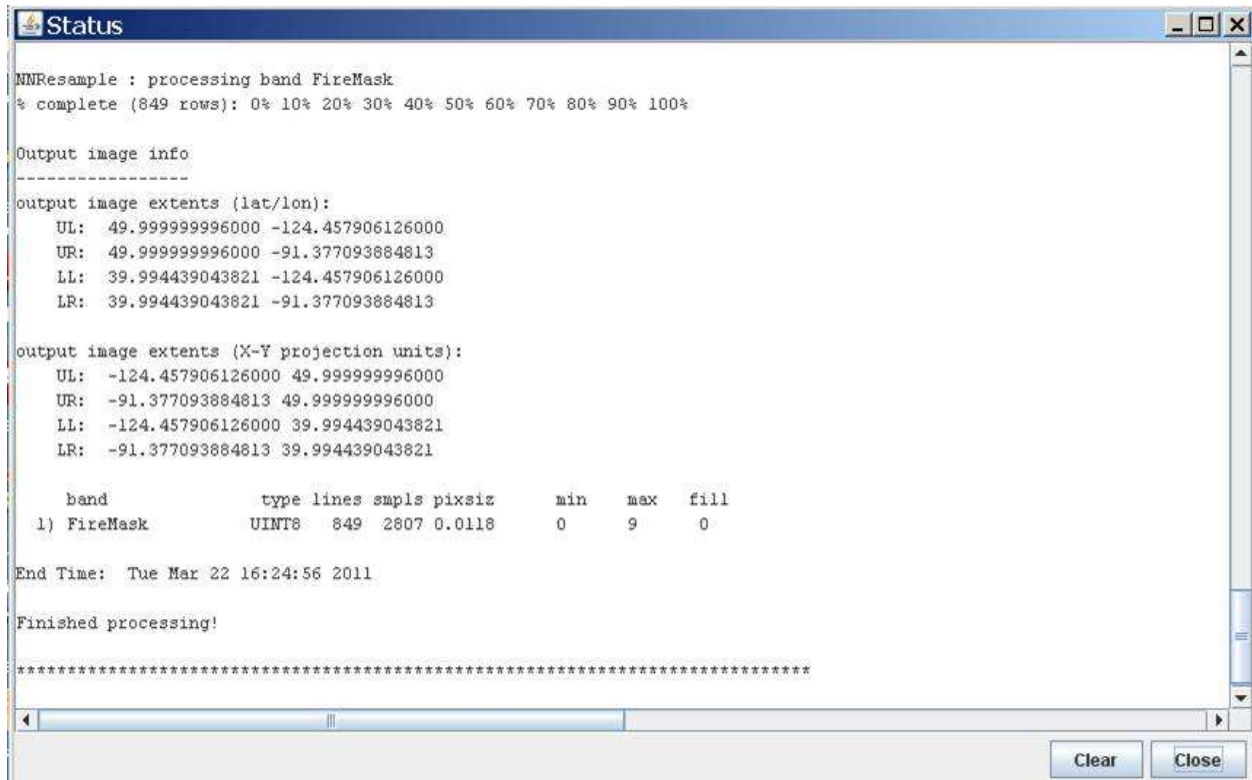
The MRT offers a method to retain all the `Source` and `Destination` parameters just entered through the steps above. This is useful if there is more than one file to process to the same specifications, if users need to input the same parameters at a later session, or if users do not want to enter all the parameter information repeatedly. To save all information just entered on the GUI, click on `Save Parameter File` and type in a file name. Unlike specifying an image output file, it is not necessary to include a file extension with the name and a file with the extension `.prm` is created in the directory of choice (see the sample parameter file included in the “Batch Processing” section). An existing file name may be used, but the file will be overwritten unless it is modified. If the last parameter file used for processing was not saved, it is retrievable from the MRT `bin` directory, where the most recently executed parameters are saved in `TmpParam.prm`.

To restore saved parameters, simply click `Load Parameter File`, navigate to the desired `.prm` and click `Open`. This will restore all the parameters in the `Source` and `Destination` fields on the GUI. The fields can be edited any time prior to clicking `Run`. The saved `.prm` can also be used to feed command line processing (see “Batch Processing” section).

Executing resample

Having entered all the desired `Source` and `Destination` parameters for projection, click `Run` either on the bottom of the GUI’s right pane, or under the `Action` menu on its top left. As `resample` executes, a `Status` window will pop up to display job progress. Its contents are simultaneously being appended to a processing log that tracks every MRT session. The log is useful for troubleshooting, and is found in the

MRT bin as `resample.log`. The log is an ASCII text file that users may edit or print with standard text file tools.



```
MRT Status
-----
MNResample : processing band FireMask
% complete (849 rows): 0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%

Output image info
-----
output image extents (lat/lon):
  UL: 49.999999996000 -124.457906126000
  UR: 49.999999996000 -91.377093884813
  LL: 39.994439043821 -124.457906126000
  LR: 39.994439043821 -91.377093884813

output image extents (X-Y projection units):
  UL: -124.457906126000 49.999999996000
  UR: -91.377093884813 49.999999996000
  LL: -124.457906126000 39.994439043821
  LR: -91.377093884813 39.994439043821

  band          type lines smpls pixsiz      min   max   fill
  1) FireMask    UINT8   849  2807 0.0118      0     9     0

End Time: Tue Mar 22 16:24:56 2011

Finished processing!

*****

[Progress bar showing 100% completion]

[Clear] [Close]
```

When the Run button is clicked, the GUI creates a temporary parameter file (`TmpParam.prm`) in the bin directory and runs `resample` using that parameter file (`resample -p TmpParam.prm`). This file is overwritten on subsequent runs, but not deleted, so it can be examined with any ASCII text file viewer/editor when processing is complete.

Exiting the GUI

To exit the MRT GUI, click the `E_xit` button or click `F_ile, E_xit` on the menu bar on the top banner on the GUI. The standard operating system commands (e.g., double click on the “X” in the upper right hand corner of the window, in Microsoft Windows).

Format Conversion

The previous section “Resampling Tool,” directed use of the MRT GUI to execute the `resample` software. The MRT also provides a file format converter, which allows conversion from input HDF-EOS to output GeoTIFF or raw binary without resampling.

Running the format converter is similar to the `resample` option (please see section above for details on parameters discussed here). The `Input File` information must be specified first, then the `Output File name` and `Output File Type`. As with resampling, the default for format conversion is to process all bands across the entire tile unless otherwise specified. Spatial and spectral subsetting are supported in format conversion, so those parameters may be input as well.

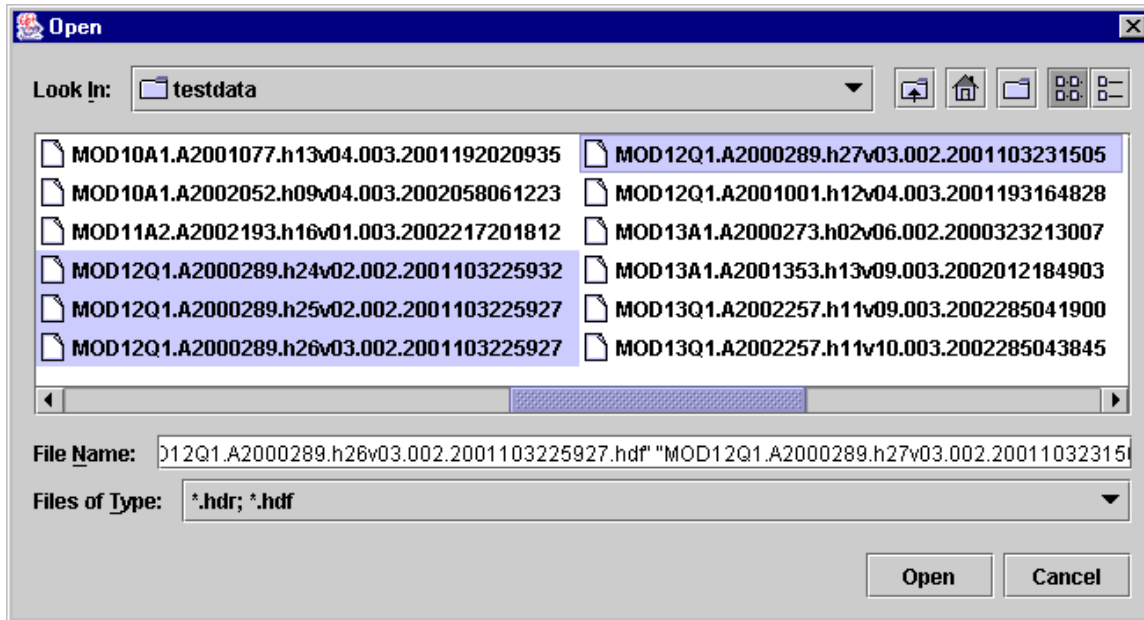
Unlike the resampling process, `Resampling Type`, `Output Projection Type`, `Output Projection Parameters`, and `Output Pixel Size` are not used for file format conversion. The converter will use output files with specifications identical to the inputs for these parameters, ignoring any entries to these fields on the GUI

To perform a file conversion, simply click the `Convert Format` button, next to the `Run` button at the bottom right of the GUI, or use the `Convert Format` option under the `Action` menu on the top banner of the GUI. Either will be grayed out (unclickable) until the information described above is entered.

Mosaic Tool

The MRT will allow the user to specify several input files to be processed. The MRT will mosaic the input tiles into one large image, then process that image as specified by output parameters entered by the user. Processing several tiles is very similar to processing a single tile. The only difference is that multiple tiles are selected in the `Open Input File` process.

The following example shows the user selecting four tiles for processing. The user may select multiple files by using the `Shift` and `Ctrl` keys along with pressing the left mouse button. The `Shift` key will select all files between the previous mouse click and the current mouse click. The `Ctrl` key will simply add the current mouse click to the list of files.



The mosaic tool requires that all input tiles are of the same product type and thus contain the same bands which remain the same size, data type, projection, pixel size, etc. for all input tiles. If the input tiles are not of the same product type, the MRT will exit with an error.

When multiple tiles are mosaicked, the output is in HDF format and named `TmpMosaic.hdf`. The `TmpMosaic.hdf` is input to subsequent resampling actions. The HDF4 libraries currently included in MRT software impose a 2 gigabyte (GB) limitation on file sizes, so it is possible for MRT to reject processing because `TmpMosaic.hdf` file size exceeds this limit. This has been observed when numerous tiles are input for mosaicking (e.g., Africa), but is dependent on the data type. A 250-m product has more volume than a 1,000-m product, and so a mosaic of twenty tiles with 1,000-m pixels may be more successful than the same for 250-m pixels.

Credits

The MODIS Reprojection Tool was developed as a collaborative effort between the Land Processes Distributed Active Archive Center at the U.S. Geological Survey Earth Resources Observation and Science Center and the South Dakota School of Mines & Technology.

Contacts

LP DAAC is the main user support facility for this tool. Download and installation assistance, bug reports, and other comments may be addressed to:

LP DAAC User Services
U.S. Geological Survey (USGS)
Earth Resources Observation and Science Center (EROS)
47914 252nd Street
Sioux Falls, SD 57198-0001

Phone: 605-594-6116

Toll Free: 866-573-3222

Fax: 605-594-6963

Email: lpdaac@usgs.gov

Web: <https://lpdaac.usgs.gov>

Appendix A: MRT Parameter File Format

Parameter files are user-editable ASCII text files that contain information required by the MODIS Reprojection Tool (MRT) for processing MODIS data. Information entered by the user in the MRT Graphical User Interface (GUI) is stored in a parameter file, either for future use, or to run the `resample` executable. In fact, the MRT GUI is basically a parameter file editing tool. This document specifies the MRT parameter file format, which for the most part stores information as field-value pairs corresponding to GUI fields and values.

File naming conventions

By convention, all related files in a data set should be given the same base filename. Different extensions indicate the file type: parameter files (`.prm`), header files (`.hdr`), metadata files (`.met`), HDF-EOS files (`.hdf`), and GeoTIFF files (`.tif`).

Editing parameter files

Parameter files may be created and modified in two ways: by directly editing the parameter file with an ASCII text editor, or by using the MRT GUI. In the MRT GUI, selecting `Load Parameter File` allows the user to load values from an existing parameter file. Selecting `Save Parameter File` allows the user to save the current parameter values to a specified parameter file.

Parameter file format

The parameter file consists of field-value pairs and comments. Comments begin with the '#' character, and extend to the end of the line. Each field must begin on a new line, and may span more than one line for convenience and readability. Fields may occur in any order. All field-value tokens must be separated by white space (including the equals and parentheses symbols)¹¹.

To verify the format of any of the fields below, enter the desired information using the GUI, then use the `Save Parameter File` button to generate a sample file that can be examined for content.

¹¹ In the command-line MRT, space around tokens is not required. List values may be separated by either white space or commas.

```
INPUT_FILENAME = < directorypath\inputfilename.hdf >
```

The input data file name is a required field that provides requisite processing information. Only MODIS HDF-EOS file names are accepted. The file name may contain a directory path. UNIX users are advised to enclose file names in “double quotes.” An invalid file name or non-swath product will generate an error.

```
SPECTRAL_SUBSET = ( band1 band2 ... bandn )
```

This field is optional; by default, all input image bands will be selected. This is an n -element array of binary values corresponding to the n bands (Scientific Data Set or SDS elements) in the input data set. In this array, a 1 indicates that a band was selected, and a 0 indicates that it was not. If there are fewer binary values than input image bands, the remaining bands will not be selected. If there are more binary values than input image bands, the extra values will be ignored.

```
SPATIAL_SUBSET_TYPE = < INPUT_LAT_LONG, INPUT_LINE_SAMPLE, OUTPUT_PROJ_COORDS >
```

This field is required only if spatial subsetting is desired. Spatial subsets may be defined using input latitude and longitude coordinates, input lines and samples, or output projection X and Y coordinates. If neither of the corner parameters (see below) is specified, this field defaults to `INPUT_LAT_LONG` and the entire image will be processed.

```
SPATIAL_SUBSET_UL_CORNER = < ( UL_line UL_sample ), ( UL_lat UL_lon ),  
(UL_proj_x UL_proj_y ) >
```

```
SPATIAL_SUBSET_LR_CORNER = < ( LR_line LR_sample ), ( LR_lat LR_lon ),  
(LR_proj_x LR_proj_y ) >
```

These fields are required for spatial subsetting and define the coordinates of the upper-left and lower-right corners in units appropriate to the selected `SPATIAL_SUBSET_TYPE`. By default, the entire input image will be selected. The bounding rectangular coordinates in the input file’s metadata are used to determine the image coordinates.

The parameter field is formatted as follows for each subset type.

```
SPATIAL_SUBSET_TYPE = INPUT_LAT_LONG  
SPATIAL_SUBSET_UL_CORNER = -1.969280203 37.840197592  
SPATIAL_SUBSET_LR_CORNER = 26.054383372 16.574757586
```

```
SPATIAL_SUBSET_TYPE = INPUT_LINE_SAMPLE  
SPATIAL_SUBSET_UL_CORNER = 20 20  
SPATIAL_SUBSET_LR_CORNER = 1353 2029
```

```
SPATIAL_SUBSET_TYPE = OUTPUT_PROJ_COORDS
SPATIAL_SUBSET_UL_CORNER = 635521.0 1235874.0
SPATIAL_SUBSET_LR_CORNER = 137256.0 82544.0
```

In the case of multi-resolution data sets, the highest resolution of any spectral band is assumed for line/sample values. MRT expects float values (containing a decimal point) to define input latitude and longitude coordinates, and integer values to indicate line/sample pairs. If any entered value is float, then INPUT_LAT_LONG will be assumed. Note that spatial subsetting takes place in the input image space, not the output image space.

```
OUTPUT_FILENAME = < directorypath\outputfilename.extension >
```

The output file name is required. The name may optionally contain a directory path. UNIX users are advised to enclose file names in “double quotes.” A file extension must be included as it is used to automatically determine output file type (.hdr = Raw Binary, .hdf = HDF-EOS, .tif = GeoTIFF). An invalid or missing extension will generate an error. Note that an existing file with the same name as specified in OUTPUT_FILENAME will be overwritten by *resample*.

```
RESAMPLING_TYPE = < NN, BI, CC >
```

This field is optional. The default resampling type is NEAREST_NEIGHBOR (NN) unless specified as BILINEAR (BI) or CUBIC_CONVOLUTION (CC).

```
OUTPUT_PROJECTION_TYPE = < AEA, ER, GEO, GOODE, HAM, ISIN, LA, LCC, MERCAT,
MOL, PS, SIN, TM, UTM>
```

This field is required for resampling, but optional for format conversion. The output projection type may be one of the following: Albers Equal Area (AEA), Equirectangular (ER), Geographic (GEO), Interrupted Goode Homolosine (GOODE), Hammer (HAM), Integerized Sinusoidal (ISIN), Lambert Azimuthal (LA), Lambert Conformal Conic (LCC), Mercator (MERCAT), Molleweide (MOL), Polar Stereographic (PS), Sinusoidal (SIN), Transverse Mercator (TM), Universal Transverse Mercator (UTM).

```
OUTPUT_PROJECTION_PARAMETER = < p1 p2 p3 p4 p5...p15 >
```

This field is optional, but recommended for resampling success. The array field contains the 15 output projection parameter values¹². By default, all projection parameter values are set to zero, with the exception of UTM. When the first two UTM projection parameters are zero, the projection will default to the scene center. Projection parameter values are floating point and any integer values entered are automatically converted to floating point. If there are fewer than 15 projection parameter values specified

¹² See Appendix C for more information on projection parameters. Or type *swath2grid -help=<projection>* at the command line.

the remaining values are set to zero. If there are more than 15 values specified, the extra values will be ignored. Coordinate values (latitudes and longitudes) should be entered in decimal degrees.

```
UTM_ZONE = < zone number >
```

This is optional for users selecting UTM output projection type. MRT will choose the appropriate zone value by default, but one may be otherwise specified. Valid values are -60 to +60. If present, the UTM_Zone overrides values specified in the output projection parameters field.

```
DATUM = < NAD27, NAD83, WGS66, WGS72, WGS84 >
```

This field is optional, and defaults to NODATUM. Otherwise a valid datum may be specified for output projections. If the NODATUM option is used, then the user must specify the spheroid semi-major and semi-minor axes in the projection parameters. See the Datum Conversions section for more information.

```
OUTPUT_PIXEL_SIZE = < value >
```

The output pixel size may be specified for each band processed, in output projection units (meters for all projections except Geographic which requires units of decimal degrees). For example, if five bands are selected for processing, and the user wishes for some reason to output one of the bands at a different resolution, the line in the parameter file would look like this below.

```
OUTPUT_PIXEL_SIZE = 1000,1200,1000,1000,1000
```

Otherwise, the default is to output to the same resolution as the input for each band, which is generally the same for each band. This field is optional, but given the native pixel resolution of MODIS swath data, users may utilize it to normalize the output pixels (e.g., change 926.6-m input to 1,000-m output).

If the number of output pixel sizes entered in this field is less than the number of bands processed, then the last pixel size entered is used as the pixel size all bands. For example, it is not necessary to write out the output resolution for all bands if the same output resolution is desired for all bands. Simply entering the following will apply the desired output pixel size to all bands selected for processing.

```
OUTPUT_PIXEL_SIZE = 1000
```

Appendix B: MRT Raw Binary File Format

This document specifies the format of the header file for the MODIS raw binary file format. In this file format, raw binary data and metadata are stored in separate files. The header file contains information required by the MODIS Reprojection Tool (MRT) for processing MODIS data in the raw binary file format. Header files consist of user-editable ASCII text in the format described below.

Raw binary MODIS data are stored in individual data files, with one file per band. In this document, the term “band” is used interchangeably with the HDF term “Scientific Data Set”, or SDS. Within files/bands, data are stored in row-major order, starting at the upper-left corner of the image. The data type may be 8-bit integer, 16-bit integer, 32-bit integer, or 32-bit float. Integer values may be signed or unsigned. Two-byte and four-byte data types are stored in machine-dependent byte order. NOTE: This is new as of version 4.0. All previous versions output raw binary products as big-endian byte-order.

A metadata file contains information about a corresponding data file. Metadata are stored in the metadata file as user-editable ASCII text in ODL format. The metadata file format specification shall be described in another document.

File naming conventions

By convention, all related files in raw binary file format are given the same base filename. Header files are given the `.hdr` extension. Data filenames are generated from the basename and the Scientific Data Set (SDS) name, and given the `.dat` extension, as follows: `basename.SDS_name.dat`.

For example:

<code>MOD09GA.A2000072.hdr</code>	- header file
<code>MOD09GA.A2000072.sur_refl_b01_1.dat</code>	- data file (band 1)
<code>MOD09GA.A2000072.sur_refl_b02_1.dat</code>	- data file (band 2)
etc.	

Because of these naming conventions, there is no need to specify raw binary data filenames inside the header file. Data filenames are automatically generated by appending the appropriate extension onto the basename of the header file.

Header file format

The header file contains information required by the MODIS Reprojection Tool for processing data in the raw binary file format. An ODL-like format is used, as illustrated by the following:

```
# Header file for MOD09GHK.A2000072.h08v05.001.2000084105003

PROJECTION_TYPE = ISIN

PROJECTION_PARAMETERS = ( p1 p2 p3 p4 p5 p6 p7 p8 p9 p10 p11 p12 p13 p14
p15 )

# COORDINATE_ORIGIN = UL

UL_CORNER_LATLON = ( lat lon )      # lat/long in decimal degrees
UR_CORNER_LATLON = ( lat lon )      # for the outer extent of the pixel
LL_CORNER_LATLON = ( lat lon )
LR_CORNER_LATLON = ( lat lon )

# UL_CORNER_XY = ( x y )             # projection coordinates for the
# UR_CORNER_XY = ( x y )             # outer extent of the pixel
# LL_CORNER_XY = ( x y )
# LR_CORNER_XY = ( x y )

NBANDS = n
BANDNAMES = ( band1 band2 ... bandn )
DATA_TYPE = ( t1 t2 ... tn )
NLINES = ( r1 r2 ... rn )
NSAMPLES = ( c1 c2 ... cn )
PIXEL_SIZE = ( s1 s2 ... sn )
MIN_VALUE = ( v1 v2 ... vn )
MAX_VALUE = ( v1 v2 ... vn )
BACKGROUND_FILL = ( f1 f2 ... fn )
DATUM = {Optional datum value}
BYTE_ORDER = {little_endian or big_endian}
```

Editing header files

Header files may be created and modified by directly editing the header file with an ASCII text editor. Running MRT and selecting raw binary file output also creates header files. In this case, the MRT automatically generates an output header file, along with output data files.

Notes

- Fields may occur in any order. Each field should begin on a new line, but may span more than one line for readability. Field values should be separated by white space. Otherwise, there are no significant restrictions on field formatting.
- All text following a “#” on a line is considered to be a comment. Some comment lines are automatically generated by MRT for reprojection to a raw binary output image, for informational purposes only.
- Field names should be largely self-explanatory. Lower-case items represent numeric and string values for the various fields. Several fields require arrays of n values for imagery containing n bands (SDS elements). Different bands may have different data types, dimensions, resolutions, etc.
- The `BANDNAMES` field is optional; by default, bands are named according to their SDS name. The `MIN_VALUE`, `MAX_VALUE`, and `BACKGROUND_FILL` fields are also optional; by default, no background fill value is used in resampling.
- Only one input projection type is permitted in this file format. All 15 projection parameters must be specified.
- When inputting UTM data types, either the `UTM_ZONE` or first two projection parameters may be used to specify the zone. Valid `UTM_ZONE` values between -60 and 60 may be entered. If both the `UTM_ZONE` and first two projection parameters are specified, then the `UTM_ZONE` is used to determine the zone.
- A `DATUM` may be specified for input data. Valid values are `NAD27`, `NAD83`, `WGS66`, `WGS72`, `WGS84`, and `NODATUM`. If not specified then `NODATUM` will be used and the first two values in the projection parameters must be entered to define the semi-major and semi-minor axes of the spheroid.

- The `BYTE_ORDER` parameter specifies the byte-ordering used for the raw binary product. Valid values are “`little_endian`” (Intel-based systems like Windows) and “`big_endian`” (Unix-based systems like Linux and Macintosh). The byte order is required since it is possible that the input file was processed on a system with a different byte-order, in which case the image would need to be byte-swapped upon ingest. As of MRT 4.0, two-byte and four-byte data types are stored in machine-dependent byte order. All previous versions output raw binary products as big-endian byte-order.
- The `COORDINATE_ORIGIN` is an optional comment field that is not used by the MRT on input, but is written on output for information purposes only. It specifies the location of the coordinate origin as one of the four corners (`UL`, `UR`, `LL`, `LR`). The resampling executable assumes the coordinate origin is `UL` for raw binary data.
- There are two sets of corner coordinate fields. The `CORNER_XY` coordinates (in projection units) are optional comment fields. These fields are not used by the MRT on input, but are automatically generated on output for informational purposes only. The `CORNER_LATLON` coordinates are required (latitude and longitude in decimal degrees).
- Valid `DATA_TYPE` values are `INT8`, `UINT8`, `INT16`, `UINT16`, `INT32`, `UINT32`, and `FLOAT32`.

Appendix C: Projection Parameters

Projection Parameters 1-8

Code & Projection Id	Array Element							
	1	2	3	4	5	6	7	8
0 Geographic								
1 U T M	Lon/Z	Lat/Z						
2 State Plane								
3 Albers Equal Area	SMajor	SMinor	STDPR1	STDPR2	CentMer	OriginLat	FE	FN
4 Lambert Conformal C	SMajor	SMinor	STDPR1	STDPR2	CentMer	OriginLat	FE	FN
5 Mercator	SMajor	SMinor			CentMer	TrueScale	FE	FN
6 Polar Stereographic	SMajor	SMinor			LongPol	TrueScale	FE	FN
7 Polyconic	SMajor	SMinor			CentMer	OriginLat	FE	FN
8 Equid. Conic A	SMajor	SMinor	STDPAR		CentMer	OriginLat	FE	FN
Equid. Conic B	SMajor	SMinor	STDPR1	STDPR2	CentMer	OriginLat	FE	FN
9 Transverse Mercator	SMajor	SMinor	Factor		CentMer	OriginLat	FE	FN
10 Stereographic	Sphere				CentLon	CenterLat	FE	FN
11 Lambert Azimuthal	Sphere				CentLon	CenterLat	FE	FN
12 Azimuthal	Sphere				CentLon	CenterLat	FE	FN
13 Gnomonic	Sphere				CentLon	CenterLat	FE	FN
14 Orthographic	Sphere				CentLon	CenterLat	FE	FN
15 Gen. Vert. Near Per	Sphere		Height		CentLon	CenterLat	FE	FN
16 Sinusoidal	Sphere				CentMer		FE	FN
17 Equirectangular	Sphere				CentMer	TrueScale	FE	FN
18 Miller Cylindrical	Sphere				CentMer		FE	FN
19 Van der Grinten	Sphere				CentMer	OriginLat	FE	FN

```

20 Hotin Oblique Merc A |SMajor|SMinor|Factor|          |          |OriginLat|FE|FN|
    Hotin Oblique Merc B |SMajor|SMinor|Factor|AziAng|AzmthPt|OriginLat|FE|FN|
21 Robinson            |Sphere|          |          |          |CentMer|          |FE|FN|
22 Space Oblique Merc A |SMajor|SMinor|          |IncAng|AscLong|          |FE|FN|
    Space Oblique Merc B |SMajor|SMinor|Satnum|Path  |          |          |FE|FN|
23 Alaska Conformal    |SMajor|SMinor|          |          |          |          |FE|FN|
24 Interrupted Goode    |Sphere|          |          |          |          |          |          |
25 Mollweide            |Sphere|          |          |          |CentMer|          |FE|FN|
26 Interrupt Mollweide |Sphere|          |          |          |          |          |          |
27 Hammer              |Sphere|          |          |          |CentMer|          |FE|FN|
28 Wagner IV           |Sphere|          |          |          |CentMer|          |FE|FN|
29 Wagner VII          |Sphere|          |          |          |CentMer|          |FE|FN|
30 Oblated Equal Area   |Sphere|          |Shapem|Shapen|CentLon|CenterLat|FE|FN|
31 Integerized Sinusoid |Sphere|          |          |          |CentMer|          |FE|FN|

```

Projection Parameters 9-15

Code & Projection Id	Array Element						
	9	10	11	12	13	14	15
0 Geographic							
1 U T M							
2 State Plane							
3 Albers Equal Area							
4 Lambert Conformal C							
5 Mercator							
6 Polar Stereographic							
7 Polyconic							

8	Equid. Conic A	zero						
	Equid. Conic B	one						
9	Transverse Mercator							
10	Stereographic							
11	Lambert Azimuthal							
12	Azimuthal							
13	Gnomonic							
14	Orthographic							
15	Gen. Vert. Near Per							
16	Sinusoidal							
17	Equirectangular							
18	Miller Cylindrical							
19	Van der Grinten							
20	Hotin Oblique Merc A	Long1 Lat1 Long2 Lat2 zero						
	Hotin Oblique Merc B					one		
21	Robinson							
22	Space Oblique Merc A	PSRev LRat PFlag	zero					
	Space Oblique Merc B					one		
23	Alaska Conformal							
24	Interrupted Goode							
25	Mollweide							
26	Interrupt Mollweide							
27	Hammer							
28	Wagner IV							
29	Wagner VII							
30	Oblated Equal Area	Angle						
31	Integerized Sinusoidal	NZone	RFlag					

Notes

- Lon/Z: Longitude of any point in the UTM zone or zero. If zero, a zone code must be specified.
- Lat/Z: Latitude of any point in the UTM zone or zero. If zero, a zone code must be specified.
- SMajor: Semi-major axis of ellipsoid. If zero, Clarke 1866 in meters is assumed.
- SMinor: If zero, a spherical form is assumed, eccentricity squared of the ellipsoid if less than one, or if greater than one, the semi-minor axis of ellipsoid.
- Sphere: Radius of reference sphere. If zero, 6370997 meters is used.
- STDPAR: Latitude of the standard parallel
- STDPR1: Latitude of the first standard parallel
- STDPR2: Latitude of the second standard parallel
- CentMer: Longitude of the central meridian
- OriginLat: Latitude of the projection origin
- FE: False easting in the same units as the semi-major axis
- FN: False northing in the same units as the semi-major axis
- TrueScale: Latitude of true scale
- LongPol: Longitude down below pole of map
- Factor: Scale factor at central meridian (Transverse Mercator) or center of projection (Hotine Oblique Mercator)
- CentLon: Longitude of center of projection
- CenterLat: Latitude of center of projection
- Height: Height of perspective point
- Long1: Longitude of first point on center line (Hotine Oblique , format A)
- Long2: Longitude of second point on center line (Hotine Oblique , format A)
- Lat1: Latitude of first point on center line (Hotine Oblique , format A)

- Lat2: Latitude of second point on center line (Hotine Oblique , format A)
- AziAng: Azimuth angle east of north of center line (Hotine Oblique , format B)
- AzmthPt: Longitude of point on central meridian where azimuth occurs Hotine Oblique Mercator, format B)
- IncAng: Inclination of orbit at ascending node, counter-clockwise equator (SOM, format A)
- AscLong: Longitude of ascending orbit at equator (SOM, format A)
- PSRev: Period of satellite revolution in minutes (SOM, format A)
- LRat: Landsat ratio to compensate for confusion at northern end orbit (SOM, format A -- use 0.5201613)
- PFlag: End of path flag for Landsat: 0 = start of path, = end of path (SOM, format A)
- Satnum: Landsat Satellite Number (SOM, format B)
- Path: Landsat Path Number (Use WRS-1 for Landsat 1, 2 and 3 and -2 for Landsat 4, 5 and 6.) (SOM, format B)
- Shapem: Oblated Equal Area oval shape parameter m
- Shapen: Oblated Equal Area oval shape parameter n
- Angle: Oblated Equal Area oval rotation angle
- NZone: Number of equally spaced latitudinal zones (rows); must be 2 or larger and even
- RFlag: Right justify columns flag is used to indicate what to do with zones with an odd number of columns. If it has a value of 0 or 1 it indicates the extra column is on the right (zero) or left (one) of the projection y-axis. If the flag is set to 2 the number of columns are calculated so there are always an even number of column in each zone.
- Array elements 14 and 15 are set to zero
- All array elements with blank fields are set to zero
- All angles (latitudes, longitudes, azimuths, etc.) are entered in decimal degrees and the MRT then converts them to packed degrees/minutes/seconds (DDDMMMSSS.SS) format for the call to GCTP.