NAME

mbimagelist − parses recursive imagelist structures, performing one or more tasks on the image files referenced in the imagelist(s). The default action is to print out out the complete list of image file paths; other possible tasks include also printing out associated timestamps and camera settings, and printing out processing parameter values embedded in the imagelist structure.

VERSION

Version 5.0

SYNOPSIS

mbimagelist
[
--absolute {**-A**}
--copy=*directory* {**-C***directory*}
--copyhere
--files {**-F**}
--help {**-H**}
--input=*file* {**-I***file*}
--left {**-L**}
--parameters {**-P**}
--right {**-RfP}**
--settings {-S}
--single
--verbose {-V}
]

DESCRIPTION

MBimagelist is a utility for parsing imagelist files. Imagelist files, or lists of seafloor photographic images, are used by a number of **MB-System** programs dealing with seafloor photography. These lists may contain references to other imagelists, making them recursive. By default, the program **mbimagelist** outputs each image filename encountered as it parses through the input imagelist tree. The output paths can be relative to the current working directory or absolute.

Imagelist files can also contain processing parameters associated with the images, including specifying a file containing the calibration parameters for the camera rig, specifying a file containing the trajectory (navigation and attitude) of the camera rig as images were collected, specifying a file with an image correction lookup table, and several other parameters. All of these parameters correspond to command line options for **mbphotomosaic** that can also be specified.

This program can be used in shellscripts to read imagelists in the same fashion as **MB-System** programs like **mbphotomosaic** and **mbphotocorrect**. This program can also be used to check and debug complex recursive imagelist structures.

MB-SYSTEM AUTHORSHIP

David W. Caress
Monterey Bay Aquarium Research Institute
Dale N. Chayes
Center for Coastal and Ocean Mapping
University of New Hampshire
Christian do Santos Ferreira
MARUM - Center for Marine Environmental Sciences
University of Bremen

OPTIONS

--copy=*directory*

Causes the image files referenced in the input imagelist structure to be copied to the specified directory and creates a imagelist (names "imagelist.mb-1") in that same directory that references the copied image files.

--copyhere

Causes the image files referenced in the input imagelist structure to be copied to the current directory and creates a imagelist (names "imagelist.mb-1") that references the copied image files.

--report

Causes a listing to be printed of the unique imagelist files referenced through the recursive imagelist structure. Each line begins with the recursion level of that imagelist file within the overall structure followed by the full path of the imagelist file indented by a number of tabs equal to the recursion level.

--format

format

Sets the data format associated with the imagelist or swath data file specified with the **--input** option. By default, this program will attempt to determine the format from the input file suffix (e.g. a file ending in .mb57 has a format id of 57, and a file ending in .mb-1 has a format id of −1). A imagelist has a format id of −1.

--help This "help" flag cause the program to print out a description of its operation and then exit immediately.

--input *FILE*

Sets the input filename. If *format* > 0 (set with the **−f** option) then the swath data filename specified by *infile* is output along with its format and a file weight of 1.0. If *format* < 0, then *infile* is treated as a imagelist file containing a list of the input swath sonar data files to be processed and their formats. The program will parse the imagelist (recursively, if necessary) and output each swath filename and the associated format and file weight.

--make-ancilliary

This argument causes **MBimagelist** to generate three types of ancillary data files ("inf", "fbt", and "fnv"). In all cases, the ancillary filenames are just the original filename with ".inf", ".fbt", or ".fnv" appended on the end. **MB-System** mak es use of ancillary data files in a number of instances. The most prominent ancillary files are metadata or "inf" files (created from the output of **mbinfo**). Programs such as**mbgrid** and **mbm_plot** try to check "inf" files to see if the corresponding data files include data within desired areas. Additional ancillary files are used to speed plotting and gridding functions. The "fast bath" or "fbt" files are generated by copying the swath bathymetry to a sparse, quickly read format (format 71). The "fast nav" or "fnv" files are just ASCII lists of navigation generated using **mblist** with a **--update-ancilliary***tMXYHSc* option. Programs such as **mbgrid**, **mbswath**, and **mbcontour** will try to read "fbt" and "fnv" files instead of the full data files whenever only bathymetry or navigation information are required.

--update-ancilliary

This argument causes **MBimagelist** to generate the three ancillary data files ("inf", "fbt", and "fnv") if these files don’t already exist or are out of date.

--processed

Normally, **mbimagelist** allows $PROCESSED and $RAW tags within the imagelist files to determine whether processed file names are reported when available ($PROCESSED) or only raw file names are reported ($RAW). The **--processed** option forces **mbimagelist** to output processed file names when they exist.

--problem

This option causes the program to check each data file for the existence of any ancillary files referenced in its mbprocess parameter file (if the parameter file exists). The relevant ancillary files include edit save files generated by **mbedit** or **mbclean**, navigation files generated by **mbnavedit** or

mbnavadjust, tide files, and svp files. An error message is output for each missing ancillary file.

--bounds

W/E/S/N

The bounds of the desired area are set in longitude and latitude using w=west, e=east, s=south, and n=north. This option causes the program to check each data file with an "inf" file for overlap with the desired bounds, and only report those files with data in the desired area (or no "inf" file to check). This behavior mimics that of **mbgrid**, allowing users to check what data files will contribute to gridding some particular area.

--status

This option causes **mbimagelist** to report the status of the files it lists, including whether the file is up to date or needs reprocessing, and if the file is locked. **MBprocess** sets locks while operating on a swath file to prevent other instances of **mbprocess** from simultaneously operating on that same file. Locking consists of creating a small text file named by appending ".lck" to the swath filename; while this file exists other programs will not modify the locked file. The locking program deletes the lock file when it is done. Orphaned lock files may be left if **mbprocess** crashes or is interrupted. These will prevent reprocessing by **mbprocess**, but can be both detected and removed using **mbimagelist**.

--raw Normally, **mbimagelist** allows $PROCESSED and $RAW tags within the imagelist files to determine whether processed file names are reported when available ($PROCESSED) or only (raw) unprocessed file names are reported ($RAW). The **--raw** option forces **mbimagelist** to only output raw file names.

--verbose

Normally, **mbimagelist** only prints out the filenames and formats. If the **--verbose** flag is given, then **mbinfo** works in a "verbose" mode and outputs the program version being used.

--unlock

This option causes **mbimagelist** to remove any processing locks on files it parses. **MBprocess** and other programs may set locks while operating on a swath file to prevent other programs from simultaneously operating on that same file.The consists of creating a small text file named by appending ".lck" to the swath filename; while this file exists other programs will not modify the locked file. The locking program deletes the lock file when it is done. Orphaned lock files may be left if **MB-System** programs crash or are interrupted. These can be detected using the **--status** option of **mbimagelist**.

--imagelistp

The **--imagelistp** option causes the program to generate a imagelist file that will first set a $PROCESSED flag and then reference the input file specified using the **--input**=*FILE* option. The output imagelist is named by adding a "p.mb-1" suffix to the root of the input file (the root is the portion before any **MB-System** suffix).

By default, the input is assumed to be a imagelist named imagelist.mb-1, resulting in an output imagelist named imagelistp.mb-1 with the following contents:

```
$PROCESSED
imagelist.mb-1 −1
```

If the input file is specified as a imagelist like imagelist_sslo.mb-1, then the output imagelist imagelist_sslop.mb-1 will have the following contents:

```
$PROCESSED
imagelist_sslo.mb-1 −1
```

If the input file is specified as a swath file like 20050916122920.mb57, then the output imagelist 20050916122920p.mb-1 will have the following contents:

```
$PROCESSED
20050916122920.mb57 57
```

EXAMPLES

Suppose we have two swath data files from an EM3000 multibeam and another two from an Hydrosweep MD multibeam. We might construct two imagelist files. For the EM3000 we might have a file imagelist_em3000.mb-1 containing:

```
0004_20010705_165004_raw.mb57 57
0005_20010705_172010_raw.mb57 57
```

For the Hydrosweep MD data we might have a file imagelist_hsmd.mb-1 containing:

```
al10107051649.mb102 102
al10107051719.mb102 102
```

Further suppose that we have found it necessary to edit the bathymetry in 0005_20010705_172010_raw.mb57 and al10107051719.mb102 using **mbedit**, and that **mbprocess** has been run on both files to generate processed files called 0005_20010705_172010_rawp.mb57 and al10107051719p.mb102.

If we run:

```
mbimagelist --input=imagelist_em3000.mb-1
```

the output is:

```
0004_20010705_165004_raw.mb57 57 1.000000
0005_20010705_172010_raw.mb57 57 1.000000
```

Here the file name is followed by the format and then by a third column containing the default file weight of 1.0.

Similarly, if we run:

```
mbimagelist --input=imagelist_hsmd.mb-1
```

the output is:

```
al10107051649.mb102 102 1.000000
al10107051719.mb102 102 1.000000
```

If we insert a line

```
$PROCESSED
```

at the top of both imagelist_hsmd.mb-1 and imagelist_em3000.mb-1, then the output of **mbimagelist** changes so that:

```
mbimagelist --input=imagelist_em3000.mb-1
```

yields:

```
0004_20010705_165004_raw.mb57 57 1.000000
0005_20010705_172010_rawp.mb57 57 1.000000 and:
mbimagelist --input=imagelist_hsmd.mb-1
```

yields:

```
al10107051649.mb102 102 1.000000
al10107051719p.mb102 102 1.000000
```

Now suppose we create a imagelist file called imagelist_all.mb-1 that refers to the two imagelists shown above (without the $PROCESSED tags). If the contents of imagelist_all.mb-1 are:

```
imagelist_em3000.mb-1 −1 100.0
imagelist_hsmd.mb-1  −1  1.0
```

where we have specified different file weights for the two imagelists, then:

```
mbimagelist --input=imagelist_all.mb-1
```

yields:

```
0004_20010705_165004_raw.mb57 57 100.000000
0005_20010705_172010_raw.mb57 57 100.000000
al10107051649.mb102 102 1.000000
al10107051719.mb102 102 1.000000
```

Now, if we use the **--processed** option to force **mbimagelist** to output processed data file names when possible, then:

```
mbimagelist --input=imagelist_all.mb-1 --processed
```

yields:

```
0004_20010705_165004_raw.mb57 57 100.000000
0005_20010705_172010_rawp.mb57 57 100.000000
al10107051649.mb102 102 1.000000
al10107051719p.mb102 102 1.000000
```

To demonstrate the imagelist file listing function, consider the imagelist file named imagelist.mb-1 that is located at the top of MBARI's shipboard swath mapping database structure. This file references imagelists under directories for each of the institutions that we have sourced survey data from (e.g. CCOM, GEOMAR, IFREMER, etc.), and each of those imagelists reference imagelist files in directories for individual surveys or expedition legs, which in turn reference swath files for those surveys (or in some cases reference more imagelists if the expedition leg is organized into multiple surveys). We use the **--report** option to obtain the following listing (which actually runs a lot longer than shown here):

yields:

```
<00> imagelist.mb-1
<01>    CCOM/imagelist.mb-1
<02>            CCOM/NR07-1/imagelist.mb-1
<01>    GEOMAR/imagelist.mb-1
<02>            GEOMAR/SONNE100/imagelist.mb-1
<02>            GEOMAR/SONNE47/imagelist.mb-1
<02>            GEOMAR/SO108/imagelist.mb-1
<02>            GEOMAR/GEOMETEP/imagelist.mb-1
<02>            GEOMAR/SO83/imagelist.mb-1
<02>            GEOMAR/SO92/imagelist.mb-1
<02>            GEOMAR/SO99/imagelist.mb-1
<02>            GEOMAR/SO109-1/imagelist.mb-1
<02>            GEOMAR/SO109-2/imagelist.mb-1
<02>            GEOMAR/SO111/imagelist.mb-1
<02>            GEOMAR/SO112/imagelist.mb-1
<02>            GEOMAR/SO141/imagelist.mb-1
<02>            GEOMAR/SO142/imagelist.mb-1
<01>    IFREMER/imagelist.mb-1
<02>            IFREMER/CHARCOT/imagelist.mb-1
<02>            IFREMER/FOUNDATION/imagelist_mb71.mb-1
<02>            IFREMER/GEOMETEP4/imagelist.mb-1
<02>            IFREMER/MANZPA/imagelist.mb-1
<02>            IFREMER/NOUPA/imagelist.mb-1
<02>            IFREMER/OLIPAC/imagelist.mb-1
<02>            IFREMER/PAPNOU87/imagelist.mb-1
<02>            IFREMER/PAPNOU99/imagelist.mb-1
<02>            IFREMER/POLYNAUT/imagelist.mb-1
<02>            IFREMER/SEAPOS/imagelist.mb-1
<02>            IFREMER/ZEPOLYF1/imagelist.mb-1
<02>            IFREMER/ZEPOLYF2/imagelist.mb-1
<02>            IFREMER/ZEPOLYF3/imagelist.mb-1
<02>            IFREMER/BENTHAUS/imagelist.mb-1
<02>            IFREMER/SISMITA/imagelist.mb-1
<02>            IFREMER/ACT/imagelist.mb-1
```

SEE ALSO

mbsystem(1)

BUGS

No true bugs here, only distantly related arthropods... Yum. Seriously, it would be better if the copy function preserved the modification times of the copied swath files and ancillary files. Copying of processed files should also be an option.