NAME

mbdatalist − parses recursive datalist files and outputs the complete list of data files, formats, and file weights.

VERSION

Version 5.0

SYNOPSIS

mbdatalist [**−C −F***format* **−I***filename* **−N −O −P −Q −R −S −U −Y −Z −V −H**]

DESCRIPTION

MBdatalist is a utility for parsing datalist files. Datalist files, or lists of swath data files and their format ids, are used by a number of **MB-System** programs. These lists may contain references to other datalists, making them recursive. See the **MB-System** manual page for details on the format and structure of datalists. The program **mbdatalist** outputs each swath data filename, format id, and file weight encountered as it descends through the input datalist tree. If a swath data file rather than a datalist is provided as input, the same swath data filename and format will be the sole output.

This program can be used in shellscripts to read datalists in the same fashion as **MB-System** programs like **mbgrid** and **mbprocess**. This program can also be used to check and debug complex recursive datalist structures.

The program **mbprocess** operates on "raw" swath data files, producing a "processed" swath data file (see the **mbprocess** man page for explanation). The **MB-System** algorithm for reading datalists will, if a flag is set, replace a swath file name with the associated "processed" file name when that "processed" file exists. This flag may be set by embedding "$PROCESSED" as a line in a datalist or it may be set first by the calling program. The flag may also be set to preclude reporting "processed" file names (embedding "$RAW" in a datalist accomplishes this). When setting this flag within datalists, the first encounter of a $PROCESSED or $RAW tag will prevail over later instances of either tag. The **−P** and **−U** options force **mbdatalist** to output processed file names when they exist (**−P**) or to only output unprocessed (raw) file names (**−U**).

Programs such as **mbgrid** try to check statistics or "inf" files to see if the corresponding data files include data within the specified geographic bounds. Other programs look for "fast bathymetry" or "fast navigation" ("fbt" or "fnv") files in order to read the data more quickly. The **−N** option causes **mbdatalist** to create these three types of ancillary files for each swath data file. The **−O** option causes **mbdatalist** to create the "inf", "fbt", and "fnv" files only when they don't already exist or are out of date (older than the data file).

Datalists may also contain a third value, called the grid weight, which is used by **mbgrid** to priortize data. The larger the grid weight, the more importance **mbgrid** attaches to the related bathymetry data. Grid weights can be applied to datalist entries which are themselves datalist files, causing these weights to be associated with all of files referenced therein. However, the default behavior is for any grid weight in a particular datalist entry to override values derived from higher levels in the recursive structure. This behavior can be reversed if a $NOLOCALWEIGHT tag is placed in the datalist, or in a datalist higher up in the structure. See the **MB-System** manual page for a more complete description.

The **−R***w/e/s/n* 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.

The **−Q** option causes the program to check each data file for the existence of any ancillary files (e.g. navigation files, edit save files, etc.) referenced in its mbprocess parameter file (if the parameter file exists). The program will list any problem found with the processing parameters, and will also list any data problem

noted in the "inf" files. The possible data problems include:
No survey data found
Zero longitude or latitude in survey data
Instantaneous speed exceeds 25 km/hr
Average speed exceeds 25 km/hr
Sounding depth exceeds 11000 m
Unsupported Simrad datagram

The **−Z** option causes the program to generate a datalist file named "datalistp.mb-1" and then exit. This datalist has the following form:
$PROCESSED
datalist.mb-1 −1
This file is a commonly used convenience because it allows users to easily reference the swath files listed (directly or recursively) through the datalist "datalist.mb-1" with the $PROCESSED flag on. So, in order to grid the processed bathymetry rather than the raw bathymetry, run **mbgrid** with "datalistp.mb-1" as the input rather than "datalist.mb-1".

The **−S** option causes **mbdatalist** 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. This allows one to run **mbprocess** multiple times simultaneously on a single datalist, either on a single multiprocessor machine or on multiple computers mounting the same filesystem. 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 **mbprocess** crashes or is interrupted. These will prevent reprocessing by **mbprocess**, but can be both detected with the **−S** option and removed using the **−Y** option.

Finally, this program can be used to copy the swath files referenced in a datalist structure to a single directory and to create a datalist there (names "datalist.mb-1") that references those swath files. This is accomplished using the *-C* option. The *-C* copy function will not be done if the *-N*, *-O*, or *-Q* options are specified, but is compatible with the *-P*, *-R*, and *-U* options.

AUTHORSHIP

David W. Caress (caress@mbari.org)
Monterey Bay Aquarium Research Institute
Dale N. Chayes (dale@ldeo.columbia.edu)
Lamont-Doherty Earth Observatory

OPTIONS

−C

Causes the swath files referenced in the input datalist structure to be copied to the current directory and creates a datalist (names "datalist.mb-1") that references the copied swath files. The copy function will not be done if the *-N*, *-O*, or *-Q* options are specified. If the *-P*, *-R*, and *-U* options are specified these functions will modify which swath files are copied. Any ancilliary files (e.g. *inf metadata files) will also be copied, but processed data files derived from the target copied files will not be copied.

−F *format*
Sets the data format associated with the datalist or swath data file specified with the **−I** 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 datalist has a format id of −1.

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

–I *filename*
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 datalist file containing a list of the input swath sonar data files to be processed and their formats. The program will parse the datalist (recursively, if necessary) and output each swath filename and the associated format and file weight.

–N This argument causes **MBdatalist** 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** makes 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 **–O***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.

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

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

–Q 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.

–R *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.

–S This option causes **mbdatalist** 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 **mbdatalist**.

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

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

–Y This option causes **mbdatalist** 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 **–S** option of **mbdatalist**.

–Z The **–Z** option causes the program to generate a datalist file that will first set a $PROCESSED flag and then reference the input file specified using the **–I***filename* option. The output datalist 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 datalist named datalist.mb-1, resulting in an output datalist named datalistp.mb-1 with the following contents:
    $PROCESSED
    datalist.mb-1 –1
If the input file is specified as a datalist like datalist_sslo.mb-1, then the output datalist datalist_sslop.mb-1 will have the following contents:
    $PROCESSED
    datalist_sslo.mb-1 –1
If the input file is specified as a swath file like 20050916122920.mb57, then the output datalist 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 datalist files. For the EM3000 we might have a file dlst_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 dlst_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:
    mbdatalist –I dlst_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:
    mbdatalist –I dlst_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 dlst_hsmd.mb-1 and dlst_em3000.mb-1, then the output of **mbdatalist** changes so that:
    mbdatalist –I dlst_em3000.mb-1

yields:

    0004_20010705_165004_raw.mb57 57 1.000000
    0005_20010705_172010_rawp.mb57 57 1.000000 and:
    mbdatalist –I dlst_hsmd.mb-1

yields:

    al10107051649.mb102 102 1.000000
    al10107051719p.mb102 102 1.000000

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

    dlst_em3000.mb-1 –1 100.0
    dlst_hsmd.mb-1   –1   1.0

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

    mbdatalist –I dlst_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 **–P** option to force **mbdatalist** to output processed data file names when possible, then:

    mbdatalist –I dlst_all.mb-1 –P

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

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 ancilliary files. Copying of processed files should also be an option.