NAME

mbpreprocess − performs preprocessing of swath sonar data as part of setting up an MB-System processing structure for a dataset.

VERSION

Version 5.0

SYNOPSIS

mbpreprocess [
--verbose
--help
--input=*FILE*
--format=*FORMATID*
--output-directory=*DIRECTORY*
--platform-file=*FILE*
--platform-target-sensor=*RECORDID*
--output-sensor-fnv
--skip-existing
--nav-file=*FILE*
--nav-file-format=*FORMATID*
--nav-async=*RECORDTYPE*
--nav-sensor=*SENSORID*
--sensordepth-file=*FILE*
--sensordepth-file-format=*FORMATID*
--sensordepth-async=*RECORDTYPE*
--sensordepth-sensor=*SENSORID*
--heading-file=*FILE*
--heading-file-format=*FORMATID*
--heading-async=*RECORDTYPE*
--heading-sensor=*SENSORID*
--altitude-file=*FILE*
--altitude-file-format=*FORMATID*
--altitude-async=*RECORDTYPE*
--altitude-sensor=*SENSORID*
--attitude-file=*FILE*
--attitude-file-format=*FORMATID*
--attitude-async=*RECORDTYPE*
--attitude-sensor=*SENSORID*
--soundspeed-file=*FILE*
--soundspeed-file-format=*FORMATID*
--soundspeed-async=*RECORDTYPE*
--soundspeed-sensor=*SENSORID*
--time-latency-file=*FILE*
--time-latency-file-format=*FORMATID*
--time-latency-constant=*SECONDS*
--time-latency-apply-nav
--time-latency-apply-sensordepth
--time-latency-apply-heading
--time-latency-apply-attitude
--time-latency-apply-altitude
--time-latency-apply-all-ancilliary
--time-latency-apply-survey
--time-latency-apply-all
--filter=*SECONDS*

--filter-apply-nav
--filter-apply-sensordepth
--filter-apply-heading
--filter-apply-attitude
--filter-apply-altitude
--filter-apply-all-ancilliary
--recalculate-bathymetry
--no-change-survey
--multibeam-sidescan-source=*SOURCE*
--sounding-amplitude-filter=*THRESHOLD*
--sounding-altitude-filter=*THRESHOLD*
--ignore-water-column
--head1-offsets=*X/Y/Z/HEADING/ROLL/PITCH*
--head2-offsets=*X/Y/Z/HEADING/ROLL/PITCH*
--kluge-time-jumps=*SECONDS*
--kluge-fix-7k-timestamps=*TARGETOFFSET*
--kluge-ancilliary-time-jumps=*SECONDS*
--kluge-mbaripressure-time-jumps=*THRESHOLD*
--kluge-beam-tweak=*FACTOR*
--kluge-soundspeed-tweak=*FACTOR*
--kluge-zero-attitude-correction
--kluge-zero-alongtrack-angles
--kluge-fix-wissl-timestamps
--kluge-auv-sentry-sensordepth
--kluge-ignore-snippets
--kluge-sonardepth-from-heave
--kluge-flipsign-roll
--kluge-flipsign-pitch
]

DESCRIPTION

MBpreprocess handles preprocessing of swath sonar data as part of setting up an MB-System processing structure for a dataset. This program replaces the several format-specific preprocessing programs found in MB-System version 5 releases with a single program for version 6. In many cases the preprocessing consists only of creating the ancillary files parallel to the raw files, but in others the data must be translated to different formats. The proprocessing step can also be used to merge navigation, attitude, sound speed, or other ancillary data with the survey data.

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

--help This option causes **mbconfig** to print out its usage options and exit.

--verbose

By default **mbpreprocess** outputs minimal information to the shell. This option causes the program to indicate it’s progress as it runs.

--input=*FILE*

Swath data file from which the input data will be read, or a datalist file containing a list of input swath data files and/or other datalist files. If *FILE* is a datalist file, then mbpreprocess will attempt to preprocess all data files identified by recursively reading the datalist structure referenced in *FILE*.

--format=*FORMATID*

Sets the **MBIO** integer format identifier for the input file *FILE* specified with the **−−input** option. By default, **mbpreprocess** infers the format from the "*.mbXX" **MB−System** suffix convention.

--output-directory=*DIRECTORY*

By default the output files written by **mbpreprocess** are located in the same directory as the input swath files. This options allows the user to specify a different directory into which the output files are written.

--platform-file=*FILE*

Causes **mbpreprocess** to input and utilize an **MB-System** platform file, which provides a complete description of the relative geometry of sensors on a survey platform. Platform files can be constructed using the program **mbmakeplatform**. Most datasets do not require the use of a platform file, and only a few data formats are supported with preprocessing functionality utilizing platform files (e.g. the Reson 7k format 88, the Teledyne 7k3 format 89, the 3D at Depth lidar formats 232 and 233). Platform files are needed only when survey data have been collected without integrating the asynchronous navigation and attitude data and when the positional and angular offsets between the survey sensor (e.g. multibeam sonar, lidar, stereo camera rig) have not been specified in the data stream. Most often this situation is associated with survey systems on submerged platforms like autonomous underwater vehicles (AUVs) or remotely operated vehicles (ROVs). In cases where the sonar control and data logging software have a full platform description and apply that to fully correct the survey data in real time, no platform file is needed for the preprocessing stage in **MB-System**. Specifically, all datasets collected using Kongsberg multibeam sonars do not require use of a platform file (the only exception to this would be if the positional offsets between sensors were incorrectly specified in the multibeam configuration).

--platform-target-sensor=*SENSORID*

If a platform file has been specified, this option defines the id of the sensor description associated with the survey data being preprocessed. This determines how lever arm calculations are done. For instance, if sensor 0 is the source of attitude and lateral position data (e.g. an inertial navigation system), sensor 1 is the source of vertical position data (e.g. a pressure sensor), and sensor 2 is a multibeam sonar for which the bathymetry data are not initially motion corrected, then one should use **--platform-target-sensor**=*2* so that the lever arm calculations produce the position and attitude of the multibeam sonar. The multibeam bathymetry are then calculated using the corrected position and attitude data. This option is necessary because in general a platform may have multiple survey sensors (e.g. multibeam sonar, lidar, stereo camera rig, subbottom profiler, sidescan sonar), each of which will have a separate data processing workflow.

--output-sensor-fnv

Causes **mbpreprocess** to output "fast navigation" or *.fnv files for each of the sensors described in the platform file specified with the **--platform-file** option. These *.fnv files will be corrected for the lever arms specified by the positional and angular offsets defined in the platform file.

--skip-existing

This option causes **mbpreprocess** to skip preprocessing when the output files already exist and are up to date relative to the inputs.

−−nav-file=*filename*

Specifies an external time series file from which to merge sonar position (navigation), replacing navigation data included in the data stream. By default **mbpreprocess** attempts to use navigation values included in the input data.

−−**nav-file-format**=*format_id*

Specifies the format of an external navigation time series file from which position values are derived (as defined with −−**nav-file**). Options for the *format_id* value are:

1: ASCCI text file with lines of the form:
time_d longitude latitude speed
where time_d is time in decimal epoch seconds (seconds since 1970), longitude and latitude are in decimal degrees, and speed is in km/hour (and is optional). South latitudes are negative. Longitude may be defined on either the -180.0 to +180.0 or 0.0 to 360.0 domains.

2: ASCCI text file with lines of the form:
year month day hour minute second longitude latitude
where year, month, day, hour, and minute values are integers, the second value is decimal, and longitude and latitude are in decimal degrees. South latitudes are negative. Longitude may be defined on either the -180.0 to +180.0 or 0.0 to 360.0 domains.

3: ASCCI text file with lines of the form:
year julian_day hour minute second longitude latitude
where year, julian_day, hour, and minute values are integers, the second value is decimal, and longitude and latitude are in decimal degrees. South latitudes are negative. Longitude may be defined on either the -180.0 to +180.0 or 0.0 to 360.0 domains.

4: ASCCI text file with lines of the form:
year julian_day day_minute second longitude latitude
where year, julian_day, and day_minute values are integers, the second value is decimal, and longitude and latitude are in decimal degrees. South latitudes are negative. Longitude may be defined on either the -180.0 to +180.0 or 0.0 to 360.0 domains.

5: ASCCI text file in the 1990's era L-DEO processed nav format with lines of the form:
yy+jjjhhmmssNddmmmmmmEdddmmmmmm
where yy is the two digit year (after 1999 the "yy+" was replaced by a four digit year "yyyy"), jjj is the julian_day, hh is the hour, mm is minutes, and ss is seconds. The latitude is given as Nddmmmmmmand where N is 'N' for north and 'S' for south, dd are integer degrees, and mmmmmm is minutes * 10000. The longitude is given as Edddmmmmmmand where E is 'E' for east and 'W' for west, ddd are integer degrees, and mmmmmm is minutes * 10000.

6 or 7: NMEA 0183 position strings
Several NMEA and NMEA-like strings containing position are recognized, and can be parsed with and without line break characters. These strings include ZDA, GLL, GGA, DAT, and UNX.

8: Simrad 90 format navigation files with lines of the form:
ddmmyy hhmmssss ddmmmmmmmN dddmmmmmmmE
where dd is day of the month, mm is the month, yy is the two digit year, hh is the hour, mm is the minute, and ssss is seconds * 100. The latitude is given as ddmmmmmmmN where dd are integer degrees, mmmmmmm is minutes * 100000, and N is 'N' for north and 'S' for south latitude. The longitude is given as dddmmmmmmmE where ddd are integer degrees, mmmmmmm is minutes * 100000, and E is 'E' for east and 'W' for west longitude.

day_minute values are integers,
the second value is decimal, and longitude and latitude are in decimal

degrees. The longitude is given as
Edddmmmmmmand where E is 'E' for east and 'W' for west, ddd are integer
degrees, and mmmmm is minutes * 10000.
9: ASCCI text file with white-space delimited lines of the form:
yr mon day hour min sec time_d lon lat heading speed sensordepth
or
yr mon day hour min sec time_d lon lat heading speed sensordepth roll pitch heave
where yr is the four digit year, mon is the month, day is the day of the
month, min is the minute, second is the decimal seconds, time_d is time in
decimal epoch seconds (seconds since 1970), lon is the longitude
in decimal degrees, lat is the latitude in decimal degrees, heading is in
decimal degrees, speed is in km/hour, and sensordepth is in meters.
South latitudes are negative. Longitude may be defined on either the
-180.0 to +180.0 or 0.0 to 360.0 domains. If present, roll and pitch are in
decimal degrees and heave is in meters.
10: R2R (Rolling deck to Repository) navigation format with ASCCI text lines
of the form:
yyyy-mm-ddThh:mm:ss.sssZ lon lat quality nsat dilution height
where the lon and lat fields are in decimal degrees with south latitudes
and west longitudes negative, and the last four quantities relating to
GPS fix quality.

−−**nav-async**=*record_kind*

Specifies the type of data records from which position values are derived (as defined with −−**nav-file-format**). Options for the *record_kind* include:
MB_DATA_DATA: 1 (survey data)
MB_DATA_NAV: 12 (navigation data)
MB_DATA_NAV1: 29 (navigation data from navigation system 1)
MB_DATA_NAV2: 30 (navigation data from navigation system 2)
MB_DATA_NAV3: 31 (navigation data from navigation system 3)
What types of data records are present is format-dependent, as is the default choice of which record type is used as the navigation source. The program **mbinfo** can be used with the **-N** option to determine the numbers of different record types present in a data file.

--nav-sensor=*SENSORID*

If a platform file has been specified, this option defines the id of the sensor description associated with the navigation data being used, whether that derives from asynchronous records in the data stream or from an external navigation file. Note that if the external navigation has been calculated to define the location of the target survey sensor, then the sensor id should be the same as the survey sensor so that a null lever arm correction is made.

--sensordepth-file=*FILE*

Specifies an external time series file from which to merge sensor depth, replacing sensor depth data included in the data stream. By default **mbpreprocess** attempts to use sensor depth values included in the input data.

--sensordepth-file-format=*FORMATID*

Specifies the format of an external sensor depth time series file from which sensor depth values are derived (as defined with −−**sensordepth-file**). Options for the *format_id* value are:
1: ASCCI text file with lines of the form:
time_d sensordepth
where time_d is time in decimal epoch seconds (seconds since 1970), and
sensordepth values are in meters positive down.
2: ASCCI text file with lines of the form:
year month day hour minute second sensordepth
where year, month, day, hour, and minute values are integers, the second
value is decimal, and sensordepth values are in meters positive down.

3: ASCCI text file with lines of the form:
year julian_day hour minute second sensordepth
where year, julian_day, hour, and minute values are integers, the second value is decimal, and sensordepth values are in meters positive down.
4: ASCCI text file with lines of the form:
year julian_day day_minute second sensordepth
where year, julian_day, and day_minute values are integers, the second value is decimal, and sensordepth values are in meters positive down.
9: ASCCI text file with white-space delimited lines of the form:
yr mon day hour min sec time_d lon lat heading speed sensordepth
or
yr mon day hour min sec time_d lon lat heading speed sensordepth roll pitch heave
where yr is the four digit year, mon is the month, day is the day of the month, min is the minute, second is the decimal seconds, time_d is time in decimal epoch seconds (seconds since 1970), lon is the longitude in decimal degrees, lat is the latitude in decimal degrees, heading is in decimal degrees, speed is in km/hour, and sensordepth is in meters. South latitudes are negative. Longitude may be defined on either the -180.0 to +180.0 or 0.0 to 360.0 domains. If present, roll and pitch are in decimal degrees and heave is in meters.

--sensordepth-async=*RECORDTYPE*

Specifies the type of data records from which sensor depth values are derived (as defined with −−**sensordepth-file-format**). Options for the *record_kind* include:
MB_DATA_DATA: 1 (survey data)
MB_DATA_NAV: 12 (navigation data)
MB_DATA_HEIGHT 16 (sensor height data)
MB_DATA_NAV1: 29 (navigation data from navigation system 1)
MB_DATA_NAV2: 30 (navigation data from navigation system 2)
MB_DATA_NAV3: 31 (navigation data from navigation system 3)
MB_DATA_SONARDEPTH 59 (sensor depth data)
What types of data records are present is format-dependent, as is the default choice of which record type is used as the sensor depth source. The program **mbinfo** can be used with the **-N** option to determine the numbers of different record types present in a data file.

--sensordepth-sensor=*SENSORID*

If a platform file has been specified, this option defines the id of the sensor description associated with the sensor depth data being used, whether that derives from asynchronous records in the data stream or from an external navigation file. Note that if the external sensor depth data have been calculated to define the vertical location of the target survey sensor, then the sensor id should be the same as the survey sensor so that a null lever arm correction is made.

--heading-file=*FILE*

Specifies an external time series file from which to merge heading, replacing heading data included in the data stream. By default **mbpreprocess** attempts to use heading values included in the input data.

--heading-file-format=*FORMATID*

Specifies the format of an external heading time series file from which heading values are derived (as defined with −−**heading-file**). Options for the *format_id* value are:
1: ASCCI text file with lines of the form:
time_d heading
where time_d is time in decimal epoch seconds (seconds since 1970), and heading values are in degrees positive clockwise from forward.
2: ASCCI text file with lines of the form:
year month day hour minute second heading
where year, month, day, hour, and minute values are integers, the second

value is decimal, and heading values in degrees positive clockwise from forward.
3: ASCCI text file with lines of the form:
year julian_day hour minute second heading
where year, julian_day, hour, and minute values are integers, the second
value is decimal, and heading values in degrees positive clockwise from forward.
4: ASCCI text file with lines of the form:
year julian_day day_minute second heading
where year, julian_day, and day_minute values are integers, the second
value is decimal, and heading values in degrees positive clockwise from forward.
9: ASCCI text file with white-space delimited lines of the form:
yr mon day hour min sec time_d lon lat heading speed sensordepth
or
yr mon day hour min sec time_d lon lat heading speed sensordepth roll pitch heave
where yr is the four digit year, mon is the month, day is the day of the
month, min is the minute, second is the decimal seconds, time_d is time in
decimal epoch seconds (seconds since 1970), lon is the longitude
in decimal degrees, lat is the latitude in decimal degrees, heading is in
decimal degrees, speed is in km/hour, and heading is in meters.
South latitudes are negative. Longitude may be defined on either the
-180.0 to +180.0 or 0.0 to 360.0 domains. If present, roll and pitch are in
decimal degrees and heave is in meters.

--heading-async=*RECORDTYPE*

Specifies the type of data records from which heading values are derived (as defined with **−−heading-file-format**). Options for the *record_kind* include:
MB_DATA_DATA: 1 (survey data)
MB_DATA_NAV: 12 (navigation data)
MB_DATA_HEADING: 17 (heading data)
MB_DATA_NAV1: 29 (navigation data from navigation system 1)
MB_DATA_NAV2: 30 (navigation data from navigation system 2)
MB_DATA_NAV3: 31 (navigation data from navigation system 3)
What types of data records are present is format-dependent, as is the default choice of which record type is used as the heading source. The program **mbinfo** can be used with the **-N** option to determine the numbers of different record types present in a data file.

--heading-sensor=*SENSORID*

If a platform file has been specified, this option defines the id of the sensor description associated with the heading data being used, whether that derives from asynchronous records in the data stream or from an external navigation file.

--altitude-file=*FILE*

Specifies an external time series file from which to merge altitude data, replacing altitude data included in the data stream. By default **mbpreprocess** attempts to use altitude values included in the input data. Typically altitude data are critical only for processing sidescan data in the absence of swath bathymetry.

--altitude-file-format=*FORMATID*

Specifies the format of an external altitude time series file from which altitude values are derived (as defined with **−−altitude-file**). Options for the *format_id* value are:
1: ASCCI text file with lines of the form:
time_d altitude
where time_d is time in decimal epoch seconds (seconds since 1970), and
altitude values are in meters.
2: ASCCI text file with lines of the form:
year month day hour minute second altitude
where year, month, day, hour, and minute values are integers, the second
value is decimal, and altitude values are in meters.

3: ASCCI text file with lines of the form:
year julian_day hour minute second altitude
where year, julian_day, hour, and minute values are integers, the second value is decimal, and altitude values are in meters.
4: ASCCI text file with lines of the form:
year julian_day day_minute second altitude
where year, julian_day, and day_minute values are integers, the second value is decimal, and altitude values are in meters.

--altitude-async=*RECORDTYPE*

Specifies the type of data records from which heading values are derived (as defined with **−−heading-file-format**). Options for the *record_kind* include:
MB_DATA_DATA: 1 (survey data)
MB_DATA_NAV: 12 (navigation data)
MB_DATA_NAV1: 29 (navigation data from navigation system 1)
MB_DATA_NAV2: 30 (navigation data from navigation system 2)
MB_DATA_NAV3: 31 (navigation data from navigation system 3)
MB_DATA_ALTITUDE: 60 (altitude data)
What types of data records are present is format-dependent, as is the default choice of which record type is used as the altitude source. The program **mbinfo** can be used with the **-N** option to determine the numbers of different record types present in a data file.

--altitude-sensor=*SENSORID*

If a platform file has been specified, this option defines the id of the sensor description associated with the navigation data being used, whether that derives from asynchronous records in the data stream or from an external navigation file.

--attitude-file=*FILE*

Specifies an external time series file from which to merge attitude data (roll and pitch, often also heave), replacing attitude data included in the data stream. By default **mbpreprocess** attempts to use attitude values included in the input data.

--attitude-file-format=*FORMATID*

Specifies the format of an external attitude time series file from which attitude values are derived (as defined with **−−attitude-file**). Options for the *format_id* value are:
1: ASCCI text file with lines of the form:
time_d roll pitch heave
where time_d is time in decimal epoch seconds (seconds since 1970), and sensordepth values are in meters positive down.
2: ASCCI text file with lines of the form:
year month day hour minute second roll pitch heave
where year, month, day, hour, and minute values are integers, the second value is decimal, and sensordepth values are in meters positive down.
3: ASCCI text file with lines of the form:
year julian_day hour minute second roll pitch heave
where year, julian_day, hour, and minute values are integers, the second value is decimal, and sensordepth values are in meters positive down.
4: ASCCI text file with lines of the form:
year julian_day day_minute second roll pitch heave
where year, julian_day, and day_minute values are integers, the second value is decimal, and sensordepth values are in meters positive down.
9: ASCCI text file with white-space delimited lines of the form:
yr mon day hour min sec time_d lon lat heading speed sensordepth roll pitch heave
where yr is the four digit year, mon is the month, day is the day of the month, min is the minute, second is the decimal seconds, time_d is time in decimal epoch seconds (seconds since 1970), lon is the longitude in decimal degrees, lat is the latitude in decimal degrees, heading is in

decimal degrees, speed is in km/hour, and sensordepth is in meters.
South latitudes are negative. Longitude may be defined on either the
-180.0 to +180.0 or 0.0 to 360.0 domains. Roll and pitch are in
decimal degrees and heave is in meters.

--attitude-async=*RECORDTYPE*

Specifies the type of data records from which attitude values are derived (as defined with **−−attitude-file-format**). Options for the *record_kind* include:
MB_DATA_DATA: 1 (survey data)
MB_DATA_NAV: 12 (navigation data)
MB_DATA_NAV1: 29 (navigation data from navigation system 1)
MB_DATA_NAV2: 30 (navigation data from navigation system 2)
MB_DATA_NAV3: 31 (navigation data from navigation system 3)
MB_DATA_ATTITUDE: 18 (attitude data)
MB_DATA_ATTITUDE1: 56 (attitude data from attitude system 1)
MB_DATA_ATTITUDE2: 57 (attitude data from attitude system 2)
MB_DATA_ATTITUDE3: 58 (attitude data from attitude system 3) #.br What types of data records are present is format-dependent, as is the default choice of which record type is used as the attitude source. The program **mbinfo** can be used with the **-N** option to determine the numbers of different record types present in a data file.

--attitude-sensor=*SENSORID*

If a platform file has been specified, this option defines the id of the sensor description associated with the navigation data being used, whether that derives from asynchronous records in the data stream or from an external navigation file.

--soundspeed-file=*FILE*

Specifies an external time series file from which to merge sound speed data, replacing sound speed data included in the data stream. By default **mbpreprocess** attempts to use sound speed values included in the input data. The sound speed values represent the speed of sound at the survey sensor (i.e. the sonar arrays); typically these are input only when the preprocessing includes recalculation of multibeam bathymetry.

--soundspeed-file-format=*FORMATID*

Specifies the format of an external soundspeed time series file from which soundspeed values are derived (as defined with **−−soundspeed-file**). Options for the *format_id* value are:
1: ASCCI text file with lines of the form:
time_d soundspeed
where time_d is time in decimal epoch seconds (seconds since 1970), and
soundspeed values are in meters positive down.
2: ASCCI text file with lines of the form:
year month day hour minute second soundspeed
where year, month, day, hour, and minute values are integers, the second
value is decimal, and soundspeed values are in meters positive down.
3: ASCCI text file with lines of the form:
year julian_day hour minute second soundspeed
where year, julian_day, hour, and minute values are integers, the second
value is decimal, and soundspeed values are in meters positive down.
4: ASCCI text file with lines of the form:
year julian_day day_minute second soundspeed
where year, julian_day, and day_minute values are integers, the second
value is decimal, and soundspeed values are in meters positive down.

--soundspeed-async=*RECORDTYPE*

Specifies the type of data records from which sound speed values are derived (as defined with **−−soundspeed-file-format**). Options for the *record_kind* include:
MB_DATA_DATA: 1 (survey data)

MB_DATA_SSV: 19 (sound speed data)
MB_DATA_CTD: 34 (sound speed data) #.br What types of data records are present is format-dependent, as is the default choice of which record type is used as the sound speed source. The program **mbinfo** can be used with the **-N** option to determine the numbers of different record types present in a data file.

--soundspeed-sensor=*SENSORID*
If a platform file has been specified, this option defines the id of the sensor description associated with the navigation data being used, whether that derives from asynchronous records in the data stream or from an external navigation file.

--time-latency-file=*FILE*
Specifies an external time series file from which to read a time latency model to be applied to some or all of the timestamps in the data stream.

--time-latency-file-format=*FORMATID*
Specifies the format of an external time latency time series file from which time latency values are derived (as defined with **−−time-latency-file**). Options for the *format_id* value are:
1: ASCCI text file with lines of the form:
time_d timeshift
where time_d is time in decimal epoch seconds (seconds since 1970), and
timeshift values are in seconds.
2: ASCCI text file with lines of the form:
year month day hour minute second timeshift
where year, month, day, hour, and minute values are integers, the second
value is decimal, and timeshift values are in seconds.
3: ASCCI text file with lines of the form:
year julian_day hour minute second timeshift
where year, julian_day, hour, and minute values are integers, the second
value is decimal, and timeshift values are in seconds.
4: ASCCI text file with lines of the form:
year julian_day day_minute second timeshift
where year, julian_day, and day_minute values are integers, the second
value is decimal, and timeshift values are in seconds.

--time-latency-constant=*SECONDS*
Specifies a constant time latency value

--time-latency-apply-nav
Specifies that the time latency correction will be applied to the time stamps associated with the navigation data merged with the survey data.

--time-latency-apply-sensordepth
Specifies that the time latency correction will be applied to the time stamps associated with the sonar depth data merged with the survey data.

--time-latency-apply-heading
Specifies that the time latency correction will be applied to the time stamps associated with the heading data merged with the survey data.

--time-latency-apply-attitude
Specifies that the time latency correction will be applied to the time stamps associated with the attitude data merged with the survey data.

--time-latency-apply-altitude
Specifies that the time latency correction will be applied to the time stamps associated with the altitude data merged with the survey data.

--time-latency-apply-all-ancilliary
Specifies that the time latency correction will be applied to the time stamps associated with all of the ancilliary data merged with the survey data (i.e. the navigation, sensor depth, heading, attitude, and altitude data).

--time-latency-apply-survey
Specifies that the time latency correction will be applied to the time stamps associated with the survey data.

--time-latency-apply-all
Specifies that the time latency correction will be applied to all time stamps associated with survey data and ancilliary data.

--filter=*SECONDS*
Specifies the half width in seconds of a Gaussian smoothing filter that can be applied to ancilliary time series data prior to merging with the survey data. This includes the navigation, heading, sensor depth, attitude and altitude data.

--filter-apply-nav
Specifies that the smoothing filtering will be applied to the navigation data merged with the survey data.

--filter-apply-sensordepth
Specifies that the smoothing filtering will be applied to the sensor depth data merged with the survey data.

--filter-apply-heading
Specifies that the smoothing filtering will be applied to the heading data merged with the survey data.

--filter-apply-attitude
Specifies that the smoothing filtering will be applied to the attitude data merged with the survey data.

--filter-apply-altitude
Specifies that the smoothing filtering will be applied to the altitude data merged with the survey data.

--filter-apply-all-ancilliary
Specifies that the smoothing filtering will be will be applied to all of the a ncilliary data merged with the survey data (i.e. the navigation, sensor depth, heading, attitude, and altitude data).

--recalculate-bathymetry
This option causes the bathymetry in the survey data to be recalculated during preprocessing. The specifics of the recalculation are format dependent.

--no-change-survey
This option forces **mbpreprocess** to not modify the survey data.

--multibeam-sidescan-source=*SOURCE*
This option defines the source type for the backscatter used in calculating the sidescan imagery associated with multibeam data for multibeam data formats that have multiple types of backscatter records and may have more that one type available in a data stream. The relevant formats are Reson 7k (88), Teledyne 7k3 (89), and Kongsberg kmall (161). If *SOURCE* = S or s, then uncalibrated snippet records will be used. If *SOURCE* = C or c, then the source records will be calibrated snippets. If *SOURCE* = B or b, then the source will be wide beam backscatter that is provided in a sidescan-like port and starboard time series. Finally, if *SOURCE* = W or w, then the source records will be calibrated wide beam backscatter. By default, for the relevant formats **mbpreprocess** will attempt to use the best available source, with a preference order of calibrated snippets, then snippets, then calibrated wide beam, then wide beam.

--sounding-amplitude-filter=*THRESHOLD*
This option enables flagging soundings that have amplitude or intensity values less than the specified threshold value for some data formats. The two relevant formats are Reson 7k (88) and 3D at Depth lidar (232 and 233).

--sounding-altitude-filter=*THRESHOLD*
This option enables flagging soundings that have altitude values less than the specified threshold value for some data formats. The relevant formats are 3D at Depth lidar (232 and 233).

--head1-offsets=*X/Y/Z/HEADING/ROLL/PITCH*
This option specifies the positional and angular offsets of optical head 1 relative to the origin for a dual head lidar system. The only relevant formats are for 3D at Depth lidars (232 and 233).

--head2-offsets=*X/Y/Z/HEADING/ROLL/PITCH*
This option specifies the positional and angular offsets of optical head 2 relative to the origin for a dual head lidar system. The only relevant formats are for 3D at Depth lidars (232 and 233).

--kluge-time-jumps=*SECONDS*

--kluge-fix-7k-timestamps=*TARGETOFFSET*
This option was created to fix some Teledyne Reson T50 multibeam data (format 89) collected in 2024 with an MBARI Mapping AUV. The sonar ping timestamps jumped considerably (by as much as a day!!) while the ancillary nav, sensor depth, and attitude records maintained accurate timing. In good data the ping records will occur in the datastream with a fairly stable delay from the most recent ancillary records (0.7 seconds in this case). This option forces the pings records to have timestamps that have the target offset from the most recent ancillary records whenever the recorded offset magnitude is more than four times the target offset.

--kluge-ancilliary-time-jumps=*SECONDS*

--kluge-mbaripressure-time-jumps=*SECONDS*

--kluge-beam-tweak=*FACTOR*

--kluge-soundspeed-tweak=*FACTOR*

--kluge-zero-attitude-correction

--kluge-zero-alongtrack-angles
This option zeroes alongtrack angles used to calculated bathymetry.

--kluge-fix-wissl-timestamps

--kluge-ignore-snippets
This option was created to handle some Kongsberg EM122 data in *.all files (supported as format 58) in which most, but not all, of the snippet backscatter records are missing. The i/o module for this format views pings without snippet records as incomplete. When **--kluge-ignore-snippets** is specified, **mbpreprocess** creates an empty snippet record for each ping, and ignores any snippet records actually in the input datastream.

--kluge-sonardepth-from-heave
This option causes heave values to be moved to the sonardepth (sometimes sensordepth) field, leaving the heave value as zero, for each survey ping. This is currently only relevant to the preprocessing of Imagenex DeltaT multibeam data in the vendor 83P format (191). The output of preprocessing format 191 data are format 192 files.

--kluge-early-mbari-mapping-auv
This option pertains only to Reson 7125 200 kHz multibeam data collected on the first MBARI Mapping AUV from 2006 through 2009. Invoking this kluge triggers **mbpreprocess** to modify the beam quality factors to be consistent with the later (>2009) standard for the original bathymetry records in Reson s7k format data.

--kluge-flipsign-roll
This option multiplies the roll values to merged by -1, flipping the sign.

--kluge-flipsign-pitch
This option multiplies the pitch values to merged by -1, flipping the sign.

EXAMPLES

To be written.....

SEE ALSO

mbsystem(1), **mbdatalist**(1), **mbprocess**(1)

BUGS

Oh yeah.