SANDIA REPORT

SAND2015-6650X
Unlimited Release
Printed June 2015

NetMOD Version 2.0 Parameters

B. John Merchant

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2015-6650x
Unlimited Release
Printed June, 2015

NetMOD Version 2.0 Parameters

B. John Merchant
Ground-based Monitoring R and E
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-MS0404

Abstract

NetMOD (Network Monitoring for Optimal Detection) is a Java-based software package for conducting simulation of seismic, hydroacoustic and infrasonic networks. Network simulations have long been used to study network resilience to station outages and to determine where additional stations are needed to reduce monitoring thresholds. NetMOD makes use of geophysical models to determine the source characteristics, signal attenuation along the path between the source and station, and the performance and noise properties of the station. These geophysical models are combined to simulate the relative amplitudes of signal and noise that are observed at each of the stations. From these signal-to-noise ratios (SNR), the probability of detection can be computed given a detection threshold.

This document describes the parameters that are used to configure the NetMOD tool and the input and output parameters that make up the simulation definitions.

ACKNOWLEDGMENTS

We would like to thank the reviewers of this document for their contributions. This document is derived from the NetSim Control Parameters and Output Data (SAIC, 2009) report.

CONTENTS

FIGURES

TABLES

NOMENCLATURE

dB	Decibel
CDF	Cumulative Distribution Function
GUI	Graphical User Interface
JVM	Java Virtual Machine
NetMOD	Network Monitoring for Optimal Detection
PDF	Probability Density Function
PSD	Power Spectral Density
SNR	Signal to Noise Ratio

1 NETMOD CONFIGURATION PARAMETERS

NetMOD has a number of configuration parameters that may be specified. The configuration parameters are contained by default within a file in the NetMOD startup directory called '*NetMOD.cfg*'. However, the file containing these configuration parameters and the individual parameters themselves may be over-ridden at the command line.

These parameters control the behavior and appearance of the NetMOD user interface, as opposed to the simulation parameters which are used to define a simulation project that may be executed within NetMOD.

However, the configuration parameters do contain some infrequently used flags that may be set in order to duplicate unique implementations of certain key algorithms in other simulation tools. These configuration parameters would only be set in order to perform comparison testing between NetMOD and other simulation tools.

The configuration parameters and description for each are included in the table below. The NetMOD User's Manual describes how these settings may be changed from within the NetMOD User Interface.

Table 1 NetMOD Configuration Parameters

Parameter Name	Parameter Value
BACKGROUND_COLOR	Background color of the NetMOD user interface. Comma separated red, green, blue, and alpha color components. Each color component can have integer values ranging from 0 to 255, i.e. '255,255,255,255'
CHART_VIEWER_ANTI_ALIASING	Enables anti alias filtering of the data lines within a chart in order to plot smoother lines. 'true' or 'false'
CHART_VIEWER_BACKGROUND_COLOR	Background color of the charts. Comma separated red, green, blue, and alpha color components. Each color component can have integer values ranging from 0 to 255, i.e. '255,255,255,255'
CHART_VIEWER_GRIDLINE_COLOR	Color of the chart gridlines. Comma separated red, green, blue, and alpha color components. Each color component can have integer values ranging from 0 to 255, i.e. '255,255,255,255'
CHART_VIEWER_GRIDLINE_WIDTH	Width of the chart gridlines in points. Integer value, i.e. '1'
CHART_VIEWER_HEIGHT	Height of the charts in pixels. Integer value, i.e. '300'
CHART_VIEWER_LABEL_FONT_SIZE	Font size of the chart labels. Integer value, i.e. '10'
CHART_VIEWER_LINE_COLOR_PALETTE	Color palette used to draw data lines on charts. Multiple color values, each separated by semi-colons, i.e. '128,0,0,255;0,0,128,255;0,128,0,255'
CHART_VIEWER_LINE_WIDTH	Width of the chart data lines in points. Integer value, i.e. '1'
CHART_VIEWER_WIDTH	Width of the charts in pixels. Integer value, i.e. '700'
DIVIDER_LOCATION	Relative location of the divider separating the simulation controls from the ap. Floating point value between 0 and 1, i.e. '0.3'
FILE_CHOOSER_DIRECTORY	Directory location used by default to open files. This parameter is updated dynamically to reflect the last location a user had opened a file.
FONT_SIZE	Font size in points. Integer value, i.e. '14'
ICON_SIZE	Icon size. 'small', 'medium', or 'large'
MAP	String identifying the map plugin, i.e. 'None', 'Basic', or 'WorldWind'

MAP_BACKGROUND_COLOR	Background color of the map. Comma separated red, green, blue, and alpha color components. Each color component can have integer values ranging from 0 to 255, i.e. ‘255,255,255,255’
MAP_BOUNDARIES	File location of the political boundary file, relative to the NetMOD installation directory.
MAP_BOUNDARIES_FILL	Flag to control whether the map boundaries are filled. ‘true’ or ‘false’
MAP_BOUNDARIES_FILL_COLOR	Color to fill the map boundaries. Comma separated red, green, blue, and alpha color components. Each color component can have integer values ranging from 0 to 255, i.e. ‘255,255,255,255’
MAP_BOUNDARIES_LINE	Flag to control whether the map boundary lines are drawn. ‘true’ or ‘false’
MAP_BOUNDARIES_LINE_COLOR	Color the draw the map boundaries. Comma separated red, green, blue, and alpha color components. Each color component can have integer values ranging from 0 to 255, i.e. ‘255,255,255,255’
MAP_BOUNDARIES_LINE_WIDTH	Line width of the map boundaries in pixels. Integer value, i.e. ‘2’
MAP_OUTPUT_SHADING	Flag to control whether the output maps are drawn as pixels (flat) or with interpolated smoothing (bilinear). ‘flat’ or ‘bilinear’
MAP_SOURCE_COLOR	Color used draw source locations on the map. Comma separated red, green, blue, and alpha color components. Each color component can have integer values ranging from 0 to 255, i.e. ‘255,255,255,255’
MAP_SOURCE_POINTSIZE	Size of the source location points on the map. Integer value, i.e. ‘10’
MAP_STATION_ENABLED_COLOR	Color used to draw enabled stations on the map. Comma separated red, green, blue, and alpha color components. Each color component can have integer values ranging from 0 to 255, i.e. ‘255,255,255,255’
MAP_STATION_DISABLED_COLOR	Color used to draw disabled stations on the map. Comma separated red, green, blue, and alpha color components. Each color component can have integer values ranging from 0 to 255, i.e. ‘255,255,255,255’
MAP_STATION_POINTSIZE	Size of the station location points on the map. Integer value, i.e. ‘10’
MAP_STATION_LABEL_COLOR	Color used to draw the station labels on the map. Comma separated red, green, blue, and alpha color components. Each color component can have integer values ranging from 0 to 255, i.e. ‘255,255,255,255’
MAP_TOPOGRAPHY	File location of the topography file, relative to the NetMOD installation directory.
MAP_WEIGHT	Parameter used to define how the map is constructed. This parameter is platform architecture dependent and reflects whether the map is constructed from purely java (light) objects or using native (heavy) libraries. ‘light’ or ‘heavy’
MEDIA_GRID_COLOR_PALETTE	Palette of colors used to draw the media grids. Multiple color values, each separated by semi-colons, i.e. ‘64,64,64,255;0,255,0,255;0,0,255,255;255,0,0,255’
NETSIM_PATH_SD	Flag to enable a NetSim bug related to path attenuation standard deviations. ‘true’ or ‘false’
NETSIM_C_INTERP	Flag to enable a difference in some NetSim implementations in how the interpolation of frequency values is performed. ‘true’ or ‘false’
NETSIM_NOISE_SD	Flag to enable a NetSim bug in which site noise standard deviations are specified as decibels instead of log10 units. ‘true’ or ‘false’

NUMBER_THREADS	Integer value specifying the number of threads that are used to execute the simulation. If the number of threads is set to a value less than or equal to zero, then NetMOD will use as many threads as there are processors on the computer.
OUTPUT_NAME	A string representing how any generated simulation output files are to be named. If this parameter is empty, then the output filename will be taken from the simulation parameter '*title*'. The definition of the *OUTPUT_NAME* may be a static string or include references to multiple simulation parameters, each contained within '*$()*'. For example, the output name definition below: *'NetMOD_Output_$(Simul-Technology)_$(Run-Type)_$(net)_$(Event-Detection-Criteria)'* Would create output files that were dynamic based upon the specific simulation that is being run, such as: NetMOD_Output_SEISMIC_DETECTION_Network_3P

The following is an example NetMOD configuration file with the parameters set to their defaults:

```
#--- NetMOD Application Properties ---
#Fri Apr 03 09:24:25 MDT 2015
BACKGROUND_COLOR=255,255,255,255
CHART_VIEWER_ANTI_ALIASING=false
CHART_VIEWER_BACKGROUND_COLOR=255,255,255,255
CHART_VIEWER_GRIDLINE_COLOR=192,192,192,255
CHART_VIEWER_GRIDLINE_WIDTH=1.0
CHART_VIEWER_HEIGHT=300
CHART_VIEWER_LABEL_FONT_SIZE=10.0
CHART_VIEWER_LINE_COLOR_PALETTE=128,0,0,255;0,0,128,255;0,128,0,255;128,128,0,255;128,0,
128,255;0,128,128,255;64,64,64,255;255,64,64,255;64,64,255,255;64,255,64,255;255,255,64,255;255,64,
255,255;64,255,255,255
CHART_VIEWER_LINE_WIDTH=1.0
CHART_VIEWER_WIDTH=700
DIVIDER_LOCATION=0.3
FILE_CHOOSER_DIRECTORY=
FONT_SIZE=14
ICON_SIZE=medium
MAP=WorldWind
MAP_BACKGROUND_COLOR=240,240,240,255
MAP_BOUNDARIES=data/polbnd15m.shp
MAP_BOUNDARIES_FILL=false
MAP_BOUNDARIES_FILL_COLOR=255,255,255,255
MAP_BOUNDARIES_LINE=true
MAP_BOUNDARIES_LINE_COLOR=255,255,255,255
MAP_BOUNDARIES_LINE_WIDTH=2.0
MAP_OUTPUT_SHADING=bilinear
MAP_SOURCE_COLOR=255,200,0,255
MAP_SOURCE_POINTSIZE=12.0
MAP_STATION_DISABLED_COLOR=255,0,0,255
MAP_STATION_ENABLED_COLOR=0,255,0,255
```

MAP_STATION_LABEL_COLOR=255,255,255,255
MAP_STATION_POINTSIZE=16.0
MAP_TOPOGRAPHY=data/world.topo.bathy.200408.3x5400x2700.jpg
MAP_WEIGHT=light
MEDIA_GRID_COLOR_PALETTE=64,64,64,255;0,255,0,255;0,0,255,255;255,0,0,255;255,200,0,255;25
5,255,0,255;255,175,175,255;0,255,255,255;255,0,255,255;255,255,255,255
NETSIM_C_INTERP=false
NETSIM_NOISE_SD_DB=false
NETSIM_PATH_SD=false
NUMBER_THREADS=8
OUTPUT_NAME=${Simul-Technology}_${Run-Type}_${net}_${Event-Detection-Criteria}

2 NETMOD SIMULATION PROJECT PARAMETERS

2.1 LibPar Format

NetMOD utilizes a *libpar* compatible ASCII file format for storing simulation projects. *Libpar* is a powerful and flexible format for handling parameters. *Libpar* parameters may be organized into a single file or into a hierarchy of files that are referenced from one file to the next.

Libpar parameters are case sensitive and take the format of:

```
<parameter>=<value>
```

The <*value*> that a parameter is assigned may be optionally enclosed within quotes (" ").
Quotes may be required if the value contains a space, such as in a path name.

Comments start with a "#" symbol.

The value assignment to a parameter may contain references to other parameters, enclosed within: $(<*parameter*>). For example:

Parameter-A=one
Parameter-B=$(Parameter-A)_two

Will result in *Parameter-B* having the value of "one_two".

Parameters may be assigned or re-assigned in any order within the parameter file. However, the file contents are processed sequentially and any parameter references are evaluated relative to the parameter assignments up to that point within the file. For example:

```
Parameter-A=one
Parameter-B=$(Parameter-A)
Parameter-A=two
```

will result in *Parameter-A* being equal to "two" and *Parameter-B* being equal to "one"

A *libpar* file may be linked to any number of other *libpar* files through the reference:

par="/path/to/file1.par"
par="/path/to/file2.par"

A reference to another *libpar* file is loaded and processed prior to processing the next line within the current *libpar* file.

2.1.1 Tables

A *libpar* file may also contain formatted tables of parameter values. A table is identified by a tag at the beginning and end of the table:

> *#!BeginTable <Table-Name>*
> ...
> *#!EndTable*

where <*Table-Name*> is the identifying name of the table.

The first row within a table contains the column names of the table with each column name delimited by a vertical bar ("|"). Subsequent rows within the table contain the space delimited tabular values. Each row is preceded by a unique row label preceded by the "@" symbol.

```
#!BeginTable ExampleTable
        column1 | column2 | column3 | column4
@row1   value1_1  value1_2  value1_3  value1_4
@row2   value2_1  value2_2  value2_3  value2_4
...
#!EndTable
```

The values may be numeric or strings and may contain references to other parameters.

The column names are case sensitive and may be defined in any order. A table is not required to contain all of the columns which are defined.

In addition, there may be multiple declarations of the same *Table-Name*. The columns in each table declaration will be joined together based upon matching row labels.

The content of a table may be reference by a parameter declaration outside of the table using the format:

> *$(<Table-Name>[<row-label>].<column-name>)*

For example, using the *ExampleTable* above:

> *Parameter-One=$(ExampleTable[row1].column1)*

would assign *Parameter-One* equal to *value1_1*.

2.2 Parameter Hierarchy

NetMOD utilizes a hierarchy of *libpar* files and formatted data files for organizational purposes when it writes out a simulation project. NetMOD can read in any structure of *libpar* files, provided they correctly abide by the specifications. The structure NetMOD writes out is shown below:

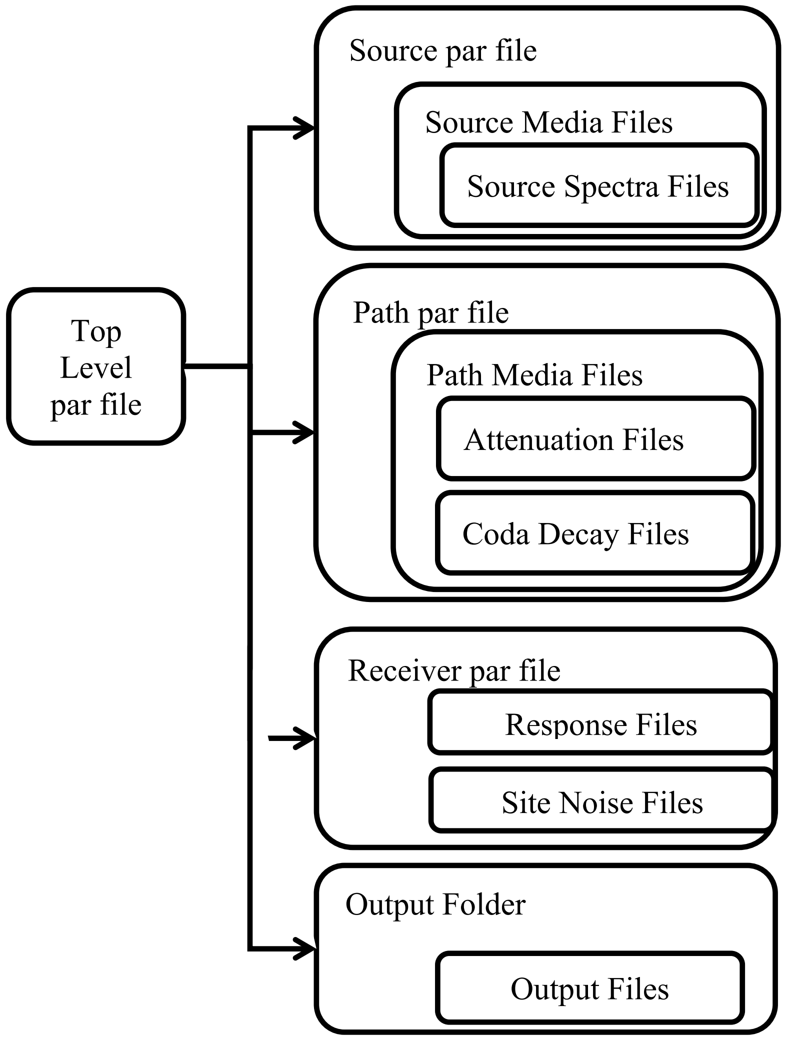

Figure 1 NetMOD Project File Hierarchy

When reading or writing a NetMOD parameter file, it is only necessary to direct NetMOD to the top-level file. This top-level file will contain the necessary references to all of the other files in the simulation project.

NetMOD implicitly defines a *libpar* parameter called *NS_CONFIG*, short for Network Simulation Configuration. This *NS_CONFIG* variable is automatically assigned the directory path of the top level parameter file and should not be re-assigned. The *NS_CONFIG* variable may be referenced throughout the parameter files.

2.3 General Parameters

Table 2 General Simulation Parameters

Parameter	Description
title	Title of the NetMOD project.
Run-Type	*NetMOD* run type. The only currently supported option is "detection".
Sub-Type	*NetMOD* run subtype. Acceptable values are "threshold" and "probability."
Simul-Technology	Simulation technology type. Acceptable values are "Seismic," "Hydro," or "Infra."
Simul-Baseline	Simulation baseline model name for the given *Simul-Technology*. It is the prefix assigned to the *NetSim* output files. (Required parameter; no default)
Event-Detection-Criteria	The minimum criteria that must be met to declare an event detection. This is a free-form mnemonic string that provides a simple mapping to more complex syntax expressed in the *EDC-Conversion-File.* Examples include "3P," for P-wave detection by at least three stations, or "2P AND 1LR," for two P and one LR wave detection by the network. The mapping of this string must exist in the *EDC-Conversion-File*. (Default: "3P")
EDC-Conversion-File	The file that contains the mapping of the *Event-Detection-Criteria* between an easily human readable name and the specific syntax required for the detection criteria.
Size-Type	Event size type. Acceptable values are "yield," "mb," "Ms," "mbLg," or "moment." If "yield" is specified, then *Event-Size*, *Min-Event-Size*, and *Max-Event-Size* are specified as yield [kT]. If a magnitude or "moment" is specified, then *Event-Size*, *Min-Event-Size*, and *Max-Event-Size* are specified as log moment [N-m]. (Default: "mb")
Event-Size	Fixed event size assumed for a probability run or in a location run with fixed event size. The units are determined by *Size-Type*. (Default: 0.01, if *Size-Type* is "yield"; 3.0, otherwise)
Min-Event-Size	Minimum event size allowed for a detection threshold run. The units are determined by *Size-Type*. (Default: 10^{-5}, if *Size-Type* is "yield"; 0.0, otherwise)
Max-Event-Size	Maximum event size allowed in a detection threshold run. The units are determined by *Size-Type*. (Default: 1.0, if *Size-Type* is "yield"; 5.0, otherwise)
conf	Network detection probability level. (Default: 0.9 → 90%)
Low-Prob-Cutoff	Low probability cutoff expressed as a fraction between 0.0 and 1.0. Any station detection probability below the cut off will be set to zero.
Regional-Xover-Distance	Cross over distance used in seismic simulations for transitioning between teleseismic and regional path attenuation.
Noise-Amplitude	Method of computing noise amplitudes. The only option available is 'NetSim'.
Num-MonteCarlo-Iter	Number of Monte Carlo location iterations (= 1 for probability weighted location uncertainties). (Default: 1)
MonteCarlo-Seed	Integer value representing a random number seed to be used for Monte Carlo simulations. If the seed value is less than or equal to zero, then the current time will be used as a seed. A static seed value allows for somewhat deterministic and reproducible random sequences.
Pct-Lo	Minimum location confidence level for Monte Carlo simulations (e.g., the median Monte Carlo realization has *Pct-Lo* of 0.50). (Default: 0.50)

Pct-Hi	Maximum location confidence level for Monte Carlo simulations (e.g., the 90th percentile Monte Carlo realization has *Pct-Hi* = 0.90). (Default: 0.90)
Depth-Fixed	Event depth constraint for location uncertainties. Acceptable values are Boolean TRUE and FALSE. (Default: TRUE)
Event-Depth	Event depth [km]. (Default: 0.0)
Freq-Sampling	List of amplitude sampling frequencies [Hz]. (Required parameter; no default; the list is limited to a maximum of 30 frequencies)
Freq-Type	Method of combining the station SNR at multiple frequencies: 'high' identifies the SNR with the highest probability and 'average' computes the mean SNR.
Verbose-Level	Flag to select whether the extended output list file (prefix.*lst*), which contains network and station details, is written. (≤ 0 for no output, > 0 file is written; = 3 for extensive debugging output). (Default: 0)
Signal-Amplitude	Method of computing signal amplitudes. Seismic: 'NetSim' Hydroacoustic: 'NetSim' Infrasound: 'NetSim', 'LANL', or 'LePichon2011'
Path-Model-Blockage	Path model name to indicate a blockage path (e.g., hydro blockage). This parameter is only used for hydroacoustic simulations.
Explosion-Type	The type of explosion to model for infrasonic simulations: 'chemical': Simulate a chemical explosion. 'nuclear': Simulate a nuclear explosion.
Wind-Mode	Method of computing the infrasound wind vector, in order of increasing accuracy and computation time: 'none': no wind. A value of 0 m/s is used in the calculations. 'sta': wind at station location 'source': wind at source location 'source_sta': average of the wind at station and source locations 'path': average of the wind along the source-receiver great circle path sampled every 50 km
Wind-Model	Wind model to use when calculating wind vectors. Only 'HWM' is currently available.
Wind-Day-Of-Year	Ordinal day of year; acceptable values are days 1–366, and -1 means no day-of-year dependence. (Default: -1)
Wind-Time-Of-Day	Hour of the day; acceptable values are 0–23 hours, and -1 means no time of day dependence. (Default: -1)

2.4 Source Parameters

Table 3 Source Simulation Parameters

Parameter	Description
Epi-Model	Specifies the epicenter source grid, defined within the *EpiSource* table.
Source-Grid-File	File name for source media grid.
Source-Media-Desc-File	File name for source media description.

2.4.1 EpiSource Table

The *EpiSource* table defines multiple epicenter source grids. The specific source grid region to be used is identified by the *Epi-Model* parameter.

Table 4 EpiSource Table Parameters

Parameter	Description
Region	Name of the source grid region, corresponding to the *Epi-Model* parameter.
Min_Lat	Minimum latitude of the source grid in degrees.
Max_Lat	Maximum latitude of the source grid in degrees.
Min_Lon	Minimum longitude of the source grid in degrees.
Max_Lon	Maximum longitude of the source grid in degrees.
Del_Lat	Latitude increment of the source grid in degrees.
Del_Lon	Longitude increment of the source grid in degrees.
Regular	Boolean flag (true or false) indicating whether the source grid should be represented by a regular rectangular grid of points or whether the source grid more equally spaced with fewer points closer to the poles.

When writing out the *EpiSource* table, NetMOD uses the *Region* as the row label.

Table 5 EpiSource Table Example

```
#!BeginTable EpiSource
          Region | Min_Lat | Max_Lat | Min_Lon | Max_Lon | Del_Lat | Del_Lon | Regular
@Global   Global   -80.0     80.0      -180.0    180.0     4.0       4.0       true
@US       US        25.00    51.00     -130.00   -64.00    1.0       1.0       true
#!EndTable
```

2.5 Source Spectra (SOR) Text File

The source spectra file specifies the source spectra across a range of amplitudes and frequencies. The file is stored as an ASCII text file. A description and example of the parameters within the file and the file format are shown below.

Files in this format are identified by having a "*.sor*" extension.

Table 6 Source Spectra File Parameters

Parameter	Description
title	Title of the source spectra file
comment	Zero or more comment lines. Each comment line must start with '#'.
n_frequencies	Integer value representing the number of frequencies

frequency(i) i = 1..*n_frequencies*	Frequencies (Hz).
n_moments	Integer value representing the number of moments.
moment(j) j = 1..*n_moments*	Log moments. The units are N-m for seismic, uPa for hydroacoustic, and Pa for infrasonic noise.
amplitude(i, j) i = 1..*n_frequencies* j = 1..*n_moments*	Log source spectra amplitudes. The units are N-m.

Table 7 Source Spectra File Format

Format	Comment
title	
comment	
n_frequency	
frequency(1) … *frequency*(*n_frequencies*)	Values are separated by white space (space, tab, or new line) and may span multiple lines.
comment	
n_moment	
moment(1) … moment(*n_moment*)	Values are separated by white space (space, tab, or new line) and may span multiple lines.
Source amplitudes for frequency = *frequency(1)*	
amplitude(1,1) … *amplitude*(1, *n_moments*)	Values are separated by white space (space, tab, or new line) and may span multiple lines.
…	
Source amplitudes for frequency = *frequency*(*n_frequencies*)	
amplitude(*n_frequencies*,1) … *amplitude*(*n_frequencies*, *n_moments*)	Values are separated by white space (space, tab, or new line) and may span multiple lines.

Table 8 Source Spectra File Example

```
Explosion Source: Mueller Murphy granite at normal scaled depth
  24
  0.010    0.050     0.100    0.200    0.300    0.400    0.600     0.800
  1.000    1.200     1.500    2.000    2.500    3.000    4.000     5.000
  6.000    7.000     8.000    9.000   10.000   12.000   15.000    20.000
  8
  11.839   12.598   13.356   14.115   14.874   15.633   16.392   17.151
Source amplitudes for frequency:    0.010
11.839    12.598    13.356    14.115    14.874    15.633    16.392    17.151
Source amplitudes for frequency:    0.050
11.839    12.598    13.356    14.115    14.874    15.633    16.394    17.160
Source amplitudes for frequency:    0.100
11.839    12.598    13.356    14.115    14.875    15.635    16.399    17.185
Source amplitudes for frequency:    0.200
11.839    12.598    13.356    14.116    14.876    15.639    16.420    17.264
Source amplitudes for frequency:    0.300
11.839    12.598    13.357    14.116    14.877    15.647    16.451    17.353
Source amplitudes for frequency:    0.400
11.839    12.598    13.357    14.117    14.880    15.657    16.487    17.436
Source amplitudes for frequency:    0.600
11.839    12.598    13.357    14.118    14.886    15.683    16.566    17.549
Source amplitudes for frequency:    0.800
```

```
11.839    12.598    13.358    14.120    14.895    15.714    16.633    17.568
Source amplitudes for frequency:    1.000
11.839    12.598    13.359    14.123    14.905    15.746    16.672    17.510
Source amplitudes for frequency:    1.200
11.839    12.599    13.360    14.127    14.917    15.775    16.676    17.419
Source amplitudes for frequency:    1.500
11.839    12.599    13.361    14.132    14.935    15.802    16.623    17.277
Source amplitudes for frequency:    2.000
11.840    12.600    13.365    14.143    14.959    15.785    16.474    17.069
Source amplitudes for frequency:    2.500
11.840    12.602    13.369    14.153    14.963    15.704    16.322    16.896
Source amplitudes for frequency:    3.000
11.841    12.603    13.373    14.159    14.937    15.600    16.188    16.750
Source amplitudes for frequency:    4.000
11.842    12.607    13.378    14.145    14.817    15.400    15.964    16.513
Source amplitudes for frequency:    5.000
11.844    12.610    13.376    14.086    14.671    15.230    15.783    16.326
Source amplitudes for frequency:    6.000
11.846    12.611    13.359    13.993    14.536    15.085    15.633    16.171
Source amplitudes for frequency:    7.000
11.847    12.610    13.323    13.889    14.415    14.961    15.504    16.040
Source amplitudes for frequency:    8.000
11.848    12.603    13.269    13.787    14.308    14.851    15.391    15.925
Source amplitudes for frequency:    9.000
11.849    12.590    13.202    13.692    14.213    14.753    15.291    15.824
Source amplitudes for frequency:    10.000
11.848    12.569    13.129    13.605    14.127    14.665    15.201    15.733
Source amplitudes for frequency:    12.000
11.841    12.503    12.982    13.452    13.975    14.511    15.045    15.576
Source amplitudes for frequency:    15.000
11.811    12.362    12.785    13.262    13.788    14.321    14.853    15.383
Source amplitudes for frequency:    20.000
11.695    12.112    12.523    13.016    13.543    14.075    14.605    15.133
```

2.6 Source Media Files

The source media is defined by two files: The first describes the source media types ("*.smd*") and the second describes the gridded regions that the media types are defined over ("*.sgr*").

2.6.1 Source Media Description (SMD)Text File

The source media description file specifies the properties of each of the source media types. A description and example of the parameters within the file and the file format are shown below.

Files in this format are identified by having a "*.smd*" extension.

Table 9 Source Media Description File Parameters

Parameter	Description
title	Title of the source spectra file.
comment	Zero or more comment lines. Each comment line must start with '#'.
Directory	Directory in which the source spectra files are contained. Note that the directory may be an absolute path or relative to the location of the SMD file.
n	Integer value representing the number of source media types.
name(i)	Source media name.

i = 1..*n*	
sourcespectra.sor(i) *i* = 1..*n*	File name containing the source spectra.
a_yield(i) b_yield(i) *a_mb(i) b_mb(i)* *a_ms(i) b_ms(i)* *a_mbLg(i) b_mbLg(i)* *a_ML(i) b_ML(i)* *a_Mw(i) b_Mw(i)* *a_Pa(i) b_Pa(i)* *i* = 1..*n*	Slope (*a*) and intercept (*b*) of log moment versus magnitude for several magnitude types: *log(moment) = a * magnitude + b* Note that in the case of yield (kT), the relationship is: *log(moment) = a_yield * log(yield) + b_yield* Moment units are N-m.
P_exc(i) pP_exc(i) Pg_exc(i) S_exc(i) *Lg_exc(i) LR_exc(i) H_exc(i) T_exc(i)* *I_exc(1) O_exc(i)* *i* = 1..*n*	Source excitation factor for each of the defined phases within NetMOD: P, pP, Pg, S, Lg, LR, H, T, I, and O (other).
P_vel(i) pP_vel(i) Pg_vel(i) S_vel(i) *Lg_vel(i) LR_vel(i) H_vel(i) T_vel(i)* *I_vel(1) O_vel(i)* *i* = 1..*n*	Source media velocity in m/s for each of the defined phases within NetMOD: P, pP, Pg, S, Lg, LR, H, T, I, and O (other).

Table 10 Source Media Description File Format

Format
title *comment* *directory* *n* *name(1)* *sourcespectra.sor(1)* *density(1)* *a_yield(1) b_yield(1) a_mb(1) b_mb(1) a_ms(1) b_ms(1) a_mbLg(1) b_mbLg(1) a_ML(i) b_ML(i) a_* *P_exc(1) pP_exc(1) Pg_exc(1) S_exc(1) Lg_exc(1) LR_exc(1) H_exc(1) T_exc(1) I_exc(1) O_exc(1)* *P_vel(1) pP_vel(1) Pg_vel(1) S_vel(1) Lg_vel(1) LR_vel(1) H_vel(1) T_vel(1) I_vel(1) O_vel(1)*   *name(n)* *sourcespectra.sor(n)* *density(n)* *a_yield(n) b_yield(n) a_mb(n) b_mb(n) a_ms(n) b_ms(n) a_mbLg(n) b_mbLg(n)* *P_exc(n) pP_exc(n) Pg_exc(n) S_exc(n) Lg_exc(n) LR_exc(n) H_exc(n) T_exc(n) I_exc(n) O_exc(n)* *P_vel(n) pP_vel(n) Pg_vel(n) S_vel(n) Lg_vel(n) LR_vel(n) H_vel(n) T_vel(n) I_vel(n) O_vel(n)*

Table 11 Source Media Description File Example

```
Source Media
.\
3
reference
-
2500.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
granite
```

```
expl-mm-granite-norm-depth.sor
2550.000
0.760 14.874 0.990 10.510 1.000 12.530 1.000 10.214 0.000 0.000 0.000 0.000
0.740  0.000 0.930  0.410 1.060  0.190 0.000  0.000 0.000 0.000
5500.000  0.000 5500.000 3175.000 3175.000 3175.000 0.000 0.000 0.000 0.000
water600
water_600m.sor
1000.000
1.000 13.762 0.833 9.200 0.930 11.390 0.000 0.000 0.000 0.000 0.000 0.000
0.300  0.000 0.380 0.000 0.000 1.680 0.000 0.000 0.000 0.000 1498.000
1498.000 1498.000  0.000 0.000 1498.000  0.000  0.000 0.000 0.000
```

2.6.2 Source Grid (SGR) Text File

The file is stored as an ASCII text file. A description and example of the parameters within the file and the file format are shown below. The file specifies a rectangular grid of latitude and longitude cells. Typically the grid spans the entire globe. Each cell of the grid is identified by the latitude and longitude at the cells lower-left corner. A description and example of the parameters within the file and the file format are shown below.

Files in this format are identified by having a "*.sgr*" extension.

Table 12 Source Grid File Parameters

Parameter	Description
title	Title of the source spectra file.
comment	Zero or more comment lines. Each comment line must start with '#'.
n_latitudes	Integer value representing the number of latitudes.
n_longitudes	Integer value representing the number of longitudes.
latitude_start	Southern most starting latitude of the grid.
longitude_start	Western most starting longitude of the grid.
latitude_delta	Latitude increment
longitude_delta	Longitude increment
media_name(i) i = *n_latitudes* * *n_longitudes*	Name of the media type for each cell.
latitude(i) i = *n_latitudes* * *n_longitudes*	Southern most latitude for each cell.
longitude(i) i = *n_latitudes* * *n_longitudes*	Western most longitude for each cell.

Table 13 Source Grid File Format

Format
title *comment* *n_latitudes n_longitudes* *latitude_start longitude_start* *latitude_delta longitude_delta* *media_name*(1) *latitude*(1) *longitude*(1) … … *media_name*(*n_latitudes* * *n_longitudes*) *latitude*(*n_latitudes* * *n_longitudes*) *longitude*(*n_latitudes* * *n_longitudes*)

Table 14 Source Grid File Example

```
Source Media
      36          72
 -90.00000000     -180.00000000
   5.00000000        5.00000000
granite      -90.00000000     -180.00000000
granite      -90.00000000     -175.00000000
granite      -90.00000000     -170.00000000
granite      -90.00000000     -165.00000000
granite      -90.00000000     -160.00000000
granite      -90.00000000     -155.00000000
granite      -90.00000000     -150.00000000
granite      -90.00000000     -145.00000000
granite      -90.00000000     -140.00000000
```

```
granite     -90.00000000     -135.00000000
granite     -90.00000000     -130.00000000
granite     -90.00000000     -125.00000000
granite     -90.00000000     -120.00000000
granite     -90.00000000     -115.00000000
granite     -90.00000000     -110.00000000
granite     -90.00000000     -105.00000000
granite     -90.00000000     -100.00000000
…
```

2.7 Path

Table 15 Path Simulation Parameters

Parameter	Description
Path-Media-Desc-File	File name for path media input (file.*pmd*). (Required parameter; no default)
Path-Grid-File	File name for path grid input (file.*pgr*). (Required parameter; no default)

2.7.1 PhaseSpec Table

The *PhaseSpec* table defines the simulation parameters that are unique to a specific phase.

Table 16 PhaseSpec Table Parameters

Parameter	Description
phase	Phase name (P, pP, Pg, S, Lg, LR, H, T, I, and O).
tech	Station technology: S (Seismic), H (Hydoacoustic), and I (Infrasonic).
TimeW	Time window length (seconds) used to convert between power spectral density and amplitude.
GVel1	Minimum group velocity (m/s) used to compute a time window length if *TimeW* is less than zero.
GVel2	Maximum group velocity (m/s) used to compute a time window length if *TimeW* is less than zero.
LogAmpCorr	Default phase specific correction to the log signal amplitude. May be overridden with a station specific correction in the *StaPhaseSpec* table.
LogAmpCorrSD	Default standard deviation of the log signal amplitude correction. May be overridden with a station specific standard deviation in the *StaPhaseSpec* table.
SnrThresh	Default signal to noise ratio threshold required for a detection. May be overridden with a station specific threshold in the *StaPhaseSpec* table. The SNR threshold is specified as a ratio and not a log.
deltim	Default travel time standard deviation (seconds) used for location simulation. May be overridden with a station specific standard deviation in the *StaPhaseSpec* table.
delaz	Default azimuth standard deviation (degrees) used for location simulation. May be overridden with a station specific standard deviation in the *StaPhaseSpec* table.
velocity	Default receiver media velocity (m/s). May be overridden with a station specific velocity in the *StaPhaseSpec* table.
beam	Beam type: ‘coh’ for a coherent beam and “inc” for an incoherent beam.
PrevPhase	Phase name of the previous phase arrival used for computing prior phase noise. If no prior phase noise is to be computed, may be set to ‘-‘.
NF_file	File name for the noise factor, or coda decay rate, file.
TT_ME_file	File name for the distance-dependent travel-time modeling error file. (not currently used).

When writing out the *PhaseSpec* table, NetMOD uses the *phase* as the row label.

2.8 Path Media Files

The path media is defined by two files: The first describes the path media types (“.*pmd*”) and the second describes the gridded regions that the media types are defined over (“.*pgr*”).

2.8.1 Path Media Description (PMD) File

The path media description file is stored as an ASCII text file. The file specifies the properties of each of the path media types. A description and example of the parameters within the file and the file format are shown below.

The path media description file defines multiple path media types. Each path media type has properties that are defined for each of the 10 defined phases within NetMOD. The defined phases and their order are:

1) P/PKP (or Pn)
2) pP
3) Pg
4) S (or Sn)
5) Lg
6) LR
7) H (Hydroacoustic-H)
8) T (Hydroacoustic-T)
9) I (Infrasound)
10) O (Other)

Files in this format are identified by having a "*.pmd*" extension.

Table 17 Path Media Description File Parameters

Parameter	Description
title	Title of the source spectra file.
comment	Zero or more comment lines. Each comment line must start with '#'.
directory	Directory in which the attenuation and travel time files are contained. Note that the directory may be an absolute path or relative to the location of the PMD file.
n	Integer value representing the number of path media types
name(i) *i* = 1..*n*	Path media name
attenuation.atn(i) *i* = 1..*n*	File name containing the path attenuation .
traveltime.tim(i) *i* = 1..*n*	File name containing the travel time. Note, this is used for location simulations which are not yet supported within NetMOD.
atn_std(i,j) *i* = 1..*n* *j* = 1..*10(# of phases)*	Standard deviation of the log attenuation for the path media type and phase.
src_corr(i,j) *i* = 1..*n* *j* = 1..*10(# of phases)*	Log source correction for teleseismic distances for the path media type and phase.
recvr_corr(i,j) *i* = 1..*n* *j* = 1..*10(# of phases)*	Log receiver correction for teleseismic distances for the path media type and phase.
time_std(i,j) *i* = 1..*n* *j* = 1..*10(# of phases)*	Standard deviation of the travel time in seconds for the path media type and phase.
az_std(i,j) *i* = 1..*n*	Standard deviation of the azimuth in degrees for the path media type and phase.

j = 1..*10(# of phases)*	

Table 18 Path Media Description File Format

Format
title *comment* *directory* *n* *name(1)* *attenuation.atn(1,1) traveltime.tim(1,1)* *atn_std(1,1) source_corr(1,1) receiver_corr(1,1) time_std(1,1)* *azimuth_std(1,1)* *…* *attenuation.atn(1,10) traveltime.tim(1,10)* *atn_std(1,10) src_corr(1,10) recvr_corr(1,10) time_std(1,10) az_std(1,10)* *…* *name(n)* *attenuation.atn(n,1) traveltime.tim(n,1)* *atn_std(n,1) src_corr(n,1) recvr_corr(n,1) time_std(n,1) az_std(n,1)* *…* *attenuation.atn(n,10) traveltime.tim(n,10)* *atn_std(n,10) src_corr(n,10) recvr_corr(n,10) time_std(n,10) az_std(n,10)*

Table 19 Path Media Description File Example

```
Path Media
.\
5
reference
ptele.atn   -
   0.300       0.000       0.000       0.000       0.000
-     -
   0.000       0.000       0.000       0.000       0.000
pg.atn      -
   0.500       0.000       0.000       0.000       0.000
stele.atn   -
   0.500      -1.000       0.000       0.000       0.000
lgtec.atn   -
   0.320       0.000       0.000       0.000       0.000
lrrpg.atn   -
   0.300       0.000       0.000       0.000       0.000
-     -
   0.000       0.000       0.000       0.000       0.000
-     -
   0.000       0.000       0.000       0.000       0.000
-     -
   0.000       0.000       0.000       0.000       0.000
-     -
   0.000       0.000       0.000       0.000       0.000
stable_land
pn.atn      -
   0.260       0.000       0.000       0.000       0.000
-     -
   0.000       0.000       0.000       0.000       0.000
```

```
pg.atn       -
   0.250       0.000       0.000       0.000       0.000
sn.atn       -
   0.190      -1.000       0.000       0.000       0.000
lg.atn       -
   0.280       0.000       0.000       0.000       0.000
lrrpg.atn    -
   0.500       0.000       0.000       0.000       0.000
-     -
   0.000       0.000       0.000       0.000       0.000
-     -
   0.000       0.000       0.000       0.000       0.000
-     -
   0.000       0.000       0.000       0.000       0.000
-     -
   0.000       0.000       0.000       0.000       0.000
tectonic_land
pntec.atn    -
   0.400       0.000       0.000       0.000       0.000
-     -
   0.000       0.000       0.000       0.000       0.000
pgtec.atn    -
   0.340       0.000       0.000       0.000       0.000
sntec.atn    -
   0.290      -1.800       0.000       0.000       0.000
lgtec.atn    -
   0.320       0.000       0.000       0.000       0.000
lrrpg.atn    -
   0.500       0.000       0.000       0.000       0.000
-     -
   0.000       0.000       0.000       0.000       0.000
-     -
   0.000       0.000       0.000       0.000       0.000
-     -
   0.000       0.000       0.000       0.000       0.000
-     -
   0.000       0.000       0.000       0.000       0.000
tectonic_deepwater
pntec.atn    -
   0.400       0.000       0.000       0.000       0.000
-     -
   0.000       0.000       0.000       0.000       0.000
pgtec.atn    -
   0.340       0.000       0.000       0.000       0.000
sntec.atn    -
   0.290      -1.000       0.000       0.000       0.000
barrier_lg.atn    -
   0.320     -99.000       0.000       0.000       0.000
lrrpg.atn    -
   0.500       0.000       0.000       0.000       0.000
-     -
   0.000       0.000       0.000       0.000       0.000
-     -
   0.000       0.000       0.000       0.000       0.000
-     -
   0.000       0.000       0.000       0.000       0.000
-     -
```

```
   0.000       0.000       0.000       0.000       0.000
stable_deepwater
pntec.atn   -
   0.400       0.000       0.000       0.000       0.000
-     -
   0.000       0.000       0.000       0.000       0.000
pgtec.atn   -
   0.340       0.000       0.000       0.000       0.000
sntec.atn   -
   0.290      -0.800       0.000       0.000       0.000
barrier_lg.atn    -
   0.320     -99.000       0.000       0.000       0.000
lrrpg.atn   -
   0.500       0.000       0.000       0.000       0.000
-     -
   0.000       0.000       0.000       0.000       0.000
-     -
   0.000       0.000       0.000       0.000       0.000
-     -
   0.000       0.000       0.000       0.000       0.000
-     -
   0.000       0.000       0.000       0.000       0.000
```

2.8.2 Path Grid (PGR) Text File

The path grid file is stored as an ASCII text file. The file specifies a rectangular grid of latitude and longitude cells. Typically the grid spans the entire globe. Each cell of the grid is identified by the latitude and longitude at the cells lower-left corner. A description and example of the parameters within the file and the file format are shown below.

Files in this format are identified by having a "*.pgr*" extension.

Table 20 Path Grid File Parameters

Parameter	Description
title	Title of the source spectra file.
comment	Zero or more comment lines. Each comment line must start with '#'.
n_latitudes	Integer value representing the number of latitudes.
n_longitudes	Integer value representing the number of longitudes.
latitude_start	Southern most starting latitude of the grid.
longitude_start	Western most starting longitude of the grid.
latitude_delta	Latitude increment
longitude_delta	Longitude increment
media_name(i) i = *n_latitudes* * *n_longitudes*	Name of the media type for each cell.
latitude(i) i = *n_latitudes* * *n_longitudes*	Southern most latitude for each cell.
longitude(i) i = *n_latitudes* * *n_longitudes*	Western most longitude for each cell.

Table 21 Path Grid File Format

Format

```
title
comment
n_latitudes n_longitudes
latitude_start longitude_start
latitude_delta longitude_delta
media_name(1) latitude(1) longitude(1)
…
…
media_name(n_latitudes * n_longitudes) latitude(n_latitudes * n_longitudes)
longitude(n_latitudes * n_longitudes)
```

Table 22 Path Grid File Example

```
Path Media
      36          72
 -90.00000000     -180.00000000
   5.00000000        5.00000000
tectonic_land            -90.00000000     -180.00000000
tectonic_land            -90.00000000     -175.00000000
tectonic_land            -90.00000000     -170.00000000
tectonic_land            -90.00000000     -165.00000000
tectonic_land            -90.00000000     -160.00000000
tectonic_land            -90.00000000     -155.00000000
tectonic_land            -90.00000000     -150.00000000
tectonic_land            -90.00000000     -145.00000000
tectonic_land            -90.00000000     -140.00000000
tectonic_land            -90.00000000     -135.00000000
```

```
tectonic_land          -90.00000000    -130.00000000
tectonic_land          -90.00000000    -125.00000000
tectonic_land          -90.00000000    -120.00000000
tectonic_land          -90.00000000    -115.00000000
tectonic_land          -90.00000000    -110.00000000
tectonic_land          -90.00000000    -105.00000000
tectonic_land          -90.00000000    -100.00000000
tectonic_land          -90.00000000     -95.00000000
tectonic_land          -90.00000000     -90.00000000
tectonic_land          -90.00000000     -85.00000000
tectonic_land          -90.00000000     -80.00000000
tectonic_land          -90.00000000     -75.00000000
tectonic_land          -90.00000000     -70.00000000
tectonic_land          -90.00000000     -65.00000000
tectonic_land          -90.00000000     -60.00000000
tectonic_land          -90.00000000     -55.00000000
tectonic_land          -90.00000000     -50.00000000
tectonic_land          -90.00000000     -45.00000000
tectonic_land          -90.00000000     -40.00000000
tectonic_land          -90.00000000     -35.00000000
stable_land            -90.00000000     -30.00000000
stable_land            -90.00000000     -25.00000000
stable_land            -90.00000000     -20.00000000
stable_land            -90.00000000     -15.00000000
stable_land            -90.00000000     -10.00000000
stable_land            -90.00000000      -5.00000000
stable_land            -90.00000000       0.00000000
stable_land            -90.00000000       5.00000000
…
```

2.9 Path Attenuation (ATN) Text File

The path attenuation file specifies the path attenuation across a range of distances and frequencies. The file is stored as an ASCII text file. A description and example of the parameters within the file and the file format are shown below.

Files in this format are identified by having an "*.atn*" extension.

Table 23 Path Attenuation File Parameters

Parameter	Description
title	Title of the source spectra file
comment	Zero or more comment lines. Each comment line must start with '#'.
n_frequencies	Integer value representing the number of frequencies
frequency(i) i = 1..*n_frequencies*	Frequencies (Hz).
n_distances	Integer value representing the number of moments.
distance(j) j = 1..*n_amplitudes*	Distances (degrees).
*Attenuation(*i, j) i = 1..*n_frequencies* j = 1..*n_distances*	Log attenuation (unitless).

Table 24 Path Attenuation File Format

Format	Comment
title	
comment	
n_frequencies	
frequency(1) … *frequency*(*n_frequencies*)	Values are separated by white space (space, tab, or new line) and may span multiple lines.
comment	
n_distances	
distance(1) … distance(n_distances)	Values are separated by white space (space, tab, or new line) and may span multiple lines.
attenuation for frequency = *frequency(1)*	Comment Line
amplitude(1,1) … *amplitude*(1, *n_amplitude*)	Values are separated by white space (space, tab, or new line) and may span multiple lines.
…	
attenuation for frequency = *frequency*(*n_frequencies*)	
amplitude(*n_frequency*,1) … *amplitude*(*n_frequencies*, *n_distances*)	Values are separated by white space (space, tab, or new line) and may span multiple lines.

Table 25 Path Attenuation File Example

```
Veith-Clawson with PKP (H Israelsson 85/07/19)
 11
   0.500    1.000    2.000    3.000    4.000    5.000    6.000    7.000    8.000    9.000   10.000
 110
   0.000    1.000    2.000    3.000    4.000    5.000    6.000    7.000    8.000    9.000
  10.000   11.000   12.000   13.000   14.000   15.000   16.000   17.000   18.000   19.000
  20.000   21.000   22.000   23.000   24.000   25.000   26.000   27.000   28.000   29.000
  30.000   31.000   32.000   33.000   34.000   35.000   36.000   37.000   38.000   39.000
  40.000   41.000   42.000   43.000   46.000   47.000   48.000   50.000   52.000   53.000
  55.000   57.000   59.000   60.000   61.000   63.000   65.000   66.000   67.000   68.000
  69.000   70.000   71.000   72.000   73.000   74.000   75.000   76.000   77.000   78.000
  79.000   80.000   81.000   82.000   83.000   84.000   85.000   86.000   87.000   88.000
  89.000   90.000   91.000   92.000   93.000   94.000   95.000   96.000   97.000   98.000
  99.000  100.000  105.000  110.000  111.000  113.000  114.000  119.000  120.000
 123.000  124.000  141.000  142.000  151.000  152.000  153.000  154.000  157.000  161.000  180.000
Attenuation for frequency:  0.500
        -2.938  -4.129  -5.309  -5.439  -5.789  -5.999  -6.139  -6.259
        -6.339  -6.389  -6.429  -6.469  -6.489  -6.499  -6.499  -6.489
        -6.449  -6.339  -6.219  -6.029  -6.009  -6.039  -6.089  -6.179
        -6.279  -6.389  -6.489  -6.589  -6.659  -6.679  -6.659  -6.619
        -6.599  -6.599  -6.589  -6.579  -6.579  -6.579  -6.569  -6.569
        -6.559  -6.559  -6.559  -6.569  -6.579  -6.589  -6.599  -6.609
        -6.619  -6.629  -6.639  -6.649  -6.659  -6.669  -6.679  -6.689
        -6.699  -6.699  -6.709  -6.719  -6.719  -6.729  -6.739  -6.739
        -6.749  -6.749  -6.759  -6.769  -6.769  -6.779  -6.779  -6.789
        -6.799  -6.809  -6.819  -6.829  -6.849  -6.879  -6.899  -6.919
        -6.959  -6.999  -7.039  -7.089  -7.149  -7.219  -7.299  -7.379
        -7.459  -7.539  -7.619  -7.699  -7.638  -7.638  -7.438  -7.438
        -7.238  -7.238  -7.188  -7.188  -7.138  -7.138  -6.438  -6.438
        -6.538  -6.588  -6.738  -6.938  -7.038  -7.638
Attenuation for frequency:  1.000
```

```
            -3.620   -4.811   -5.991   -6.121   -6.471   -6.681   -6.821   -6.941
            -7.021   -7.071   -7.111   -7.151   -7.171   -7.181   -7.181   -7.171
            -7.131   -7.021   -6.901   -6.711   -6.691   -6.721   -6.771   -6.861
            -6.961   -7.071   -7.171   -7.271   -7.341   -7.361   -7.341   -7.301
            -7.281   -7.281   -7.271   -7.261   -7.261   -7.261   -7.251   -7.251
            -7.241   -7.241   -7.241   -7.251   -7.261   -7.271   -7.281   -7.291
            -7.301   -7.311   -7.321   -7.331   -7.341   -7.351   -7.361   -7.371
            -7.381   -7.381   -7.391   -7.401   -7.401   -7.411   -7.421   -7.421
            -7.431   -7.431   -7.441   -7.451   -7.451   -7.461   -7.461   -7.471
            -7.481   -7.491   -7.501   -7.511   -7.531   -7.561   -7.581   -7.601
            -7.641   -7.681   -7.721   -7.771   -7.831   -7.901   -7.981   -8.061
            -8.141   -8.221   -8.301   -8.381   -8.320   -8.320   -8.120   -8.120
            -7.920   -7.920   -7.870   -7.870   -7.820   -7.820   -7.120   -7.120
            -7.220   -7.270   -7.420   -7.620   -7.720   -8.320
Attenuation for frequency:  2.000
            -4.984   -6.175   -7.355   -7.485   -7.835   -8.045   -8.185   -8.305
            -8.385   -8.435   -8.475   -8.515   -8.535   -8.545   -8.545   -8.535
            -8.495   -8.385   -8.265   -8.075   -8.055   -8.085   -8.135   -8.225
            -8.325   -8.435   -8.535   -8.635   -8.705   -8.725   -8.705   -8.665
            -8.645   -8.645   -8.635   -8.625   -8.625   -8.625   -8.615   -8.615
            -8.605   -8.605   -8.605   -8.615   -8.625   -8.635   -8.645   -8.655
            -8.665   -8.675   -8.685   -8.695   -8.705   -8.715   -8.725   -8.735
            -8.745   -8.745   -8.755   -8.765   -8.765   -8.775   -8.785   -8.785
            -8.795   -8.795   -8.805   -8.815   -8.815   -8.825   -8.825   -8.835
            -8.845   -8.855   -8.865   -8.875   -8.895   -8.925   -8.945   -8.965
            -9.005   -9.045   -9.085   -9.135   -9.195   -9.265   -9.345   -9.425
            -9.505   -9.585   -9.665   -9.745   -9.684   -9.684   -9.484   -9.484
            -9.284   -9.284   -9.234   -9.234   -9.184   -9.184   -8.484   -8.484
            -8.584   -8.634   -8.784   -8.984   -9.084   -9.684
...
```

2.10 Noise Factor (NFC) Text File

The noise factor file specifies the amount of decayed coda from prior signals that is present as pre-signal noise in secondary phases. The noise factor is represented across a range of distances between the source and receiver. The file is stored as an ASCII text file. A description and example of the parameters within the file and the file format are shown below.

Files in this format are identified by having an "*.nfc*" extension.

Table 26 Noise Factor File Parameters

Parameter	Description
title	Title of the noise factor file
comment	Zero or more comment lines. Each comment line must start with '#'.
n_distances	Integer value representing the number of moments.
distance(i) i = 1..*n_distances*	Distances (degrees) between the source and receiver.
*decay_rate(*i) i = 1..*n_distances*	Decay rate expressed as a fraction.

Table 27 Noise Factor File Format

Format
title
comment
n_distances
distance(1) decay_rate(1)
…
distance(n_distances) decay_rate(n_distances)

Table 28 Noise Factor File Example

Pn to Sn noise factor		
	9	
	2.000	*0.360*
	4.000	*0.340*
	6.000	*0.310*
	8.000	*0.290*
	10.000	*0.270*
	12.000	*0.250*
	14.000	*0.220*
	34.176	*0.000*
	60.000	*0.000*

2.11 Receiver

Table 29 Receiver Simulation Parameters

Parameter	Description
default-station-density	Default station media density (kg/m^3).
default-station-elev	Default station elevation (kilometers).
default-station-rely	Default station reliability expressed as a fraction [0.0 … 1.0].
net	Network name from the *StaNet* table.
list-of-stations	Comma separated list of station names used for simulation

Both the *net* and *list-of-stations* parameters may be used to define the stations to be used for simulation. If *list-of-stations* is defined, then NetMOD will automatically create a network with the name equal to the value of the *net* parameter. If there is already a network with that name, then it will be replaced. If *list-of-stations* is not defined, then NetMOD will automatically assign *list-of-stations* to be equal to the *list-of-network-stations* variable within the *StaNet* table.

2.11.1 StaSpec Table

The *StaSpec* table defines the simulation parameters that are unique to a specific station.

Table 30 StaSpec Table Parameters

Parameter	Description
sta	Station name.
group	Station group name. If empty, this defaults to the station name. Multiple stations may share the same group name. Stations with the same group name are treated as being statistically dependent (i.e. elements of a hydroacoustic station).
lat	Station latitude (degrees).
lon	Station longitude (degrees).
elev	Station elevation (kilometers).
tech	Station technology: S (Seismic), H (Hydoacoustic), and I (Infrasonic).
statype	Station type: ss (single station), ar (array), hydrophone, or 'tphase_sta'.
nchan	Number of channels.
rely	Station reliability expressed as a fraction [0.0 … 1.0].
density	Density of the receiver medium (kg/m^3)

When writing out the *StaSpec* table, NetMOD uses the *sta* as the row label.

2.11.2 StaPhaseSpec Table

The *StaPhaseSpec* table defiles parameters that are unique to a combination of station and phase.

Table 31 StaPhaseSpec Table Parameters

Parameter	Description
sta	Station name.
phase	Phase name.
LogAmpCorr	Station and phase specific correction to the log signal amplitude. If this value is 0, then the default from the *PhaseSpec* table will be used.
LogAmpCorrSD	Station and phase specific standard deviation of the log signal amplitude correction. If this value is 0, then the default from the *PhaseSpec* table will be used.

SnrThresh	Station and phase specific signal to noise ratio threshold required for a detection. If this value is 0, then the default from the *PhaseSpec* table will be used.
deltim	Station and phase specific travel time standard deviation (seconds) used for location simulation. If this value is 0, then the default from the *PhaseSpec* table will be used.
delaz	Station and phase specific azimuth standard deviation (degrees) used for location simulation. If this value is 0, then the default from the *PhaseSpec* table will be used.
velocity	Station and phase specific receiver media velocity (m/s). If this value is 0, then the default from the *PhaseSpec* table will be used.
response	Station and phase specific response (array gain) file. See 2.13 Site Response (SRS) Text File.

When writing out the *StaPhaseSpec* table, NetMOD uses the *sta_phase* as the row label.

2.11.3 StaNet Table

The *StaNet* table allows for multiple network of stations to be defined. The network in use for a simulation may then be selected using the *net* parameter.

Table 32 StaNet Table Parameters

Parameter	Description
net	Network name
list-of-network-stations	Comma separated list of station names that are assigned to this network.

When writing out the *StaNet* table, NetMOD uses the *net* as the row label.

2.11.4 NoiseSpec Table

The *NoiseSpec* table defines the parameters for the station's site noise.

Table 33 NoiseSpec Table Parameters

Parameter	Description
sta	Station name.
logNoiseSD	Standard deviation of the log site noise for this station. The units for this parameter are log(nm) for seismic, log(uPa) for hydroacoustic, and log(Pa) for infrasonic. Note that this value is in units of amplitude and power spectral density.
surg_type	Name of the surrogate station be used for site noise if this station does not have a known noise file or 'none' if this noise file is to be used.
display_name	Display name (unused).
file	Name of the file containing the site noise. See 2.12 Site Noise File for a description of the file format.

When writing out the *NoiseSpec* table, NetMOD uses the *sta* as the row label.

2.12 Site Noise File

The site noise file specifies the amount of ambient background noise that is present at each receiver location. Several formats are available for storing site noise, depending upon the desired level of detail.

2.12.1 NOI Text File

The site noise stored in a NOI Text File is represented as a log PSD (and optional standard deviation) across a range of frequencies. NetMOD interprets frequencies outside of the defined frequency range as having a high noise level of 999.0.

The file is stored as an ASCII text file. A description and example of the parameters within the file and the file format are shown below.

Files in this format are identified by having a "*.noi*" extension.

Table 34 Noise Factor File Parameters

Parameter	Description
title	Title of the site noise file
comment	Zero or more comment lines. Each comment line must start with '#'.
n_frequencies	Integer value representing the number of moments.
frequency(i) i = 1..*n_frequencies*	Frequency (Hz).
*psd(*i) i = 1..*n_frequencies*	Log noise expressed as a power spectral density. The units are $\log_{10}$ nm^2/Hz for seismic, $\log_{10}$ uPa^2/Hz for hydroacoustic, and $\log_{10}$ Pa^2/Hz for infrasonic noise.
std(i) i = 1..*n_frequencies*	Standard Deviation of log noise expressed as a power spectral density. The units are $\log_{10}$ nm^2/Hz for seismic, $\log_{10}$ uPa^2/Hz for hydroacoustic, and $\log_{10}$ Pa^2/Hz for infrasonic noise. The standard deviation is optional.

Table 35 Noise Factor File Format

Format
title *comment* *n_frequenices* *frequency(1) psd(1) [std(1)]* *…* *frequency(n_frequencies) psd(n_frequencies) [std(n_frequencies)]*

Table 36 Noise Factor File Example

```
Peterson NLNM: power spectral density (in displacement,log power,nm^2/Hz)
22
0.000010      24.494   0.0
0.000100      15.619   0.0
0.001667      7.482    0.0
0.003049      6.122    0.0
0.006494      5.059    0.0
0.009901      4.325    0.0
0.014286      3.437    0.0
0.022222      2.670    0.0
0.031646      2.307    0.0
0.045662      2.419    0.0
0.064103      3.367    0.0
0.083333      2.499    0.0
0.100000      2.432    0.0
0.166667      3.020    0.0
0.200000      3.494    0.0
```

0.232558	*3.231*	*0.0*
0.416667	*1.464*	*0.0*
0.806452	*-1.189*	*0.0*
1.250000	*-2.500*	*0.0*
2.500000	*-3.454*	*0.0*
5.882353	*-4.941*	*0.0*
10.000000	*-5.993*	*0.0*

2.12.2 Cumulative Distribution Function Text File

The site noise stored in a Cumulative Distribution Function Text File extends the NOI text file format to include cumulative distribution of the power spectral density. The file is stored as an ASCII text file. A description and example of the parameters within the file and the file format are shown below.

Files in this format are identified by having a "*.noi_cdf*" extension.

Table 37 Cumulative Distribution Function Text File Parameters

Parameter	Description
title	Title of the site noise file
comment	Zero or more comment lines. Each comment line must start with '#'.
n_percentiles	Integer value representing the number of percentiles
percentile(j) *j = 1..n_percentiles*	Percentile levels as a fraction from [0.0 … 1.0].
n_freq	Integer value representing the number of moments.
frequency(i) i = 1..*n_freq*	Frequency (Hz).
*psd(*i) i = 1..*n_freq*	Log noise expressed as a power spectral density. The units are $\log_{10}$ nm^2/Hz for seismic, $\log_{10}$ uPa^2/Hz for hydroacoustic, and $\log_{10}$ Pa^2/Hz for infrasonic noise.
std(i) i = 1..*n_freq*	Standard Deviation of log noise expressed as a power spectral density. The units are $\log_{10}$ nm^2/Hz for seismic, $\log_{10}$ uPa^2/Hz for hydroacoustic, and $\log_{10}$ Pa^2/Hz for infrasonic noise. The standard deviation is optional.
cdf(i.j) i = 1..*n_freq* *j = 1..n_percentiles*	Cumulative distribution values for each frequency and percentile. The units are $\log_{10}$ nm^2/Hz for seismic, $\log_{10}$ uPa^2/Hz for hydroacoustic, and $\log_{10}$ Pa^2/Hz for infrasonic noise.

Table 38 Cumulative Distribution Function Text File Format

Format
title *comment* *n_percentiles* *percentile(1) … percentile(n_percentiles)* *n_frequenices* *frequency(1) psd(1) std(1) cdf(1,1) … cdf(1,n_percentiles)* … *frequency(n_freq) psd(n_freq) std(n_freq) cdf(n_freq,1) cdf(n_freq,n_percentiles)*

Table 39 Cumulative Distribution Function Text File Example

```
power spectral density (in displacement,log power,nm^2/Hz)
#  Percentiles every 10% from 0% to 100%
11
 0.0  0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9  1.0
22
0.000010 24.49  0.07172   24.42   24.45   24.46   24.47   24.47   24.51   24.52   24.52   24.52   24.53   24.54
0.000100 15.62  0.163     15.51   15.54   15.59   15.6    15.61   15.69   15.73   15.77   15.78   15.87   15.93
0.001667 7.482  0.04889   7.422   7.439   7.474   7.493   7.497   7.498   7.498   7.5     7.525   7.527   7.613
0.003049 6.122  0.1035    5.955   5.964   6.037   6.141   6.157   6.177   6.179   6.225   6.31    6.312   6.331
0.006494 5.059  0.07269   4.948   4.971   4.985   5.007   5.017   5.058   5.062   5.067   5.069   5.071   5.089
0.009901 4.325  0.03034   4.298   4.307   4.314   4.326   4.338   4.342   4.343   4.346   4.353   4.355   4.356
0.014286 3.437  0.02939   3.402   3.42    3.422   3.424   3.429   3.431   3.432   3.434   3.462   3.468   3.473
0.022222 2.67   0.07873   2.478   2.519   2.555   2.596   2.617   2.62    2.636   2.64    2.661   2.694   2.863
0.031646 2.307  0.08884   2.197   2.261   2.299   2.312   2.318   2.329   2.341   2.355   2.402   2.431   2.451
0.045662 2.419  0.1147    2.227   2.301   2.308   2.34    2.363   2.371   2.437   2.459   2.465   2.514   2.592
0.064103 3.367  0.1069    3.295   3.329   3.376   3.377   3.382   3.389   3.403   3.407   3.467   3.536   3.651
```

0.083333	2.499	0.08095	2.308	2.369	2.442	2.486	2.487	2.518	2.526	2.564	2.57	2.583	2.633
0.100000	2.432	0.2944	1.816	2.179	2.255	2.264	2.381	2.419	2.425	2.438	2.572	2.688	2.816
0.166667	3.02	0.1438	2.832	2.846	2.855	2.866	2.973	3.011	3.03	3.149	3.235	3.282	3.349
0.200000	3.494	0.03252	3.449	3.45	3.465	3.488	3.494	3.499	3.504	3.506	3.515	3.531	3.552
0.232558	3.231	0.07549	3.082	3.18	3.184	3.201	3.22	3.23	3.265	3.27	3.294	3.303	3.317
0.416667	1.464	0.137	1.175	1.237	1.282	1.389	1.477	1.479	1.507	1.523	1.525	1.594	1.616
0.806452	-1.189	0.1712	-1.396	-1.393	-1.295	-1.294	-1.256	-1.196	-1.18	-1.146	-1.096	-1.089	-0.8667
1.250000	-2.5	0.01022	-2.519	-2.509	-2.507	-2.507	-2.504	-2.499	-2.492	-2.487	-2.485	-2.484	-2.478
2.500000	-3.454	0.02414	-3.506	-3.496	-3.495	-3.49	-3.481	-3.472	-3.455	-3.438	-3.436	-3.425	-3.421
5.882353	-4.941	0.03192	-4.986	-4.985	-4.94	-4.94	-4.915	-4.914	-4.91	-4.902	-4.891	-4.89	-4.885
10.000000	-5.993	0.03129	-6.056	-6.021	-6.02	-6.008	-6.004	-5.997	-5.981	-5.98	-5.967	-5.961	-5.953

2.12.3 NetCDF File

The site noise may be stored in a NetCDF (Network Common Data Form not to be confused with Cumulative Distribution Function) binary file. The NetCDF format allows for storing a receivers ambient site noise as a cumulative distribution function (CDF) of the power spectral density for multiple time periods.

NetMOD is able to utilize the mean and standard deviation of the power spectral density when performing probabilistic simulations. When performing Monte Carlo simulations, NetMOD can take advantage of the CDF definition of the power spectral density since there is no longer an assumption that the variables being simulated are normally distributed. In addition, when performing a simulation of a specific time period, NetMOD will automatically select the closest matching time of day and day of year contained within the NetCDF file.

NetCDF is an open standard that consists of a set of interfaces, libraries, and platform-independent formats for representing scientific data. Additional information about NetCDF may be obtained from:

http://www.unidata.ucar.edu/software/netcdf/

NetMOD supports the use of site noise contained within NetCDF v4 format, provided the data is organized with the following structure:

Global Attributes	**Value**
station_name	String containing the station name.
data_type	String containing the type of data: 'seis'.
station_latitude	Double value containing the station latitude in degrees.
station_longitude	Double value containing the station longitude in degrees.
units	String containing the unit of the data: 'db re (m^2/s^4)/Hz'
spectral_window_type	String containing the type of spectral window used
spectral_window_equivalent_noise_bandwidth	Double value containing the noise bandwidth
spectral_window_relative_ovelap_value	Double value containing the fractional overlap of the windows
spectral_main_window_length	Double value containing the spectral window length in seconds
smoothing	String containing a description of the smoothing algorithm
code_version	Version of the CTBTO/IDC code used to generate the noise.

An example of the global attributes is shown below:

```
Global Attributes:
  station_name                               = 'ASAR'
  data_type                                  = 'seis'
  station_latitude                           = -23.7
  station_longitude                          = 134
  units                                      = 'db re (m^2/s^4)/Hz'
  spectral_window_type                       = 'nutall4a'
  spectral_window_equivalent_noise_bandwidth = 2.13
```

```
  spectral_window_relative_ovelap_value      = 0.68
  spectral_main_window_length                = 3.6e+003
  smoothing                                  = 'Savitzky-Golay filter: 11pt
6order'
  code_version                               = '1.0.0'
```

Dimensions	**Value**
STRLEN	Integer length limit of string, 0 is unlimited.
number_of_sensors	Number of sensors elements contained within this NetCDF file
number_of_percentiles	Number of percentiles that make up each Cumulative Distribution Function (CDF).
samples	Number of frequency samples that make up each of the noise definitions

An example of the dimensions is shown below:

```
Dimensions:
           STRLEN                = 0     (UNLIMITED)
           number_of_sensors     = 22
           number_of_percentiles = 21
           samples               = 100
```

Global Variables	**Value**
percentiles	A NetCDF variable containing an array of the percentile values that define each CDF. These values typically range from 0 to 100 in 5% increments.
frequencies	A NetCDF variable containing an array of the frequency values that define each CDF.
sensors	A NetCDF variable containing an array of the identifiers for the sensors contained within this NetCDF file.

An example of the global variables is shown below:

```
Variables:
    percentiles
           Size:        21x1
           Dimensions: number_of_percentiles
           Datatype:   double
    frequencies
           Size:        100x1
           Dimensions: samples
           Datatype:   double
    sensors
           Size:        22x1
           Dimensions: number_of_sensors
           Datatype:   UNSUPPORTED DATATYPE
```

There is a specific hierarchy of group definitions within the NetCDF noise file format that must be adhered to. The top level group has the same station name as is defined in the corresponding global attribute:

```
      /<station_name>/
```

Below the top level group, there can be any number of arbitrarily named groups that represent the site noise at different time spans:

```
/<station_name>/<Time Span 1>/

/<station_name>/<Time Span 2>/

…
```

Each of the time spans contains a "*time_attributes*" variable that specifies the time span for which this group applies:

```
/<station_name>/<Time Span 1>/
          Variables:
              time_attributes
                     Size:       []
                     Dimensions: STRLEN
                     Datatype:   int32
                     Attributes:
                                 begin_time = '2010001 02:34:22'
                                 end_time   = '2010031 03:34:22'
                                 month      = 'January'
```

The "*begin_time*" and "*end_time*" attributes each contain two parts: A concatenated year, day of year, and time representing a GMT hour:minutes:second time. The begin day of year and end day of year define the range of days of the year that this noise file spans. The begin time and end time represents the range of times of the day that this noise file spans. In this way, the NetCDF file can contain multiple noise definitions for different times of day that each span a larger number of days, such as a month.

Within each time span group, there can be any number of arbitrarily named groups for each sensor element of the station. However, NetMOD only makes use of the first group that is present. There are four variables that are present in each of the sensor element groups:

Variables	**Value**
log_PSD	A NetCDF variable containing an array of the average log PSD value in units of $\log_{10}$ ($(nm/s^2)^2/Hz$) at each of the defined frequencies.
log_PSD_std	A NetCDF variable containing an array of the standard deviation log PSD value in units of $\log_{10}$ ($(nm/s^2)^2/Hz$) at each of the defined frequencies.
PDF	A NetCDF variable containing an array of the PSD values in units of dB ($(m/s^2)^2/Hz$) of acceleration at each frequency and CDF level.
sensor_attributes	A number of attributes about the sensor. See the example below.

```
/<station_name>/<Time Span 1>/<Sensor element 1>
    Variables:
```

```
log_PSD
       Size:       100x1
       Dimensions: samples
       Datatype:   double
log_PSD_std
       Size:       100x1
       Dimensions: samples
       Datatype:   double
PDF
       Size:       100x21
       Dimensions: samples,number_of_percentiles
       Datatype:   double
sensor_attributes
       Size:       []
       Dimensions: STRLEN
       Datatype:   int32
       Attributes:
                   sensor_name                = 'AS01'
                   channel_name               = 'SHZ'
                   decon_file                 = 'filename'
                   recon_file                 = 'none'
                   calib                      = 0.00925
                   calper                     = 1
                   spectral_sub_window_length = 500
                   number_in_average          = 28
```

Example NetCDF noise files are provided with the distribution of NetMOD and may be used as a template for generating additional NetCDF noise files.

2.13 Site Response (SRS) Text File

The site response file specifies the array gain due to beam forming. The response is specified across a range of frequencies. NetMOD interprets frequencies outside of the defined frequency range as having an array gain of -999.0.

A description and example of the parameters within the file and the file format are shown below.

Files in this format are identified by having a "*.srs*" extension.

Table 40 Site Response File Parameters

Parameter	Description
title	Title of the site noise file
comment	Zero or more comment lines. Each comment line must start with '#'.
n_frequencies	Integer value representing the number of moments.
frequency(i) i = 1..*n_frequencies*	Frequency (Hz).
*gain(*i) i = 1..*n_frequencies*	Log amplitude of the site response. The array gain is unit-less.

Table 41 Site Response File Format

Format
title *comment* *n_frequenices* *frequency(1) gain(1)* … *frequency(n_frequencies) gain(n_frequencies)*

Table 42 Site Response File Example

```
Reg FINESA-type array (a=1.6, n=9)
22
  0.250    0.017
  0.750    0.255
  1.250    0.366
  1.750    0.439
  2.250    0.477
  2.750    0.477
  3.250    0.477
  3.750    0.477
  4.250    0.477
  4.750    0.477
  5.250    0.477
  5.750    0.477
  6.250    0.477
  6.750    0.460
  7.250    0.445
  7.750    0.430
  8.250    0.417
  8.750    0.404
  9.250    0.392
  9.750    0.380
```

10.250	*0.370*
20.000	*0.227*

2.14 Output

Parameter	Description
Output-Dir	Directory for writing output files.

Within the output directory, NetMOD will save the files generated for any simulations that are run. A simulation output is saved as a set of three ASCII text files: A key file, a station file, and a grid file.

2.14.1 Simul Key File

The Simul Key file contains some of the key parameters that were used to generate the simulation. The intent is that the parameters in the key file can approximate the state of the simulation. The Simul Key parameters share the same naming and definitions as the simulation parameters from which they original. The key file is identified as ending with a "*.simul_key*" extension and is an ASCII text file with a single parameter per line:

<parameter> = <value>

Output Parameter
Simul-Baseline
Simul-Technology
Run-Type
Sub-Type
Freq-Sampling
Freq-Type
Signal-Amplitude
Noise-Amplitude
Event-Detection-Criteria
Wind-Mode
Wind-Day-Of-Year
Wind-Time-Of-Day
Size-Type
Event-Size
Min-Event-Size
Max-Event-Size
Conf
Num-MonteCarlo-Iter
Epi-Model
EpiGrid-MinLat
EpiGrid-MaxLat
EpiGrid-MinLon
EpiGrid-MaxLon
EpiGrid-DelLat
EpiGrid-DelLon
Min-Epi-Size
Max-Epi-Size
net

2.14.2 Simul Sta File

The Simul Sta file contains the set of stations that were used to generate the simulation. The intent is that the parameters in the key file can approximate the state of the simulation. A Simul Sta file is identified as ending with a "*.simul_sta*" extension and is an ASCII text file with a single station per row and the following columns:

Table 43 Simul Sta Parameters

Parameter	Description
sta	Station name.
lon	Station longitude (degrees).
lat	Station latitude (degrees).
rely	Station reliability expressed as a fraction [0.0 … 1.0].
nchan	Number of channels.
statype	Station type: ss (single station), ar (array), hydrophone, or 'tphase_sta'.

Table 44 Simul Sta File Format

```
sta(1) tech(1) lon(1) lat(1) rely(1) nchan(1) statype(1)
:
:
Sta(n) tech(n) lon(n) lat(n) rely(n) nchan(n) statype(n)
```

2.14.3 Simul Grid File

The Simul Grid file contains the simulation output at each of the epicenter source locations. A Simul Grid file is identified as ending with a "*.simul_grid*" extension. The file is an ASCII text file with a single epicenter result per row

The key file is an ASCII text file with the following parameters:

Table 45 Simul Grid Parameters

Parameter	Description
num_epicenters	Number of epicenters.
lat	Epicenter latitude (degrees).
lon	Epicenter longitude (degrees).
size	Event size.
probability	Network detection probability.
fracloc	Fraction of the Monte Carlo iterations with sufficient data to constrain location.
smaj1	Length (km) of the semi-major axes determined by the confidence parameter *Pct-Lo.*
smin1	Length (km) of the semi-minor axes determined by the confidence parameter *Pct-Lo.*
strk1	Strike (degrees from North) of the semi-major axes determined by the confidence .parameter *Pct-Lo.*
depth1	Depth uncertainty (km) determined by the confidence parameter *Pct-Lo*.
smaj2	Length (km) of the semi-major axes determined by the confidence parameter *Pct-Hi.*
smin2	Length (km) of the semi-minor axes determined by the confidence parameter *Pct-Hi.*
strk2	Strike (degrees from North) of the semi-major axes determined by the confidence .parameter *Pct-Hi.*
depth2	Depth uncertainty (km) determined by the confidence parameter *Pct-Hi*.

Table 46 Simul Grid File Format

```
num_epicenters
lat(1) lon(1) size(1) probability(1) . . .
:
:
lat(n) lon(n) size(n) probability(n) . . .
```

3 REFERENCES

SAIC-09/3007, NetSim Control Parameters and Output Data, Science Applications International Corporation Technical Report SAIC-09/3007, 2009.

Sereno, T.J., S.R. Bratt, and G. Yee (1990). NetSim: a computer program for simulating detection and location capability of regional seismic networks, SAIC Annual Technical Report SAIC90/1163.

DISTRIBUTION

1	MS0899	Technical Library	9536 (electronic copy)

Sandia National Laboratories