Raytracer
v5.3.14
Documentation

Tsunami Development
tsunamidevelopment.com
713-783-1435

Tsunami Development
713-783-1435
tsunamidevelopment.com

CONTENTS

Raytracer Introduction

Raytracer is a ray tracing program for generating travel times for the PSDM algorithm. It advances the individual rays using a fourth order Runga-Kutta solution and supports three methods for selecting the times, minimum time, minimum distance and maximum energy. Raytracer uses the wavefront reconstruction method to maintain accuracy as the rays move through the velocity model. It provides user control for interpolation and smoothing of the travel times, as well as for the density of recording locations. Virtually any size velocity model can be used since it does not need to fit into memory. Though if the velocity model is much greater than the memory on the compute nodes, the performance of the software will suffer.

Raytracer advances the individual rays as members of beams, or triangular sets of three adjacent rays. As the rays advance through the model, the individual ray times are extrapolated to a common point within the beam, and are then compared with each other. If the times deviate by more than the allowed amount a new ray and associated beam are created. Therefore, as the rays scatter, the accuracy of the ray front is maintained.

The step size for advancing a given ray is determined automatically, and is adjusted based on the local velocity gradient. As the velocity gradient becomes larger the step size becomes smaller. This is done because the error in the Runga-Kutta equation increases as the angle change, dependent on delta v, becomes greater.

The rays advance radially from the shot location, with all rays within a beam advancing to the next step. At that point a search is made for the grids that are within each beam. The travel time is then calculated for any grids within a given beam. If a grid is within multiple beams, then the selection criteria are used, minimum time, minimum distance, or maximum energy, to determine which beam to use to calculate the travel time.

You can use multiple Raytracer output files with a single PSDM job. You might have multiple files because of file sizes or because you have different shot spacing for different areas of the survey.

All travel time interpolations, both within Raytracer and within PSDM, are done using slowness rather than seconds. This greatly improves the accuracy of the interpolation and allows the increments between shots and subsurface locations to be larger without a loss of accuracy.

This application may be run either from the application GUI, or from the command line. If run from the GUI, the GUI will build a parameter file, and then execute the job using the parameter file. You may alternatively create the parameter file by hand and execute the job from the command line. Both workflows will be presented in this document. For each step, there will be a detailed description of the GUI and each parameter it needs followed by a boxed section describing the same parameters as they would appear in the parameter file if using the program by the command line.

Frequently Asked Questions

- **I have java 1.4.2 or higher defined in my .cshrc (.bashrc) file but the GUI doesn't recognize it.**

  1) Look for the following line in your .cshrc (.bashrc) file:

  source ~ Epos3_env

  This will override the java that you have set as $JAVA_HOME and use the java for Paradigm / Geodepth.

  2) Some versions of Linux OS have a place holder for java. To override this, put your $JAVA_HOME/bin path first in your $PATH definition in your .cshrc/.bashrc file.

- **My node.db file won't display in the Node Selection window of GUI.**

  Check that there are no hidden characters in your node.db file with a text editor. If your node.db file was created on a pc , you may need to use the dos2unix command on the file.

- **I keep getting the following error:** OS Error: could not map rld from file /lib32/rld

  There are not enough file descriptors available. To resolve this issue:

  in csh, do following command: limit descriptors 500
  verify with: limit

  in bash, do following command: ulimit –n 500
  verify with: ulimit –a

- How do I set up rsh for my system?

  http://evuraan.blogspot.com/2005/02/how-to-turn-on-rsh-and-rlogin-on.html will help you get started.

- I've issued the start command from the GUI but the job doesn't run

  Things to check:

  1) Test rsh: rsh node_name date

  2) Test rcp: rcp some_file node_name: /tmp

3) Check xterm where GUI was launched to see command that was issued.

   If you see “null” in the command line, there may be a typo in your node.db file, hidden characters or some other issue with the node names.

4) Make sure you have the correct node.db file set in your parameter file.

5) Node names must be in sync. Check /etc to make sure names are correct.

Installation

System Requirements

Tsunami will run on any combination of Linux, SGI, Opteron, Solaris and Itanium systems.
Linux must be: version 2.4 or greater, with the gcc compiler
SGI must be: 6.2 of IRIX or greater
Itanium: must have the Intel v8.0 compiler

Tsunami is very flexible such that all systems within a cluster need not be running the same version of the operating system. Even different versions of Linux can be accommodated by using the features of the node database file.
Please contact Tsunami Development for if you need to mix multiple versions of Linux operating systems.

The amount of memory suggested is 256 MBytes per processor for PSTM, and 512 MBytes per processor for PSDM and Raytracer. Smaller amounts may work with some performance penalty, or if the jobs to be run are small.

No disk space is required on any of the compute nodes, only the master node needs to be able to see the data files and the file system for the output files. The /tmp file system needs to exist on the compute nodes and permissions need to be open to write to /tmp. Tsunami will put the executable and local logfiles for the compute nodes in /tmp.

It's suggested that NFS not be used within the cluster. NFS is not used by Tsunami, and can cause problems as the clusters get larger. This is especially true as the cluster exceeds 100 processors, as NFS can cause the systems to hang.

The rsh and rcp commands must be enabled. Tsunami uses the rsh and rcp commands to copy and start the executable on the compute nodes. Therefore permissions must be set, so that rsh and rcp can work in the users accounts. You can test the rsh command by typing the following at the prompt to get the current date:

Example: <prompt> : rsh < node name > date

You can test the rcp by copying a file to the node's /tmp directory:

Example: <prompt>: rcp <file> node_name: /tmp

For most installations a 100 Mbit network is sufficient to support Tsunami. Each node should be on a 100 Mbit switch. This will be sufficient for most jobs.

Installation of Tsunami

1. Obtain tar file of application from the ftp site provided by Tsunami Development.

2. Change directory to the apps directory.

   Example: <prompt>: cd /apps

3. Create a directory named tsunami.

   Example: <prompt>: mkdir tsunami

4. Copy the tar file into the tsunami directory created in step 3.
   Untar the Tsunami tar file.

   Example: tar -xvpf tsunami.tar

You should now have the following structure:

```
apps/tsunami/tsunami_5.3.14/pstm_5.3.14
                            psdm_5.3.14
                            rays_5.3.14
                            tomo_5.3.14
```

Your license file will control whether or not you can run all executables. Your final directory structure should look like the following:

```
/apps/tsunami/tsunami_5.3.14/rays_5.3.14/itanium
                                         itanium_3.5
                                         linux
                                         linux_3.4
                                         opteron
                                         opteron32
                                         opteron_3.4
                                         sgi
```

5. Obtain license file from Tsunami Development via email. Save license file in the TSUNAMI directory created in step 3. Please see the *Floating License* section for more information on license files.

GUI Installation

1. No installation of the GUI is required. In order to run the GUI (/apps/tsunami/tsunami_5.3.14/tsunami.jar), you must have Version 1.4.2 or higher of Java 2 Platform, Standard Edition (J2SE) installed. To determine your java information use the following commands:

   a) java –version
      This will give you the version number of your java installation.

   b) which java
      This will tell you where your java installation is located.

   Should these commands return no information or your have a lower version of java, please have your system administrator ensure that the correct version of java is installed on your machine and the JAVA_HOME environment variable is set in the users .cshrc or .bashrc file. Should you need to install java, please see *Appendix C: Java Installation* for more information.

2. Edit the users ".cshrc" or ".bashrc" file to include the following variables:

   a) If the environment variable JAVA_HOME does not already exist, add to .cshrc or .bashrc file.

      For .bashrc: export JAVA_HOME="java directory"
      Where java directory is the java install directory determined in step 1.

      For .cshrc: setenv JAVA_HOME java directory
      Where java directory is the java install directory determined in step 1.

   b) Add JAVA_HOME/bin to PATH variable.

      For .bashrc: PATH is located in the .bash_profile file. Add JAVA_HOME/bin to the end of the existing PATH variable.
      Ex: export PATH=$PATH:$HOME/bin:$JAVA_HOME/bin

      For .cshrc: PATH is located in the .login file.
      Add the JAVA_HOME directory/bin to the end of the existing set path= variable.
      Ex: set path=(/bin /usr/bin /sbin /usr/etc /usr/local/bin /usr/java2/bin)
      Where /usr/java2 is JAVA_HOME

c) Add an environment variable TSUNAMI that points to the tsunami installation directory.

For .bashrc: export TSUNAMI="/apps/tsunami"

For .cshrc: setenv TSUNAMI /apps/tsunami

d) Add an alias that points to the tsunami GUI executable.

For .bashrc: alias tsunami="/apps/tsunami/tsunami_5.3.14/tsunami.sh"

For .cshrc: alias tsunami /apps/tsunami/tsunami_5.3.14/tsunami.sh

Floating License

The license file registers the nodes that are licensed for the products. Up until 3.1.7 the user could run the application on any and all nodes licensed in the file. With the floating license, the user may have all their nodes in the license file but only check out how many nodes that they have purchased. With version 4.15.9, cpu's will also be taken into consideration to accommodate those users who have more than 2 cpu's per system.

The floating license allows the user to have any number of nodes in the license file, and then license the number that they purchased. The software will check out the licenses from the license file when a job runs, and check in the license when the job completes, or is aborted. Any number of jobs can run on the nodes that are checked out, but only the number allowed by the license can be checked out. An example would be to have 50 nodes, license 20, and be able to use any 20 out of those 50 at a given time.

For the floating cpu license, if only one job is running on a node the license manager will only check out the number of cpus specified by the node.db file. If multiple jobs are running on a node then all cpus licensed for that node will be checked out and a warning will be printed to the logfile.

If not all the cpus on a node are licensed then the user will not be allowed to specify more cpus in the node.db file than are licensed for that node. If not all cpus are licensed for a node, then the user will not be able to run multiple jobs on the node. The software checks the number of physical cpus on the node, and compares it to the number licensed, and the number in the node db file.

You can see the status of the license file by running the utility: check_license_file.
It will give you a list of the node name, the mac address, the expiration date, the number of cpus licensed for each node and how many jobs have the node checked out.

Example: <prompt>: check_license_file <license file name>

If for some reason a job fails to check in the license you can reset the license file for a group of nodes by running clear_nodes. Clear nodes will kill jobs running on the nodes listed in the node.db file you submitted to clear_nodes, as well as reset the licenses for only those nodes in the node.db submitted to the clear_nodes command. Clear nodes has been changed for 3.1.7 and no longer uses a user id for an argument, it will now clear all Tsunami processes on the nodes requested. The clear_nodes will now create a clear_nodes.log file in $TSUNAMI.

Example: <prompt>: clear_nodes < node db file >

The location of the license file has changed from 3.1.6. It is now in the /apps/tsunami directory instead of the application directory.

Cluster Configuration

A node description file must be created for tsunami. This node db will be the master node list. It is an ASCII list of the nodes for the job in the following format:

node name	number of processors	memory in Mbytes	speed factor	operating system

The node name is the name of the system as listed in the /etc/hosts file.

The speed factor is the relative speed of the processor compared with the other nodes. The speed factor and the number of processors are used to balance the workload between nodes. This allows one to mix nodes with different number of processors, and different speeds in the same cluster.

Typically the fastest processors are given a factor of 1.0, and slower ones numbers less than one. For instance, a 3000 MHz processor might have a factor of 1.0, and a 1500 MHz processor a factor of .5.

At the end of the log file for a job, statistics are provided for each node. You should use the relative values for "Millions of shift and sums per second kernel time" as the factor. This gives the speed of migrating the data when the code is executing the kernel.

The operating system tells the software which one of the executables to use. The supported operating systems are linux, sgi, solaris, opteron and itanium (linux).

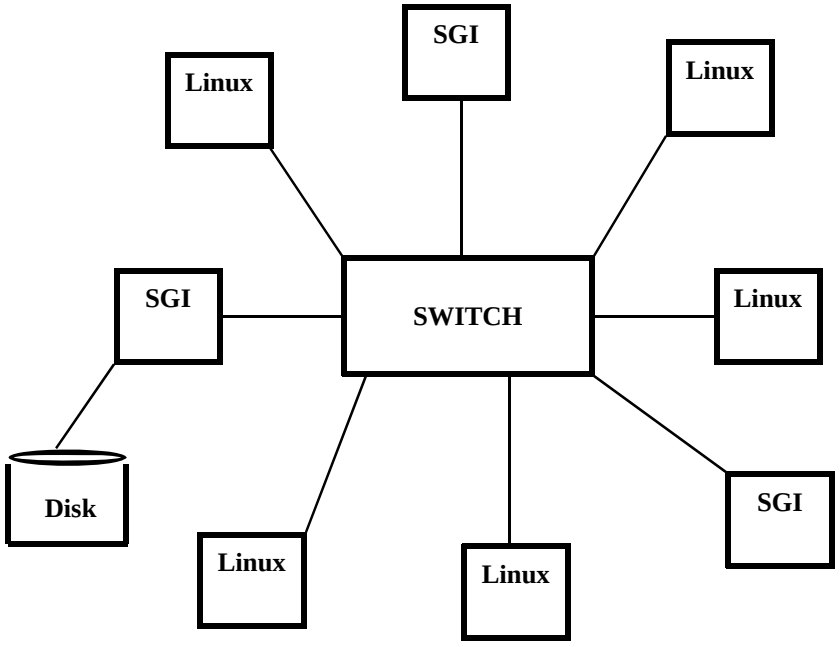
The server node must be the first entry in the db. This is the node from which the job is launched, and is the only node that must have access to the input data, and the file system where the output data will be written. The server may also be a compute node, in that case the server should be listed twice in the first two lines of the db file.

The following is an example of a node db, there is one record per line with spaces as the delimiter. You can use the # sign to comment out a line if necessary for a particular job.

server1	2	512	1.0	linux
linux1	2	512	1.0	linux
linux2	2	512	1.0	linux
sgi1	4	512	.75	sgi
itanium1	8	1000	.65	itanium

In order for the program to execute, it must be possible to issue a remote shell from the node where the program is initiated to all the nodes in the node database. If you have questions about this see your system administrator or contact Tsunami Development. There also must be a /tmp directory on all the nodes. This is where the executable is copied to when the program begins. The executable is removed at the end of the job.

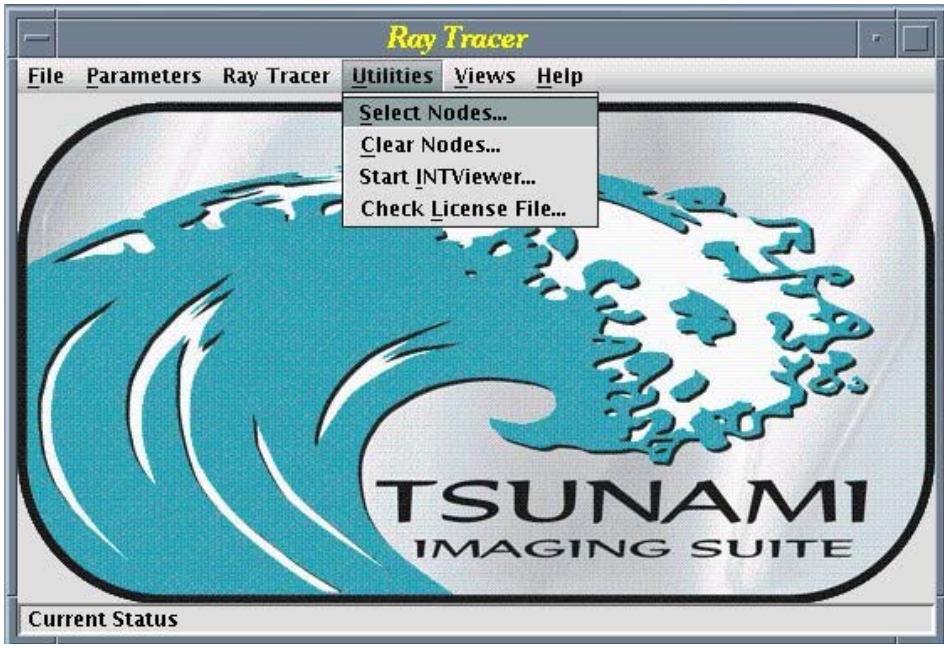
Example Cluster Configuration



Node Utility

Once the master node database has been created, you can use the Select Node Utility from the GUI. *See Getting Started for GUI start up steps.

1. Select **Utility** —> **Select Nodes**.



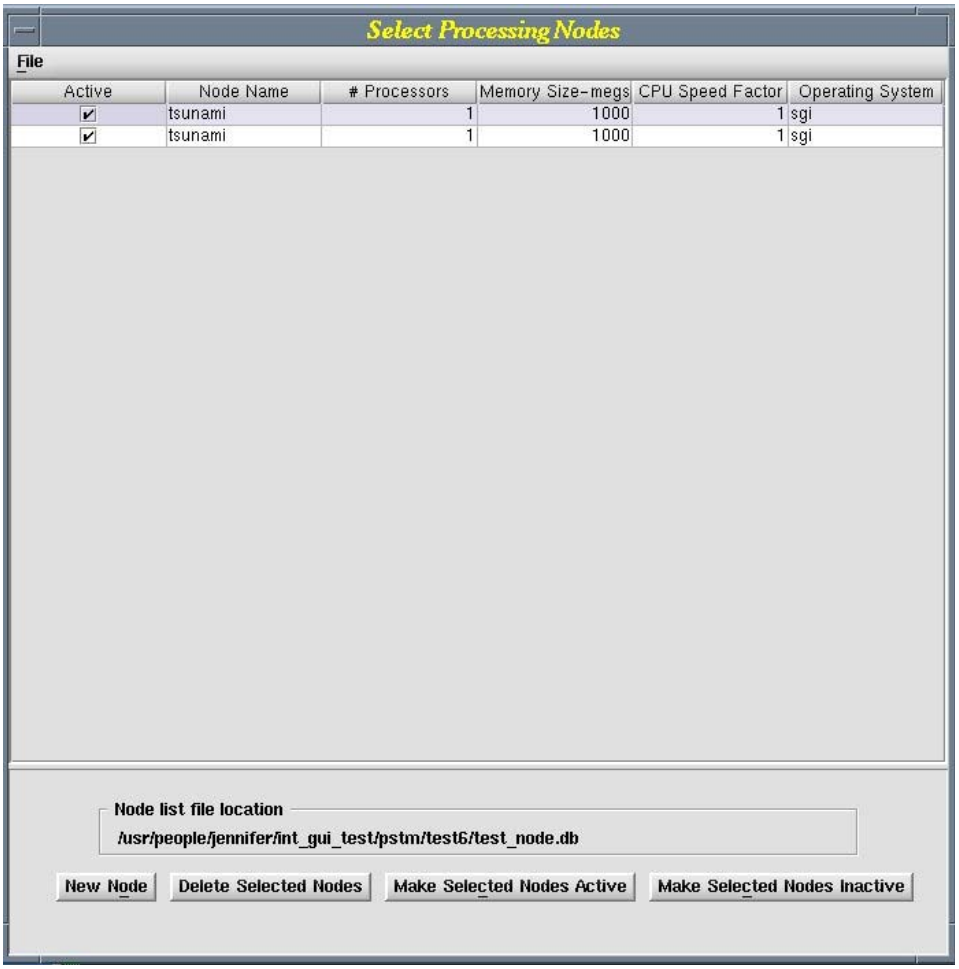
The select nodes dialog box opens and displays node files, as well as creates new node files.

To open a node list file or to create a new one, the program must first search for a master node list file. The master node list should be located in $TSUNAMI. Where $TSUNAMI = the tsunami directory - such as "/apps/tsunami". If no master node list is found, the user will then be prompted to enter the correct path (where the master node list resides) into the Application Directory (hmdr) on the files panel of the parameters edit box.

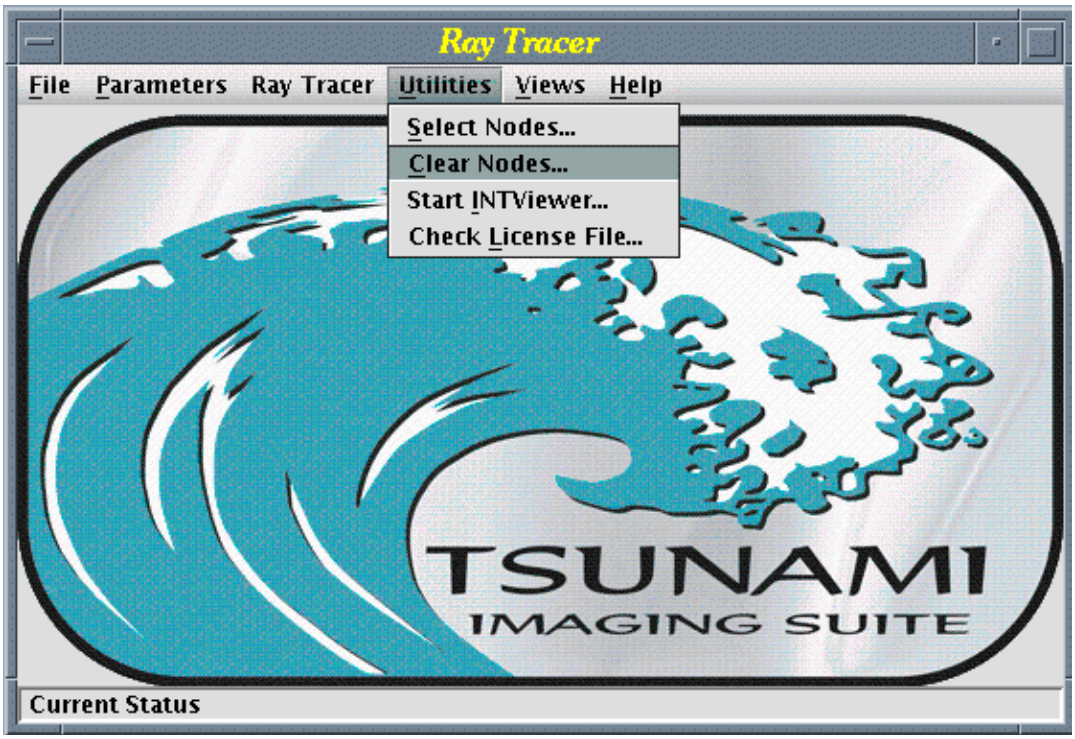
If a master node list is found and a project node list file is supplied in the node list (nddb) parameter of the parameter file (located on the processing panel of the parameters edit box), the dialog box will open and display the selected nodes. If no project node list is supplied, the dialog box will open and the user can select from the available nodes in order to create a new node list file.

When the user selects to save the node list file, the current file will automatically be entered into the node list (nddb) parameter of the parameter file (located on the processing panel of the parameters edit box).

For more information and help on the node list file and scaling a job on multiple nodes see the *Performance Considerations and Advanced Features* section.



2. When a job is aborted or is stopped due to a hardware problem, Tsunami makes every effort to kill all the processes on the compute nodes. Sometimes it is not successful in killing all orphans so you must use the Clear Nodes Utility. Clear nodes will kill jobs running on the nodes listed in the node.db file you submitted to the clear_nodes command, as well as reset the licenses for only those nodes in the node.db submitted to the clear_nodes command. Clear nodes will not kill any processes on the master node. It is very likely that processes on the master node will abort as a result of killing the processes on the compute nodes, however you may need to verify this. Please see the *Floating License* section for more information. The clear nodes utility will create a clear_nodes.log file in $TSUNAMI.



Clear Nodes from Command Line:

At the prompt type:

clear_nodes < node database file >

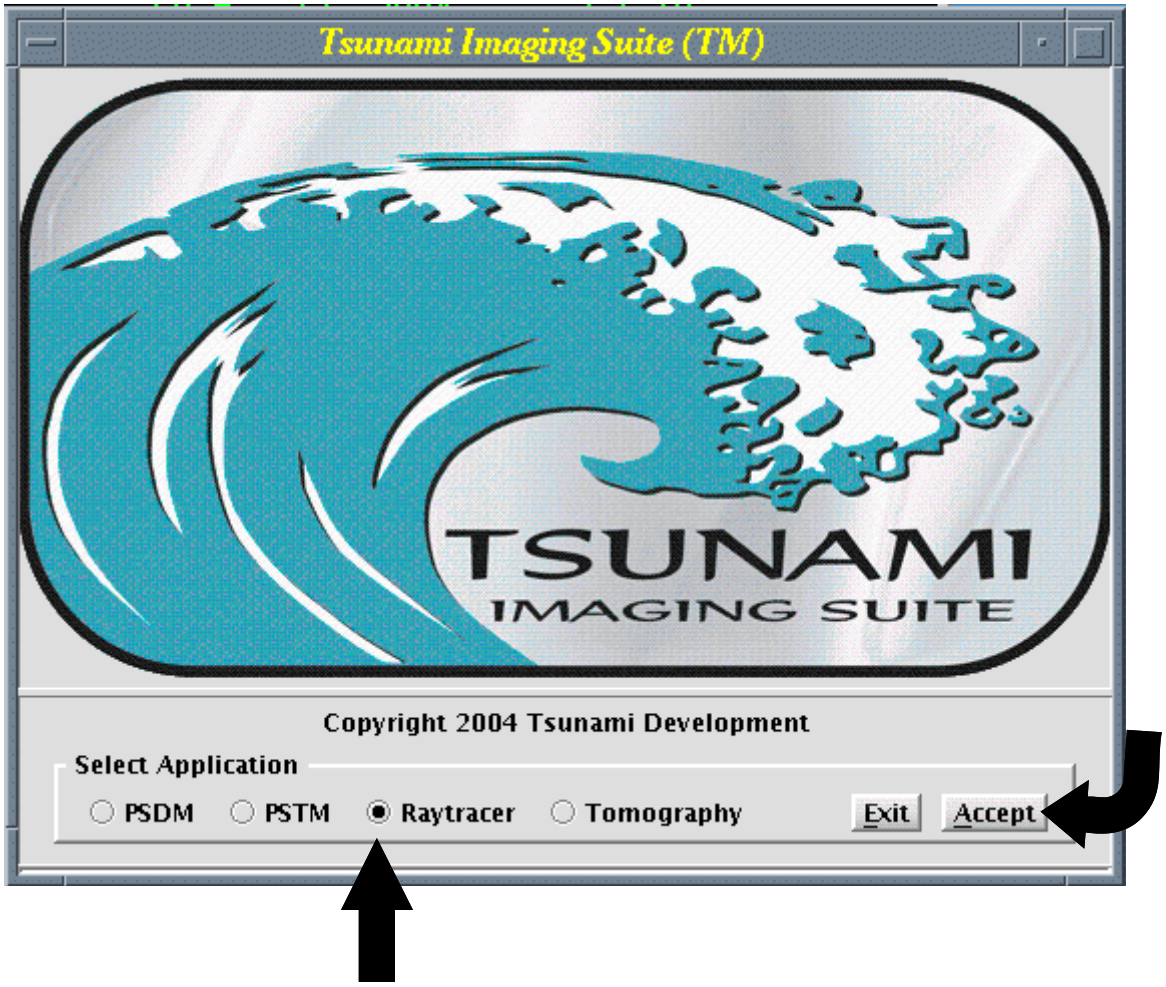
Clear nodes will kill jobs running on the nodes listed in the node.db file you submitted to clear_nodes, as well as reset the licenses for only those nodes in the node.db submitted to the clear_nodes command. Clear nodes will not kill any processes on the master node. It is very likely that processes on the master node will abort as a result of killing the processes on the compute nodes, however you may need to verify this. Please see the *Floating License* section for more information.

Getting Started

1. Use the alias set up in step 2 of the GUI Installation section to start the Tsunami GUI.

   Example: <prompt>: tsunami

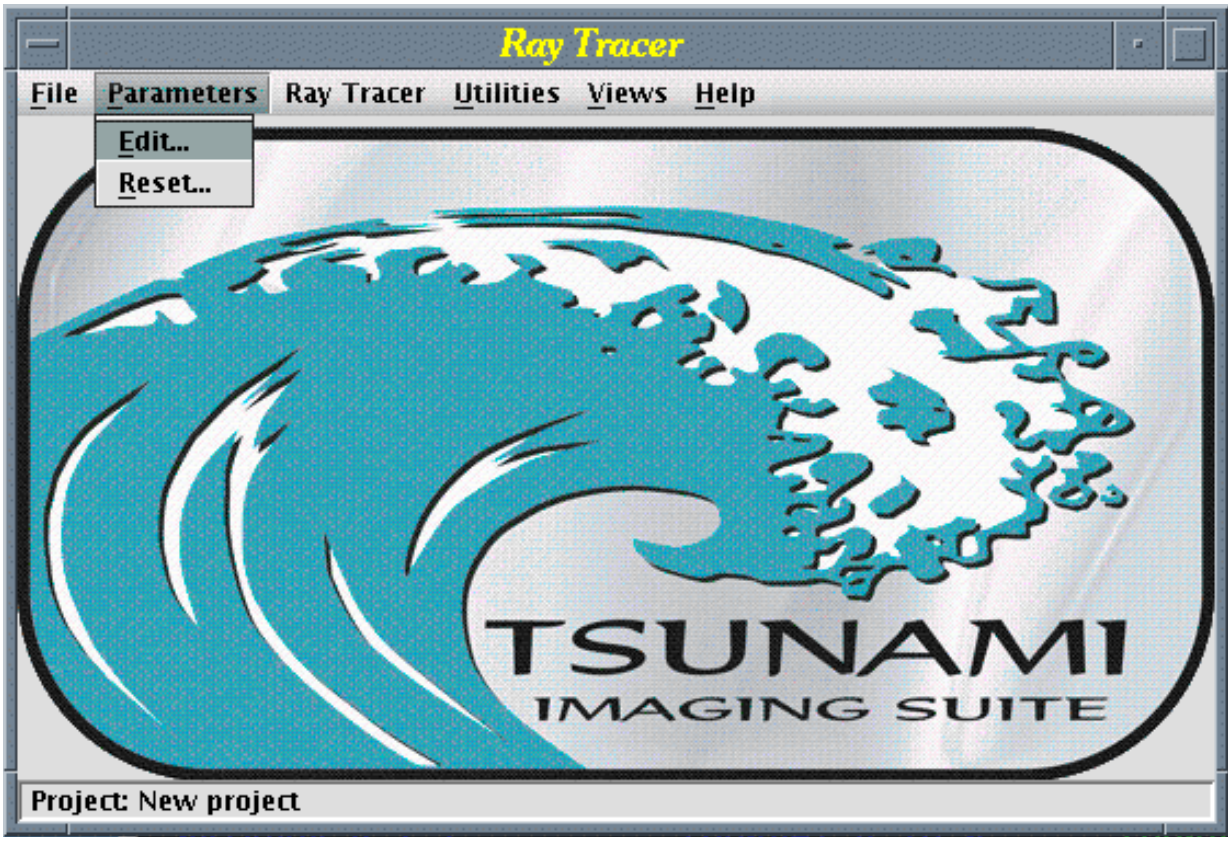
2. Toggle the Raytracer button on the Startup Window and click Accept.



3. Select File —> New to begin new job.

Generating a Travel Time File

1. Select **Parameter —> Edit** from the Ray Tracer startup window.



2. Select the **Files** tab in the **Parameter Selection Box**. Required information is highlighted in yellow.

File Settings:

Application Directory (hmdr): Points to the version of RAYS that you plan to use. Ex. For version 3.1.7 it should point to ~/tsunami_3.1.7/rays_3.1.7.

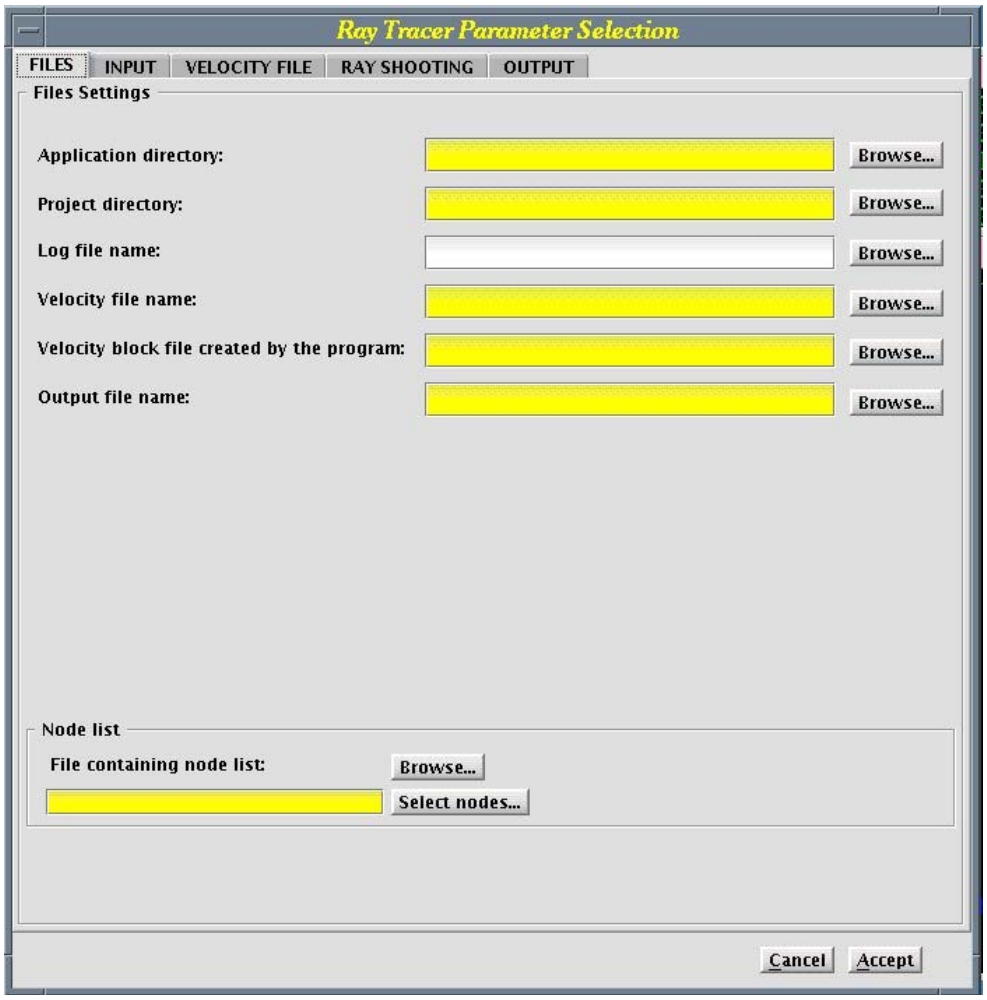
Project Directory (prjdr): The directory where the logfile will be located as well as other temporary files created by the job. It is also the directory where the scratch velocity block file is stored.

Log file name (logfile): If no log file name is entered, log file name defaults to project_dir/jobname.log. If designating a specific name for your logfile, it is recommended that you use the full path to that logfile name.

Velocity File Name (velf): This is a SEGY interval velocity file. Must be regularly sampled in inline, xline and depth.

Velocity Block File Created by the program (blkf): The name and location of the velocity block file created by the program. If left blank, default is to create the file named vels.blk in the project directory. Please see the *Performance Considerations and Advanced Features: Velocity Scratch Block File* section for more information.

Output File Name (outf): The name of the output travel time file.



Node List:

File containing node list (nddb): List of node names to use in job. For more information on the node list file please refer to the *Cluster Configuration* section.

Browse: Opens dialog box to select node file.

Select Nodes: Allows you to edit the selected node file. From this window you may add new nodes to your list, delete selected nodes, and turn nodes on/off for your job.

Parameter File For Command Line Use:

The **Files Setting** page would be represented in the command line parameter file by the following list of parameters:

hmdr= home directory of software
prjdr= project directory
logfile= logfile name
velf= SEGY velocity file
blkf= Velocity Block File Created by the program
outf= name of output travel time file
nddb= node database file

For more information on creating the command line parameter file and complete list of required parameters please see *Appendix B: Parameter File Information.*

3. Select the **Input** tab on the **Parameter Selection** box. Enter in necessary information. Required information is highlighted in yellow.

Dataset Dimensions (twod): Controls whether job is a 2D or 3D job. If the user is running rays for a 2D line, all shots will be on a single inline and a single inline of travel times will be output for each shot. The default is 3D.

RAYS works by shooting shots over the area covered by the source and receivers for the seismic dataset. For a given PSDM job, the area covered by the shots should cover the area of the source/receivers of the input seismic data. You can have multiple RAYS travel time files for a given PSDM job. The spacing between shots need not be the same for the multiple input files.

Shot Locations:

First Inline to Shoot (ilfrst): first inline location to shoot

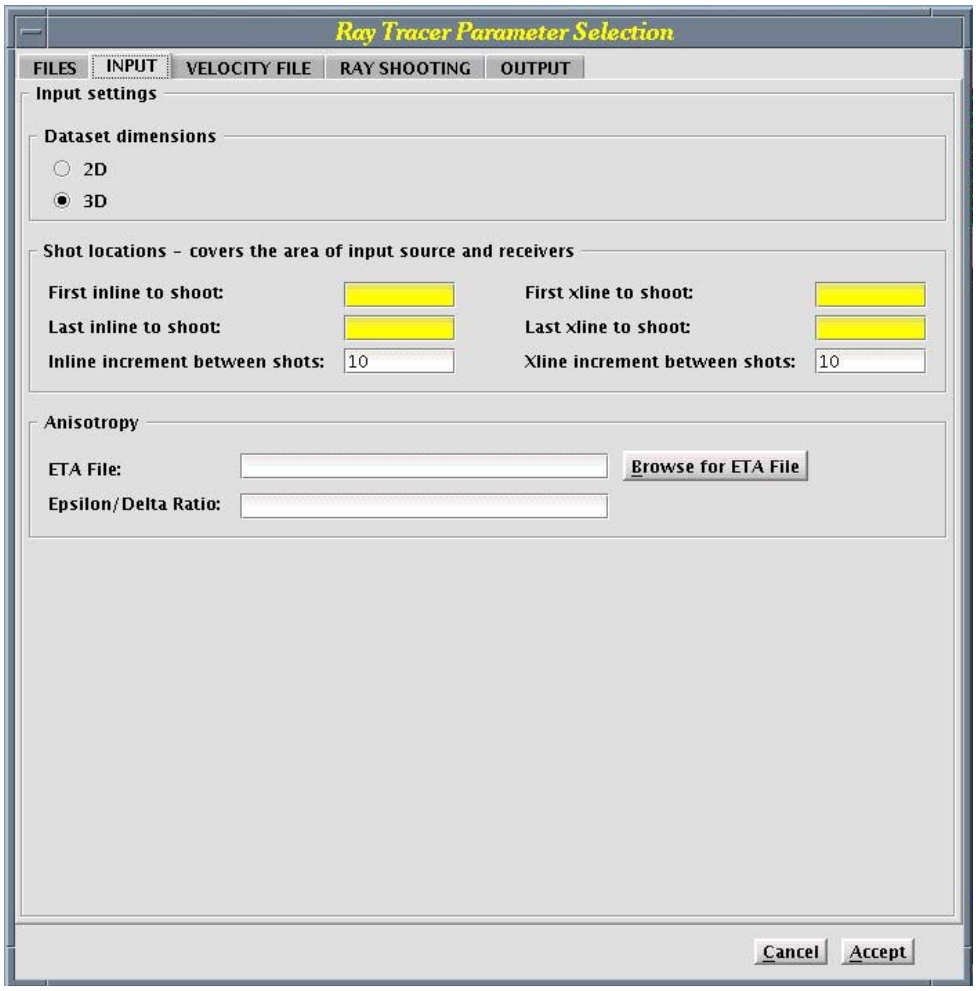
Last Inline to Shoot (illst): last inline location to shoot

Inline Increment Between Shots (ilinc): inline increment to shoot (default is 10)

First Xline to Shoot (xlfrst): first xline location to shoot

Last Xline to Shoot (xllst): last xline location to shoot

Xline Increment Between Shots (xlinc) : xline increment to shoot (default is 10)



Anisotropy:

ETA File (etaf): Name of segy file of ETA values. ETA values must be sampled at the same inline/xline increment and depth increment as the velocity model. They must be in segy format and the inline/xline header values need to be in the same location and same format as the velocity model. These things will be checked and verified by the ray tracer before running.

Epsilon/Delta Ratio (epdelr): Epsilon/Delta Ratio.

Both the ETA file and the Epsilon/Delta Ratio are required for the anisotropy function.

Please Note: The anisotropy function is not in the eikonal version of the travel time solver and would give you an error if you tried to run in that mode. There are no changes necessary when you run the PSDM migration.

Please Note: If you are using the Velocity Smoother and wish to use Anisotropy in your PSDM migration, you will need to create the block model in the Raytracer and not the Velocity Smoother.

Parameter File For Command Line Use:

The **Input** page would be represented in the command line parameter file by the following list of parameters:

twod= 2D(1) or 3D(0). Default is 0 or 3D.
xlfrst= first xline location to shoot
xllst= last xline location to shoot
xlinc= xline increment to shoot (default is 10)
ilfrst= first inline location to shoot
illst= last inline location to shoot
ilinc= inline increment to shoot (default is 10)
etaf= ETA File in segy format
epdelr= Epsilon / Delta ratio

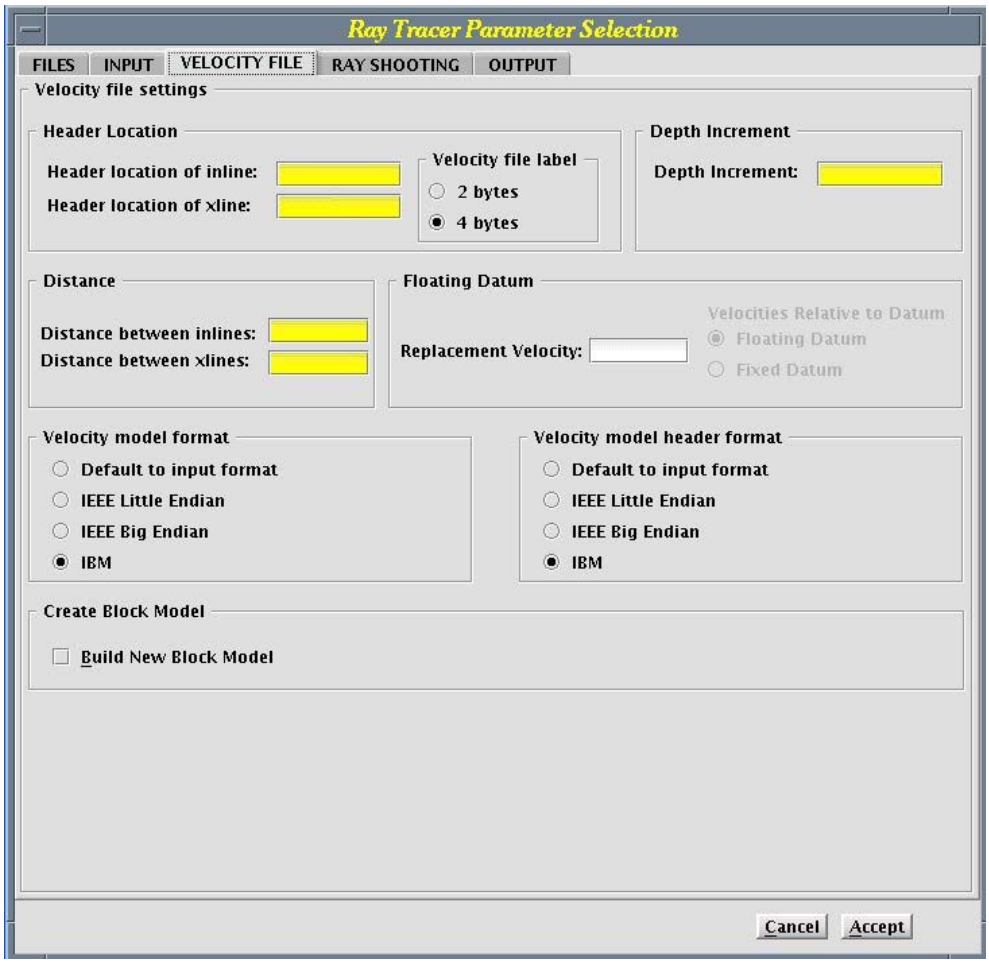
For more information on creating the command line parameter file and complete list of required parameters please see *Appendix B: Parameter File Information*.

4. Select the **Velocity File** tab from the **Parameter Selection** Box. Required items are highlighted in yellow.

The velocity model file is a set of interval velocities in a SEGY format. They must be stored in regular increments in the depth, xline, and inline directions. They also must be sorted by depth, xline and inline, with depth being the most rapidly varying dimension. It is important that the model be sorted by inline. Most processing systems will not sort the model by default. It is also important that the model provided be rectangular. All inlines must have the same number of xline traces

The format can be either IEEE or IBM format. If the model is in IEEE format it can be in either big or little endian format.

The velocity model need not extend to all needed inline and xline locations. If need be, the model will automatically be extended by simply taking the closest existing model location.



Header Locations:

Header loc of Xline (vxlb): The byte location of Xline labels in velocity model headers. (No Default)

Header loc of inline (vilb): The byte location of the inline labels in the model headers. (No Default)

Depth Increment (vdz): Depth increment of the velocity model.

Velocity File Label (vxliltyp): Velocity file label integer. Default is 0 or 4 byte integer. 0 = 4 byte integer and 1= 2 byte integer. (Default is 4 byte integer)

Distance:

Distance between xlines (xldist): distance between xlines in the survey. (No Default)

Distance between inlines (ildist): distance between inlines in the survey. (No Default)

Floating Datum:

Rays allows the user to image data from a floating datum or topography. Please see the *Performance Considerations and Advanced Features: Imaging From Floating Datum or Topography* section for more information.

Replacement Velocity (vdtm): The replacement velocity used between the fixed and floating datums.

Floating or Fixed Datum (frdtm): Velocities hung from fixed or floating datum. Default is 1 or floating datum. 0 = fixed datum, 1 = floating datum

Velocity Model Format (dataf): Indicates the format of the velocity model headers. Default to the input format or IEEE_LE, IEEE_BE or IBM. (Default is IBM).

Velocity Model Header Format (hdrfmt): Indicates the format of the velocity model header. Default to the input format or IEEE_LE, IEEE_BE or IBM. (Default is IEEE_BE).

Create Block Model:
Please see the *Performance Considerations and Advanced Features: Velocity Scratch Block File* and *the Smooth Velocity Documentation* for more information.

> **Build New Block Model (newb):** Default is off. GUI versions prior to 4.17.12 will have newb set to on.

Please Note: If you are using the Velocity Smoother and wish to use Anisotropy in your PSDM migration, you will need to create the block model in the Raytracer and not the Velocity Smoother.

Parameter File For Command Line Use:

The **Velocity File** page would be represented in the command line parameter file by the following list of parameters:

vxlb= The byte location of Xline labels in velocity model headers. (No Default)
vilb= The byte location of the inline labels in the model headers. (No Default)
vdz= Depth increment of the velocity model.
xldist= Distance between xlines in the survey. (No Default)
ildist= Distance between inlines in the survey. (No Default)
dataf= Indicates the format of the velocity model headers. Default to the input format or IEEE_LE, IEEE_BE or IBM. (Default is IBM).
hdrfmt= Indicates the format of the velocity model header. Default to the input format or IEEE_LE, IEEE_BE or IBM. (Default is IBM).
vxliltyp= Velocity file label integer. (Default is 4 byte integer)
newb= Default is on.
smfctr= By default Rays does no smoothing.

For more information on creating the command line parameter file and complete list of required parameters please see *Appendix B: Parameter File Information*.

5. Select the **Ray Shooting** tab on the **Parameter Selection** Box. Required fields will be highlighted in yellow.

Travel Time Algorithm set to Ray Tracing:

The shots are created by shooting rays in even increments in two dimensions, a radial plane or azimuth angle, and a vertical plane or phi angle. The number of initial rays is controlled by:

Angle Increments to Shoot:

Angle increment in the vertical plane (phinc): (default is 2.5)

Angle increment in the horizontal plane (azinc): angle increment in the azimuth plane (default is 2.5)

The rays are shot in groups of three or beams. As a beam advances through the model, the travel times of the three rays comprising the beam are compared with each other. If the travel times of the three rays differ by more than an allowed amount, the beam is divided; a new ray and a new beam are created. The user can control how often a beam divides by using the accuracy factor, and can control the maximum number of live rays per beams.

Accuracy factor to control creation of new rays (accfac): accuracy factor, a lower number increases the accuracy and beam divisions. Making the accuracy factor smaller will create more new rays. If you increase the number of live rays it may increase the accuracy, it will increase the runtime. You can check the increase in accuracy by look ing at the statistics for the shots. The median time difference should decrease. Otherwise you are just increasing the runtime, without gaining any benefit. You will also see the number of fill-in rays that get created. (default 4.0)

Maximum rays per beam (maxray): maximum number of live rays within a beam. (default 75)

DT of seismic data (dtacc): dt of seismic data in milliseconds (default 4)

The logfile provides statistics on the accuracy of the travel times to guide the selection of these parameters. As each travel time is stored in an output location, the difference between each of the three rays generating the output time is tabulated. In addition, the distance between the three rays generating the output time is also tabulated. The log file then reports the median time difference, the 80 percentile time difference, the median distance between rays, the 80 percentile distance between rays. You can increase the accuracy of the times by adjusting these factors. Of course you are also affecting the runtime of the job as well.

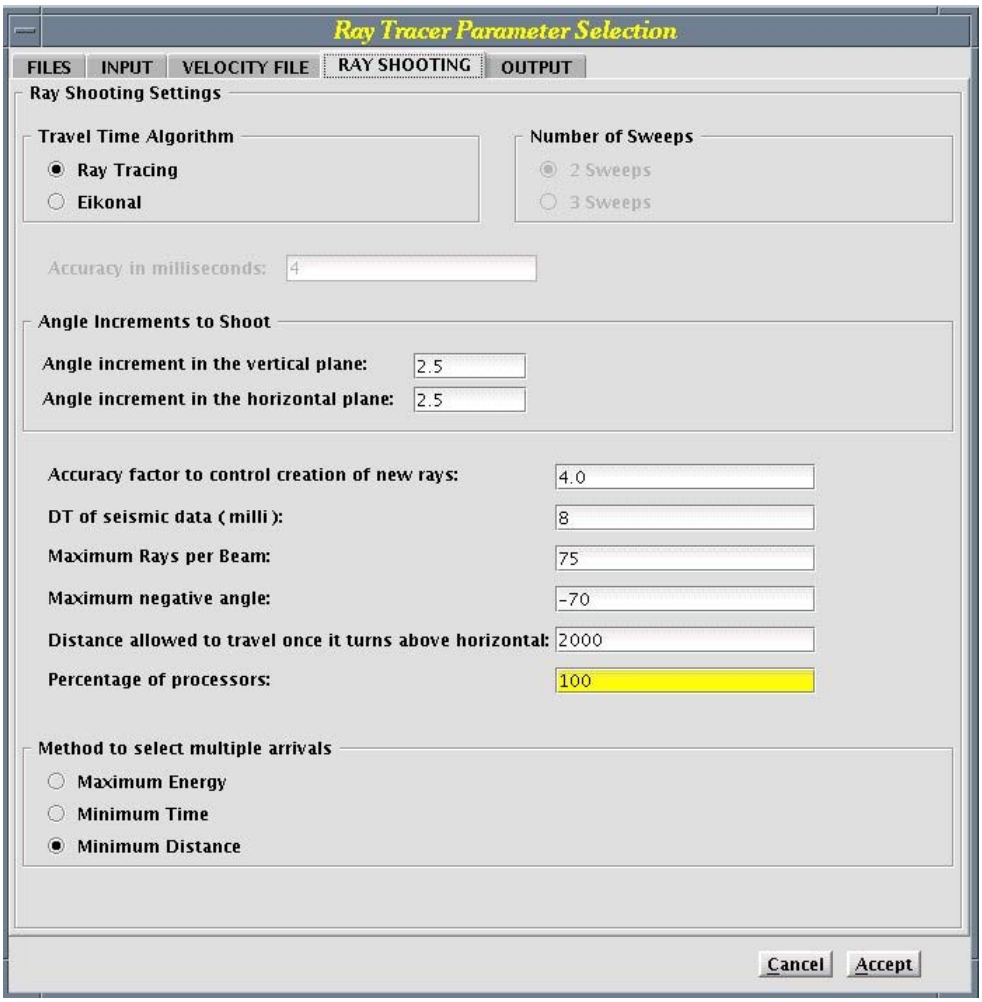
Turning Rays

RayTracer allows the existence of turning rays. These are rays that turn up beyond the horizontal. These rays are needed to image salt overhangs. However they also can greatly increase the runtime, if they are not controlled. Therefore the user can limit the maximum angle the rays can turn towards the vertical, and also the maximum distance the ray can travel once it turns up past the horizontal. The convention for this parameter is that 0 is horizontal, and –90 is vertical.

Maximum Negative Angle (mnegang): from 0 to –90 (default is –60)

Distance allowed to travel once it turns above horizontal (mxnegdst): from 0 to 10000 (default is 2000)

Method to select mulitple arrivals (mode): Maximum Energy, Minimum Distance or Minimum Time



Parameter File For Command Line Use:

The **Ray Shooting** page would be represented in the command line parameter file by the following list of parameters:

eiknl= Should be set to 0 for Ray Tracing option
phinc= (default is 2.5)
azinc= angle increment in the azimuth plane (default is 2.5)
accfac= accuracy factor, a lower number increases the accuracy and beam divisions. (default 4.0)
maxray= maximum number of live rays within a beam. (default 75)
dtacc= dt of seismic data in milliseconds (default 4)
mnegang= from 0 to –90 (default is –60)
mxnegdst= from 0 to 10000 (default is 2000)
mode= Maximum Energy, Minimum Distance or Minimum Time
vdtm= replacement velocity between datums
frdtm= fixed or floating datum
rqpcnt= required percent of processors

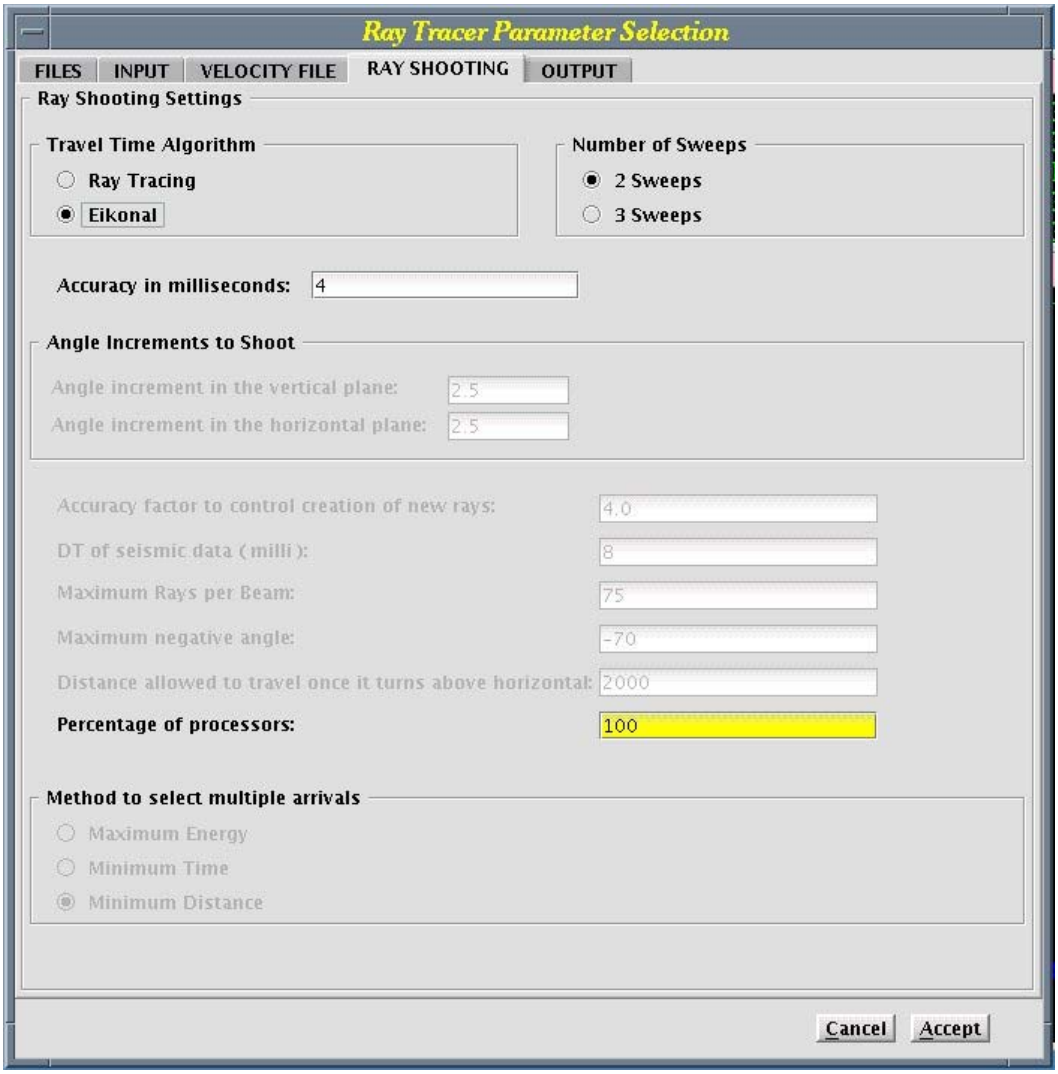
For more information on creating the command line parameter file and complete list of required parameters please see *Appendix B: Parameter File Information*.

Travel Time Algorithm set to Eikonal:

The eikonal solver uses a variable density grid that increases accuracy in more complicated areas.

Accuracy in milliseconds (tres): This parameter sets the threshold for subdividing grids. The parameter is in milliseconds and should be used in the range of 4—8. Making this parameter small can substantially increase the run time.

Number of Sweeps (nsweep): To refine the accuracy and check whether the grid should be subdivided, multiple sweeps through the grid are used. Set this parameter to either 2 or 3. Setting it to 3 substantially increases run time.



Parameter File For Command Line Use:

The **Ray Shooting page** set for the **Eikonal Solver** would be represented in the command line parameter file by the following list of parameters:

eiknl=	Should be set to 1for Eikonal Solver option
rqpcnt=	Required percent of processors
tres=	Accuracy in milliseconds
nsweep=	Number of sweeps

For more information on creating the command line parameter file and complete list of required parameters please see *Appendix B: Parameter File Information*.

6. Select the **Output Tab** on the **Parameter Selection** box. Required items will be high-lighted in yellow.

Describing Ray Image Locations – cover area of migration output

The image locations must cover the area of the migration output locations. The increments for the image locations must be evenly divisible into the shot increments. So if shots are 10 inlines apart the subsurface locations may be 10,5,2,1 etc. These locations are the locations where the travel time values are recorded.

If you are using multiple travel time files, all files must have the same subsurface locations, and increments.

The output increments need not match the migration output increments. In fact it is advised that when outputting every inline, or xline, that the travel time output increment should be in the neighborhood of four to five. This will improve the performance and prevent the travel time file from becoming extremely large.

If you are using the migration for velocity analysis, and the output CDPs are very widely spaced, you should have the subsurface locations match the migration output CDP locations. It is advisable in this case for the first, last and increments to match, to both improve performance, and minimize the interpolation distances.

First inline to output (ilfrstout): first subsurface inline

Last inline to output (illstout): last subsurface inline

Inline increment to output (ilincout): increment between subsurface inlines (default is 5)

First xline to output (xlfrstout): first subsurface xline

Last xline to output (xllstout): last subsurface xline

Xline increment to output (xlincout): increment between subsurface xlines (default is 5)

Migration Aperture and Maximum Offset

RAYS determines the output size of the travel time volume based on the migration aperture and maximum offset of the planned migration. The travel time volume never needs to be larger in diameter than two times the migration aperture plus the maximum offset. The migration aperture is defined as the maximum distance from an input trace to an image trace it contributes to. Based on these parameters the size of the travel time volume is automatically calculated, as well as the number of subsurface recording locations within the volumes. These are then reported in the log to the user and in the header of the travel time output volume.

Migration aperture (aper): aperture used in migration job

Maximum offset (mxoff): maximum offset of the seismic dataset

Depth Sampling Increment

You can either specify a fixed output depth increment, or allow RAYS to calculate a variable depth sampling increment based on a set of frequency depth pairs. The frequency depth pairs define the output data frequency you expect to recover, and then RAYS calculates the necessary depth increment to recover this frequency. **Generally this method will allow the migration job to run faster**. Either way, you will be able to specify any constant output depth increment for your PSDM job.

By using the fixed depth sampling increment you override any depth frequency pairs. If neither is specified, then the application will generate a depth increment based on Nyquist.

Fixed depth sampling increment (dinc): Depth sampling increment. The default for metric is 30 meters, and for English is 90 feet.

Depth frequency pairs (frqpr): '1000 90 5000 70 20000 60' Depth frequency pairs enclosed in single quotes.

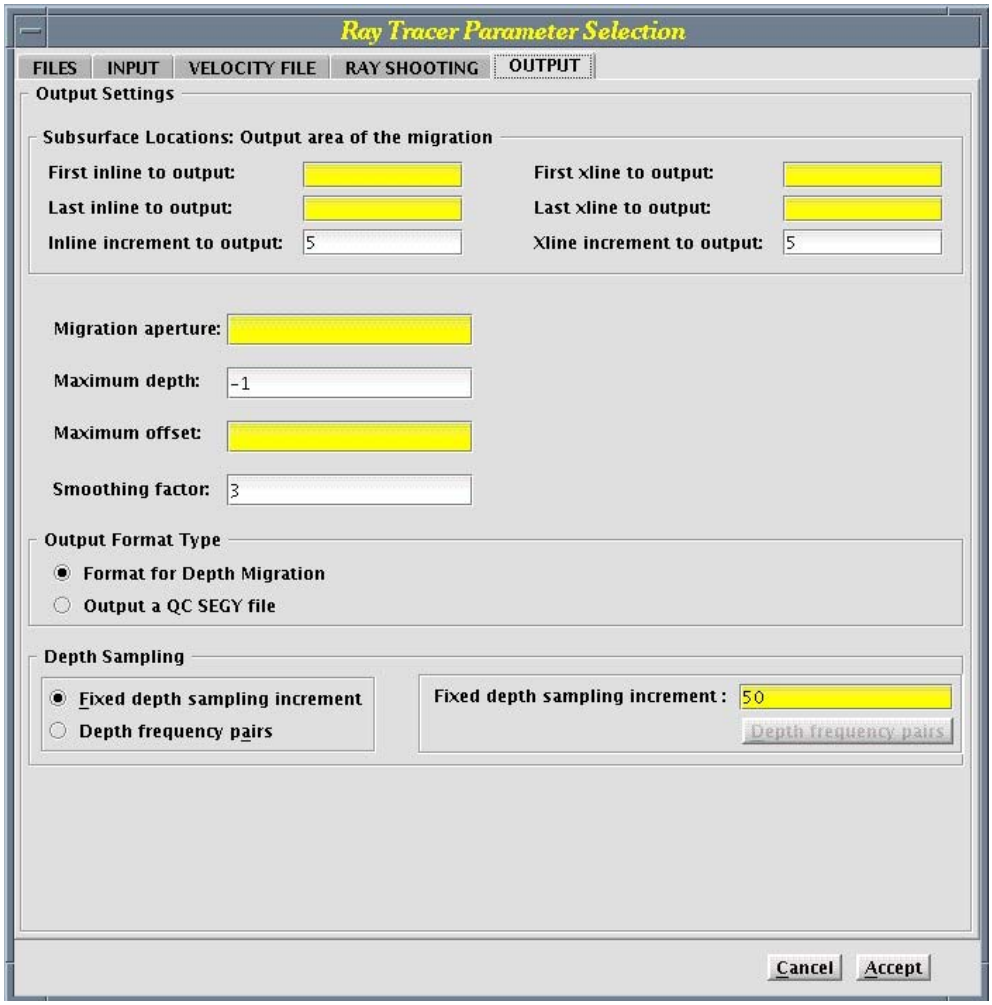
Maximum depth (dmax): maximum depth to output

Smoothing Factor (smth): smoothing of the output travel times

Output Format Type (fmto):

Beginning in Ray Tracer 3.1.9 the user can optionally create a SEGY file of travel times. This can be used to QC the times before producing a full travel time file for the migration.
The SEGY file cannot be read by the migration, it is only for display purposes. In the file the inline, xline and shot number are stored as 4 byte integers in bytes 181,185, and 17 respectively.

You can set the output to segy by using the parameter **fmto=** 2. The default is **fmto=** 1 which is



Parameter File For Command Line Use:

The **Output** page would be represented in the command line parameter file by the following list of parameters:

xlfrstout= first subsurface xline
xllstout= last subsurface xline
xlincout= increment between subsurface xlines. (default is 5)
ilfrstout= first subsurface inline
illstout= last subsurface inline
ilincout= increment between subsurface inlines. (default is 5)
aper= aperture used in migration job
mxoff= maximum offset of the seismic dataset
dinc= depth sampling increment
frqpr= '1000 90 5000 70 20000 60' Depth frequency pairs enclosed in single quotes.
smth= smoothing of the output travel times
dmax= maximum depth to output
fmto= output format

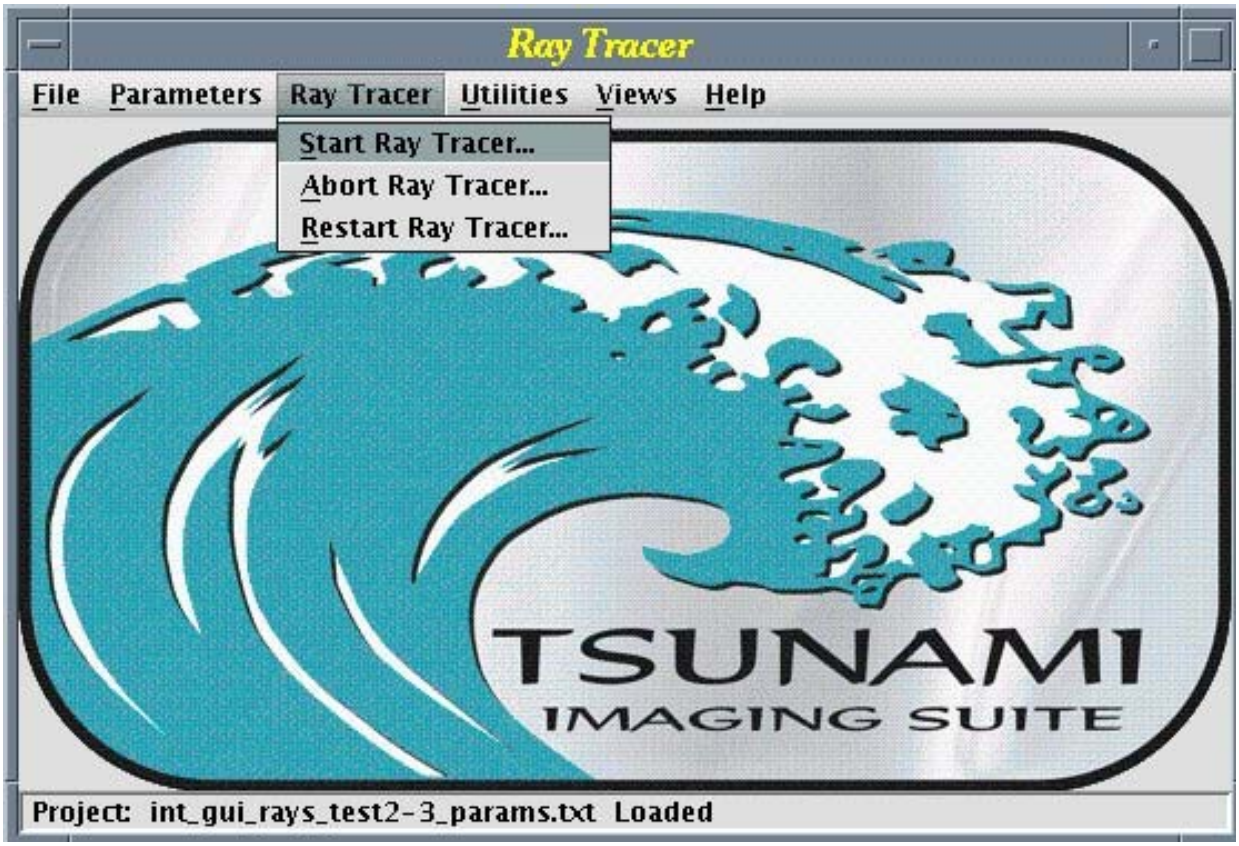
For more information on creating the command line parameter file and complete list of required parameters please see *Appendix B: Parameter File Information*.

Stopping and Restarting a Job

Stop/Restart Job from GUI

Ray Tracer - Start

Starts a job with the current parameter file. The current parameter file is displayed on the top bar of the user interface. Once the file is started, the status bar will display "Starting Ray Tracer for file: (file name)."



Rays Tracer - Abort

Stops the job immediately. This will abort the job, perform necessary cleanup and prevent the job from restarting.

Ray Tracer - Restart

Select 'Yes' to restart the job.

Once the file is restarted, the status bar will display "Restarting Ray Tracer for file: (file name)."

Please Note: If your job will not restart, it may be missing the *.pids file. Check the project directory for the param_file_name.pids file. If it is not in the project directory, you can recreate the file by using the touch command. Then retry the restart.

Abort/Restart from Command line:

To start a job type:
rays_start < parameter file name > &
Jobs started in this manner will automatically restart if they fail to complete successfully.

To abort a job type:
rays_abort < parameter file name >
This will abort the job, perform all the cleanup and prevent the job from restarting.

To restart a job that you aborted type:
rays_restart < parameter file name >

Performance Considerations and Advanced Features

Velocity Scratch Block File

Ray Tracer is designed to be able to use a velocity file of any size, including one that is larger that the system's memory. If the system is a shared memory multiprocessing system it will share the portions of the velocity file in memory, among the processors, thereby holding as much of the velocity file in memory as possible. It will automatically manage the portions of the velocity file in memory.

In order to achieve this, and maintain a high level of performance Ray Tracer reformats the velocity file into a blocked or tiled velocity file. Since all blocks are the same size, it will round up the dimensions of the velocity model to the next block boundary. It will then pack the velocities into the block model as shot integers. As a result, this block file will be somewhat greater that one half the size of the velocity model size and it will be reformatted for rapid access. The user needs to make sure that sufficient disk space is available for this file. The location and name of the file can be specified with the **blkf=** parameter.

The exact size of the tiled velocity file depends on the geometry of the original file. The velocity file will be rounded up to the nearest increment of 25 in each dimension. While this takes up additional disk space it allows the program to run using a velocity file that is much larger than the system's memory and still maintain performance. It also facilitates sharing the model among many processors in an efficient manner.

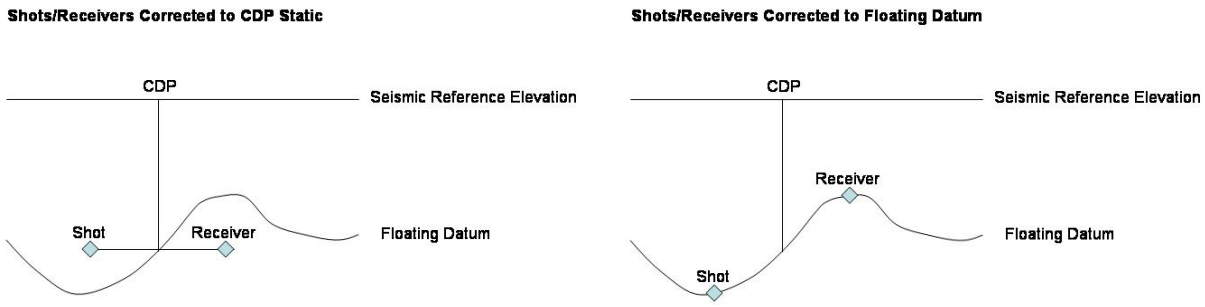
Since building the block model can sometimes take a good deal of time, you can use the **newb**= parameter to signal to the program that it should reuse an existing block file.

> **newb=** 0 use an existing block file, 1 create a new block file (default is 0) If using something other than the default block file (project_directory/vels.blk), you will need to specify it in the blkf= parameter.

In order to minimize the ray scattering it is advised that you smooth the interval velocities before running Ray Tracer on a velocity model. You can either smooth the model with the software you used to produce the model, or you can use the Smooth Velocity module to smooth the model and create a new velocity file and/or block file. By default Ray Tracer does no smoothing. Please see the *Smooth Velocity* documentation for more information.

Imaging From Floating Datum or Topography

When a survey has been acquired over an irregular topography it is common to image the data from a smoothed floating datum. When imaging from a floating datum the input traces are assumed to be corrected to either the cdp static or to the floating datum. Please see pictures. For the output migrated traces, the first sample will be at the flat seismic reference datum.



Tsunami PSTM supports two methods for imaging from a floating datum. For Method 1, the input trace headers must contain the two way static or time shift (in milliseconds) between the floating datum and the flat seismic reference datum at the cdp. For Method 2 the input trace headers must contain the elevation (in feet or meters) of the floating datum at the cdp.

If using a floating datum, the floating datum parameters must be specified when building the trace.db since the static or elevation values will be stored there. When you build the trace.db it is important to check the log file. The floating datum value for the first trace of each input data file will be listed. This value will be in milliseconds if using Method 1 and in feet or meters if using Method 2. After all input traces are read, the minimum and maximum static shift will be printed. For both methods, this will be converted to time in milliseconds. Be sure to check that the range is reasonable and the signs correct for your input data.

When imaging from a floating datum the velocity traces may be referenced (or hung) from either the floating datum or the Seismic Reference Datum. Different processing systems will output the velocities differently, so you should check that you know which way it is being done for your data. Use the "Velocities Relative to datum" button to select either Floating Datum or Flat Datum. If using command line entry, **frdtm=** 0 is for velocities relative the Seismic Reference Datum and **frdtm**= 1 is for velocities relative to the floating datum. This parameter works the same way for Method 1 and Method 2.

For PSTM, all required parameters are located on the INPUT parameter tab.

For prestack depth imaging, the Replacement Velocity (**vdtm**) parameter is located on both the VELOCITY parameter tab in Raytracer and the INPUT tab in PSDM. Velocities Relative to datum parameter (**frdtm**) is located on the VELOCITY parameter tab in the Raytracer program. All other floating datum parameters on the INPUT parameter tab of PSDM.

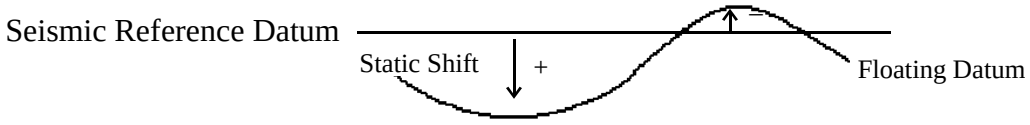
Floating Datum Method 1 – Static Shift in Time

This method is used when Time is selected for the "Static Shift in" parameter. If using command line entry, the parameter is **staticd=** 0.

The input traces must have the two way static or time shift (in milliseconds) from the floating datum to the seismic reference datum. The values must be written in the trace headers as 4 byte integer. All traces in the same cdp should have the same static value. The parameter that describes the byte location for the static shift in PSTM is **fldtm=**. Inputting a value for this parameter turns on the imaging from floating datum option.

Use the drawing below as a reference for calculating the static shift values to place in the trace headers.

Floating Datum Method 1



If the floating datum is below the Seismic Reference Datum, the static values should be positive. If the floating datum is above the Seismic Reference Datum, the static values will be negative. The static value at a cdp can be calculated with the following formula:

If the input data is in Paradigm format you must calculate the static value byte location by examination of the .pds file. First locate the header field name you are using for the static shift. Multiply the sequential number of this header field and add 1. See Appendix D: Paradigm File Information for an example. The header field used must be a four byte integer value.

The Reference Elevation parameter (**dtmelev=**) is always set to zero for Method 1. This means that for the output migrated traces, the first sample will be at the elevation used to calculate the static values in the trace headers.

The Replacement Velocity parameter (**vdtm=**) is not used for Method 1 for PSTM. For depth imaging, it is required and must be provided for both Raytracer and PSDM.

Before running a Raytracer job from a floating datum, you should first build the trace database using PSDM.

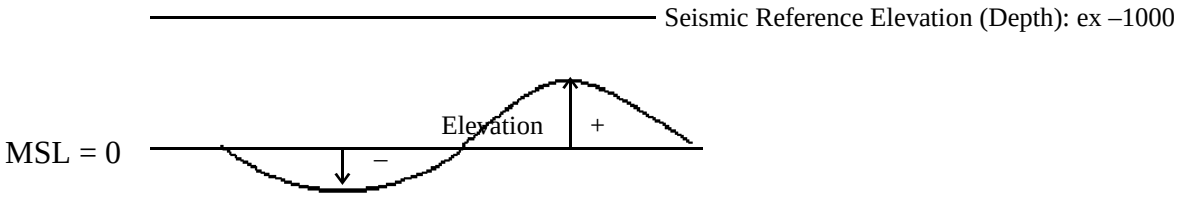
Floating Datum Method 2 – Static Shift in Depth

This method is used when Depth is selected for the "Static Shift in" parameter. If using command line entry, the parameter is **staticd=** 1.

The input traces must have the elevation of the floating datum at the cdp. The values must be written in the trace headers as 4 byte integer. All traces in the same cdp should have the same elevation value. The parameter that describes the byte location for the datum elevation in PSDM is **fldtm=**. Inputting a value for this parameter turns on the imaging from floating datum option.

Use the drawing below as a reference for calculating the static shift values to place in the trace headers.

Floating Datum Method 2



Elevations for the floating datum are measured from mean sea level. Elevations above MSL are positive. Elevations below MSL are negative.

If the input data is in Paradigm format you must calculate the datum elevation value byte location by examination of the .pds file. First locate the header field name you are using for the datum elevation. Multiply the sequential number of this header field and add 1. See Appendix D: Paradigm File Information for an example. The header field used must be a four byte integer value.

The Reference Elevation parameter (**dtmelev=**) is the Seismic Reference Elevation for the output traces. The migrated traces will be relative to (or hung) from this elevation. Seismic Reference Elevations above sea level will be negative. Seismic Reference Elevations below sea level will be positive.

The Reference Velocity parameter (**vdtm=**) is the replacement velocity that is to be used between the floating datum and the Seismic Reference Elevation. It is required for Method 2 in PSTM, Raytracer and PSDM.

OBC Data. Specifying your data as OBC was added in Tsunami Version 5.2.13. This option allows you to specify if the input data is acquired using Ocean Bottom Cable methods. In turning this option on, the program assumes the source is at a flat datum or water surface and the receiver is at the floating datum or water bottom. This will turn the Shots / Receivers Corrected To option to Floating Datum. The static at cdp location is what should be in the headers however. Using this option properly angles the impulse response, much like what happens when imaging from topography. Approximations can be made with statics, but they aren't as accurate. Please see the Performance Considerations and Advanced Features: Imaging From Floating Datum or Topography section.

Displaying Travel Times

In rays 3.1.9 the user can optionally create a SEGY file of travel times. This can be used to QC the times before producing a full travel time file for the migration.
The SEGY file cannot be read by the migration, it is only for display purposes. In the file the inline, xline and shot number are stored as 4 byte integers in bytes 181,185, and 17 respectively.

You can set the output to segy by using the parameter **fmto=** 2. The default is **fmto=** 1 which is the correct format for the depth migration.

Appendix A: Log file Information

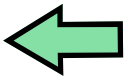
The log file is located in ~proj_dir/parameter_file.log

Raytracer provides a great deal of information in the log file. If you are running a new dataset through Raytracer it is critical that you verify that the program is reading the trace data and velocity data correctly.

To assist in this effort, at the start of the job the program will print out the range of inlines and xlines covered in the velocity file. It will print out the minimum and maximum velocity values used in the job, and it will print out header data from the first trace it finds in the input data files. These should be checked to verify that the data formats are being read correctly.

```
Version 4.11 Tsunami Ray Tracer
Parameter file: new_ttime.example

nddb= '/home/example/new_tests/node02.db'
hmdr= '/home/software/tsunami_4.11.6/rays_4.11.6'
prjdr= '/home/example/new_tests'
logfile= '/home/example/new_tests/ttime.log'
velf= '/home/example/velocities.segy'
blkf= '/home/example/new_tests/tmp.blk'
outf= '/home/example/ttimes.out'
xlfrst= 3920
xllst= 5541
ilinc= 20
ilfrst= 1982
illst= 3325
xlinc= 20
vxlb= 5
vilb= 1
vdz= 50
newb= 1
dataf= 2
hdrfmt= 2
eiknl= 0
nswep= 2
mode= 2
xlfrstout= 3920
xllstout= 5541
xlincout= 10
ilfrstout= 1982
illstout= 3325
ilincout= 10
aper= 23000
mxoff= 21000
xldist= 55
ildist= 55
rqpcnt= 60
```



CHECK 1
Shows parameters you input either through GUI or in your parameter file. Check the vdz= , vilb=, vxlb=, fmt=, xldist=, ildist=, aper= and mxoff= parameters for correct values. Please see *Appendix B: Parameter File Information* for parameter definitions.

```
Jun 23  15:54
Reading license file /home/software/psdm_license.dat
Current license expires month    1 day     31 year   2006
Reading the velocity model
Checking the sort order, and the limits of model
Velocity model coverage:  First xline 3920  Last xline 5540 xline inc 20
                  First inline 1982 Last inline 3322 inline inc 20
Veloicty model z increment: 50


Model dimensions

Model header size 3600
First x      0.0   Number of x values    82  X distance 1100.0
First y      0.0   Number of y values    68  Y distance 1100.0
First z      0.0   Number of z values   680  Z distance  50.0

Minimum velocity in model   4971
Maximum velocity in model  14486
Building velocity block model
Smoothing operator x dimen 1
Smoothing operator y dimen 1
Smoothing operator z dimen 1
Allocated  5.0 percent of the block model
Allocated 10.0 percent of the block model
Allocated 15.0 percent of the block model
Allocated 20.0 percent of the block model
Allocated 25.0 percent of the block model
Allocated 30.0 percent of the block model
Allocated 35.0 percent of the block model
Allocated 40.0 percent of the block model
Allocated 45.0 percent of the block model
Allocated 50.0 percent of the block model
Allocated 55.0 percent of the block model
Allocated 60.0 percent of the block model
Allocated 65.0 percent of the block model
Allocated 70.0 percent of the block model
Allocated 75.0 percent of the block model
Allocated 80.0 percent of the block model
Allocated 85.0 percent of the block model
Allocated 90.0 percent of the block model
Allocated 95.0 percent of the block model
Allocated 100.0 percent of the block model
Max Derivatives in Smoothed Model
dx       0.6   dy       0.4  dz     10.4

Some calculated parameters:

Diameter of ttime cube x direction:   67000
Diameter of ttime cube y direction:   67000
Number of subsurface locations in x direction 122
Number of subsurface locations in y direction 122
Number of output depth levels 307
```

CHECK 2
Verify that license file is correct.

CHECK 3
Verify Velocity Model Coverage.

```
Outputting file format 1
Header successfully written

Server has connected to node node01
Server has connected to node node02
Server has connected to node node03
Server has connected to node node04
Server has connected to node node05
Server has connected to node node06
Server has connected to node node07
Server has connected to node node08
Server has connected to node node09
Server has connected to node node10
Server has connected to node node11
Server has connected to node node12
Server has connected to node node13
Server has connected to 13 nodes with 26 processors
Server failed to connect to 0 nodes
Server has connected to 100.0 percent of processing power
Min vel in model 4971.097656
Max vel in model 14485.589844
Node node01 successfully allocated 250 mbytes of memory
Node node02 successfully allocated 250 mbytes of memory
Node node03 successfully allocated 250 mbytes of memory
Node node04 successfully allocated 250 mbytes of memory
Node node05 successfully allocated 250 mbytes of memory
Node node06 successfully allocated 250 mbytes of memory
Node node07 successfully allocated 250 mbytes of memory
Node node08 successfully allocated 250 mbytes of memory
Node node09 successfully allocated 250 mbytes of memory
Node node10 successfully allocated 250 mbytes of memory
Node node11 successfully allocated 250 mbytes of memory
Node node12 successfully allocated 250 mbytes of memory
Node node13 successfully allocated 250 mbytes of memory


Total number of shots 5727


 ******* BEGIN SHOOTING RAYS *******

Processing shot 1 at inline 1982, xline 3920 at      0.0 min, node node01
Processing shot 2 at inline 1982, xline 3940 at      0.0 min, node node01
Processing shot 13 at inline 1982, xline 4160 at     0.0 min, node node07
Processing shot 14 at inline 1982, xline 4180 at     0.0 min, node node07
Processing shot 11 at inline 1982, xline 4120 at     0.0 min, node node06
Processing shot 12 at inline 1982, xline 4140 at     0.0 min, node node06
```

Lines Deleted From Original Log File for Space Saving in Documentation

```
Processing shot 2323 at inline 2522, xline 5540 at     0.0 min, node node12
Processing shot 2324 at inline 2522, xline 5560 at     0.0 min, node node13
Processing shot 2325 at inline 2542, xline 3920 at     0.0 min, node node13
Shot 2325 number of fill in rays 16524
```

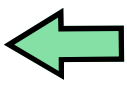
Lines Deleted From Original Log File for Space Saving in Documentation

Shot 5717 median ray diff 0.0070 80 percentile diff 0.0137
Shot 5717 percent empty grids 2.45
Shot 5725 number of fill in rays 14111
Shot 5725 median dist to ray 453 80 percentile dist 710
Shot 5725 median ray diff 0.0068 80 percentile diff 0.0130
Shot 5725 percent empty grids 1.46
Shot 5718 number of fill in rays 13664
Shot 5718 median dist to ray 449 80 percentile dist 719
Shot 5718 median ray diff 0.0070 80 percentile diff 0.0135
Shot 5718 percent empty grids 2.20
Shot 5724 number of fill in rays 13758
Shot 5724 median dist to ray 453 80 percentile dist 711
Shot 5724 median ray diff 0.0067 80 percentile diff 0.0130
Shot 5724 percent empty grids 1.58
Shot 5723 number of fill in rays 13863
Shot 5723 median dist to ray 455 80 percentile dist 715
Shot 5723 median ray diff 0.0066 80 percentile diff 0.0131
Shot 5723 percent empty grids 1.79
Shot 5722 number of fill in rays 13802
Shot 5722 median dist to ray 454 80 percentile dist 718
Shot 5722 median ray diff 0.0066 80 percentile diff 0.0130
Shot 5722 percent empty grids 2.66
Shot 5721 number of fill in rays 13637
Shot 5721 median dist to ray 453 80 percentile dist 714
Shot 5721 median ray diff 0.0067 80 percentile diff 0.0131
Shot 5721 percent empty grids 2.92
Shot 5720 number of fill in rays 13665
Shot 5720 median dist to ray 443 80 percentile dist 707
Shot 5720 median ray diff 0.0071 80 percentile diff 0.0136
Shot 5720 percent empty grids 3.65
Shot 5719 number of fill in rays 13555
Shot 5719 median dist to ray 447 80 percentile dist 715
Shot 5719 median ray diff 0.0073 80 percentile diff 0.0139
Shot 5719 percent empty grids 3.42

******RAY SHOOTING COMPLETED*******

******** Median Stats on Shots *********
Median distance between rays 422 80 percentile distance 690
Median difference between rays 0.0063 80 percenfile difference 0.0124

RAYS Main: Successful Completion
Jun 25 15:40



CHECK 4
Check for successful completion of process.

Example Check License Log File:

```
Product name rays
Creation date 010104
Start date 010104
End date 123104
Length of hw record 96
Number of licensed nodes 100
   Mac Address          Node Name    Checked Out

 00:A0:C9:FB:25:B4    dual450
 00:42:52:00:13:E1     rlx-0-0-1
 00:42:52:00:0F:37     rlx-0-0-2
 00:42:52:00:13:36     rlx-0-0-3              1
 00:42:52:00:11:38     rlx-0-0-4
 00:42:52:00:23:0B     rlx-0-0-5
 00:42:52:00:0B:EF     rlx-0-0-6
 00:42:52:00:18:FD     rlx-0-0-7              1
 00:42:52:00:1C:3C     rlx-0-0-8              1
 00:42:52:00:19:27     rlx-0-0-9
 00:42:52:00:25:B4     rlx-0-0-11
 00:42:52:00:13:24     rlx-0-0-13
 00:42:52:00:17:26     rlx-0-0-15
 00:42:52:00:1A:35     rlx-0-0-17
 00:42:52:00:1A:0B     rlx-0-0-19
 00:42:52:00:18:7C     rlx-0-0-21
 00:E0:81:03:38:C6     linux1
 00:03:47:71:D2:54     linux2                 1
 00:03:47:71:E9:F5     linux3
 00:03:47:6B:45:47     linux4                 1
 00:E0:81:02:B4:3E     linux5
 00:03:47:71:62:FC     linux6
 00:03:47:71:5B:73     linux7
 00:03:47:71:D2:41     linux8
 00:03:47:71:65:23     linux9
 00:03:47:71:62:2D     linux10
 00:03:47:71:5D:B9     linux11
 00:03:47:71:5D:BB     linux12                1
 00:03:47:71:5B:7C     linux13                1
 00:03:47:71:D2:49     linux14                1
}
```

NOTE
Beginning and ending date of license.

NOTE
A 1 indicates that the node is checked out.

Example Clear Nodes Log File:

```
clearing node rlx-0-0-1
clearing node rlx-0-0-2
clearing node rlx-0-0-3
clearing node rlx-0-0-4
clearing node rlx-0-0-5
clearing node rlx-0-0-6
clearing node rlx-0-0-8
clearing node rlx-0-0-9
clearing node rlx-0-0-11
clearing node rlx-0-0-13
clearing node rlx-0-0-15
clearing node rlx-0-0-17
clearing node rlx-0-0-19
clearing node rlx-0-0-21
Resetting license file /apps/tsunami/pstm_license.dat
Reading license file /apps/tsunami/pstm_license.dat
Resetting license file /apps/tsunami/psdm_license.dat
Reading license file /apps/tsunami/psdm_license.dat
```

Appendix B: Parameter File Information

Required Parameters

hmdr= Application home directory
Required

prjdr= Project directory
Required

velf= This is a segy interval velocity file
Same rules apply, must be regularly sampled in inline, xline and depth
Required
No Default

blkf= This is the name of the velocity block file created by the program
Required
No Default

outf= The name of the output travel time file
Required
No Default

xldist= Distance between xlines
Required
No Default

ildist= Distance between inlines
Required
No Default

xlfrst= First xline to shoot
Required
No Default

xllst= Last xline to shot
Required
No Default

ilfrst= First inline to shoot
Required
No Default

illst= Last inline to shoot
Required
No Default

ilfrstout= First inline to output
Required
No Default

illstout= Last inline to output
Required
No Default

xlfrstout= First xline to output
Required
No Default

xllstout= Last xline to output
Required
No Default

aper= Migration aperture to use
Required
No Default

mxoff= Maximum offset to use
Required
No Default

vilb= Header location of inline in velocity model
Required
No Default

vxlb= Header loccation of xline in velocity model
Required
No Default

vdz= Depth increment of the velocity model
Required
No Default

nddb= Node database - See Cluster Configuration for a description
Required
No Default

ilincout= Inline inc to output
Default = 5

xlincout= Xline inc to output
Default = 5

Alphabetical List of Parameters

accfac= Accuracy factor to control creation of new rays. Changes accuracy as a function of angle to shot.
Default = 4.0

angnum= Number of angle increments in spherical shell
Default = 10

ani= Anisotropy flat

aper= Migration aperture to use
Required
No Default

azinc= Angle increment in the horizontal plane
Default = 2.5

blkf= This is the name of the velocity block file created by the program
Required
No Default

dataf= Format of velocity model
0= IEEE_LE
1= IEEE_BE
2 = IBM
Default = 2 or IBM

dinc= Fixed depth sampling increment. If dinc= is used it will override frqpr=
Default for metric is 30 meters, and for English is 90 feet

dmax= Maximum depth to output
Default = depth of the model

dtacc= dt of seismic data. Accuracy in milliseconds
Default = 8

eiknl= Eikonal or Ray Tracer flag
0 = eikonal
1 = ray tracer
Default = 0 or eikonal

epdelr= Epsilon / Delta ratio
No Default

etaf= ETA file name

fmto= Option to create QC SEGY file of travel times
1 = correct format for Depth Migration
2 = output to SEGY

frdtm= Velocities hung from fixed or floating datum.
0 = fixed datum
1 = floating datum
Default = 1 or floating datum

frqpr= Depth frequency pairs, defines vertical sampling increment.
Pairs enclosed in single quotes.
No Default

hdrfmt= Format of velocity model headers
0= IEEE_LE
1= IEEE_BE
2 = IBM
Default = 1

hmdr= Application home directory

ildist= Distance between inlines
Required
No Default

ilfrst= First inline to shoot
Required
No Default

ilfrstout= First inline to output
Required
No Default

ilinc= Inline increment between shots
No Default

ilincout= Inline inc to output
No Default

illst= Last inline to shoot
Required
No Default

illstout= Last inline to output
Required
No Default

logfile= User specified logfile name

maxray= Maximum rays per beam
Default = 75

mnegang= Maximum negative angle, ie rays are allowed to turn above the horizontal by this amount. 90 would be a vertical ray traveling towards the surface.
Default = –70
Max = 0
Min = –89.9

mode= Method to select multiple arrivals
0 = maximum energy
1 = minimum time
2 = minimum distance
Default = 1

mxnegdst= Distance the ray is allowed to travel once it turns above the horizontal
Default = 2000
Max = 10000
Min = 0

mxoff= Maximum offset to use
Required
No Default

nddb= Node database - See Cluster Configuration for a description
Required
No Default

newb= Controls whether the block model gets created.
1 = Create new block model.
0 = Use existing block model
Default = 1

nswep= Number of sweeps
Default = 2

outf= The name of the output travel time file
Required
No Default

phinc= Angle increment in the vertical plane
Default = 2.5

prjdr= Project directory

rqpcnt= Required percent of nodes
Default = 100

rystff= Ray damping factor, stiffness adjustment
Default = 0

smfctr= Smoothing Factor for Velocity Model
0 = off
1 = on
Default = 0

smth= Smoothing of the output travel times.
Default = 3.
0 = off

stpsz= Step size of ray advance
Default calculated

tres= Accuracy factor for eikonal
Default = 4

twod= 2D/3D switch
0 = 3D (threed)
1 = 2D (twod)
Default= 0 or 3D

vdtm= When using a floating datum you will also need to specify a correction velocity or a velocity to use between the fixed datum and the floating datum. This should be the same velocity that was used when Rays was run.
Default = 0 or not used

vdz= Depth increment of the velocity mode
Required
No Default

velf= This is a segy interval velocity file. Same rules apply, must be regularly sampled in Inline, xline and depth
Required
No Default

vilb= Header loc of inline in velocity model
Required
No Default

vxlb= Header location of xline in velocity model
Required
No Default

vxliltyp= Velocity file label integer
Default = 0
0 = 4 byte integer
1 = 2 byte integer

xldist= Distance between xlines
Required
No Default

xlfrst= First xline to shoot
Required
No Default

xlfrstout First xline to output
Required
No Default

xlinc= Xline increment between shots
No Default

xlincout= Xline inc to output
No Default

xllst= Last xline to shot
Required
No Default

xllstout= Last xline to output
Required
No Default

Example Parameter File

The parameters can be in the parameter file in any order. Anything not a parameter is taken as a comment. All filenames, and pairs of numbers must be enclosed with single quotes. All parameters must include the equal sign, that is attached to the parameter string with no spaces. Parameter values should be separated from the equal sign by a space.

```
velf=  '/data/velocity.segy'          input velocity file
blkf=  '/data/velocity.blk'           velocity block file
logfile=  '/data/logfile.out'         logfile
outf=  '/data/times.out'              output travel timed
nddb=  'node_database.db'             node database file
vdz=   30                             depth increment in velocity file
vilb=   180                           byte offset of inline label in velocity header
vxlb=  184                            byte offset of xline label in velocity header
xldist= 25.0                          distance between xlines
ildist= 50.0                          distance between inlines
xlfrst= 10                            first shot xline
xllst=   100                          last short xline
xlinc=  5                             xline increment between shots
ilfrst= 20                            first shot inline
illst=  100                           last shot inline
ilinc=  5                             inline increment between shots
ilfrstout=   20                       first migration output inline
illstout=   100                       last migration output inline
ilincout=   5                         output increment of travel times
xlfrstout= 10                         first migration output xline
xllstout= 100                         last migration output xline
xlincout= 5                           output increment of travel times
aper=  6000                           migration aperture
mxoff= 6000                           maximum migration offset
dinc=100                              depth increment
newb=   1                             create a new velocity block model
mode= 1                               use minimum distance travel times
```

Appendix C: Java Installation Information

For the Tsunami GUI and Viewer you must have version 1.4.2 or higher of Java 2 Platform, Standard Edition (J2SE) installed along with Java 3D. Should java not be installed on your system, your system administrator can download and install using the following steps.

To Check For Correct Java Version on Your Machine:

1) java –version
   This will give you the version number of your java installation.

2) which java
   This will tell you where your java installation is located.

Should these commands return no information, please ensure that the JAVA_HOME environment variable is set in the users .cshrc or .bashrc file. Your system administrator may need to be consulted in order to correctly set the path for JAVA_HOME.

To Download and Install Java and Java 3D on Linux and Itanium Platforms:

1) Go to Java website via following link.
   http://java.sun.com/j2se/index.jsp

2) Select the latest release (non Beta version). Must be J2SE SDK version 1.4.2 or higher. Please note: you must download J2SE version 5.0 if using AMD Opteron 64 bit linux.

3) Please read all installation documentation on the Java website regarding J2SE SDK for your particular platform. The self extracting binary file method is recommended.

4) Download self extracting binary file to desired location on your system. Please be sure to download the appropriate executable file for your platform.
   Example: Linux 32 bit vs. Linux 64 bit Itanium 2

5) Follow java installation directions to correctly install J2SE SDK version 1.4.2 or higher on your machine.

6) Edit users .cshrc or .bashrc file to include JAVA_HOME. Set path to directory created in java install in step 5.

   For .cshrc: setenv JAVA_HOME path

   For .bashrc: export JAVA_HOME=“path”

7) Add JAVA_HOME/bin to PATH variable.

For cshell: PATH is located in the **.login** file.
Add the JAVA_HOME directory/bin to the end of the existing
set path= variable.
Ex: set path=(/bin /usr/bin /sbin /usr/etc /usr/local/bin /usr/j2se_1.4.2/bin)
Where /usr/j2se_1.4.2 is JAVA_HOME

For bourne shell: PATH is located in the .bash_profile file. Add
JAVA_HOME/bin to the end of the existing PATH variable.
Ex: export PATH=$PATH:$HOME/bin:$JAVA_HOME/bin

8) To ensure correct set up, please perform the following commands:

a) java –version
This will give you the version number of your java installation.

b) which java
This will tell you where your java installation is located.

9) Once J2SE SDK version 1.4.2 or higher has been installed correctly, download Java 3D SDK Version 1.3.1 or higher (fcs or non-Beta version only) into the directory created in step 5 from the following website:

http://www.blackdown.org/java-linux/java-linux-d1.html

Please ensure that you have downloaded the correct version for your platform.
Example: Linux 32 bit vs. Linux 64 bit Itanium 2
Please be sure to download the Java 3D license and Readme file as well.

10) Please read all installation documentation on the Java/Blackdown website regarding Java 3D SDK Version 1.3.1 or higher for your particular platform before install to ensure all requirements met.

11) Run tests detailed in installation documentation to verify correct installation of Java 3D SDK Version 1.3.1 or higher on your system.

To Download and Install Java and Java 3D on SGI :

1) Ensure that Java SDK version 1.4.1_06 or higher for IRIX is not installed. This install will **overwrite** any previous installation of Java SDK version 1.4.1 or lower.

2) Please read and verify all requirements and installation instructions from the SGI website before downloading the executable.

   http://www.sgi.com/products/evaluation/6.5_java2_1.4.1_06/

   Select the **Check Requirements** button at bottom of page.

   After viewing requirements, please read all installation documentation. Select the **Install** button at bottom of Check Requirements page. Agree to license agreement by selecting the **Accept License** button at bottom of page. Fully read the Installation Instructions page and then select the **Troubleshooter** button at bottom of page to see more install tips. Once all install documentation has been read, go back to the Installation Instructions page and select the **Install** button for java2_eoe.

3) After installation, edit users .cshrc or .bashrc file to include JAVA_HOME. Set path to directory created in java install in step 2.

   For .cshrc: setenv JAVA_HOME path

   For .bashrc: export JAVA_HOME="path"

4) Add JAVA_HOME/bin to PATH variable.

   For cshell: PATH is located in the **.login** file.
   Add the JAVA_HOME directory/bin to the end of the existing
   set path= variable.
   Ex: set path=(/bin /usr/bin /sbin /usr/etc /usr/local/bin /usr/java2/bin)
   Where /usr/java2 is JAVA_HOME

   For bourne shell: PATH is located in the .bash_profile file. Add JAVA_HOME/bin to the end of the existing PATH variable.
   Ex: export PATH=$PATH:$HOME/bin:$JAVA_HOME/bin

5) To ensure correct set up, please perform the following commands:

   a) java –version
      This will give you the version number of your java installation.

   b) which java
      This will tell you where your java installation is located.

6) Prepare to install Java 3D version 1.3.1 or higher for IRIX by checking requirements on the following page:

http://www.sgi.com/products/evaluation/6.5_java3d_1.3.1/

Select the **Check Requirements** button at bottom of page.

After viewing requirements, select the **Continue** button at bottom of page. Agree to license agreement by selecting the **Accept License** button at bottom of page. Fully read the Installation Instructions page and the **More Instructions** page located in step 3 of Installation Instructions page. Then select the **Troubleshooter** button at bottom of page to see more install tips. Once all install documentation has been read, go back to the Installation Instructions page and select the **Install** button located in step 2.