



**Tomography
V5.3.14
Documentation**

Tsunami Development
tsunamidevelopment.com
713-783-1435

Tsunami Development

713-783-1435

tsunamidevelopment.com

The Information contained in this document is subject to change without notice. Tsunami Development assumes no responsibility for any error that may appear in this manual. Tsunami Development does not warrant that this document is error free. Please report any errors in this document in writing to Tsunami Development. Some states or jurisdictions do not allow disclaimer of expressed or implied warranties in certain transactions; therefore this statement may not apply to you.

CONTENTS

Introduction.....	4
Frequently Asked Questions.....	6
Installation.....	8
System Requirements.....	9
Installation of Tsunami.....	9
GUI Installation.....	10
INTViewer Installation.....	12
Floating License.....	20
Cluster Configuration.....	21
Node Utility.....	23
Getting Started.....	26
Start INTViewer.....	26
Picking Moveout on Gathers.....	27
Start Tomography.....	37
Generating an Internal Velocity File.....	38
Files Tab.....	39
Stack File Tab.....	41
Velocity File Tab.....	43
Processing Tab.....	45
Starting and Stopping a Job.....	47
Appendix A : Log file Information.....	49
Example Log file.....	49
Example Check License Log.....	54
Example Clear Nodes Log.....	55
Appendix B : Parameter File Information	56
Required Parameters.....	56
Alphabetical List of Parameters.....	58
Example Parameter File.....	61
Appendix C: Java Installation Information.....	62

Tomography Introduction

The objective of Tomography is to provide an easy to use tool for generating accurate velocity models for PSDM. The program is used in conjunction with Tsunami's PSDM and Raytracer programs. The first step in the process is to run the Tsunami PSDM with an existing velocity model. This is an interval velocity model which can be created from any number of sources. The user may use a velocity modeling package, or a conversion of RMS velocities from a model used for PSTM. A great deal depends on the complexity of the geology.

Once the initial PSDM is complete, the user picks the moveout on the depth gathers and saves the picks for input to Tomography. See the Getting Started chapter of this user manual for details on picking gathers and saving the picks.

The moveout traced by the user describes to Tomography the error in the velocity model. The tomography program uses the ray traced travel time tables, saved from the migration, and the user picks, to construct source-receiver ray paths at the pick locations. The moveout described by the picks is then analyzed, in conjunction with the ray paths, to determine the necessary change in velocity (Δv) which will remove the moveout from the gather. Because Tsunami Tomography uses the travel time tables from the depth migration, there is a tremendous time savings.

The input to the tomography program then becomes the picks made by the user, the previous velocity model, the travel time tables saved from the depth migration, and optionally a stack volume. If a stack volume is input, then the tomography program will automatically calculate the structural dip at the pick locations. The dip is used to skew the raypaths to more accurately represent the source and receiver locations for the picks.

The user can optionally set a minimum and/or maximum velocity to update. This is used to restrict the process from updating water, or salt velocities. It is also possible to restrict the update to an area below a digitized horizon. This allows the user to update the velocity model in a top down sequence, holding the shallow velocities constant, while updating velocities at depth. Tomography produces an updated interval velocity model which is then used to remigrate the data using Raytracer and PSDM.

There are significant advantages to the way the Tsunami Tomography is implemented. The most important advantage, from an accuracy perspective, is that the full solution is calculated in the depth domain. Many applications convert the gathers to the time domain before calculating semblances, or use the traditional normal moveout equation to determine the necessary velocity correction. These calculations are then in error because they are made using time migration equations. In addition the conversion from depth to time uses the velocity model that is being updated, and is therefore incorrect. Tsunami Tomography does no depth to time conversion, but calculates the necessary changes in the velocity model in the depth domain, using the ray paths derived from the existing travel time tables. So it is using the same travel times that the migration used, and is therefore consistent with the migration gathers that are being used for the model update.

Some vendors suggest that the tomography process only be used when the velocities are within a couple of percent of being correct. They also often suggest that the number of input locations be relatively few. These suggestions are made because the inversion process that is used is relatively unstable, and large changes in velocities, or lots of data points can create wildly unpredictable velocity results or extremely long runtimes. Tsunami Tomography does not have these limitations. The more pick locations the user provides the more accurate and stable the result. Of course the runtimes will increase as the number of pick locations increase, but at a modest rate.

Relatively large velocity corrections can also be made. For example, local corrections of eight to ten percent can be made in the velocities, while still producing a stable and accurate result. Velocities can be simultaneously decreased at one depth and increased at another depths in a consistent manner.

This method does rely on the signal being good enough that the user can see the moveout on the gathers. Depth migration is normally applied to surveys with at least decent signal to noise ratios. Improvements in acquisition technology have also improved the expected signal quality in surveys. However, in the event that the signal strength is weak, one can take any number of steps to improve signal quality. Processes like creating super gathers from neighboring cdps, or filtering the gathers to remove noise will improve the user's ability to accurately pick the moveout. The moveout being picked is a fairly low frequency event, so aggressive filtering will not hurt the process.

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
                             tomo_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/tomo_5.3.14/README_RAYS
                                     itanium
                                     linux
                                     opteron
                                     opteron32
                                     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 1.5.1 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 you 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.

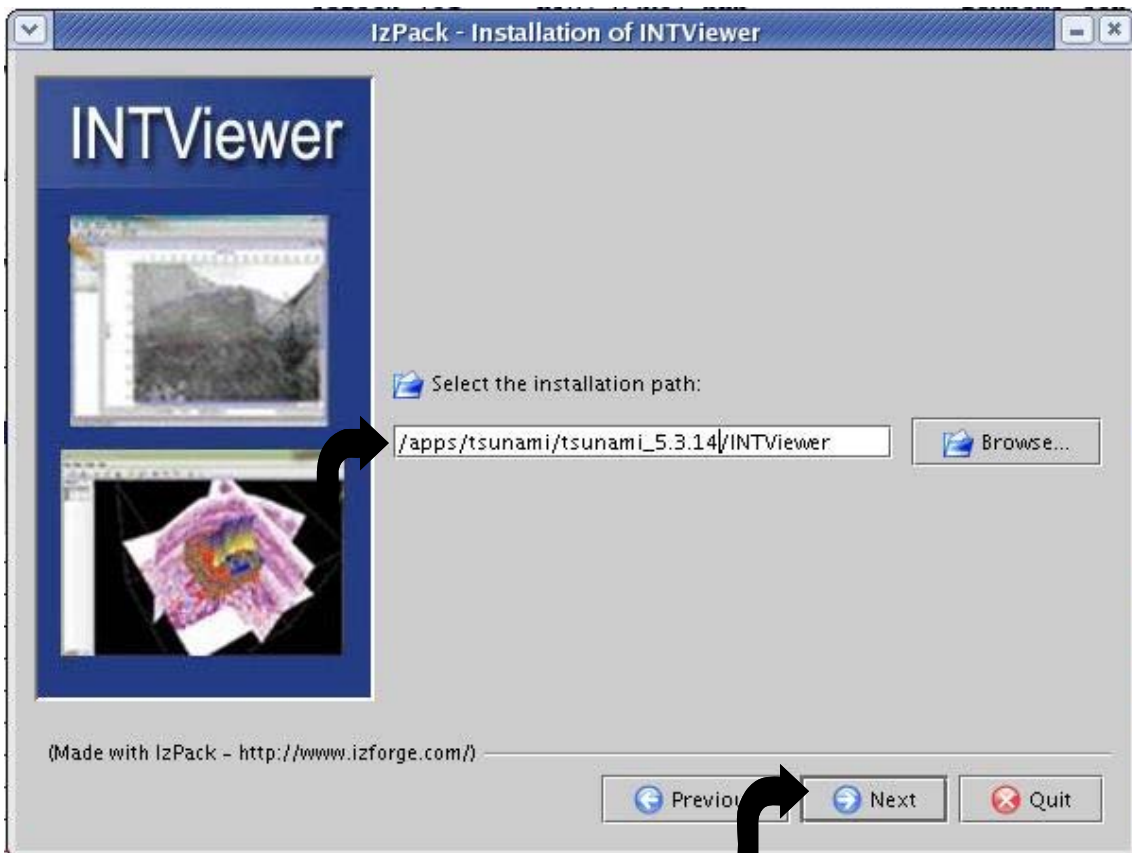
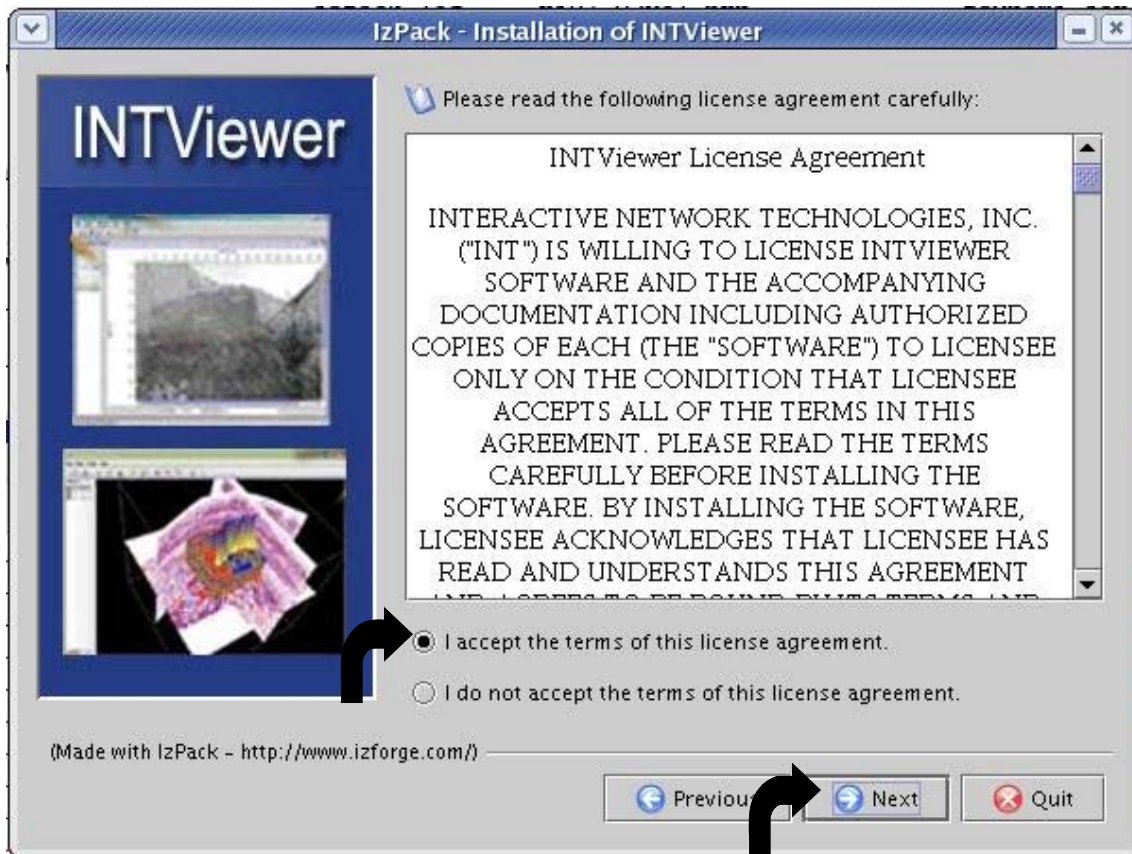
For .bashrc: `alias tsunami="/apps/tsunami/tsunami_5.3.14/tsunami.sh"`

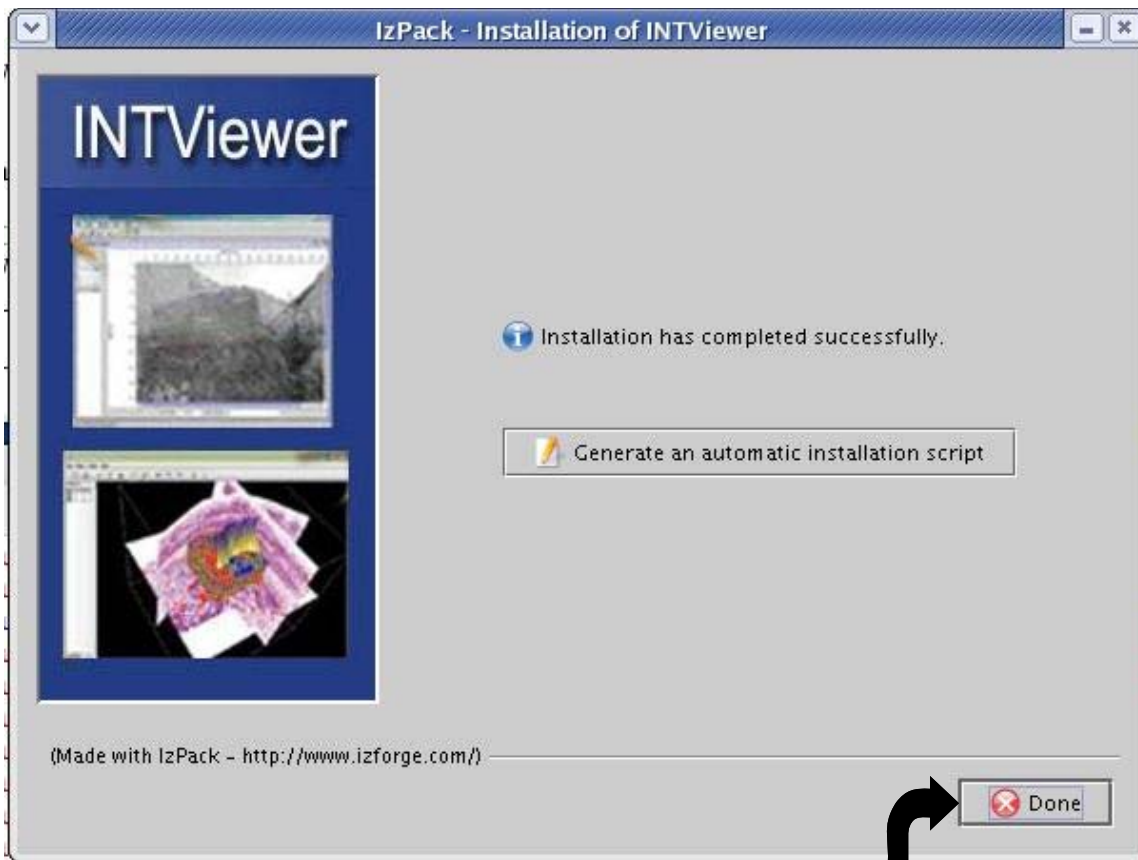
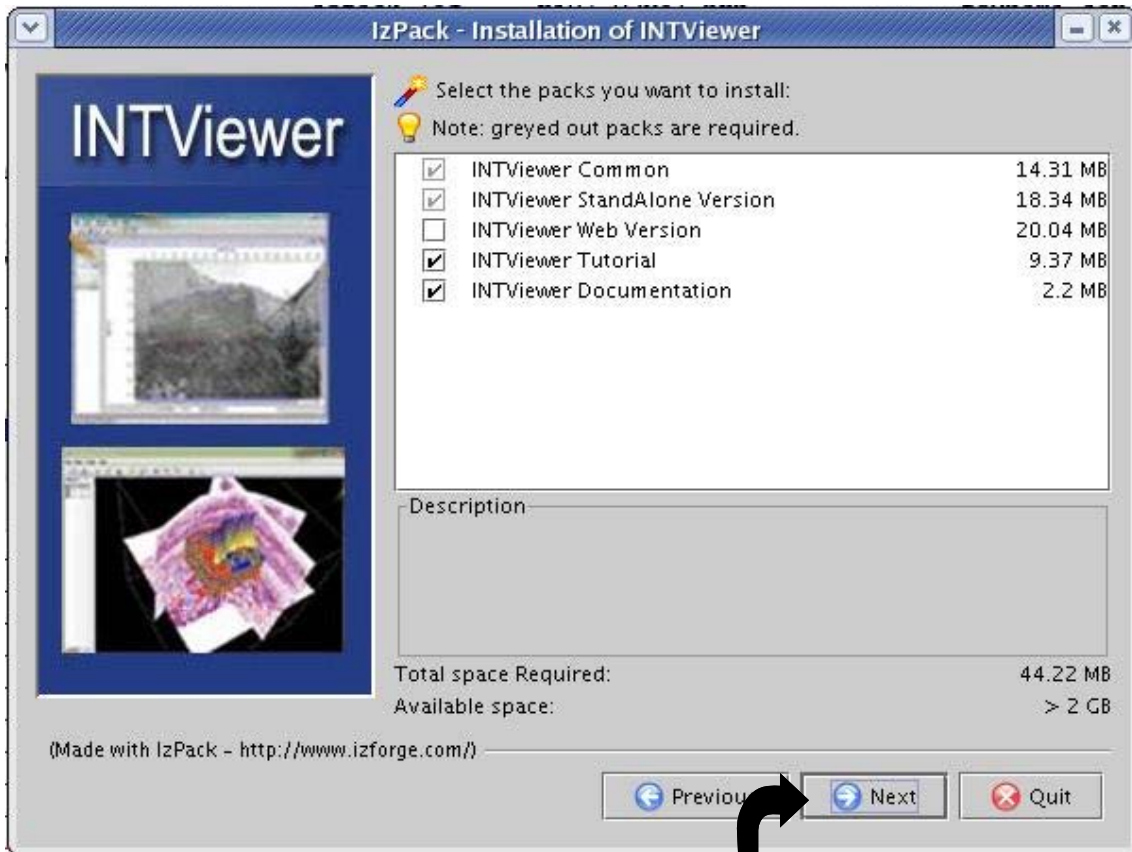
For .cshrc: `alias tsunami /apps/tsunami/tsunami_5.3.14/tsunami.sh`

INTViewer Installation Procedures For Windows

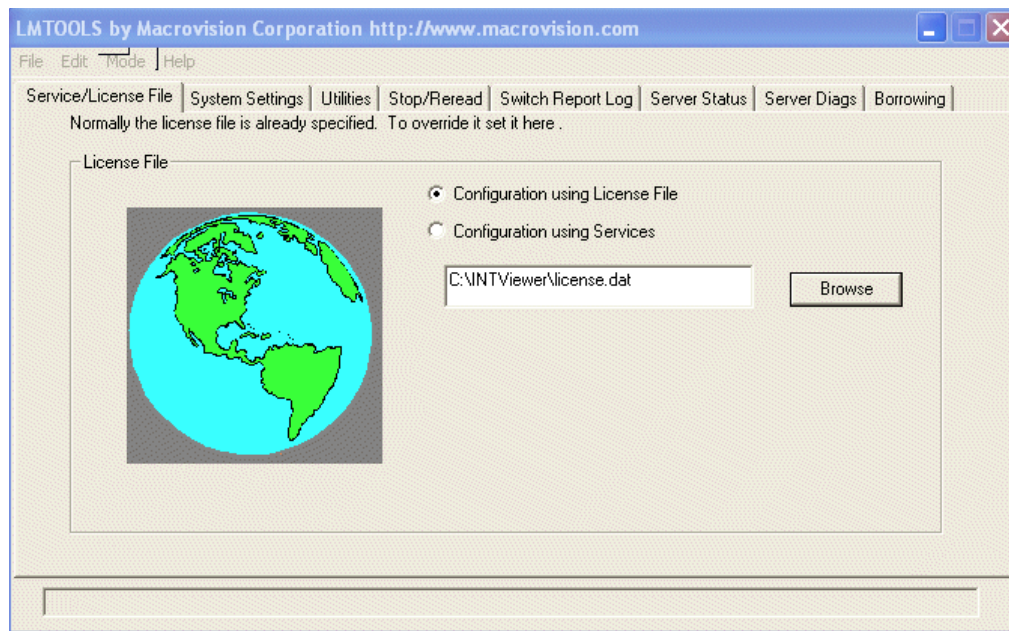
- 1) Ensure Java SDK version 1.5.1 or higher installed on your machine.
- 2) Obtain latest version of INTViewer from Tsunami Development.
- 3) Double click on executable jar file from step 2: intviewer-flex_install.jar
- 4) Go through GUI install.







- 5) Move license.dat file into the INTViewer directory created in the Choose Install Folder step.
- 6) Open the flexnet/Windows directory.
- 7) Double click the lmtools icon.
- 8) On the Service/License File tab, set to Configuration using License File. Browse for the license.dat file you moved in step 5. Then exit lmtools.



- 9) Start the INTViewer by double clicking on the INTViewer.bat file in the C:\\INTViewer directory or the directory structure specified in the Choose Install Folder step.

INTViewer Installation Procedures For UNIX

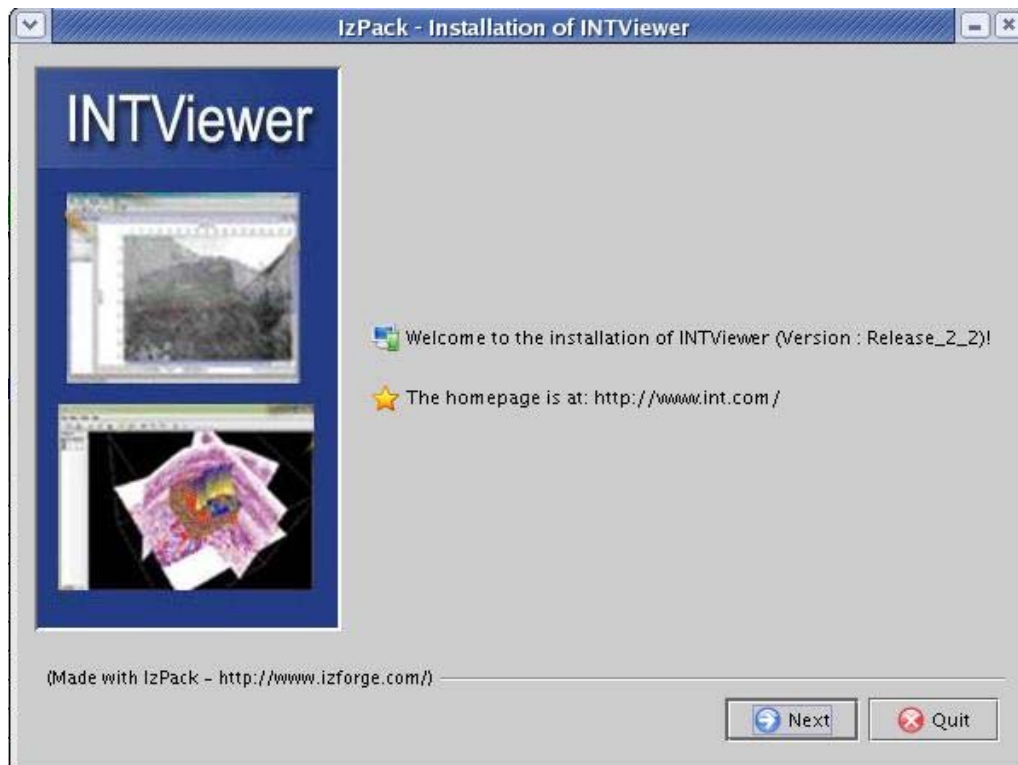
- 1) Ensure Java SDK version 1.5.1 or higher installed on your machine.
- 2) Change directory to \$TSUNAMI/tsunami_5.x.x/INTViewer
- 3) At prompt, issue following command:

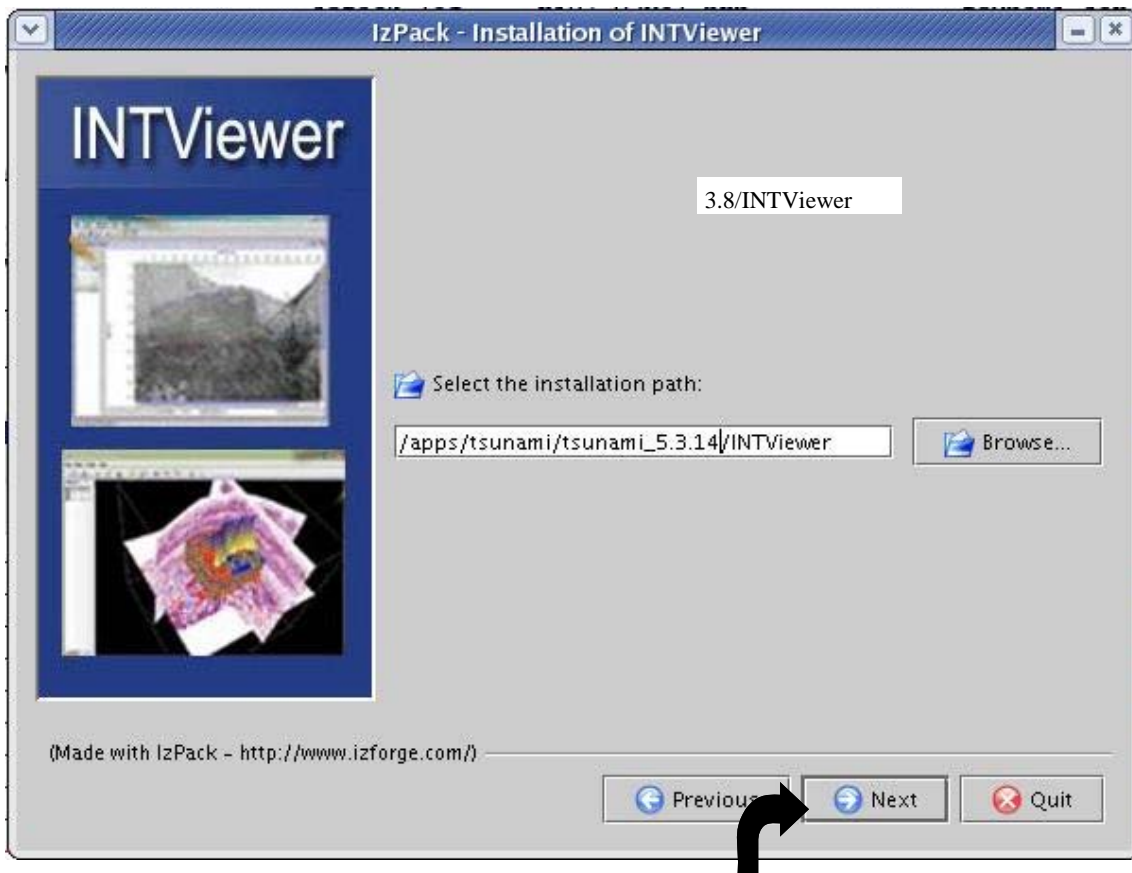
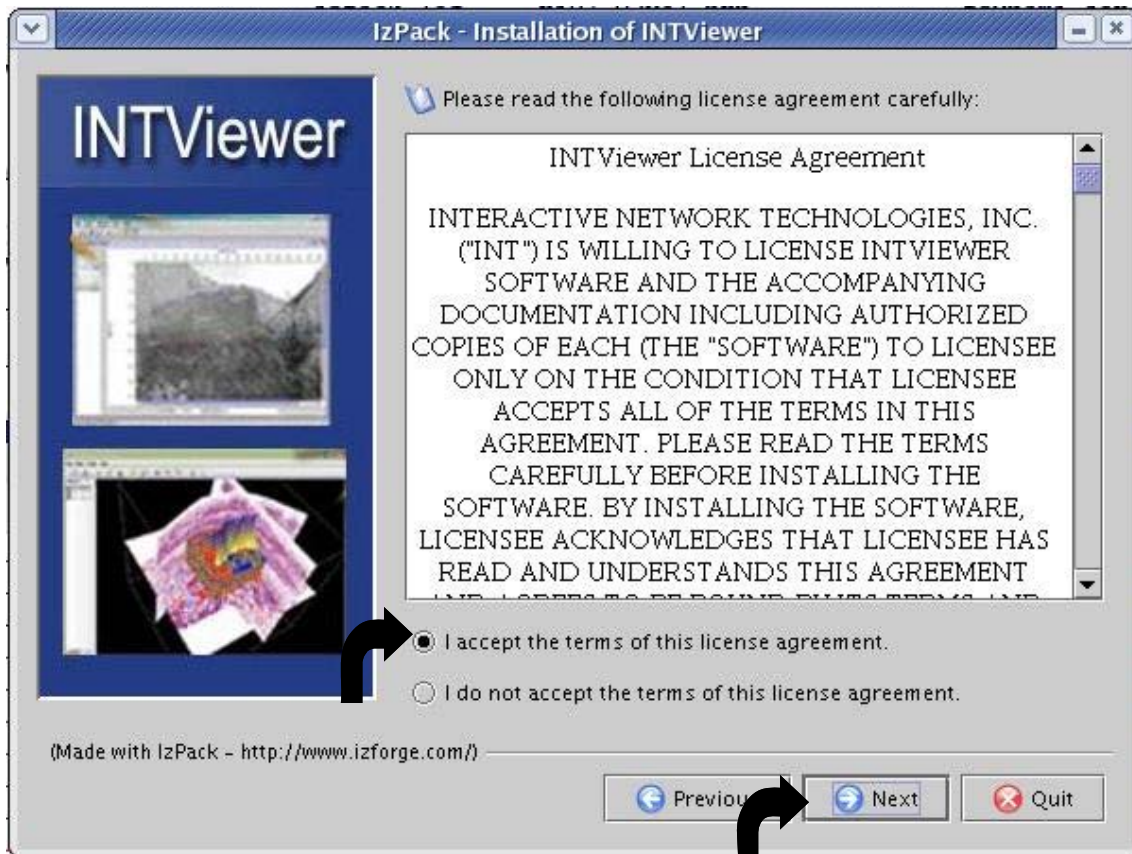
```
java -jar intviewer-flex_install.jar
```

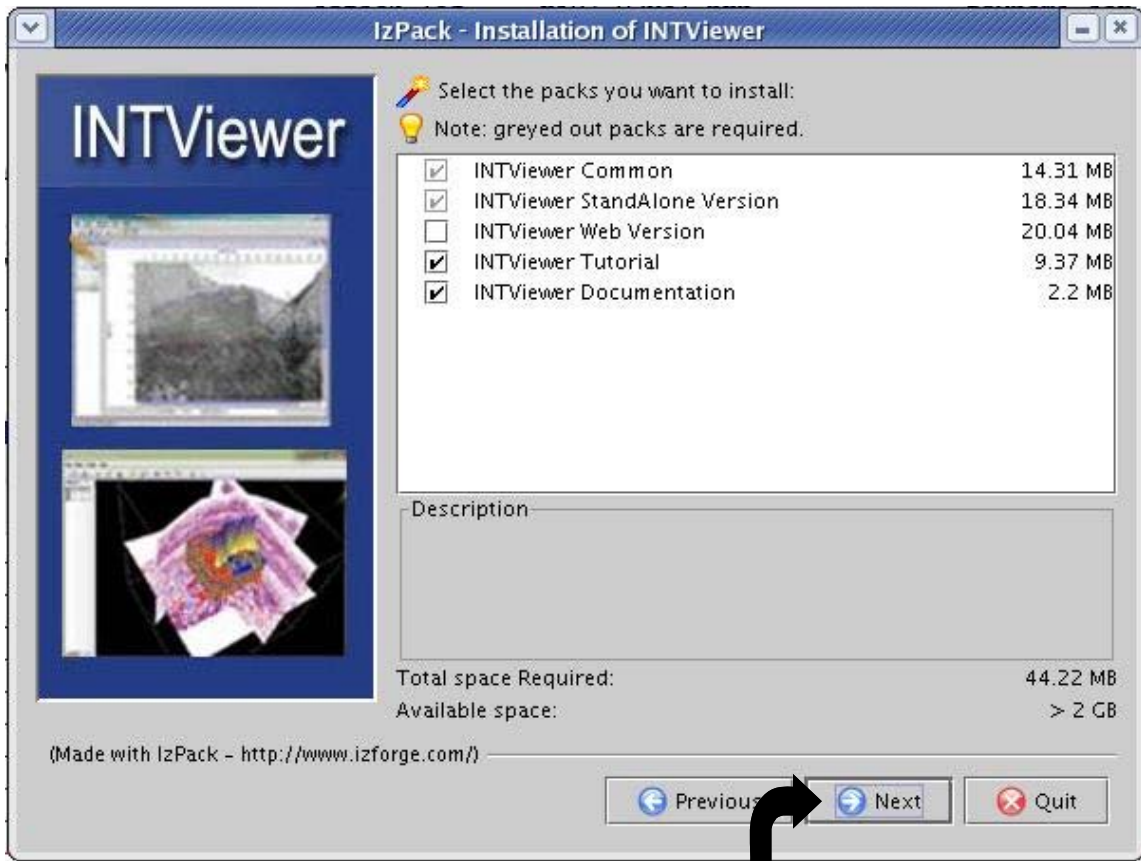
where intviewer-flex_install.jar is the file located in:

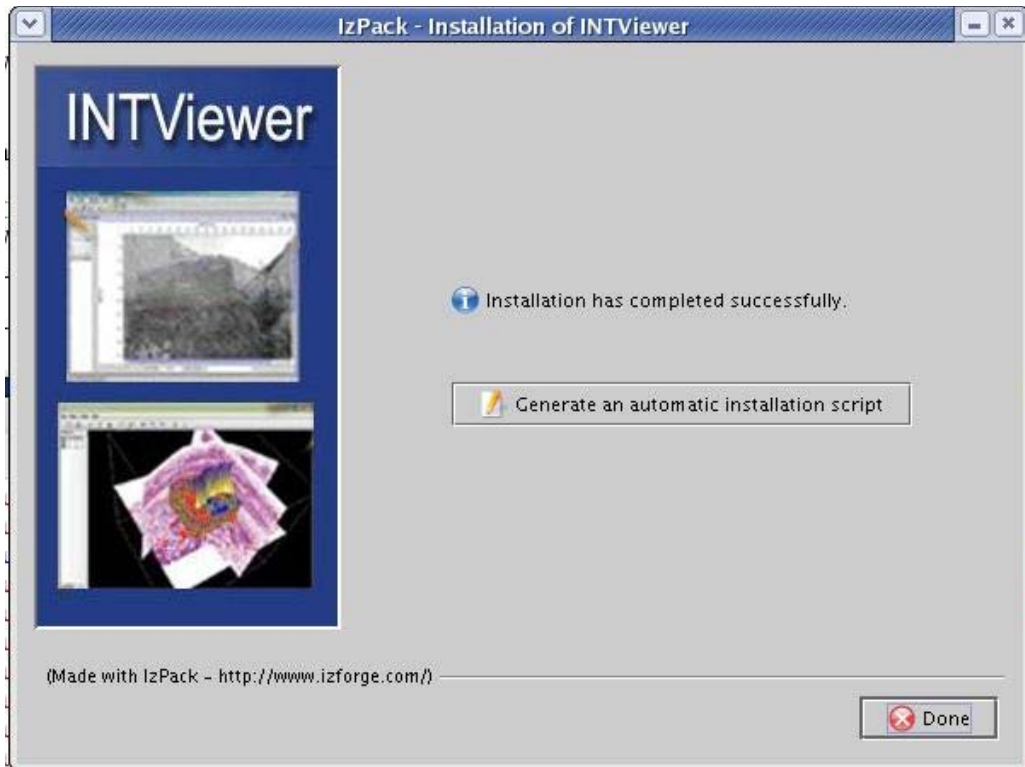
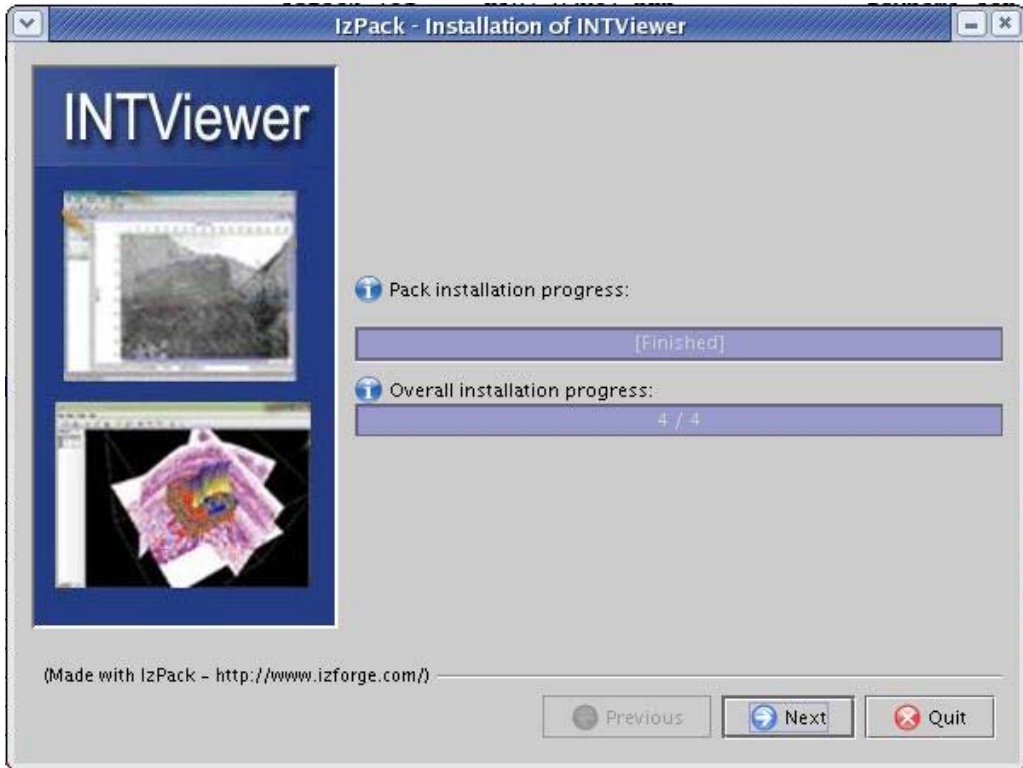
```
$TSUNAMI/tsunami_5.x.x/INTViewer
```

- 4) Go through GUI install.









- 5) Move license.dat file into INTViewer directory created in Choose Install Folder step.
- 6) Cd to .../INTViewer/flexnet and then into the directory that best describes your system: Linux, Unix or Solaris.
- 7) To start the license server use the following command:
`./lmgrd -c $TSUNAMI/INTViewer/license.dat`
- 8) To start INTViewer, run the following command in your INTViewer directory:
`./INTViewer.sh`
- 9) To be able to start the INTViewer from any location you will need to edit the INTViewer.sh file. Change the INTVIEWERPATH= variable in the file to be the absolute path to the INTViewer directory.

ex. Original: INTVIEWERPATH=.

New: INTVIEWERPATH= /apps/tsunami/tsunami_5.x.x/INTViewer

- 10) Once step 9 has been done, you will be able to set up an alias in your .cshrc/.bashrc file to start up the INTViewer.

ex for .cshrc: `alias intviewer $TSUNAMI/INTViewer/INTViewer.sh`

ex for .bashrc: `alias intviewer="$TSUNAMI/INTViewer/INTViewer.sh"`

- 11) To be able to start the INTViewer from the GUI, you must have the following variable set in the users .cshrc/.bashrc:

ex. For .cshrc: `setenv INTVIEWER_HOME /apps/tsunami/tsunami_5.x.x/INTViewer/INTViewer.sh`

ex. For .bashrc: `export INTVIEWER_HOME="/apps/tsunami/tsunami_5.x.x/INTViewer/INTViewer.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).

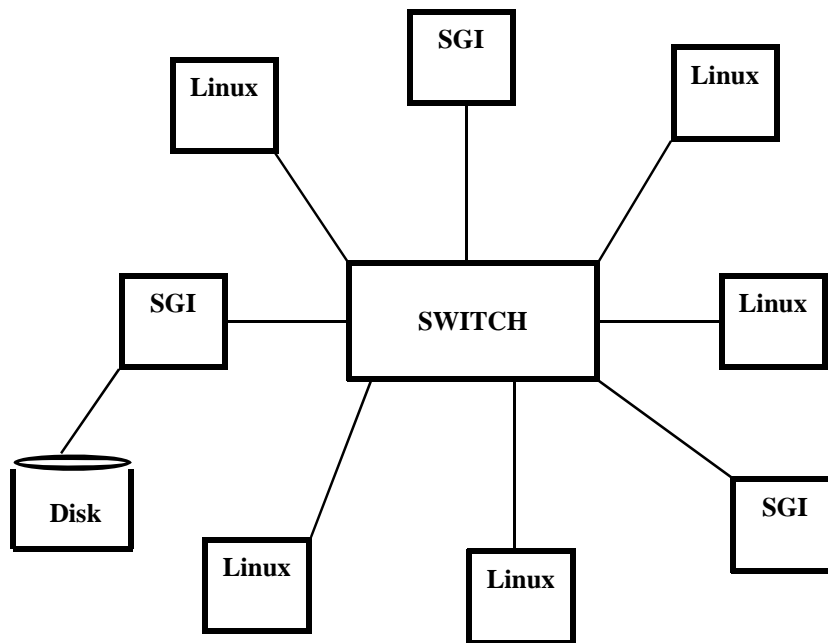
The server node must be the first entry in the db. This is the node from which the job is launched, and is the only node that must have access to the input data, and the file system where the output data will be written. The server may also be a compute node, in that case the server should be listed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

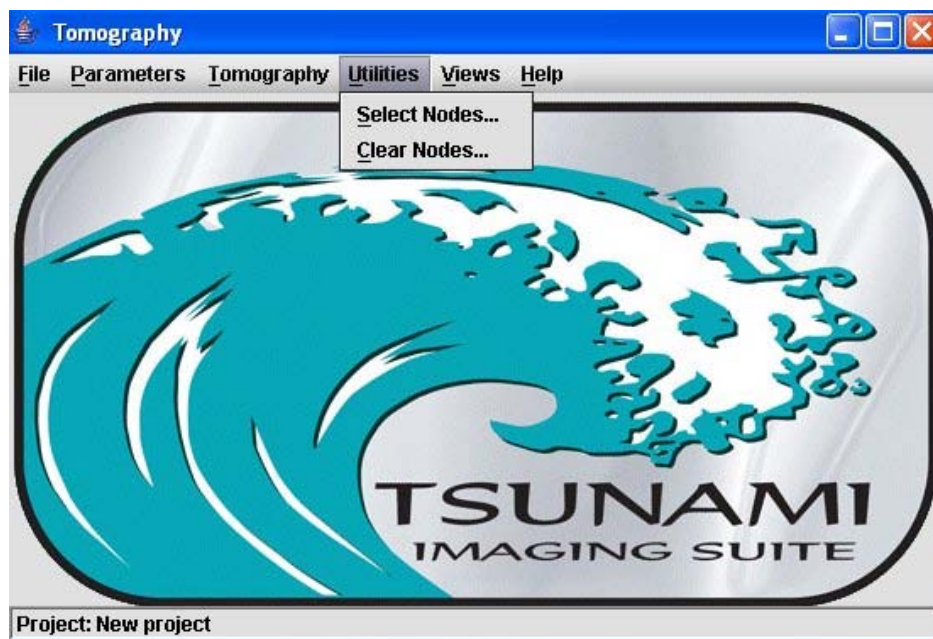
Example Cluster Configuration



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.

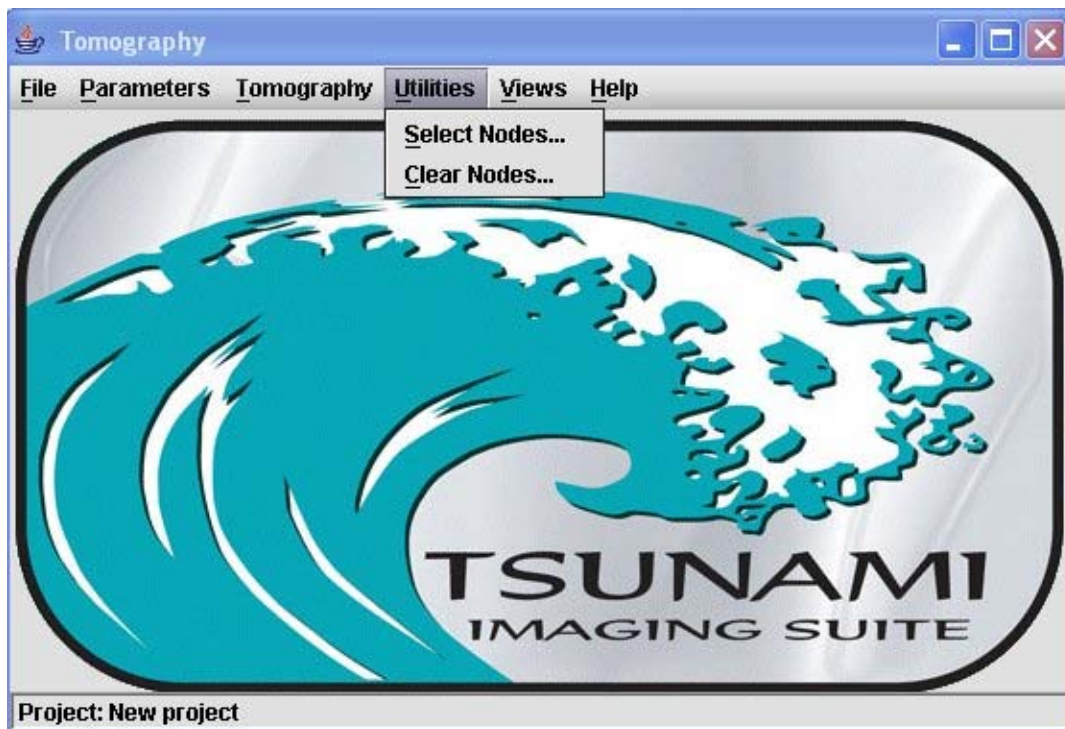
To open a node list file or to create a new one, the program must first search for a master node list file. The master node list should be located in \$TSUNAMI. Where \$TSUNAMI = the tsunami directory - such as "/apps/tsunami". If no master node list is found, the user will then be prompted to enter 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).



- 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

Tomography uses the residual moveout from PSDM migrated gathers to update the velocity model. Before running Tomography, you must first run Tsunami PSDM with an existing velocity model. This is an interval velocity model which can be created from any number of sources. The user may use a velocity modeling packing, or a conversion of RMS velocities from a model used for PSTM. A great deal depends on the complexity of the geology.

Once the initial PSDM is complete, the user displays the depth gathers using the INTViewer, a SEG-Y display tool distributed by Tsunami Development. The user then picks the moveout on the depth gathers and saves the picks for input to Tomography.

The input to the tomography program then becomes the picks made by the user, the previous velocity model, the travel time tables saved from the depth migration, and optionally a stack volume. The output from the tomography will be an updated velocity model. The updated velocity model is used to calculate new travel times and depth migrated gathers using Tsunami's Raytracer and PSDM.

Start INTViewer:

The INTViewer can be started by

- 1) selecting Utilities → Start INTViewer from the Tomography main window.
- 2) By changing directories to `$TSUNAMI/tsunami_5.2.13/INTViewer` and running the `INTViewer.sh` file.

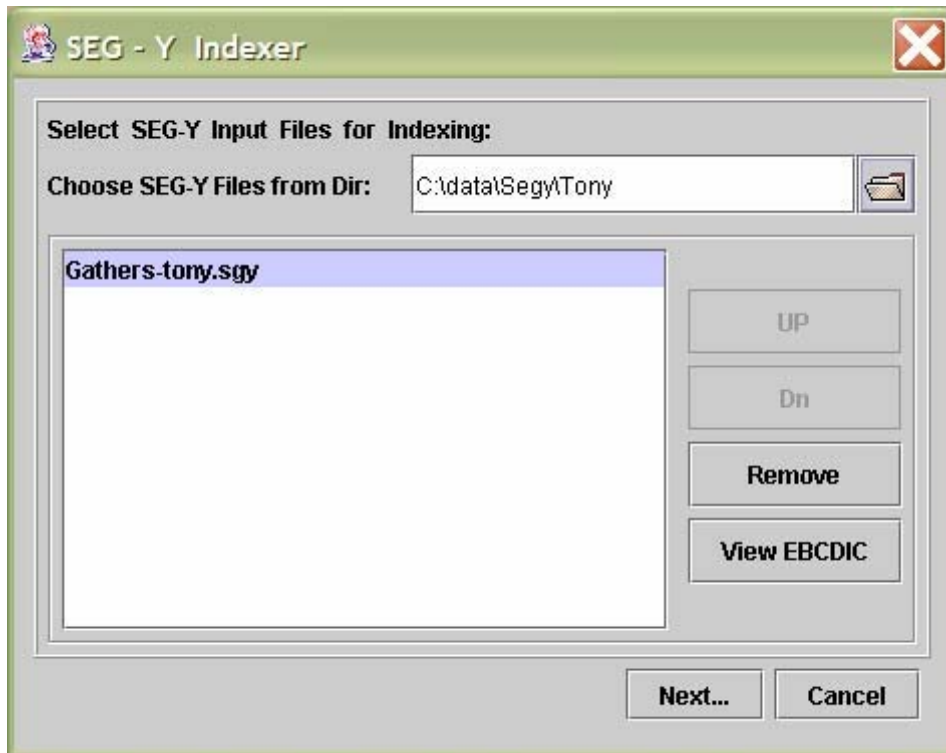
```
ex. cd $TSUNAMI/tsunami_5.2.13/INTViewer
     ./INTViewer.sh
```

- 3) Use the `intviewer` alias set up in the INTViewer installation to start program from desired directory.
Please see the *INTViewer Installation* section for more information.

Picking Moveout on Gathers:

Step 1 – Indexing the gathers file

The first step is to index your gathers file using the SegyIndexer utility. Run the utility and select the Segy file to index using the File selection dialog as shown below.



Since trace header values in most Segy files are not always standard, you will need first to find out which keys are used to encode the Inline, Xline and Offset values. This information can often be found in the EBCDIC header, which you can view using the View EBCDIC option. The description of the trace header is encoded into a template file. Two examples are provided in your distribution: StandardSegy.xml and PreStackSegy.xml. Those files are XMLformatted files and you can edit them using any ASCII editor. They can also be viewed using most web-browsers.

For our example gather file we use template file PreStackSegy.xml. The Inline value is encoded in field INLINE at byte location 180 (counting from 0), the Xline value is encoded in field XLINE at byte location 184 and the Offset value is encoded in field at byte location 36. Please note that the actual names for the keys (INLINE, XLINE and OFFSET in this example) have no particular meaning and can be anything you want.

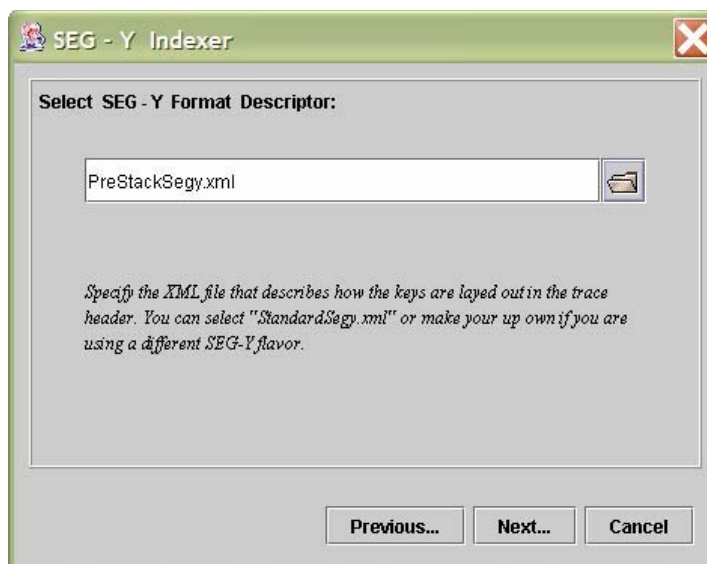
```

-<Segy>
  <Header Name="EbcDic" Format="EBCDIC" Size="3200" Place="Line" />
  <Header Name="BinaryHeader" Format="BINARY" Size="400" Place="Line" Fields="Segy"
  />
-<Header Name="TraceHeader" Format="BINARY" Size="240" Place="Trace">
  <Field Name="OFFSET" Format="UINT32" Offset="36" />
  <Field Name="INLINE" Format="UINT32" Offset="180" />
  <Field Name="XLINE" Format="UINT32" Offset="184" />
  <Field Name="FIELD REC" Format="UINT32" Offset="8" />
  <Field Name="FIELD TR" Format="UINT32" Offset="12" />
  <Field Name="CDP" Format="UINT32" Offset="20" />
  <Field Name="CDPTR" Format="UINT32" Offset="24" />
  <Field Name="RCV ELEV" Format="UINT32" Offset="40" />
  <Field Name="SRCX" Format="UINT32" Offset="72" />
  <Field Name="SRCY" Format="UINT32" Offset="76" />
  <Field Name="RCVX" Format="UINT32" Offset="80" />
  <Field Name="RCVY" Format="UINT32" Offset="84" />
  <Field Name="CDPX" Format="UINT32" Offset="188" />
  <Field Name="CDPY" Format="UINT32" Offset="192" />
  <Field Name="SHTPT ID" Format="UINT32" Offset="16" />
  <Field Name="SHTPT HUM" Format="UINT32" Offset="196" />
Header> Segy>

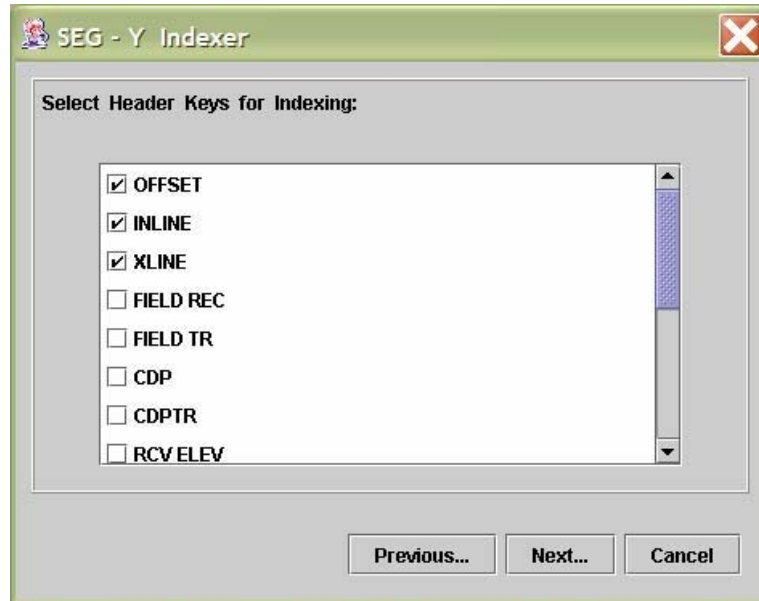
```

Hopefully you will be able to use one of the templates provided. If not, create your own by copying one of the provided template and make modification to match the header fields in your dataset. Please remember that the Offset field starts at 0, so you may have to subtract 1 compared to what you are used to. Refer to the INTViewer ReadMe.html for a more complete description of the template format.

The next step is to select the template file you want to use using the file selection dialog as shown below: Once you have selected the template, you will be presented with the list of header keys specified in the template. Select the keys you want to use to index the Segy data file. In this particular case we have selected INLINE, XLINE and OFFSET.

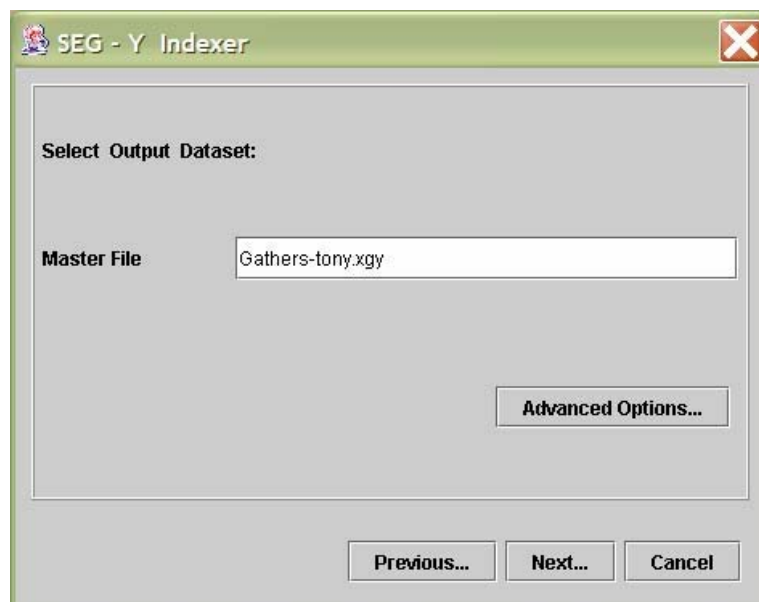


As a general rule to indexing, you want to specify the minimum set of keys to uniquely access your data.

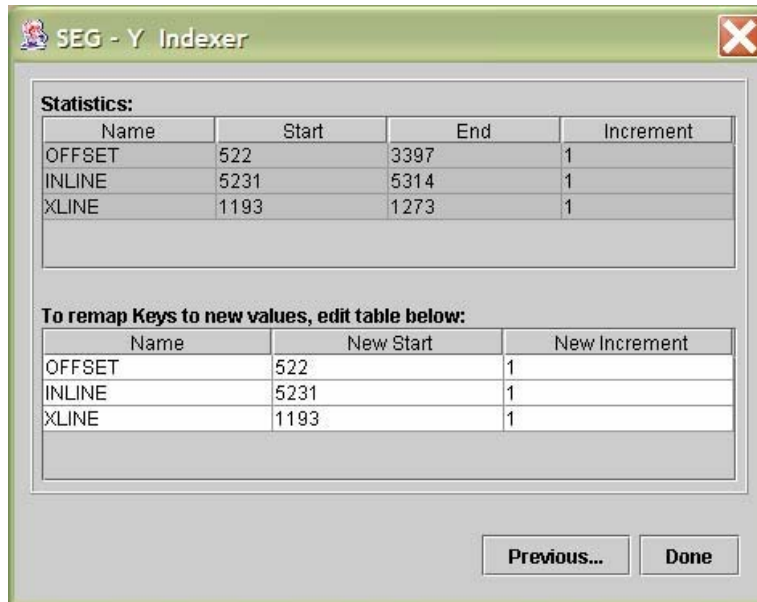


In the following step, you need to specify the output dataset or master file. In our case, we just accept the default name. The master file has suffix “.xgy”. This is the file that we will reference when building the display in INTViewer.

Please note that indexing does not create a new copy of the data. It simply creates a binary index file (less than 1% of the original Segy file size) and a master file. The master file is a very small XML formatted file that ties together the original Segy file, the index file and provides some statistics about the indexing.



You are now ready to start the indexing job. Simply press Next... and then the Start button. A progress status is shown during indexing. When finished, you will be presented with statistics about the indexing. In our test dataset, as shown below, there are 13 unique OFFSET values, 6804 unique INLINE values and 81 unique XLINE values. Finally, the second table let you specify a new start and increment value for any of the indexed keys.



Step 2 – Building the gather View

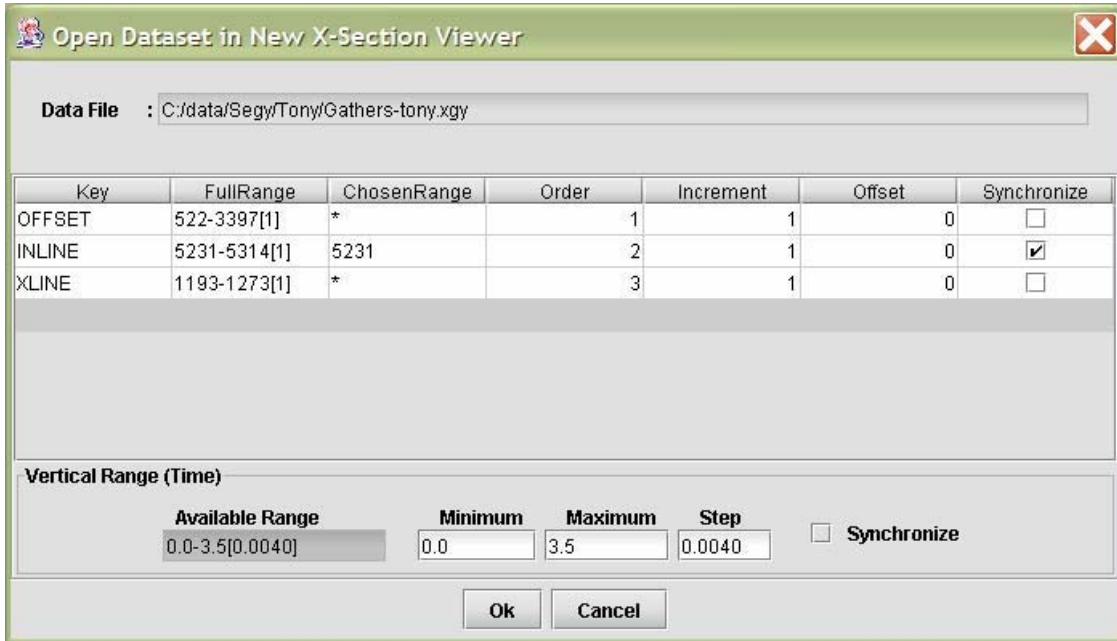
a. Load the data

Start INTViewer and select option:

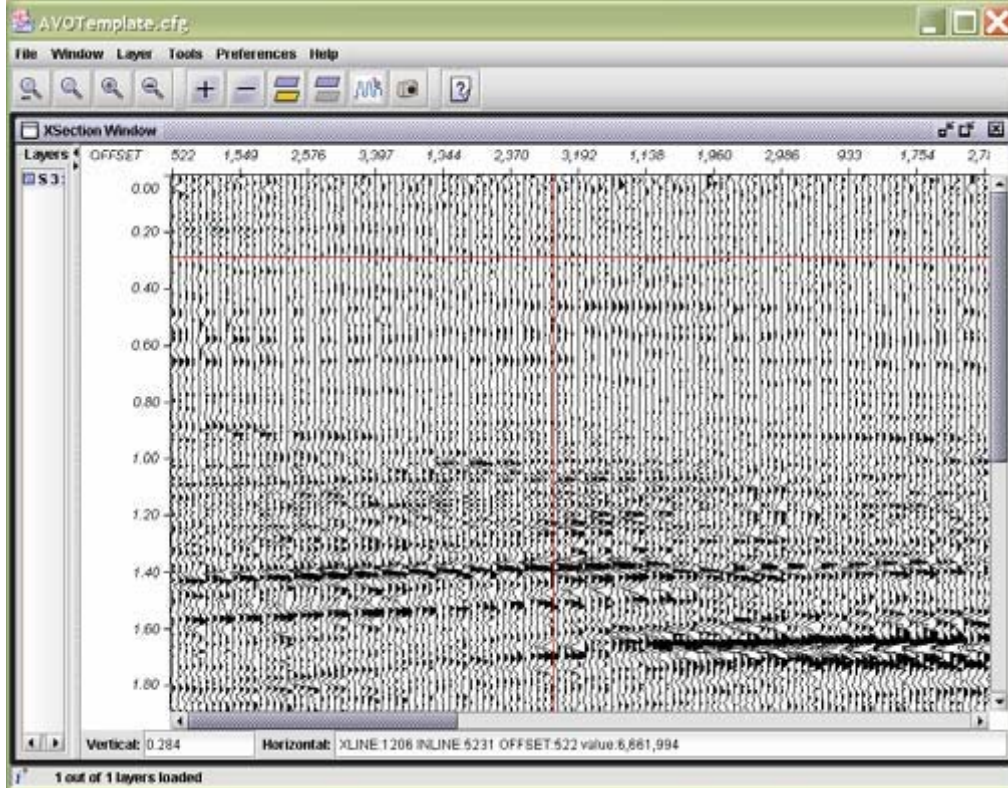
File->Open Data In New X-Section Window ->Indexed-Segy...

Select the master file (Gathers-tony.xgy in our case) that you have created in Step 1. Once this is done, you will be presented with a dialog window similar to the one shown below.

Enter an INLINE value under the ChosenRange field (we used the first INLINE, 5231 in our example) and also check the Synchronize box for INLINE.



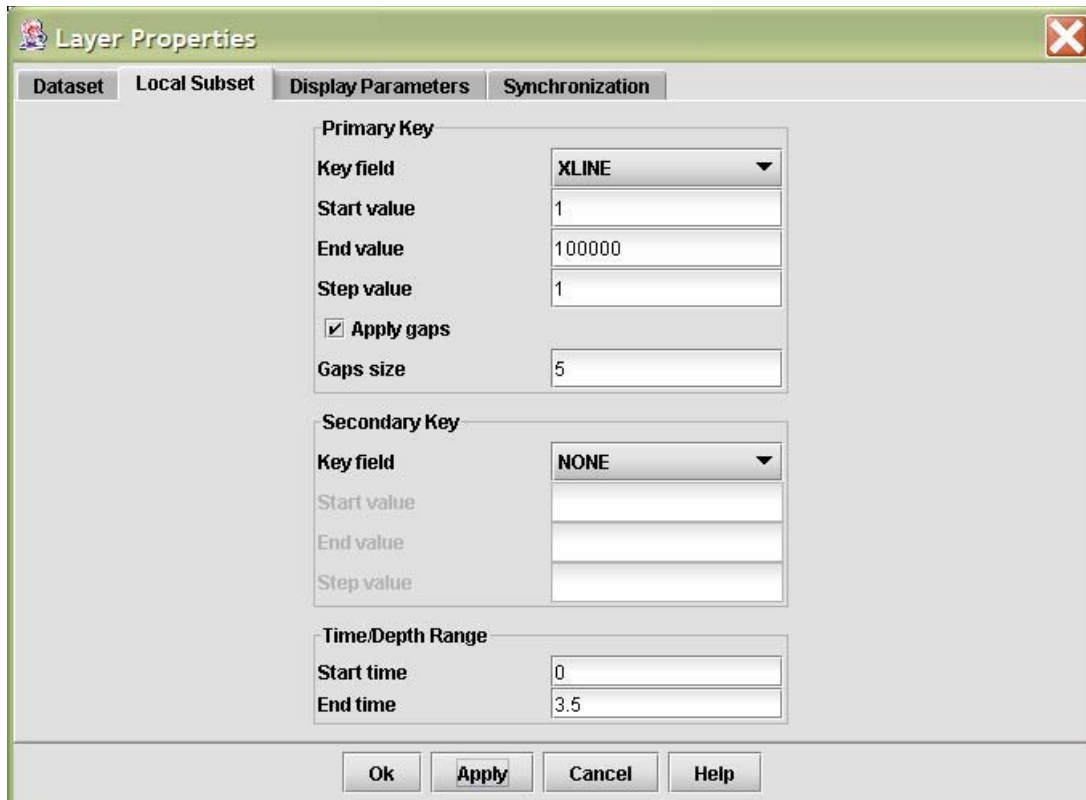
Press OK and you will get the first INLINE of gathers to display as shown below. Please note that if you want to pick along the XLINE direction, you can just select a XLINE value instead. Also make sure you check the Synchronize check box for the XLINE key.



b. Adding the gaps

To specify the gaps, double click on the layer name in the upper left corner (S0: Segy) of the newly created window, or choose menu option Layer->Property... You will obtain a new dialog box. Select tab Local Subset as shown below.

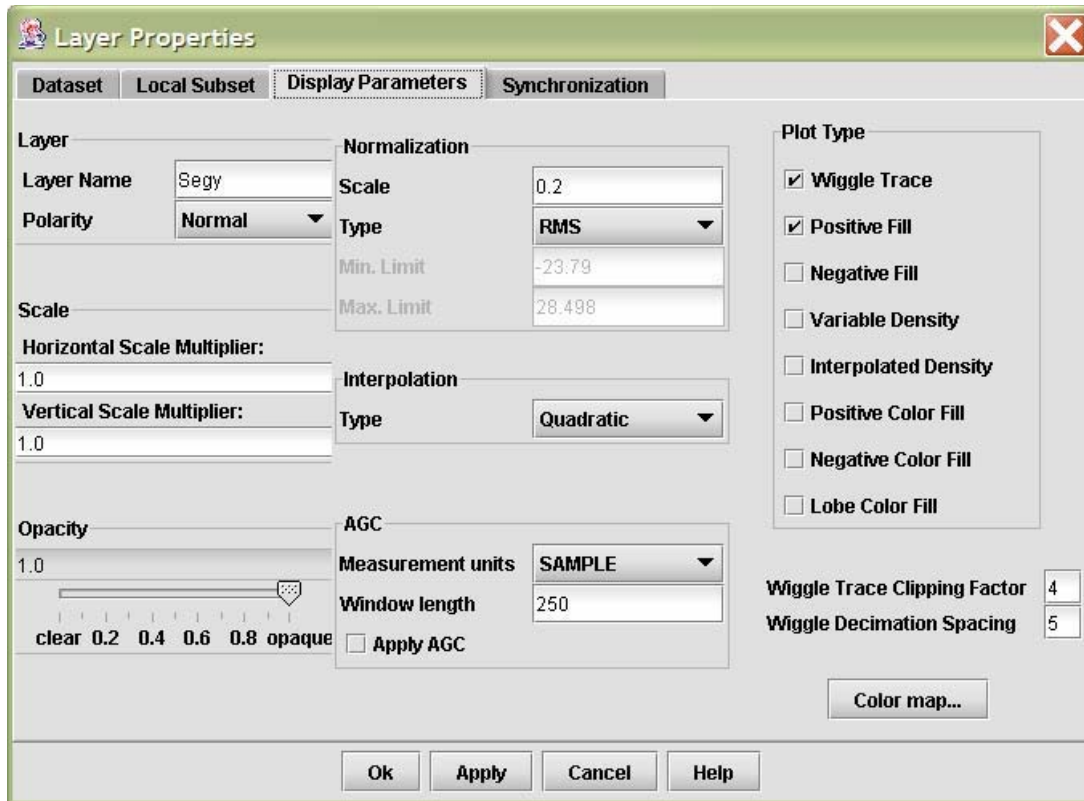
This panel lets you select a subset of the data already selected using any header key value. It can also be used to turn gaps on. To do so, select for the primary Key field value XLINE (if you have selected an INLINE view) or INLINE (if you have selected a XLINE view).



c. Configure the Display Parameters

Select tab Display Parameters as shown on next page.

First, you may want to play with the Normalization or AGC options to control the gain. Then depending on the scale you selected you may notice that in wiggle mode not all the traces are visible. This is controlled by Wiggle Decimation Spacing at the bottom right of the panel. This field controls the minimum spacing in pixels between traces. Change it to 3 if you want to see more traces. Select Ok when you are done.



d. Configure the Labels and Annotation

The default view provides an annotation for the first key in your dataset (OFFSET in our case). We want to change it to display INLINE and XLINE values. First select menu option Window->Annotation... and select the Horizontal tab option. You will obtain a dialog as shown below.



To add an INLINE and XLINE axis, simply select INLINE in the list of Available Keys and press the \hat{I} key. You can then do the same for XLINE. To remove OFFSET, select it from the list of Selected Keys and press the \hat{A} key. You can then use the Up and Down keys to reorder the axes. Press Ok when done.

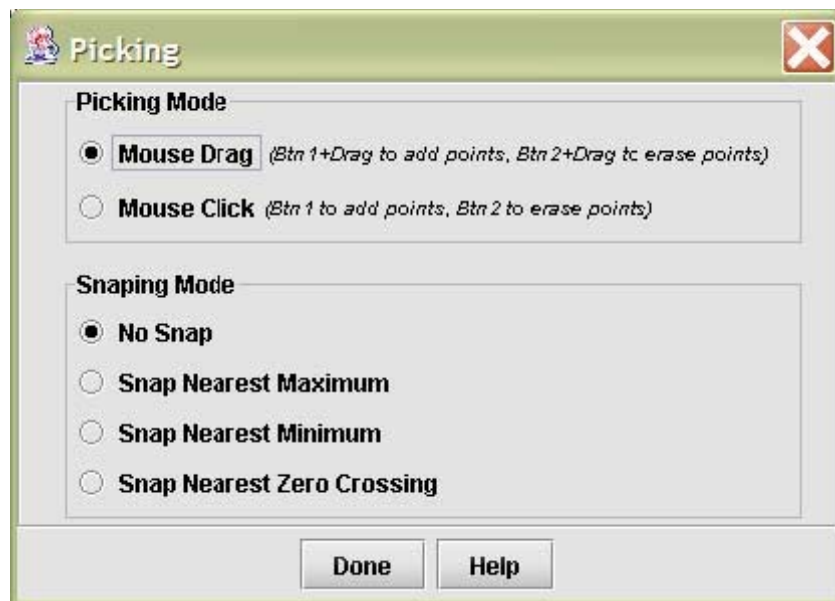
You can also check the Misc. tab for additional annotation options, including specifying a plot title, horizontal and vertical annotation, etc..

Step 3 – Create a Horizon

To create a horizon, select Layer->Create New. You will get a dialog that asks you if you want to collect amplitudes for this horizon. Select No. Once this is done, a new layer name should appear in the Layers list.

You can start picking on your gathers using mouse Left button (picking is active when the picking layer name is selected). The default picking behavior is to track the mouse while the Left button is pressed (Mouse Drag mode) with No Snap set as your Snapping Mode.

Other options are available using option Preferences->Picking or by selecting the icon in the menu bar.



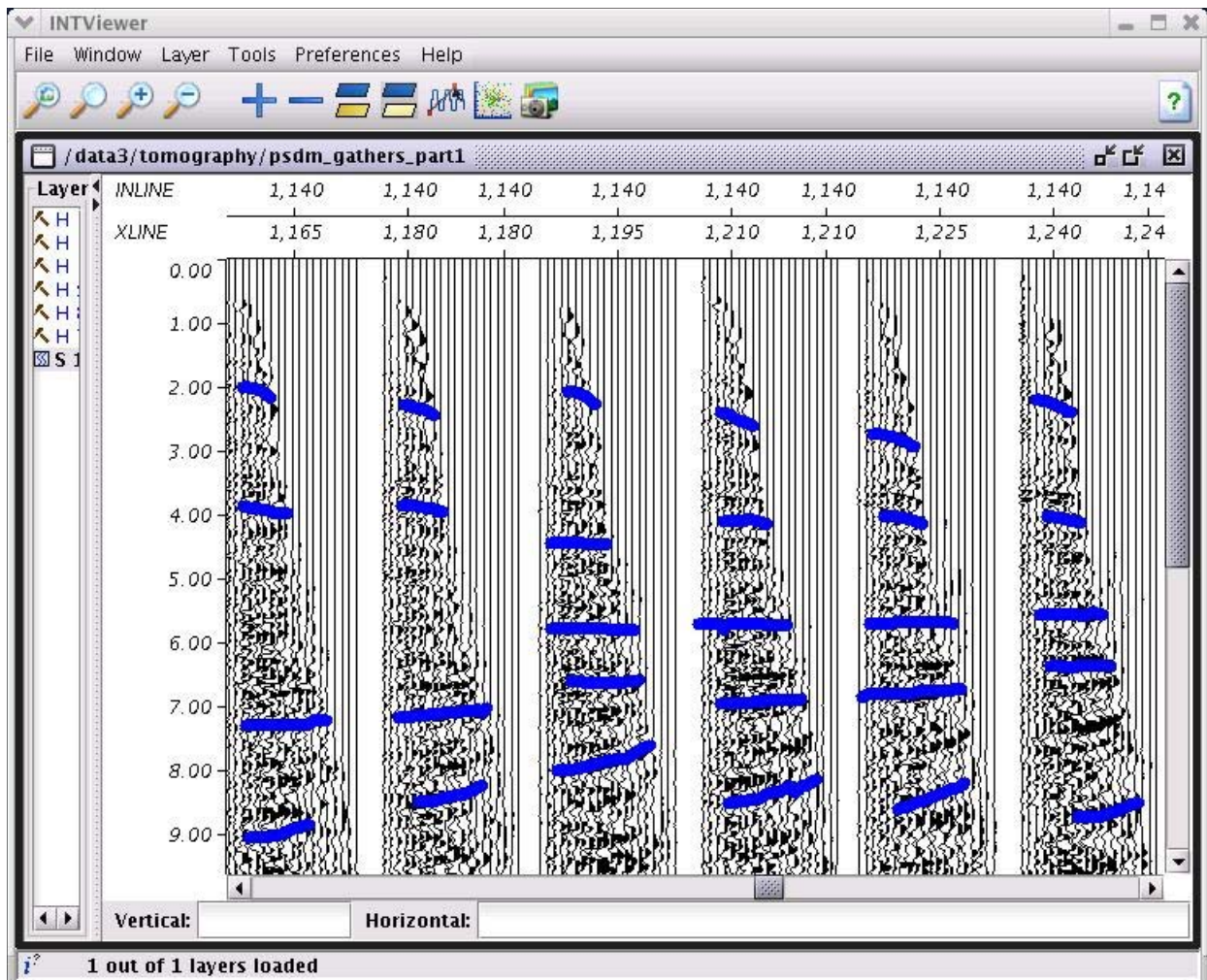
Please note that the picking is active as long as the horizon is selected in the Layers list.

Horizon Attributes

You can edit the attribute of the horizon (line style, line color, line thickness, etc.) by double clicking on the picking layer name in the Layers list.

When picking the depth migrated gathers, the following rules should be kept in mind:

- 1 Not every gather must be picked. Usually every 10-20 inline and crossline are sufficient.
- 2 Not every reflection event must be picked. Only events where the moveout changes should be picked. The tomography application will assume the same velocity correction is needed vertically between picks. See the example below.
- 3 Since the tomography is grid based, picks do not need to track horizons. No attempt is made to process the picks on a horizon basis.
- 4 For most cases only 3-5 picks are required for each picked gather. Too many picks will unnecessarily increase the tomography run time.
- 5 A stacked image of the migrated data can be used as a guide for locating the gathers within the survey. The stack and gather images can also be synchronized for scale, and display parameters.



Step 4 – Saving your work

First you can save the horizon using the Layer->Save Horizon option. Since multiple layers can be saved into a single file, you are prompted for the list of layers you want to save. Horizon data is stored in a simple XML formatted file

You can then save the entire display you just created using option File->Save Session... You will be prompted for a filename where to save the session. The session can be later restored using option File->Open Session...

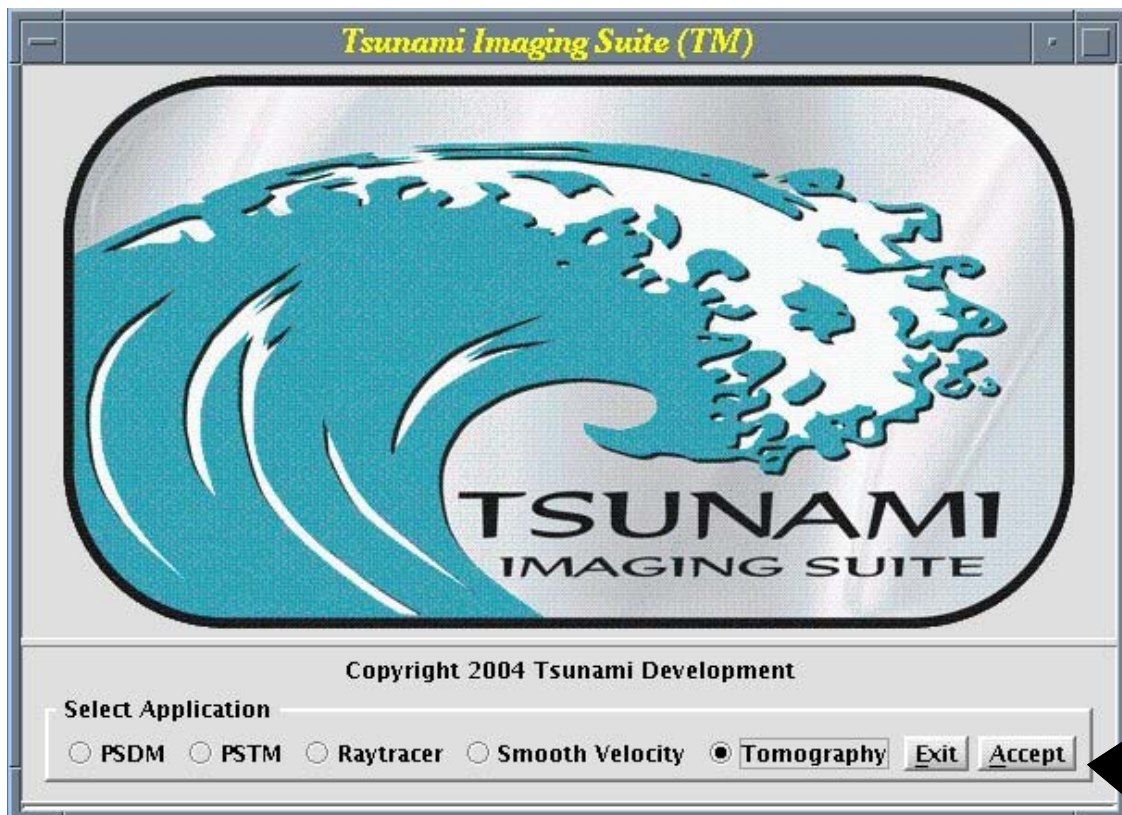
****Please Note:** INTViewer version 1.3 does not read in your saved horizon file when you re-open your session. As a best practice, you should delete all horizons when you re-open your session and then read in your saved horizon file (Layers—> Add Layers—>XML-Horizon).

Start Tomography:

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

Example: <prompt>: tsunami

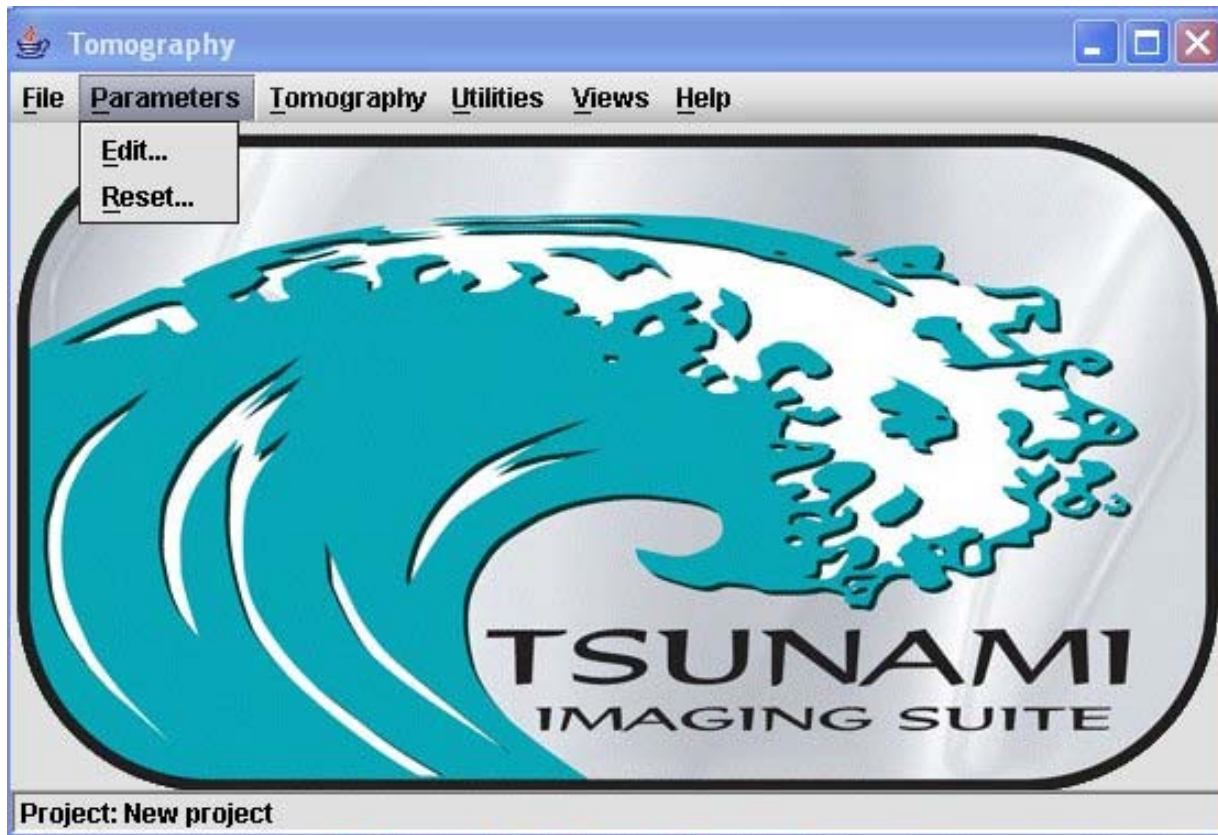
2. Toggle the Tomography button on the Startup Window and click Accept.



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

Generating an Internal Velocity File

1. Select **Parameter** —> **Edit** from the Tomography 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 TOMO that you plan to use. For version 5.2.13 it should point to ~/tsunami/rays_5.2.13.

Project Directory (prjdr): The directory where the logfile will be located as well as other temporary files created by the job.

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

Stack File Name (stkf): The name and location of the SEG Y stack file used for dip control of the tomography..

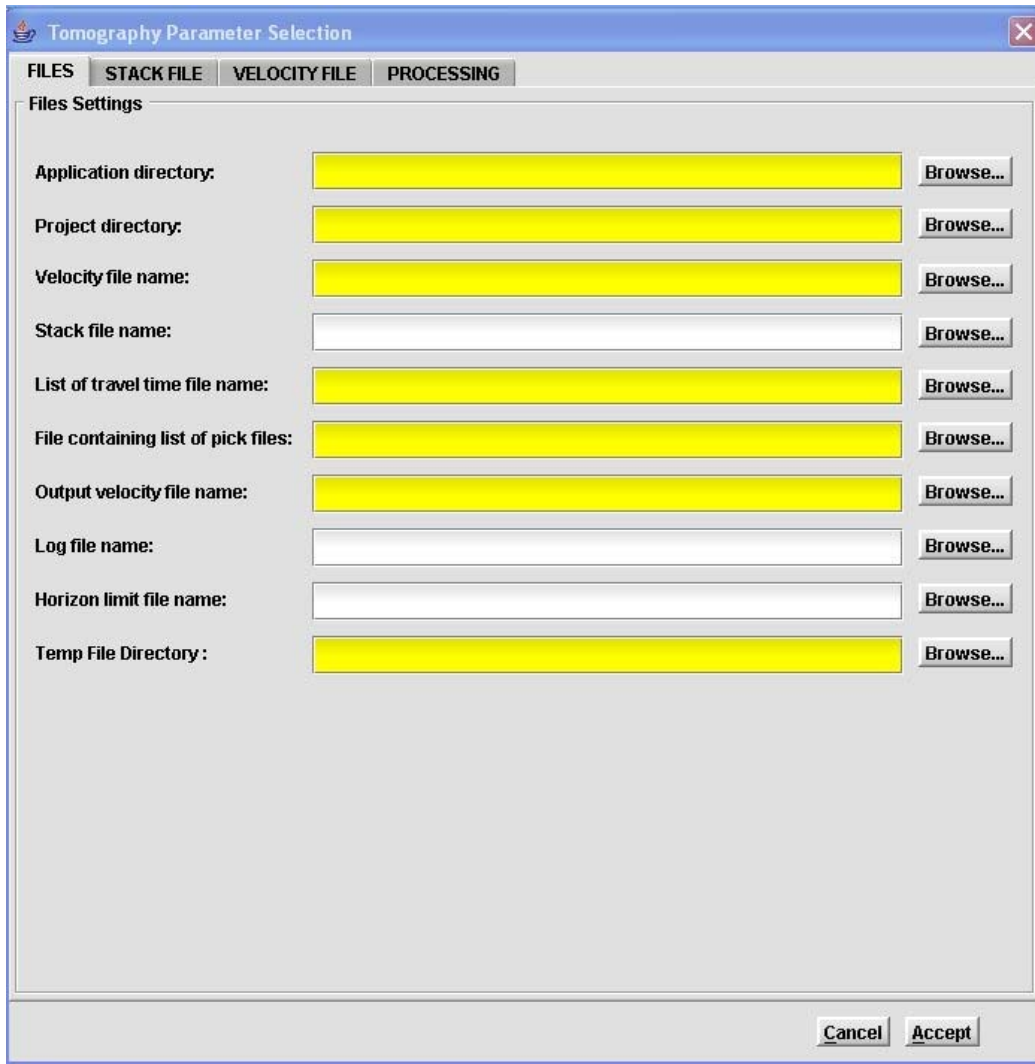
List of Travel Time File Name (ttmf): List of travel time files created with Tsunami Rays. There is no limit to the number of travel time files that can be used. Please refer to the Rays documentation for more information.

File Containing List of Pick Files (inpf): List of input pick files. The pick files are generated by picking the residual moveout on depth migrated gathers using the INT Viewer. Currently, only the INTViewer pick format is supported. See the Picking Tutorial in the INTViewer manual for more information.

Output Velocity File Name (outf): The name of the output SEG Y interval velocity file.
Log file name (logf): 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.

Horizontal Limit File Name (horizf): Velocities above this digitized horizon file will not be modified by the tomography. This allows the updated velocity model to be constructed in a top down manner. The file is created by picking a single horizontal horizon on a depth migrated section using the INTViewer. Currently, only the INTViewer format is supported.

Temp File Directory (ttmf): Directory for temporary files created during the job run.



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=	Application home directory
prjdr=	Project directory
velf=	Input velocity filename
stkf=	Stack filename
ttmf=	List of travel time files
inpf=	List of pick files
outf=	The name of the output travel time file
logf=	User specified logfile name
horizf=	Horizon limit filename
tmpdir=	Temp file directory

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

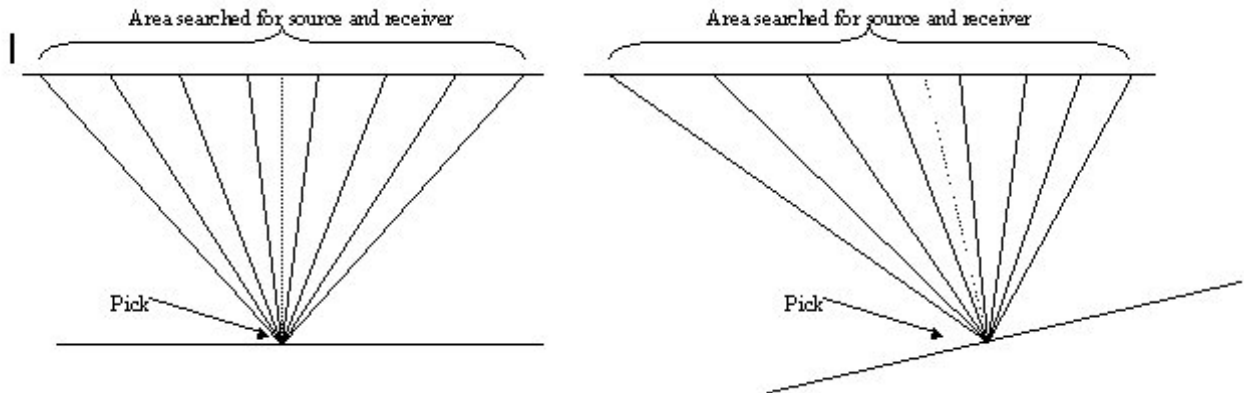
3. Select the **Stack File** tab from the **Parameter Selection Box**. Use of a stack file for the tomography calculations is optional. The stack file is analyzed to determine the dip present in the data. The dip is then used to adjust the ray paths. If no stack file is used, the dips are assumed to be flat. In this case, the rays would be symmetrical about a vertical line at the pick point. However, with a dip and azimuth the rays can be skewed to more closely represent the actual source and receiver locations that image the point.

The dip is auto calculated using a tau-p transform. It finds the maximum energy from slant stacks made with different angles both vertically and in azimuth. The slant stack with the most energy represents the maximum dip present and is the dip used to adjust the ray paths.

The stack file must be in SEGY format. The inline and xline numbers must be in the trace headers as described by the parameters below.

If used, the stack file must be identified on the **Files** tab and all parameters on the **Stack File** tab are required. If not used, all parameters on this tab can be left blank.

The left picture below shows the ray path area assuming no dip. The right side shows the shift in the ray paths when dip is included.



Byte Locations:

Byte Location of Xline (xlnb): The byte location of Xline labels in the stack file trace headers. (No Default)

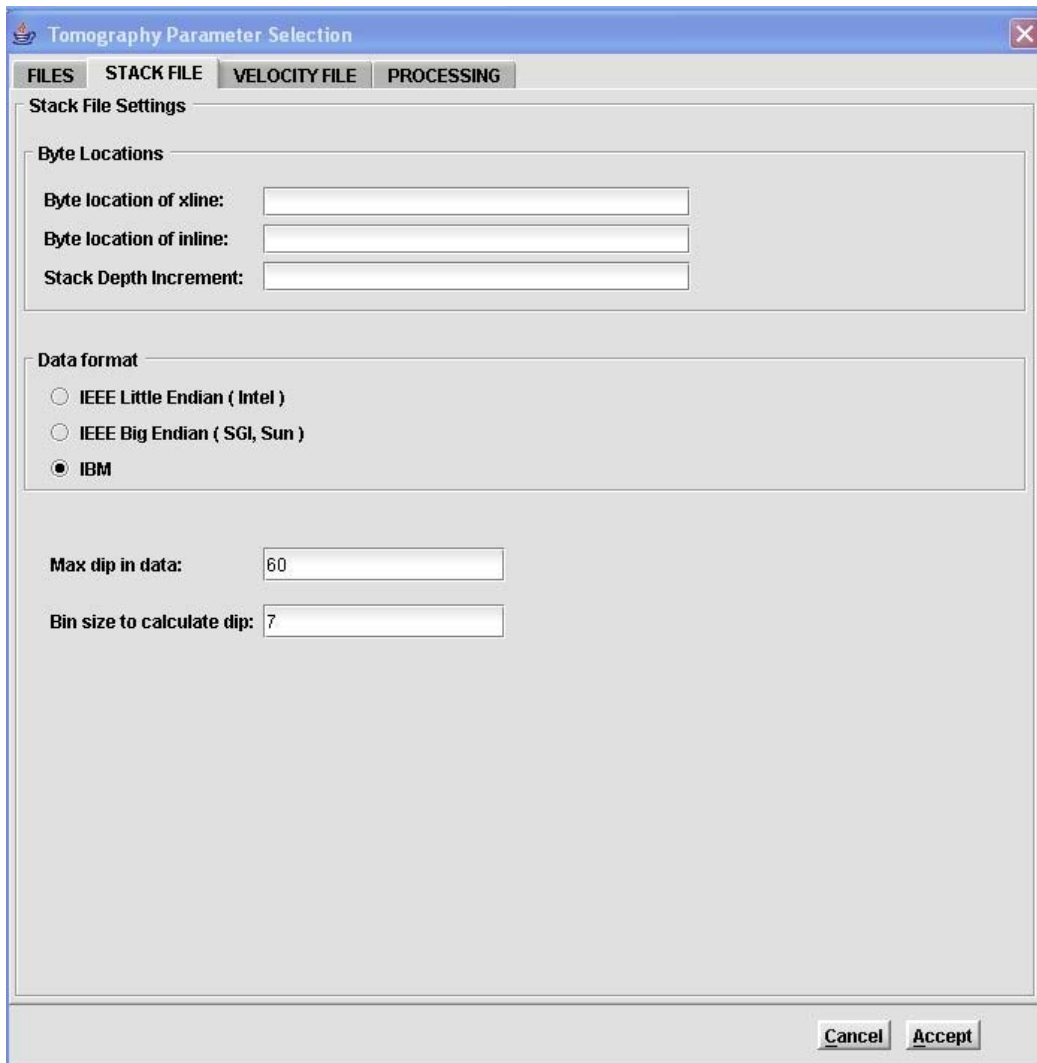
Byte Location of Inline (ilnb): The byte location of the Inline labels in the stack file trace headers. (No Default)

Stack Depth Increment (stkdzdz): Depth increment of the stack file.

Data Format (dataf): Indicates the format of the SEGY velocity model and stack file trace headers and trace data. Default is IBM.

Max Dip in Data (maxdip): Maximum dip, in degrees, for the dip correction analysis of the stack section.

Bin Size to Calculate Dip (binsz): The number of CDP's to use in the slant stack for calculating the maximum dip. Default is 5. The run time for calculating maximum dip will increase by the square of binsz.



Parameter File For Command Line Use:

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

- xlnb=** Stack file xline byte location
- ilnb=** Stack file inline byte location
- stkdz=** Depth increment of the stack file
- dataf=** Format of stack file
- maxdip=** Max dip in stack data
- binsz=** Bin size to calculate dip

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 SEG-Y format. This should be the same velocity model that generated the PSDM gathers that were used to pick for input to Tomography.

Header Location:

Header Location of Inline (vilb): The byte location of Inline labels in velocity model headers. (No Default)

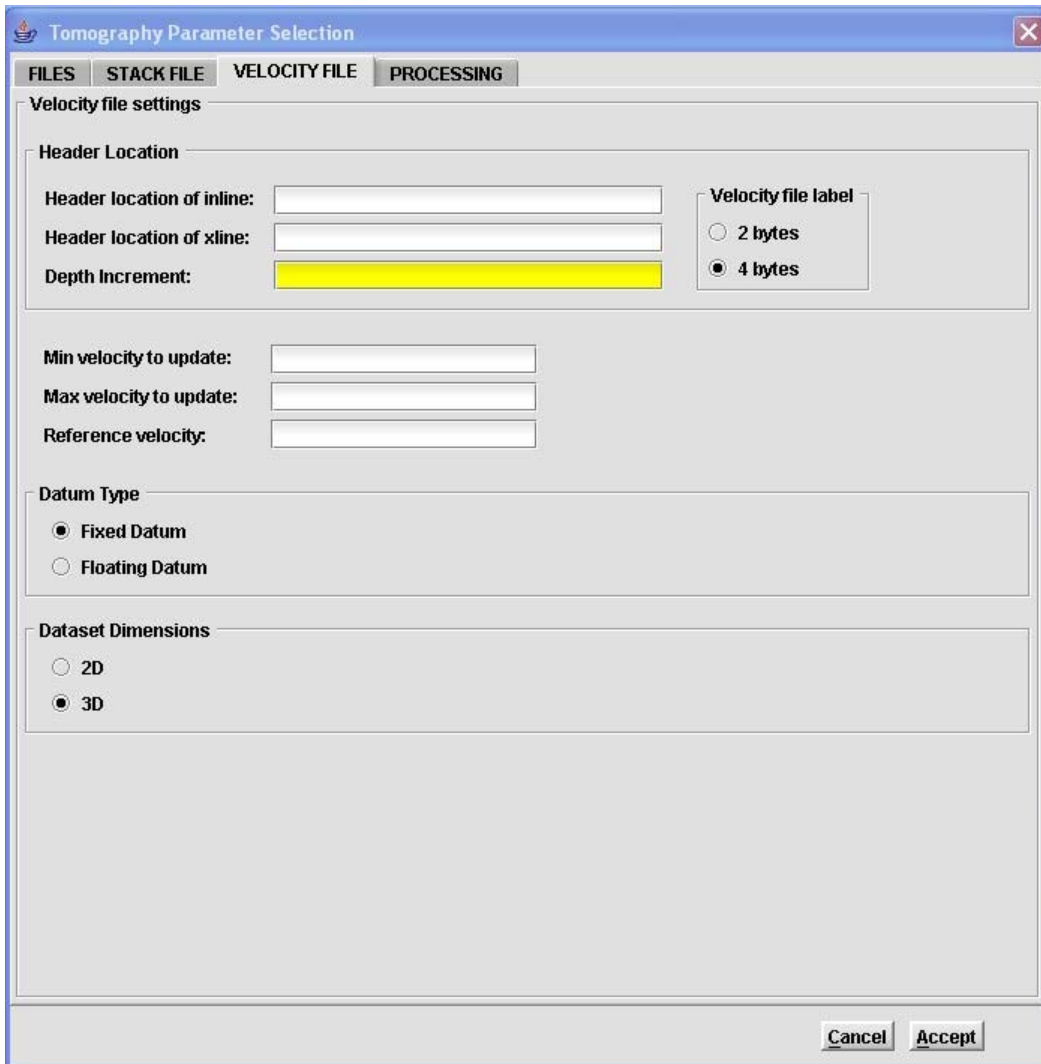
Header Location of Xline (vxlb): The byte location of the Xline labels in the model headers. (No Default)

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

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

Min Velocity to Update (minv): Model velocities slower than this value will not be modified. For example, to preserve the water velocities for a marine model enter a velocity slightly faster than the water velocity. The default is no lower limit on velocities to be updated.

Max Velocity to Update (maxv): Model velocities faster than this value will not be modified. For example, if the original model contains a salt velocity that you do not want modified by tomography, enter a velocity slightly slower than the salt. The default is no upper limit on velocities to be updated.



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:

vilb=	Header loc of inline in velocity model
vxlb=	Header loc of xline in velocity model
vdz=	Depth increment of the velocity model
vxlitype=	Velocity file label integer
minv=	Minimum velocity to update
maxv=	Maximum velocity to update
frdtm=	Fixed or floating datum
threed=	Data set dimensions

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 **Processing** tab on the **Parameter Selection Box**. Required fields will be highlighted in yellow.

Processing Settings:

Number of Rays per Pick (numpick): The number of rays calculated for each pick from the pick files. (Default is 5)

Max Percentage Velocity Change (maxchg): The maximum amount of change allowed between the input and output velocity models. (Default is 10)

Max Depth to Update (maxdepth): The maximum depth for velocity model changes. Velocities below this depth will be identical for the input and output velocity models. The default is to modify the entire depth range of the velocity model.

Migration aperture (aper): Should be the same aperture used for PSDM. (No Default)

Nominal Inline Spacing (ildist): The distance between adjacent inlines. (No Default)

Nominal Xline Spacing (xldist): The distance between adjacent xlines. (No Default)

The following two parameters are used to do smoothing of the delta V values (or change in velocity) before it is applied to the velocities. Using a larger number will produce more smoothing, a smaller number less smoothing.

Average Distance Between Picks in the Inline Direction (aveil): Default is 20

Average Distance Between Picks in the Xline Direction (avexl): Default is 20

Subsampling of Deltav calculations: Subsampling the velocity model will result in faster run times for the tomography, however, some accuracy will be sacrificed.

Xline Subsampling (subxl): Xline subsampling increment for calculating deltav. The default is no subsampling.

Inline Subsampling (subil): Inline subsampling increment for calculating deltav. The default is no subsampling.

Depth Subsampling (subz): Depth subsampling increment for calculating deltav. A value of two means that deltav will be calculated at a depth interval twice that of the input velocity model. The default is no subsampling.

Smoothing of Deltav Values: Smoothing applied to the calculated deltav values. The default is 1 or no smoothing.

Node List:

File containing node list (nddb): List of node names to use in the 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.

The image shows a screenshot of the "Tomography Parameter Selection" dialog box. The dialog has a title bar with a dropdown arrow on the left and a close button on the right. Below the title bar are four tabs: "FILES", "STACK FILE", "VELOCITY FILE", and "PROCESSING". The "PROCESSING" tab is selected. The main area is divided into three sections:

- Processing Settings:** Contains several parameters with input fields:
 - Number of rays per pick: 5
 - Max percentage velocity change: 10
 - Max depth to update: (empty)
 - Migration aperture: (yellow highlight)
 - Nominal Inline Spacing: (yellow highlight)
 - Nominal Xline Spacing: (yellow highlight)
 - Average Distance Between Picks in the Inline Direction: 20
 - Average Distance Between Picks in the Xline Direction: 20
- Subsampling of deltav calculations:** Contains three parameters with input fields:
 - Xline Subsampling: 1
 - Inline Subsampling: 1
 - Depth Subsampling: 1
- Smoothing of Deltav values:** Contains one parameter with an input field:
 - Smoothing of Deltav values: 1
- Node list:** Contains two buttons:
 - File containing node list: (yellow highlight)
 -

At the bottom right of the dialog are two buttons: "Cancel" and "Accept".

Parameter File For Command Line Use:

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

numpick=	Number of rays per pick
maxchng=	Max percentage velocity change
maxdepth=	Max depth to update
aper=	Migration aperture to use
ildist=	Distance between inlines
xldist=	Distance between xlines
Subxl=	Xline subsampling
Subil=	Inline subsampling
Subz=	Depth subsampling
smfctr=	Smoothing of deltav values
nddb=	Node database - See Cluster Configuration for a description

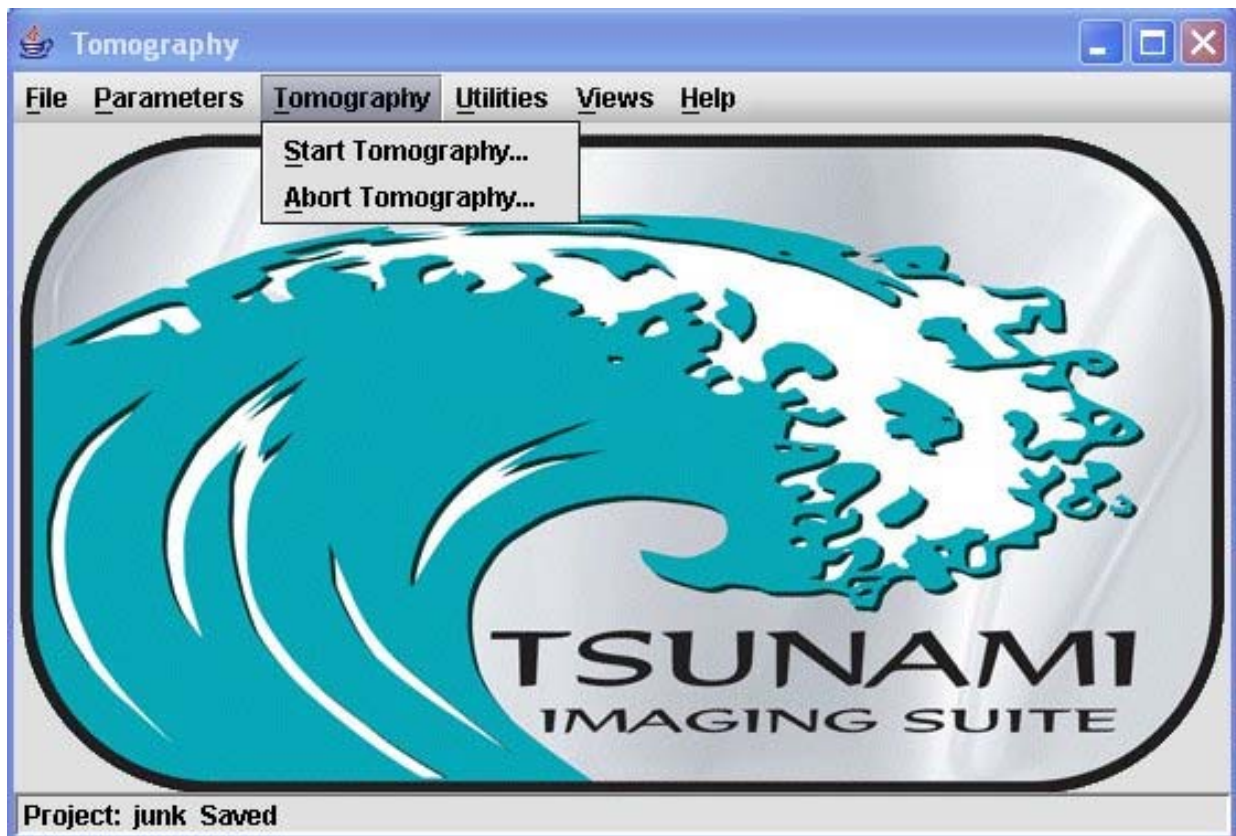
For more information on creating the command line parameter file and complete list of required parameters please see *Appendix B: Parameter File Information*.

Starting / Stopping a Job

Stop Job from GUI

Tomography - 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 Tomography 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.

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.

Appendix A: Log file Information

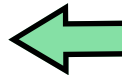
The log file is located in `~proj_dir/parameter_file.log`

Tomography provides a great deal of information in the log file. If you are running a new data-set through Tomography 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.

<pre>Reading license file /itanium1/software/tomo_license.dat Current license expires month 1 day 1 year 2006 Version 4.13 Tsunami Tomography Parameter file: tomo.vol</pre>		CHECK 1 Verify License file is correct.
<pre>hmdr= '/itanium1/software/tsunami_5.2.13/tomo_5.2.13' prjdr= '/itan21/data_itanium2/kendall_tomo' velf= '/data1/summit/grid32_Vint_ibm.segy' ttmf= '/itan21/data_itanium2/kendall_tomo/ttime_list' inpf= '/itan21/data_itanium2/kendall_tomo/pick.files' outf= '/itan21/data_itanium2/kendall_tomo/summit_mod1.segy' tmpdir= '/itan21/data_itanium2/tomo_tmp' dataf= 2 vdz= 32 vxl= 17 vilb= 9 maxv= 14000 minv= 1000 vxlityp= 0 aper= 10000 ildist= 82.5 xldist= 82.5 subxl= 4 subil= 4 subz= 4 nddb= '/itan21/data_itanium2/kendall_tomo/node15itan.db' smfctr= 2.0 logf= '/itan21/data_itanium2/kendall_tomo/tomo.log'</pre>		CHECK 2 Shows parameters you input either through GUI or in your parameter file.
<pre>Stack data format is IBM segy Velocity file format is IBM segy Reading the velocity model Checking the sort order, and the limits of model Read 10000 velocity traces Read 20000 velocity traces Read 30000 velocity traces</pre>		CHECK 3 Verify correct data formats.

Percent complete 90.0 Jul 15 14:52
 Percent complete 100.0 Jul 15 14:52
 Total raypaths indexed 258101
 Maximum depth to modify 17003
 Updating model level 640
 Number of grids held constant 0
 Apply ray paths to model grids
 Sorting 20917702 ray grid intercepts
 Beginning sort
 Sorting file using 2 partitions
 Sorting partition 1
 Sorting partition 2
 Merging partitions
 Percent complete 10.0 Jul 15 15:00
 Percent complete 20.0 Jul 15 15:01
 Percent complete 30.0 Jul 15 15:01
 Percent complete 40.0 Jul 15 15:02
 Percent complete 50.0 Jul 15 15:02
 Percent complete 60.0 Jul 15 15:03
 Percent complete 70.0 Jul 15 15:04
 Percent complete 80.0 Jul 15 15:04
 Percent complete 90.0 Jul 15 15:05
 Percent complete 100.0 Jul 15 15:05
 Sort completed
 Depth 0 Min deltav 0.955 Max deltav 1.014 ave deltav 0.979
 Depth 128 Min deltav 0.949 Max deltav 1.014 ave deltav 0.979
 Depth 256 Min deltav 0.949 Max deltav 1.014 ave deltav 0.979
 Depth 384 Min deltav 0.947 Max deltav 1.014 ave deltav 0.979
 Depth 512 Min deltav 0.946 Max deltav 1.014 ave deltav 0.980
 Updating ray paths
 Percent complete 10.0 Jul 15 15:07
 Percent complete 20.0 Jul 15 15:07
 Percent complete 30.0 Jul 15 15:08
 Percent complete 40.0 Jul 15 15:08
 Percent complete 50.0 Jul 15 15:08
 Percent complete 60.0 Jul 15 15:08
 Percent complete 70.0 Jul 15 15:09
 Percent complete 80.0 Jul 15 15:09
 Percent complete 90.0 Jul 15 15:09
 Percent complete 100.0 Jul 15 15:09
 Completed updating ray paths
 Updating model level 1024
 Number of grids held constant 0
 Apply ray paths to model grids
 Sorting 17387248 ray grid intercepts
 Beginning sort
 Sorting file in memory
 Sort completed
 Depth 640 Min deltav 0.946 Max deltav 1.014 ave deltav 0.980
 Depth 768 Min deltav 0.943 Max deltav 1.014 ave deltav 0.980
 Depth 896 Min deltav 0.939 Max deltav 1.014 ave deltav 0.980
 Updating model level 1152
 Apply ray paths to model grids
 Sorting 6508195 ray grid intercepts



CHECK 5
 Prints minimum, maximum and average deltav for each depth. Verify that values are reasonable.

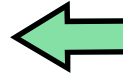
Beginning sort
Sorting file in memory
Sort completed
Depth 1024 Min deltav 0.950 Max deltav 1.014 ave deltav 0.981
Updating model level 1280
Apply ray paths to model grids
Sorting 3921073 ray grid intercepts
Beginning sort
Sorting file in memory
Sort completed
Depth 1152 Min deltav 0.947 Max deltav 1.014 ave deltav 0.981
Updating ray paths
Percent complete 10.0 Jul 15 15:21
Percent complete 20.0 Jul 15 15:21
Percent complete 30.0 Jul 15 15:21
Percent complete 40.0 Jul 15 15:22
Percent complete 50.0 Jul 15 15:22
Percent complete 60.0 Jul 15 15:22
Percent complete 70.0 Jul 15 15:22
Percent complete 80.0 Jul 15 15:23
Percent complete 90.0 Jul 15 15:23
Percent complete 100.0 Jul 15 15:23
Completed updating ray paths
Updating model level 1408
Number of grids held constant 0
Apply ray paths to model grids
Sorting 6010039 ray grid intercepts
Beginning sort
Sorting file in memory
Sort completed
Depth 1280 Min deltav 0.946 Max deltav 1.015 ave deltav 0.982
Updating model level 1536
Apply ray paths to model grids
Sorting 6860137 ray grid intercepts
Beginning sort
Sorting file in memory

Lines Deleted From Original Log File for Space Saving in Documentation

Depth 16640 Min deltav 0.924 Max deltav 1.008 ave deltav 0.986
Depth 16768 Min deltav 0.924 Max deltav 1.008 ave deltav 0.986
Updating model level 17024
Apply ray paths to model grids
Sorting 2246 ray grid intercepts
Beginning sort
Sorting file in memory
Sort completed
Depth 16896 Min deltav 0.933 Max deltav 0.958 ave deltav 0.945
Updating final velocity model

Smoothing deltav model
Smoothing operator x dimen 7
Smoothing operator y dimen 7
Smoothing operator z dimen 1
Printing error of fit for poorest 5.0 percent
Inline Xline Depth Percent Error

1170	1270	6846.7	6.68
1110	1255	2996.5	6.52



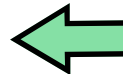
CHECK 6

Prints error of fit for the poorest 5.0 percent of picks. Verify numbers are small. Large numbers indicate poor picks.

Lines Deleted From Original Log File for Space Saving in Documentation

1320	1165	4355.4	0.50
1185	1195	2090.6	0.50
1350	1300	6533.1	0.50
1290	1120	4407.7	0.50
1350	1090	7578.4	0.50

TOMGRAPHY MAIN: Successful completion

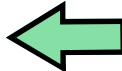


CHECK 7

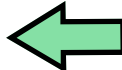
Check for successful completion of process.

Example Check License Log File:

Product name tomo		
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

- aper=** Migration aperture to use
Required
No Default
- hmdr=** Application home directory
Required
- ildist=** Distance between inlines
Required
No Default
- ilnb=** Stack file inline byte location
Required only if using stack file
No Default
- inpf=** List of pick files
Required
- nddb=** Node database - See Cluster Configuration for a description
Required
No Default
- outf=** The name of the output travel time file
Required
No Default
- prjdr=** Project directory
Required
- stkdz=** Depth increment of the stack file
Required only if using stack file
No Default
- tmpdir=**Temp file directory
Required
- ttmf=** List of travel time files
Required
- vdz=** Depth increment of the velocity model
Required
No Default

- velf=** Input velocity filename
Required
- vilb=** Header loc of inline in velocity model
Required
No Default
- vxlb=** Header loc of xline in velocity model
Required
No Default
- xldist=** Distance between xlines
Required
No Default
- xlnb=** Stack file xline byte location
Required only if using stack file
No Default

Alphabetical List of Parameters

- aper=** Migration aperture to use
Required
No Default
- binsz=** Bin size to calculate dip
Default = 7
- dataf=** Format of stack and velocity file
0= IEEE_LE
1= IEEE_BE
2 = IBM
Default = 2 or IBM
- frdtm=** Fixed or floating datum
0= Fixed datum
1= Floating datum
Default = 0 or Fixed datum
- hmdr=** Application home directory
Required
- horizf=** Horizon limit filename
- ildist=** Distance between inlines
Required
No Default
- ilnb=** Stack file inline byte location
Required only if using stack file
No Default
- inpf=** List of pick files
Required
- logf=** User specified logfile name
Default = parameter_file_name.log
- maxchnng=**Max percentage velocity change
Default = 10
- maxdip=**Max dip in stack data
Default = 60
- maxdepth=**Max depth to update
Default to update full depth
- maxv=** Maximum velocity to update
No lower velocity limit for update

minv= Minimum velocity to update
No upper velocity limit for update

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

numpick=Number of rays per pick
Default = 5

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

prjdr= Project directory
Required

smfctr=Smoothing of deltav values
Default = 1

stkf= Stack filename

stkdz= Depth increment of the stack file
Required only if using stack file
No Default

subil= Inline subsampling
Default = 1

subxl= Xline subsampling
Default = 1

subz= Depth subsampling
Default = 1

threed= Data set dimensions
0= 2D
1= 3D
Default = 1 or 3D

tmpdir= Temp file directory
Required

ttmf= List of travel time files
Required

vdz= Depth increment of the velocity model
Required
No Default

velf= Input velocity filename
Required

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

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

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

xldist= Distance between xlines
Required
No Default

xlnb= Stack file xline byte location
Required only if using stack file
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.

hmdir= '/software/tsunami_4.13.8/tomo_4.13.8'	home directory of software
prjdr= '/data/tomography'	project directory
velf= '/data/summit/Vint_mod0.segy'	input velocity model
ttime= '/data/tomography/ttime_files'	list of input travel time files
inpf= '/data/tomography/pick_files'	list of input pick time files
outf= '/data/tomography/Vint_mod1.segy'	output velocity model
tmpdir= '/data/tomography'	temporary file directory
nddb= '/data/tomography/node.db'	nodes to use for this job
logf= '/data/tomography/tomography.log'	log file for this job
dataf= 2	format of the velocity and stack files
vdz= 32	velocity model depth increment
vxlb= 17	velocity model byte offset of xline
vilb= 9	velocity model byte offset of inline
vxliltp= 0	line label size
maxv= 14000	maximum velocity to update
minv= 1000	minimum velocity to update
aper= 10000	migration aperture
ildist= 82.5	inline spacing
xldist= 82.5	xline spacing
subxl= 4	xline subsampling
subil= 4	inline subsampling
subz= 4	depth subsampling
smfctr= 2.0	smoothing of deltax values

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.