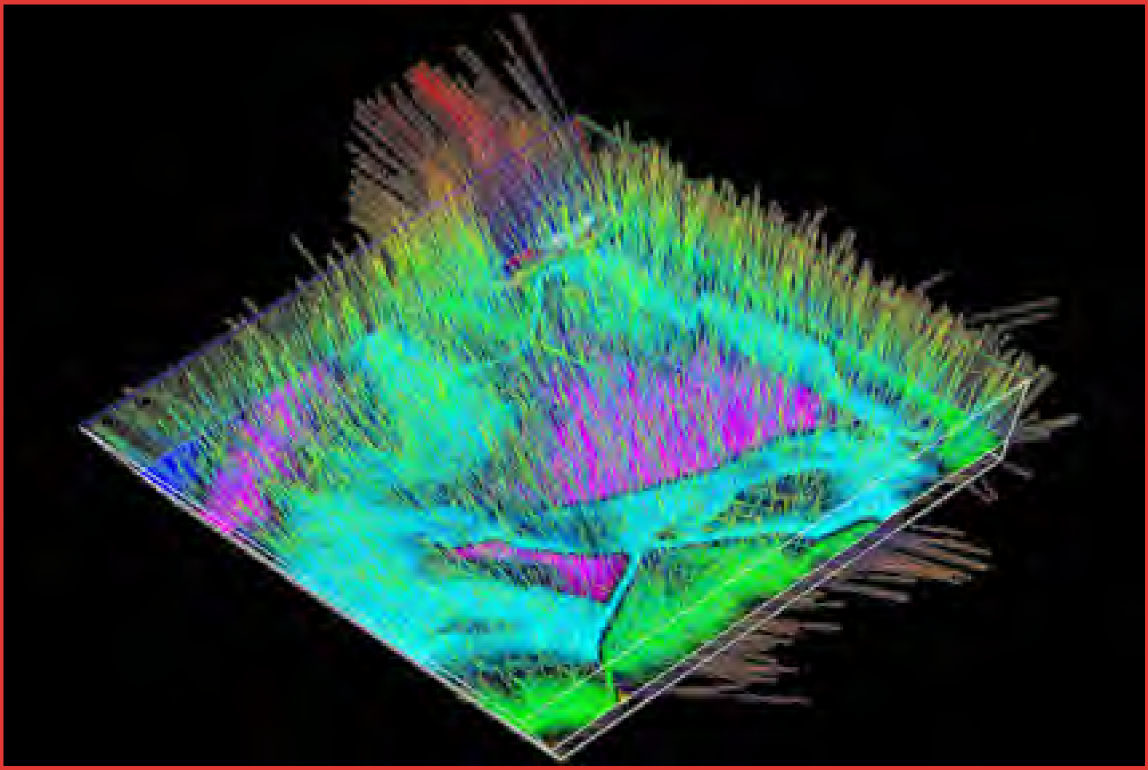
MARVEL® Users Guide

Version 2.1.1



Panorama Technologies

www.panoramatech.com

MARVEL User Guide

Contents

About This Book

This book contains information about installing and licensing the MARVEL Migration and Residual Velocity Analysis program developed by Panorama Technologies.

The book is intended for people doing seismographic data analysis, specifically migration, residual velocity analysis, and interpretation. This includes students in applied geophysics and employees of the petroleum industry, particularly those who deal with seismic migration as processors or interpreters, as well as anyone who is interested in the methodology of subsurface imaging using reflection seismology.

The book assumes that you are familiar with the following topics:

- using the operating systems and applications in your enterprise
- editing and saving text files
- performing seismographic data analysis

Online books are formatted as Portable Document Format (PDF) documents. To view, print, or copy PDF books, use the free Acrobat Reader from Adobe Systems. If you do not have the reader on your system, you can obtain the reader at http://www.adobe.com.

Conventions

This book uses the following special conventions:

- All syntax, operating system terms, and literal examples are presented in this typeface.
- File names and paths are displayed as path/filename.
- Variable text in path names, system messages, or syntax is displayed in italic text: testsys/instance/*fileName*.
- The symbol => connects items in a menu sequence. For example, Actions => Create Test instructs you to choose the Create Test command from the Actions menu.
- If you move the cursor over text shown in blue, such as Chapter 1, the cursor will change to a pointer. If you then click the cursor, Acrobat Reader displays the location indicated by the text.
- Web and email addresses are shown as underlined blue text, such as www.panoramatech.com, and will open your default web browser when you click the mouse cursor on the link.

Chapter 1

Installing and Licensing MARVEL

This chapter explains the process of installing and licensing the MARVEL. To do this, you must install MARVEL on each computer on which you want to run the program, and then run the Panorama License Manager on the computer which you want to use as the Panorama License Server.

Installing MARVEL

MARVEL is supplied as a single tar file.

1. Change to the directory in which you want to install MARVEL. For example,

   #cd /usr/jpm/apps

2. Mount the MARVEL software distribution CDROM on your system. For example,

   #mount t iso9660 /dev/cdrom /mnt/cdrom

3. Verify the name of the MARVEL installation tar file by entering the command:

   #ls /mnt/cdrom

   The MARVEL tar file will have a name similar to

   marvel-i86-AMD-7.8.0-2007.08.15.tar.

4. Copy the MARVEL tar file from the CDROM to the directory in which you want to install MARVEL. For example,

   #cp marvel-i86-AMD-7.8.0-2007.08.15.tar /usr/jpm/apps

Note:

If you plan to you MARVEL on a cluster, you must install MARVEL into exactly the same directory structure on each node in the cluster.

5. Decompress the MARVEL files using the tar command. For example,

   #tar zxvf marvel-i86-AMD-7.8.0-2007.08.15.tar

   This will create a marvel subdirectory and expand the tar file into that subdirectory.

6. Change to the subdirectory in which you have installed MARVEL. For example,

   #cd marvel

7. Use a text editor, such as vim, to open the run script and change the INSTALL_TOP line to point to the directory where MARVEL is located. For example,

   INSTALL_TOP=/usr/jpm/marvel/

Note:

The final / is required.

8. Use a text editor, such as vim, to open the config.sh script and change the PANLM_SERVER line to point to the local host where the Panorama License Server will be located, and to the port to be used for communications. For example,

   PANLM_SERVER=hpotter:33933

9. Panorama Technologies suggests that you also add the MARVEL directory to the PATH statement on each computer where you install MARVEL.

10. Repeat Steps 1 through 9 on each computer on which you want to install MARVEL.

Obtaining the License File

To use the Panorama License Manager, you must first decide on the computer you want to use as the Panorama License Server. The Panorama License Manager is installed automatically as part of the MARVEL installation procedure, but you must install a license file on the computer you want to use as the Panorama License Server. The Panorama License Manager is then run as a daemon on the computer on which the license file has been installed.

> **Note:**
>
> These steps must be performed on the computer you want to use as the Panorama License Server.

To obtain your license file, perform the following steps:

1. Change to the directory in which you have installed MARVEL.

2. Enter the following command:

   ```
   # ./run panlm &
   ```

   Several lines of text are displayed in the window, similar to the following:

   ```
   Aug 11 18:20:43 schmendrake.local panlm[379]: WARNING: License file □/Applications/marvel/install/config/panlm.conf□ does not exist
   Aug 11 18:20:43 schmendrake.local panlm[379]: Panorama Licence Manager version 0.04
   Aug 11 18:20:43 schmendrake.local panlm[379]: Listening to port 33933
   Aug 11 18:20:43 schmendrake.local panlm[379]: Host ID □5910d332e3ca1da8b927a403bf3ab8a5□
   Aug 11 18:20:43 schmendrake.local panlm[379]: WARNING: Couldn□t read config file □/Applications/marvel/install/config/panlm.conf□
   Aug 11 18:20:43 schmendrake.local panlm[379]: Status:
   Aug 11 18:20:43 schmendrake.local panlm[379]: Max Queued: 0
   ```

3. Copy this text into an email message, and send the message to: cjb@panoramatech.com.

   Panorama Technologies will send you a license file specific to the computer where you ran the panlm command.

> **Note:**
>
> Please include your telephone number in the body of the message in case Panorama Technologies needs to provide any special installation information.

4. Save the text file attachment, panlm.conf, you receive from Panorama Technologies.

5. Copy the panlm.conf file to the MARVEL installation configuration directory. That is, if you have installed MARVEL in /usr/jpm/apps/marvel, copy the file to /usr/jpm/apps/marvel/install/config.

Starting the License Manager

To start the Panorama License Manager daemon, enter the following command:

```
#run panlm >& panlm.log &
```

Run the nodeInfo Daemon

Although most of the work will be done on the computer on which you run MARVEL itself, computationally heavy tasks will be spread automatically over multiple computers in a cluster. You must run the nodeInfo daemon on each computer you want to use in the cluster. To start the nodeInfo daemon, enter the following command on every computer you want to use in the cluster:

```
#run nodeInfo daemon=1 >& /dev/null &
```

Chapter 2

Building and Refining a Model

This chapter guides you through picking data in MARVEL, building a basic model based on those picks, running a job against the seismic data using that model, and then refining the picks to build a better model in an iterative process.

Building a Model

The general procedure for building a model follows these nine steps:

1. Create a project to specify the basic information you want to use.
2. Pick a limited set of data to start the process.
3. Build a model of the data using the MARVEL Build Model window (see Building a Model), or using the MARVEL jobBuilder (see To Build a Model as a Job).
4. Since the first model is based on a very limited set of the ensembles, pick all of the ensembles you want to examine.
5. Build a new model of the data based on the results of the full pick.
6. Run a Kirchhoff time migration job against the data using the new model.
7. Run a normal moveout (NMO) job against the output from the Kirchhoff time migration.
8. Pick the data generated by the moveout, and build a new model using these picks.
9. If the resulting inline or crossline data looks reasonable, stop the process; otherwise, return to Step 3 to go through another iteration of the process, using the data from Step 4 to begin the new iteration.

Step 1. Create a Project

You define a project to specify the data you want to analyze, and to provide certain base information to be used in the calculations.

Before You Begin

As you create a project, you will need the following information:

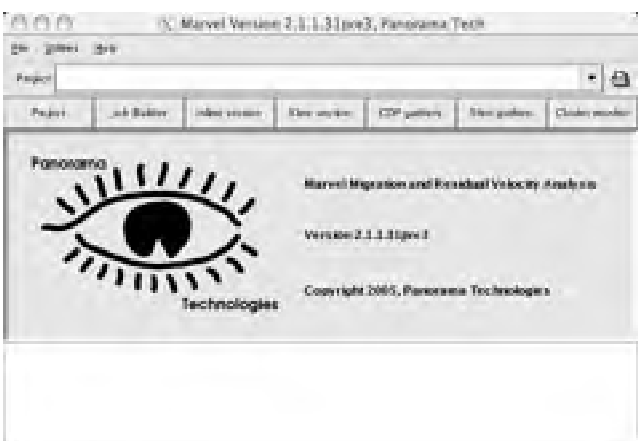
- datafile or dataset names
- basic information about the data: CDP 0, line 0, x0, y0, dx, dy, and angle

To Create a Project

1. Start MARVEL, as explained in Chapter 1.

   MARVEL displays the Start Window shown in Figure 1.

Figure 1. Start Window



2. Click the Project button in the Startup panel.

   MARVEL displays the Project window shown in Figure 2. This window enables you to select input datasets, a model (if you already have one defined), an output data set, and basic information about the coordinate system you want to use.

Figure 2. Project Window

3. Select File => Open...

   MARVEL displays a standard file Open window. Use the window to select the .segy files you want to use for this project. You can select multiple files by holding the Shift key as you click the mouse cursor on each file you want to select.

Note:

You can also create an XML dataset that identifies all of the data files that you want to use in the project and information about headers in the data files. Refer to Appendix A, File Formats, in the *MARVEL Reference Manual*.

4. If you have an existing model of the data, click on the Model tab.

   MARVEL displays the Model selection panel of the Project window, as shown in Figure 3.

Figure 3. Project Window with Model Pane

5. Select File => Open Model...

   MARVEL displays a standard file Open window. Use the window to select the .model file you want to use for this project.

6. Select File => Save As...

   MARVEL displays a standard Save As window. Enter the project name you want to use, and click Save.

Step 2. Picking the Data

Step 2a. Autopicking the Data

Velocity picking involves identifying and recording the position of specific reflection events in a seismic image. It involves placing a segmented line through a series of high energy locations in a semblance. You perform picking by selecting a limited number of reflectors for picking, and you then attempt to pick those reflectors everywhere they are present throughout the seismic image. You can use autopicking, select the picks manually, or use a combination of the two methods. This user's guide does not attempt to describe manual picking. If you are using manual picking, pick the data, and then go to Section .

Note:

For information about how MARVEL performs autopicking, see the autopick Module in the *MARVEL Reference Manual*.

Before You Begin

Before you start create and save picks, you will need the following information:

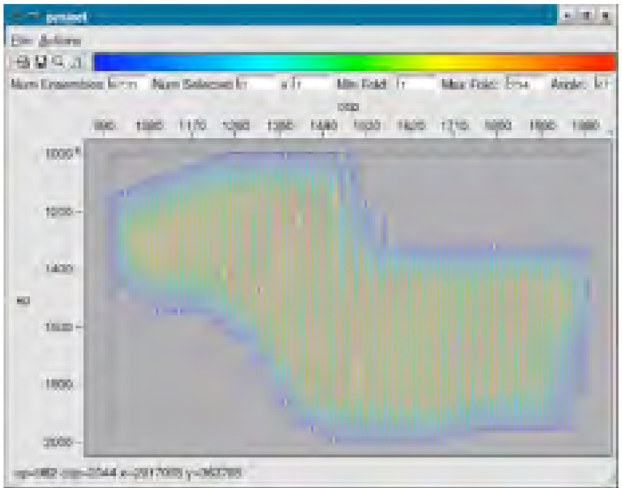
- whether you want to work with the CDP or shot map
- the number of ensembles you want to use
- the minimum and maximum velocities
- the number of velocities in the semblance
- minimum and maximum picking times
- the time increment
- the velocities at the top and bottom of the selected area and the corresponding velocity widths

To Create and Save Picks

1. Select Actions => View cdp map in the Project window (Figure 2).

   MARVEL displays a CDP map of the data, such as that shown in Figure 4.

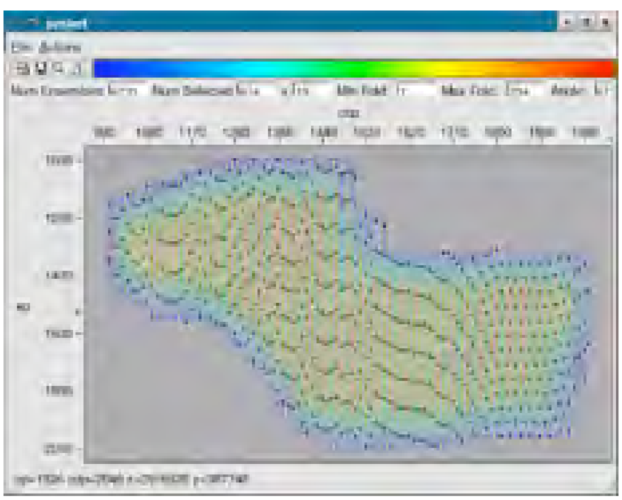
Figure 4. CDP Map



2. Select the area you want to view by the following method:

   a) Type the selection frequency, that is, do you want to select every ensemble, or only every twentieth ensemble, or some other frequency. Figure 5 shows that a selection set to every tenth ensemble.

Figure 5. CDP Map with Selections



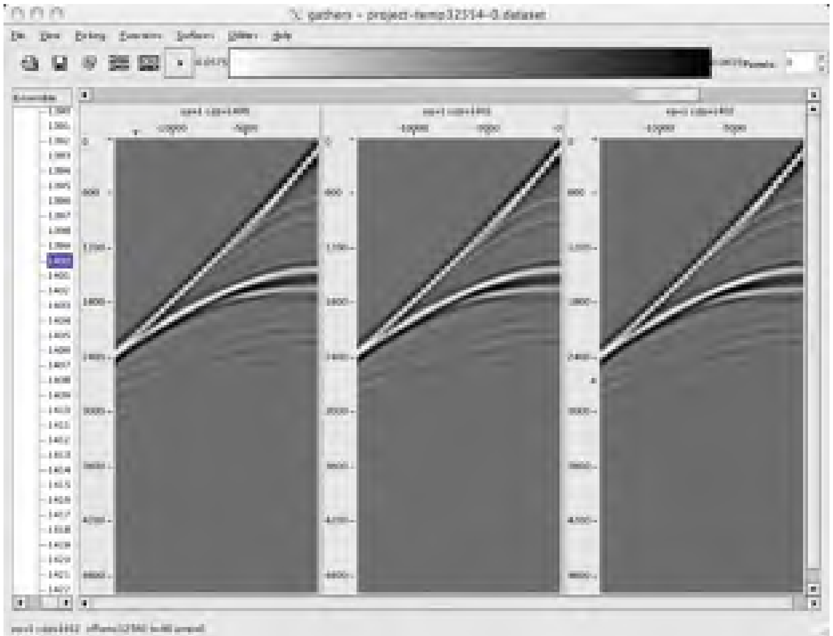
   b) Press Enter.

   c) Click the mouse in the upper left corner of the area you want to select, drag the cursor to the lower right corner, and then release the mouse button.

3. Select Actions => View selection, or click on the magnifying glass in the toolbar.

   After a few moments, MARVEL displays a Gathers window, as shown in Figure 6.

Figure 6. Gathers Window

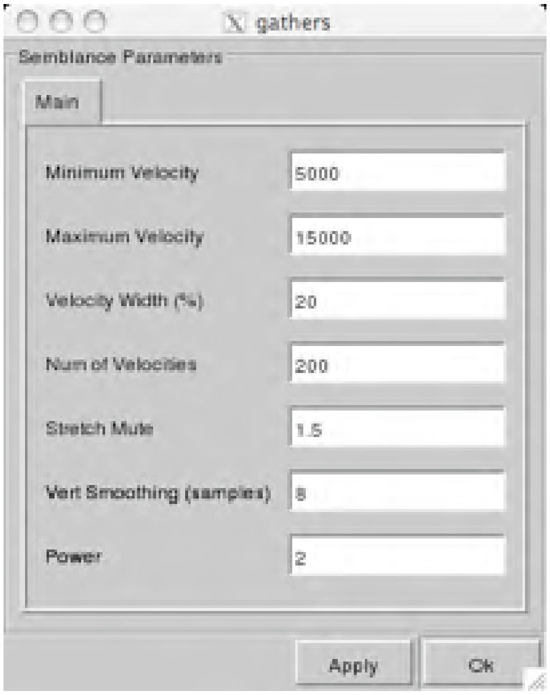


You can change the number of ensembles displayed at one time by changing the number in the Panels field on the right side of the toolbar, and then pressing Enter.

4. Select Functions => Semblance parameters... in the Gathers window.

   MARVEL displays the Semblance Parameters window shown in Figure 7.

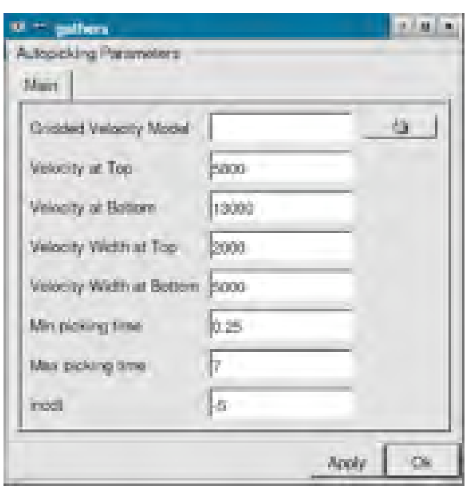
Figure 7. Semblance Parameters Window



5. Modify the displayed parameters to suit your project.
6. Click Ok in the Semblance Parameters window to accept the entries and close the window.
7. Select Picking => Autopicking parameters... in the Gathers window.

   MARVEL displays the Autopicking Parameters window shown in Figure 8.

Figure 8. Autopicking Parameters Window

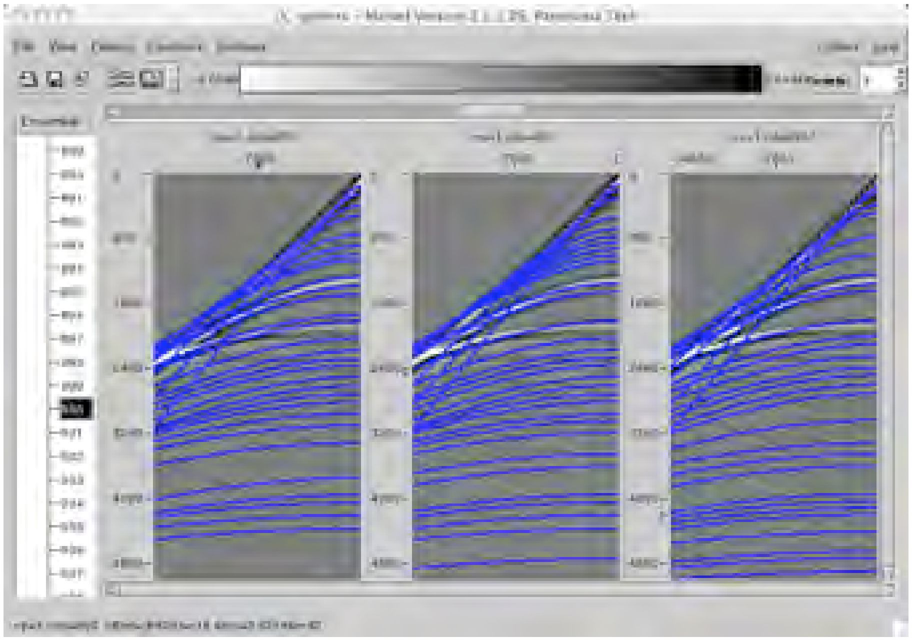


8. Enter the desired values for the autopicking parameters.

9. Click OK in the Autopicking Parameters window to accept the entries.
10. Select Functions => Autopick in the Gathers window.

    MARVEL draws the autopick curves, as shown in Figure 9.

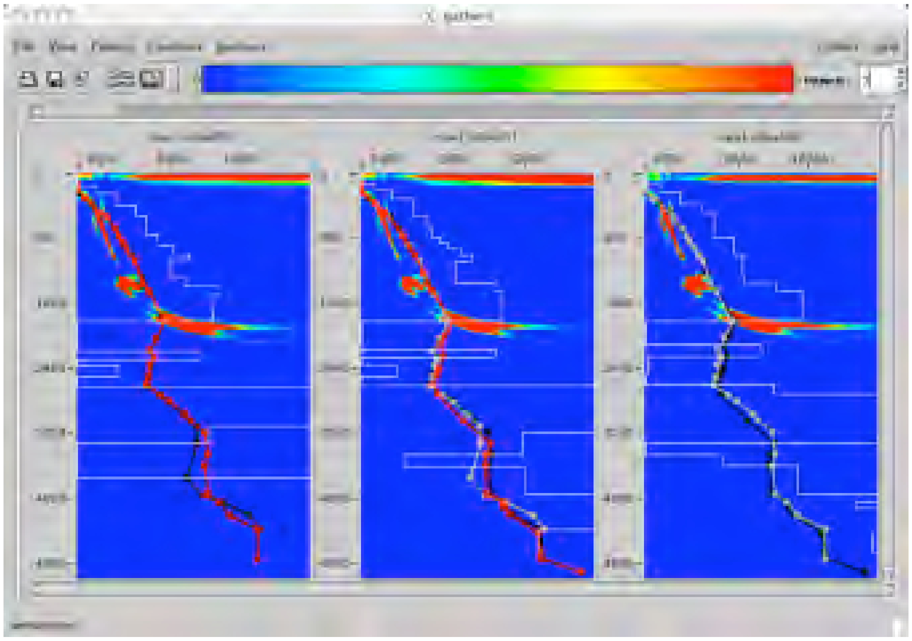
Figure 9. Autopick Curves



11. Select Functions => Semblance in the Gathers window.

    MARVEL displays a Semblance window like that shown in Figure 10.

Figure 10. Semblance Window

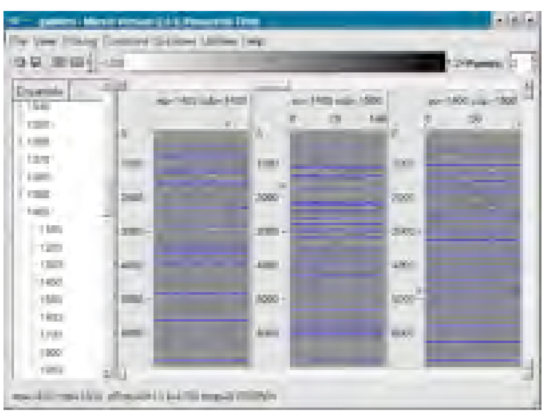


12. Select Functions => NMO to perform a normal moveout of the picks.

13. Switch to the display of the autopicks.

    If the picks have been selected correctly, they should be approximately horizontal after normal moveout, as shown in Figure 11.

Figure 11. Autopicks after NMO



14. Save the picks.

Step 2b. Picking Data Manually

Step 3. Build an Initial Model

You need to build a model before you can create and run a job analyzing the data. You can build the model using the MARVEL Build Model window (shown in this section), or you can create the model using the MARVEL jobBuilder (see To Build a Model as a Job). Although both methods are shown in this manual, you will probably find it much easier to use jobBuilder because you can rerun the job easily without reentering all of the required data.

Before You Begin

Before you start building a model, you will need the following information:

- output line range
- output crossline range
- output Z/T range
- minimum Y (line) smoothing
- minimum X (CDP) smoothing

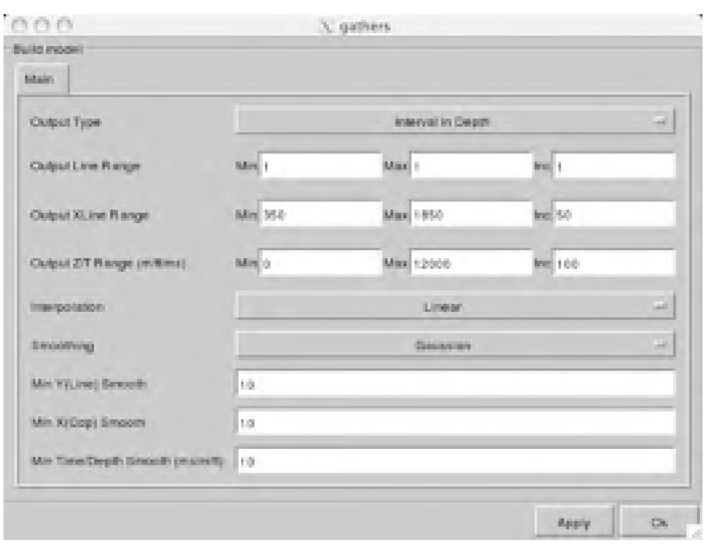
To Build a Model Using the Build Model Screen

Perform the following steps,

1. On Figure 11, select Functions => Build model...

   MARVEL displays the Build model window shown in Figure 12.

Figure 12. Build Model Window



2. Select an Output Type of RMS in Time.

3. Enter values for the Output Line Range, Output XLine Range, and the Output Z/T Range. Sample values are shown in Figure 12.

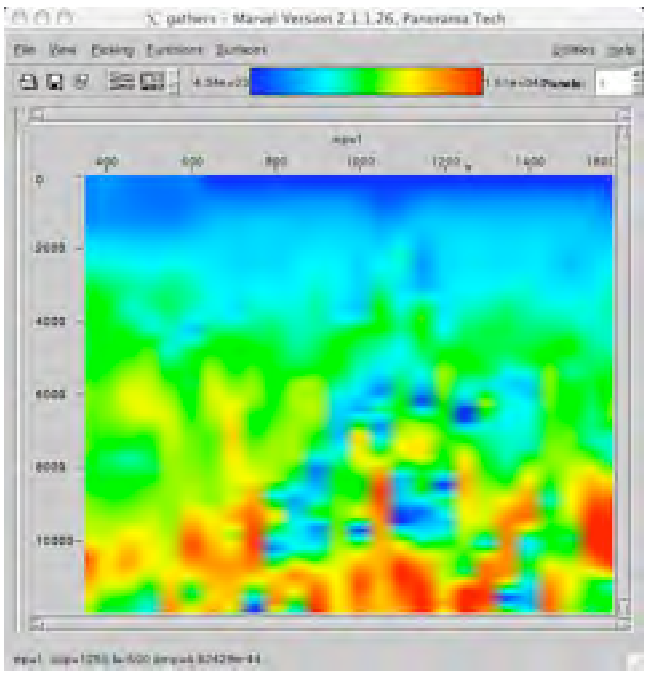
4. Click OK, to accept the default values for the other fields.

   MARVEL displays a Save As window.

5. Enter a file name for the model and click Save.

   The data is saved with an extension of .segy, and after a few moments, MARVEL displays the model, such as that shown in Figure 13.

Figure 13. Model Window



To Build a Model as a Job

You can also build a model using the MARVEL jobBuilder process. This has an advantage because you can rerun the job easily without reentering the parameters that control the process.

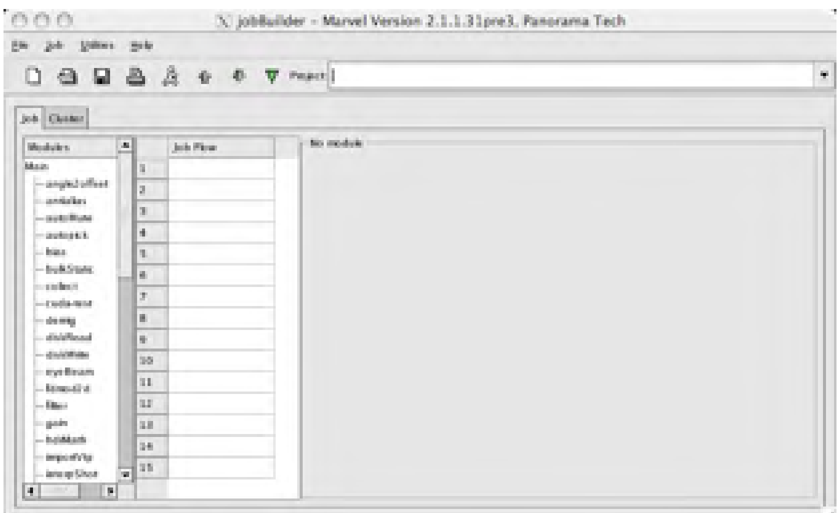
Note:

You need to make sure that the directory to which you want the job to write output has the correct read and write permissions.

1. In the Project window (Figure 2), select Actions => Run job builder...

   MARVEL displays a blank jobBuilder window as shown in Figure 14.

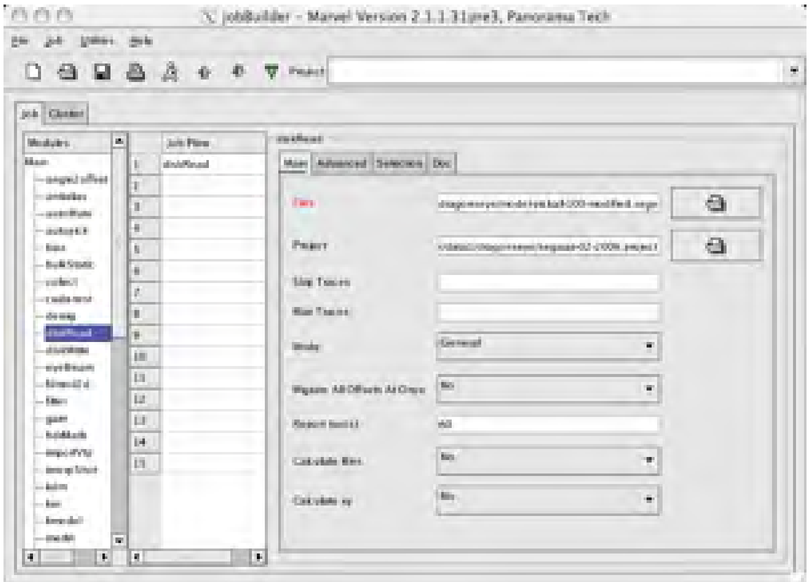
Figure 14. JobBuilder Window



2. Double-click on diskRead in the Modules tree on the left side of the window.

   MARVEL adds diskRead to the Job Flow, and displays the diskRead pane in the jobBuilder window, as shown in Figure 16. Notice that the default project information in the Project field is the current project name.

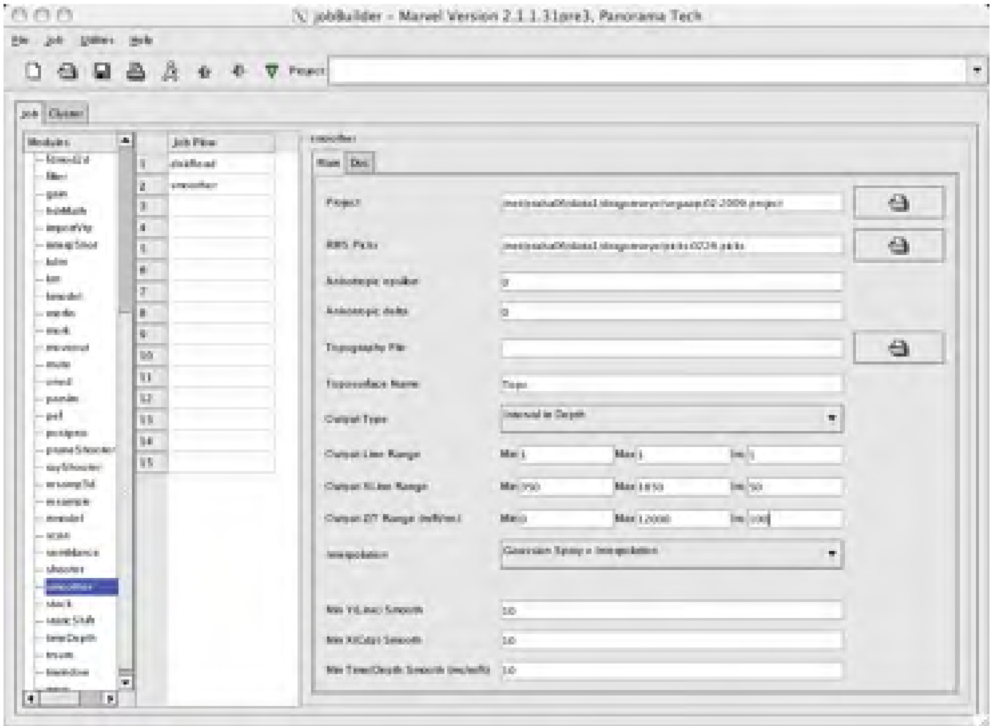
Figure 15. JobBuilder Window with diskRead Pane



3. Fill in the information in the fields as appropriate for your project.
4. Double-click on smoother in the Modules tree on the left side of the window.

   MARVEL adds smoother to the Job Flow, and displays the smoother pane in the jobBuilder window, as shown in Figure 16. Notice that the default project information in the Project field is the current project name.

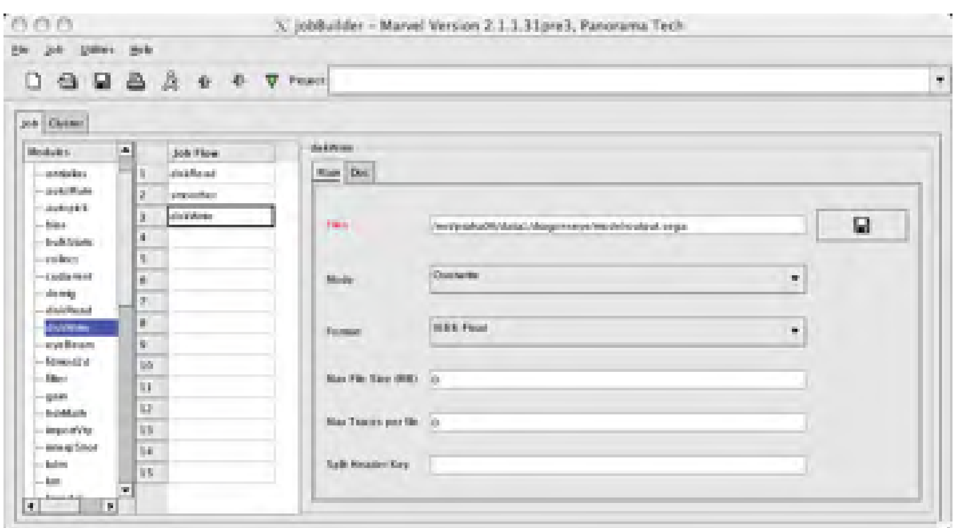
Figure 16. JobBuilder Window with smoother Pane



5. Fill in the information in the fields as appropriate for your project.
6. Double-click diskWrite in the Modules tree.

   MARVEL adds diskWrite to the Job Flow and displays the diskWrite pane in the job-Builder window, as shown in Figure 17.

Figure 17. JobBuilder diskwrite Pane



7. Enter the file name for the dataset to which you want to write the output data.

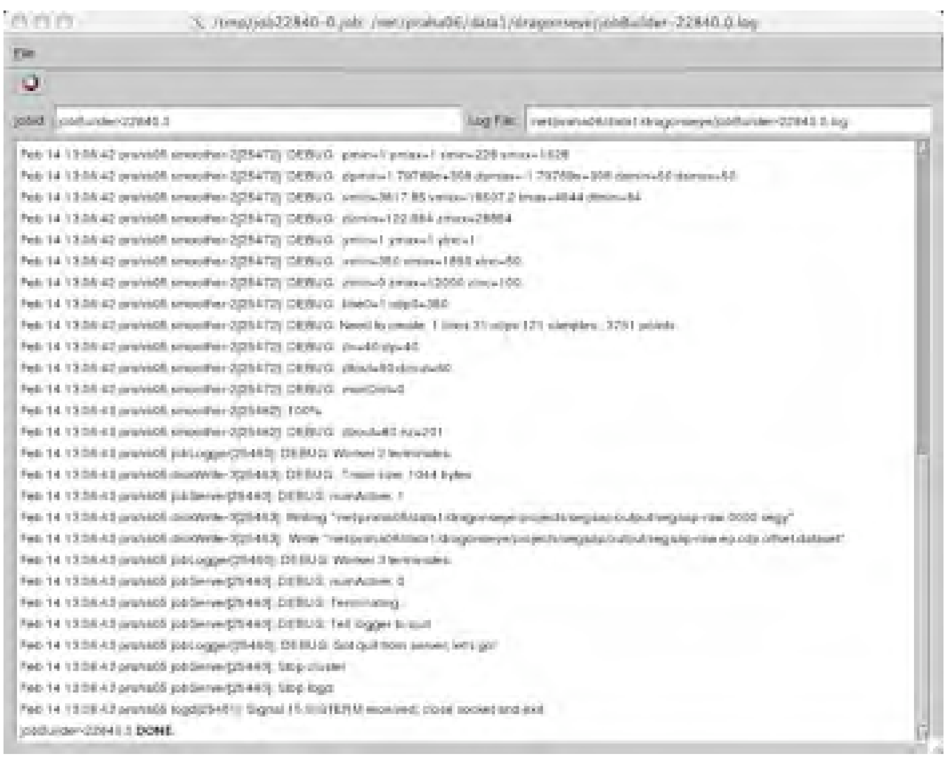
   The extension .0000.segy is added automatically to the file name. The maximum file size is determined by the Max File Size and Max Traces per File parameters. If more than one dataset is required to hold all of the data, the extension will be incremented as needed, such as .0001.segy and .0002.segy.

8. Select File => Save as... to save the job with an extension of .job.

9. To run the job, select the Job => Run menu item, or press Ctrl+R, or press the Run icon in the toolbar.

   MARVEL displays a Job Log window, as shown in Figure 18. The figure shows an example Job Log window as it looks when the job has run to completion.

Figure 18. JobBuilder Log Window



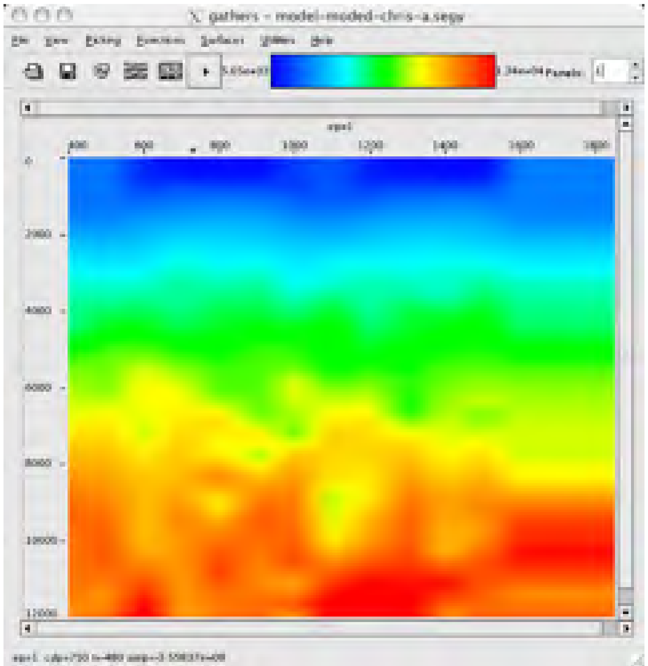
10. To display the model you have just built, click the Inline section button in the MARVEL Start Window (Figure 1).

    MARVEL displays a blank gathers window.

11. Select File => Open or click the Open button.

    MARVEL displays an Open dialog box that enables you to select the output file you specified in Step 7. MARVEL displays the inline data, as shown in Figure 19.

Figure 19. Inline View of the Model



12. To zoom the view, click the middle mouse button in the upper left corner of the zoom area, drag the cursor to the lower right corner of the zoom area and release the mouse button.

Step 4. Run a Kirchhoff Time Migration Job

The next step is migrating the data using the model you built in the previous procedure.

Note:

For more information about the specific tasks that you can perform using the MARVEL JobBuilder, refer to Chapter 4, JobBuilder Operations, in the *MARVEL Reference Manual*.

The job in this example does the following three tasks:

- read the data files
- perform a Kirchhoff time migration
- write the new information to disk

Before You Begin

You need to make sure that the directory to which you want the job to write output has the correct read and write permissions.

Before you start building the job, you will need the following information:

- name and location of the data file containing the seismic information
- project coordinates (Cdp 0, Line 0, x0, y0, dx, dy, angle)
- output mode for the Kirchhoff migration
- the offset bins
- the output lines and crosslines
- the line and crossline apertures
- the model and velocity types
- the model name
- output lines, crosslines, and times

Note:

If there is a file with extension .restart in the directory where the processing job locates the backup file (see Figure 12), your job will not run. You must delete the file manually before you can run the job.

To Build a Kirchhoff Time Migration Job

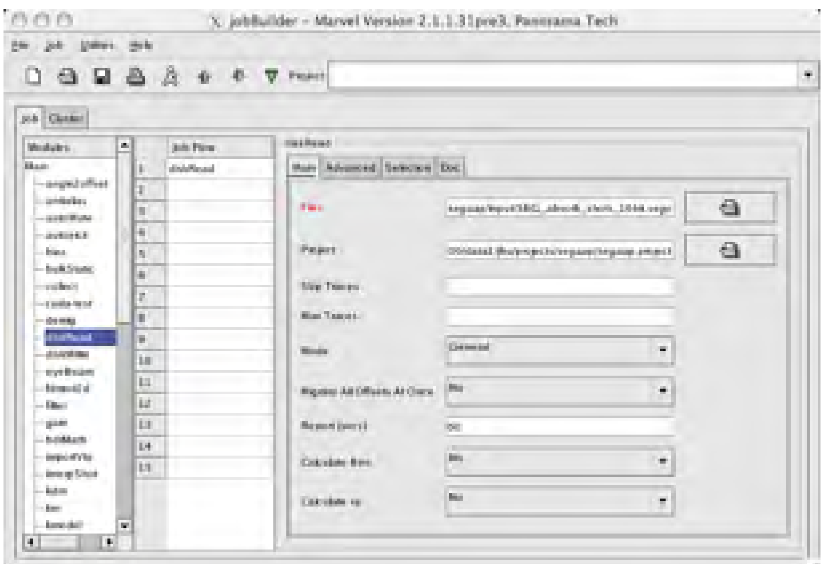
1. In the Project window (Figure 2), select Actions => Run job builder...

   MARVEL displays the jobBuilder window shown in Figure 14.

2. Double-click on diskRead in the Modules tree on the left side of the window.

   MARVEL adds diskRead to the Job Flow, and displays the diskRead pane in the jobBuilder window, as shown in Figure 20. Notice that the default project information in the Project field is the current project name.

Figure 20. JobBuilder diskread Window



Note:

Field names in red, such as Files in Figure 20, indicate that the fields are required entries. Also, if a tab name is underlined, such as the Main tab in the figure, at least one of the fields in the tab is a required entry.

3. Select the File Open button to the right of the Files field.

   MARVEL displays an Open window that enables you to select the dataset you saved when you created the project (Step 1. Create a Project).

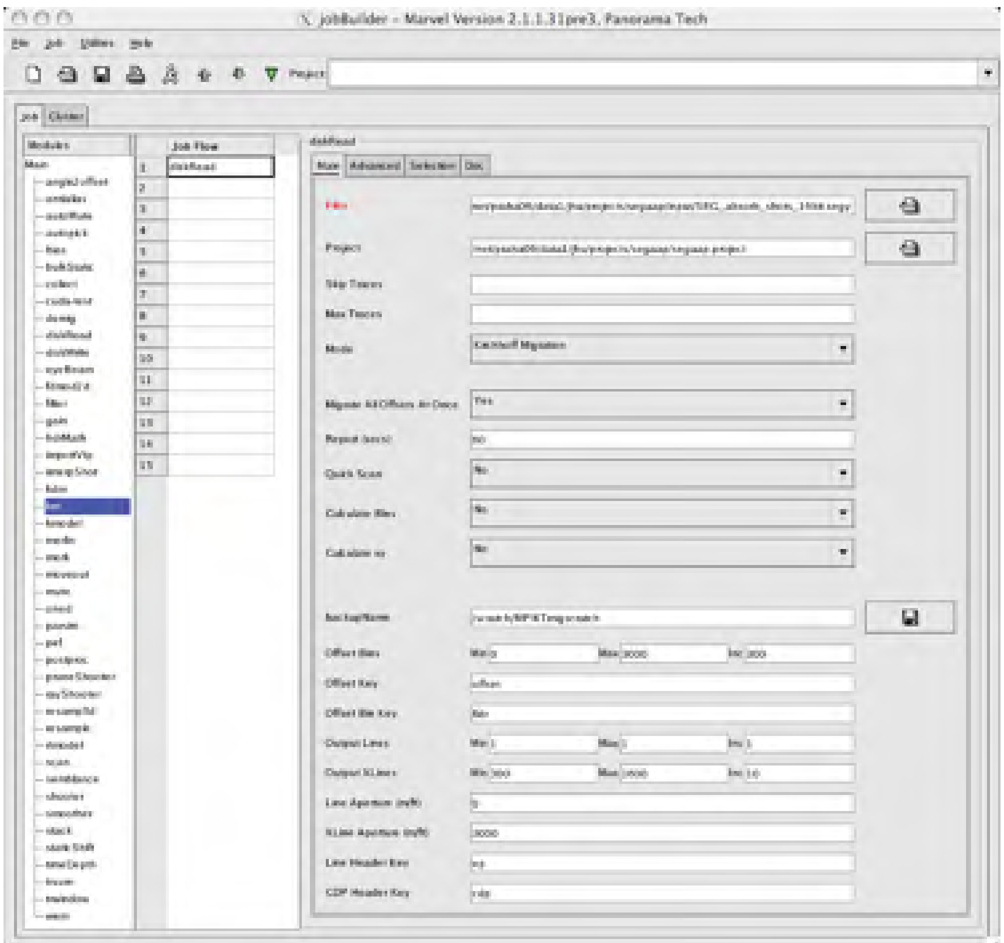
4. Select Kirchhoff Migration for the Mode.

Note:

You should use the Kirchhoff Migration Mode only if you are going to be performing either a Kirchhoff depth migration using the kdm module or a Kirchhoff curved-ray time migration using the km module; otherwise, you should use the General Mode or Shot Migration.

The JobBuilder window is redisplayed for the Kirchhoff migration mode, as shown in Figure 21.

Figure 21. JobBuilder Window, Kirchhoff Migration Mode



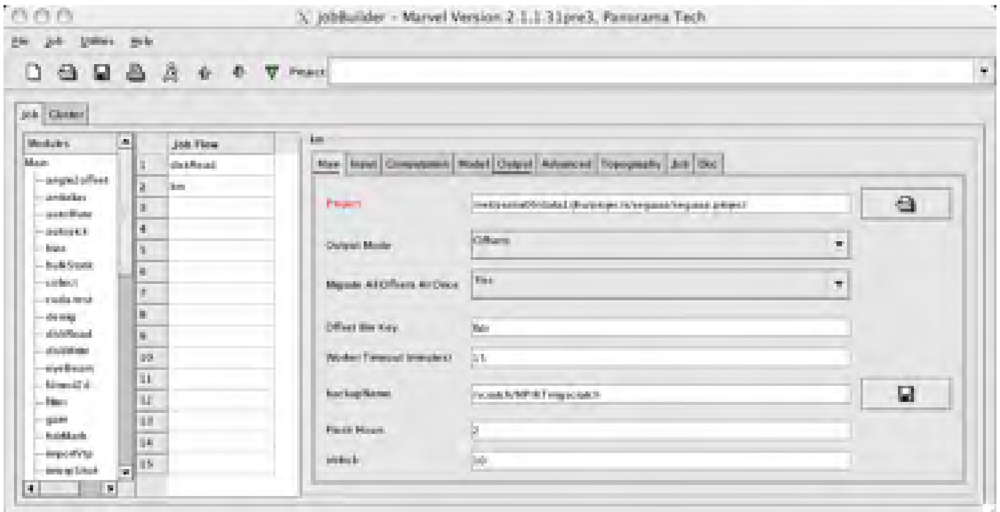
5. Enter the appropriate values for this panel, including:
   - data files
   - offset bins
   - output lines and crosslines

- line and crossline apertures

6. Double-click on the km module in the Modules tree.

   MARVEL adds the km module to the Job Flow, and displays the km pane in the jobBuilder window, as shown in Figure 22.

Figure 22. JobBuilder Window with Main Pane

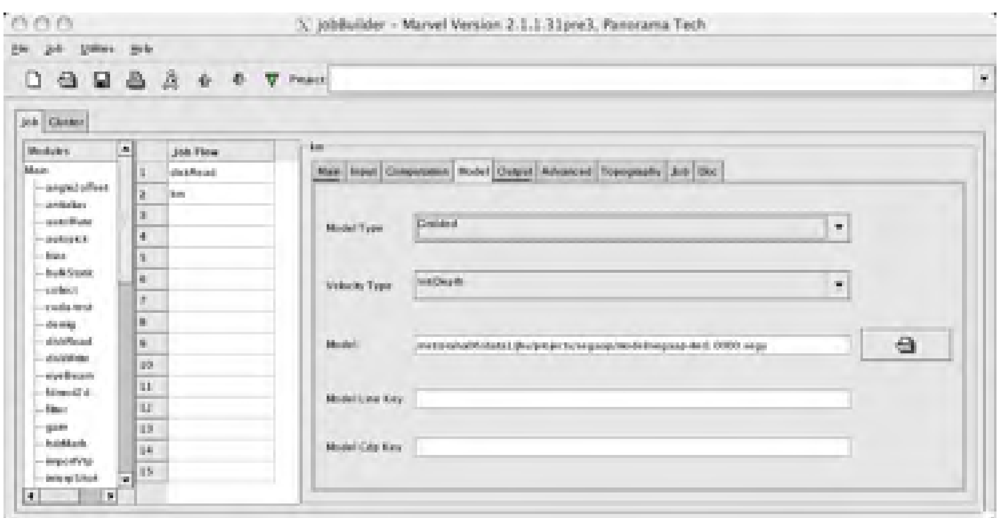


7. Select Offsets in the Output Mode field.

8. Select the Model tab in the km pane.

   MARVEL displays the JobBuilder window with the Model pane as shown in Figure 23.

Figure 23. JobBuilder Window with Model Pane



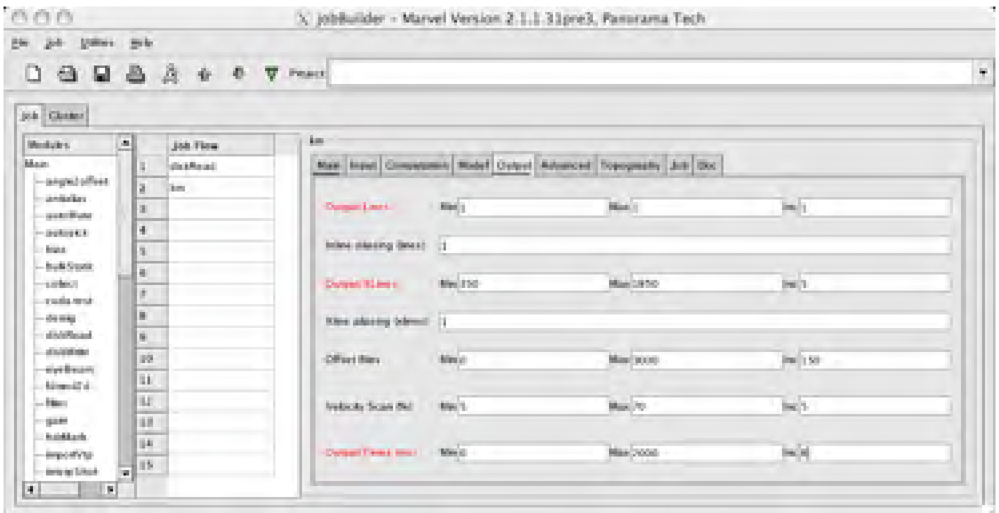
9. Select the File Open button at the right side of the Model field.

   MARVEL displays an Open window that enables you to select the output dataset you saved in Building a Model.

10. Select the Output tab in the JobBuilder window.

    MARVEL displays the JobBuilder window with the Output pane as shown in Figure 24.

Figure 24. JobBuilder Window with Output Pane

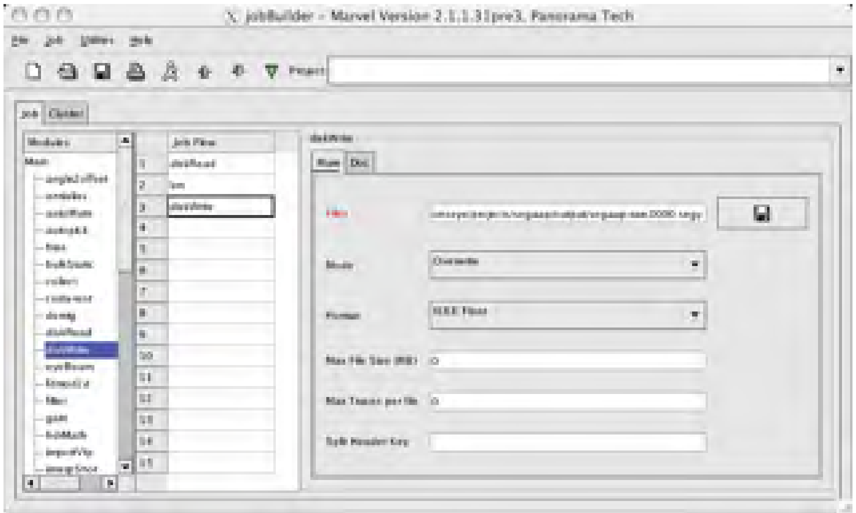


11. Enter the values for the Output Lines, Output XLines, and Output Times in the appropriate fields.

12. Double-click on the diskWrite module in the Modules tree.

    MARVEL adds the diskWrite module to the Job Flow list, and displays the diskWrite pane in the jobBuilder window, as shown in Figure 25.

Figure 25. jobBuilder Window with diskWrite Pane



13. Select the File Open button to the right of the Files field.

    MARVEL displays an Open window that enables you to select the file you want to use for storing the output of the job.

> **Note:**
>
> Be sure that the directory to which the job is writing output has the appropriate read and write permissions. Also, be sure the job can overwrite any files already in the directory.

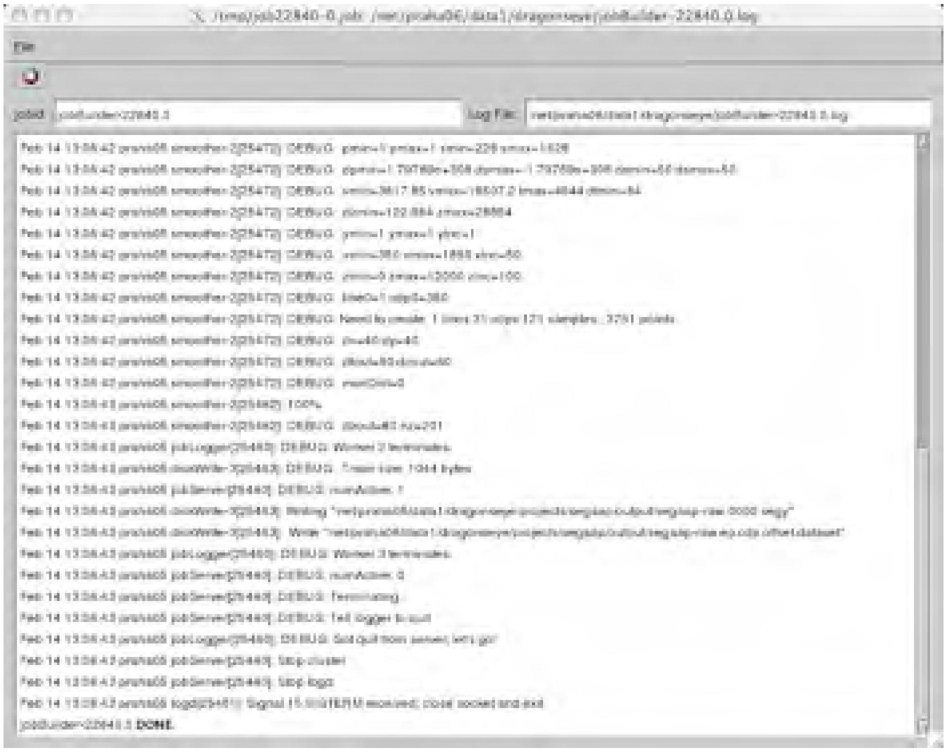
14. Select File => Save as... to save the job with an extension of .job.

15. Select Job => Run or click the Run icon in the toolbar.

> **Note:**
>
> If there is a file with extension .restart in the directory where the job writes the backup file (Figure 22), your job will not run. You must delete the file manually before you can run the job.

16. MARVEL displays the Job Log window shown in Figure 26. The figure shows an example Job Log window as it looks when the job has run to completion.

Figure 26. Job Log Window



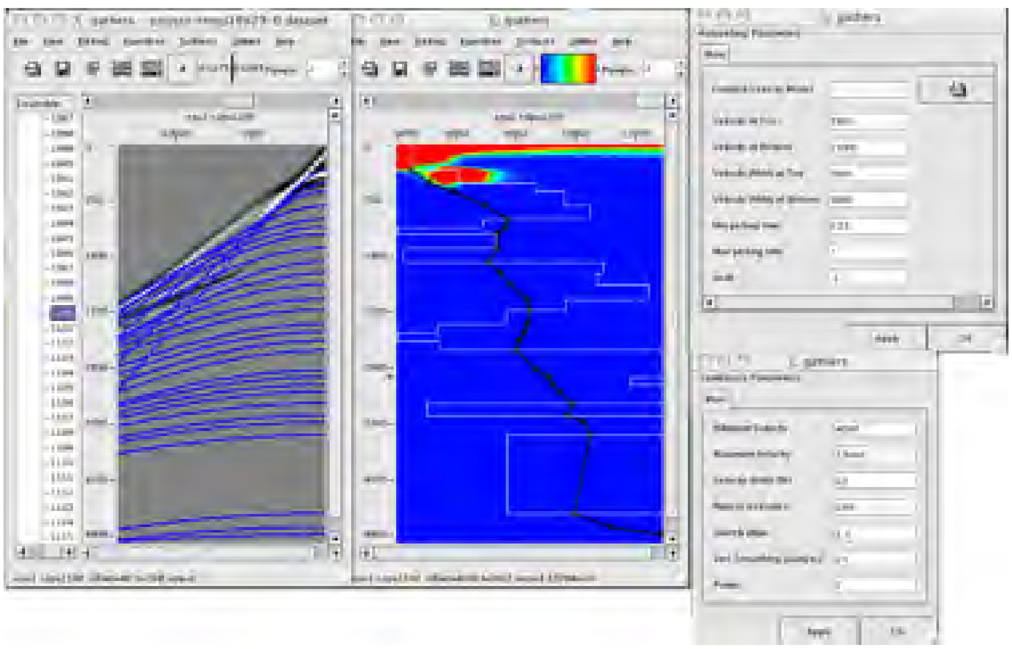
Step 5. Update the Picks

The first step of the iterative process for improving the model is to develop a new set of picks that includes all of the ensembles you want to examine.

1. Select the ensembles you want to use on the CDP map, see Figure 4.
2. Change the autopick parameters and semblance parameters for the gathers windows (Figure 6 and Figure 7), as desired.
3. Select Picking => Autopick All to autopick all of the ensembles.

   Figure 27 shows a new set of autopick and semblance parameters, along with the matching pick and semblance views.

Figure 27. Pick and Semblance Views



4. Save the picks.

Step 6. Build a New Model

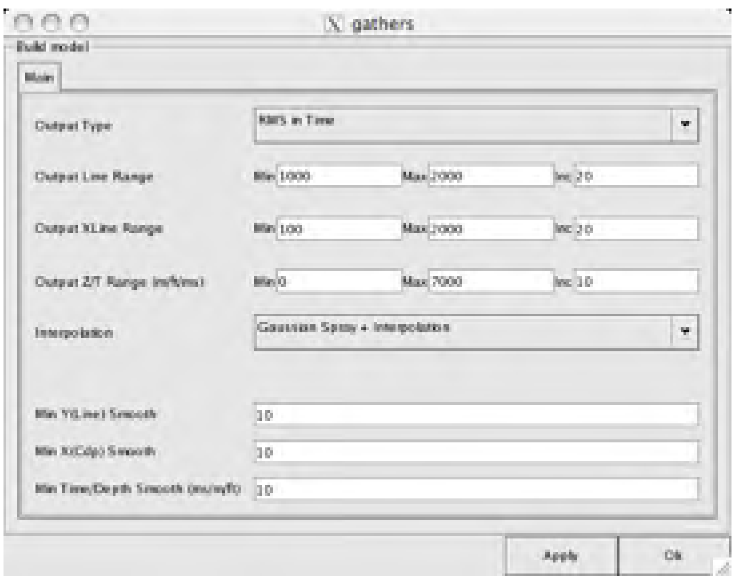
Now that you have updated the picks, you can build a model that will represent your data more accurately than the original model.

To Build the New Model

1. Select Functions => Build model...

   MARVEL displays the Build model window shown in Figure 28.

Figure 28. Build Model Window



2. Enter values for the Output Line Range, Output XLine Range, and the Output Z/T Range. Sample values are shown in the figure.

3. Click OK.

   MARVEL displays a Save As window.

4. Enter a file name for the model and click Save.

   The data is saved with an extension of .segy, and after a few moments, MARVEL displays the new model as that shown in Figure 29.

Figure 29. New Model Window

Step 7. Run a New Kirchhoff Time Migration Job

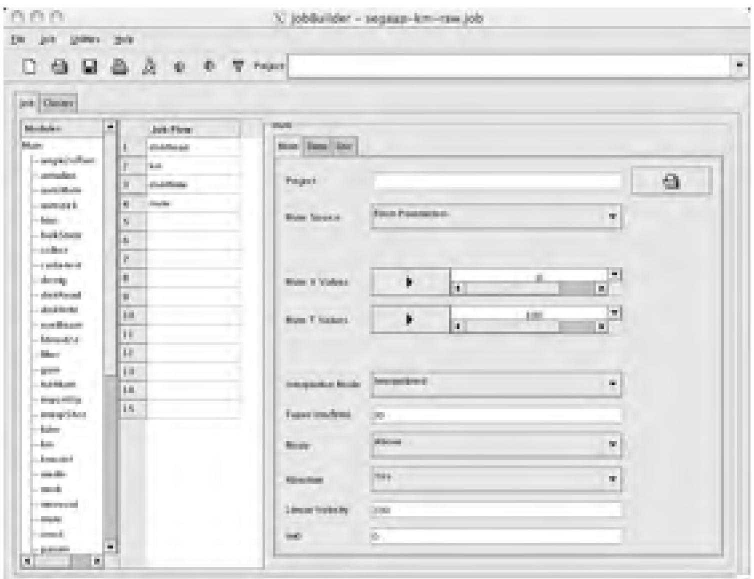
You will now build another job to analyze the data using the model you just created. You will also use the MARVEL automute feature to mute the first arrivals on the traces.

1. Open the job you saved in Step 14 on page 29.

2. Double-click the autoMute module in the module tree.

   The autoMute module is placed at the end of your Job Flow list.

Figure 30. Add autoMute Module



3. Click the up arrow in the toolbar to move the autoMute module to become the second module in the list.

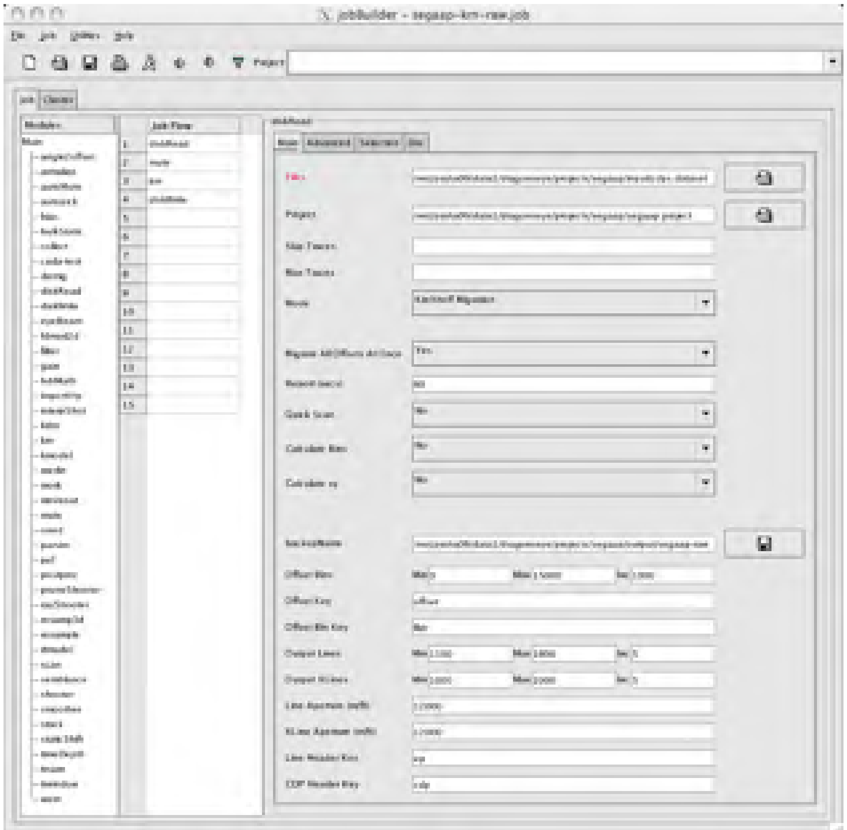
4. Select From Data Headers in the Mute Source field.

5. Select the diskRead module in the Job Flow list.

6. Change the Offset Bins setting to reduce the increment, as shown in Figure 31.

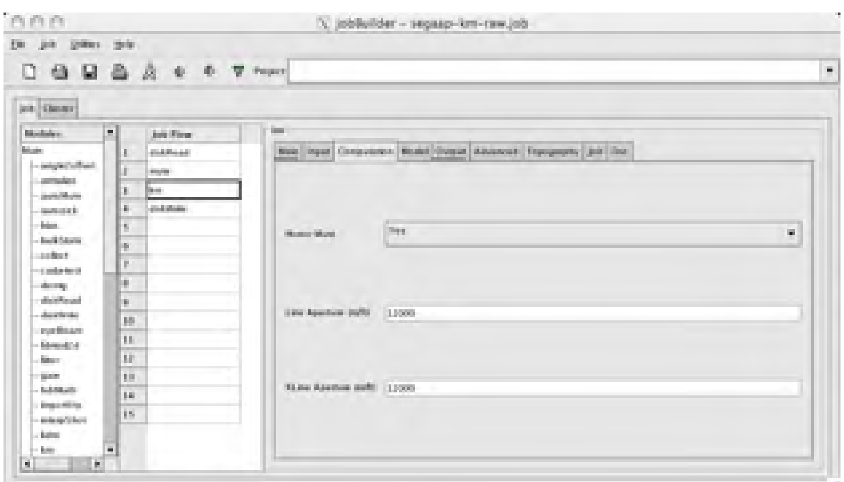
7. Change the values for the Output Lines, Output XLines, Line Aperture, and XLine Aperture to reflect the values you used in building the new model.

Figure 31. jobBuilder Window with diskRead Pane



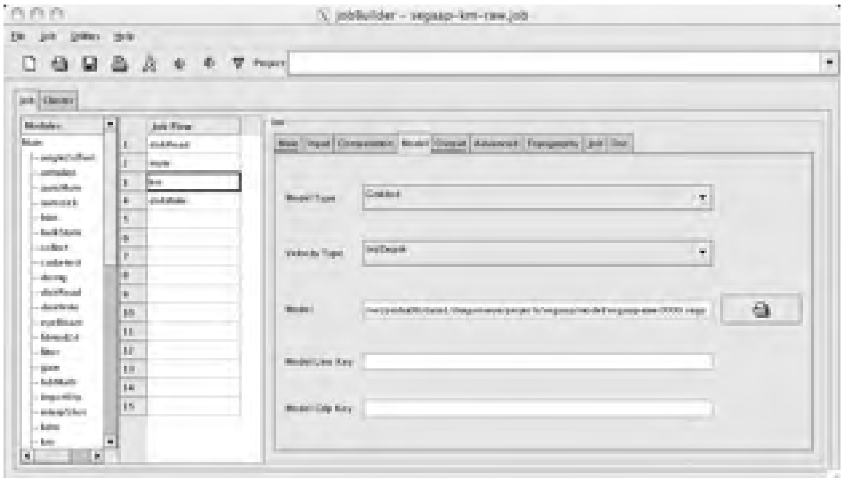
8. Select the Computation tab in that module, as shown in Figure 32.

Figure 32. jobBuilder Window with Computation Pane



9. Change the Honor Mute setting to Yes.
10. Select the Model tab in the km pane.

Figure 33. jobBuilder Window with km Pane



11. Enter the name of the model that you built based on Autopick All.

12. Click on the diskWrite module in the Job Flow list.

13. Select the File Open button to the right of the Files field. MARVEL displays an Open window that enables you to select the file you want to use for storing the output of the job.

> **Note:**
>
> Be sure that the directory to which the job is writing output has the appropriate read and write permissions. Also, the job must be able to overwrite any files already in that directory.

14. Select File => Save as... to save the job with an extension of .job.

15. Select Job => Run or click the Run icon in the toolbar.

> **Note:**
>
> If there is a file with extension .restart in the directory where the job writes the backup file, your job will not run. You must delete the file manually before you can run the job.

MARVEL displays the Job Log window when the job has run to completion.

16. Click the Inline section button in the MARVEL Start Window.

MARVEL displays a blank gathers window.

17. Select File => Open or click the button.

    MARVEL displays an Open dialog box that enables you to select the output file you generated in Step 14 on page 35.

    MARVEL displays the inline data.

Step 8. Build a Normal Moveout Job

You will now build another job to perform a normal moveout on the data you generated in the previous section.

1. Open the job you saved in Step 13 on page 35.
2. Right click on the km module in the Job Flow list.

   A Delete pop-up box displays beside the cursor.
3. Click on Delete to delete the km module from the Job List.
4. Double-click on the moveout module in the module tree to move it to the Job Flow list.
5. If necessary, click the up arrow in the toolbar to reposition the moveout module so it becomes the second module in the list.
6. Select the project.
7. Enter the name of the model that you built based on Autopick All in the Velocity Model field.
8. In the Direction field, select Inverse.
9. Click on the diskWrite module in the Job Flow list.
10. Select the File Open button to the right of the Files field.

    MARVEL displays an Open window that enables you to select the file you want to use for storing the output of the job.

> **Note:**
>
> Be sure that the directory to which the job is writing output has the appropriate read and write permissions. Also, be sure that the job can overwrite any existing files in the directory.

11. Select File => Save as... to save the job with an extension of .job.
12. Select Job => Run or click the Run icon in the toolbar.

    MARVEL displays the Job Log window when the job has run to completion.

Step 9. Pick the Updated Data

1. Click CDP Gathers in the Panorama Technologies MARVEL Start window.

   MARVEL displays a blank Gathers window.

2. Click the File Open button in the tool bar.

3. Select the dataset you just created in the previous section.

4. Select Picking => Autopicking parameters... in the Gathers window.

   MARVEL displays an Autopicking Parameters window.

Note:

For information about how MARVEL performs autopicking, see autopick Module in the *MARVEL Reference Manual*.

5. Enter the desired values for the autopicking parameters, and click OK to accept the entries.

6. Select Functions => Semblance Parameters... in the Gathers window.

7. Enter the desired semblance parameters.

   In this case, they were the same as what was used in the earlier semblances.

8. Select Functions => Autopick All in the Gathers window.

Step 10. Build an Updated Model

You can now build an updated model from the autopicked data.

1. Select Functions => Build model... in the Gathers window.
2. Select Functions => Build model... in the Gathers window.

   MARVEL displays the Build model window.
3. Select an Output Type of RMS in Time.
4. Make sure the values for the Output Line Range and the Output XLine Range blank.
5. Enter values for the Output Z/T range.
6. Click **OK**.

   MARVEL displays a Save As window.
7. Enter a file name for the model and click Save. The data is saved with an extension of .segy, and after a few moments, MARVEL displays the model (Figure 34).

Figure 34. Model Window

8. Click the Inline section button in the MARVEL Start Window.

   MARVEL displays a blank gathers window.
9. Select File => Open or click the button.

   MARVEL displays an Open dialog box that enables you to select the output file you generated in Step 12 on Page 48. MARVEL displays the inline data. A zoomed view of the inline data is shown in Figure 35.

Figure 35. Inline Data

If this view of the data meets your requirements, you are finished with the data modeling process.

If you want a better view of the data, return to Step 6. Build a New Model to perform another iteration of the modeling process, using the dataset you generated with the normal moveout (see Step 8. Build a Normal Moveout Job) as the starting point for the next iteration.

Appendix A

Topographic Migration Velocity Analysis

This appendix provides a work flow on which topographic migration velocity analysis can be based. As such, the focus is on the direct estimation of a reasonable first velocity volume from datum-corrected data, followed by a general recipe for the subsequent velocity analysis from the true, or nearly true, topographic surface.

Introduction

It is necessary to migrate seismic data from the Earth's surface in areas of significant elevation variation. Proper application of this technology has two necessary requirements:

- The topographic surface of elevations must be part of the migration algorithm. Whether performing prestack-time or prestack-depth migrations, travel times must be calculated from this surface rather than from datum.
- The topographic surface represents time zero, and, as such, all velocity analysis must be performed from this surface.

This second statement is true regardless of whether or not migration is part of the analysis procedure. Rough topographic surfaces generally make the initial velocity analysis difficult or impossible, and as a result, velocity estimation is usually based on data to which elevation statics have been applied. While this method frequently produces an acceptable image, velocities estimated in this way are not normally useful in a migration from topography. Consequently, when the initial model is estimated from data at the datum, the image must be shifted back to the true topographic surface to be useful in the migration velocity analysis stage.

After a reasonable initial model is available, subsequent migration from topography velocity analysis can update the initial model to improve the image until differences between neighboring iterations are negligible. Thus, a major problem with migration from topography is not how to perform the process, but, instead, how to estimate a reasonable starting velocity to start the iterative process that determines a reasonable final imaging velocity volume.

Table 1 identifies the terms used in the equations in the rest of this appendix.

Table 1. Equation Terms Used in the Appendix

Term	Description
selev	source elevation
sdepth	source depth
gelev	receiver elevation
wevel	weathering velocity
swevel	sub weathering velocity
sdel	datum elevation
sut	source up hole time
sstat	source static (time)
gstat	receiver static (time)

continues on next page

Table 1. Equation Terms Used in the Appendix – continued

Variable	Description
tstat	total static (time)
τ	time from datum

Elevation Statics

There are three simple equations for computing elevation statics. These equations are:

Equation 1:

$$sstat = \frac{sdel - (selev - sdepth)}{swevel}$$

Equation 2:

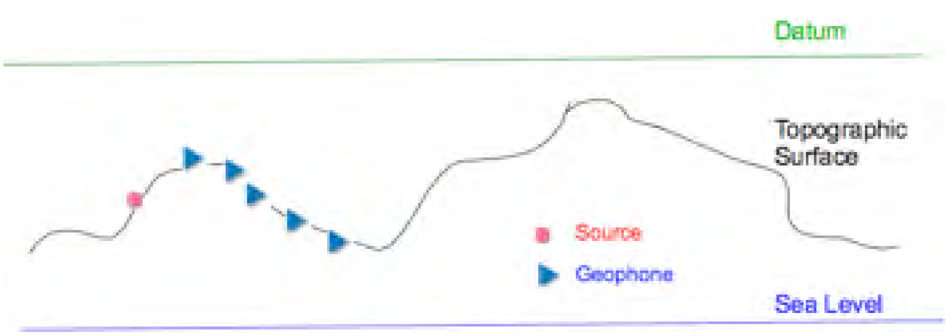
$$gstat = sstat - sut$$

Equation 3:

$$tstat = sstat + gstat + \frac{selev - gelev}{wevel}$$

To fully understand this process, consider Figure 36, where you can see all of the features of a typical land or ocean bottom cable acquisition. It is a simple topographic schematic with sources and receiver on the topographic surface. There is a zero datum or sea level, a datum that is above all source and receiver elevations, and a topographic surface on which the data is actually recorded. Normally, the datum would be set equal to the highest source or receiver elevation, but this is purely a matter of choice. In fact, the datum could be chosen anywhere on the model. It could even be below sea level, but this is rarely done.

Figure 36. Simple Topographic Schematic



Elevation static correction consists of two basic steps. As shown in Figure 37, the first step vertically shifts all of the receivers so they are theoretically aligned on the source. The second step simply shifts this entire shot profile to the desired datum.

Figure 37. Simple Topographic Schematic After Correction to Source Position

Figure 38 shows what happens if you set the datum elevation to zero, that is, sea level.

Figure 38. Simple Topographic Schematic After Correction to Sea Level

In this case, the term $\frac{selev - gelev}{wevel}$ shifts the trace samples so that, in effect, both source and receiver are at the same level. Then, Equation 4 shifts the trace samples so that the sources and receivers now lie on the surface representing zero datum or sea level.

Equation 4:

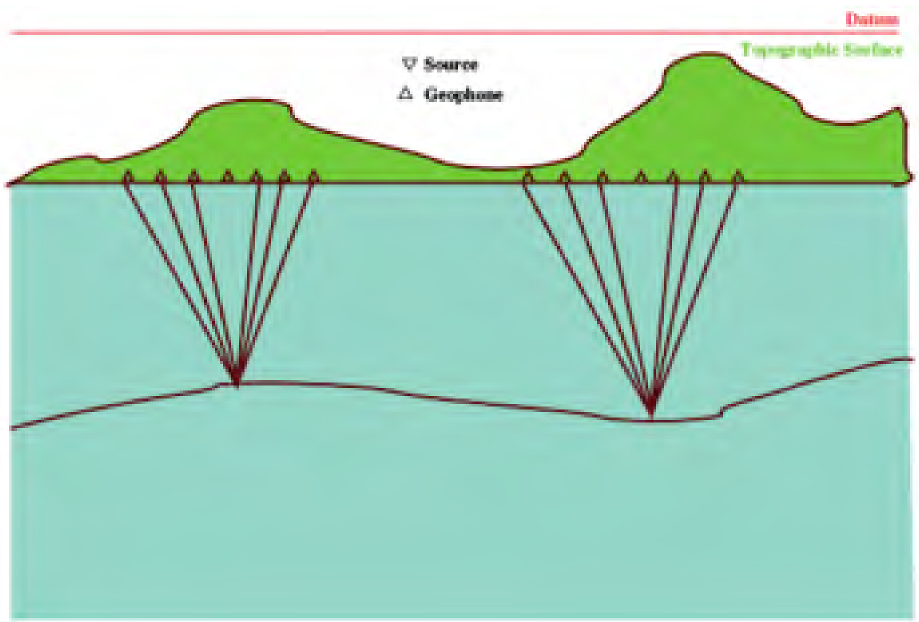
$$sstat + gstat = 2.0\left[\frac{sdel - (selev - sdepth)}{swevel} - sut\right]$$

After the trace has been shifted so that it resides at sea level, the datum elevations term, *sdel* in Equation 1 takes over and shifts the traces so that sources and receivers are now essentially on the datum surface. This is illustrated in Figure 39.

Figure 39. Simple Topographic Schematic After Correction to Datum Elevation

Figure 40 represents how data just prior to the beginning of the imaging or migration part of the processing sequence. At this point, you will probably be able to make reasonable estimates of an initial velocity volume. The area above the flat datum is assigned a constant velocity, while velocities are estimated using traditional-style, hyperbolic-arrival assumptions below the datum.

Figure 40. Building an initial velocity model from topography using a single datum



Initial Velocity Models

It is important to note that velocity analysis can only take place when and if the midpoints of each trace within a given CDP are located on the same level. As Figure 40 and Figure 41 show, elevation static corrections can be applied trace-by-trace to produce a data set for which each CDP has this property. Figure 40 shows data that are shifted to a flat datum. In this figure, the datum is at the minimum topographic elevation, but any datum, including the true topographic surface, will suffice, as shown in Figure 41.

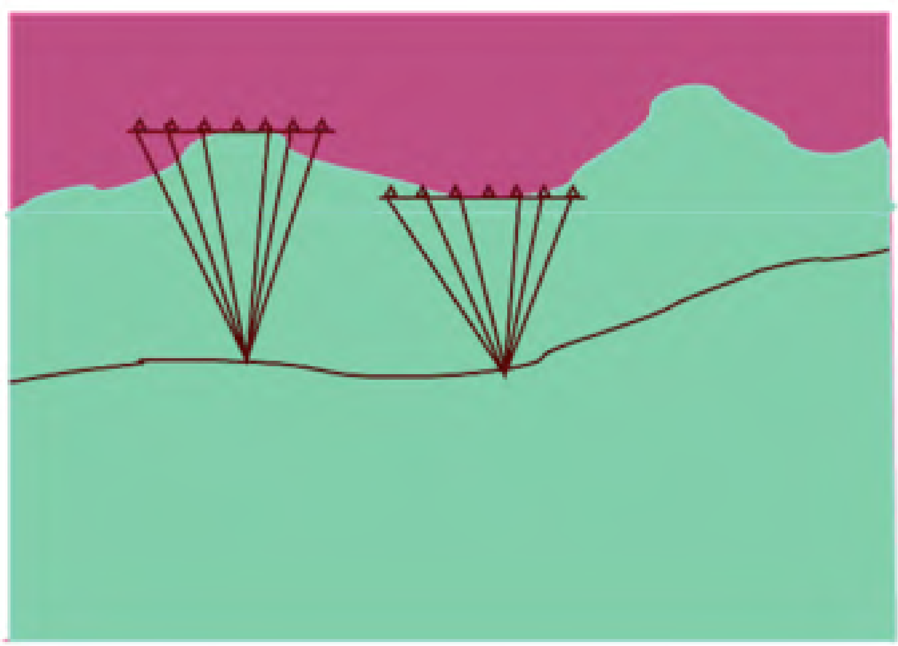
Of course, you should chose *swevel* and *wevel* as close to the true weathering and subweathering velocities as possible, but the real objective is to produce CDP's with arrivals that are as close to hyperbolic as you can achieve. If *swevel* and *wevel* satisfy this requirement, arrivals will be sufficiently hyperbolic so that traditional velocity analysis can proceed, at least in the short-offset case. Moreover, if you choose the CDP-by-CDP shift properly (in other words, to the right surface), this suggests that the initial velocities will be more accurate than those estimated from datum.

Thus, it is clear that to obtain a reasonable initial velocity field, you must first shift the input data to a suitably chosen horizon.

To make this as simple for you as possible, Panorama Technologies has chosen to shift all the traces automatically within any given CDP to whatever you specify as the “true” topographic surface.

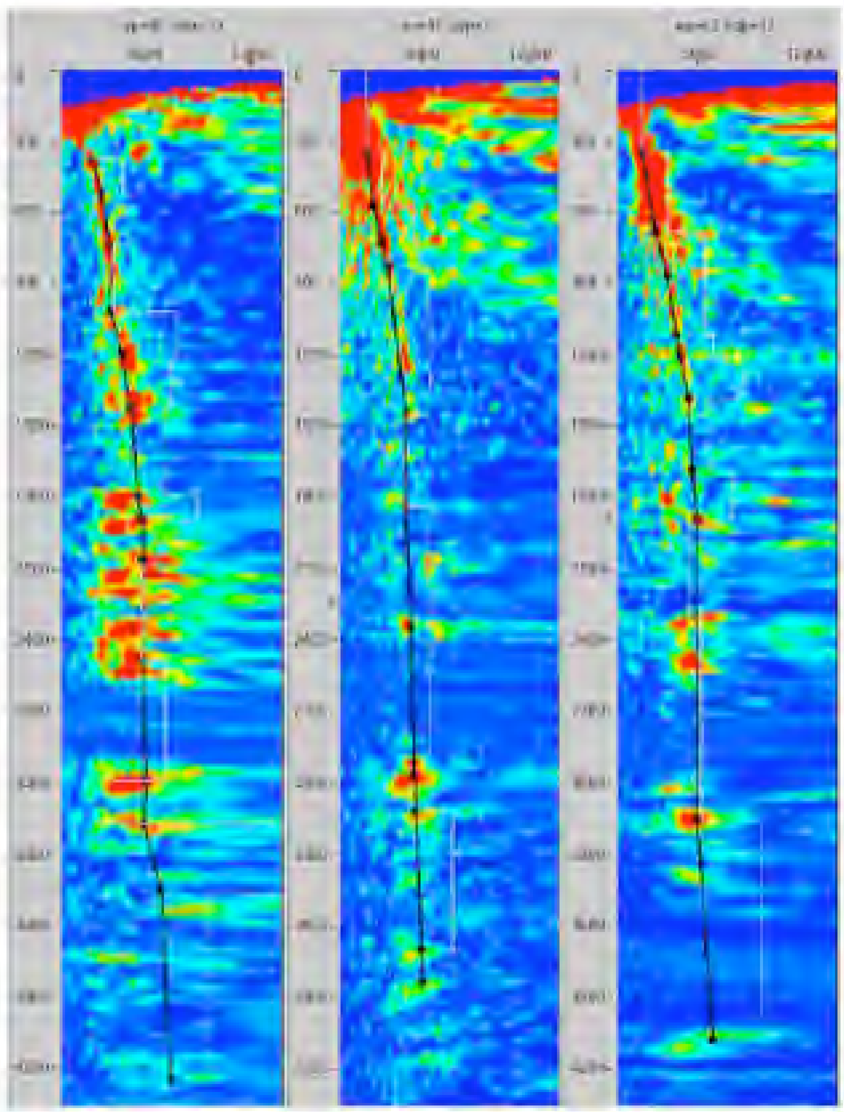
Thus, assuming that appropriate values are in proper header locations, the Gathers module, which is the velocity analysis module supplied by Panorama Technologies, performs the shift automatically to be consistent with Figure 41. You should recognize that you can allow *swevel* and *wevel* to vary across the entire midpoint range, although this is seldom done.

Figure 41. Initial Velocity Model



After the data headers have been set properly, the initial velocity analysis is straightforward. Semblance panels are analyzed and picked in exactly the same manner as you do when the topographic elevation variations are negligible. The picking process, as illustrated in Figure 42, is exactly the same as in a normal processing mode. If the topographic variation within a CDP is relatively small, this process should work well. Note that you will not see the shift to a topographic surface.

Figure 42. Picking an initial velocity model from topography



Migrating from Topography

Before you migrate from topography, you need to make sure that the observed seismic data are shifted to their original time zero locations. This means that sources and receivers must be at the topographic surface as indicated in Figure 36.

If the input data is hung from datum, you must first apply the appropriate static shift module to place the data back at the correct surface elevations. Since the Gathers module automatically applies the proper static shifts to the data on input, it is also necessary that data trace headers be set correctly. When these objectives have been met, it is quite simple to apply the Panorama Technologies time and depth migration modules in topographic mode.

The time migration module handles all topographic variations automatically, and calculates the proper travel times from the given topographic model. To produce accurate travel times, Panorama Technologies' most energetic raytracer must be provided with a suitable topographic model and surface. On successful execution of either of the three raytrace modules, shooter, pruneshooter or raytracer, the resulting travel time tables contain all of the necessary information for an accurate migration of the input data.

Migration Velocity Analysis

Migration velocity analysis from topography proceeds through a normal iterative process. Since migration has the net effect of producing coincident source and receiver data, you can think of the migration output as having been topographically shifted, as indicated in Figure 41, which shows building an initial velocity model from topography using multiple datums. Note that at each midpoint, the datum is exactly equal to the midpoint's elevation. In this case, velocity estimation is directly *from topography*. Thus, migration velocity analysis proceeds in exactly the same manner as it did for the initial stacking velocity analysis of the preceding section. The only serious concern is that you have to perform all of these tasks without error.

Workflow

The following simple workflow provides a procedure for migrating from topography. The sequence is as follows:

1. Prepare the data so that header locations corresponding to *selev*, *sdepth*, *gelev*, *wevel*, *swevel*, *sdel*, *sut*, *sstat*, *gstat*, and *tstat* are set correctly.

2. Remove any elevation statics that may have been applied to the data. Residual statics need not be removed, but refraction statics must not have been applied.

3. Perform an initial stacking velocity analysis to determine an initial migration velocity model. (Make sure that the Gathers module is in topographic mode.)

4. If running a depth migration, generate travel times from topography.

5. Migrate using the current velocity model.

6. Repeat the migration velocity analysis until you obtain suitable results.

Appendix B

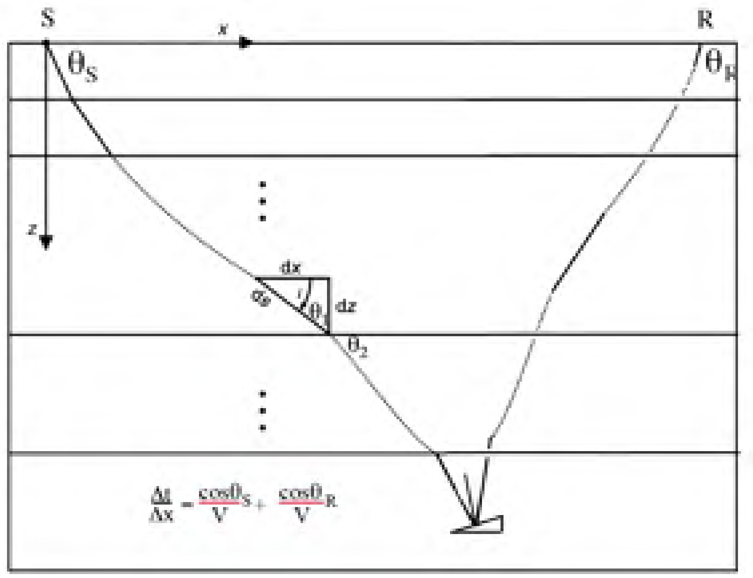
Using the eyeBeam Module

This appendix discusses the optimum utilization of Panorama Technologies eyeBeam module and provides a brief explanation of the assumptions underlying the process. The article is divided into three parts: First, a brief overview summarizes the fundamental principles on which the module is based; second, the focus is directed toward data preparation for optimum results; and, third, the module's parameters are explained in detail.

Overview

In contrast to the Kirchhoff smear stack approach, eyeBeam estimates local dip elements, or beamlets, from the input data and then applies classical imaging principles to produce a properly migrated image. To a large extent, the process is based on Figure 43. Together with the near surface velocity, V, Equation 5 provides the precise relationship between the apparent dip, $\frac{\Delta t}{\Delta x}$, and the source and receiver take-off angles, θ_S and θ_R, respectively. As indicated in the figure, raytracing is used to locate the reflector that gave rise to the apparent dip. The opening angle, or either incidence angle at the intersection of the two rays, determines reflector dip. The sum of the source and receiver take-off angles are directly related to the apparent dip of a subsurface reflector. The opening angle (sum of the incidence angles) define the local reflector dip.

Figure 43. Fundamental components of Panorama Technologies eyeBeam module.



Equation 5:

$$\frac{\Delta t}{\Delta x} = \frac{cos\theta_S}{V} + \frac{cos\theta_R}{V}$$

The ratios $\frac{cos\theta_S}{V}$ and $\frac{cos\theta_R}{V}$ are the derivatives, $\frac{d\tau_S}{dx}$ and $\frac{d\tau_R}{dx}$, of the source and receiver traveltimes, that is, the gradients in 3D. This fact allows you to compute dynamic traveltimes and provide proper amplitude correction to preserve AVO response and approximate true amplitude imaging. Figure 43 shows the schema for two-dimensions, but the general concept remains valid in three dimensions. In 3D, the source and receiver take-offs are specified by an azimuth and dip.

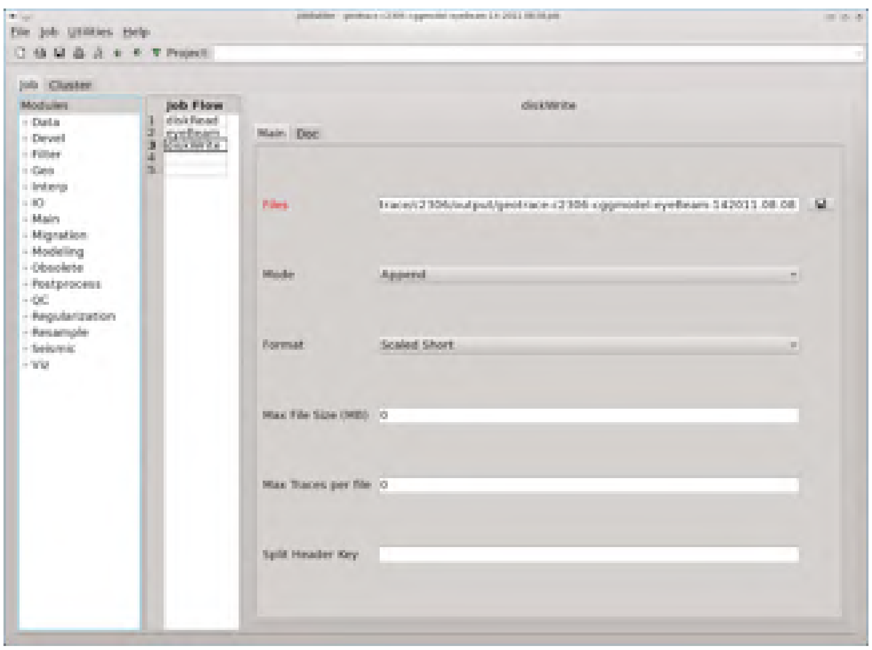
Key features of eyeBeam and its dynamic raytracer include

- Migration From Topography
- Full TTI imaging based on the dynamic raytracer
- Raytrace arrivals include
  - maximum energy
  - minimum velocity
  - minimum distance

Data Preparation

The quality of the final image produced by eyeBeam is controlled by the quality of the input data. In many cases, the quality of the final image can be controlled by proper preprocessing of the input data volume. Because estimates of local apparent dip are paramount, data preparation can and should focus on input data coherency and frequency content. In many cases, coherency be increased through signal enhancement applications and also careful choice of the frequency bandwidth. Spectral analysis should reveal the extent to which the data can be resampled to a larger sampling increment. Efficiency is highest when the maximum frequency can be limited to 31.5 Hz. To maximize throughput, data preparation should also include storing the properly processed data in a scaled 16-bit format. This reduces the input data size and results in faster data reads and writes. As shown in Figure 44, you do this by setting the output data format to Scaled Short. This setting effectively halves the output file size, thereby improving overall performance and efficiency of eyeBeam migrations.

Figure 44. Setting the output file format to scaled-short in diskWrite



Parameterization

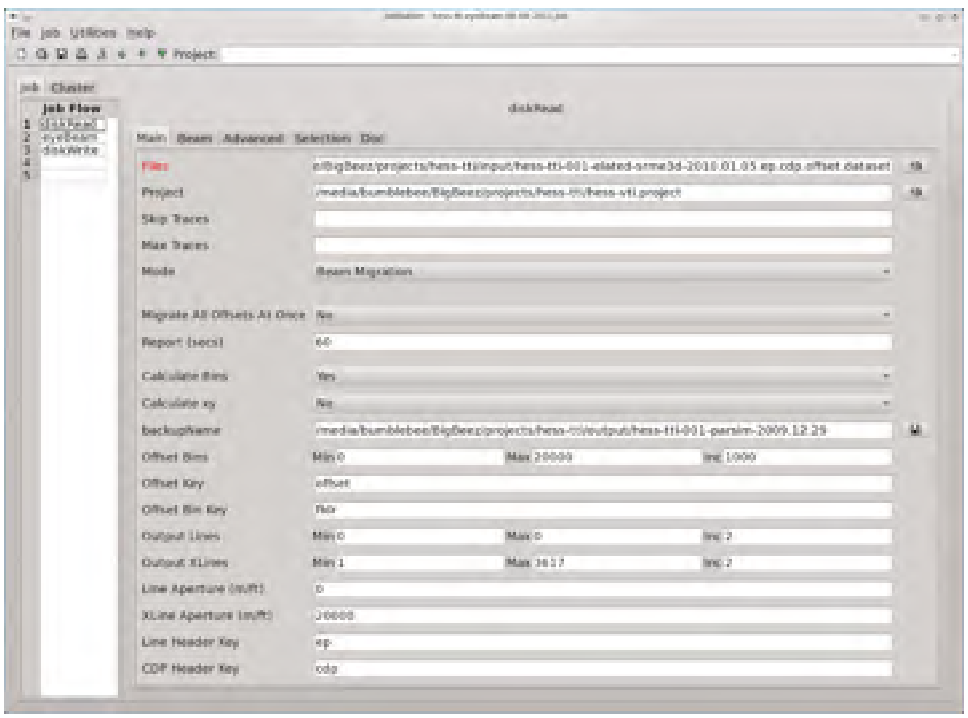
Parameterization of an eyeBeam project uses both the diskRead module and the eyeBeam module.

diskRead

Parameterization of an eyeBeam project begins with the diskRead module. Figure 45 shows the Main panel of diskRead along with the parameters required to ensure that diskRead will construct appropriately-sized super-gathers (patches) to feed to the eyeBeam module. At your discretion, the parameters below diskRead's Mode field can also be set in the eyeBeam module. While several of these fields have reasonable defaults, they should always be reviewed before job submission. The fields are defined in Table 2.

The panel shows selection of Beam Migration in the Mode field. This ensures that diskRead constructs appropriate patches to feed to the eyeBeam module after the sort has been performed.

Figure 45. diskRead Main Panel



The Files field in Figure 45 points to either a .segy file with properly filed headers or to a .dataset file containing the appropriate coordinate information to enable proper sorting of the input data into offset binned super gathers. Depending on the computer system being used, the sort may take awhile.

Table 2. diskRead Main Panel Fields

Field	Description
Migrate all Offsets at Once	If set to YES, traces are read in exactly the order they are stored. If set to NO, traces are read in common offset order. This should be set to NO for Beam Migrations.
Report (secs)	This is the report time for diskRead. For example, a value of 60 causes diskRead to report every minute.
Calculate Bins	If the input data has only CDPs and lines in the headers, this will calculate the real world coordinates for selecting super-gathers.
Calculate xy	If the input data does not have CDPs and lines set in the header, set this parameter to YES to have diskRead calculate CDP and line information from the world coordinates in the trace headers.
backupName	The name and location of the backup file used in restarts and recovery operations.
Offset Bins	These three fields are the minimum, maximum, and increment defining the leading edges of the offset bins. For example, if these values are set to 0, 20000, 100, then 199 offset bins beginning at 50 will be output.
Offset Bin Key	The header key for storing the Offset Bin value.
Output Lines	These three fields are the minimum, maximum, and increment for the desired output line range. Figure 45 is a 2D setup so the values are set to output a single 2D line.
Output XLines	The minimum, maximum and increment for the desired output line range.
Line and Xline Aperture	The desired line and crossline half-apertures.
Line and CDP Header Keys	The header locations where this information is stored.

The diskRead Beam panel is shown in Figure 46. The most important field in diskRead's Beam panel is the Patch Size field. This field defines the size of the patch in both line and crossline directions. Reasonable values are project dependent, but are typically in the neighborhood of 200 meters. The diskRead Beam fields are defined in Table 3.

Figure 46. diskRead Beam Panel

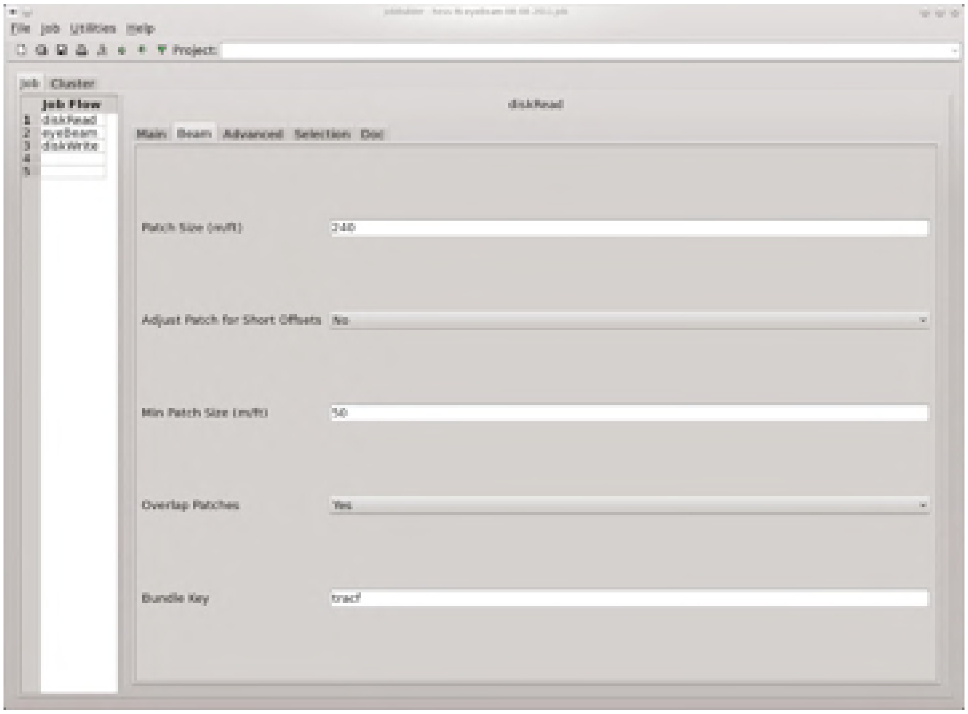
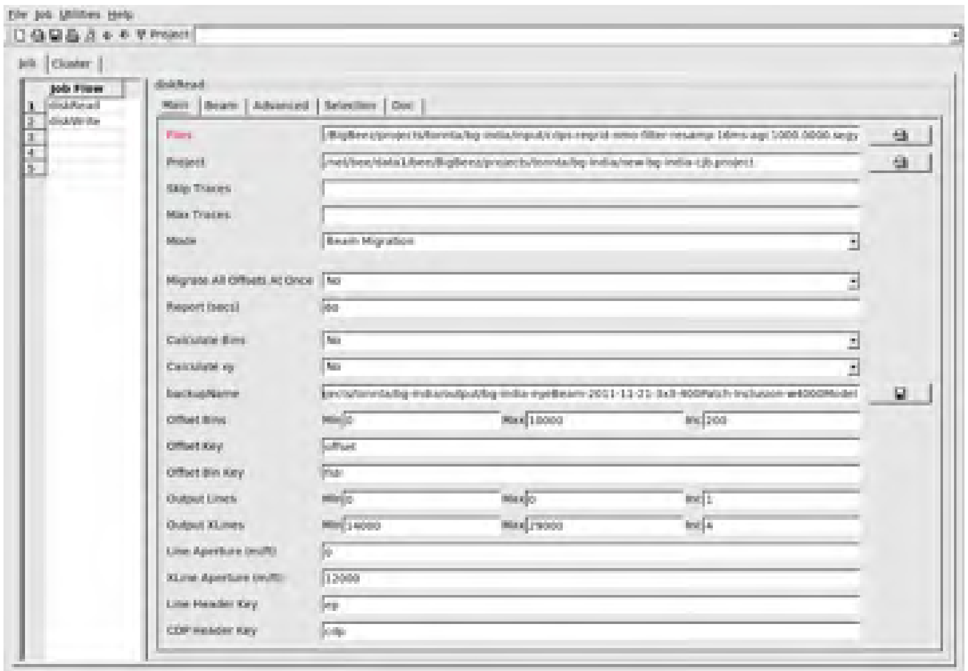


Table 3. diskRead Beam Panel Fields

Module	Description
Patch Size (m/ft)	The actual patch or super gather size. The value defines a square around sources and receivers. Every trace with a source and a receiver in the corresponding square is included in the patch.
Adjust Patch for Short Offsets	This switch specifies whether or not eyeBeam reduces the patch size for small offsets. It enhances coherence to improve beamlet estimation.
Min Patch Size (m/ft)	The minimum patch size allowed for short offset reduction when the previous field is set to YES.
Overlap Patches	Set this to YES to increase the coherence at shallow depths.
Bundle Key	The header word location containing the unique value assigned to each patch.

If you plan to do multiple migrations using the same input, the diskRead data stream of super gathers can be output in presorted form by feeding the output from diskRead directly into diskWrite, as shown in Figure 47. The diskWrite module writes the data to the specified super-gather-sorted output file. In subsequent applications of eyeBeam, diskRead's input file field must contain the .segy super-gather-sorted output file and the Mode field must be set to General. Note that this approach can also be used effectively during the data preparation step.

Figure 47. diskRead General Panel



eyeBeam

Figure 48 shows the eyeBeam Main panel. The fields are explained in Table 4, and give the necessary information for a successful run of the module.

Figure 48. eyeBeam Main Panel

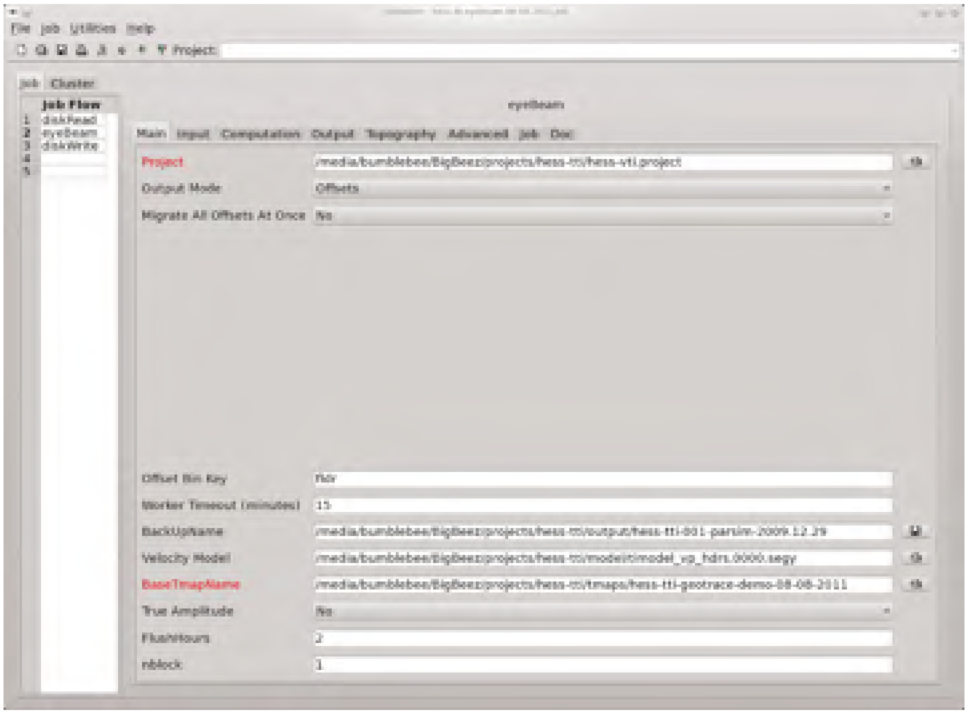


Table 4. eyeBeam Main Panel Fields

Module	Description
Project	This required field defines the input data geometric coordinates.
Output Mode	The output mode is normally Offsets or Stack. Use Offset mode to produce a range of offsets, and Stack to produce a stack of all offsets
Migrate All Offsets At Once	This should always be set to NO for eyeBeam migrations.
Offset Bin Key	It is recommended that this field be left at the default value, but any available header key can be used. **Note: Avoid using cdp, line, sx, sy, gx or gy used in other header related fields.**

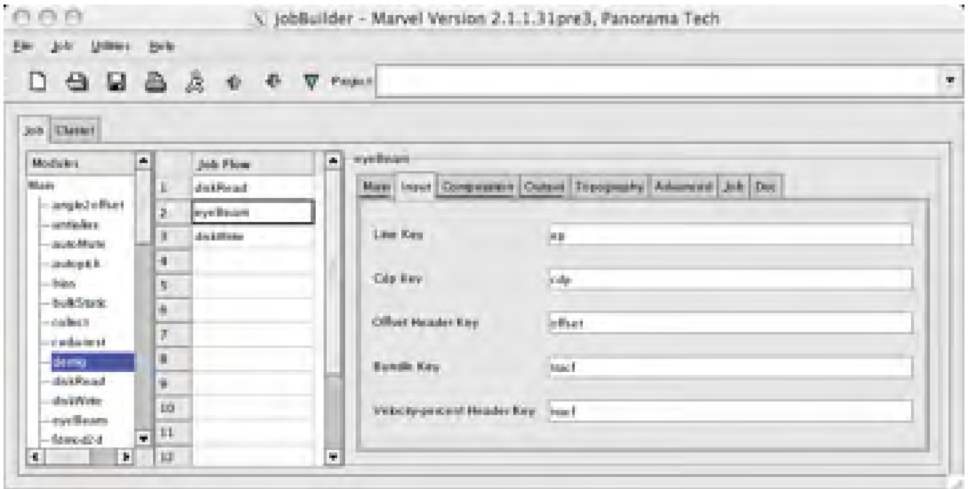
continues on next page

Table 4. eyeBeam Main Panel Fields–continued

Tab	Description
Worker Timeout (minutes)	This field defines the length of time eyeBeam will allow a worker to be inactive. If communication is not achieved after this time limit, the eyeBeam master assumes that the worker is dead.
BackUpName	This file is used to store backup information to enable a restart after any suspension of execution.
Velocity Model	When given, the velocity model serves two purposes. It provides the necessary velocity information for differential moveout of the traces in each super gather to the average offset of that ensemble. It also defines the near surface velocity for computation of take-off angles. This is a seismic file in any format MARVEL can read, typically, SEG-Y. The units are m/s, or ft/s, but can be anything else, depending on the choice of units in the seismic data.
BaseTmapName	The base name of the traveltime file(s). This name is the tmap file name without the .tmap extension. **Note: This is a REQUIRED entry.**
True Amplitude	When set to YES, eyeBeam will use the amplitudes computed during the generation of traveltimes by rayShooter.
Flush Hours	The length of time between each output of backup information to the BackUpFile.
nblock	**Note: This parameter should not be set when using eyeBeam.**

The eyeBeam Input Panel in Figure 49 defines the header storage locations for the various indicated parameters.

Figure 49. eyeBeam Input Panel



The eyeBeam Computation Panel is shown in three forms in Figure 50, Figure 51, and Figure 52. The fields on these panels are the most important eyeBeam parameters, and are defined in Table 5.

Figure 50. eyeBeam Computation Panel, Application Range Utilization field set to All

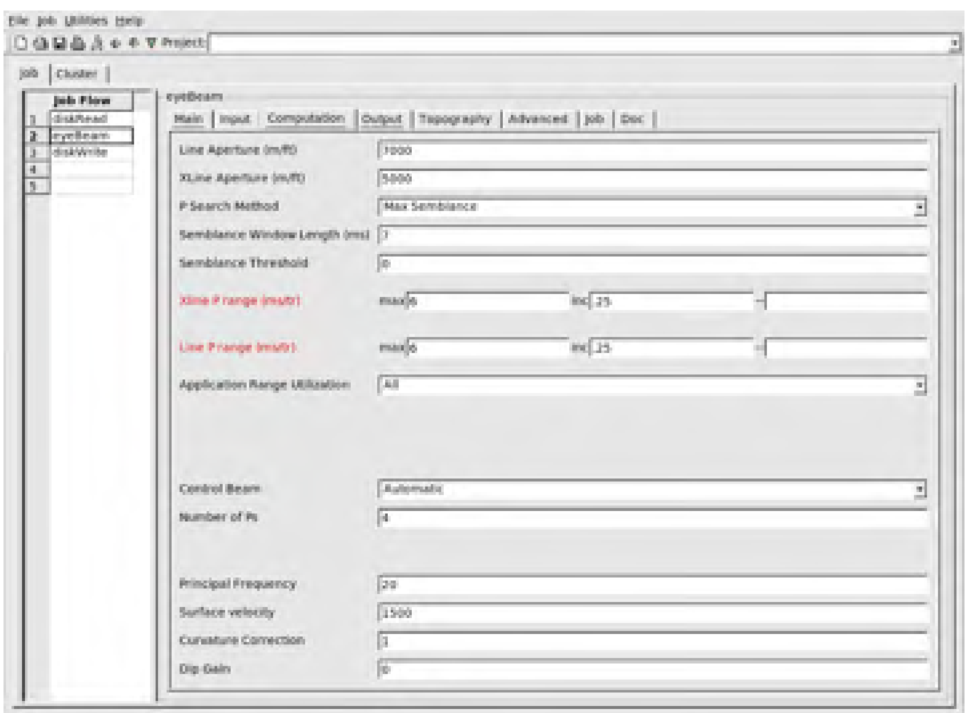


Figure 51. The eyeBeam Computation Panel, Application Range Utilization field set to Exclude

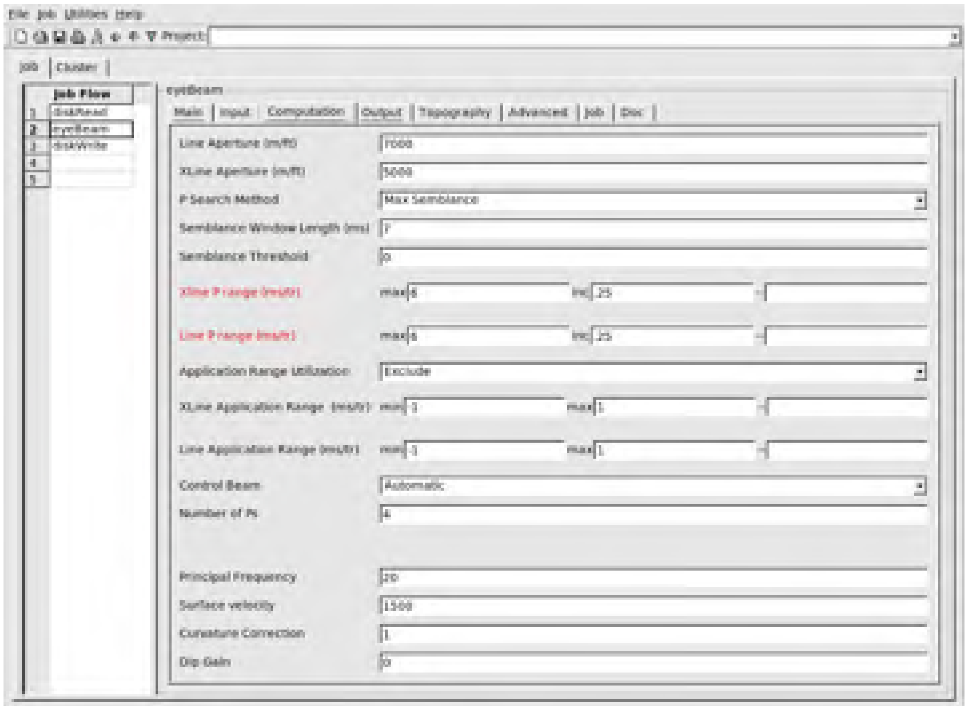


Figure 52. The eyeBeam Computation Panel, Application Range Utilization field set to Include

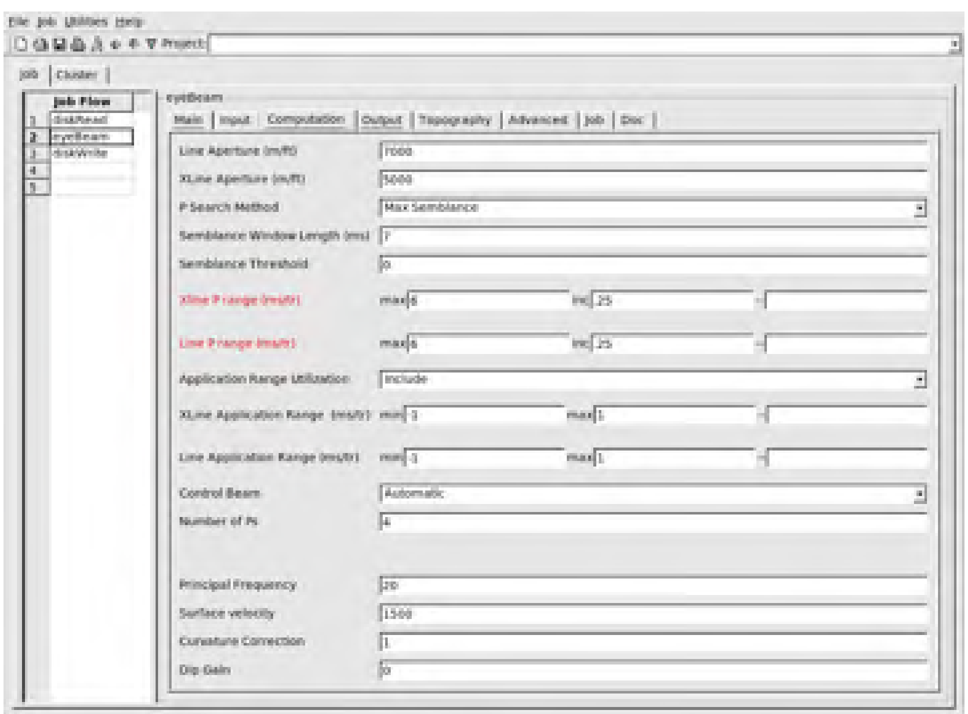


Table 5. eyeBeam Main Panel Fields

Module	Description
Line and Xline Aperture	These fields define the migration half-aperture for the migration
pSearch Method.	This field has two options. You can choose the p-values based on either the maximum semblance or maximum slant stack amplitudes. The recommendation is to set this field to Max Semblance. • When set to Max Semblance, the two parameters Semblance Window Length and Semblance Threshold appear. Semblance Window Length controls the length of the vertical window from which p-values are selected. Semblance Threshold provides a threshold for rejecting p-values whose semblance is below this value. • When set to Max Stack, p-values are determined from the slant stack volume or field. No other parameters need be set.
Xline P Range	The max value and increment. The range is the defined from negative max to max. **Note: This is a REQUIRED entry.**
Line P Range	The max value and increment. The range is then defined from negative max to max. **Note: This is a REQUIRED entry.**
Application Range Utilization	This field can be set to All, Exclude, or Include. • When set to All, the p-search is performed over the entire set of slant-stack p values. • When set to Exclude, the p-search is performed over those p-values outside the range defined by the Xline Application Range and Line Application Range fields. • When set to Include, the only allowable p-values are based on the range defined within the Xline and Line Application Ranges.
Control Beam	The three options are Fixed P Range, Automatic and No Control.
Number of Ps	The actual number of p-values to migrate when Control Beam is Automatic.
Principal Frequency	Principal Frequency is used to define the Fresnel zone.

continues on next page

Table 5. eyeBeam Main Panel Fields–continued

Tab	Description
Surface velocity	If no initial velocity volume is provided, this value defines the near surface velocity.
Curvature Correction	When set to 1, this flag causes the eyeBeam algorithm to correct for local curvature.
Dip Gain	When set, a linearly increasing scale factor will be applied to higher dips.

Figure 53 shows the eyeBeam Output Panel, which defines the range and limits of the desired output data volume. The meaning of these parameters are defined in Table 6.

Figure 53. eyeBeam Output Panel

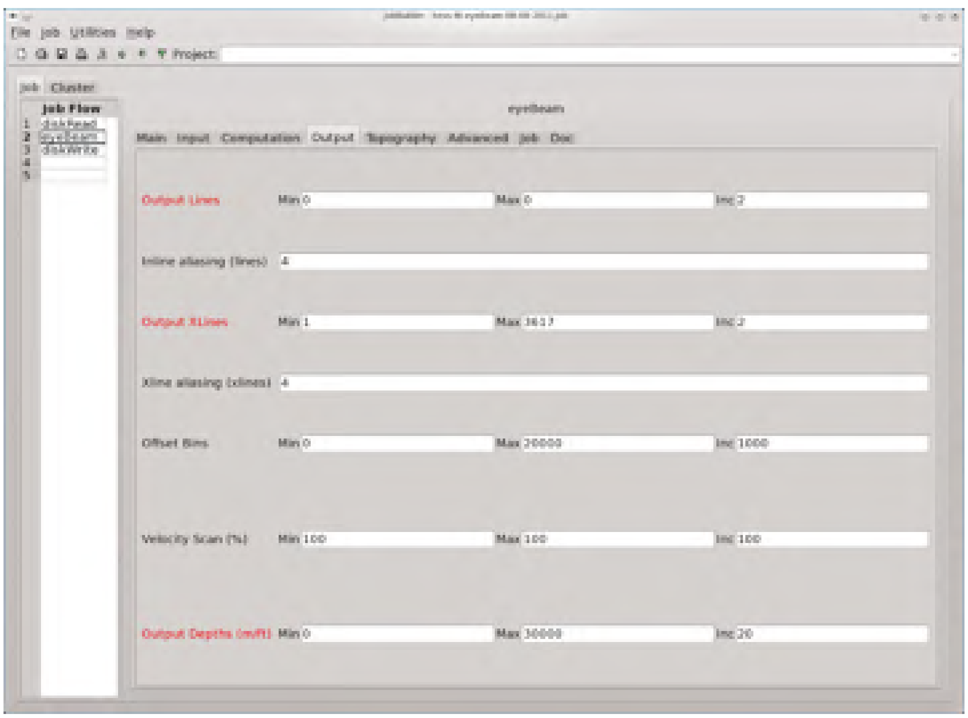


Table 6. eyeBeam Output Panel Fields

Module	Description
Output Lines	The output line range. **Note: This is a REQUIRED entry.**
Inline aliasing (lines)	Defines the desired spacing length for antialiasing. The larger this value is, the more anti-aliasing is applied.

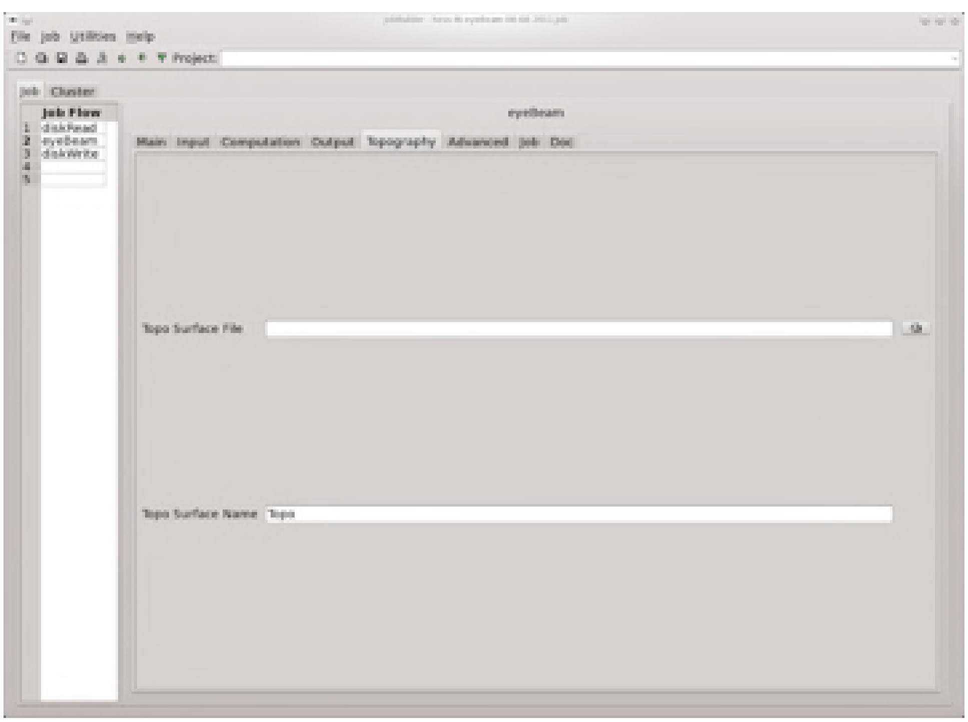
continues on next page

Table 6. eyeBeam Output Panel Fields–continued

Tab	Description
Output XLines	The output xLine range. **Note: This is a REQUIRED entry.**
Crossline aliasing (xlines)	Defines the desired spacing length for antialiasing. The larger this value is, the more anti-aliasing is applied.
Offset Bins	Defines the endpoints of the desired output offset bins. For example, the values 0, 20,000, and 1000, define output offsets ranging from 500 to 19,500.
Velocity Scan (%)	Setting the minimum, maximum, and increment provide the percentage range over which migrations should be performed. For example, setting this range to 90, 110, 5 will produce output volumes using 90, 95, 100, 105, and 110 percent of the original velocity field.
Output Depths	Defines the minimum, maximum, and increment for the each output trace. **Note: This is a REQUIRED entry.**

The eyeBeam Topography Panel fields in Figure 54 define the topographic surface file and the surface name.

Figure 54. eyeBeam Topography Panel



The eyeBeam Advanced Panel in is shown in Figure 55. The fields are described in Table 7. It is best to simply use the default values for these fields.

Figure 55. eyeBeam Advanced Panel

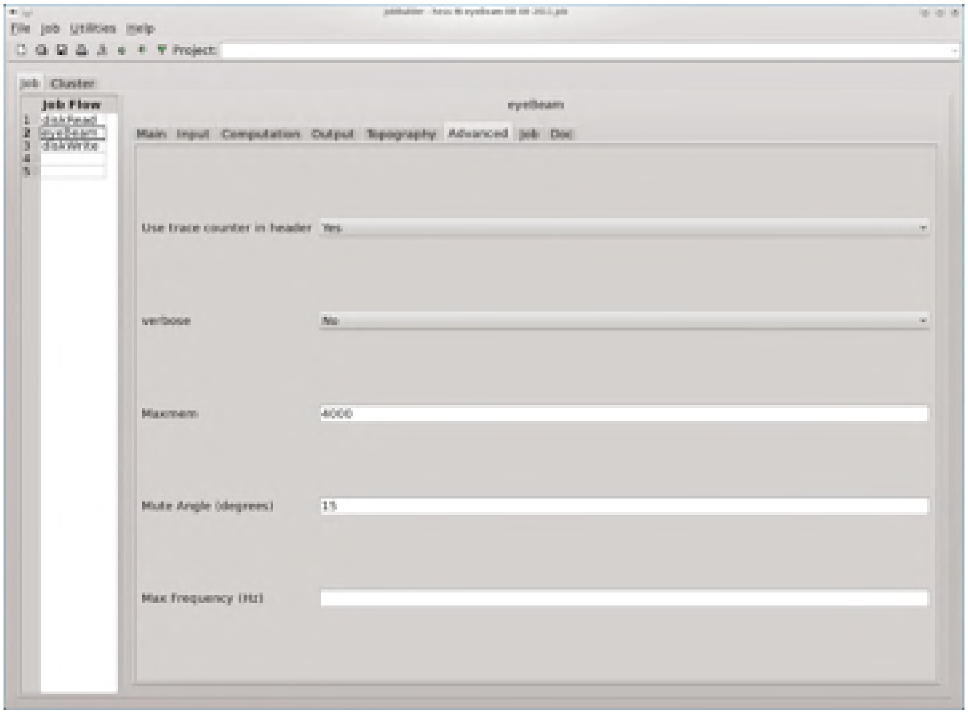


Table 7. eyeBeam Advanced Panel Fields

Module	Description
Use trace counter in header.	When set to YES, the trace counter is set.
Verbose	When set to YES, eyeBeam will print debug information to the log file.
Maxmem	The amount of memory, in megabytes, to allow for each process on each node.
Mute Angle (degrees)	The value applies an automatic mute at this angle on each output CDP and line gather.
Max Frequency (Hz)	This field applies a low-pass filter to the input with this value as the highest frequency. It is best to use the default value and avoid the extra calculations. If you want to filter the data, it is better to do so before migration.

The eyeBeam Job Panel shown in Figure 56 defines computer based parameters related to how the process generates output. The terms are defined in Table 8.

Figure 56. eyeBeam Job Panel

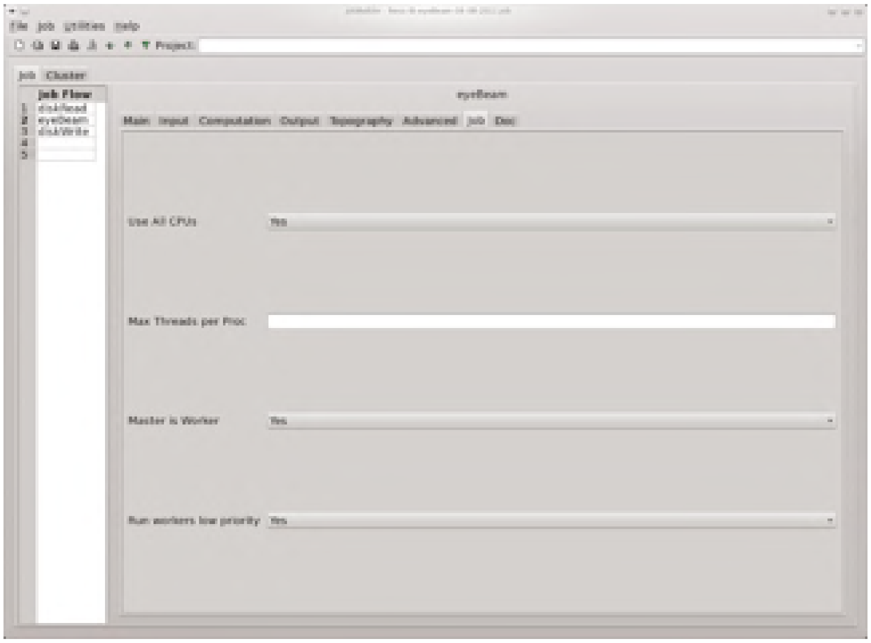


Table 8. eyeBeam Advanced Panel Fields

Module	Description
Use All CPUs	When this is set to YES, Marvel initializes eyeBeam as a single process utilizing all cores on the node. When set to NO, Marvel initializes eyeBeam as multiple processes, where each process uses a single core.
Max Threads per Proc	When Use All CPUs is set to NO, you can set this value to the number of cores per process. For example, when running on an 8 core node, setting Max Threads per Proc to 4 would cause two instances of eyeBeam slaves to each use exactly 4 cores during execution.
Master is Worker	When set to YES, the master node will also be used as a worker. In this case, the master node will have one master process and at least one slave process in execution. When set to NO, the master node does little or no actual computation. Its primary purpose is to send and receive parameters and monitor and handle input and output. The recommended setting is NO.
Run Workers Low Priority	When set to Yes, all processes, except the master, will be run in a reduced priority *nice* mode.

Index

Panorama Technologies

February 18, 2013