

# Fragmentation Library

for Split-Online 4.0



Rev. February, 2009



Digital Image Analysis Systems • Software • Service

Split Engineering LLC  
110 S. Church Avenue, Suite #8312  
Tucson, Arizona, USA 85701  
Phone: 520-327-3773  
Fax: 520-326-7532  
[www.spliteng.com](http://www.spliteng.com)

# Fragmentation Library

The Split-Fragmentation library is used to delineate particles in images of broken rock and then analyze particle size statistics. The library installation includes:



**Split Fragmentation Module** – Converts grayscale images into delineated binary images



**Fragmentation Distribution Module** – Generates distribution statistics from delineated binary images



**SplitParts Scriptable ActiveX Library** – Allows accessing intermediate data used in the fragmentation calculations

## Deprecated Modules



**Distribution Analysis Multi Module** – Generates a distribution histogram from delineated binary images



**Sieve Analysis Module** – Generates sieve series and percent-passing data from distribution histogram

## Installing the Library

Select **Fragmentation** from the component list in the Split-Online installer. The installer will automatically put the installed files in a subfolder beneath the Split-Online installation folder.

This is an extended library that requires a product **registration code** to perform the installation (see *Stamping Extended Libraries* in the Split-Online documentation). The modules in the library will operate only on a computer system that has the **dongle** for which the library is stamped. Contact Split Engineering to get the registration code for your system.

After the fragmentation library has been installed, the four new module icons shown above will appear in the Palette Window of Split-Online (the VBScript icon is already there).

## Adding the Components to a Channel

Following the drag and drop method outlined in *Adding Modules to a Channel* in the Split-Online documentation, place the Split Fragmentation module into a channel in a position that follows (to the right of) any module which creates grayscale frames, such as the Live Images module. To the right of the Split Fragmentation module, add either the Fragmentation Distribution module (recommended), or, if using the deprecated modules, add both the Distribution Analysis module and the Sieve Analysis module as shown below.



**Channel using standard fragmenting modules**



**Channel using deprecated fragmentation modules**

# Split Fragmentation



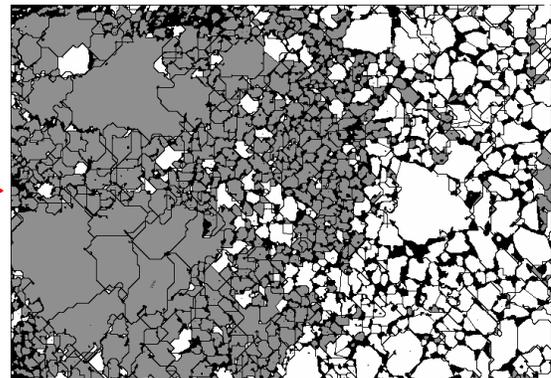
**Prerequisite:** This module must follow a source of scaled grayscale frames.

The Split Fragmentation module generates a frameset of binary, delineated frames using a frameset of grayscale frames as input. (In reality, the *binary* frames are not truly binary, but are 16-color frames. However we continue use the term *binary* to make the distinction between these frames and the grayscale frames that are used as input to this module.)

An integral part of the delineation process is the identification of **fin**es material. In the example delineation shown below, the fines material is colored gray in the delineated image.



Original grayscale image

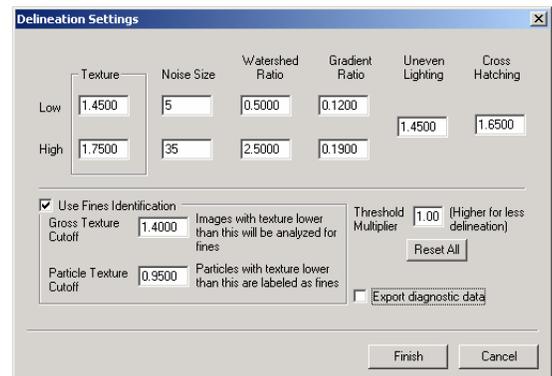


Delineated image with fines identified

This module operates on each grayscale frame independently. It starts by calculating a **texture factor** for each entire frame. The texture factor is meant to characterize the image and is subsequently used to adjust a variety of **delineation parameters**.

## Adjusting Delineations

The first step in configuring this module is to determine the range of texture factors that can be expected from the material that is being delineated. Typically this number is determined by collecting a comprehensive set of sample images that represents the full range of material that will be delineated by this module. Establish a channel that will run the sample images through this module, and then output the **SDRatio 10/5** per-frame data (contributed by this module) to a convenient device (such as the *Report* pane of the *Output Window*). The resulting texture range should be roughly equivalent to the range of reported values with *reasonable* padding to account for possible extremes that weren't reflected in the sample images.



Split Fragmentation Settings

The next step is to make adjustments to improve the delineations. Delineations are affected by adjusting the following parameters:

- **Noise Size** Black particles smaller than the noise size are ignored. The noise size is scaled between the **Low** and **High** values, based on where the measured texture factor falls within the configured range for the texture. Larger values decrease delineations.
- **Watershed Ratio** Adjusts the sensitivity of the watershed algorithm. The watershed ratio is scaled between the **Low** and **High** values, based on where the measured texture factor falls within the configured range for the texture. Larger values decrease delineations.

- **Gradient Ratio** The gradient ratio is scaled between the **Low** and **High** values, based on where the measured texture factor falls within the configured range for the texture. Larger values decrease delineations.
- **Uneven Lighting** When frame texture is below this value, special processing is employed for uneven lighting conditions. Larger values increase the use of uneven lighting, which will generally result in less delineations.
- **Cross Hatching** When frame texture is below this value, the frame is smoothed before processing. This may improve processing of poor images and generally results in decreased delineations. Smaller values increase the use of cross hatching.
- **Threshold Multiplier** This is the only parameter not tied to the texture factor, and affects how the images are thresholded during processing. Larger values decrease delineations.

## Adjusting Fines Identification

The **Gross Cutoff Texture** determines whether the frame is analyzed for fines particles. If the measured texture factor is lower than this parameter, the frame will be analyzed for fines, thus larger values increase fines identification. The **Particle Cutoff Texture** determines whether individual particles are classified as fines. If the texture factor for the entire frame is above the gross cutoff, the particle cutoff has no effect because the frame will not be analyzed for fines. If the texture factor for the entire frame is below the gross cutoff, each delineated particle in the frame is analyzed and if the particles' texture is below the particle cutoff, the particle is classified as fines. Therefore, larger values increase fine identification.

## Summary of Adjustments

### To increase delineations

- decrease noise size
- decrease watershed ratio
- decrease the gradient ratio
- increase cross hatching
- decrease uneven lighting
- decrease threshold multiplier.

### To decrease delineations

- increase noise size
- increase watershed ratio
- increase gradient ratio
- decrease cross hatching
- increase uneven lighting
- increase threshold multiplier.

### To increase fines

- increase the gross cutoff
- increase the particle cutoff

### To decrease fines

- decrease the gross cutoff
- decrease the particle cutoff

### Channel Data Produced by this Module

-  **Qualified Frame Count** – The number of valid frames delineated
-  **SDRatio 10/5** – A fuzzy number that represents the gross texture of image frames.
-  **SDRatio 10/5** – A per-frame fuzzy number that represents the gross texture the frame
-  **AutoNoiseSize** – Used internally
-  **AutoShedSize** – Used internally
-  **AutoGradRatio** – Used internally
-  **AutoEvenLight** – Used internally
-  **AutoMinXHatch** – Used internally
-  **AutoFinesID** – Used internally

Data names in **Blue** are produced only when the **Export Diagnostic Data** option is checked

### Script-Modifiable Parameters in this Module

- crosshatchthreshold** [float] – Value in the *Cross Hatching* box
- enablefinesid** [boolean] – Value of the *Use Fines Identification*. Checkbox
- gradienthigh** [float] – Value in the top (*Low*) *Gradient Ratio* box
- gradientlow** [float] – Value in the bottom (*High*) *Gradient Ratio* box
- grossfinesthreshold** [float] – Value in the *Gross Texture Cutoff* box
- noiseshigh** [float] – Value in the top (*Low*) *Noise Size* box
- noiselow** [float] – Value in the bottom (*High*) *Noise Size* box
- particlefinesthreshold** [float] – Value in the *Particle Texture Cutoff* box
- texturehigh** [float] – Value in the top (*Low*) *Texture* box
- texturelow** [float] – Value in the bottom (*High*) *Texture* box
- threshold** [float] – Value in the bottom *Threshold Multiplier* box
- unevenhreshold** [float] – Value in the *Uneven Lighting* box
- watershedhigh** [float] – Value in the top (*Low*) *Watershed Ratio* box
- watershedlow** [float] – Value in the bottom (*High*) *Watershed Ratio* box

# Fragmentation Distribution



**Prerequisite:** This module must follow the Split Fragmentation module or some other source of scaled *binary* images

This module takes the results of the Split Fragmentation module, which are essentially one or more binary delineated images (as defined above in the documentation for the Split Fragmentation module), and produces quantitative size, shape and surface area statistics on the delineated particles. This module can be run without the Split Fragmentation module in the channel as long as the channel contains a frameset of scaled binary frames.

**Note:** This module originated with the release of Split-Online 3.0. It replaces the two deprecated modules, Distribution Analysis Multi, and Sieve Analysis.

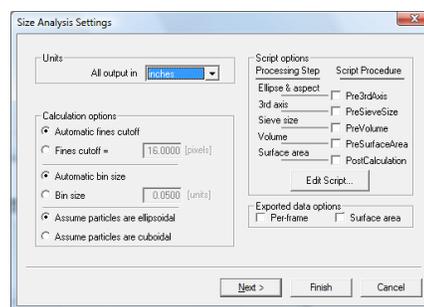
In addition to generating size statistics, the Fragmentation Distribution module is designed to help calibrate a Split-Online system when laboratory sieve results are available. The module also provides a scripting interface to allow clients to customize – or even totally replace – the particle calculations in the module.

The description below is given in two parts: first we'll cover the setup options for the module in their approximate order of appearance, and second we'll go through the recommended calibration procedure in detail.

## Setup Parameters – Page 1

### Units

Select the units to be used for reporting the size statistics. This selection is doesn't have to be the same as the units used to scale the input images.



Setup – page 1

### Calibration Options – Fines cutoff

There is a minimum particle size below which the particle delineations are mostly unreliable. Particles below this size are considered *fines*, and the fines cutoff value dictates this size in units of square pixels. Delineated particles below the fines cutoff are ignored by this module.

The optimum fines cutoff size is inexact, difficult to pinpoint, and is likely to change as the imaging environment changes, but under normal conditions it will generally be in the range of 8-16 square pixels. We recommend using the default option of *Automatic fines cutoff*, however you can select to define a static value with the *Fines cutoff = ... [pixels]* option. When using the automatic option you can calculate the selected cutoff from the exported *Fines Cutoff* value. This value is reported in the units selected in the option above, so in the unlikely event that you want the fines cutoff in square pixels rather than real units you need to multiply this value by the frame scale.

### Calibration Options – Bin size

The bin size refers to the size of divisions in the master histogram used internally by the software. Usually the bin size should be about half of the smallest precision reported in the results. For example, if you set up a sieve series (see below) with the sizes 0.10, 0.25, 0.50, and 1.0, the smallest precision in these sizes is the 1/100 place, so the bin size should be half of that, or 0.005. Larger than that and you may incur quantization errors; smaller than that and you may degrade performance with no gain in accuracy.

We recommend that you use the default option of *Automatic bin size*. With the automatic option, the software will select a bin size based on the sieve series you are using (see below). You can inspect the calculated bin size with the exported *Bin Width* value. If you have not entered a sieve series or want to specify the bin size for any other reason, use the *Bin size ... [units]* option.

### **Calibration Options – Particle shape**

The only calculated values affected by the particle shape are the surface area results. Select either the *ellipsoidal* or *cuboidal* option.

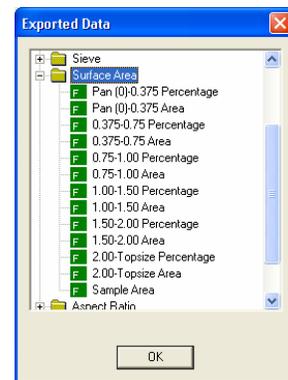
### **Exported Data Options – Per Frame**

Check the *Per-frame* box to export channel data for each frame in the input frameset. See the section on Per-Frame data in the Split-Online documentation.

### **Exported Data Options – Surface Area**

Check the *Surface Area* box to export surface area statistics. When enabled, the exported data for the module will include a *Surface Area* folder as shown here. The child items in the folder are not reported in individual sieve sizes, but in sieve *ranges* instead. A *sieve range* is just the range between two adjacent sieve sizes. For each sieve range there are two values reported: the *Percentage* and the *Area*.

**Important:** Surface area results are derived from just the delineated particles above the fines cutoff. The surface area of particles below the fines cutoff is assigned a value of zero, even if a fines estimation is in use (see below). These calculations are appropriate only for screened samples or samples with very little fines material.



**Exported surface area values**

The *Percentage* value tells us, for any sieve range, what percentage of the sample's total surface area comes from particles within that range. The sum of *Percentage* values over the entire sample should equal 100 (which is why there is no final *Sample Percentage* value, like there is a *Sample Area* value).

The *Area* value for a sieve range is the *unit surface area* of particles within that range. The final *Sample Area* value tells the unit surface area of the entire sample. The unit surface area is the (surface area) / (volume). If using units of mm, the unit surface area would be in [mm<sup>2</sup>/mm<sup>3</sup>], or [1/mm].

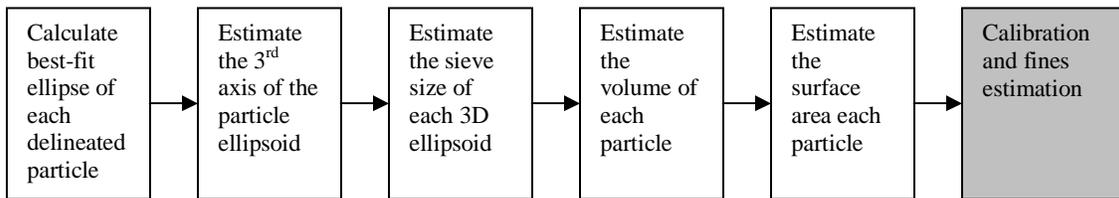
The *Area* allows you to compare two samples for relative surface area (the *Percentage* does not). Also, if you know the volume of a sample, you can get its total surface area by multiplying the *Sample Area* by the sample volume. In most installations of Split-Online it is unreasonable to assume a sample volume.

### **Setup Parameters – Scripting**

Clients with their own particle size models may want to modify the way that this module calculates particle size statistics. This module provides a method of doing that in the way of a VBScript interface. At five different points in the calculation pipeline, you can select to execute a VBScript subroutine that has access to the list of delineated particles. The subroutine can examine and change any of the particle statistics that have already been calculated by this module.

### **The Particle Pipeline**

The conversion of binary images to size statistics follows a processing pipeline similar to this:



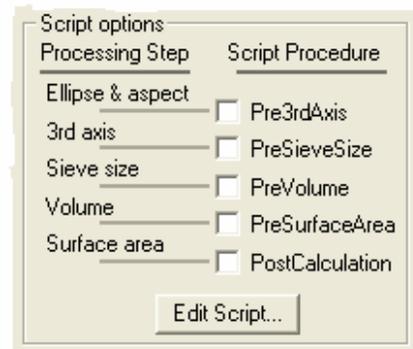
**Particle pipeline**

The first step in the pipeline produces a list of particle descriptions, and the descriptions include statistics such as those listed here. When the list is initially made in the first step, most of these statistics are undefined:

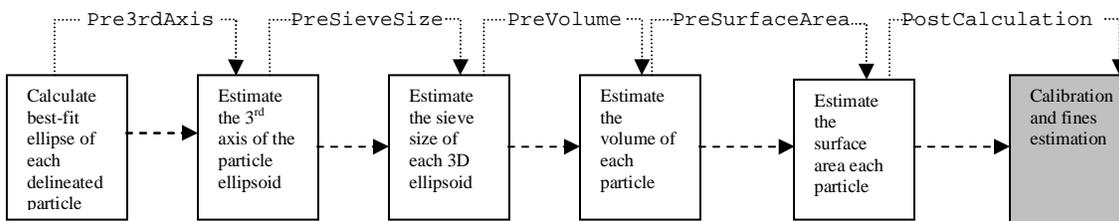
- particle location
- major and minor axis length
- axis orientation and aspect ratio
- 3<sup>rd</sup> axis – normal to the major and minor, coming out of the screen
- sieve size
- volume
- surface area

As the list of particles gets passed along to each successive step in the pipeline, the statistics are calculated and filled in for each particle’s description. By the end of the pipeline we have a full description of every delineated particle in an image.

The five VBScript subroutine points correspond to the position of the five arrows shown in the particle pipeline graphic. The interface for scripting options, shown here, attempts to illustrate this concept by listing the processing steps down the left side and showing the in-between scripting points down the right side. This is illustrated again in the particle pipeline below, showing the name and position of the VBScript subroutines.



**Scripting user interface**



**Particle pipeline showing VBScript subroutine names and positions**

**Guidelines to selecting a VBScript subroutine**

There is a significant amount of overhead required for each call into a VBScript subroutine, so remember the old adage of *use what you need, but need what you use*. Enable as few subroutines as possible to satisfy your calculation requirements and, generally, the earlier in the pipeline the better.

Scripting doesn’t disable any of the standard calculations; this module is going to go through all of its calculations regardless of which scripting subroutines have been enabled. The script subroutine should be

selected so that the module has already calculated the particle statistics that the script is going to modify. If the module hasn't already calculated the statistics, the script-modified statistics will be overwritten by the standard module calculations.

As an example, let's say that you want to replace the algorithm that calculates the sieve size of the particles. The first consideration is that we need to enter into the script after the module has calculated the sieve size, otherwise the module will overwrite our script calculations. Referring to the pipeline, the module calculates the sieve size in the third step, so our script subroutine has to execute after step three. This leaves us with the option of using one of the `PreVolume`, `PreSurfaceArea`, or `PostCalculation` subroutines. Above we stated that earlier is better, so we would select the `PreVolume` subroutine to perform our calculations.

### **Adding a VBScript subroutine**

There are two criteria that must be met for a script subroutine to get executed: first, the subroutine name must exist in the module's script code as a valid VBScript subroutine and second, the box beside the subroutine's name has to be checked under *Script Options*.

Initially this module includes skeletal script code that already has all of the subroutine names in it. You can edit this script code at any time by pressing the *Edit Script...* button. The section of the skeletal code that contains the `PreVolume` subroutine is shown in the box below.

```
Sub PreVolume(Particles)
  Dim particle
  For Each particle in Particles
    '*** TODO *** calculate any of the following particle parameters
    'particle.SieveSize = ??
    'particle.Ellipse3rd = ??
    'particle.EllipseMajor = ??
    'particle.EllipseMinor = ??
    'particle.EllipseAngle = ??
    'particle.EllipseX = ??
    'particle.EllipseY = ??
  Next ' particle
End Sub
```

### **The *PreVolume* skeletal script code**

Each of the five subroutines has the same initial skeletal format shown in the script above. The subroutine takes one argument, *Particles*, which is the list of delineated particles. Inside the subroutine is a *For Each* loop that iterates over every particle object in the list. Particle objects are described in detail below.

The properties of the particle objects that have already been calculated are indicated in the comments of each skeletal subroutine. (Not listed are the read-only properties of the particles. These are always available but cannot be changed.) Continuing the example we started above, we might change the `PreVolume` subroutine as shown here. We're replacing this module's calculation of the particle *SieveSize* with our own equation, which just computes the average of the major and 3<sup>rd</sup> axes. (We don't mean to suggest that this is a valid method of calculating the sieve size, it's used only as an example of how to modify a particle's parameter.)

```

Sub PreVolume(Particles)
  Dim particle
  For Each particle in Particles
    particle.SieveSize=(particle.EllipseMajor+particle.Ellipse3rd)/2
  Next ' particle
End Sub

```

### Our modified *PreVolume* subroutine

You are free to modify the skeletal code in any way that you want, as long as it is valid VBScript code. If there are any errors in the script code, the code will silently fail to run. In the nonsensical example code below, we use a conditional *If* block to help calculate the *SieveSize* of each particle. Particles oriented between 0 and 90 degrees have their *SieveSize* doubled, and all others have their *SieveSize* halved.

```

Sub PreVolume(Particles)
  Dim particle
  For Each particle in Particles
    If particle.EllipseAngle > 90 Then
      particle.SieveSize = particle.SieveSize * 2
    Else
      particle.SieveSize = particle.SieveSize / 2
    End If
  Next ' particle
End Sub

```

### A silly example of a modified *SieveSize* calculation

#### The Particle Object

Particle definitions are passed to and from script using a VBScript *object* to represent each particle. The particle object has no methods, but it does have the properties listed below. The following abbreviations are used in the descriptions of the particle properties:

BFE – Best fit ellipse

<sup>RO</sup> – Read Only

Some properties are in pixels and are noted; otherwise, the properties are in the units defined by the *Units* option of this module. Where applicable, the origin of the frame's coordinate system is the upper left.

As with all VBScript objects and keywords, spelling counts but case does not. In the example code we preserve case to improve readability, but it isn't necessary.

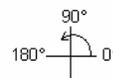
**PixelCount**<sup>RO</sup> – Number of pixels in delineated particle

**DelineatedArea**<sup>RO</sup> –Area of delineated particle (not the best fit ellipse) in *units*

**EllipseMajor** – Length of the BFE major axis

**EllipseMinor** – Length of the BFE minor axis

**EllipseAngle** – Angle of the major axis of the BFE, in degrees. Zero degrees is to the right and 180° is to the left as shown.



**EllipseX** –X coordinate of the center of the BFE

**EllipseY** –Y coordinate of the center of the BFE

**EllipseArea**<sup>RO</sup> – Area of the BFE

**MaterialType** – 0 means particle is rock, 1 means particle is a patch of fines material

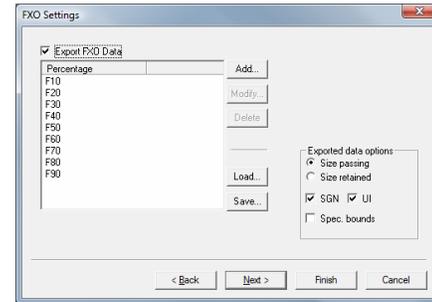
**Ellipse3rd** – Length of the 3D ellipsoid's 3<sup>rd</sup> axis, which is normal to the plane of the image

**SieveSize** – The sieve size of the 3D particle, which is the smallest opening through which the particle can be passed at any orientation

**Volume** – Volume of the particle, in *units*<sup>3</sup>  
**SurfaceArea** – Surface area of the particle, in *units*<sup>2</sup>

## Setup Parameters – Page 2, FXO Settings

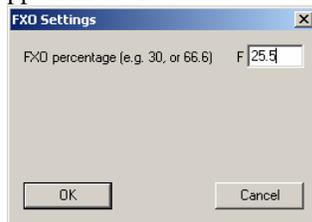
The second page of the module's setup defines the *FXO* statistics that will be exported by the module. *FXO* statistics are the inverse of sieve results: for a given percentage passing (or retained), the *FXO* result is the particle size. Where a sieve tells you a percentage that is a given size, the *FXO* tells you a size that is a given percentage. When *FXO* exports are enabled, the module will always export a *Topsize* value along with the other *FXO* data.



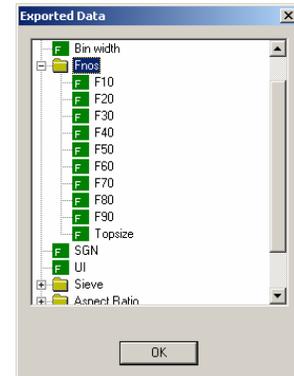
Setup page 2, FXO settings

### Export FXO Data

This check box enables the *FXO* calculations and exports the results. When checked, the module will export a folder titled *Fnos* as shown to the right. Within the *Fnos* folder is the always-exported *Topsize* value plus any percentages that appear in the list.



Entering an *FXO* value



Exported *FXO* data

### The Percentage List

The list box below the *Export FXO Data* option lists the series of percentages for which *FXO* will be calculated. To modify the series, select the *Custom Series* option below the list and use the *Add...*, *Modify...*, and *Delete* buttons to the right of the box to modify the list. By convention, *FXO* percentages always begin with an *F*, followed by the numeric percentage. When specifying an *FXO* percentage, enter the numeric value of the percentage without the leading *F* as shown in the example above where 25.5% is the percentage. The module doesn't allow you to enter duplicate values, and it will automatically sort the series when you modify a percentage. Use the *Save* button to store the series as an *fxo* file, and *Load* to recall previously saved series.

The entire string that you use to specify a percentage is retained and used as the name of the exported data. If you use F10.00 instead of just F10, for example, the exported value will have the name *F10.00*.

### Exported Data Options - Passing vs. Retained

You have the option of reporting *FXO* values in either *Size passing* or *Size retained* by selecting the appropriate option on the setup page. For a given percentage, say 90%, the *Size passing* option reports the statistical particle size that is larger than exactly 90% of the sample, and *Size retained* reports the particle size that is smaller than 90% of the sample.

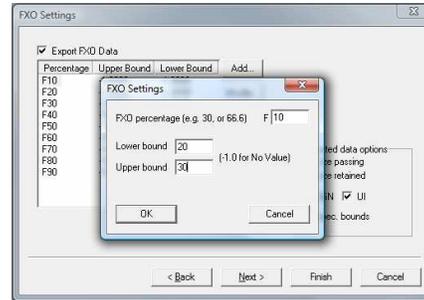
For percentage P%,  $Size\ of\ P\% \ passing = Size\ of\ (100 - P\%) \ retained$   
 $Size\ of\ P\% \ retained = Size\ of\ (100 - P\%) \ passing$

### **Exported Data Options – SGN and UI**

The Size Guide Number *SGN* is simply the size of 50% passing – which is the same as the size of 50% retained. It is a standard measure in some industries so it is included as an export option. The Uniformity Index *UI* is the ratio of (95% retained) / (10% retained), or (5% passing) / (90 % passing). The numerator will always be a size less than or equal to the denominator, so the UI is always between 0 and 1. The closer to 1, the more uniform the size distribution.

### **Exported Data Options – Spec. Bounds**

Turning on the *Spec. Bounds* options enables you to enter upper and lower specification bounds for each percentage in the FXO series. When enabled, two additional columns appear in the list: the *Upper Bound* and the *Lower Bound* columns. Also, the interface for editing a percentage will have two additional entries: the *Upper bound* and the *Lower bound*, as shown here.



### **Entering an FXO value with *Spec. Bounds* enabled**

The specification bounds are currently used on two occasions. 1) to assign a true or false value to the *Within Bounds* exported value and 2) for automatically drawing boundary lines on trend plots or X-Y plots.

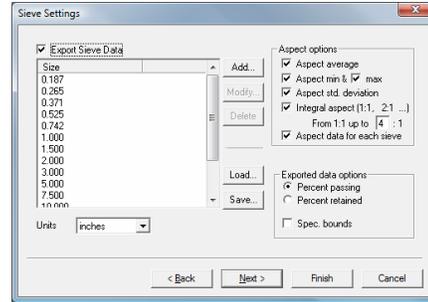
When *Spec. Bounds* is enabled for either the FXO series or for the sieve series (see below), this module exports a boolean value *Within Bounds*. *Within Bounds* is true if all of the items, including all FXO percentages and all sieve sizes, are within their upper and lower bounds; otherwise it will be false. The exception to this is when a bounds value is set to -1, it is ignored and doesn't enter into the calculation of *Within Bounds*.

The trend plots and X-Y plots in Split-Online support an *Auto Bounds* feature. When this feature is enabled on a plot, the plot will automatically draw upper and lower boundaries, based on the boundary values that have been entered here.

**Note:** When you switch between the *Size passing* and *Size retained* options, the *Spec. Bounds* values do not automatically change so they will no longer be valid.

## Setup Parameters – Page 3, Sieve Settings

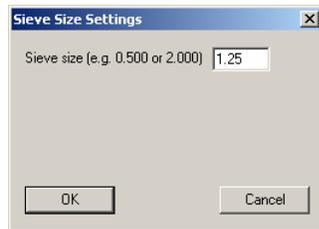
The third page of the module's setup defines the *Sieve* statistics that will be exported by the module. The sieve statistics are modeled after a physical set of sieve screens; you establish a set of virtual sieves and this module will calculate the fraction of sample that passes (or is retained by) each sieve size. Unlike a real sieve set, however, you can define any number of virtual sieves and there is no limit on their size.



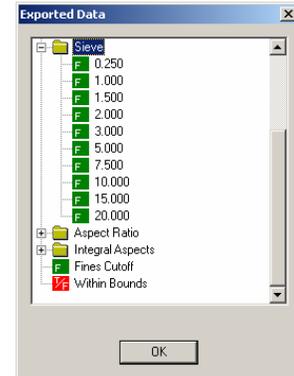
Setup page 3, Sieve settings

## Export Sieve Data

This check box enables the sieve calculations and exports the results. When checked, the module will export a folder titled *Sieve* as shown to the right. Within the *Sieve* folder is the series of virtual sieve sizes that have been configured on this page.



Entering an sieve value



Exported sieve data

## The Size List

The list box below the *Export Sieve Data* option lists the series of sieve sizes for which percent passing will be calculated. To modify the series, select the *Custom Series* option below the list and use the *Add...*, *Modify...*, and *Delete* buttons to the right of the box to modify the list. In the example above, the sieve size 1.25 is being added. The module doesn't allow you to enter duplicate values, and it will automatically sort the series after any modification. Use the *Save* button to save the series as an *siv* file, and *Load* to recall previously saved series.

The entire string that you use to specify a percentage is retained and used as the name of the exported data. If you use 1.2500 instead of just 1.25, for example, the exported value will have the name *1.2500*.

## Exported data options - Passing vs. Retained

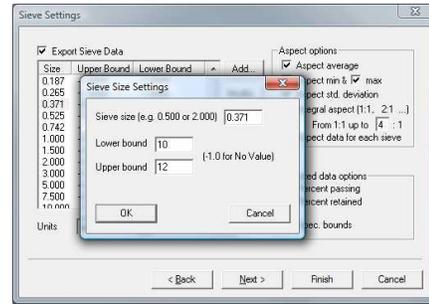
You have the option of reporting sieve values in either *Percent passing* or *Percent retained* by selecting the appropriate option on the setup page. For a given size, say 1.00", the *Percent passing* option reports the percentage of the sample that is smaller than 1.00", and *Percent retained* reports the percentage of the sample that is larger than 1.00".

$$\text{For a given size } S, \text{ Percent passing } S = (100 - \text{Percent retained } S)$$

$$\text{Percent retained } S = (100 - \text{Percent passing } S)$$

### Exported data options – Spec. Bounds

Turning on the *Spec. Bounds* option enables you to enter upper and lower specification bounds for each size in the sieve series. When enabled, two additional columns appear in the list: the *Upper Bound* and the *Lower Bound* columns. Also, the interface for editing a size will have two additional entries: the *Upper bound* and the *Lower bound*, as shown here.



**Entering a sieve value with *Spec. Bounds* enabled**

The specification bounds are currently used on two occasions. 1) to assign a true or false value to the *Within Bounds* exported value and 2) for automatically drawing boundary lines on trend plots or X-Y plots.

When *Spec. Bounds* is enabled for either the sieve series or for the FXO series (see above), this module exports a Boolean value *Within Bounds*. *Within Bounds* is true if all of the items, including all sieve sizes and all FXO percentages, are within their upper and lower bounds. The exception to this is when a bounds value is set to -1, it is ignored and doesn't enter into the calculation of *Within Bounds*.

The trend plots and X-Y plots in Split-Online support an *Auto Bounds* feature. When this feature is enabled on a plot, the plot will automatically draw upper and lower boundaries, based on the boundary values that have been entered here.

**Note:** When you switch between the *Percent passing* and *Percent retained* options, the *Spec. Bounds* values do not automatically change so they will no longer be valid.

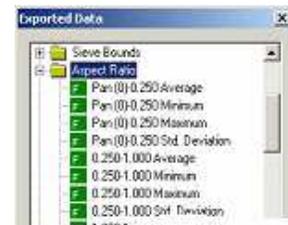
### Setup Parameters – Aspect Options

Aspect calculations are enabled only when the sieve calculations are enabled. There is no mathematic reason for this; it is done out of convenience and will likely change in future releases of this library.

**Important:** Aspect results are derived from just the delineated particles above the fines cutoff. The aspect statistics of particles below the fines cutoff is assigned a value of zero, even if a fines estimation is in use (see below).

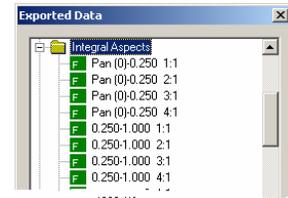
### Export folder names

There are two different folders used to export aspect information. The standard statistics for the average, min, max and standard deviation of aspect ratios is exported through a folder named *Aspect Ratio*. When any of these options are checked, the *Aspect Ratio* folder is added to this module's output and the selected statistics will appear as child items in the folder. The child items are not reported in individual sieve sizes, but in sieve *ranges* instead. A *sieve range* is just the range between two adjacent sieve sizes.



***Aspect Ratio* folder**

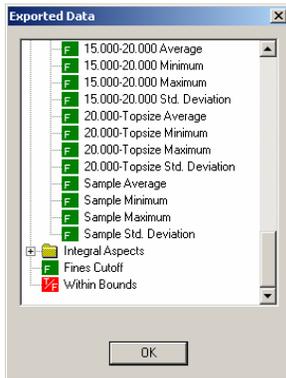
When the *Integral Aspect...* option is checked, this module exports a folder titled *Integral Aspects*. As with the *Aspect Ratio* folder, the child items in this folder are reported in sieve ranges, not sieve sizes.



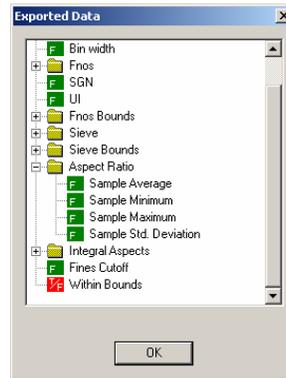
**Integral Aspects folder**

**Per-sieve results**

The *Aspect data for each sieve* checkbox lets you control whether aspect results are reported for each sieve range plus the entire sample, or for just the entire sample. When checked, every aspect statistic that is enabled is reported for each sieve range, plus for the entire sample. When unchecked, all enabled aspect statistics are reported for just the entire sample, and not for the individual sieve ranges.



**Aspect data for each sieve enabled**



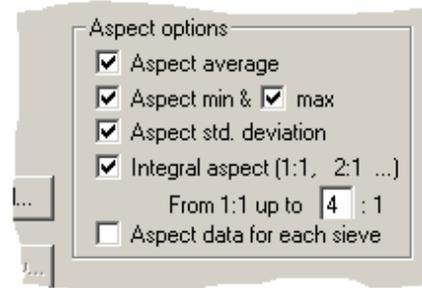
**Aspect data for each sieve disabled**

**Aspect average, min, max and standard deviation**

These are standard statistics and warrant little discussion. They are reported under the *Aspect Ratio* folder, and keep in mind that they are measured only on delineated particles larger than the fines cutoff value.

**Integral Aspect**

Integral aspects are percentages of particles whose aspect falls within an integral range. For example, a sample might have 70% of its particles within the 1:1 aspect range, 20% within the 2:1 range, 5% within the 3:1 range, and so on. The integers are the highlighted values in the ratios. The number of integral ratios reported depends on the value entered into the box *From 1:1 up to ... :1*. In the illustration this value is 4, so the percentage would be reported for the ratios 1:1, 2:1, 3:1, and 4:1.



**Integral aspect detail**

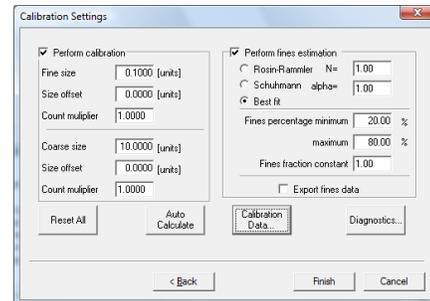
Aspect ratios are not rounded up to the next integral range. For example, a particle with an aspect of 1.999 : 1, it is still considered to be within the 1:1 range.

## Setup Parameters – Page 4, Calibration Settings

The fourth and final page of the module's setup is where we enter the calibration information. This is usually done with the *Calibration mode* option enabled. The calibration process is described cookbook fashion in the next section.

### Commentary on Calibrations

Calibrating this module is the process of reconciling the difference between the calculated size distribution curve and the size distribution *lab results* measured by hand in a sieve test. Calibration can't be performed without at least one lab sample.



Setup page 4, Calibration settings

Calibration is almost always required because of the three main sources of error:

- The mass and sieve size of irregularly shaped rocks is difficult to accurately measure from 2 dimensional images. Assumptions about the 3<sup>rd</sup> axis have to be made, and particles often overlap.
- There is almost always some fraction of the sample that cannot be seen in the images because it is buried beneath the top surface
- The *finest fraction* of a sample cannot be delineated, therefore it cannot be measured

To perform the best calibration, our lab results should be able to tell us:

- the error in our sieve size calculations
- the error in our volume (hence, weight) calculations
- the *nature* of the fines distribution – meaning the shape of the distribution curve at the fines end

If we can measure these three factors then we should be able to perform an acceptable calibration of this module and have it produce robust and reliable size distributions. In reality, we rarely have the luxury of a direct measurement of these factors; instead we are usually given a few data points that make up the lab results size distribution. Through the cookbook calibration process outlined in the next section we implicitly extract these factors from the lab results and apply them to our calibration.

### When to Calibrate

Any major change in either the Split-Online configuration or in the *imaging environment* warrants a calibration. The imaging environment includes the camera zoom and focus, light intensity and location, and even the material that is being measured. Ideally, the Split Fragmentation module should first be adjusted after any environmental change, and then this module should be recalibrated.

### Perform calibration

This section controls the corrections made to particle size and volume calculations. When the checkbox is not checked, no corrections are made to the delineated particle sizes.

The calibration process works on the assumption that errors in finer particles are different than the errors in coarser particles, and that the differences between fine and coarse are linear, or close enough to linear that using a linear approximation is sufficient. Based on this assumption, you are given two calibration points: one for the fine end and one for the coarse end, and the software will interpolate in between.

The *Fine size* and *Coarse size* parameters define the low and high calibration points. Other than that the fine size has to be smaller than the coarse size, your selection of these two points is up to you, however you may find benefit from selecting two sizes that correspond to sieve sizes in the lab results. Usually the fine size should be at or above the fines cutoff. If the lab results include screen sizes below the fines cutoff, we recommend setting the *Fine size* to either the fines cutoff or the smallest screen size that is larger than the fines cutoff. The *Coarse size* should usually be the largest screen size or the actual material topsize, if it is known.

### **Size offsets**

Each calibration point has a *Size offset* parameter, which is a positive or negative bias in particle size. This parameter is meant to address the error in sieve size of delineated particles. If you enter a size offset of +1[unit], delineated particles will have their sieve size adjusted by +1[unit].

The *Size offset* parameter is best used when you are able to compare to lab results of a screened sample that has very small size variation. For example, if your *Fine size* is 1", you could collect the retained material from a 1" sieve screen and place it into the normal imaging environment and capture images of the material. With all calibration parameters at their default, nominal values, you can compare the difference between the lab results and this module's output. Since we know that this experiment has effectively zero error from either the hidden material problem or from fines estimation, nearly all of the error must be from the sieve size measurements.

If you aren't able to perform such a calibration process for the *Size offset*, you still might be able to make reasonable adjustments if you can be fairly certain of the smallest and largest sizes in an imaged sample, such as in a screened sample. If this is not possible, we advise leaving the *Size offset* parameter at zero.

### **Count multipliers**

Both calibration points also have a *Count multiplier* parameter which address the hidden material source of error. Generally, smaller particles are under-represented in an image because they tend to get hidden through natural sorting. In a typical calibration, the *Count multiplier* of the *Coarse size* is kept at one and the *Count multiplier* of the *Fine size* is increased.

Note that the net effect of the *Count multipliers* is the ratio of the values at the coarse size and the fine size. Setting both to 5, for example, is the same as setting both to 1, or to 0.5, or any other value because the ratio between them is still 1:1. Similarly, if the value at the fine size is set to 10 and at the coarse size to 0.1 (ratio = 100:1), this is the same as using 100 and 1 instead.

### **Perform fines estimation**

The *Perform fines estimation* checkbox turns on or off the estimation of sample volume below the fines cutoff value. Generally this option should be enabled because even though it is only an estimate, it will improve the distribution statistics of finer material in almost every case. When you are sampling screened material with no fines fraction you might get better results with this option off.

Fines estimation is done by fitting one of two curves to the measured size distribution and extrapolating the curve into the fines realm. Using industry standards as a guide, we offer two types of curves for fines estimation: the *Rosin-Rammler* and the *Schuhmann*. You can pick one of these two curve types, or we recommend that you pick the *Best fit* option and use whichever curve has the best fit with the measured data.

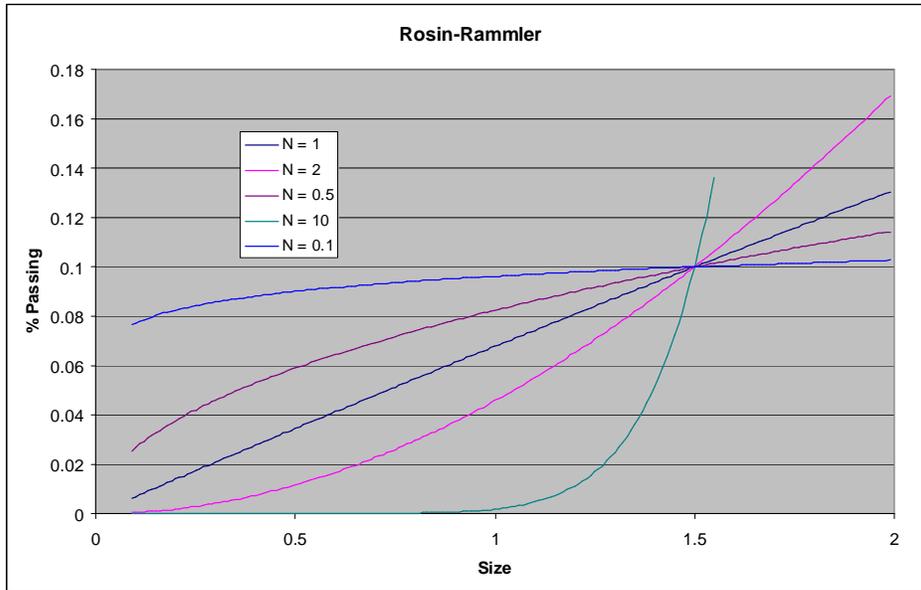
The Rosin-Rammler function is given by:

$$\rho_1 = 1 - e^{-\ln(1-\rho_0) \left(\frac{x_1}{x_0}\right)^n}$$

$x_0$  = Characteristic size, the fines cutoff in our case

$x_1$  = target size – what is the fraction passing at this size?  
 $\rho_0$  = Fraction passing at size  $x_0$  (known)  
 $\rho_1$  = Fraction passing at size  $x_1$  (unknown)  
 $n$  = Rosin Rammler uniformity coefficient

The Rosin-Rammler uniformity coefficient  $n$  defines the general shape of the curve. The fines estimation method works on the assumption that  $n$  is constant at a given site, and in calibration mode, the software will calculate the ideal value of  $n$  based on the lab results. Some examples of Rosin-Rammler curves are shown here at various values of  $n$  and a fines cutoff ( $x_0$ ) of 1.5 and  $\rho_0$  of 0.1.



The Schuhmann function is given by the equation:

$$\rho_1 = \left( \frac{x_1}{k} \right)^\alpha$$

$x_1$  = target size – what is the fraction passing at this size?  
 $\rho_1$  = Fraction passing at size  $x_1$  (unknown)  
 $\alpha$  = Schuhmann slope, given by:

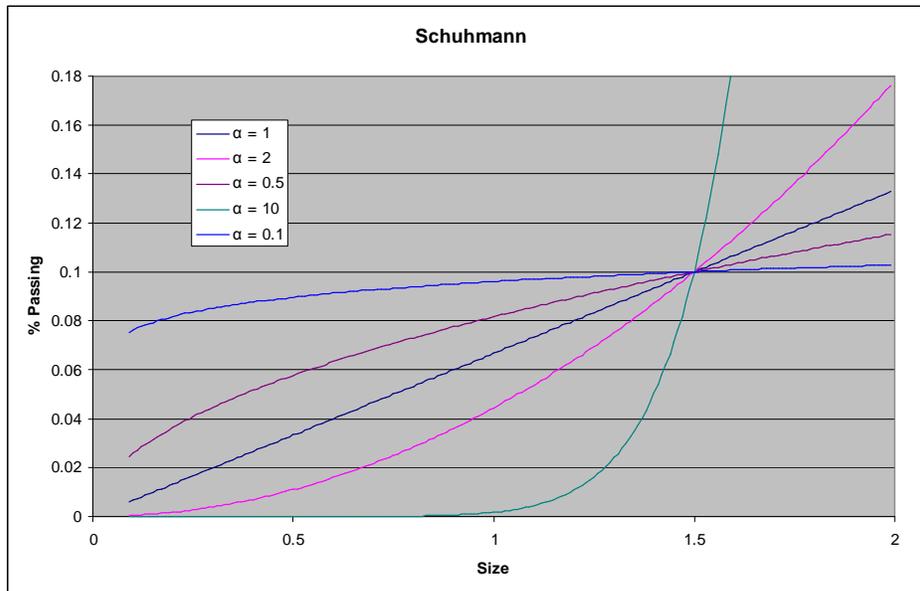
$$\alpha = \frac{\ln\left(\frac{\rho_1}{\rho_0}\right)}{\ln\left(\frac{x_1}{x_0}\right)}$$

$x_0$  = Characteristic size, the fines cutoff in our case  
 $\rho_0$  = Fraction passing at size  $x_0$  (known)

Through rearrangement we can express the constant  $k$  as:

$$k = e^{\left(\ln(x_1) - \left(\frac{\ln(\rho_1)}{\alpha}\right)\right)}$$

which makes  $\alpha$  the only coefficient in the equation. As with the Rosin-Rammler coefficient, the fines estimation method works on the assumption that the Schuhmann slope  $\alpha$  is constant at a given site, and in calibration mode, the software will calculate the best value of  $\alpha$  based on the lab results. Some examples of Schuhmann curves are shown here at various values of  $\alpha$ , a fines cutoff of 1.5 and  $\rho_0 = 0.1$ .



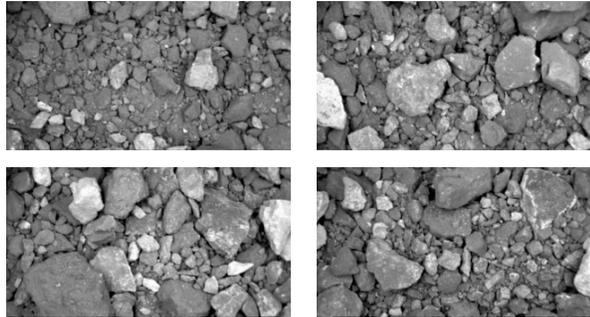
### **Comparing Rosin-Rammler and Schuhmann**

Using the *Best fit* option of fines estimation will cause the module to calculate which curve is the best fit for the measured, delineated particles and apply the best-fit curve to the fines estimation. As you can see from the curves in the two graphs above, there is very little difference between the two.

## Calibration – Step-by-Step Instructions

### (1) Collect sample images and lab results

A full discussion of the proper field methods for collecting a calibration sample is beyond the scope of these instructions. In the following steps it is assumed that you have performed an accurate sieve test on a sufficiently large sample, and that you are using one or more images representative of that sample to calibrate this module.



Sample  
2004031-2

Screen	Lab
5.000	98.70
4.000	95.02
3.000	84.18
2.000	71.50
1.000	53.63
0.500	37.66
0.375	32.79
0.250	25.63

Collect images and lab results

### (2) Decide whether to use pre-delineated images

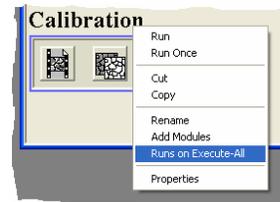
Once you have collected the grayscale images, there are two options for how you will use them to perform calibrations:

- Option A – Delineate them once and save the delineated binary images (this was the only option in previous versions of Split-Online)
- Option B – Re-delineate the grayscale images each time you perform a calibration step. **This is the recommended method** because even though it is slightly slower, it re-processes the images through all of the modules in the channel each time, ensuring that changes made to other modules are reflected in the calibration data.

**If you chose option A above, follow the steps 3a-3d. If you chose option B, skip to step 4a.**

### (3a) Set up a calibration channel

It is usually easiest to make a new channel to be used just for calibration. If you disable the *Runs on Execute-All* option of the channel you can just leave the channel in the project. The first module should be a File Images module which opens the grayscale or color images that correspond to the sieve lab results. The second module will probably be the Split Fragmentation module.

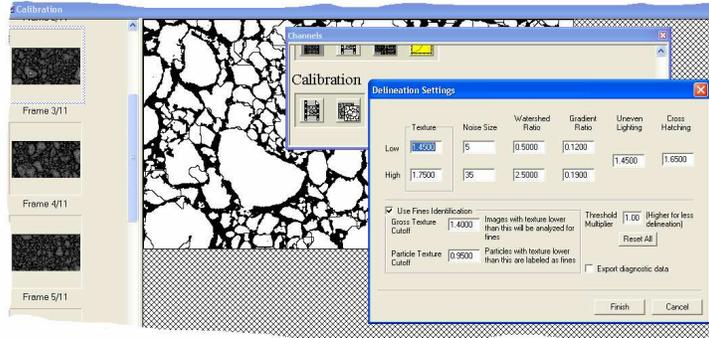


Calibration channel

Alternatively, you can avoid creating a separate channel by putting both a *Live Images* module and a *File Images* module in the same channel. In normal operation, disable the *File Images* module and enable the *Live Images* module; when generating delineations for calibration, disable the *Live Images* module and enable the *File Images* module.

### **(3b) Get good delineations**

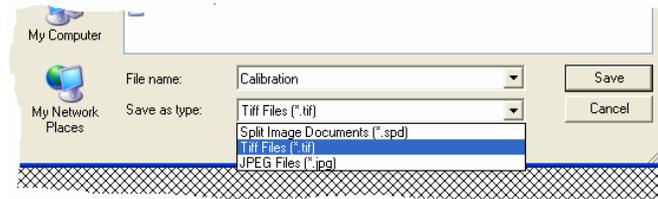
Using the documentation in this manual for the Split Fragmentation module, configure the module to provide the best delineation as possible, possibly including fines identification. Do not use the lab results to gauge the delineations; use only your eyes and your judgment.



**Get good delineations from Split Fragmentation**

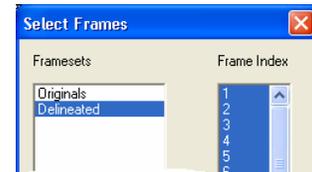
### **(3c) Save the delineated binary frames**

When the Split Fragmentation module is configured to your satisfaction, allow it to delineate all of the calibration images and then save the delineation results. From the menu, select *File / Save As...* and, for convenience, save the delineated images in the same directory as the original images. **Important: Save the images in *tiff* format, not *jpeg*!** Jpeg format is a *lossy* format, and the binary delineated images saved in jpeg format will not work correctly.



**Save delineated files in *tiff* format**

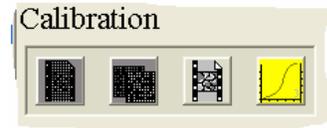
After you OK the *Save As* dialog you will see the *Select Frames* box as shown here. Pick just the *Delineated* frameset from the list of framesets and then OK.



**Save just the delineated frames**

### **(3d) Prepare channel for the size distribution module**

We are done with the delineation step so, unless we make any modifications to the sample images or to the Split Fragmentation settings, we will deal only with the delineated binary images from this point forward. We recommend keeping the current modules in the calibration channel, but disabling them so they don't execute. Then, add a new File Images module to read in the delineated images saved in the previous step, and add a Fragmentation Distribution module. **Important:** change the frameset name from *Originals* to *Delineated* in the File Images module.



**New modules**



**New frameset name in File Images module**

**Steps 4a-4c are only if you chose option B in step 2.**

**(4a) Put a *File Images* module in the channel**

The intent is to set up the channel to run with *either* live video or images from the calibration files, so put a module for each in the same channel. One of the modules will always be disabled.



**Channel has two modules that produce images**

**(4b) Temporarily disable the *Live Images* module**

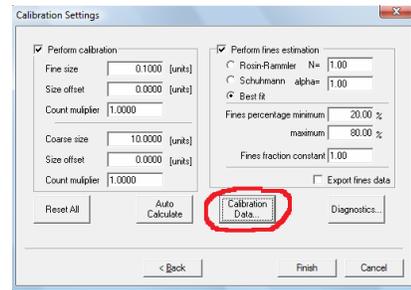
For the purpose of calibration, disable the *Live Images* module and enable the *File Images* module. Don't forget to reverse that when calibration is completed!

**(4c) Populate the *File Images* module with the grayscale calibration images**

Use all of the images from step (1) above. Ideally, the images will have their scale embedded because this allows you to change the scale and still use older calibration images.

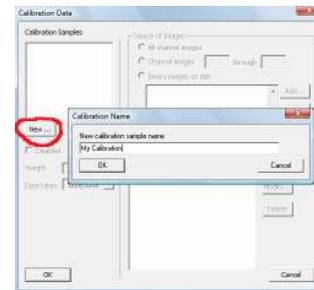
**(5) Enter the calibration information**

Advance to the 4th page of the module's settings and press the *Calibration Data* button, shown here. Enable *Calibration mode* on the first page of this module's settings. Advance to the *Sieve Settings* page and make sure that the sieve series is the same as that used in the lab results. Enter the results of the lab's sieve test as the *Calibration standard* for the sieve series.



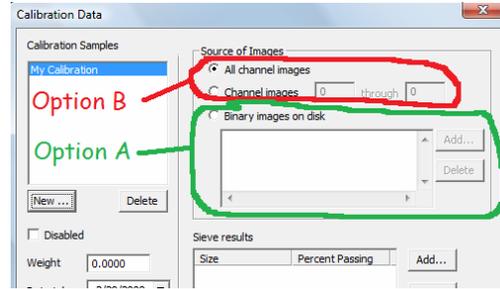
**Calibration Data button**

Split-Online supports any number of *Calibration Samples*. A calibration sample is comprised of the images and lab results as explained in step (1) above. To add a new sample, press the *New* button and enter a name for the sample, as shown here. The *Weight* and *Date Taken* fields are optional. If a weight other than zero is specified, the sample is assigned that weight in the final calibration parameters. For example, with two calibration samples, if you give the first a weight of 1 and the second a weight of 2, the second sample will have twice as much effect on the final calibration. The date is for reference only and has no effect on the results.



**Add a new calibration sample**

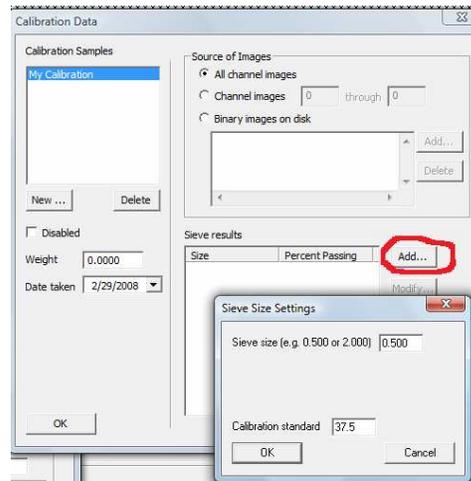
In step 2 above, you decided whether to pre-delineate your images and save the delineated binary images to disk, or to re-delineate the grayscale images as needed. That decision affects which of the options you choose here. If you went with option A then select the third choice, *Binary images on Disk* and use the *Add* button to browse for the binary delineated images.



Select images from the channel or from disk

If you selected option B then select either the first choice, *All channel images*, or the second choice *Channel images xx through xx*. The difference between these two is that *All Channel Images* will use every image processed by the channel for this calibration set, and the second choice lets you dictate that only some of the images processed by the channel apply to this calibration set. The two numbers associated with the second choice refer to the image order in the *File Images* module.

For each calibration set, you need to enter the results of the lab sample sieve results. For each sieve size in the results, hit the *Add* button as shown here, and enter the Sieve size (in the same units as specified on the first page of the module's setup) and the percent passing or retained for that size.

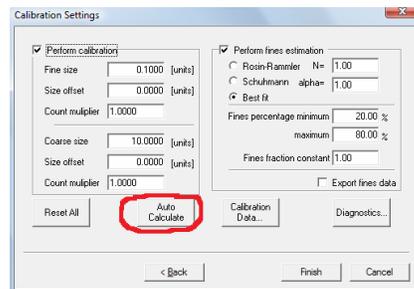


Enter the lab results for each sieve size

**(6) Calculate the calibration parameters**

On the 4th page of the module's setup, press the *Auto Calculate* button shown here. After a slight pause, Split-Online will fill in all of the calibration fields with calculated values.

If you are using option B from step 2 for any of the calibration sets, Split-Online will automatically run the channel that contains the module. This generates the binary delineations for calibration.

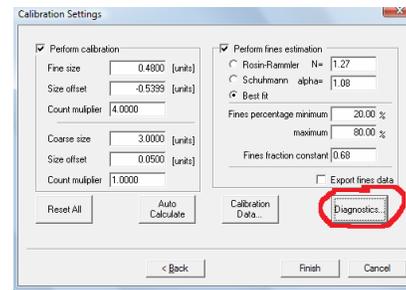


Calculate calibration parameters

There may be times you want to manually adjust the calibration parameters instead of using the fully automatic method. See the section *Special instructions for manually adjusting calibration parameters* on page 25 for instructions.

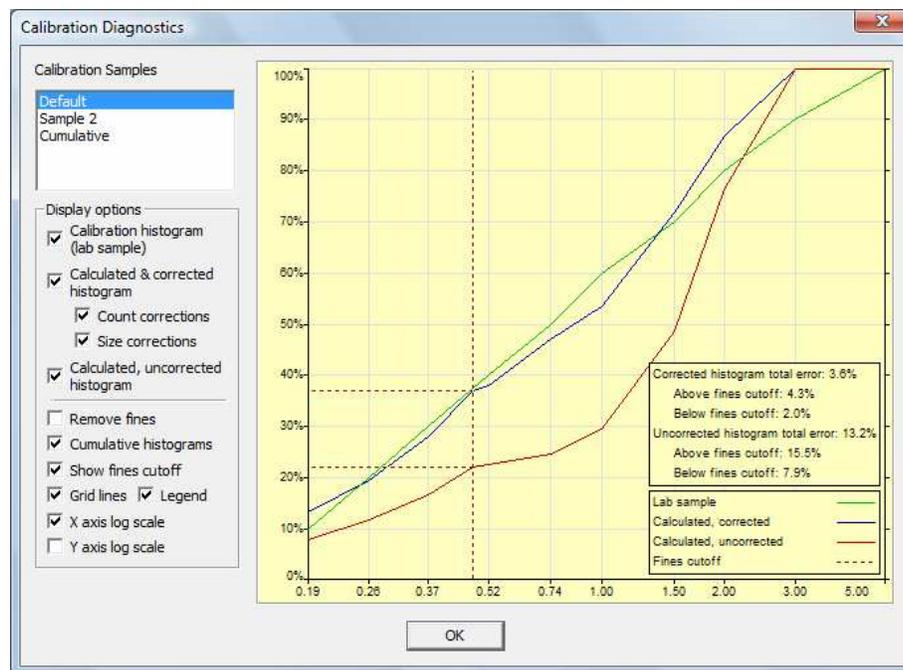
### (7) Inspect the final results

Hit the *Diagnostics* button as shown here. Split-Online will execute the channel if necessary to generate binary delineations.



### Show diagnostics

The large graph on this window shows information about the calibration set that is selected in the *Calibration Samples* box. This box contains the name of each calibration set from step 5, except those that are disabled. If there is more than one enabled calibration set, Split-Online will add the *Cumulative* set to the list. The cumulative set is a calculated combination of all of the other enabled calibration sets.



### Diagnostics graph

Display options on the diagnostics window are explained below.

- **Calibration histogram (lab sample)** – Toggles display of the measured sieve values entered in step 5.
- **Calculated & corrected histogram** – Toggles display of the calibrated size distribution histogram that Split-Online generates for the calibration images. The two checkboxes beneath this option let you inspect the effects of the multiplier and offset calibration parameters by removing them from the calibration. When both are un-checked, this histogram is identical to the uncorrected histogram (below).
  - **Count corrections** – Uncheck to remove the *Count multiplier* from the corrected calibration
  - **Size corrections** – Uncheck to remove the *Size offset* from the corrected calibration
- **Calculated, uncorrected histogram** – Toggles display of the uncalibrated size distribution that Split-Online generates for the calibration images.
- **Remove Fines** – Removes the fines fraction from all histograms and recalculates the cumulative curves. The calibration of the Split-Online generated histogram is performed differently for particles below the fines cutoff, and this option lets you inspect just the above-fines fraction of all histograms..
- **Cumulative histograms** – Uncheck this to show normal, non-accumulated histograms. This is useful for diagnosing large errors between the lab sample and the Split-Online histograms.
- **Show fines cutoff** – Toggles display of the fines cutoff indicators
- **Grid lines** – Toggles display of grid lines on the graph
- **Legend** – Toggles display of the legend and histogram error information on the graph
- **X axis log scale** – Check for log scale on the X axis, uncheck for linear
- **Y axis log scale** – Check for log scale on the Y axis, uncheck for linear

## Special instructions for manually adjusting calibration parameters

These instructions are for if you chose to manually modify the calibration parameters rather than rely on the automatically calculated values. This process replaces or augments step 6 above.

1. Calculate the *Fine size*. You can do this by either running the Auto Calculate once and letting it fill in the value, or by running the Diagnostics and recording the fine size from the legend.
2. Manually modify the calibration parameters on the left side of the page.
3. This step adjusts the *Fines factor constant* to match your manual modifications. Un-check the *Perform calibration* box as shown here, and run the *Auto Calculate* button. Note: the *Auto Calculate* function is significantly faster when you have un-checked the *Perform calibration* box.

When the *Auto Calculate* function completes, the fines estimation constants will be updated.



**Uncheck the Perform calibration box**

4. Re-check the *Perform calibration* box and inspect the results (normally done by running the Diagnostics feature)

5. If you need to modify any of the calibration parameters again, return to step 2 above. Otherwise you are done.

## Channel Data Produced by this Module

-  **Distribution size units** – Enumerated distance units of results: 0 = m, 2 = mm, 8 = in. 9 =- ft
-  **Size units name** – The name of the size units of results, such as “meters”
-  **Bin width** – The width of the internal histogram bins, in size units
-  **Fines cutoff** – Fines cutoff size, in size units
-  **Fnos**<sup>1</sup> – The FXO series
  -  **(FXO percent)** – One entry for each percentage in the FXO series
-  **Fnos**<sup>1, F</sup> – Per-frame FXO series
  -  **(FXO percent)** – One entry for each percentage in the FXO series
-  **Fnos Bounds**<sup>1, 5</sup> – The FXO series min and max specification boundaries
  -  **(FXO percent) upper** – Upper bound, one entry for each percentage in the FXO series
  -  **(FXO percent) lower** – Lower bound, one entry for each percentage in the FXO series
-  **Sieve**<sup>2</sup> – The sieve series
  -  **(sieve size)** – One entry for each size in the sieve series.
-  **Sieve**<sup>2, F</sup> – Per-frame sieve series
  -  **(sieve size)** – One entry for each size in the size series.
-  **Sieve Bounds**<sup>2, 5</sup> – The sieve series min and max specification boundaries
  -  **(sieve size) upper** – Upper bound, one entry for each percentage in the sieve series
  -  **(sieve size) lower** – Lower bound, one entry for each percentage in the sieve series
-  **SGN**<sup>3</sup> – Size Guide Number, in size units
-  **SGN**<sup>3, F</sup> – Per-frame Size Guide Number, in size units
-  **UI**<sup>4</sup> – Uniformity Index (no units)
-  **UI**<sup>4, F</sup> – Per-frame Uniformity Index (no units)
-  **Within Bounds**<sup>5</sup> – True when all bounds-enabled data are within the min and max bounds

- 1 Exported only when *Export FXO Data* enabled
- 2 Exported only when *Export Sieve Data* enabled
- 3 Exported only when *SGN* enabled
- 4 Exported only when *UI* enabled
- 5 Exported only when *Spec. Bounds* enabled
- F Exported only when *Per-frame* enabled

(Continued...)

Channel Data Produced by this Module

(...continued)

-  **Surface Area**<sup>6</sup> – Surface area statistics
  -  **(sieve range) Percentage** – % of total sample surface area in this sieve range
  -  **(sieve range) Area** – Unit area of this sieve range (units<sup>2</sup>/units<sup>3</sup>)
  -  **Sample Area** – Unit area of the entire sample (units<sup>2</sup>/units<sup>3</sup>)
-  **Aspect Ratio**<sup>7</sup> – Aspect ratio statistics
  -  **(sieve range) Average**<sup>A, S</sup> – Average aspect of delineated particles in this sieve range
  -  **(sieve range) Minimum**<sup>B, S</sup> – Minimum aspect of delineated particles in this sieve range
  -  **(sieve range) Maximum**<sup>C, S</sup> – Maximum aspect of delineated particles in this sieve range
  -  **(sieve range) Std. Deviation**<sup>D, S</sup> – Standard deviation of particle aspects in this sieve range
  -  **Sample Average**<sup>A</sup> – Average aspect of all delineated particles in this sample
  -  **Sample Minimum**<sup>B</sup> – Minimum aspect of all delineated particles in this sample
  -  **Sample Maximum**<sup>C</sup> – Maximum aspect of all delineated particles in this sample
  -  **Sample Std. Deviation**<sup>D</sup> – Standard deviation of particle aspects in this sample
-  **Integral Aspects**<sup>8</sup> – Aspect ratio statistics
  -  **(sieve range) N: 1**<sup>S</sup> – Percentage of N:1 particles in this sieve range
  -  **Sample N: 1** – Percentage of N:1 particles in the entire sample
-  **Fines Fit Coefficient**<sup>9</sup> – The least-squares correlation coefficient of the fit of the fines curve
-  **Fines Fit Coefficient**<sup>9, F</sup> – Per-frame least-squares correlation coefficient of the fit of the fines curve
-  **Fines Fit Type**<sup>9</sup> – The name of the best-fit fines curve: “Schumann” or “Rosin-Rammler”
-  **Fines Fit Type**<sup>9, F</sup> – Per-frame name of the best-fit fines curve
-  **Fines Fraction**<sup>9</sup> – Fraction of material calculated at the fines cutoff size
-  **Fines Fraction**<sup>9, F</sup> – Per-frame fraction of material calculated at the fines cutoff size

6 Exported only when *Surface area* enabled

7 Exported only when any of the following standard aspect statistics are enabled:

A – Exported only when *Aspect Average* enabled

B – Exported only when *Aspect min* enabled

C – Exported only when *Aspect max* enabled

D – Exported only when *Aspect std. deviation* enabled

S Exported only when *Aspect data for each sieve* enabled

8 Exported only when *Integral aspect* enabled

9 Exported only when *Export fines data* enabled

F Exported only when *Per-frame* enabled

# Distribution Analysis Multi



**Prerequisite:** This module must follow the Split Fragmentation module

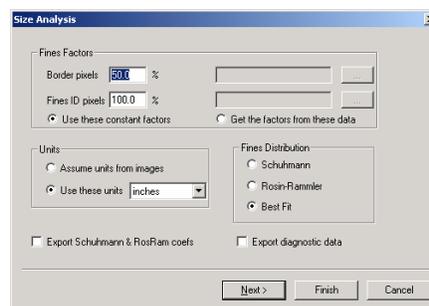
**Note:** This module has been replaced by the Fragmentation Distribution module described above.

This module is an intermediate step between the Split-Fragmentation module and the Sieve Analysis module. It takes a frameset of binary delineated frames as input and generates a size distribution histogram. This histogram is used internally by the Sieve Analysis module and isn't available for use by any of the other modules.

## Fines Factors

The fines estimation parameters are at the top of the first page of the setup wizard as shown. For each frame processed, the amount of estimated fines in the frame is affected by the area of *border pixels* (shadows and edges in the binary frames) and by the area of *fines ID pixels* (patches identified as fines by the Split-Fragmentation module).

By adjusting the fines factors for each of these pixel types, you can modify the fine-end of the resulting distribution histogram. The minimum value of 0% results in the coarsest distribution. There is no maximum value for either pixel type, but generally it shouldn't exceed 150 to 200%, and it should be that high only when there is reason to suspect that the fraction of fines beneath the visible surface is greater than the visible fraction.



**Size Analysis: First Page**

Instead of using constant fines factors, you can extract the fines factors from the channel's runtime data. Select the **Get the factors from these data** option and use the two browse buttons to select channel data for each pixel type.

## Units

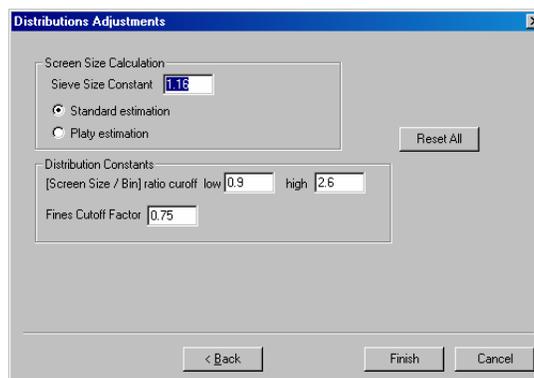
The units option refers to the length units used in the distribution histogram. The choice of units has no effect on the final results and is available here only for diagnostics. If the **Assume units from images** option is selected, the scaling units of the first frame processed are used.

## Fines Distribution

This allows you to force the use of one of the two curve fitting models in the fines calculation. In the absence of some compelling reason to do otherwise, leave this option on **Best Fit**.

## Distribution Adjustments

Options on the second page of the wizard let you modify the way that the size distribution is calculated for the delineated particles. Even minor changes to these parameters can have a significant impact on the size distribution, therefore they shouldn't be modified except at the direction of Split Engineering.



**Size Analysis: Second Page**

## Exported Data

The complexity of the data exports is due to our support of legacy systems that may be using deprecated data. The two checkboxes at the bottom of the first page of the setup wizard control how much data this module exports. They should be left *unchecked* unless your installation is using any of the deprecated data.

**Channel Data Produced by this Module**

-  **SizeAnalysis Distance Units** – Enumerated distance units used in size analysis calculation
-  **Size Analysis Composite** – Composite statistical data totaled or averaged for all frames.
  -  **Best Fit** – Simple reflection of whether the Best Fit option was selected on the first page of the wizard.
  -  **Bin Count** – The number of bins in the distribution histogram.
  -  **Bin Width** – The width of each histogram bin, in the units selected from the first page of the wizard.
  -  **Best Fit Type** – Either “RosRam” or “Schuhmann”
  -  **Number Fine Bins** – Used internally
  -  **Border Fines Factor** – The fines factor that was used, as configured on the first page of the wizard.
  -  **FinesID Factor** – The Fines ID factor that was used, as configured on the first page of the wizard.
  -  **Fit Correlation** – The correlation of the fit of whichever fines distribution method (either RosRam or Schuhmann) was used.
-  **Schuhmann** – Statistics of Schuhmann fit
  -  **X50** – Fit coefficient
  -  **N** – Fit coefficient
  -  **R2** – Fit correlation
-  **RosRam** – Statistics of Rosin-Rammler fit
  -  **X50** – Fit coefficient
  -  **N** – Fit coefficient
  -  **R2** – Fit correlation
-  **Size Analysis** – Per-frame statistics
  -  **Std Dev. Ratio** – Unused
  -  **Particle Count** – Number of particles in delineated frame
  -  **Fines Cutoff** – Minimum discernable particle size. Less than this size is considered fines.
  -  **Number Fine Bins** – Number of histogram bins below the fines cutoff
  -  **Fit Correlation** – The correlation of the fit of whichever fines distribution method (either RosRam or Schuhmann) was used.
  -  **Best Fit Type** – Either “RosRam” or “Schuhmann”
  -  **Black Pixels** – Number of black (edge and shadow) pixels in the frame
  -  **White Pixels** – Number of white (particle) pixels in the frame
  -  **Gray Pixels** – Number of gray (fines-ID) pixels in the frame
-  **Fit Coefficients** – Curve fit statistics
  -  **Schuhmann** – Statistics of Schuhmann fit
    -  **X50** – Fit coefficient
    -  **N** – Fit coefficient
    -  **R2** – Fit correlation
  -  **RosRam** – Statistics of Rosin-Rammler fit
    -  **X50** – Fit coefficient
    -  **N** – Fit coefficient
    -  **R2** – Fit correlation

Data names in **Blue** are produced only when the **Export Diagnostic Data** option is checked  
Data names in **Orange** are produced only when the **Export Schuhmann & RosRam Coefs** option is checked

# Sieve Analysis

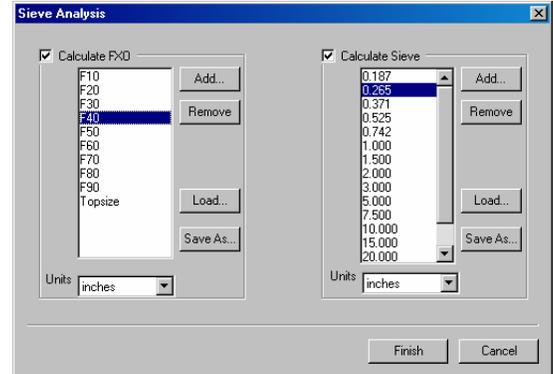


**Prerequisite:** This module must follow the distribution analysis.

**Note:** This module has been replaced by the Fragmentation Distribution module described above.

The Sieve Analysis module takes the distribution histogram produced by the Distribution Analysis module and calculates a **sieve** series and/or a percent-passing series (**FXO** series) for the distribution.

Each series is independently configurable through the setup wizard. Use the Calculate...checkboxes at the top to turn off the series calculation. The Units dropdown box below each list lets you select the units for each series.



Sieve Analysis Setup

## Modifying the Series

Use the Add... and Remove buttons beside each list to modify the members of the series. When adding new items, bear in mind that the value typed in is preserved *verbatim* to assign a name to the item. The two examples shown below, for instance, will both result in a new sieve item of 1/2 inch, but the first one will be named 0.5 and the second will be named 0.500000.



Two identical sieve items with different names

## Loading and Saving

Sieve and FXO series can be persisted to and restored from disk with the Load... and Save As... buttons. The effect of loading a series is immediate; the file is not referenced while the channel is running.

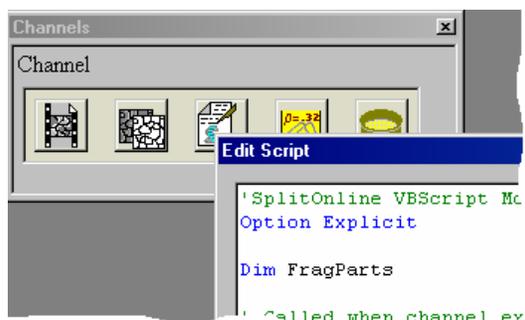
Channel Data Produced by this Module

- Fnos** – The composite FXO series configured in the setup wizard
  - F (FXO percent)** – One entry for each percent in the FXO series.
- Sieve** – The sieve series configured in the setup wizard
  - F (sieve size)** – One entry for each size in the size series.
- Fnos** – Per-frame FXO series configured in the setup wizard
  - F (FXO percent)** – One entry for each percent in the FXO series.
- Sieve** – Per-frame sieve series configured in the setup wizard
  - F (sieve size)** – One entry for each size in the size series.
- Fnos Units** – Enumerated units of Fnos data
- Sieve Units** – Enumerated units of Sieve data

## SplitParts Script Interface

The fragmentation library installation includes a scriptable ActiveX library *SplitParts*. The SplitParts library allows you to gain access to the lower level, intermediate size calculation data that isn't available otherwise. A full working example of how to use the SplitParts library is shown at the end of this section.

The script module that references the SplitParts library should be located in the channel after the fragmentation module. It may appear either before or after the Distribution Analysis module, but be aware that the Distribution Analysis modifies some of the particle information that is provided by this library, as noted in the descriptions below. To avoid confusion and for consistency, we recommend that the script module be located as shown in the graphic here, after the Fragmentation module and before the Distribution Analysis module.



Script Module in a Channel

## Object Reference

---

### **Object Name:** FragParts

Encapsulates the collection of fragmentation particle information for the channel. The FragParts object can only be created via the CreateObject command (VBScript) as in the following example:

```
Dim FragParts
Sub OnChannelExecute
    Set FragParts = CreateObject("SplitParts.FragParts")
    FragParts.Setchannel Channel ' Always do this!
End Sub
```

### Creating the FragParts object

### **Methods:**

#### **SetChannel**(*channel*)

Mandatory initialization. No return value. As shown in the sample code above and below, always initialize the FragParts object with the current channel, using the **Channel** keyword added to the scripting language by Split-Online.

#### **GetParticleCount**( [*nFrame*] )

Returns the integer number of particles available for the 0-based frame # *nFrame*. If the optional *nFrame* argument isn't given, the default 0 is used. If *nFrame* is not a valid frame index the channel terminates with a stop exception.

#### **GetParticle**(*nParticle* [, *nFrame*])

Returns a FragParticle object with index *nParticle*. The 0-based particle index *nParticle* should not exceed the value returned from **GetParticleCount** or the channel will terminate with a stop exception. If the optional *nFrame* argument isn't given the default of 0 is used. If *nFrame* is not a valid frame index the channel terminates with a stop exception.

---

---

**Object Name:** FragParticle

Represents a single delineated particle. Created from the **GetParticle** method of a FragParts object. The Fragmentation module delineates particles from a binary image and then fits an ideal ellipse to each particle. This object exposes both the raw particle information and information about the fitted ellipse. See the script code below for an example of how to create and use this object.

**Methods:**

**PixelCount**

Returns the integer pixel count for the delineated particle (not the fitted ellipse).

**Area**

Returns the floating-point area of the delineated particle (not the fitted ellipse). If this script occurs before the Distribution Analysis module then the area is in square pixels; if this script occurs after the Distribution Analysis module then the area is in the units used by the Distribution Analysis module.

**Texture**

Returns the floating-point texture value for the particle. This is the value that is used to identify the particle as fines or rock.

**IsFine**

Returns **true** if the particle is considered *fines* based on its texture, **false** otherwise.

**RectTop, RectBottom, RectLeft, RectRight**

These return the pixel (un-scaled) coordinates of the particle's bounding rectangle.

**EllipseX, EllipseY**

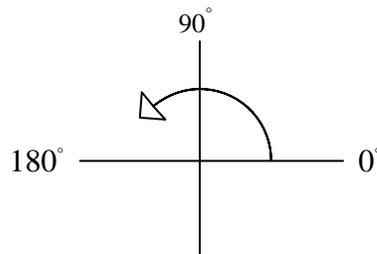
These return the floating-point coordinates of the center of the fitted ellipse. If this script occurs before the Distribution Analysis module then these values are un-scaled and in pixels; if this script occurs after the Distribution Analysis module then these values are scaled and in the units used by the Distribution Analysis module.

**EllipseMajor, EllipseMinor**

Return the floating-point length of the major and minor axes of the fitted ellipse. If this script occurs before the Distribution Analysis module then these values are un-scaled and in pixels; if this script occurs after the Distribution Analysis module then these values are scaled and in the units used by the Distribution Analysis module.

**EllipseAngle**

Returns the angle of the major axis of the fitted ellipse, in degrees. Zero degrees is to the right and 180° is to the left as shown below.



```

Option Explicit
Dim FragParts

' Called when channel execution begins
Sub OnChannelExecute
    Set FragParts = CreateObject("SplitParts.FragParts")
    FragParts.Setchannel Channel ' Always do this!
End Sub

' Called when channel execution terminates
Sub OnChannelTerminate
    Set FragParts = Nothing
End Sub

' Called when this module is executed in each channel iteration
Sub OnIteration
    Dim frameSet, frame, f, scale
    Dim partCount, particle, p

    ' Get the original grayscale frameset.
    Set frameSet = Channel.GetFrameSet("Originals")

    ' Look at each frame
    For f = 0 To (frameSet.GetFrameCount - 1)
        Set frame = frameSet.GetFrame(f)
        scale = frame.GetScale ' always returned in pixels per inch

        ' Square the scale to convert the area from square pixels to
        ' square inches
        Dim scaleDiv
        scaleDiv = scale * scale

        ' Get the number of particles delineated in this frame
        partCount = FragParts.GetParticleCount(f)

        ' Look at each particle
        Dim areaSum, pixelSum
        areaSum = 0
        For p = 0 To (partCount - 1)
            Set particle = FragParts.GetParticle(p, f)
            areaSum = areaSum + particle.Area / scaleDiv
            Set particle = Nothing
        Next 'p
        SplitApp.Log "Frame " & f & ", avg particle area = " &
        CSng(areaSum / partCount)& " sq. in."
        Set frame = Nothing
    Next ' f
    Set frameSet = Nothing
End Sub

```

**Sample Code: Using the SplitParts library**