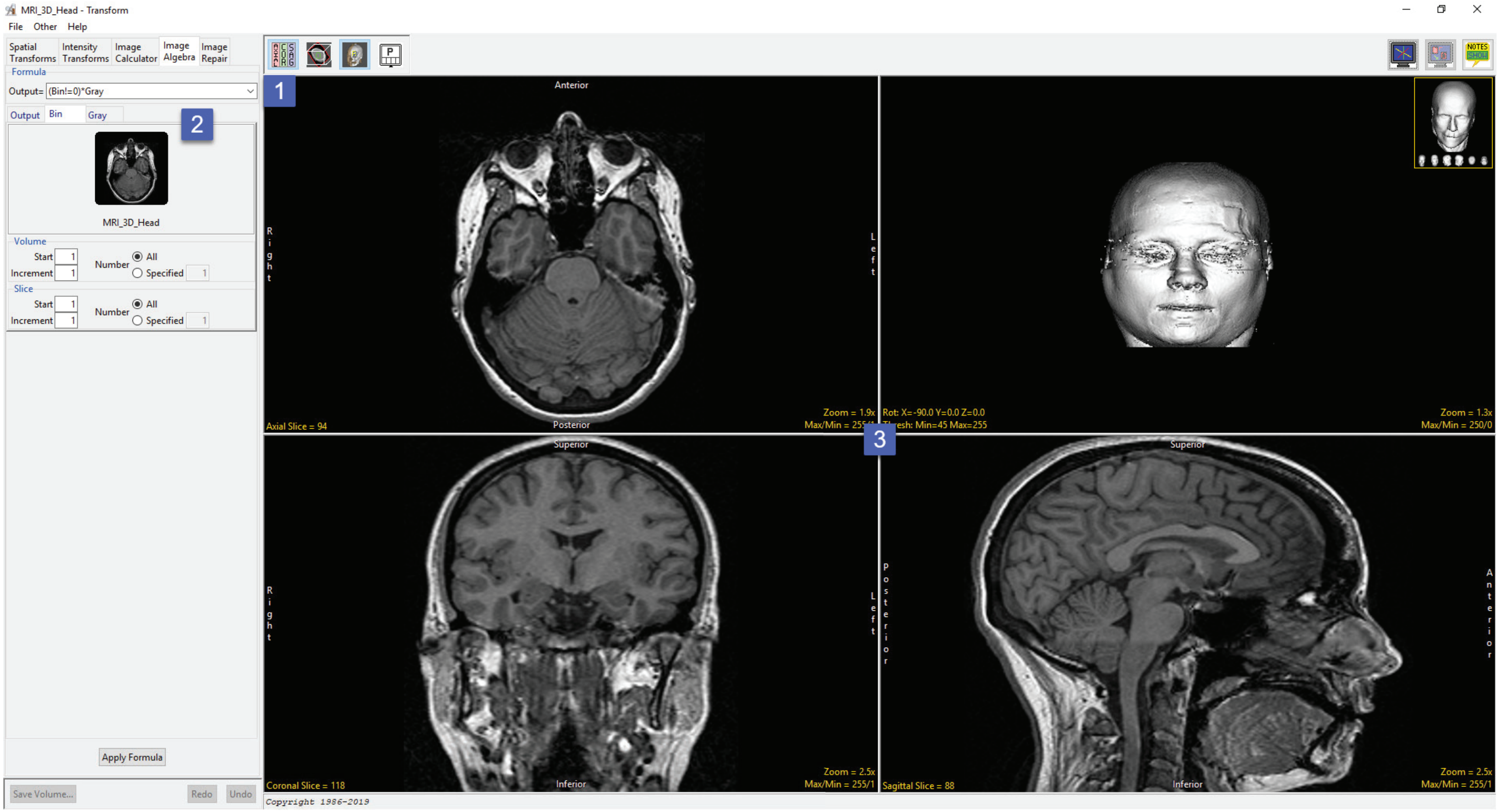
# Image Algebra

Image Algebra performs mathematical operations on volumes and constant numerical values by means of an algebraic formula parser. There are several useful formulas preloaded, as detailed in the table below.

The Image Algebra interface provides access to a Formula parser [1], variable tabs [2], and the standard display area [3].



MRI_3D_Head - Transform

Formula

Output= (Bin!=0)*Gray

Volume

Start 1

Increment 1

Number ● All ○ Specified 1

Slice

Start 1

Increment 1

Number ● All ○ Specified 1

Copyright 1986-2019

# Formula Parser

The Formula field allows users to enter a formula to process image data, alternatively users can choose from a list of predefined formulas.

A formula may be as simple as Output=Input, which copies a file, or as complex as to contain up to 1000 variables and 1000 constants. When entering complex formulas parenthesis should be used literally to group mathematical operations.

The following operators are supported; +, -, *, /, ( ), <, >, <= ,>= , ==, !=

| Symbol | Name | Meaning/Definition | Example |
|---|---|---|---|
| = | Equals sign | Equality; is equal to, equals | **Output = Input** --- Copies a file |
| != | Equality operator: Inequality | Not equal to operator; is not equal to, does not equal. Returns true if the operands to not have the same value; otherwise, it returns false. | **Output = (Input!=145)*Input**<br>Removes voxels equal to 145 from the image. '(Input!=145)' assigns all voxels not equal to 145 to 1 and all voxels equal to 145 to 0. '*Input' then multiples the resulting binary image with the original grayscale image, all voxels assigned to 1 are kept while all voxels assigned to 0 are removed from the final image. |
| == | Equality operator: Equality | Equal to operator returns true (1) if both operands have the same value; otherwise it returns false (0). | **Output = (Mask==0)*Volume**<br>Inverts the binary mask input image so all voxels set to 1 are removed from the final image while all voxels set to 0 are kept. Normally when multiplying a grayscale and binary the voxels set to 1 in the binary mask are kept while the voxels set to 0 are removed.  'Mask==0' keeps all the voxels in the binary image with a value of 0. '*Volume' multiples the (inverted) binary image with the grayscale images, the resulting grayscale image contains the voxels that were set to 0 in the binary image. |
| < | Strict inequality | Less than | **Output = (Input<155)*128**<br>In this formula if the voxel intensity value is less than 155, TRUE(1) is returned and these voxels are assigned an intensity value of 128, else False(0) is returned and the result is 0 and the original grayscale value is preserved. '(Input<155)' assigns all voxels less than 145 to 1 and all other voxels to 0. '*128' then assigns all voxels with a value of TRUE(1) an intensity value of 128. |

# Formula Parser (continued)

| Symbol | Name | Meaning/Definition | Example |
|---|---|---|---|
| > | Strict inequality | Greater than | **Output = (Input>45)*Input**<br>Keeps all voxels greater than 145 (146 and higher) in the image and removes all other voxels. '(Input>145)' assigns all voxels greater than 145 to 1 and all other voxels to 0. '*Input' then multiples the resulting binary image with the original grayscale image, all voxels assigned to 1 are kept while all voxels assigned to 0 are removed from the final image. |
| <= | Inequality | Less than or equal to | **Output = (Input<=145)*Input**<br>Keeps all voxels equal to and less than 145 in the image and removes all other voxels. |
| >= | Inequality | Greater than or equal to | **Output = (Input>=145)*Input**<br>Keeps all voxels equal to and greater than 145 in the image and removes all other voxels. |
| + | Plus sign | Addition | **Output = File1 + File2**<br>Adds two files. Summing data is for useful for averaging data for noise reduction. |
| - | Minus sign | Subtraction | **Output = File1 – File2**<br>Subtracts File2 from File1. Subtraction is useful for creating a difference image or removing background noise. |
| * | Asterisk | Multiplication | **Output = Binary*Grayscale**<br>Multiplies two data sets, in this example the binary image is multiplied by a greyscale image. All voxels in the binary mask equal to 1 will be kept in the resulting greyscale image while all voxels with a value of 0 will be removed. Useful for masking data. |
| / | Division slash | Division | **Output = (File1+File2)/2**<br>Averages 2 data sets. Useful for averaging images to reduce noise. |
| ( ) | Parentheses | Calculate expression inside first | **Output = (Input-intercept)/slope**<br>Scales image data based on the slope and intercept values from a linear regression line. Useful for rescaling CT data to HU's or BMD units. '(Input-intercept)' minuses the intercept value from the input value prior to dividing by the slope '/slope'. |

# Formula Parser (continued)

**Application examples:**

***Data Normalization:*** Output=(input-(min(input)))/(max(input))-(min(input))

***Scaling data to a phantom:*** Output = (Input-#intercept#)/#slope#

***Masking data and setting the excluded voxels to a specific intensity value:*** Output=(Mask==0)*Volume+(Mask==1)*#value#

The following functions are supported:

| Function | Notes |
|---|---|
| sqrt(equation) | Equation can be a variable or a constant. |
| log(equation) | Equation can be a variable or a constant. |
| exp(equation) | Equation can be a variable or a constant. |
| count() | Returns the output slice number. |
| max(variable) | Returns the maximum value. |
| min(variable) | Returns the minimum value. |
| avg(variable) | Returns the average value. |
| abs(variable) | Returns absolute value. |
| int(variable) | Converts to integer, truncating floating point values. |
| float(variable) | Converts to floating point values. |
| xpos() | Returns the x position in the input image. |
| ypos() | Returns the y position in the input image. |
| ADDTOT(num_add, inc) | Returns the sum of the added images as a floating point image. |
| ADDAVG(num_add, inc) | Returns the result of the added and averaged images. |
| RED(variable) | Returns the red portion of the RGB image if the variable specifies an RGB image or an image with a colormap. |
| GREEN(variable) | Returns the green portion of the RGB image if the variable specifies an RGB image or an image with a colormap. |
| BLUE(variable) | Returns the blue portion of the RGB image if the variable specifies an RGB image or an image with a colormap. |

Note: **ADDTOT()** and **ADDAVG()** are specialized functions that allow large groups of images to be summed or averaged. Unlike other functions which use a single image as input, these functions allow for multiple image inputs, with a single output image generated. The options in the variables menu all users to specify which input image is passed to a function. With these functions, the values specified in the variables menu specify only the starting slice for a group of images. num_add specifies the total number of images to add together, and inc specifies the increment between these images. Special status messages are generated when these functions are executed. Review these messages carefully if you experience problems accomplishing the desired total and averaging functions.

# Formula Parser (continued)

**Additional information when defining formulas:**

- Surrounding variables with #'s, will cause prompting for the information that is to be put in the formula in place of the variables.
- Variable names must begin with an alpha character and have no more than 15 alphanumeric characters.
- <,>,<=,>=,==,!= are boolean operators that can be used in user specified formulas. The following is an example showing the use of boolean operators:

There are several useful formulas preloaded, as detailed in the table below.

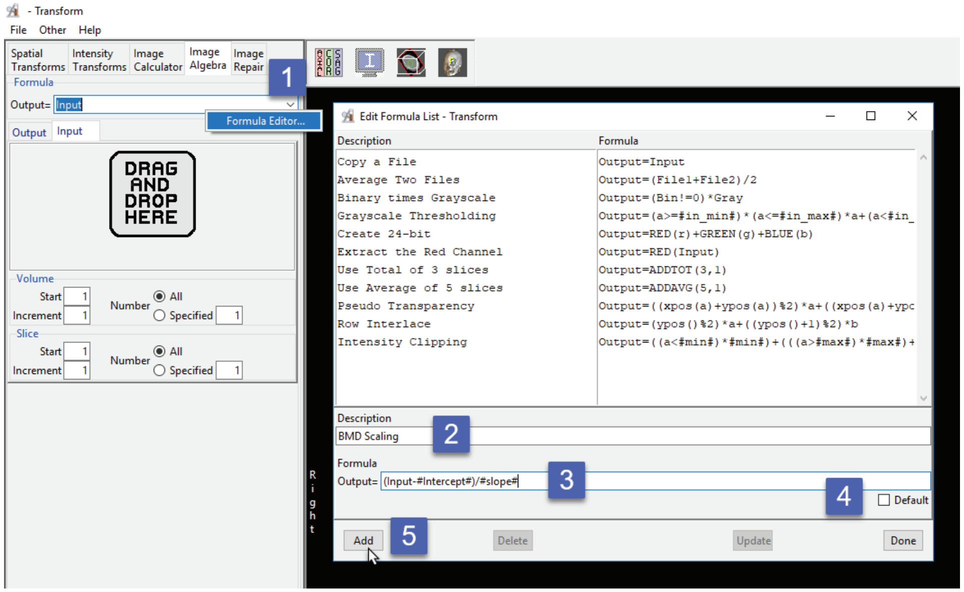
| Formula | Output= |
|---|---|
| Copy a File | Input |
| Average Two Files | (File1+File2)/2 |
| Binary times Grayscale | (Bin!=0)*Gray |
| Grayscale Thresholding | (a>=#in_min#)*(a<=#in_max#)*a+(a<#in_min#)*#out_min#+(a>#in_max#)*#out_max# |
| Create 24-bit | RED(r)+GREEN(g)+BLUE(b) |
| Extract the Red Channel | RED(Input) |
| Use Total of 3 slices | ADDTOT(3,1) |
| Use Average of 5 slices | ADDAVG(5,1) |
| Pseudo Transparency | ((xpos(a)+ypos(a))%2)*a+((xpos(a)+ypos(a)+1)%2)*b |
| Row Interlace | (ypos()%2)*a+((ypos()+1)%2)*b |
| Intensity Clipping | ((a<#min#)*#min#)+(((a>#max#)*#max#)+((a>=#min#)*(a<=#max#))*a) |

# Formula Parser (continued)

### Adding Formulas

To add your own formula to the list of formulas available from the formula drop down menu, do the following.

- Right click in the Formula field [1].
- In the Edit Formula List window returned enter a description for your formula [2].
- Next, enter the formula [3]. Remember to surround variables with #'s to make them imputable.
- Uncheck the Default option [4], unless you wish your formula to be the default formula that appears in the formula bar when Image Algebra is selected.
- Click Add [5].
- The formula will be added to the list [6] and will be selectable from the formula drop down menu [7].
- Apply Formula: Applies the formula to the image data.
- Save Volume: Opens the Save Transform window



Output= Input

Volume: Start 1, Increment 1, Number ● All ○ Specified 1

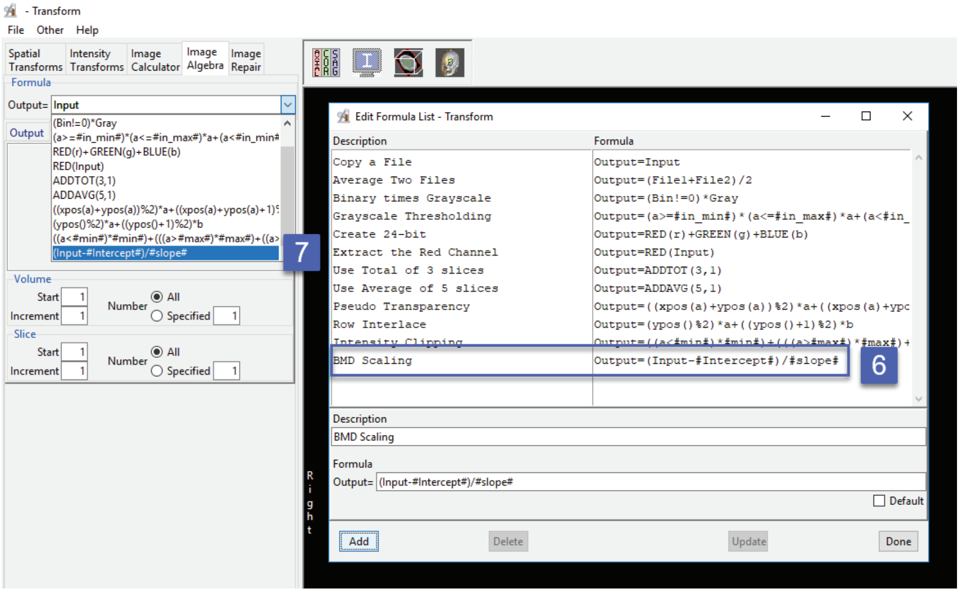
Slice: Start 1, Increment 1, Number ● All ○ Specified 1

| Description | Formula |
|---|---|
| Copy a File | Output=Input |
| Average Two Files | Output=(File1+File2)/2 |
| Binary times Grayscale | Output=(Bin!=0)*Gray |
| Grayscale Thresholding | Output=(a>=#in_min#)*(a<=#in_max#)*a+(a<#in_ |
| Create 24-bit | Output=RED(r)+GREEN(g)+BLUE(b) |
| Extract the Red Channel | Output=RED(Input) |
| Use Total of 3 slices | Output=ADDTOT(3,1) |
| Use Average of 5 slices | Output=ADDAVG(5,1) |
| Pseudo Transparency | Output=((xpos(a)+ypos(a))%2)*a+((xpos(a)+ypc |
| Row Interlace | Output=(ypos()%2)*a+((ypos()+1)%2)*b |
| Intensity Clipping | Output=((a<#min#)*#min#)+(((a>#max#)*#max#)+ |

Description: BMD Scaling

Formula: Output= (Input-#Intercept#)/#slope#

□ Default



Output= Input

(Bin!=0)*Gray
(a>=#in_min#)*(a<=#in_max#)*a+(a<#in_min#
RED(r)+GREEN(g)+BLUE(b)
RED(Input)
ADDTOT(3,1)
ADDAVG(5,1)
((xpos(a)+ypos(a))%2)*a+((xpos(a)+ypos(a)+1)%
(ypos()%2)*a+((ypos()+1)%2)*b
((a<#min#)*#min#)+(((a>#max#)*#max#)+((a>
(Input-#Intercept#)/#slope#

Volume: Start 1, Increment 1, Number ● All ○ Specified 1

Slice: Start 1, Increment 1, Number ● All ○ Specified 1

| Description | Formula |
|---|---|
| Copy a File | Output=Input |
| Average Two Files | Output=(File1+File2)/2 |
| Binary times Grayscale | Output=(Bin!=0)*Gray |
| Grayscale Thresholding | Output=(a>=#in_min#)*(a<=#in_max#)*a+(a<#in_ |
| Create 24-bit | Output=RED(r)+GREEN(g)+BLUE(b) |
| Extract the Red Channel | Output=RED(Input) |
| Use Total of 3 slices | Output=ADDTOT(3,1) |
| Use Average of 5 slices | Output=ADDAVG(5,1) |
| Pseudo Transparency | Output=((xpos(a)+ypos(a))%2)*a+((xpos(a)+ypc |
| Row Interlace | Output=(ypos()%2)*a+((ypos()+1)%2)*b |
| Intensity Clipping | Output=((a<#min#)*#min#)+(((a>#max#)*#max#)+ |
| BMD Scaling | Output=(Input-#Intercept#)/#slope# |

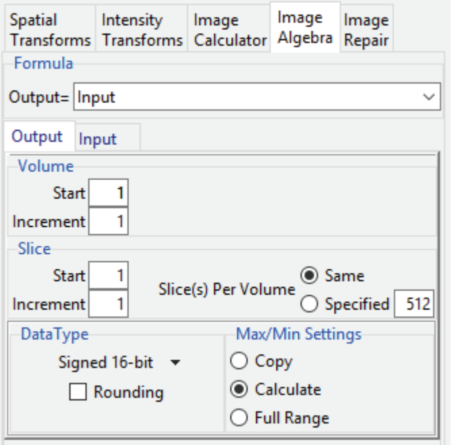
Description: BMD Scaling

Formula: Output= (Input-#Intercept#)/#slope#

□ Default

# Variables

Variables are dynamically generated when a formula is created or selected. The entries allow for the specification and/or modification of data values and parameters. Should a parameter fail to be provided and a default value cannot be determined, the formula parser will prompt the user for the value. The following options are available (depending on the current formula):



Formula

Output= Input

Volume

Start 1

Increment 1

Slice

Start 1

Increment 1

Slice(s) Per Volume ● Same ○ Specified 512

DataType

Signed 16-bit

□ Rounding

Max/Min Settings

○ Copy

● Calculate

○ Full Range

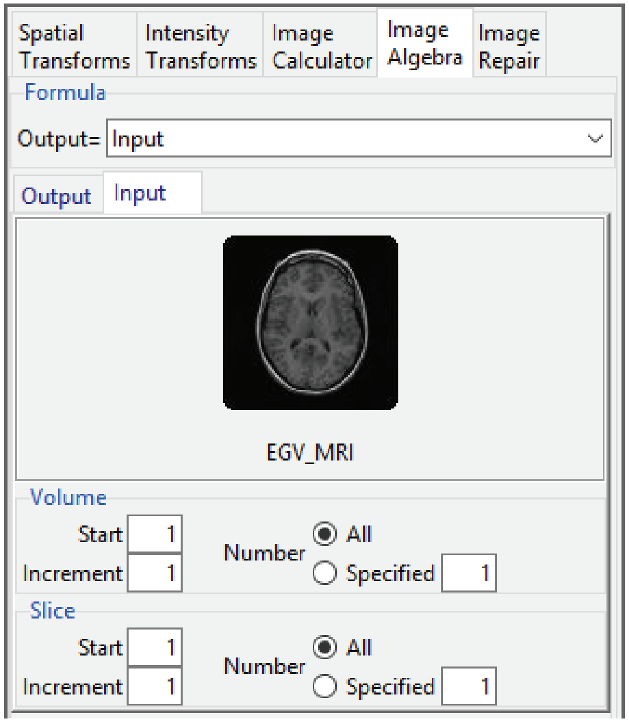
**Output:** allows for the specification of the output image attributes. The following options are available:

- Volume: The volume field enables specification of the volume attributes for the output file
  - Start: Specifies the starting volume number for the output file.
  - Increment: Specifies the increment used between volumes for the output file.
- Slice: The slice field enables specification of the slice attributes for the output file.
  - Start: Specifies the starting slice number in the output volume for each volume processed.
  - Increment: Specifies the increment between slices for the output volume.
  - Slice(s) Per Volume: Specifies the number of slices in each output volume. Options are: *Same:* When selected, indicates that the number of slices in the first input file is used as the number of slices in the output file; *Specified:* When selected, enables the specification of the number of slices in the output file.
- DataType: Specifies the data type of the output file. Users can choose from a list of available data types, including; unsigned 8-bit, signed 8-bit, unsigned 16-bit, signed 16-bit, unsigned 32-bit, signed 32-bit, float, RGB, and double.
  - Rounding: When enabled, floating values are converted to integers by rounding to the nearest integer, otherwise the number will be truncation at the decimal point. This option has no effect if the calculated value is an integer or the Output Datatype is float.
- Max/Min: Specifies the method used to set the maximum and minimum values for the output file. The following options are available:
  - Copy: Specifies that the maximum and minimum values are copied from the input file.
  - Calculate: Specifies that the maximum and minimum values are calculated for every slice processed.
  - Full Range: Specifies that the maximum and minimum are set to the full range possible for the current data type.

# Variables (continued)

**Variable:** The variable parameters, named assigned from the variable defined in the formula, allows for the specification of the variable (e.g. Input) volume attributes. The following options are available:

- Volume: The Volume field enables specification of the volume attributes for the variable volume.
    - Start: Specifies the starting volume number for the variable volume.
    - Increment: Specifies the increment used between volumes for the variable volume.
    - Number: Choose from the following to specify the number of volumes to be processed:
      *All:* Specifies that all volumes in the variable volume will be processed;
      *Specified:* Indicates the number of volumes in the variable volume to be processed.
- Slice: The slice field enables specification of the slice attributes for the variable volume.
    - Start: Specifies the starting slice number in the variable volume for each volume processed.
    - Increment: Specifies the increment between slices for the variable volume.
    - Number: Choose from the following to specify the number of slices to be processed: *All:* Specifies that all slices in the variable volume will be processed; *Specified:* Indicates the number of slices in the variable volume to be processed.



Output= Input

Volume
Start 1
Increment 1
Number ● All ○ Specified 1

Slice
Start 1
Increment 1
Number ● All ○ Specified 1