VERBOSE EXPLANATIONS

of ACToolkit objects for Max/MSP 6-8

Written by Jeyong Jung *https://jey-noise.com*

v. 0.9.2

Table of Contents

jeynoise~

Arguments

Setting the object to the ‘exp’ mode

1st (required): ***exp*** (mode specifier)

The object can perform the band-limited oscillation. However, when it's set to the *exp* mode, the object maybe involves aliasing on generating the phase signals so is not anymore entirely band-limited. Nevertheless, this *exp* mode can be used for producing White Noise (flat-spectrum noise) or exponentially-descending spectrum noise. The command *exp*, when it is identified as the first argument, sets the object to this *exp* mode. Notably, in this mode, all the other arguments and the methods (message selectors) of the object are inactivated. Moreover, the same inactivation applies to all the attributes except for the *@active* attribute and the signal inlets except for the second inlet. Therefore, in this mode, the object performs the single-parameter oscillation, and the only one parameter available is the downscaling factor for frequencies. (see also the *'exp' mode* tab in the object's help file) The signal coming through the second inlet sets the downscaling factor in the *exp* mode, and, the factor is set to 1 by default. The downscaling-factor smaller than 1 is replaced by 1.

Setting the object to perform band-limited oscillation

1st: Period Duration (optional, number)

The object generates noise by using a routine that randomly setting the frequencies of sine and cosine waves per desired amount of period(s). That is, when the object begins to produce the new sine-wave-period(s) of a certain frequency, the duration of the sine-wave-period(s) is/are derived from multiplying "the wavelength of the sine wave" by the period-duration factor. For instance, if the given period-duration is 1 constantly, the frequencies of sine waves can be changed whenever a sine-wave-period is ended. Moreover, if the derived duration of the sine-wave-period(s) is shorter than 1 sample, it will be replaced by 1 sample (see also the Period-by-Period Synthesis tab of the object's help file). The first argument of the object sets the period-duration factor. This argument can also be set by the signal coming through the second inlet. Negative values turn to positives, and the default setting is 1.

2nd: Phase Offset (optional, number)

The second argument determines the offsets applied to the phase signals generated by the object. The phase signals are outputted through the second outlet and used in oscillating sine and cosine waves outputted through the first and fourth outlets. In terms of sine and cosine waves, 1 means the phase shift of positive 360 degrees and -1 means the negative 360 degrees. Negative phase offsets equal to or less than -1 will be wrapped into the range 0 to -1, and positive ones equal to or greater than 1 will be wrapped into the range 0 to 1. The default setting is 0.

Attributes

@err_msg

[affects the object only when it receives external frequencies] The frequencies of the sine and cosine waves that the object generates can be set by using the external signals coming through the first inlet. In this case, thus the object is only capable of operating within a limited frequency range, it is probable that the external frequencies are in excess. Accordingly, the object examines the incoming frequencies and folds up the ones in excess into the allowed range so that they will all become available. In relation to such a feature of the object, the *@err_msg* attribute determines whether the object will notify the external frequencies are in excess. Furthermore, the upper and lower limits of the external frequencies are automatically set according to the

Sampling Rate. For more information about how the object derives the allowed range for the external frequencies, please refer to the *fold0* item in the Messages chapter. Any value greater than 0 turns on the attribute. The attribute is turned off by default.

@proc_time

The time required for processing a PDF-list (prescribes the distribution of frequencies) of a fixed size can vary according to the performance of computers. If such a performance factor is a concern, enable the @*proc_time* attribute to measure the time required for processing incoming PDF-lists. The information will be posted on the Max Console. Any value greater than 0 turns on the attribute. The attribute is turned off by default.

NB) On Windows, the measurement tends to result in 0 or 1 millisecond. This is because the rate of the clock-ticks used in the "Microsoft C Runtime Library" is not enough high.

@active

If the @*active* attribute is turned off, then all the signal outlets of the object will only output 0s. Though the PDF-list can be renewed even when the attribute is turned off, outputting done-bang through the rightmost outlet is delayed until the attribute is turned on. Any value greater than 0 turns on the attribute. The attribute is turned off by default.

@ignore_0PDF

[affects the object only in the **stand-alone** ***mode]*** In the stand-alone mode, turning on the @*ignore_0PDF* attribute makes the object to ignore the null PDF-lists such as {0, 0} and ones comprise 0s of any number. Therefore, when the attribute is on, the null PDF-lists do not affect the object, but the last non-null PDF-list will still be in effect. If the attribute is off, then feeding a null PDF-list makes the object to oscillate at the constant lowest-frequency-possible set by the *f0* and *f1* methods (message selectors). The attribute is turned off by default.

@f0

[affects the object only in the* stand-alone *mode] The *@f0* attribute performs the exact same feature as what the *f0* method (message selector) does. Please refer to the "stand-alone mode 2" and the “arguments & misc.” tabs of the object's help file and the *f0* item in the Messages chapter.

@f1

[affects the object only in the* stand-alone *mode] The *@f1* attribute performs the exact same feature as what the *f1* method (message selector) does. Please refer to the "stand-alone mode 2" and the “arguments & misc.” tabs of the object's help file and the *f0* item in the Messages chapter.

@fold0

[affects the object only in the* stand-alone *mode] The *@fold0* attribute performs the exact same feature as what the *fold0* method (message selector) does. Please refer to the "stand-alone mode 2" and the “arguments & misc.” tabs of the object's help file and the *fold0* item in the Messages chapter.

@fold1

[affects the object only in the* stand-alone *mode] The *@fold1* attribute performs the exact same feature as what the fold1 method (message selector) does. Please refer to the "stand-alone mode 2" and the “arguments & misc.” tabs of the object's help file and the *fold0* item in the Messages chapter.

Messages

f0

[affects the object only in the* stand-alone *mode] When it operates in the stand-alone mode (it does not receive external frequencies), the object derives PseudoRandom Numbers (PRNs) from the *rand()* function of C libraries, redistributes the PRNs, and then applies the redistributed PRNs to the frequencies of sine and cosine wave products. In this case, the *f0* and *f1* values set the range of the random frequencies. When a value among the *f0* and *f1* is entered, the object will first compare the two values and then sets both upper and lower frequency-boundaries according to the result of the comparison. If the *f0* and *f1* values exceed the folding boundaries determined by *fold0* and *fold1* values, then the values are folded into the folding-boundary (see also the *fold0* item below and the "stand-alone mode 2" tab of the object's help file). The default settings of *f0* and *f1* are 1 and 22050.

f1

[affects the object only in the* stand-alone *mode] Please refer to the *f0* item above.

fold0

[affects the object only in the* stand-alone *mode] The *fold0* and *fold1* values set the upper and lower limits on the *f0* and *f1* values (frequency boundaries). So, the frequency-boundary values that exceed the folding boundaries are folded into the folding boundaries (see also the *f0* item above and the "stand-alone mode 2" tab of the object's help file). When the one among the two *fold0* and *fold1* values is entered, the object will first compare the two values and then sets both upper and lower folding-boundaries according to the result of the comparison. Furthermore, there exist the ultimate limits on the folding boundaries, and they are derived from the Sampling Rate. The ultimate upper-limit of the folding boundaries is *SR ÷ 2*, and the lower one is the difference between *SR ÷ 2* and "the largest double-precision floating-point-number smaller than *SR ÷ 2*". For instance, when the SR is 44100, the ultimate lower limit is 0.0000000000073 (Hz.) approximately (see also the *info* item below). Those ultimate limits on the *fold0* and *fold1* values are also the limits on the external frequencies (see also the *@err_msg* item in the Attributes chapter and the "feeding frequencies" tab in the object's help file). Negative values turn to positives, and the values are limited to the ultimate limits derived from the SR. The default settings of *fold0* and *fold1* are 1 and 22050.

fold1

[affects the object only in the* stand-alone *mode] Please refer to the *fold0* item above.

info (no value is needed)

This message lets the object to post the information regarding the folding-boundaries and the folded frequency-boundaries on the Max Console.

rectify

[affects the object only in the* stand-alone *mode] Description is not prepared yet, so please refer to the "stand-alone mode 2" tab of the object's help file. Any value greater than 0 turns on the flag. The flag is turned off by default.

sine^2_spread

[affects the object only in the* stand-alone *mode] When it operates in the stand-alone mode (it does not receive external frequencies), the object randomly sets the frequencies of sine and cosine wave products based on a user-defined PDF-list (list of probabilities). In such a case, it is possible to interpolate the PDF-list that prescribes the distribution of frequencies by using three interpolation components—*sine^2*, *arcsec*, and *gauss*. Therefore, the *sine^2_spread*, *arcsec_spread*, and *gauss_spread* methods set each interpolation-component's continuity factor respectively. When this factor of a certain interpolation-component increases, the PDF of the frequencies—that is derived from using the chosen component—will have narrower or no gaps along the range set by the *f0* and *f1* methods (see also "all modes at once" tab in the object's help file). Negative values turn to positives, and the values greater than 1 will be wrapped. The default settings of *sine^2_spread*, *arcsec_spread*, and *gauss_spread* values are 1, 0, and 0.

arcsec_spread

[affects the object only in the* stand-alone *mode] Please refer to the *sine^2_spread* item above.

gauss_spread

[affects the object only in the* stand-alone *mode] Please refer to the *sine^2_spread* item above.

sine^2_ratio

[affects the object only in the* stand-alone *mode] When there are no external frequencies supplied to the object (in the stand-alone mode), the object derives the frequencies of sine and cosine waves from the internally generated random numbers. In this case, thus the PDF-list (list of probabilities) that prescribes the distribution of random frequencies can be interpolated by using three components, determining the proportion of each PDF-list-interpolation component is doable. The weight of each component is set by "a ratio value of a component" over "the total sum of the three *sine^2_ratio*, *arcsec_ratio*, and *gauss_ratio* values". Moreover, if the three ratio values are all set to 0, then 0 will be outputted through the first outlet and the lower frequency-boundary value will be outputted through the second outlet. Negative values turn to positives, and the values greater than 1 will be wrapped. The default settings of *sine^2_ratio*, *arcsec_ratio*, and *gauss_ratio* values are 1, 0, and 0.

arcsec_ratio

[affects the object only in the* stand-alone *mode] Please refer to the *sine^2_ratio* item above.

gauss_ratio

[affects the object only in the* stand-alone *mode] Please refer to the *sine^2_ratio* item above.

gauss_shape

[affects the object only in the* stand-alone *mode] This parameter sets the shape of the *gauss* component (bell-shaped curve) used in the PDF-list interpolation. Negative values turn to positives, and the values greater than 1 will be wrapped. The default setting is 0.5.

jeyrand~

Arguments

Setting the object to the 'sine^2' mode

1st (required): ***sine^2*** (mode specifier)

The argument, when it is identified as the first argument, sets the object to the *sine^2* mode. In this mode, the PDF-list (list of probabilities) interpolation of the object uses only the *sine^2* component (an upside-down cosine shaped curve).

2nd: *spread* factor of *sine^2* component (optional, 0-1)

3rd: *int/ext* factor of the object (optional, 0-1)

Setting the object to the 'arcsec' mode

1st (required): ***arcsec*** (mode specifier)

The argument, when it is identified as the first argument, sets the object to the *arcsec* mode. In this mode, the PDF-list (list of probabilities) interpolation of the object uses only the *arcsec* component, so the interpolated PDF-list will only result in comb-filtered shapes.

2nd: *spread* factor of *arcsec* component (optional, 0-1)

3rd: *int/ext* factor of the object (optional, 0-1)

Setting the object to the 'gauss' mode

1st (required): ***gauss*** (mode specifier)

The argument, when it is identified as the first argument, sets the object to the *gauss* mode. In this mode, the PDF-list (list of probabilities) interpolation of the object uses only the *gauss* component (a bell-shaped curve). This *gauss* component is shape-variable (see also the *shape* and *gauss_shape* items in the Messages chapter).

2nd: *spread* factor of *gauss* component (optional, 0-1)

3rd: *shape* factor of *gauss* component (optional, 0-1)

4th: *int/ext* factor of the object (optional, 0-1)

Setting the object to support all the three modes mentioned above ('all-at-once' mode)

When there's no argument or a numeric argument identified as the first argument written within it, the object is set to the all-at-once mode (see also the "all modes at once" tab in the object's help file). Then all the three modes of the object explained above are available at the same time.

1st: *spread* factor of *sine^2* component (optional, 0-1)

2nd: *ratio* factor of *sine^2* component (optional, 0-1)

3rd: *spread* factor of *arcsec* component (optional, 0-1)

4th: *ratio* factor of *arcsec* component (optional, 0-1)

5th: *spread* factor of *gauss* component (optional, 0-1)

6th: *ratio* factor of *gauss* component (optional, 0-1)

7th: *shape* factor of *gauss* component (optional, 0-1)

8th: *int/ext* factor of the object (optional, 0-1)

Attributes

@err_msg

Basically, the object can transform the incoming signal in the range 0-1 by redistributing the signal's discrete samples. However, the incoming signal maybe includes the samples that exceed the range 0-1. Nevertheless, feeding the samples in excess into the object does not make the object to be malfunctioning. Instead the samples in excess will be replaced by the pseudorandom numbers derived from the *rand()* function of C libraries. If whether "the incoming signal exceeds the allowed range 0-1" is a concern, enable the @*err_msg* attribute. Then, when signal sample(s) out of the range 0-1 is/are detected, the object posts error message(s) on the Max Console. Any value greater than 0 turns on the attribute. The attribute is turned off by default.

@proc_time

The time required for processing a PDF-list (prescribes how incoming samples will be redistributed) of a fixed size can vary according to the performance of computers. If such a performance factor is a concern, enable the @*proc_time* attribute to measure the time required for processing incoming PDF-lists. The information will be posted on the Max Console. Any value greater than 0 turns on the attribute. The attribute is turned off by default.

NB) On Windows, the measurement tends to result in 0 or 1 millisecond. This is because the rate of the clock-ticks used in the "Microsoft C Runtime Library" is not enough high.

@active

If the @*active* attribute is turned off, then the first and second outlet of the object will only output 0s. Though the PDF-list can be renewed even when the attribute is turned off, outputting done-bang through the third outlet is delayed until when the attribute is turned on. Any value greater than 0 turns on the attribute. The attribute is turned off by default.

@ignore_0PDF

Turning on the @*ignore_0PDF* attribute makes the object to ignore the null PDF-lists such as {0, 0} and ones comprise 0s of any number. Therefore, when the attribute is on, the null PDF-lists do not affect the object, but the last non-null PDF-list will still be in effect. If the attribute is off, then feeding a null PDF-list makes the object to output 0 through the first outlet and the lowest-frequency-possible set by the *f0* and *f1* methods (message selectors) through the second outlet. The attribute is turned off by default.

@f0

The @*f0* attribute performs the exact same feature as what the *f0* method (message selector) does. Please refer to the "frequency boundaries" and the 'misc.' tabs of the object's help file and the *f0* item in the Messages chapter.

@f1

The @*f1* attribute performs the exact same feature as what the *f1* method (message selector) does. Please refer to the "frequency boundaries" and the 'misc.' tabs of the object's help file and the *f0* item in the Messages chapter.

@fold0

The @*fold0* attribute performs the exact same feature as what the *fold0* method (message selector) does. Please refer to the "frequency boundaries" and the 'misc.' tabs of the object's help file and the *fold0* item in the Messages chapter.

@fold1

The @*fold1* attribute performs the exact same feature as what the fold1 method (message selector) does. Please refer to the "frequency boundaries" and the 'misc.' tabs of the object's help file and the *fold0* item in the Messages chapter.

Messages

f0

The *f0* and *f1* values set the range of frequencies outputted through the second outlet. When a value among the two is entered, the object will first compare the value with the other value and then sets both upper and lower frequency-boundaries according to the result of the comparison. If the *f0* and *f1* values exceed the folding boundaries determined by *fold0* and *fold1* values, then the values are folded into the folding-boundary (see also the *fold0* item below and the "frequency boundaries" tab of the object's help file). The default settings of *f0* and *f1* are 1 and 22050.

f1

Please refer to the *f0* item above.

fold0

The *fold0* and *fold1* values set the upper and lower limits of the frequency-boundaries (*f0* and *f1* values). So, the frequency-boundary values that exceed the folding boundaries are folded into the folding boundaries (see also the *f0* item above and the "frequency boundaries" tab of the object's help file). When the one among the two *fold0* and *fold1* values is entered, the object compares the value with the other value and then sets both upper and lower folding-boundaries according to the result of the comparison. Furthermore, there exist the ultimate limits on the folding boundaries, and they are derived from the Sampling Rate. The ultimate upper limit of the folding boundaries is *SR ÷ 2*, and the lower one is the difference between *SR ÷ 2* and "the largest double-precision floating-point-number smaller than *SR ÷ 2*". For instance, when the SR is 44100, the ultimate lower limit is 0.0000000000073 (Hz.) approximately (see also the *info* item below). Negative values turn to positives, and the values are limited to the ultimate limits derived from the SR. The default settings of *fold0* and *fold1* are 1 and 22050.

fold1

Please refer to the *fold0* item above.

info (no value is needed)

This message lets the object to post the information regarding the folding-boundaries and the folded frequency-boundaries on the Max Console.

rectify

Description is not prepared yet, so please refer to the "frequency boundaries" tab of the object's help file. Any value greater than 0 turns on the flag. The flag is turned off by default.

spread

[not available in the* all-at-once *mode] When the object is set to the one among the *sine^2*, *arcsec*, and *gauss* modes (see also the Arguments chapter), the *spread* method determines the discontinuity factor for the PDF-list interpolation of the object. 0 means that the PDF-list (list of probabilities) will not be interpolated. After that the output samples will show strictly discontinuous distribution due to the discontinuously defined probabilities in the PDF-list are not at all interpolated. When the value increases, the PDF of the redistributed samples will have narrower or no gaps along the sample space 0-1. Negative values turn to positives, and the values greater than 1 will be wrapped. The default setting is 1.

shape

[only available in the* gauss *mode] The method sets the shape of the *gauss* component (bell-shaped curve) used in the PDF-list interpolation. Negative values turn to positives, and the values greater than 1 will be wrapped. The default setting is 0.5.

int/ext

By default, when a/some signal wire(s) is/are connected to it, the object transforms the incoming signal by redistributing the discrete samples. Also, in this case, the object can randomly choose per sampling moment what to redistribute between the external signal and the PseudoRandom Numbers (PRNs) generated by the *rand()* function of C libraries. The *int/ext* method sets the proportion of the external signal and the one of the PRNs (White Noise) based on a given value. The less the value the greater the chance of redistributing the PRNs the object gets, and the more the value the greater the chance of redistributing the external signal. Negative values turn to positives, and the values greater than 1 will be wrapped. The default setting is 1.

sine^2_spread

[only available in the* all-at-once *mode] When the object is set to the *all-at-once* mode (see also the Arguments chapter), the object can provide all the three available PDF-list interpolation components at the same time. In this case, we can determine the discontinuity factor of each component—*sine^2*, *arcsec*, and *gauss*—by using *sine^2_spread*, *arcsec_spread*, and *gauss_spread* methods (message selectors) respectively. 0 means that the samples redistributed by the chosen component will result in the strictly discontinuous distribution due to the intervals between every two consecutive probabilities in the PDF-list are not at all interpolated. When the value increases, the PDF of the redistributed samples will have narrower or no gaps along the sample space 0-1. Negative values turn to positives, and the values greater than 1 will be wrapped. The default settings of *sine^2_spread*, *arcsec_spread*, and *gauss_spread* values are 1, 0, and 0.

arcsec_spread

[only available in the* all-at-once *mode] Please refer to the *sine^2_spread* item above.

gauss_spread

[only available in the* all-at-once *mode] Please refer to the *sine^2_spread* item above.

sine^2_ratio

[only available in the* all-at-once *mode] The *sine^2_ratio*, *arcsec_ratio*, and *gauss_ratio* methods are used only when the object is set to the all-at-once mode. In the all-at-once mode (see also the Arguments chapter), thus the PDF-list interpolation of the object can use all the three available components, determining the proportion of each interpolation component is doable. The weight of each component is derived from "a ratio value of a component" over "the total sum of three *sine^2_ratio*, *arcsec_ratio*, and *gauss_ratio* values". Moreover, if the three ratio values are all set to 0, then 0s will be outputted through the first outlet and the value of the lower frequency-boundary will be outputted through the second outlet. Negative values turn to positives, and the values greater than 1 will be wrapped. The default settings of *sine^2_ratio*, *arcsec_ratio*, and *gauss_ratio* values are 1, 0, and 0.

arcsec_ratio

[only available in the* all-at-once *mode] Please refer to the *sine^2_ratio* item above.

gauss_ratio

[only available in the all-at-once mode] Please refer to the *sine^2_ratio* item above.

gauss_shape

[only available in the all-at-once mode] The method sets the shape of the *gauss* component (bell-shaped curve) used in the PDF-list interpolation. Negative values turn to positives, and the values greater than 1 will be wrapped. The default setting is 0.5.

out0_enable

When the *out0_enable* flag is turned off, the first outlet of the object only outputs 0. If the *rectify* flag is turned on, then turning off the *out0_enable* flag makes the object to consume little less CPU resources. Any value greater than 0 turns on the flag. The flag is turned on by default.

out1_enable

When the *out1_enable* flag is turned off, the second outlet of the object only outputs the value of the lower frequency-boundary. Any value greater than 0 turns on the flag. The flag is turned on by default.

Arguments

Setting the object to the 'sine^2' mode

1st (required): ***sine^2*** (mode specifier)

The argument, when it is identified as the first argument, sets the object to the *sine^2* mode. In this mode, the PDF-list (list of probabilities) interpolation of the object uses only the *sine^2* component (an upside-down cosine shaped curve).

2nd: *spread* factor of *sine^2* component (optional, 0-1)

3rd: *int/ext* factor of the object (optional, 0-1)

Setting the object to the 'arcsec' mode

1st (required): ***arcsec*** (mode specifier)

The argument, when it is identified as the first argument, sets the object to the *arcsec* mode. In this mode, the PDF-list (list of probabilities) interpolation of the object uses only the *arcsec* component, so the interpolated PDF-list will only result in comb-filtered shapes.

2nd: *spread* factor of *arcsec* component (optional, 0-1)

3rd: *int/ext* factor of the object (optional, 0-1)

Setting the object to the 'gauss' mode

1st (required): ***gauss*** (mode specifier)

The argument, when it is identified as the first argument, sets the object to the *gauss* mode. In this mode, the PDF-list (list of probabilities) interpolation of the object uses only the *gauss* component (a bell-shaped curve). This *gauss* component is shape-variable (see also the *shape* and *gauss_shape* items in the Messages chapter).

2nd: *spread* factor of *gauss* component (optional, 0-1)

3rd: *shape* factor of *gauss* component (optional, 0-1)

4th: *int/ext* factor of the object (optional, 0-1)

Setting the object to support all the three modes mentioned above ('all-at-once' mode)

When there's no argument or a numeric argument identified as the first argument written within it, the object is set to the all-at-once mode (see also the "all modes at once" tab in the object's help file). Then all the three modes of the object explained above are available at the same time.

1st: *spread* factor of *sine^2* component (optional, 0-1)

2nd: *ratio* factor of *sine^2* component (optional, 0-1)

3rd: *spread* factor of *arcsec* component (optional, 0-1)

4th: *ratio* factor of *arcsec* component (optional, 0-1)

5th: *spread* factor of *gauss* component (optional, 0-1)

6th: *ratio* factor of *gauss* component (optional, 0-1)

7th: *shape* factor of *gauss* component (optional, 0-1)

8th: *int/ext* factor of the object (optional, 0-1)

Attributes

@err_msg

Basically, the object redistributes incoming floating-point numbers in the range 0-1. However, the incoming numbers maybe exceed the range 0-1. Nevertheless, feeding the numbers in excess into the object does not make the object to be malfunctioning. Instead the numbers in excess will be replaced by the pseudorandom numbers derived from the *rand()* function of C libraries. If whether "the incoming numbers exceed the allowed range 0-1" is a concern, enable the *@err_msg* attribute. Then, when the incoming number(s) out of the range 0-1 is/are detected, the object posts error message(s) on the Max Console. Any value greater than 0 turns on the attribute. The attribute is turned off by default.

@proc_time

The time required for processing a PDF-list (prescribes how incoming numbers will be redistributed) of a fixed size can vary according to the performance of computers. If such a performance factor is a concern, enable the *@proc_time* attribute to measure how much time is required for processing incoming PDF-lists. The information will be posted on the Max Console. Any value greater than 0 turns on the attribute. The attribute is turned off by default.

NB) On Windows, the measurement tends to result in 0 or 1 millisecond. This is because the rate of the clock-ticks used in the "Microsoft C Runtime Library" is not enough high.

@ignore_0PDF

Turning on the *@ignore_0PDF* attribute makes the object to ignore the null PDF-lists such as {0, 0} and ones comprise 0s of any number. Therefore, when the attribute is on, the null PDF-lists do not affect the object, but the last non-null PDF-list will still be in effect. If the attribute is off, then feeding a null PDF-list makes the object only output 0 through the first outlet or the lists consist of only 0s through the second outlet. The attribute is turned off by default.

Messages

bang

When received a bang, the object first generates a PseudoRandom Number (PRN) internally, redistribute the PRN according to the PDF-list (list of probabilities), and then outputs the redistributed PRN through the first (leftmost) outlet.

spread

[not available in the* all-at-once *mode] When the object is set to the one among the *sine^2*, *arcsec*, and *gauss* modes (see also the Arguments chapter), the *spread* method determines the discontinuity factor for the PDF-list interpolation of the object. 0 means that the PDF-list (list of probabilities) will not be interpolated. After that the output numbers will show strictly discontinuous distribution due to the discontinuously defined probabilities in the PDF-list are not at all interpolated. When the value increases, the PDF of the redistributed samples will have narrower or no gaps along the sample space 0-1. Negative values turn to positives, and the values greater than 1 will be wrapped. The default setting is 1.

shape

[only available in the* gauss *mode] The method sets the shape of the *gauss* component (bell-shaped curve) used in the PDF-list interpolation. Negative values turn to positives, and the values greater than 1 will be wrapped. The default setting is 0.5.

int/ext

Basically, the object performs the redistribution of the incoming floating-point numbers in the range 0-1. Also, the object can randomly choose per number reception what to redistribute between the received number and the PseudoRandom Number (PRN) generated by the *rand()* function of C libraries. The *int/ext* method sets the proportion of the incoming number and the one of the PRNs (White Noise) based on a given value. The less the value the greater the chance of redistributing the PRNs the object gets, and the more the value the greater the chance of redistributing the incoming numbers. Negative values turn to positives, and the values greater than 1 will be wrapped. The default setting is 1.

sine^2_spread

[only available in the* all-at-once *mode] When the object is set to the all-at-once mode (see also the Arguments chapter), the object can provide all the three available PDF-list interpolation components at the same time. In this case, we can determine the discontinuity factor of each component—*sine^2*, *arcsec*, and *gauss*—by using *sine^2_spread*, *arcsec_spread*, and *gauss_spread* methods (message selectors) respectively. 0 means that the samples redistributed by the chosen component will result in the strictly discontinuous distribution due to the intervals between every two consecutive probabilities in the PDF-list are not at all interpolated. When the value increases, the PDF of the redistributed numbers will have narrower or no gaps along the sample space 0-1. Negative values turn to positives, and the values greater than 1 will be wrapped. The default settings of *sine^2_spread*, *arcsec_spread*, and *gauss_spread* values are 1, 0, and 0.

arcsec_spread

[only available in the* all-at-once *mode] Please refer to the *sine^2_spread* item above.

gauss_spread

[only available in the* all-at-once *mode] Please refer to the *sine^2_spread* item above.

sine^2_ratio

[only available in the* all-at-once *mode] The *sine^2_ratio*, *arcsec_ratio*, and *gauss_ratio* methods are used only when the object is set to the all-at-once mode. In the all-at-once mode (see also the Arguments chapter), thus the PDF-list interpolation task of the object can use all the three available components, determining the proportion of each interpolation component is doable. The weight of each component is derived from "a ratio value of a component" over "the total sum of three *sine^2_ratio*, *arcsec_ratio*, and *gauss_ratio* values". Moreover, if the three ratio values are all set to 0, then 0s will be outputted through the first outlet and the lists consist of only 0s through the second outlet. Negative values turn to positives, and the values greater than 1 will be wrapped. The default settings of *sine^2_ratio*, *arcsec_ratio*, and *gauss_ratio* values are 1, 0, and 0.

arcsec_ratio

[only available in the* all-at-once *mode] Please refer to the *sine^2_ratio* item above.

gauss_ratio

[only available in the* all-at-once *mode] Please refer to the *sine^2_ratio* item above.

gauss_shape

[only available in the* all-at-once *mode] The method sets the shape of the *gauss* component (bell-shaped curve) used in the PDF-list interpolation. Negative values turn to positives, and the values greater than 1 will be wrapped. The default setting is 0.5.

jeyhisto~ & jeyhisto

Arguments

1st: Resolution (optional, integer 2-32767)

The first argument determines the size of the histograms. It can also be understood as the resolution of the histograms due to the object is only capable of analyse the samples in the fixed range 0-1. The more the value the greater the precision on Binning (or Razoring) the incoming samples the object gets. Furthermore, this argument can also be modified by feeding integers into the object. Please note that the latest value entered to the object will be applied to the histograms only after the histogram is reset (by outputting it or giving the 'reset' message to the object). The maximum size of the histograms is 32767, and the minimum is 2. Negative values turn to positives, and the values that exceed the range 2-32767 will be ignored. The default setting is 128.

2nd: Analysis Method (optional, 0-1)

The second argument determines which analysis method between Binning and Razoring the object will use. The two methods can be combined by applying a floating-point value greater than 0 and lower than 1 to this argument. A graphical explanation regarding how the two analyses methods work is prepared in the 'binning/razoring' tab of the object's help file. Negative values turn to positives, and the values greater than 1 will be wrapped. The default setting is to use Binning only (1).

3rd: Auto-reset Flag (optional, on/off)

The third argument determines whether the object will reset the histogram after outputting it. If this flag is turned off, then the analysis data of the object will not be reset even after outputting histograms so that the following analyses will be derived from all the data collected since the previous resetting or the loading of the object. Any value greater than 0 turns on the flag. The flag is turned on by default.

Attributes

@err_msg

Basically, the object analyses the samples in the range 0-1. Therefore, the object simply ignores the incoming samples exceeding the range 0-1. However, if whether "the incoming samples exceed the allowed range 0-1" is a concern, enable the *@err_msg* attribute. Then, when the incoming sample(s) out of the range 0-1 is/are detected, the object posts an error message on the Max Console after outputting a histogram. Any value greater than 0 turns on the attribute. The attribute is turned off by default.

@normalize

When the *@normalize* attribute is turned on, the probabilities included in a histogram product of the object will be rescaled to be in the fixed range 0-1. Any value greater than 0 turns on the attribute. The attribute is turned on by default.

@autoreset

The *@autoreset* attribute performs the same feature as the third argument (see also the “Auto-reset Flag” item in the Arguments chapter) but is provided for setting the flag through the *attrui* object of Max. Any value greater than 0 turns on the attribute. The attribute is turned on by default.

Messages

bang

If the number of the samples that the object analysed is greater than 1, then we can let the object to output a histogram by sending a bang. When the *autoreset* flag is turned on, the number of the analysed samples and the analysis data will be initialized after outputting a histogram. On the contrary, by turning off the *autoreset* flag, we can preserve the number of the analysed samples and the analysis data even after outputting a histogram (see also the 'Auto-reset Flag' item in the Arguments chapter).

mode

The *mode* method performs the same feature as the second argument (see also the 'Analysis Method' item in the Arguments chapter). Negative values turn to positives, and the values greater than 1 will be wrapped. The default setting is 1.

autoreset

The *autoreset* method performs the same feature as the third argument (see also the 'Auto-reset Flag' item in the Arguments chapter). Any value greater than 0 turns on the flag. The flag is turned on by default.

col

When the *col* flag is turned off, the incoming samples will not be analysed at all. Any value greater than 0 turns on the flag. The flag is turned on by default.

reset (no value is needed)

When this message is entered, the object initializes the analysis data without outputting a histogram.