Detailed here is a reference for all types of signal processing nodes that are built into the language intrinsically. More can be defined at user-level by using the “subgraph” functionality described in the tutorial.

Each page has a brief description of the operation of nodes of this type, as well as a listing of input ports, output ports, and parameters. Understand the difference between parameters and ports: ports process audio dynamically through the graph during runtime of the language, whereas parameters exist to initialise nodes. When parameters are “linked” to a port, this means that the parameter will be used if no input connections lead to this port. For example, a static frequency parameter may be provided to an oscillator, which will be used if there is no input signal controlling the oscillator frequency.

Index:

Sine_Oscillator~

Description: Generates a simple sine wave at a given frequency and a given phase offset. A gate signal can be used to sync the oscillator phase to another oscillator or clock.

Input Ports	0|	Frequency	The frequency of sinusoidal oscillation
	1|	Sync gate	If a gate signal is received, the internal phase count is reset to zero
	2|	Phase offset	Modulate the phase offset
Output Ports	|0	Signal output	The generated sine signal
Parameters	0	Frequency	Linked to port 0
	1	Phase	Linked to port 2

Saw_Oscillator~

Description: Generates an aliased saw wave at a given frequency. Antialiasing will be provided in the future.

Input Ports	0|	Frequency	The frequency of saw oscillation
	1|	Sync gate	If a gate signal is received, the internal phase count is reset to zero
Output Ports	|0	Signal output	The generated saw signal
Parameters	0	Frequency	Linked to port 0

Square_Oscillator~

Description: Generates an aliased square wave at a given frequency and pulse width. Antialiasing will be provided in the future.

Input Ports	0|	Frequency	The frequency of square oscillation
	1|	Pulse width	Controls the duty cycle, ranges from -1 to 1, where 0 is 50% duty cycle.
Output Ports	|0	Signal output	The generated square signal
Parameters	0	Frequency	Linked to port 0
	1	Pulse Width	Linked to port 1

Triangle_Oscillator~

Description: Generates an aliased triangle wave at a given frequency. Antialiasing will be provided in the future.

Input Ports	`0|`	Frequency	The frequency of triangle oscillation
	`1|`	Sync gate	If a gate signal is received, the internal phase count is reset to zero
Output Ports	`|0`	Signal output	The generated triangle signal
Parameters	`0`	Frequency	Linked to port `0`

Noise~

Description: Produces white noise. Each output sample is a random value between -1 and 1.

Input Ports		None	
Output Ports	`|0`	Signal output	The generated noise signal
Parameters		None	

Constant~

Description: Always outputs a constant value.

Input Ports		None	
Output Ports	|0	Signal output	The generated constant signal
Parameters	0	Constant value	The constant value to be output

Clock~

Description: Outputs clock pulses with a certain time interval

Input Ports		None	
Output Ports	|0	Signal output	The generated clock signal
Parameters	0	Rate	The clock interval in samples

Timer~

Description: Counts up in seconds from the start of the program. Can be reset

Input Ports	`0|`	Reset	A gate signal here will reset the clock to `0`
Output Ports	`|0`	Signal output	The generated timer signal in seconds
Parameters		None	

Phasor~

Description: Generates a ramp from 0 to 1 with a certain period

Input Ports	`0|`	Period	How long should it take for the phase to make one rotation
	`1|`	Sync	Reset the phasor to zero with a gate
	`2|`	Phase offset	Phase modulate the phasor
Output Ports	`|0`	Signal output	The generated phasor signal
Parameters	`0`	Period	Linked to input port 1

Add~

Description: Sums together two signals, or a signal and a constant parameter

Input Ports	0|	Signal	The signal that represents the left operand of the addition operation
	1|	Offset	The signal that should be added to the first
Output Ports	|0	Signal output	The summed output signal
Parameters	0	Default value	Linked to port 1

Multiply~

Description: Multiplies together two signals, or a signal and a constant parameter

Input Ports	0|	Signal	The signal that represents the left operand of the multiplication operation
	1|	Multiplier	The signal that should be multiplied with the first
Output Ports	|0	Signal output	The output product
Parameters	0	Default value	Linked to port 1

Subtract~

Description: Subtract two signals, or a signal and a constant parameter

Input Ports	0|	Signal	The signal that represents the left operand of the subtraction operation
	1|	Subtrahend	The signal that should be subtracted from the first
Output Ports	|0	Signal output	The output signal
Parameters	0	Default value	Linked to port 1

Divide~

Description: Divide two signals, or a signal and a constant parameter

Input Ports	0|	Signal	The signal that represents the left operand of the division operation
	1|	Divider	The signal that should be divided with the first
Output Ports	|0	Signal output	The output signal
Parameters	0	Default value	Linked to port 1

Power~

Description: Raises one signal to the power of another, or raises a signal to the power of a constant

Input Ports	0|	Signal	The signal that represents the left operand of the exponentiation operation
	1|	Power	The signal that should be first should be raised to
Output Ports	|0	Signal output	The output product
Parameters	0	Default value	Linked to port 1

Exponentiate~

Description: Constant to the power of a signal

Input Ports	0|	Signal	The signal that represents the exponent
	1|	Base	The signal that should be raised to the first signal's power
Output Ports	|0	Signal output	The output product
Parameters	0	Default value	Linked to port 1

Delay_Line~

Description: Represents a delay buffer which a signal can be written to. The delay time can be varied up to the maximum specified size of the buffer.

Input Ports	0|	Signal	The signal that should be written to the delay buffer
	1|	Tap position	The length of the buffer that should be used
Output Ports	|0	Signal output	The result of reading from the buffer
Parameters	0	Maximum size	How much space should be allocated

Sample_And_Hold~

Description: Holds the same value from an input signal, updates when a gate is received

Input Ports	0|	Signal	The signal that the value is selected from
	1|	Hold	Hold a new value
Output Ports	|0	Signal output	The output held value
Parameters		None	

Envelope_Follower~

Description: Extracts the amplitude envelope of a signal. Allows you to specify attack and decay time

Input Ports	0|	Signal	The signal from which to extract the time-amplitude envelope information
	1|	Attack time	Set the time constant for rising signals in samples
	2|	Decay time	Set the time constant for falling signals in samples
Output Ports	|0	Signal output	The generated clock signal
Parameters	0	Attack	Linked to port 1
	1	Decay	Linked to port 2

Envelope_Generator~

Description: Generates a single linear envelope section with a given time length in samples. It is possible to create quadratic or other tapered time-envelopes using arithmetic operations from this

Input Ports	0|	Trigger	A gate to this input will start trigger the envelope
	1|	Length	The length of the envelope in samples
	2|	Start	The start value of the envelope section
	3|	End	The end value of the envelope section
Output Ports	|0	Signal output	The generated envelope signal
Parameters	0	Length	Linked to port 1
	1	Start	Linked to port 2
	2	End	Linked to port 3

Clamp~

Description: Clamp a signal to force it to stay in a specific range. Defaults to 0 - 1

Input Ports	0|	Signal	The signal that should be restrained
	1|	Minimum value	The lower value to clamp
	2|	Maximum value	The upper value to clamp
Output Ports	|0	Signal output	The clamped signal
Parameters	0	Minimum	Linked to port 1
	1	Maximum	Linked to port 2

Inverse~

Description: Calculate the additive inverse of a signal, equal to a 180-degree phase rotation

Input Ports	0|	Signal	The signal that should be inverted
Output Ports	|0	Signal output	The inverted signal
Parameters		None	

Comparator~

Description: Output a DC signal of value 0 or 1, depending on whether the input is greater than another value

Input Ports	`0|`	Signal	The signal to compare
	`1|`	Comparand	The value to compare to
Output Ports	`|0`	Signal output	0 or 1, result of the comparison
Parameters	`0`	Comparand	Linked to input 1

Reciprocal~

Description: Calculates 1/z for signal z. May result in signals valued `inf` or `nan` according to IEEE 32bit floating point specifications

Input Ports	`0|`	Signal	The signal to reciprocate
Output Ports	`|0`	Signal output	The result of the reciprocation operation
Parameters		None	

Bi_to_Unipolar~

Description: A utility which remaps a signal from the range (-1, 1) to (0, 1) by scaling and shifting

Input Ports	0|	Signal	The signal to map
Output Ports	|0	Signal output	The result of the scaling operation
Parameters		None	

Sin~

Description: Calculates the sine of a signal

Input Ports	0|	Signal	The signal to which the sine function is applied
Output Ports	|0	Signal output	The result of the sine operation
Parameters		None	

Cos~

Description: Calculates the cosine of a signal

Input Ports	0|	Signal	The signal to which the cosine function is applied
Output Ports	|0	Signal output	The result of the cosine operation
Parameters		None	

Tanh~

Description: Calculates the hyperbolic tangent of a signal

Input Ports	0|	Signal	The signal to which the tanh function is applied
Output Ports	|0	Signal output	The result of the tanh operation
Parameters		None	

Modulo~

Description: Calculates the remainder of a division of a signal by another signal, or a signal by a constant

Input Ports	0|	Signal	The signal to which the modulo function is applied
	1|	Divisor	The signal to divide by
Output Ports	|0	Signal output	The result of the modulo operation
Parameters	0	Divisor	Linked to port 1

Abs~

Description: Calculates the magnitude of a signal

Input Ports	0|	Signal	The signal of which to calculate the absolute value (removing the sign)
Output Ports	|0	Signal output	The result of the modulus operation
Parameters		None	

Floor~

Description: Apply the floor function to a signal, removing the decimal part. Used for indexing buffers with a signal

Input Ports	0|	Signal	The signal to floor
Output Ports	|0	Signal output	The floored signal
Parameters		None	

Ceil~

Description: Apply the ceiling function to a signal, rounding up to the nearest integer. Used for indexing buffers with a signal

Input Ports	0|	Signal	The signal to ceil
Output Ports	|0	Signal output	The ceiled signal
Parameters		None	

Sign~

Description: Calculate the sign of the signal, returning -1, 0 or 1

Input Ports	0|	Signal	The signal to sign
Output Ports	|0	Signal output	The signed signal
Parameters		None	

Log~

Description: Calculate the logarithm of a signal, with an arbitrary base

Input Ports	0|	Signal	The signal to get the logarithm of
	1|	Logarithmic base	The base of the logarithm operation
Output Ports	|0	Signal output	The signed signal
Parameters	0	Base	Linked to port 1. Defaults to Euler's number

Atan~

Description: Calculates the arctangent tangent of a signal

Input Ports	0|	Signal	The signal to which the arctangent function is applied
Output Ports	|0	Signal output	The result of the arctangent operation
Parameters		None	

Write_File~

Description: Write the input signal to a file on disk. The file will be created if it is not available. File size needs to be specified. The format will be raw bytes of 32bit floating point LPCM, little-endian, 44100hz sampling rate

Input Ports	0|	Signal	The signal which should be written to the file
Output Ports		None	
Parameters	0	Filename	The name of the file to maybe create and write to
	1	File size	How much audio data should be written. The file will be resized if necessary

Read_File~

Description: Read a signal from a file. The format should be raw bytes of 32bit floating point LPCM, little-endian, 44100hz sampling rate. Outputs 0s after reaching the end of the file

Input Ports		None	
Output Ports	|0	Signal	The signal which has been read from the file
Parameters	0	Filename	The name of the file to maybe create and write to

---

Step_Sequence~

Description: Holds a list of numbers, which can be stepped through using a gate. Cycles when it gets to the end

Input Ports	0|	Trigger	Gate signal reads the next value from the list
Output Ports	|0	Value	The value that is currently being indexed (DC)
Parameters	0	Sequence	The sequence to be stepped

Index_Sequence~

Description: Holds a list of numbers, which can be indexed using a signal. Performs linear interpolation between values in the case of fractional signals. This node can be used for wavetable synthesis and audio buffer playback

Input Ports	0|	Index	The index value into the sequence
Output Ports	|0	Value	The interpolated value read from the sequence
Parameters	0	Sequence	The sequence to be indexed

Smooth~

Description: Linear, first-order lowpass feedback filter which is suitable as a slew rate limiter or for general parameter smoothing

Input Ports	0|	Signal	The signal to be smoothed
	1|	Corner frequency	The cutoff frequency of the lowpass filtering operation
Output Ports	|0	Signal	The filtered signal
Parameters	0	Frequency	Linked to port 1

Lowpass_Filter~

Highpass_Filter~

Bandpass_Filter~

Allpass_Filter~

Description: A set of second-order biquadratic filters, representing several common filter shapes. Allows you to specify resonance (filter quality factor) and the cutoff frequency

Input Ports	0|	Signal	The signal to be smoothed
	1|	Frequency	The cutoff frequency or centre frequency of the filtering operation
	2|	Resonance	The Q of the filter measured linearly
Output Ports	|0	Signal	The filtered signal
Parameters	0	Frequency	Linked to port 1
	1	Resonance	Linked to port 2

Convolver~

Description: Convolves an arbitrary impulse response with a signal, producing a finite impulse response filter. Implements time-domain convolution and is not suited for impulse responses much longer than 128 points

Input Ports	0|	Signal	The signal to be filtered
Output Ports	|0	Signal	The filtered signal
Parameters	0	Impulse response	A sequence representing the desired filter impulse response

---

Pole~

Description: Allows the user to place a resonant pole on the complex plane, for direct-z-plane filter design

Input Ports	0|	Signal	The signal to be filtered
	1|	Real component	Real position of the filter pole
	2|	Imaginary component	Imaginary position of the filter pole
Output Ports	|0	Signal	The filtered signal
Parameters	0	Position	Linked to both ports 1 and 2, real and imaginary parts of the parameter respectively

Zero~

Description: Allows the user to place a zero on the complex plane, for direct-z-plane filter design

Input Ports	0|	Signal	The signal to be filtered
	1|	Real component	Real position of the filter zero
	2|	Imaginary component	Imaginary position of the filter zero
Output Ports	|0	Signal	The filtered signal
Parameters	0	Position	Linked to both ports 1 and 2, real and imaginary parts of the parameter respectively