## 1. ListenerPoint (~ Web Audio API: AudioListener)

AudioListener: represents the position and orientation of the person listening to the audio scene.

Attributes:

| | |
|---|---|
| positionX , positionY , positionZ | The positionX , positionY , and positionZ fields represent the location of the listener in 3D Cartesian coordinate space. SpatialSound node uses this position relative to individual audio sources for spatialization. The default value is 0,0,0. |
| forwardX, forwardY, forwardZ | The forwardX, forwardY, forwardZ parameters represent a direction vector in 3D space. Both a forward vector and an up vector are used to determine the orientation of the listener. In simple human terms, the forward vector represents which direction the person's nose is pointing. |
| upX, upY, upZ | The up vector represents the direction the top of a person's head is pointing. These two vectors are expected to be linearly independent. |
| **gain (extra in ListenerPoint)** | represents a change in volume. The default value is 1. |
| **isViewpoint (extra in ListenerPoint)** | specifies if the listener position is the viewpoint of camera. If the isViewpoint field is FALSE, the user uses the other fields to determine the listener position. The default value is TRUE. |

## 2. AcousticProperties

Definition: determines acoustic effects including surface reflection, physical phenomena such as absorption, specular, diffuse and refraction coefficient of materials.

Attributes:

| | |
|---|---|
| absorption | specifies the sound absorption coefficient of a surface which is the ratio of the sound intensity absorbed or otherwise not reflected by a specific surface that of the initial sound intensity. This characteristic depends on the nature and thickness of the material. Particularly, the sound is absorbed when it encounters fibrous or porous materials, panels that have some flexibility, volumes of air that resonate, openings in the room boundaries (e.g. doorway). Moreover, the absorption of sound by a particular material/panel depends on the frequency and angle of incidence of the sound wave. |
| specular | describes the sound specular coefficient, which is one of the physical phenomena of sound that occurs when a sound wave strikes a plane surface, and a part of the sound energy is reflected back into space but the angle of reflection is equal to the angle of incidence. |
| diffuse | determines the sound diffusion coefficient, which aims to measure the degree of scattering produced on reflection. Specifically, it is produced in the same way as the specular reflection, but in this case, the sound wavelength is comparable with the corrugation dimensions of an irregular reflection surface and the incident sound wave will be scattered in all directions. In other words, it is a measure of the surface's ability to uniformly scatter in all directions. |
| refraction | describes the sound refraction coefficient of a medium, which determines the propagation speed of the wave. This, for a wave traveling from medium one into |

| | |
|---|---|
| | medium two, then the ratio of the refractive indices is equal to the inverse of the velocity ratios. |

## 3. SpatialSound (~ Web Audio API: PannerNode)

PannerNode: represents a processing node which positions / spatializes an incoming audio stream in three-dimensional space. The spatialization is in relation to the ListenerPoint.

Attributes:

| | |
|---|---|
| positionX , positionY , positionZ | The positionX , positionY , and positionZ fields set the x, y, z coordinates position of the audio source in a 3D Cartesian system. The default value is 0,0,0. |
| orientationX , orientationY , orientationZ | The orientationX , orientationY , and orientationZ describe the x, y, z components of the vector of the direction the audio source is pointing in 3D Cartesian coordinate space. Depending on how directional the sound is (controlled by the cone attributes), a sound pointing away from the listener can be very quiet or completely silent. The default value is 1,0,0. |
| coneInnerAngle | is an angle, in degrees, inside of which there will be no volume reduction. The default value is 360. The behavior is undefined if the angle is outside the interval [0, 360]. |
| coneOuterAngle | is an angle, in degrees, outside of which the volume will be reduced to a constant value of coneOuterGain. The default value is 360. The behavior is undefined if the angle is outside the interval [0, 360]. |
| coneOuterGain | is the gain outside of the coneOuterAngle. The default value is 0. It is a linear value (not dB) in the range [0, 1]. |
| distanceModel | The distanceModel field specifies the distance model used by this SpatialSound. It is an enumerated value determining which algorithm to use to reduce the volume of the audio source as it moves away from the listener. The possible values are:<br>a. Linear: A linear distance model calculating the gain induced by the distance according to: 1 - rolloffFactor * (distance - refDistance) / (maxDistance - refDistance)<br>b. Inverse: An inverse distance model calculating the gain induced by the distance according to: refDistance / (refDistance + rolloffFactor * (Math.max(distance, refDistance) - refDistance))<br>c. Exponential: An exponential distance model calculating the gain induced by the distance according to: pow((Math.max(distance, refDistance) / refDistance, -rolloffFactor)<br>The default value is "inverse". |
| maxDistance | is the maximum distance between source and listener, after which the volume will not be reduced any further. The default value is 10000. |
| panningModel | The panningModel field specifies the panning model used by this SpatialSound. The possible values are<br>a. equalpower: Represents the equal-power panning algorithm, generally regarded as simple and efficient.<br>b. Head-Related Transfer Function (HRTF): Renders a stereo output of higher quality than equalpower — it uses a convolution with measured impulse responses from human subjects. |

| | |
|---|---|
| | The default value is "equalpower". |
| refDistance | is a reference distance for reducing volume as source moves further from the listener. The default value is 1. |
| rolloffFactor | describes how quickly the volume is reduced as source moves away from listener. The default value is 1. |
| Gain **(extra in SpatialSound)** | represents a change in volume. The default value is 1. |
| Source **(extra in SpatialSound)** | specifies the sound source for the Sound node. If the source field is not specified, the Sound node will not emit audio. The source field shall specify either an AudioClip node or a MovieTexture node. If a MovieTexture node is specified as the sound source, the MovieTexture shall refer to a movie format that supports sound (EXAMPLE MPEG-1Systems, see ISO/IEC 11172-1). |

`<!-- Heritage from AudioNode -->`

| | |
|---|---|
| numberOfInputs | represents the number of inputs feeding the node. |
| numberOfOutputs | represents the number of outputs coming out of the node. |
| channelCount | represents an integer used to determine how many channels are used when up-mixing and down-mixing connections to any inputs to the node. |
| `channelCountMode` | represents an enumerated value describing the way channels must be matched between the node's inputs and outputs. |
| `channelInterpretation` | represents an enumerated value describing the meaning of the channels. This interpretation will define how audio up-mixing and down-mixing will happen.<br>The possible values are "speakers" or "discrete". |

## 4. AudioBufferSource (~ Web Audio API: AudioBuffer & AudioBufferSourceNode)

AudioBuffer: represents a short audio asset residing in memory.

Attributes:

| | |
|---|---|
| duration | Duration of the Pulse Code Modulation (PCM) audio data (in seconds).<br>PCM describes a process that's used to convert analog audio signals into digital audio signals. |
| length | Length of the PCM audio data (in sample-frames). |
| numberOfChannels | The number of discrete audio channels. |
| sampleRate | The sample-rate for the PCM audio data (in samples per second). |

AudioBufferSourceNode: represents an audio source from an in-memory audio asset in an AudioBuffer.

Attributes:

| | |
|---|---|
| buffer | represents a memory-resident audio asset (for one-shot sounds and other short audio clips). Its format is non-interleaved 32-bit linear floating-point PCM values with a normal range of [−1,1], but values are not limited to this range. It can contain one or more channels. Typically, it would be expected that the length of the PCM data would be fairly short (usually somewhat less than a minute). For |

| | |
|---|---|
| | longer sounds, such as music soundtracks, streaming should be used with the <audio> HTML element and AudioClip. |
| detune | modulate the speed at which is rendered the audio stream (in cents).<br>For example, values of +100 and -100 detune the source up or down by one semitone, while +1200 and -1200 detune it up or down by one octave. |
| loop | indicates if the audio asset must be replayed when the end of the AudioBuffer is reached. |
| loopEnd | indicates the time (in seconds) at which playback of the AudioBuffer stops and loops back to the time indicated by loopStart, if loop is true. |
| loopStart | indicates the time (in seconds) at which playback of the AudioBuffer must begin when loop is true. |
| playbackRate | the speed at which to render the audio stream. |

`<!-- Heritage from AudioNode -->`

| | |
|---|---|
| numberOfInputs | represents the number of inputs feeding the node. |
| numberOfOutputs | represents the number of outputs coming out of the node. |
| channelCount | represents an integer used to determine how many channels are used when up-mixing and down-mixing connections to any inputs to the node. |
| `channelCountMode` | represents an enumerated value describing the way channels must be matched between the node's inputs and outputs. |
| `channelInterpretation` | represents an enumerated value describing the meaning of the channels. This interpretation will define how audio up-mixing and down-mixing will happen.<br>The possible values are "speakers" or "discrete". |

## 5. OscillattorSource (~ Web Audio API: OscillatorNode)

OscillatorNode: represents an audio source generating a periodic waveform. It can replace the AudioBufferSourceNode. It enables us to create our own synths.

Attributes:

| | |
|---|---|
| detune | A detuning value (in cents) which will offset the frequency by the given amount. |
| frequency | The frequency (in Hertz) of the periodic waveform. Its default value is 440. |
| type | The shape of the periodic waveform. ("sine", "square", "sawtooth", "triangle", "custom") |

`<!-- Heritage from AudioNode -->`

| | |
|---|---|
| numberOfInputs | represents the number of inputs feeding the node. |
| numberOfOutputs | represents the number of outputs coming out of the node. |
| channelCount | represents an integer used to determine how many channels are used when up-mixing and down-mixing connections to any inputs to the node. |
| `channelCountMode` | represents an enumerated value describing the way channels must be matched between the node's inputs and outputs. |

| | |
|---|---|
| channelInterpretation | represents an enumerated value describing the meaning of the channels. This interpretation will define how audio up-mixing and down-mixing will happen.<br>The possible values are "speakers" or "discrete". |

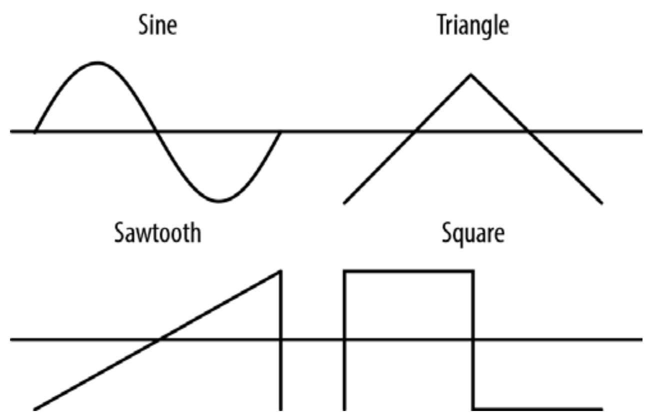



*Figure 1: Types of basic soundwave shapes that the oscillator can generate*

## 6. StreamAudioSource (~ Web Audio API: MediaStreamAudioSourceNode)

MediaStreamAudioSourceNode: operates as an audio source whose media is received from a MediaStream obtained using the WebRTC or Media Capture and Streams APIs. This media could be from a microphone or from a remote peer on a WebRTC call.

Attributes:

| | |
|---|---|
| mediaStream | represents a memory-resident audio asset (for one-shot sounds and other short audio clips). Its format is non-interleaved 32-bit linear floating-point PCM values with a normal range of [−1,1], but values are not limited to this range. It can contain one or more channels. Typically, it would be expected that the length of the PCM data would be fairly short (usually somewhat less than a minute). For longer sounds, such as music soundtracks, streaming should be used with the <audio> HTML element and AudioClip. |

```
<!-- Heritage from AudioNode -->
```

| | |
|---|---|
| numberOfInputs | represents the number of inputs feeding the node. |
| numberOfOutputs | represents the number of outputs coming out of the node. |
| channelCount | represents an integer used to determine how many channels are used when up-mixing and down-mixing connections to any inputs to the node. |
| channelCountMode | represents an enumerated value describing the way channels must be matched between the node's inputs and outputs. |
| channelInterpretation | represents an enumerated value describing the meaning of the channels. This interpretation will define how audio up-mixing and down-mixing will happen.<br>The possible values are "speakers" or "discrete". |

MediaElementAudioSourceNode: represents an audio source from an HTML5 <audio> or <video> element. → AudioClip

Attributes:

| | |
|---|---|
| mediaElement | The HTMLMediaElement (HTMLVideoElement and HTMLAudioElement) used when constructing this MediaStreamAudioSourceNode.<br>HTMLVideoElement: provides special properties and methods for manipulating video objects.<br>HTMLAudioElement: provides access to the properties of <audio> elements |

Web Audio API makes a clear distinction between buffers and source nodes, buffers are like records and sources are like play-heads. For example, if you want multiple bouncing ball, you need to load the bounce buffer only once and schedule multiple sources of playback.

When you want to use a soundfile as your audio source, you need to load your soundfile into a AudioBuffer. An AudioBuffer represents a reference to a soundfile and can be used by multiple BufferSourceNodes for playback. The AudioBuffer can be thought of as a record and a BufferSourceNode can be thought of as a record player.

AudioBuffer is designed to hold small audio snippets, typically less than 45 s. For longer sounds, objects implementing the MediaElementAudioSourceNode are more suitable. This interface represents an audio source consisting of an HTML5 <audio> or <video> element.

This small example applies a low-pass filter to the <audio> tag:

```
function onLoad() {
    var audio = new Audio();
    source = context.createMediaElementSource(audio);
    var filter = context.createBiquadFilter();
    filter.type = filter.LOWPASS;
    filter.frequency.value = 440;
    source.connect(this.filter);
    filter.connect(context.destination);
    audio.src = 'http://example.com/the.mp3';
    audio.play();
    }
```

**AudioBuffer**→ record

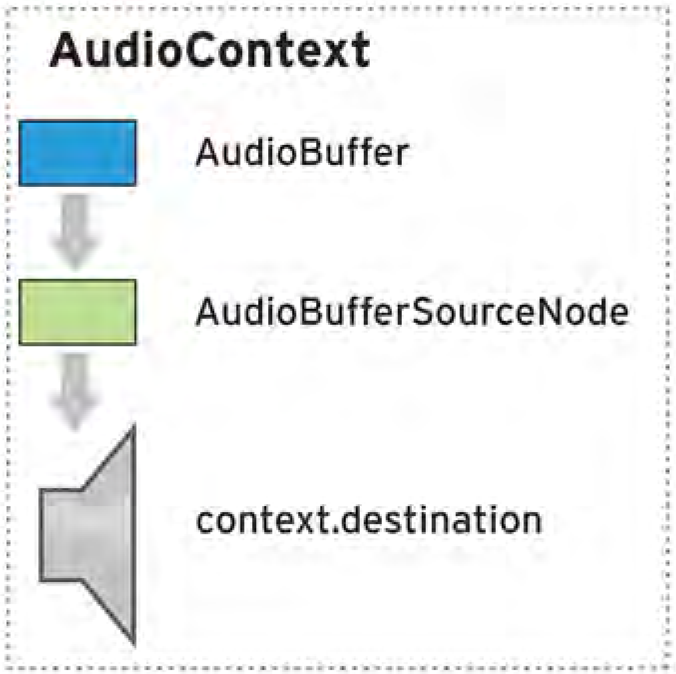
**BufferSourceNode** → record player

Example:

```
// Fix up prefixing
window.AudioContext = window.AudioContext || window.webkitAudioContext;
var context = new AudioContext();

function playSound(buffer) {
```

```
// creates a sound source
var source = context.createBufferSource();
// tell the source which sound to play
source.buffer = buffer;
// connect the source to the context's destination (the speakers)
source.connect(context.destination);
// play the source now
source.start(0);}
```



## 7. MicrophoneSource

Definition: captures input from a built-in (physical) microphone.

Attributes:

| | |
|---|---|
| isActive | A Boolean value that returns true if the device is active, or false otherwise. |
| mediaDevicesid | A unique identifier for the represented device. |

## 8. AudioDestination (~ Web Audio API: AudioDestinationNode)

AudioDestinationNode: represents the final audio destination and is what the user will ultimately hear - usually the speakers of user device.

Attributes:

| | |
|---|---|
| maxChannelCount | The maximum number of channels. An AudioDestinationNode representing the audio hardware end-point (the normal case) can potentially output more than 2 channels of audio if the audio hardware is multi-channel. maxChannelCount is the maximum number of channels that this hardware is capable of supporting. |

```
<!-- Heritage from AudioNode -->
```

| | |
|---|---|
| numberOfInputs | represents the number of inputs feeding the node. |
| numberOfOutputs | represents the number of outputs coming out of the node. |
| channelCount | represents an integer used to determine how many channels are used when up-mixing and down-mixing connections to any inputs to the node. |
| channelCountMode | represents an enumerated value describing the way channels must be matched between the node's inputs and outputs. |
| channelInterpretation | represents an enumerated value describing the meaning of the channels. This interpretation will define how audio up-mixing and down-mixing will happen.<br>The possible values are "speakers" or "discrete". |

Example:

```
var audioCtx = new AudioContext();
var source = audioCtx.createMediaElementSource(myMediaElement);
source.connect(gainNode);
gainNode.connect(audioCtx.destination);
```

## 9. StreamAudioDestination (~ Web Audio API: MediaStreamAudioDestinationNode)

MediaStreamAudioDestinationNode: is an audio destination representing a MediaStream with a single MediaStreamTrack whose kind is "audio".

Attributes:

| | |
|---|---|
| stream | represents a memory-resident audio asset (for one-shot sounds and other short audio clips). Its format is non-interleaved 32-bit linear floating-point PCM values with a normal range of [−1,1], but values are not limited to this range. It can contain one or more channels. Typically, it would be expected that the length of the PCM data would be fairly short (usually somewhat less than a minute). For longer sounds, such as music soundtracks, streaming should be used with the <audio> HTML element and AudioClip. |

```
<!-- Heritage from AudioNode -->
```

| | |
|---|---|
| numberOfInputs | represents the number of inputs feeding the node. |
| numberOfOutputs | represents the number of outputs coming out of the node. |
| channelCount | represents an integer used to determine how many channels are used when up-mixing and down-mixing connections to any inputs to the node. |
| `channelCountMode` | represents an enumerated value describing the way channels must be matched between the node's inputs and outputs. |
| `channelInterpretation` | represents an enumerated value describing the meaning of the channels. This interpretation will define how audio up-mixing and down-mixing will happen.<br>The possible values are "speakers" or "discrete". |

## 10. BiquadFilter (~ Web Audio API: BiquadFilterNode)

BiquadFilterNode: represent different kinds of filters, tone control devices, and graphic equalizers.

Attributes:

| | |
|---|---|
| Q | Quality Factor (Q) of the filter. The default value is 1 |
| detune | a detune value, in cents, for the frequency. The default value is 0. |
| frequency | the frequency at which the BiquadFilterNode will operate, in Hz. The default value is 350. |
| gain | the gain of the filter. Its value is in dB units. The gain is only used for **lowshelf**, **highshelf**, and **peaking** filters. The default value is 0. |
| type | the type of this BiquadFilterNode. Its default value is "lowpass". |

Type→ The meaning of the different properties (frequency, detune and Q) differs depending on the type of the filter you use There are many kinds of filters that can be used to achieve certain kinds of effects:

**"lowpass"**: Makes sounds more muffled

**"highpass"**: Makes sounds more tinny

**"bandpass"**: Cuts off lows and highs (e.g., telephone filter)

**"lowshelf"**: Affects the amount of bass in a sound (like the bass knob on a stereo)

**"highshelf"**: Affects the amount of treble in a sound (like the treble knob on a stereo)

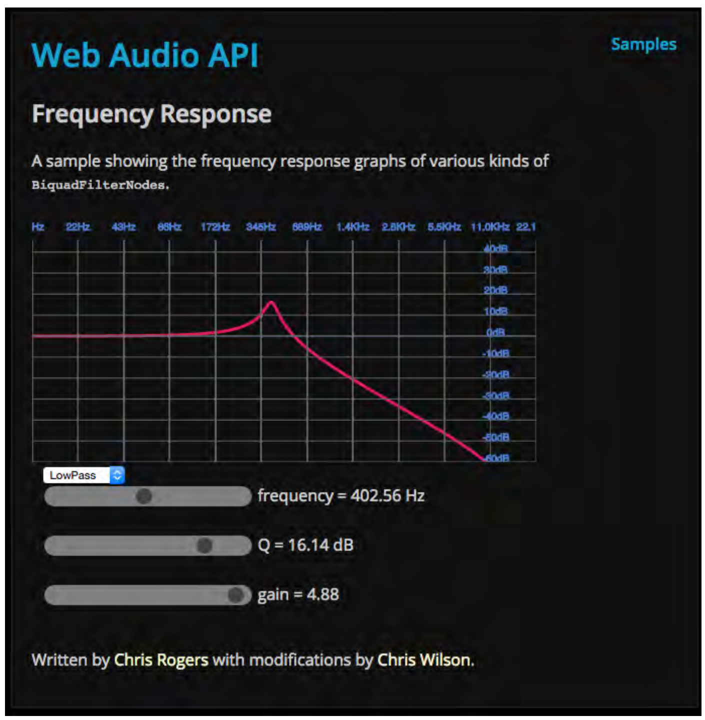
**"peaking"**: Affects the amount of midrange in a sound (like the mid knob on a stereo)

**"notch"**: Removes unwanted sounds in a narrow frequency range

**"allpass"**: Creates phaser effects

`<!-- Heritage from AudioNode -->`

| | |
|---|---|
| numberOfInputs | represents the number of inputs feeding the node. |
| numberOfOutputs | represents the number of outputs coming out of the node. |
| channelCount | represents an integer used to determine how many channels are used when up-mixing and down-mixing connections to any inputs to the node. |
| channelCountMode | represents an enumerated value describing the way channels must be matched between the node's inputs and outputs. |
| channelInterpretation | represents an enumerated value describing the meaning of the channels. This interpretation will define how audio up-mixing and down-mixing will happen.<br>The possible values are "speakers" or "discrete". |




## 11. Convolver (~Web Audio API: ConvolverNode)

ConvolverNode: performs a Linear Convolution on a given AudioBuffer, often used to achieve a reverb effect.

Examples of effects that you can get out of the convolution engine include chorus effects, reverberation, and telephone-like speech.

The idea for producing room effects is to play back a reference sound in a room, record it, and then (metaphorically) take the difference between the original sound and the recorded one. The result of this is an impulse response that captures the effect that the room has on a sound. These impulse responses are painstakingly recorded in very specific studio settings, and doing this on your own requires serious dedication. There are sites that host many of these pre-recorded impulse response files (stored as audio files). The Web Audio API provides an easy way to apply these impulse responses to your sounds using the ConvolverNode.

Attributes:

| | |
|---|---|
| buffer | represents a memory-resident audio asset (for one-shot sounds and other short audio clips). Its format is non-interleaved 32-bit linear floating-point PCM values with a normal range of [−1,1], but values are not limited to this range. It can contain one or more channels. Typically, it would be expected that the length of the PCM data would be fairly short (usually somewhat less than a minute). For longer sounds, such as music soundtracks, streaming should be used with the <audio> HTML element and AudioClip. |
| normalize | a boolean that controls whether the impulse response from the buffer will be scaled by an equal-power normalization when the buffer attribute is set, or not. |

<!-- Heritage from AudioNode -->

| | |
|---|---|
| numberOfInputs | represents the number of inputs feeding the node. |
| numberOfOutputs | represents the number of outputs coming out of the node. |
| channelCount | represents an integer used to determine how many channels are used when up-mixing and down-mixing connections to any inputs to the node. |
| channelCountMode | represents an enumerated value describing the way channels must be matched between the node's inputs and outputs. |
| channelInterpretation | represents an enumerated value describing the meaning of the channels. This interpretation will define how audio up-mixing and down-mixing will happen.<br>The possible values are "speakers" or "discrete". |

The convolver node “smushes” the input sound and its impulse response by computing a convolution, a mathematically intensive function. The result is something that sounds as if it was produced in the room where the impulse response was recorded. In practice, it often makes sense to mix the original sound (called the dry mix) with the convolved sound (called the wet mix), and use an equal-power crossfade to control how much of the effect you want to apply.

## 12. Delay (~Web Audio API: DelayNode)

DelayNode: causes a delay between the arrival of an input data and its propagation to the output.

Attributes:

| | |
|---|---|
| delayTime | represents the amount of delay (in seconds) to apply. Its default value is 0 (no delay). |

<!-- Heritage from AudioNode -->

| | |
|---|---|
| numberOfInputs | represents the number of inputs feeding the node. |
| numberOfOutputs | represents the number of outputs coming out of the node. |

| | |
|---|---|
| channelCount | represents an integer used to determine how many channels are used when up-mixing and down-mixing connections to any inputs to the node. |
| channelCountMode | represents an enumerated value describing the way channels must be matched between the node's inputs and outputs. |
| channelInterpretation | represents an enumerated value describing the meaning of the channels. This interpretation will define how audio up-mixing and down-mixing will happen.<br>The possible values are "speakers" or "discrete". |

## 13. DynamicsCompressor (~Web Audio API: DynamicsCompressorNode)

DynamicsCompressorNode: implements a dynamics compression effect.

Dynamics compression is very commonly used in <u>musical production</u> and <u>game audio</u>. It <u>lowers</u> the volume of the <u>loudest</u> parts of the signal and <u>raises</u> the volume of the <u>softest</u> parts. Overall, a louder, richer, and fuller sound can be achieved. It is especially important in games and musical applications where <u>large numbers of individual sounds</u> are played simultaneous to control the overall signal level and help avoid clipping (distorting) the audio output to the speakers.

Attributes:

| | |
|---|---|
| attack | the amount of time (in seconds) to reduce the gain by 10dB. Its default value is .003 |
| knee | contains a decibel value representing the range above the threshold where the curve smoothly transitions to the compressed portion. Its default value is 30 |
| ratio | represents the amount of change, in dB, needed in the input for a 1 dB change in the output. Its default value is 12 |
| reduction | represents the amount of gain reduction currently applied by the compressor to the signal |
| release | Represents the amount of time (in seconds) to increase the gain by 10dB. Its default value is 0.25 |
| threshold | represents the decibel value above which the compression will start taking effect. Its default value is -24 |

```
<!-- Heritage from AudioNode -->
```

| | |
|---|---|
| numberOfInputs | represents the number of inputs feeding the node. |
| numberOfOutputs | represents the number of outputs coming out of the node. |
| channelCount | represents an integer used to determine how many channels are used when up-mixing and down-mixing connections to any inputs to the node. |
| channelCountMode | represents an enumerated value describing the way channels must be matched between the node's inputs and outputs. |
| channelInterpretation | represents an enumerated value describing the meaning of the channels. This interpretation will define how audio up-mixing and down-mixing will happen.<br>The possible values are "speakers" or "discrete". |

## 14. Gain (~Web Audio API : GainNode)

GainNode: represents a change in volume.

Attributes:

| | |
|---|---|
| gain | represents the amount of gain to apply. Its default value is 1 |

`<!-- Heritage from AudioNode -->`

| | |
|---|---|
| numberOfInputs | represents the number of inputs feeding the node. |
| numberOfOutputs | represents the number of outputs coming out of the node. |
| channelCount | represents an integer used to determine how many channels are used when up-mixing and down-mixing connections to any inputs to the node. |
| `channelCountMode` | represents an enumerated value describing the way channels must be matched between the node's inputs and outputs. |
| `channelInterpretation` | represents an enumerated value describing the meaning of the channels. This interpretation will define how audio up-mixing and down-mixing will happen.<br>The possible values are "speakers" or "discrete". |

## 15. WaveShaper (~Web Audio API: WaveShaperNode)

WaveShaperNode: represents a non-linear distorter. It uses a curve to apply a wave shaping distortion to the signal. Beside obvious distortion effects, it is often used to add a warm feeling to the signal.

Attributes:

| | |
|---|---|
| curve | is an Array of floats numbers describing the distortion to apply |
| oversample | specifies what type of oversampling (if any) should be used when applying the shaping curve. The default value is "**none**", meaning the curve will be applied directly to the input samples. A value of "2x" or "4x" can improve the quality of the processing by avoiding some aliasing, with the "4x" value yielding the highest quality. For some applications, it's better to use no oversampling in order to get a very precise shaping curve. |

```
<!-- Heritage from AudioNode -->
```

| | |
|---|---|
| numberOfInputs | represents the number of inputs feeding the node. |
| numberOfOutputs | represents the number of outputs coming out of the node. |
| channelCount | represents an integer used to determine how many channels are used when up-mixing and down-mixing connections to any inputs to the node. |
| channelCountMode | represents an enumerated value describing the way channels must be matched between the node's inputs and outputs. |
| channelInterpretation | represents an enumerated value describing the meaning of the channels. This interpretation will define how audio up-mixing and down-mixing will happen.<br>The possible values are "speakers" or "discrete". |

## 16. PeriodicWave (~Web Audio API: PeriodicWave)

PeriodicWave: defines a periodic waveform that can be used to shape the output of an OscillatorNode.

Attributes:

| | |
|---|---|
| none | |

## 17. Analyser (~Web Audio API: AnalyserNode)

AnalyserNode: ables to provide real-time frequency and time-domain analysis information. It passes the audio stream unchanged from the input to the output, but allows you to take the generated data, process it, and create audio visualizations.

Attributes:

| | |
|---|---|
| fftSize | presents the size of the FFT (Fast Fourier Transform) to be used to determine the frequency domain (in sample-frames). |
| frequencyBinCount | is half that of the FFT size. This generally equates to the number of data values that you will have to play with for the visualization. |
| maxDecibels | is the maximum power value in the scaling range for the FFT analysis data for conversion to unsigned byte values. The default value is -30. |

| | |
|---|---|
| minDecibels | is the minimum power value in the scaling range for the FFT analysis data for conversion to unsigned byte values. The default value is -100. |
| smoothingTimeConstant | represents the averaging constant with the last analysis frame — basically, it makes the transition between values over time smoother. |

<!-- Heritage from AudioNode -->

| | |
|---|---|
| numberOfInputs | represents the number of inputs feeding the node. |
| numberOfOutputs | represents the number of outputs coming out of the node. |
| channelCount | represents an integer used to determine how many channels are used when up-mixing and down-mixing connections to any inputs to the node. |
| channelCountMode | represents an enumerated value describing the way channels must be matched between the node's inputs and outputs. |
| channelInterpretation | represents an enumerated value describing the meaning of the channels. This interpretation will define how audio up-mixing and down-mixing will happen.<br>The possible values are "speakers" or "discrete". |

FFT converts a signal into individual spectral components and thereby provides frequency information about the signal.

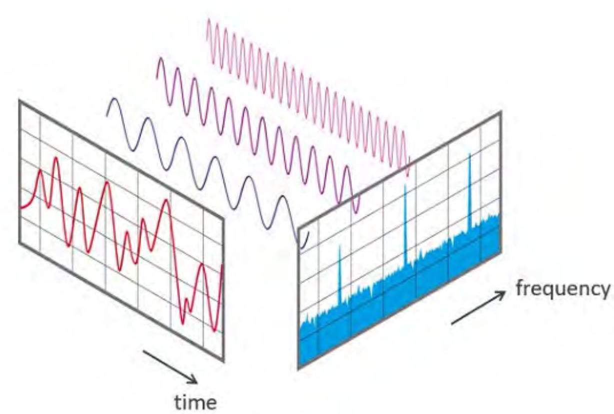



*Figure 2: View of a signal in the time and frequency domain*

## 18. ChannelSplitter (~Web Audio API: ChannelSplitterNode)

ChannelSplitterNode: separates the different channels of an audio source into a set of mono outputs.

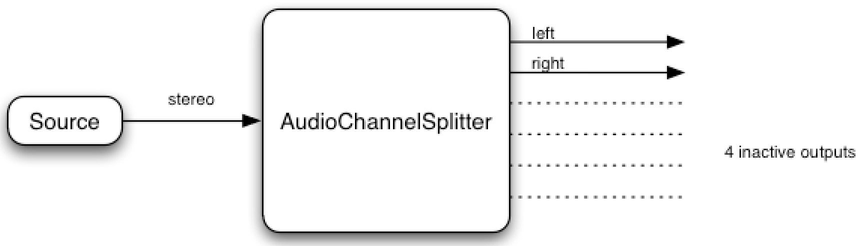
Attributes:

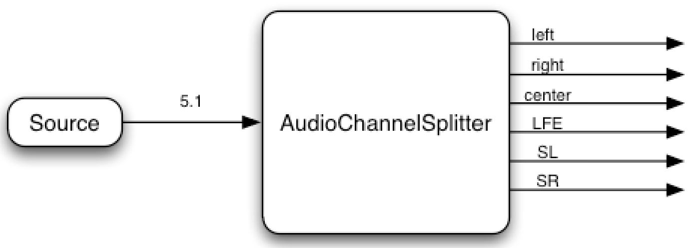
| | |
|---|---|
| none | |

<!-- Heritage from AudioNode -->

| | |
|---|---|
| numberOfInputs | represents the number of inputs feeding the node. |
| numberOfOutputs | represents the number of outputs coming out of the node. |

| | |
|---|---|
| channelCount | represents an integer used to determine how many channels are used when up-mixing and down-mixing connections to any inputs to the node. |
| `channelCountMode` | represents an enumerated value describing the way channels must be matched between the node's inputs and outputs. |
| `channelInterpretation` | represents an enumerated value describing the meaning of the channels. This interpretation will define how audio up-mixing and down-mixing will happen.<br>The possible values are "speakers" or "discrete". |




## 19. ChannelMerger (~Web Audio API: ChannelMergerNode)

ChannelMergerNode: unites different mono inputs into a single output. Often used in conjunction with its opposite.

Attributes:

| | |
|---|---|
| none | |

<!-- Heritage from AudioNode -->

| | |
|---|---|
| numberOfInputs | represents the number of inputs feeding the node. |
| numberOfOutputs | represents the number of outputs coming out of the node. |
| channelCount | represents an integer used to determine how many channels are used when up-mixing and down-mixing connections to any inputs to the node. |
| channelCountMode | represents an enumerated value describing the way channels must be matched between the node's inputs and outputs. |
| channelInterpretation | represents an enumerated value describing the meaning of the channels. This interpretation will define how audio up-mixing and down-mixing will happen.<br>The possible values are "speakers" or "discrete". |