BiquadFilterNode

The `BiquadFilterNode` interface represents a simple low-order filter, and is created using the [BaseAudioContext/createBiquadFilter](#) method. It is an [AudioNode](#) that can represent different kinds of filters, tone control devices, and graphic equalizers. A `BiquadFilterNode` always has exactly one input and one output.

Number of inputs
1
Number of outputs
1
Channel count mode
`"max"`
Channel count
2 (not used in the default count mode)
Channel interpretation
`"speakers"`

Constructor

[**BiquadFilterNode()**](#)
    Creates a new instance of a `BiquadFilterNode` object.

Properties

Inherits properties from its parent, [AudioNode](#) .

> **Note:** Though the `AudioParam` objects returned are read-only, the values they represent are not.

BiquadFilterNode.frequency   Read only

Is an [a-rate](#) [AudioParam](#), a double representing a frequency in the current filtering algorithm measured in hertz (Hz).

BiquadFilterNode.detune   Read only

Is an [a-rate](#) [AudioParam](#) representing detuning of the frequency in [cents](#) .

BiquadFilterNode.Q   Read only

Is an [a-rate](#) [AudioParam](#), a double representing a [Q factor](#) , or *quality factor*.

BiquadFilterNode.gain   Read only

Is an [a-rate](#) [AudioParam](#), a double representing the [gain](#) used in the current filtering algorithm.

BiquadFilterNode.type

Is a string value defining the kind of filtering algorithm the node is implementing.

The meaning of the different parameters depending of the type of the filter (detune has the same meaning regardless, so isn't listed below)

type	Description	frequency	Q	gain
lowpass	Standard second-order resonant lowpass filter with 12dB/octave rolloff. Frequencies below the cutoff pass through; frequencies above it are attenuated.	The cutoff frequency.	Indicates how peaked the frequency is around the cutoff. The greater the value is, the greater is the peak.	*Not used*
highpass	Standard second-order resonant highpass filter with 12dB/octave rolloff. Frequencies below the cutoff are attenuated; frequencies above it pass through.	The cutoff frequency.	Indicates how peaked the frequency is around the cutoff. The greater the value, the greater the peak.	*Not used*

type	Description	frequency	Q	gain
~~type~~	**Description**	~~frequency~~	Q	~~gain~~
bandpass	Standard second-order bandpass filter. Frequencies outside the given range of frequencies are attenuated; the frequencies inside it pass through.	The center of the range of frequencies.	Controls the width of the frequency band. The greater the Q value, the smaller the frequency band.	*Not used*
lowshelf	Standard second-order lowshelf filter. Frequencies lower than the frequency get a boost, or an attenuation; frequencies over it are unchanged.	The upper limit of the frequencies getting a boost or an attenuation.	*Not used*	The boost, in dB, to be applied; if negative, it will be an attenuation.
highshelf	Standard second-order highshelf filter. Frequencies higher than the frequency get a boost or an attenuation; frequencies lower than it are unchanged.	The lower limit of the frequencies getting a boost or an attenuation.	*Not used*	The boost, in dB, to be applied; if negative, it will be an attenuation.
peaking	Frequencies inside the range get a boost or an attenuation; frequencies outside it are unchanged.	The middle of the frequency range getting a boost or an attenuation.	Controls the width of the frequency band. The greater the Q value, the smaller the frequency band.	The boost, in dB, to be applied; if negative, it will be an attenuation.

type	Description	frequency	Q	gain

type	Description	frequency	Q	gain
notch	Standard notch filter, also called a *band-stop* or *band-rejection* filter. It is the opposite of a bandpass filter: frequencies outside the give range of frequencies pass through; frequencies inside it are attenuated.	The center of the range of frequencies.	Controls the width of the frequency band. The greater the Q value, the smaller the frequency band.	*Not used*
allpass	Standard second-order allpass filter. It lets all frequencies through, but changes the phase-relationship between the various frequencies.	The frequency with the maximal group delay, that is, the frequency where the center of the phase transition occurs.	Controls how sharp the transition is at the medium frequency. The larger this parameter is, the sharper and larger the transition will be.	*Not used*

Methods

Inherits methods from its parent, AudioNode .

BiquadFilterNode.getFrequencyResponse()
  From the current filter parameter settings this method calculates the frequency response for frequencies specified in the provided array of frequencies.

Example

See `AudioContext.createBiquadFilter` for example code that shows how to use an `AudioContext` to create a Biquad filter node.

Specifications

Specification
Web Audio API   # BiquadFilterNode

Browser compatibility

Report problems with this compatibility data on GitHub

`BiquadFilterNode`

Chrome	14
Edge	12
Firefox	25
Internet Explorer	**No**
Opera	15
Safari	6
WebView Android	4.4.3
Chrome Android	18
Firefox for Android	25
Opera Android	14
Safari on iOS	6
Samsung Internet	1.0

`BiquadFilterNode()` constructor

Chrome	55
Edge	79
Firefox	53
Internet Explorer	**No**
Opera	42

Safari	14.1
WebView Android	55
Chrome Android	55
Firefox for Android	53
Opera Android	42
Safari on iOS	14.5
Samsung Internet	6.0

Q

Chrome	14
Edge	12
Firefox	25
Internet Explorer	**No**
Opera	15
Safari	6
WebView Android	4.4.3
Chrome Android	18
Firefox for Android	25
Opera Android	14
Safari on iOS	6
Samsung Internet	1.0

detune

Chrome	14
Edge	12
Firefox	25
Internet Explorer	**No**
Opera	15
Safari	6
WebView Android	37
Chrome Android	18

Firefox for Android	25
Opera Android	14
Safari on iOS	6
Samsung Internet	1.0

frequency

Chrome	14
Edge	12
Firefox	25
Internet Explorer	**No**
Opera	15
Safari	6
WebView Android	4.4.3
Chrome Android	18
Firefox for Android	25
Opera Android	14
Safari on iOS	6
Samsung Internet	1.0

gain

Chrome	14
Edge	12
Firefox	25
Internet Explorer	**No**
Opera	15
Safari	6
WebView Android	4.4.3
Chrome Android	18
Firefox for Android	25
Opera Android	14
Safari on iOS	6

Samsung Internet	1.0

getFrequencyResponse

Chrome	14
Edge	12
Firefox	25
Internet Explorer	**No**
Opera	15
Safari	6
WebView Android	4.4.3
Chrome Android	18
Firefox for Android	25
Opera Android	14
Safari on iOS	6
Samsung Internet	1.0

type

Chrome	14
Edge	12
Firefox	25
Internet Explorer	**No**
Opera	15
Safari	6
WebView Android	4.4.3
Chrome Android	18
Firefox for Android	25
Opera Android	14
Safari on iOS	6
Samsung Internet	1.0

getFrequencyResponse

Full support

No support

See implementation notes.

See also

- [Using the Web Audio API](#)

Last modified: Sep 14, 2021, [by MDN contributors](#)