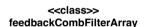
class ReverbEffect

<<class>> ReverbEffect

- + allpassReverberatorChainInstance : allpassReverberatorChain
- + feedbackCombFilterArrayInstance : feedbackCombFilterChain
- + mix : coeff t
- + initReverbEffect(): void
- + **process**(data_t inputData, data_t* outputData) : void
- + tuneMix(coeff t newMix): void
- + tuneReverbSize(reverbSize_t

reverbSize): void



- + feedbackCombFilterSections : feedbackCombFilterSection[]
- + aValuesCombFilter : coeff t[]
- + kValuesCombFilter : int[]
- + initfeedbackCombFilterArray(): void
- + process(data_t inputData, data_t*
- outputData): void
- + tuneFeedbackCombFilter(reverbSize_t

reverbSize): void

<<class>> feedbackCombFilterSection

- + aValues : coeff_t + kValue: int
- + initfeedbackCombFilterArray(): void
- + process(data_t inputData, data_t*
- outputData): void
- + tuneFeedbackCombFilter(reverbSize_t

reverbSize): void

<<class>> allpassReverberatorChain

- + allpassReverberatorSections : allpassReverberatorSection[]
- + aValuesAllPass : coeff_t[]
- + delayValuesAllPass : int[]
- + initReverbEffect(): void
- + process(data_t inputData, data_t*
- outputData): void
- $+ tune All pass Reverberator (reverb Size_t$

reverbSize) : void

<<class>> allpassReverberatorSection

- + aValue: coeff_t + delayValues : int
- + initReverbEffect(): void
- + **process**(data_t inputData, data_t* outputData) : void
- $+ tune All pass Reverberator (reverb Size_t$

reverbSize) : void