

init
initOnDevice:
initOnDevice:withParameters:
setDevice:
device
parameters
free

Activating and Deactivating activate

deactivate

Controlling the stream abort:

abortAtTime:
pause:
pauseAtTime:
resume:
resumeAtTime:

Querying the object bytesProcessed

abort:sender

Stops the NXSoundStream's playback or recording (after the current DMA transfer buffer has been processed and its remaining enqueued buffers, and sends the object's delegate a `soundStreamDidAbort:deviceReason` message. The `deviceReason` argument is ignored. It's included so the method can be used in Interface Builder as an action method. Follow this method with an invocation of `lastError` to see if an error occurred. If the NXSoundStream is inactive, this does nothing. Returns `self`.

See also: `abortAtTime:`

(NXSoundDeviceError)abortAtTime:(NXSoundStreamTime *)time

Schedules the NXSoundStream to be aborted (as described in the `abort:` method, above) at the time specified by the `time` structure. The NXSoundStreamTime structure is a cover for the familiar `timeval` structure:

abort:

(NXSoundDeviceError)activate

Adds the NXSoundStream to the sound driver's list of active streams. You must invoke this method before you can enqueue buffers on the stream. When you've finished recording or playing, you should send the NXSoundStream a `stop` message. The NXSoundStream must be connected to an NXSoundDevice for this method to have any effect. Returns `0` if successful, or an error code is returned.

See also: `deactivate`, `isActive`

(unsigned int)bytesProcessed

Returns the number of bytes of sound that the NXSoundStream has recorded or played since it was activated. Returns `0` if the object is inactive, or if an error occurs. You should follow this method with an invocation of `lastError` to see if an error occurred.

(NXSoundDeviceError)deactivate

Aborts the NXSoundStream's current activity and removes the object from the sound driver's list of active streams.

See also: `activate`, `isActive`

Returns the NXSoundDevice object that the NXSoundStream is connected to.

See also: `initWithDevice:`, `setDevice:`

`free`

Deactivates and frees the NXSoundStream.

`init`

Initializes the NXSoundStream without connecting it to an NXSoundDevice. Returns self.

See also: `initWithDevice:`, `setDevice:`

`initWithDevice:aDevice`

Initializes the NXSoundStream and connects it to `aDevice`, which should be an instance of an NXSoundDevice subclass. Returns self.

See also: `init`

`initWithDevice:aDevice withParameters:(<NXSoundParameters>)params`

Initializes the NXSoundStream, configures it by reading the appropriate parameters from `params`, `aDevice`, which should be an instance of an NXSoundDevice subclass. Returns self.

See also: `init`

`(BOOL)isActive`

Returns YES if the NXSoundStream is currently activate otherwise, NO.

See also: `activate`

`(BOOL)isPaused`

Returns YES if the NXSoundStream is currently paused otherwise, NO.

See also: `pause:`, `isActive`

`(NXSoundDeviceError)lastError`

Returns the NXSoundParameters object that was used to configure the NXSoundStream.

pause:sender

Pauses the NXSoundStream's recording or playback (after the current DMA transfer buffer has been used). It sends a soundStreamDidPause: message to the object's delegate. The argument is ignored. It's included in Interface Builder as an action method. You should follow this method with an invocation of the delegate's delegate to see if an error occurred. If the NXSoundStream isn't currently active or if it's already paused, this does nothing.

See also: pauseAtTime:, resume:

(NXSoundDeviceError)pauseAtTime:(NXSoundStreamTime *)time

Schedules the NXSoundStream to be paused (as described in the pause: method, above) at the time specified by the time parameter. See the abortAtTime: method for an explanation of the NXSoundStreamTime type. It doesn't wait for the stream to pause. A soundStreamDidPause: message is sent to the delegate at the time that the stream is paused. This does nothing if the NXSoundStream isn't currently active. An error code is returned.

See also: pause:, abortAtTime:

resume:sender

Resumes the NXSoundStream's recording or playback (after the current DMA transfer buffer has been used). It sends a soundStreamDidResume: message to the object's delegate. The argument is ignored. It's included in Interface Builder as an action method. You should follow this method with an invocation of the delegate's delegate to see if an error occurred. If the NXSoundStream isn't currently active or if it isn't paused, this does nothing.

See also: resumeAtTime:, pause:

(NXSoundDeviceError)resumeAtTime:(NXSoundStreamTime *)time

Schedules the NXSoundStream to be resumed (as described in the resume: method, above) at the time specified by the time parameter. See the abortAtTime: method for an explanation of the NXSoundStreamTime type. It doesn't wait for the stream to resume. A soundStreamDidResume: message is sent to the NXSoundStream's delegate at the time that the stream is resumed. This does nothing if the NXSoundStream is currently active. An error code is returned.

See also: pause:, abortAtTime:

setDelegate:anObject

Assigns anObject as the NXSoundStream's delegate.

See also: delegate

(port_t)streamPort

Returns the port that the NXSoundStream uses to connect to the sound driver. You can't set this port, but you normally don't need to note its identity this method is provided in case you want to pass the port as an argument to a function such as port_status(). (Note that this device port isn't understood by the old Sound/DSP driver and shouldn't be used as an argument to the sound driver functions.)

soundStream:sender didCompleteBuffer:(int)tag

Invoked when the driver finishes playing or recording the sound buffer identified by tag (as assigned to the soundStream:sender property). The return value is ignored.

See also: recordSize:tag: (NXRecordStream), playBuffer:size:tag: (NXPlayStream)

soundStream:sender didStartBuffer:(int)tag

Invoked when the driver starts playing or recording the sound buffer identified by tag (as assigned to the soundStream:sender property). The return value is ignored.

See also: recordSize:tag: (NXRecordStream), playBuffer:size:tag: (NXPlayStream)

soundStreamDidAbort:sender deviceReserved:(BOOL)flag

Invoked when the driver aborts the stream. If the stream was aborted because the NXSoundDevice is reserved, the flag will be YES, otherwise it will be NO. The return value is ignored.

See also: abort:, abortAtTime:, setReserved: (NXSoundDevice)

soundStreamDidPause:sender

Invoked when the NXSoundStream sender is paused. The return value is ignored.

See also: pause:, pauseAtTime:

soundStreamDidResume:sender

Invoked when the NXSoundStream sender is resumed. The return value is ignored.

See also: resume:, resumeAtTime: