boolValueForParameter:

floatValueForParameter:
getParameters:count:
getValues:count:forParameter:
intValueForParameter:

init
initOnHost:
free

Using a separate thread+ replyThread
+ isUsingSeparateThread
+ setThreadThreshold:
+ setUseSeparateThread:
+ threadThreshold

Examining ports devicePort
+ replyPort
streamOwnerPort

Identifying the host computer host

Configuring the object isReserved
setReserved:
setParameters:
parameters
name
+ setTimeout:
+ timeout

Retrieving peak amplitudes getPeakLeft:right:
clipCount

Setting up streams acceptsContinuousStreamSamplingRates
getStreamChannelCountsLimit:
getStreamSamplingRates:low:high:
getStreamSamplingRates:count:
getStreamDataEncodings:count:

Controlling streams abortStreams:
pauseStreams:
resumeStreams:

Handling errors lastError
+ textForError:

lastError

abortStreams:sender

Aborts all streams that are connected to the NXSoundDevice.  You should check the return value o
invoking this method to see if an error occurred.  Returns self.

abort: (NXSoundStream),  lastError

(BOOL)acceptsContinuousStreamSamplingRates

Returns the number of sample frames that have been clipped since the activation of the oldest conn
occurs when the amplitude of a sample is great than the greatest representable value for this device
reset to 0 when the last stream is deactivated.

getPeakLeft:right:

(port_t)devicePort

Returns the port that the NXSoundDevice uses to communicate with the sound driver. You can't s
and you normally don't need to note its identity this method is provided so you can pass the port as
function such as port_status().

free

Deallocates the NXSoundDevice's ports and frees the object. If the NXSoundDevice had reserved
device, it's made available again.

setReserved:

(unsigned int)getStreamChannelCountLimit

Returns the maximum number of channels of sound data that are accepted by the underlying hardw
you must set the channel count of any NXSoundStream object that's attached to this NXSoundDev
greater than that returned by this method.

getStreamSamplingRates:count:, getStreamSamplingRatesLow:high:, getStreamDataEncodings:

(NXSoundDeviceError)getStreamDataEncodings:
(const NXSoundParameterTag **)encodings
count:(unsigned int *)numEncodings

Returns, by reference in encodings, a list of the data encoding values that are accepted by the unde
encodings are represented by paramater value tags the number of encodings in the list is returned b
numEncodings. You then set an NXSoundStream object's encoding to a value plucked from the li

getStreamChannelCountLimit:, getStreamSamplingRatesLow:high:, getStreamSamplingRates:co

(NXSoundDeviceError)getStreamSamplingRates:(const float **)rates
count:(unsigned int *)numRates

Returns, by reference in rates, a list of the discrete sampling rates that are accepted by the underlyi
number of sampling rates in the list is returned by reference in numRates. You then set an NXSou
sampling rate to a value plucked from the list. You invoke this method only if the NXSoundDevic

Returns, by reference, the lowest and highest sampling rates that are accepted by the underlying ha
an NXSoundStream object's sampling rate to a value in the returned range. You invoke this metho
NXSoundDevice accepts continuous sampling rate values (as determined by the acceptsContinuou
method). If it only accepts discrete values, use the getStreamSamplingRates:count: method to retu

acceptsContinuousStreamSamplingRates, getStreamSamplingRates:count:, getStreamChannelCo
getStreamDataEncodings:count:

(NXSoundDeviceError)getPeakLeft:(float *)leftAmp right:(float *)rightAmp

Returns the most recent peak amplitudes detected by the NXSoundDevice's underlying sound devi
peaks are detected independently for the two channels and returned by reference in leftAmp and rig
monophonic sounds, the same value is returned in both arguments. The peak values returned in the
normalized to fall within (0.0, 1.0), where 0.0 is no amplitude and 1.0 is the maximum amplitude s
format. See the class description for more information on peak detection. An error code is returne

clipCount (NXSoundOut)

(const char *)host

Returns the name of the computer on which the NXSoundDevice was initialized, or nil if it's the lo

initOnHost:

init

Initializes the NXSoundDevice on the machine specified by the NXHost default (normally the loca
sound resources cannot be accessed otherwise returns self.

initOnHost:

initOnHost:(const char *)hostName

Initializes the NXSoundDevice on the machine named hostName. Returns nil if the sound resourc
otherwise returns self.

init

(BOOL)isReserved

Returns YES if the device is reserved for exclusive access by this NXSoundDevice otherwise, retu

setReserved:

(NXSoundDeviceError)lastError

Returns the name of the NXSoundDevice's underlying device, as registered with the network.

(id <NXSoundParameters>)parameters

Returns an object that contains the NXSoundDevice's parameter settings. The parameters take def
freshly initialized NXSoundDeivce.

setParameters:

pauseStreams:sender

Pauses all streams that are connected to the NXSoundDevice. You should check the return value o
invoking this method to see if an error occurred. Returns self.

pause: (NXSoundStream), lastError

resumeStreams:sender

Resumes all streams that are connected to the NXSoundDevice. You should check the return valu
invoking this method to see if an error occurred. Returns self.

resume: (NXSoundStream), lastError

(NXSoundDeviceError)setParameters:(id <NXSoundParameters>)parameterObject

Sets the NXSoundDevice's parameter values to those specified in the argument.

parameters

(NXSoundDeviceError)setReserved:(BOOL)flag

If flag is YES, reserves the underlying device for exclusive access by the NXSoundDevice (even i
by another NXSoundDeviceÐthe current owner is forced to yield). No other application, nor any
within your application, will be able access the device while it's reserved. Any currently active str
this NXSoundDevice instance are aborted. If flag is NO the device is made available to all NXSou
NXSoundDevices are unreserved by default. An error code is returned.

isReserved, soundStreamDidAbort:deviceReserved: (NXSoundStream delegate)

(port_t)streamOwnerPort

Returns the port that the NXSoundDevice uses to connect to the sound driver. You can't set this p
normally don't need to note its identity this method is provided in case you want to pass the port as
function such as port_status().