tOnDevice:anObjectInitializes a newly allocated NXPlayStream

XSoundDeviceError)activatePrepares the NXPlayStream for playback
XSoundDeviceError)playBuffer:(void *)dataPlays a buffer of sound
size:(unsigned int)bytes
tag:(int)aTag

XSoundDeviceError)setGainLeft:(float)leftAmp Sets the NXPlayStream's stereo gain
right:(float)rightAmp
GainLeft:(float *)leftScale Returns the NXPlayStream's gain by reference
right:(float *)rightScale
XSoundDeviceError)getPeakLeft:(float *)leftAmp
right:(float *)rightAmpReturns the most recent peak amplitudes by reference

undStreamDidUnderrun:senderInvoked when the sound driver underruns

XSoundDeviceError)recordSize:(unsigned int)bytes
tag:(int)anIntEnqueues a recording buffer

XSoundDeviceError)sendRecordedDataToDelegate
Sends the current buffer to the delegate

undStreamDidOverrun:senderInvoked when the sound driver overruns
undStreamDidRecordData:(void *)data Delivers a buffer of recorded data
size:(unsigned int)numBytes
forBuffer:(int)tag

ort_t)devicePortReturns the port used to communicate with the sound driver
port_t)replyPortReturns the port to which the sound driver sends messages
ort_t)streamOwnerPortReturns the port used to connect to the sound driver

nst char *)hostReturns the name of the computer on which the NXSoundDevice was initialized

<NXSoundParameters>)parametersReturns the object's parameters
XSoundDeviceError)setParameters:(id <NXSoundParameters>)params
OOL)acceptsContinuousStreamSamplingRates
XSoundDeviceError)getStreamSamplingRatesLow:(float *)lowRate
high:(float *)highRate
XSoundDeviceError)getStreamSamplingRates:(const float **)rates
count:(unsigned int *)numRates
XSoundDeviceError)getStreamDataEncodings:(const NXSoundParameterTag **)encodings
count:(unsigned int *)numEncodings
nsigned int)streamChannelCountLimit
XSoundDeviceError)setReserved:(BOOL)flag
s whether the underlying device is reserved
OOL)isReservedReturns whether the underlying device is reserved
etTimeout:(unsigned int)millisecondsSets the timeout period
unsigned int)timeoutReturns the timeout period

XSoundDeviceError)getPeakLeft:(float *)leftAmp
right:(float *)rightAmpReturns the most recent peak amplitudes by reference

ortStreams:senderAborts all streams connected to this NXSoundDevice
useStreams:senderPauses all streams connected to this NXSoundDevice
umeStreams:senderResumes all streams connected to this NXSoundDevice

XSoundDeviceError)lastErrorReturns the most recent error
const char *)textForError:(NXSoundDeviceError)errorCode
Returns a string that describes the given error

t
tFromSound:aSound
tFromSoundStruct:(SNDSoundStruct *)soundStruct
oid)configureSoundStruct:(SNDSoundStruct *)soundStruct

tInitializes the NXSoundStream
tOnDevice:aDevice withParameters:paramsInitializes the NXSoundStream and connects it to aDevice
eDeactivates and frees the NXSoundStream.

XSoundDeviceError)setDevice:aDeviceConnects the NXSoundStream to aDevice
viceReturns the NXSoundDevice that the NXSoundStream is connected to.

XSoundDeviceError)activateAdds the NXSoundStream to the list of active streams
XSoundDeviceError)deactivateAborts the NXSoundStream's current activity

ort:senderStops the NXSoundStream's playback or recording
XSoundDeviceError)abortAtTime:(NXSoundStreamTime *)time
Schedules the NXSoundStream to be aborted
use:senderPauses the NXSoundStream's recording or playback
XSoundDeviceError)pauseAtTime:(NXSoundStreamTime *)time
Schedules the NXSoundStream to be paused
ume:senderResumes the NXSoundStream's recording or playback
XSoundDeviceError)resumeAtTime:(NXSoundStreamTime *)time
Schedules the NXSoundStream to be resumed

signed int)bytesProcessedReturns the number of bytes of sound that the NXSoundStream has recorded or pla
OOL)isActiveReturns whether the NXSoundStream is currently activate
OOL)isPausedReturns whether the NXSoundStream is currently paused
ort_t)streamPortReturns the port used to connect to the sound driver
XSoundDeviceError)lastErrorReturns the most recent sound device error

Invoked when the driver finishes playing or recording
ndStream:sender didStartBuffer:(int)tagInvoked when the driver starts playing or recording
ndStreamDidAbort:sender Invoked when the driver aborts the stream
deviceReserved:(BOOL)flag
ndStreamDidPause:senderInvoked when the NXSoundStream is paused
ndStreamDidResume:senderInvoked when the NXSoundStream is resumed

tFromSection:(const char *)sectionNameCreates a Sound object from the sectionName section of the sound se
application executable file
tFromPasteboard:(Pasteboard *)thePboardCreates a Sound object from the named pasteboard
tFromSoundfile:(const char *)filenameCreates a Sound object from filename
eFrees the Sound object

Name:(const char *)aNameSet's the Sound object's name
nst char *)nameReturn's the Sound object's name

t)readSoundfile:(const char *)filenameReplaces the Sound's data with that in filename
dSoundFromStream:(NXStream *)streamReplaces the Sound's data with that read from stream
t)writeSoundfile:(const char *)filenameWrites the Sound's data to filename
teSoundToStream:(NXStream *)streamWrites the Sound's data to stream
t)writeToPasteboard:(Pasteboard *)pboardWrites the Sound's data to the named pasteboard

t)convertToFormat:(int)newFormatConverts the Sound's data to the specified format,
samplingRate:(double)newRatesampling rate, and channel count
channelCount:(int)newChannelCount
t)convertToFormat:(int)newFormatConvert's the Sound's data to the specified format

NDSoundStruct *)soundStructReturns the Sound's sound structure
t)soundStructSizeGives the size of the Sound's sound structure
signed char *)dataReturns a pointer to the Sound's sound data
t)dataFormatReturns the Sound's data format
t)dataSizeReturns the size in bytes of the Sound's data
t)channelCountReturns the number of channels of sound
uble)samplingRateReturns the sound data's sampling rate
t)sampleCountReturns the number of sample frames in the sound data
uble)durationReturns the sound's length in seconds
ar *)infoReturns a pointer to the Sound's info string
t)infoSizeReturns the length in bytes of the Sound's info string
OOL)isEmptyReturns whether the Sound contains any sound data
OOL)compatibleWith:aSoundReturns whether the Sound's format is compatiable with that of aSound

t)pausePauses the Sound's activity
se:senderPauses the Sound's activity
OOL)isPlayableReturns whether the Sound can be played
t)playPlays the Sound
y:senderPlays the Sound
t)recordRecords into the Sound
ord:senderRecords into the Sound
t)resumeResumes the Sound's activity
ume:senderResumes the Sound's activity
t)stopStops the Sound's activity
p:senderStops the Sound's activity
t)samplesProcessedReturns the number of sample frames played or recorded
t)statusReturns the Sound's activity code
ndBeingProcessedReturns self
NDSoundStruct *)soundStructBeingProcessed
Returns the sound structure that's being played or recorded
t)processingErrorReturns the last error code that was generated

OOL)isEditableReturns whether the Sound's data can be edited
t)copySamples:aSoundCopies a range of samples from aSound into the receiver
at:(int)startSample
count:(int)sampleCount
t)copySound:aSoundReplaces the Sound's data with that in aSound
t)deleteSamplesRemoves the Sound's data
t)deleteSamplesAt:(int)startSample Removes a range of samples from the Sound's data
count:(int)sampleCount
t)insertSamples:aSound at:(int)startSampleInserts aSound's data into the Sound's data
OOL)needsCompactingReturns whether the Sound's data needs to be compacted
t)compactSamplesCompacts the Sound's data

Delegate:anObjectSets the Sound's delegate object
egateReturns the Sound's delegate
Delegate:(SEL)theMessageSends theMessage to the delegate

Play:senderSent to the delegate when the Sound stops playing
Record:senderSent to the delegate when the Sound stops recording
lError:senderSent to the delegate if an error occurs during recording or playback
lPlay:senderSent to the delegate when the Sound begins to play
lRecord:senderSent to the delegate when the Sound begins to record

tFrame:(const NXRect *)frameRectInitializes the SoundMeter

BackgroundGray:(float)aValueSets the SoundMeter's background color
oat)backgroundGrayReturns the SoundMeter's background color
ForegroundGray:(float)aValueSets the SoundMeter's running bar color
oat)foregroundGrayReturns the color of the running bar
Bezeled:(BOOL)aFlagSets whether a bezeled border is drawn
OOL)isBezeledReturns whether the SoundMeter has a border
PeakGray:(float)aValueSets the SoundMeter's peak bubble color
oat)peakGrayReturns the SoundMeter's peak bubble gray

Sound:aSoundSets the SoundMeter's Sound object
undReturns the Sound object that the SoundMeter is metering
FloatValue:(float)aValueSets the current running value
HoldTime:(float)secondsSets the SoundMeter's peak value hold time in seconds
oat)holdTimeReturns the SoundMeter's peak hold time

oat)floatValueReturns the current running amplitude value

wCurrentValueDraws the SoundMeter's running bar and peak bubble
wSelf:(const NXRect *)rects :(int)rectCountDraws all the components of the SoundMeter

d:(NXTypedStream *)aStreamUnarchives the SoundMeter by reading it from aStream
te:(NXTypedStream *)aStreamArchives the SoundMeter by writing it to aStream

tFrame:(const NXRect *)frameRectInitializes the SoundView
eFrees the SoundView

leToFitFits the sound data within the current frame
BackgroundGray:(float)aGraySets the SoundView's background gray
Bezeled:(BOOL)aFlagSets whether the SoundView has a bezeled border
Continuous:(BOOL)aFlagSets the state of continuous action messages.
Delegate:anObjectSets the SoundView's delegate
DisplayMode:(int)aModeSets the SoundView's display mode
Enabled:(BOOL)aFlagEnables or disables the SoundView
ForegroundGray:(float)aGraySets the SoundView's foreground gray
OptimizedForSpeed:(BOOL)flagSets whether the SoundView's display mechanism is optimized
Sound:aSoundSets the SoundView's Sound object
eToFitResizes the SoundView's frame to maintain a constant reduction factor

oat)backgroundGrayReturns the SoundView's background gray
egateReturns the SoundView's delegate
t)displayModeReturns the SoundView's display mode
oat)foregroundGrayReturns the SoundView's foreground gray
Selection:(int *)firstSample
size:(int *)sampleCountReturns the selection by reference
OOL)isAutoScaleReturns whether the SoundView is in autoscaling mode
OOL)isBezeledReturns whether the SoundView has a bezeled border
OOL)isContinuousReturns whether the SoundView responds to mouse-dragged events
OOL)isEnabledReturns whether the SoundView is enabled
OOL)isOptimizedForSpeedReturns whether the SoundView is optimized for speedy display
oat)reductionFactorReturns the SoundView's reduction factor
undReturns the SoundView's Sound object.

Selection:(int)firstSample size:(int)sampleCount
Sets the selection
OOL)isEditableReturns whether the SoundView's data can be edited.
Editable:(BOOL)aFlagSets whether the SoundView can be edited

steboard:thePasteboard provideData:(const char *)pboardType
Places the SoundView's sound on the given pasteboard
dSelectionFromPasteboard:thePasteboardReplaces the SoundView's current selection
idRequestorForSendType:(NXAtom)sendType
andReturnType:(NXAtom)returnTypeYou never invoke this method
teSelectionToPasteboard:thePasteboard Places a copy of the SoundView's current selection
types:(NXAtom *)pboardTypes on the given pasteboard

Autoscale:(BOOL)aFlagSets the SoundView's automatic scaling mode
ReductionFactor:(float)reductionFactorRecomputes the size of the SoundView's frame

wSelf:(const NXRect *)rects :(int)rectCountDisplays the SoundView's sound data
wSamplesFrom:(int)first to:(int)lastRedisplays the given range of samples
eCursorHides the SoundView's cursor
wCursorDisplays the SoundView's cursor
eTo:(NXCoord)width :(NXCoord)heightSets the width and height of the SoundView's frame

OOL)acceptsFirstResponderReturns YES
comeFirstResponderPromotes the SoundView to first responder
ignFirstResponderResigns the position of first responder

OOL)isPlayableReturns whether the SoundView's sound can be played
y:senderPlay the current selection
ord:senderReplaces the SoundView's current selection
se:senderPauses the current playback or recording
ume:senderResumes the current playback or recording
p:senderStops the SoundView's current recording or playback
ndBeingProcessedReturns the Sound object that's currently being played or recorded into

d:(void *)streamUnarchives the SoundView by reading it from stream
te:(void *)streamArchives the SoundView by writing it to stream

Play:senderSent to the delegate just after the SoundView is played.
Record:senderSent to the delegate just after the SoundView is recorded into.
lError:senderSent to the delegate if an error is encountered .
ectionChanged:senderSent to the delegate when the SoundView's selection changes.
ndDidChange:senderSent to the delegate when the SoundView's sound is edited
lFree:senderSent to the delegate when the SoundView is freed.
lPlay:senderSent to the delegate just before the SoundView's sound is played.
lRecord:senderSent to the delegate just before the SoundView's sound is recorded into.