	initFrame:
Freeing a SoundView instance	free
Modifying the object	scaleToFit
	setBackgroundGray:
	setBezeled:
	setContinuous:
	setDelegate:
	setDisplayMode:
	setEnabled:
	setForegroundGray:
	setOptimizedForSpeed:
	setSound:
	sizeToFit
Querying the object	backgroundGray
	delegate
	displayMode
	foregroundGray
	getSelection:size:
	isAutoScale
	isBezeled
	isContinuous
	isEnabled
	isOptimizedForSpeed
	reductionFactor
	sound
Selecting and editing the sound data	
	copy:
	cut:
	delete:
	mouseDown:
	paste:
	selectAll:
	setSelection:size:
	isEditable
	setEditable:
Pasteboard and Services support	pasteboard:provideData:
	readSelectionFromPasteboard:
	validRequestorForSendType:andReturnType:
	writeSelectionToPasteboard:types:
Modifying the display coordinates	
	setAutoscale:
	setReductionFactor:
Drawing the object	drawSelf::
	drawSamplesFrom:to:
	hideCursor
	showCursor
	sizeTo::
Responding to events	acceptsFirstResponder
	becomeFirstResponder
	resignFirstResponder
Performing the sound data	pause:

didRecord:
hadError:
tellDelegate:
willPlay:
willRecord:

(BOOL)acceptsFirstResponder

If the SoundView is enabled, this returns YES, allowing the SoundView to become the first respon returns NO. This method is automatically invoked by objects defined by the Application Kit you s invoke it directly.

(float)backgroundGray

Returns the SoundView's background gray value (NX_WHITE by default).

becomeFirstResponder

Promotes the SoundView to first responder. You never invoke this method directly. Returns self.

copy:sender

Copies the current selection to the pasteboard. Returns self.

cut:sender

Deletes the current selection from the SoundView, copies it to the pasteboard, and sends a soundD the delegate. The insertion point is positioned to where the selection used to start. Returns self.

delegate

Returns the SoundView's delegate object.

delete:sender

Deletes the current selection from the SoundView's Sound and sends the soundDidChange: messag deletion isn't placed on the pasteboard. Returns self.

(int)displayMode

Returns the SoundView's display mode, one of NX_SOUNDVIEW_WAVE (oscilloscopic display
NX_SOUNDVIEW_MINMAX (minimum/maximum display this is the default).

drawSamplesFrom:(int)first to:(int)last

Redisplays the given range of samples. Return self.

drawSelf:(const NXRect *)rects :(int)rectCount

Displays the SoundView's sound data. The selection is highlighted and the cursor is drawn (if it is
Returns self.

You never send the drawSelf:: message directly to a SoundView object. To cause a SoundView to
of the display messages defined by the View class.

(float)foregroundGray

Returns the SoundView's foreground gray value (NX_BLACK by default).

free

Frees the SoundView but not its Sound object nor its delegate. The willFree: message is sent to th

getSelection:(int *)firstSample size:(int *)sampleCount

Returns the selection by reference. The index of the selection's first sample (counting from 0) is re
The size of the selection in samples is returned in sampleCount. The method itself returns self.

hadError:sender

Used to redirect delegate messages from the SoundView's Sound object you never invoke this met

hideCursor

Hides the SoundView's cursor. This is usually handled automatically. Returns self.

Returns YES if the SoundView is in autoscaling mode, otherwise returns NO.

(BOOL)isBezeled

Returns YES if the SoundView has a bezeled border, otherwise returns NO (the default).

(BOOL)isContinuous

Returns YES if the SoundView responds to mouse-dragged events (as set through setContinuous:)

(BOOL)isEditable

Returns YES if the SoundView's sound data can be edited.

(BOOL)isEnabled

Returns YES if the SoundView is enabled, otherwise returns NO. The mouse has no effect in a dis
default, a SoundView is enabled.

(BOOL)isOptimizedForSpeed

Returns YES if the SoundView is optimized for speedy display. SoundViews are optimized by de

(BOOL)isPlayable

Returns YES if the SoundView's sound data can be played without first being converted.

mouseDown:(NXEvent *)theEvent

Allows a selection to be defined by clicking and dragging the mouse. This method takes control u
While dragging, the selected region is highlighted. On mouse up, the delegate is sent the selection
isContinuous is YES, selectionChanged: messages are also sent while the mouse is being dragged.
method it's invoked automatically in response to the user's actions. Returns self.

paste:sender

Replaces the current selection with a copy of the sound data currently on the pasteboard. If there i
pasteboard data is inserted at the cursor position. The pasteboard data must be compatible with the

pause:sender

Pauses the current playback or recording session by invoking Sound's pause: method. If no sound
returns nil otherwise, returns self.

play:sender

Play the current selection by invoking Sound's play: method. If there is no selection, the SoundVi
played. The willPlay: message is sent to the delegate before the selection is played didPlay: is sen
done playing. Returns self.

read:(void *)stream

Unarchives the SoundView by reading it from stream. Returns self.

readSelectionFromPasteboard:thePasteboard

Replaces the SoundView's current selection with the sound data on the given pasteboard. The pas
converted to the format of the data in the SoundView (if possible). If the SoundView has no select
data is inserted at the cursor position. Sets the current error code for the SoundView's Sound obje
retrieve by sending processingError to the Sound) and returns self.

record:sender

Replaces the SoundView's current selection with newly recorded material. If there is no selection,
at the cursor. The willRecord: message is sent to the delegate before the recording is started didRe
recording has completed. Recorded data is always taken from the CODEC microphone input. Ret

(float)reductionFactor

Returns the SoundView's reduction factor, computed as

reductionFactor = sampleCount / displayUnits

resignFirstResponder

Resigns the position of first responder. Returns self.

automatically when the SoundView's data changes and the SoundView is in autoscale mode. If th
autoscale mode, sizeToFit is invoked when the data changes. You never invoke this method direct
reimplement this method to provide specialized behavior. Returns self.

selectAll:sender

Creates a selection over the SoundView's entire Sound. Returns self.

setAutoscale:(BOOL)aFlag

Sets the SoundView's automatic scaling mode, used to determine how the SoundView is redisplay
changes. With autoscaling enabled (aFlag is YES), the SoundView's reduction factor is recomput
within the view frame. If it's disabled (aFlag is NO), the frame is resized and the reduction factor
SoundView is in a ScrollingView, autoScaling should be disabled (autoscaling is disabled by defau

setBackgroundGray:(float)aGray

Sets the SoundView's background gray value to aGray the default is NX_WHITE. Returns self.

setBezeled:(BOOL)aFlag

If aFlag is YES, the display is given a bezeled border. By default, the border of a SoundView disp
autodisplaying is enabled, the Sound is automatically redisplayed. Returns self.

setContinuous:(BOOL)aFlag

Sets the state of continuous action messages. If aFlag is YES, selectionChanged: messages are ser
mouse is being dragged. If NO, the message is sent only on mouse up. The default is NO. Return

setDelegate:anObject

Sets the SoundView's delegate to anObject. The delegate is sent messages when the user changes
Returns self.

setDisplayMode:(int)aMode

Sets the SoundView's display mode, either NX_SOUNDVIEW_WAVE or NX_SOUNDVIEW_M
If autodisplaying is enabled, the Sound is automatically redisplayed. Returns self.

default, a SoundView is enabled. Returns self.

setForegroundGray:(float)aGray

Sets the SoundView's foreground gray value to aGray. The default is NX_BLACK. Returns self.

setOptimizedForSpeed:(BOOL)flag

Sets the SoundView to optimize its display mechanism. Optimization greatly increases the speed v drawn, particularly for large sounds. It does so at the loss of some precision in representing the sou inaccuracies are corrected as you zoom in on the data. All SoundView's are optimized by default.

setReductionFactor:(float)reductionFactor

Recomputes the size of the SoundView's frame, if autoscaling is disabled. The frame's size (in dis according to the formula

displayUnits = sampleCount / reductionFactor

Increasing the reduction factor zooms out, decreasing zooms in on the data. If autodisplaying is en automatically redisplayed.

If the SoundView is in autoscaling mode, or reductionFactor is less than 1.0, the method avoids co and returns nil. (In autoscaling mode, the reduction factor is automatically recomputed when the s changesÐsee scaleToFit:.) Otherwise, the method returns self. If reductionFactor is the same as t factor, the method returns immediately without recomputing the frame size.

setSelection:(int)firstSample size:(int)sampleCount

Sets the selection to be sampleCount samples wide, starting with sample firstSample (samples are Returns self.

setSound:aSound

Sets the SoundView's Sound object to aSound. If autoscaling is enabled, the drawing coordinate s aSound's data fits within the current frame. Otherwise, the frame is resized to accommodate the le autodisplaying is enabled, the SoundView is automatically redisplayed. Returns self.

showCursor

Displays the SoundView's cursor. This is usually handled automatically. Returns self.

Resizes the SoundView's frame (horizontally) to maintain a constant reduction factor. This metho automatically when the SoundView's data changes and the SoundView isn't in autoscale mode. If autoscale mode, scaleToFit is invoked when the data changes. You never invoke this method dire reimplement this method to provide specialized behavior. Returns self.

sound

Returns a pointer to the SoundView's Sound object.

soundBeingProcessed

Returns the Sound object that's currently being played or recorded into. Note that the actual Soun performed isn't necessarily the SoundView's sound (the object returned by the sound method) for creates a private performance Sound object. While this is generally an implementation detail, this case the SoundView's delegate needs to know exactly which object will be (or was) performed.

stop:sender

Stops the SoundView's current recording or playback. Returns self.

tellDelegate:(SEL)theMessage

Sends theMessage to the SoundView's delegate with the SoundView as the argument. If the deleg the message, then it isn't sent. You normally never invoke this method it's invoked automatically playing or editing, is performed. However, you can invoke it in the design of a SoundView subcla

validRequestorForSendType:(NXAtom)sendType
andReturnType:(NXAtom)returnType

You never invoke this method it's implemented to support services that act on sound data.

willPlay:sender

Used to redirect delegate messages from the SoundView's Sound object you never invoke this met

willRecord:sender

Used to redirect delegate messages from the SoundView's Sound object you never invoke this met ignored. Returns self.

didPlay:sender

Sent to the delegate just after the SoundView's sound is played.

didRecord:sender

Sent to the delegate just after the SoundView's sound is recorded into.

hadError:sender

Sent to the delegate if an error is encountered during recording or playback of the SoundView's so

selectionChanged:sender

Sent to the delegate when the SoundView's selection changes.

soundDidChange:sender

Sent to the delegate when the SoundView's sound data is edited.

willFree:sender

Sent to the delegate when the SoundView is freed.

willPlay:sender

Sent to the delegate just before the SoundView's sound is played.

willRecord:sender

Sent to the delegate just before the SoundView's sound is recorded into.