ExtSoundHeader structure

#include <Sound.h>

typedef struct **ExtSoundHeader** {

		Size	Offset	Description
Ptr	samplePtr;	4	0	**if** NIL then samples are in sample area
unsigned long	numChannels;	4	4	number of channels, ie mono = 1
Fixed	sampleRate;	4	8	sample rate in Fixed point representation
unsigned long	loopStart;	4	12	start of looping portion
unsigned long	loopEnd;	4	16	end of looping portion
unsigned char	encode;	1	20	data structure used , stdSH, extSH, or cmpSH
unsigned char	baseFrequency;	1	21	baseFrequency value
unsigned long	numFrames;	4	22	length in total number of frames
extended	AIFFSampleRate;	10	26	IEEE sample rate
Ptr	markerChunk;	4	36	sync track
Ptr	instrumentChunks;	4	40	AIFF instrument chunks
Ptr	AESRecording;	4	44	
unsigned short	sampleSize;	2	48	number of bits in sample
unsigned short	futureUse1;	2	50	reserved by Apple
unsigned long	futureUse2;	4	52	reserved by Apple
unsigned long	futureUse3;	4	56	reserved by Apple
unsigned long	futureUse4;	4	60	reserved by Apple
char	sampleArea[1];	1	64	space for when samples follow directly

} **ExtSoundHeader**; 66

typedef ExtSoundHeader ***ExtSoundHeaderPtr**;

Field descriptions

samplePtr — A pointer to the sampled sound data. If the sampled sound is located in memory immediately after the futureUse4 field, then this field should be set to NIL. Otherwise, this field is a pointer to the memory location of the sampled sound data.

numChannels — The number of channels in the sampled sound data.

sampleRate — The rate at which the sample was originally recorded. The approximate sample rates are shown in the Table "Sample Rates". Note that the sample rate is declared as a Fixed data type, but the most significant bit is not treated as a sign bit; instead, that bit is interpreted as having the value 32,768.

sampleRate

Rate (kHz)	**Rate (Hz)**	**value (Fixed)**
5 kHz	H5563.6363	0x15BBA2E8
7 kHz	H7418.1818	0x1CFA2E8B

11 kHz	11127.2727	0x2B7745D1
22 kHz	22254.5454	0x56EE8BA3
44 kHz	44100.0000	0xAC440000

loopStart — The starting point of the portion of the extended sampled sound header that is to be used by the **Sound Manager** when determining the duration of freqDurationCmd. These loop points specify the byte numbers in the sampled data to be used as the beginning and end points to cycle through when playing the sound.

loopEnd — The end point of the portion of the extended sampled sound header that is to be used by the **Sound Manager** when determining the duration of freqDurationCmd.

encode — The method of encoding used to generate the sampled sound data. For an extended sound header, you should specify the constant extSH. Encode option values in the ranges 0 through 63 and 128 to 255 are reserved for use by Apple. You are free to use numbers in the range 64 through 127 for your own encode options.

baseFrequency — The pitch at which the original sample was taken. This value must be in the range 1 through 127. The Table "Midi Values" in **Playing Frequencies** lists the possible baseFrequency values.

The baseFrequency value allows the **Sound Manager** to calculate the proper playback rate of the sample when an application uses the freqDurationCmd command. Applications should not alter the baseFrequency field of a sampled sound; to play the sample at different pitches, use freqDurationCmd or freqCmd.

numFrames — The number of frames in the sampled sound data.

AIFFSampleRate — The sample rate at which the frames were sampled before compression, as expressed in an extended data type representation.

markerChunk — Synchronization information. The markerChunk field is not presently used and should be set to NULL.

instrumentChunks — Instrument information.

AESRecording — Audio information.

sampleSize — The number of bits in each sample frame.

futureUse1 — Reserved.

futureUse2	Reserved.
futureUse3	Reserved.
futureUse4	The four futureUse fields are reserved for use by Apple. To maintain compatibility with future releases of system software, you should always set these fields to 0.
sampleArea	An array of bytes, each of which contains a value similar to the values in a wave-table description. These values are interpreted as offset values, where 0x80 represents an amplitude of 0. The value 0x00 is the largest negative amplitude and 0xFF is the largest positive amplitude.