CmpSoundHeader structure

#include <Sound.h>

typedef struct **CmpSoundHeader** {

Type	Field	Size	Offset	Description
Ptr	samplePtr;	4	0	if NIL then samples are in sampleArea
unsigned long	numChannels;	4	4	number of channels in sample
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	futureUse1;	4	40	reserved by Apple
Ptr	futureUse2;	4	44	reserved by Apple
StateBlockPtr	stateVars;	4	48	pointer to StateBlock
LeftOverBlockPtr	leftOverSamples;	4	52	used to save truncated samples between compression calls
unsigned short	compressionID;	2	56	0 means no compression, non zero means compressionID
unsigned short	packetSize;	2	58	number of bits in compressed sample packet
unsigned short	snthID;	2	60	resource ID of **Sound Manager** 'snth' that contains NRT C/E
unsigned short	sampleSize;	2	62	number of bits in non-compressed sample
char	sampleArea[1];	1	64	space for when samples follow directly

} **CmpSoundHeader**; 66

typedef CmpSoundHeader ***CmpSoundHeaderPtr**;

Field descriptions

samplePtr — Indicates the location of the compressed sound frames. If samplePtr is NIL, then the frames are located in the sampleArea field of the compressed sound header. Otherwise, samplePtr points to a buffer that contains the frames.

numChannels — Indicates how many channels are in the sample.

sampleRate — Indicates the sample rate at which the frames were sampled before compression. The approximate sample rates are shown in the Table "Sample Rates". under the **SoundHeader** entry. 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.

loopStart	Indicates the beginning of the loop points of the sound before compression.
loopEnd	Indicates the end of the loop points of the sound before compression.
encode	Indicates the method of encoding (if any) used to generate the sampled sound data. For a compressed sound header, you should specify the constant cmpSH 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	Indicates the pitch of the original sampled sound. It is not used by bufferCmd. If you wish to make use of baseFrequency with a compressed sound, you must first expand it and then play it with soundCmd and freqDurationCmd.
numFrames	Indicates the number of frames contained in the compressed sound header. When you store multiple channels of uncompressed sound, store them as interleaved sample frames (as in AIFF). When you store multiple channels of compressed sounds, store them as interleaved packet frames.
AIFFSampleRate	Indicates the sample rate at which the frames were sampled before compression, as expressed in an extended data type representation.
markerChunk	Specifies synchronization information. The markerChunk field is not presently used and should be set to NIL.
futureUse1	Reserved.
futureUse2	The two 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.
stateVars	Points to a state block record. The stateVars field is used to store the state variables for a given algorithm across consecutive calls.
leftOverSamples	Points to a left over block record. You can use this block to store samples that will be truncated across algorithm invocations.
compressionID	Identifies the compression algorithm used on the samples in the compressed sound header. You can use a constant to define the compression algorithm.

notCompressed	noncompressed samples
threeToOne	3:1 compressed samples
sixToOne	6:1 compressed samples

Apple reserves the right to use compression IDs in the range 0 through 511.

packetSize — Indicates the size, specified in bits, of the smallest element that a given expansion algorithm can work with. You can use a constant to define the packet size.

sixToOnePacketSize	size for 6:1
threeToOnePacketSize	size for 3:1

snthID — Indicates the resource ID number of the 'snth' resource that was used to compress the packets contained in the compressed sound header. A 3:1 'snd ' resource would have a *snthID* of 11, and a 6:1 'snd ' would have a *snthID* of 13. If a compressed sound header contains samples that are not compressed, you should set the *snthID* field to 0.

sampleSize — Indicates the size of the sample before it was compressed. Currently, the **Sound Manager** works only with 8-bit samples. The samples should be in offset binary format; applications that read their data from AIFF files must convert the samples from two's complement format to the binary format. The samples passed in the compressed sound header should always be byte-aligned, and any padding done to achieve byte alignment should be done from the left with zeros.

sampleArea — Contains the sample frames, but only when the samplePtr field is NULL. Otherwise, the sample frames are in the location indicated by samplePtr.

The code example in **Playing Sampled Sounds** illustrates the structure of an 'snd ' resource that contains compressed sound data.

This resource has the same general structure as the 'snd ' resource illustrated in **The Format 1 'snd ' Resource** in the section entitled **Sound Resources**. The principal difference is that the standard sound header is replaced by the compressed sound header. This example resource specifies a monophonic sound compressed by using the 3:1 compression algorithm.