

Sound Functions

Accessing Sound Devices and Hardware

Access sound devices

```
int          SNDAcquire(int soundResource, int priority, int preempt, int timeout,
                        SNDNegotiationFun negFun, void *arg, port_t *devicePort, port_t
                        *ownerPort)

int          SNDReset(int soundResource, port_t devicePort, port_t ownerPort)
int          SNDRelease(int soundResource, port_t *devicePort, port_t *ownerPort)
```

Reserve sound devices for recording or playback

```
int          SNDReserve(int soundResource, int priority)
int          SNDUnreserve(int soundResource)
```

Set the host computer for subsequent playback or recording

```
int          SNDSetHost(char *newHostname)
```

Sound playback utilities

```
int          SNDSetVolume(int left, int right)
int          SNDGetVolume(int *left, int *right)
int          SNDSetMute(int speakerOn)
int          SNDGetMute(int *speakerOn)
int          SNDSetFilter(int filterOn)
int          SNDGetFilter(int *filterOn)
```

Recording and Playing

Play a soundfile

```
int SNDPlaySoundfile(char *path, int priority)
```

Recording and playing a sound

```
int          SNDStartPlaying(SNDSoundStruct *sound, int tag, int priority, int preempt,
                             SNDNotificationFun beginFun, SNDNotificationFun endFun)
int          SNDVerifyPlayable(SNDSoundStruct *sound)
int          SNDStartRecording(SNDSoundStruct *sound, int tag, int priority, int preempt,
                             SNDNotificationFun beginFun, SNDNotificationFun endFun)
int          SNDStartRecordingFile(char *fileName, SNDSoundStruct *sound, int tag, int
                             priority, int preempt, SNDNotificationFun beginFun,
                             SNDNotificationFun endFun)
int          SNDStop(int tag)
```

int	SNDWait (int <i>tag</i>)
int	SNDSamplesProcessed (int <i>tag</i>)
int	SNDModifyPriority (int <i>tag</i> , int <i>newPriority</i>)

Reading and Writing Soundfiles

Read a sound from a file

int	SNDReadSoundfile (char * <i>path</i> , SNDSoundStruct ** <i>sound</i>)
int	SNDRead (int <i>fd</i> , SNDSoundStruct ** <i>sound</i>)
int	SNDReadHeader (int <i>fd</i> , SNDSoundStruct ** <i>sound</i>)
int	SNDReadDSPfile (char * <i>path</i> , SNDSoundStruct ** <i>sound</i> , char * <i>info</i>)

Write a sound to a file

int	SNDWriteSoundfile (char * <i>path</i> , SNDSoundStruct * <i>sound</i>)
int	SNDWrite (int <i>fd</i> , SNDSoundStruct * <i>sound</i>)
int	SNDWriteHeader (int <i>fd</i> , SNDSoundStruct * <i>sound</i>)

Accessing Sound Data

Create and free a sound structure

int	SNDAlloc (SNDSoundStruct ** <i>sound</i> , int <i>dataSize</i> , int <i>dataFormat</i> , int <i>samplingRate</i> , int <i>channelCount</i> , int <i>infoSize</i>)
int	SNDFree (SNDSoundStruct * <i>sound</i>)

Gain access to sampled sound data

int	SNDGetDataPointer (SNDSoundStruct * <i>sound</i> , char ** <i>ptr</i> , int * <i>size</i> , int * <i>width</i>)
-----	---

Measure samples in a sound

int	SNDSampleCount (SNDSoundStruct * <i>sound</i>)
int	SNDBytesToSamples (int <i>byteCount</i> , int <i>channelCount</i> , int <i>dataFormat</i>)
int	SNDSamplesToBytes (int <i>sampleCount</i> , int <i>channelCount</i> , int <i>dataFormat</i>)

Accessing the DSP

Boot the DSP

int	SNDBootDSP (port_t * <i>devicePort</i> , port_t * <i>ownerPort</i> , SNDSoundStruct * <i>dspCore</i>)
-----	---

Run the DSP

int	SNDRunDSP (SNDSoundStruct * <i>dspCore</i> , char * <i>toDSP</i> , int <i>toCount</i> , int <i>toWidth</i> , int <i>toBufferSize</i> , char ** <i>fromDSP</i> , int * <i>fromCount</i> , int <i>fromWidth</i> , int <i>negotiationTimeout</i> , int <i>flushTimeout</i> , int <i>conversionTimeout</i>)
-----	---

Compressing Sound Data

Compress or decompress a sound

int

SNDCompressSound(SNDSoundStruct **fromSound*, SNDSoundStruct ***toSound*,
BOOL *bitFaithful*, int *compressionAmount*)

Query for frequency bands used by Audio Transform Compression

int

SNDGetNumberOfATCBands(int **numBands*)

int

SNDGetATCBandFrequencies(int *numBands*, float **centerFreqs*)

int

SNDGetATCBandwidths(int *numBands*, float **bandwidths*)

Speed up or slow down playback of ATC sound

int

SNDDropATCSamples(int *numSamples*, int *bySamples*)

int

SNDInsertATCSamples(int *numSamples*, int *bySamples*)

Modify volume or equalization for ATC playback

int

SNDSetATCGain(float *level*)

int

SNDGetATCGain(float **level*)

int

SNDSetATCEqualizerGains(int *numBands*, float **gains*)

int

SNDGetATCEqualizerGains(int *numBands*, float **gains*)

int

SNDScaleATCEqualizerGains(int *numBands*, float **gainScalars*)

Set or get ATC parameters

int

SNDSetATCSquelchThresholds(int *numBands*, float **thresholds*)

int

SNDGetATCSquelchThresholds(int *numBands*, float **thresholds*)

int

SNDUseDefaultATCSquelchThresholds(void)

Set and get compression attributes used in recording

int

SNDSetCompressionOptions(SNDSoundStruct **sound*, int *bitFaithful*, int
compressionAmount)

int

SNDGetCompressionOptions(SNDSoundStruct **sound*, int **bitFaithful*, int
**compressionAmount*)

Converting Sound Data

Convert between logarithmic and linear units

float

SNDConvertDecibelsToLinear(float *dB*)

float

SNDConvertLinearToDecibels(float *linear*)

Convert a sound's attributes

int

SNDConvertSound(SNDSoundStruct **fromSound*, SNDSoundStruct ***toSound*)

unsigned char

SNDMulaw(short *linearValue*)

short

SNDiMulaw(unsigned char *mulawValue*)

Editing Sound Data

Copy all or part of a sound

int	SNDCopySound (SNDSoundStruct ** <i>toSound</i> , SNDSoundStruct * <i>fromSound</i>)
int	SNDCopySamples (SNDSoundStruct ** <i>toSound</i> , SNDSoundStruct * <i>fromSound</i> , int <i>startSample</i> , int <i>sampleCount</i>)

Edit a sampled sound

int	SNDInsertSamples (SNDSoundStruct * <i>toSound</i> , SNDSoundStruct * <i>fromSound</i> , int <i>startSample</i>)
int	SNDDeleteSamples (SNDSoundStruct * <i>sound</i> , int <i>startSample</i> , int <i>sampleCount</i>)
int	SNDCompactSamples (SNDSoundStruct ** <i>toSound</i> , SNDSoundStruct * <i>fromSound</i>)

Sound Errors

Describe a sound error

char	*SNDSoundError (int <i>err</i>)
------	---