Copyright ©1995 by NeXT Computer, Inc. All Rights Reserved.

Defined Types

NXSoundDeviceError

DECLARED IN soundkit/NXSoundDevice.h

SYNOPSIS

typedef enum _NXSoundDeviceError {
    NX_SoundDeviceErrorNone,
    NX_SoundDeviceErrorKernel,
    NX_SoundDeviceErrorTimeout,
    NX_SoundDeviceErrorLookUp,
    NX_SoundDeviceErrorHost,
    NX_SoundDeviceErrorNoDevice,
    NX_SoundDeviceErrorNotActive,
    NX_SoundDeviceErrorTag,
    NX_SoundDeviceErrorParameter,
    NX_SoundDeviceErrorMax
} **NXSoundDeviceError**

DESCRIPTION Error codes returned by Sound Kit methods that access the sound driver.

NXSoundParameterTag (Sound Device Keys)

DECLARED IN soundkit/NXSoundParameterTags.h

SYNOPSIS

typedef enum _NXSoundParameterTag {
    NX_SoundDeviceBufferSize,
    NX_SoundDeviceBufferCount,
    NX_SoundDeviceDetectPeaks,
    NX_SoundDeviceRampUp,
    NX_SoundDeviceRampDown,
    NX_SoundDeviceInsertZeros,
    NX_SoundDeviceDeemphasize,
    NX_SoundDeviceMuteSpeaker,
    NX_SoundDeviceMuteHeadphone,
    NX_SoundDeviceMuteLineOut,
    NX_SoundDeviceOutputLoudness,
    NX_SoundDeviceOutputAttenuationStereo,
    NX_SoundDeviceOutputAttenuationLeft,
    NX_SoundDeviceOutputAttenuationRight,
    NX_SoundDeviceAnalogInputSource,
    NX_SoundDeviceMonitorAttenuation,
    NX_SoundDeviceInputGainStereo,
    NX_SoundDeviceInputGainLeft,
    NX_SoundDeviceInputGainRight,

... (continued under the next heading)
} **NXSoundParameterTag**

DESCRIPTION Sound parameter key tags that apply to NXSoundDevice objects.

NXSoundParameterTag (Sound Device Values)

DECLARED IN soundkit/NXSoundParameterTags.h

SYNOPSIS typedef enum _NXSoundParameterTag {
... (continued from the previous heading)
NX_SoundDeviceAnalogInputSource_Microphone,
NX_SoundDeviceAnalogInputSource_LineIn,
... (continued under the next heading)
} **NXSoundParameterTag**

DESCRIPTION Sound parameter value tags that apply to NXSoundDevice objects. These are acceptable values for the NX_SoundDeviceAnalogInputSource parameter.

NXSoundParameterTag (Sound Stream Keys)

DECLARED IN soundkit/NXSoundParameterTags.h

SYNOPSIS typedef enum _NXSoundParameterTag {
... (continued from the previous heading)
NX_SoundStreamDataEncoding,
NX_SoundStreamSamplingRate,
NX_SoundStreamChannelCount,
NX_SoundStreamHighWaterMark,
NX_SoundStreamLowWaterMark,
NX_SoundStreamSource,
NX_SoundStreamSink,
NX_SoundStreamDetectPeaks,
NX_SoundStreamGainStereo,
NX_SoundStreamGainLeft,
NX_SoundStreamGainRight,
... (continued under the next heading)
} **NXSoundParameterTag**

DESCRIPTION Sound parameter key tags that apply to NXSoundStream objects.

NXSoundParameterTag (Sound Stream Values)

DECLARED IN soundkit/NXSoundParameterTags.h

SYNOPSIS typedef enum _NXSoundParameterTag {
... (continued from the previous heading)
NX_SoundStreamDataEncoding_Linear16,
NX_SoundStreamDataEncoding_Linear8,

NX_SoundStreamDataEncoding_Mulaw8,
NX_SoundStreamDataEncoding_Alaw8,
NX_SoundStreamDataEncoding_AES,
NX_SoundStreamSource_Analog,
NX_SoundStreamSource_AES,
NX_SoundStreamSink_Analog,
NX_SoundStreamSink_AES
} **NXSoundParameterTag**

DESCRIPTION Sound parameter value tags that apply to NXSoundStream objects. The first five are values for the NX_SoundStreamDataEncoding parameter; the next two are for the NX_SoundStreamSource parameter; the last two are for the NX_SoundStreamSink parameter.

NXSoundStatus

DECLARED IN soundkit/Sound.h

SYNOPSIS

typedef enum {
NX_SoundStopped,
NX_SoundRecording,
NX_SoundPlaying,
NX_SoundInitialized,
NX_SoundRecordingPaused,
NX_SoundPlayingPaused,
NX_SoundRecordingPending,
NX_SoundPlayingPending,
NX_SoundFreed,
} **NXSoundStatus**;

DESCRIPTION These represent the activities of a Sound object, as returned by Sound's **status** method.

SNDCompressionSubheader

DECLARED IN sound/soundstruct.h

SYNOPSIS

typedef struct {
int **originalSize**
int **method**;
int **numDropped**;
int **encodeLength**;
} **SNDCompressionSubheader**;

DESCRIPTION This structure describes the attributes of a compressed sound. It immediately follows the general SNDSoundStruct header. If the sound data isn't compressed, this subheader is absent. The structure's fields are

originalSize	The size of the uncompressed data, in bytes
method	The compression format (see ªCompression Formats,º below)
numDropped	The number of dropped bits, if applicable
encodeLength	The number of samples represented by an encoded block

SNDError

DECLARED IN sound/sounderror.h

SYNOPSIS

typedef enum {
SND_ERR_NONE,
SND_ERR_NOT_SOUND,
SND_ERR_BAD_FORMAT,
SND_ERR_BAD_RATE,
SND_ERR_BAD_CHANNEL,
SND_ERR_BAD_SIZE,
SND_ERR_BAD_FILENAME,
SND_ERR_CANNOT_OPEN,
SND_ERR_CANNOT_WRITE,
SND_ERR_CANNOT_READ,
SND_ERR_CANNOT_ALLOC,
SND_ERR_CANNOT_FREE,
SND_ERR_CANNOT_COPY,
SND_ERR_CANNOT_RESERVE,
SND_ERR_NOT_RESERVED,
SND_ERR_CANNOT_RECORD,
SND_ERR_ALREADY_RECORDING,
SND_ERR_NOT_RECORDING,
SND_ERR_CANNOT_PLAY,
SND_ERR_ALREADY_PLAYING,
SND_ERR_NOT_IMPLEMENTED,
SND_ERR_NOT_PLAYING,
SND_ERR_CANNOT_FIND,
SND_ERR_CANNOT_EDIT,
SND_ERR_BAD_SPACE,
SND_ERR_KERNEL,
SND_ERR_BAD_CONFIGURATION,
SND_ERR_CANNOT_CONFIGURE,
SND_ERR_UNDERRUN,
SND_ERR_ABORTED,
SND_ERR_BAD_TAG,
SND_ERR_CANNOT_ACCESS,
SND_ERR_TIMEOUT,
SND_ERR_BUSY,
SND_ERR_CANNOT_ABORT,
SND_ERR_INFO_TOO_BIG,
SND_ERR_UNKNOWN,
} **SNDError**;

DESCRIPTION These are the sound error codes returned by many sound functions. The **SNDSoundError()** function returns a pointer to a string that describes the error given one of these codes as an argument.

SNDNotificationFun

DECLARED IN sound/performsound.h

SYNOPSIS

typedef int (***SNDNotificationFun**)
(SNDSoundStruct **s*,
int *tag*,
int *err*);

DESCRIPTION This is the notification function required as an argument to methods such as **SNDStartPlaying()** and **SNDStartRecording()**.

SNDSoundStruct

DECLARED IN sound/performsound.h

SYNOPSIS

typedef struct {
    int **magic**;
    int **dataLocation**;
    int **dataSize**;
    int **dataFormat**;
    int **samplingRate**;
    int **channelCount**;
    char **info**[4];
} **SNDSoundStruct**;

DESCRIPTION This structure defines the header for sound data. It's thoroughly explained in the description of the **SNDAlloc()** function.

snddriver_handlers

DECLARED IN sound/sounddriver.h

SYNOPSIS

typedef struct snddriver_handlers {
    void ***arg**;
    int **timeout**;
    sndreply_tagged_t **started**;
    sndreply_tagged_t **completed**;
    sndreply_tagged_t **aborted**;
    sndreply_tagged_t **paused**;
    sndreply_tagged_t **resumed**;
    sndreply_tagged_t **overflow**;
    sndreply_recorded_data_t **recorded_data**;
    sndreply_dsp_cond_true_t **condition_true**;
    sndreply_dsp_msg_t **dsp_message**;
    sndreply_dsp_msg_t **dsp_error**;
} **snddriver_handlers_t**;

DESCRIPTION This structure is required as an argument by the **snddriver_reply_handler()** function. It declares, primarily, a series of call-back functions that are used by the sound driver to communicate with your program.

sndreply_dsp_cond_true_t

DECLARED IN sound/sounddriver.h

SYNOPSIS

typedef void
(***sndreply_dsp_cond_true_t**)
    (void **arg*,
    unsigned int *mask*,

unsigned int *flags*,
unsigned int *regs*);

DESCRIPTION Function type used by the sound driver's reply handler.

sndreply_dsp_msg_t

DECLARED IN sound/sounddriver.h

SYNOPSIS typedef void (***sndreply_dsp_msg_t**)
(void **arg*,
int *data,*
int *size*);

DESCRIPTION Function type used by the sound driver's reply handler.

sndreply_recorded_data_t

DECLARED IN sound/sounddriver.h

SYNOPSIS typedef void (***sndreply_recorded_data_t**)
(void **arg*,
int *tag,*
void **data,*
int *size*);

DESCRIPTION Function type used by the sound driver's reply handler.

sndreply_tagged_t

DECLARED IN sound/sounddriver.h

SYNOPSIS typedef void (***sndreply_tagged_t**)
(void **arg*,
int *tag*);

DESCRIPTION Function type used by the sound driver's reply handler.

Symbolic Constants

ATC Frame Size

DECLARED IN sound/soundstruct.h

SYNOPSIS ATC_FRAME_SIZE

DESCRIPTION This constant represents the size of a single ATC (Audio Transform Compression) frame.

Compression Formats

DECLARED IN sound/soundstruct.h

SYNOPSIS SND_CFORMAT_BITS_DROPPED
SND_CFORMAT_BIT_FAITHFUL
SND_CFORMAT_ATC

DESCRIPTION These constants represent the three types of sound data compression.

DSP Host Commands

DECLARED IN sound/sounddriver.h

SYNOPSIS SNDDRIVER_DSP_HC_HOST_RD
SNDDRIVER_DSP_HC_HOST_WD
SNDDRIVER_DSP_HC_SYS_CALL

DESCRIPTION These constants represent the DSP host commands that can be passed as an argument to **snddriver_dsp_host_cmd()**.

DSP Protocol Options

DECLARED IN sound/sounddriver.h

SYNOPSIS SNDDRIVER_DSP_PROTO_DSPERR
SNDDRIVER_DSP_PROTO_C_DMA
SNDDRIVER_DSP_PROTO_S_DMA
SNDDRIVER_DSP_PROTO_HFABORT
SNDDRIVER_DSP_PROTO_DSPMSG
SNDDRIVER_DSP_PROTO_RAW

DESCRIPTION These constants represent the DSP protocols that can be passed as an argument to **snddriver_dsp_protocol()**.

Executable File Segment Name

DECLARED IN soundkit/Sound.h

SYNOPSIS NX_SOUND_SEGMENT_NAME

DESCRIPTION This represents the segment of an executable file in which sounds are stored.

Null Notification Function

DECLARED IN sound/performsound.h

SYNOPSIS SND_NULL_FUN

DESCRIPTION Used to pass a null **SNDNotificationFun()** function as an argument to functions such as **SNDStartPlaying()** and **SNDStartRecording()**.

Sampling Rates

DECLARED IN sound/soundstruct.h

SYNOPSIS

Code	Rate
SND_RATE_CODEC	8012.8210513 Hz
SND_RATE_LOW	22050.0 Hz
SND_RATE_HIGH	44100.0 Hz

DESCRIPTION These constants represent the three sampling rates that are directly supported by the sound software and hardware.

Sound Device Access Codes

DECLARED IN sound/accesssound.h

SYNOPSIS

Code	Device
SND_ACCESS_IN	Sound-in
SND_ACCESS_DSP	DSP
SND_ACCESS_OUT	Sound-out

DESCRIPTION Used by the sound device access methods, such as **SNDAcquire()** and **SNDReserve()**, to represent specific devices.

Sound Device Timeout Limit

DECLARED IN soundkit/NXSoundDevice.h

SYNOPSIS NX_SOUNDDEVICE_TIMEOUT_MAX

DESCRIPTION The default timeout limit for communication with the sound driver. The value is, essentially, infinity. You can reset the timeout limit through NXSoundDriver's **setTimeout:** method.

Sound Device Error Code Limits

DECLARED IN soundkit/NXSoundDevice.h

SYNOPSIS NX_SOUNDDEVICE_ERROR_MIN
NX_SOUNDDEVICE_ERROR_MAX

DESCRIPTION The minimum and maximum NXSoundDeviceError values.

Sound Parameter Tag Bases

DECLARED IN soundkit/NXSoundParameterTags.h

SYNOPSIS NX_SoundDeviceParameterKeyBase
NX_SoundDeviceParameterValueBase
NX_SoundStreamParameterKeyBase
NX_SoundStreamParameterValueBase
NX_SoundParameterTagMax

DESCRIPTION Lowest tag values for the four sets of parameter tags. The parameter tag values start at 0; NX_SoundParmaeterTagMax is the highest parameter tag value reserved by the Sound Kit.

Sound Stream Control Codes

DECLARED IN sound/sounddriver.h

SYNOPSIS SNDDRIVER_AWAIT_STREAM
SNDDRIVER_ABORT_STREAM
SNDDRIVER_PAUSE_STREAM
SNDDRIVER_RESUME_STREAM

DESCRIPTION These constants represent the controlling operations that are specified as an argument to **snddriver_stream_control()**.

Sound Stream Null Time

DECLARED IN soundkit/NXSoundStream.h

SYNOPSIS NX_SOUNDSTREAM_TIME_NULL

DESCRIPTION Provides a **timeval** value (as defined in **sys/time.h**) that indicates the present time.

Sound Stream Path Codes

DECLARED IN sound/sounddriver.h

SYNOPSIS SNDDRIVER_STREAM_FROM_SNDIN
SNDDRIVER_STREAM_TO_SNDOUT_22
SNDDRIVER_STREAM_TO_SNDOUT_44
SNDDRIVER_STREAM_FROM_DSP
SNDDRIVER_STREAM_TO_DSP
SNDDRIVER_STREAM_DSP_TO_SNDOUT_22
SNDDRIVER_STREAM_DSP_TO_SNDOUT_44
SNDDRIVER_STREAM_THROUGH_DSP_TO_SNDOUT_22

SNDDRIVER_STREAM_THROUGH_DSP_TO_SNDOUT_44
SNDDRIVER_DMA_STREAM_TO_DSP
SNDDRIVER_DMA_STREAM_FROM_DSP
SNDDRIVER_DMA_STREAM_THROUGH_DSP_TO_SNDOUT_22
SNDDRIVER_DMA_STREAM_THROUGH_DSP_TO_SNDOUT_44

DESCRIPTION These constants represent the sound stream paths that can be specified as an argument to the **snddriver_stream_setup()** function.

Sound Structure Formats

DECLARED IN sound/soundstruct.h

SYNOPSIS

SND_FORMAT_UNSPECIFIED
SND_FORMAT_MULAW_8
SND_FORMAT_LINEAR_8
SND_FORMAT_LINEAR_16
SND_FORMAT_LINEAR_24
SND_FORMAT_LINEAR_32
SND_FORMAT_FLOAT
SND_FORMAT_DOUBLE
SND_FORMAT_INDIRECT
SND_FORMAT_DSP_CORE
SND_FORMAT_DSP_DATA_8
SND_FORMAT_DSP_DATA_16
SND_FORMAT_DSP_DATA_24
SND_FORMAT_DSP_DATA_32
SND_FORMAT_DISPLAY
SND_FORMAT_MULAW_SQUELCH
SND_FORMAT_EMPHASIZED
SND_FORMAT_COMPRESSED
SND_FORMAT_COMPRESSED_EMPHASIZED
SND_FORMAT_DSP_COMMANDS

DESCRIPTION These constants represent the various sound formats in which sound data can be stored. Note that not all formats are playable without conversion.

Sound Structure Magic Number

DECLARED IN sound/soundstruct.h

SYNOPSIS

SND_MAGIC

DESCRIPTION This constant is used to identify a sound structure. It's the value of the **magic** field of all valid SNDSoundStruct structures.

SoundView Display Modes

DECLARED IN soundkit/SoundView.h

SYNOPSIS

NX_SOUNDVIEW_MINMAX
NX_SOUNDVIEW_WAVE

DESCRIPTION These constants represent the two display modes offered by the SoundView class. See the SoundView class specification for details.

Global Variables

NXSoundPboardType

DECLARED IN soundkit/Sound.h

SYNOPSIS extern NXAtom **NXSoundPboardType**;

DESCRIPTION This is the sound pasteboard type.