3.2 Release Notes: Sound Kit

This file contains release notes for Release 3 of the Sound Kit.   Items specific to each of the minor releasesÐRelease 3.2, Release 3.1, and Release 3.0Ðare listed in separate sections, starting with the latest release (3.2).

Notes Specific to Release 3.2

New Features

The following new features have been added to the Sound Kit since Release 3.1.

· Third party developers can now write custom sound drivers.   See the Driver Kit documentation and examples.

· Three new sound drivers are provided in addition to the ProAudioSpectrum Ð

CompaqBusinessAudio, IntelGXProAudio, and MSWSoundSystem.

Bugs Fixed in Release 3.2

These bugs have been fixed in Release 3.2:

Reference	34627
Problem	Soundfile compression doesn't work.
Description	Playing back a compressed soundfile produces noise.
Reference	36341
Problem	8-bit recording and playback not supported.
Description	The ProAudioSpectrum driver supports 16-bit data only.
Reference	none
Problem	Sound doesn't work on ISA machines.
Description	Sound is supported on EISA machines only.

Known Problems in Release 3.2

The following known problems exist in the Sound Kit only on NEXTSTEP for Intel Processors.

·   Playback of stereo (two channel) sounds at the high sample rate (44100.0 Hz) is unreliable.

·   Playback ramping doesn't work.

·   Underrun conditions aren't detected.

Notes Specific to Release 3.1

New Features

The following new feature has been added to the Sound Kit since Release 3.0.

·   A ªsound parameterº facility has been added to the Sound Kit.   With this facility, certain hardware and software sound parametersÐsuch as speaker volume and sound data sampling rateÐare identified by pre-defined constants ("parameter tags").   To set the value of a particular parameter, you use the general parameter-setting methods defined by the NXSoundParameters protocol.   These methods look a lot like hash table operations,   where the value that you specify is applied to the parameter that's identified by a particular parameter tag.   However, the data type of the value s specific; furthermore, because of the quantitative nature of

sound parameters, only numeric data-setting methods are provided.   The methods to set a parameter value are

- **setParameter:toInt:**
- **setParameter:toFloat:**
- **setParameter:toBool:**

Analogous methods for retrieving values are also provided.
The methods defined by NXSoundParameters has made some of the methods defined by NXSoundDevice and NXSoundStream (and their subclasses) obsolete. These are:

NXSoundDevice
- **setBufferSize:**
- **bufferSize**
- **setBufferCount:**
- **bufferCount**
- **setPeakHistory:**
- **peakHistory**
- **setDetectPeaks:**
- **isDetectingPeaks**

NXSoundIn
all methods

NXSoundOut
all methods

NXPlayStream
- playBuffer:size:tag:channelCount:samplingRate:...
- setDetectPeaks:
- isDetectingPeaks
- setPeakHistory:
- peakHistory

NXRecordStream
- recordSize:tag:lowWaterMark:highWaterMark:

Known Problems

The following known problems exist in the Sound Kit only on NEXTSTEP for Intel Processors.

· Sound currently doesn't work on ISA machines.   *Problem fixed in 3.2.   See "Bugs Fixed in Release 3.2" section above.*

· Playback and recording at the high sample rate (44100.0 Hz) are unreliable in some configurations.   Only a problem for 44100.0 Hz stereo in 3.2.   *See "Known Problems in Release 3.2" section above.*

- Soundfile compression doesn't work. *Problem fixed in 3.2. See "Bugs Fixed in Release 3.2" section above.*

- 8-bit data recording and playback doesn't work. *Problem fixed in 3.2. See "Bugs Fixed in Release 3.2" section above.*

- Playback ramping doesn't work.

- Underrun conditions aren't detected.

Incompatible Changes

The following incompatible changes have been made to the Sound Kit since Release 3.0.

- **SoundView** now expects floating-point samples (SND_FORMAT_FLOAT and SND_FORMAT_DOUBLE) to be in the range -1.0 to +1.0. In Release 3.0, floating-point samples were expected to be in the range -32768.0 to +32767.0.

Notes Specific to Release 3.0

These notes were included with the Release 3.0 version of the Sound Kit.

New Features

The following new features have been added to the Sound Kit since Release 2.1.

- SoundView has been significantly optimized (especially when viewing large sounds).

- Several new objects have been added to provide a low-level interface to sound recording and playback facilities, including the new mixing and scaling features of the sound driver.   These objects do not provide access to the DSP.   See the documentation and header files for NXSoundDevice, NXSoundStream, NXSoundIn, NXSoundOut, NXPlayStream, and NXRecordStream.

- Four new methods have been added to **SoundView** for Services support:

```
+ initialize;
- (BOOL)writeSelectionToPasteboard:pboard
              types:(NXAtom *)types;
- readSelectionFromPasteboard:pboard;
- validRequestorForSendType:(NXAtom)sendType
andReturnType:(NXAtom)returnType;
```

- The new method

```
- isPlayable:aSound;
```

has been added for determining if a sound object can be played using **play** method.   The method is implemented in **Sound,** and **SoundView** forwards it.

· A new method,

```
+ addName:(const char *)aName fromBundle:(NXBundle
*)aBundle;
```

has been added to the Sound object.

· The method `convertToFormat:samplingRate:channelCount:` has been generalized to many more cases, as provided by SNDConvertSound().

· A convenience method `convertToFormat:newFormat` has been added to **Sound** which converts the data format without modifying the sampling rate or channel count.

· The **SoundView** method `setSound:aSound` now returns `nil` if a sound format cannot be displayed.   When this is detected, `convertToFormat:` can be used to create a displayable format.   The most efficient display format is `SND_FORMAT_LINEAR_16.`

Incompatible Changes

The following incompatible changes have been made to the Sound Kit since Release 2.1.

· The following Sound methods

```
+ addName:(char *)name
          fromMachO:(char *)sectionName;
- initFromMachO:(char *)sectionName;
```

have been changed to

```
+ addName:(char *)name
          fromSection:(char *)sectionName;
- initFromSection:(char *)sectionName;
```

to conform with the corresponding methods of other NEXTSTEP classes.

· The following Sound methods

```
- initFromPasteboard;
- (int)writeToPasteboard;
```

have been changed to

```
- initFromPasteboard:(NXPasteboard *)thePboard;
- (int)writeToPasteboard:(NXPasteboard *)thePboard;
```

to conform with the corresponding methods of other NEXTSTEP classes.

· The following three methods in SoundView have been removed from the API. The ªreductionº concept was never utilized in 2.1 and its predecessors and the calcDrawInfo method is no longer necessary due to enhancements in SoundView's internal architecture.

```
- reduction;
- setReduction:aDisplayReduction;
- calcDrawInfo;
```

Other Changes

The following additional changes have been made to the Sound Kit since Release 2.1.

· SoundView

SoundView has been optimized for much greater performance.   The SoundView

can now load and display any size file in a few seconds.   For reference, 16M file used to take about a minute to load and display.   Two new methods have been added to determine whether full optimization will be used.   When optimizing for speed, a zoomed out display may miss samples -- zooming in always shows all detail.

```
- (BOOL)isOptimizedForSpeed;
- setOptimizedForSpeed:(BOOL)aFlag;
```

A new method has been added to SoundView to make it easier to subclass SoundView to draw the visual appearance of the samples in the sound without having to keep track of other parts of the SoundView (e.g. the bezel, the cursor, etc.):

```
- drawSamplesFrom:(int)startX
             to:(int)endX;
```

· Constants using the "SK" prefix have been changed to use the "NX" prefix.   The old contants have been retained for compatiblity, but new code should use the new prefixes.   See the Sound.h and SoundView.h header files.

Known Problems

The following are known problems in the current release of the Sound Kit.   They will

be fixed in a future release.

· Your application may crash if the sound driver sends a message to an **NXSoundStream** object that has been freed.   To avoid this, send the **-abort:** and **-deactivate** messages to the stream, then free the object only after your **soundStreamDidAbort:deviceReserved:** delegate method has been called.

· Setting the gain to a value greater than 1.0 on an NXPlayStream can cause a numerical "wrap-around" which actually reduces the amplitude.   Only soft sounds should be boosted with a gain greater than 1.0.