Q: How can I determine a Sound's duration?

A: Send the Sound the sampleCount and samplingRate methods (if necessary) and divide the former number by the latter.

Q: How can I get Sounds to repeat one after the other, or the same one multiple times?

A: There are several approaches; however, none are guaranteed to work in all possible situations.

1) You can compute the duration of each sound just before you play it (following the approach in the answer to the first question). Set up a timed entry (see documentation on DPSAddTimedEntry) that occurs at a time equal to the sound's duration plus any desired delay between playings. When you get this timed entry, you tell the next Sound to play, and so forth.

2) If you don't require an Application event loop, you can reliably chain sounds together using the Sound library C functions. See the example file /NextDeveloper/Examples/Sound/chaintest.c, and the README file in the same directory. You are advised to use the SNDNotificationFun

approach, as in this example, and to avoid the SNDWait() function.

3) If you require an Application with an event loop (the normal case), you might have success using the SoundKit delegation method didPlay:. Create an object that is the Sound's delegate, or use an existing object. Provide it with these methods:

```
int sndCtr;

- init {
[mySound setDelegate:self]; /* mySound is the Sound to be looped */
sndCtr = 1;
}

- playIt {
[mySound play];
}

- didPlay:sender {
```

```
        if (sndCtr++ < NUMTIMES)
            [self playIt]; /* play it NUMTIMES in a row.  Alternately, use some other
terminating condition, or trigger other Sounds here */
        }
```

This approach sometimes causes problems in the playback, however, particularly if the user interrupts the playing with an event such as a mouse-down. You'll have to try it for your specific case and see whether it works.

4) If none of these approaches is suitable, the workaround is to create a new Sound composed of multiple copies of the desired sound(s), using the SoundKit's copy/paste functionality. You can do this splicing programmatically with the insertSamples:at: method, or manually by using the SoundEditor example.

5) As a final option, you can combine the chaintest strategy with the SoundKit example above. Override Sound's "play" method and directly enqueue the soundstruct with SNDStartPlaying, giving it your own special SNDNotificationFun to do the chaining as in chaintest. The

drawback of this approach is that you cannot safely send the delegate a soundDidPlay: message when the repeat count runs out.

QA267

Valid for 1.0, 2.0, 3.0