RRA: readily readable audio

John C. Lusth

Revision Date: November 7, 2011

1 Introduction

The purpose of this document is to specify the audio format used by *songlib*. The RRA (Readily Readable Audio) format is designed to be easily viewed (and debugged) and extensible. Originally suggested by Ian Taylor for use in a High School Programming Contest, it has been adopted as the fundamental format by *songlib* to allow for less-experienced programmers to improve and extend the library.

An RRA file is composed of two sections, a header section and a data section.

2 Grammar

An RRA file has the following grammar. Non-terminals are in lowercase and terminals are in UPPERCASE. An RRA file begins with the seven ASCII characters RRAUDIO.

```
file : "RRAUDIO" optTags "%%" data

optTags : *empty*
        | tag optTags

tag : attribute ":" value

value : IDENTIFIER | INTEGER | STRING

data : INTEGER
     | INTEGER data
```

An RRA file is free-format and whitespace delimited. Comments begin with a bang (!) and continue until the end of line. Here is an example file:

```
RRAUDIO
! Hohner Pocket Pal Harmonica
! D#
! recorded by Becky Smith, February 2011
sampleRate: 44100
bitsPerSample: 24
channels: 1
replicates: "1203928,2593821"
%%
! amplitude data begins
0
0
```

```
1
-2
...
```

Note that blank lines, whitespace before a colon, and tags spanning multiple lines are all permissible. Comments may appear in the data section as well.

The token '%%' separates the header from the data section of an RRA file.

3 Required tags

No tags are required but all applications must support the following tags:

```
channels: <positive integer>
sampleRate: <positive integer>
bitsPerSample: <postive integer>
samples: <a non-negative integer>
skip: <a non-negative integer>
```

If not present, the following values are assumed for the the supported attributes:

```
channels: 1
sampleRate: 44100
bitsPerSample: 16
samples: 0
skip: 0
```

The value for *samples* indicates the number of samples in one channel. To find the total number of samples across all channels, one would multiply *samples* by *channels*.

A zero value for the *samples* attribute indicates that the number of samples is not specified and an application should continue to read data until the end-of-file. If at all possible, an application should set the *samples* attribute accurately. It is up to the application to decide how to handle a *samples* value that doesn't match the number actual amount of amplitude data. At a minimum, a warning message should be generated.

The *skip* attribute indicates how many samples at the beginning of the audio stream should be ignored by the application.

4 Duplicate tags

The meaning of duplicate tags is not specified. Three reasonable alternatives come to mind:

- all subsequent tags are ignored
- all previous tags are ignored
- the attribute is variadic and the values accumulate

For the first two cases, duplicate warning messages should be generated.

Instead of using duplicate tags for multiple values, one could also use a string to encapsulate multiple values. For example, the *replicates* tag might be specified thusly:

```
replicates : "87231,169842,27154"
```

...meaning that while the note starts at sample 0, the first replicate starts at sample 87231, the second at 169,842, and so on. It is up to the application to decide how multiple values are to be communicated.

5 Unknown tags

An application should handle unknown tags gracefully. At a mininum, an unknown tag should be ignored but a warning message should be generated.

6 Multiple channels

The number of channels specifies how many different audio streams are combined to make the whole. For example, monaural audio (channels: 1) has a single channel. Stereo audio (channels: 2) has two channels (left and right), while quadraphonic audio (channels: 4) has four channels, and so on. The amplitude data for multiple channels are interleaved. For stereo, the amplitude data/samples looks like this:

```
Amplitude_0_Channel_0
Amplitude_0_Channel_1
Amplitude_1_Channel_0
Amplitude_1_Channel_1
Amplitude_2_Channel_0
Amplitude_2_Channel_1
...
Amplitude_N-1_Channel_0
Amplitude_N-1_Channel_1
```

Channels are numbered starting at zero.

A warning message should be generated if the total number of samples is not evenly divisible by the number of channels.

7 Warning messages

An application should allow the user to turn off warning messages.