**Audio Descriptors**

The text document containing the audio descriptors prints a separate line for each audio file. Each line contains the following information in this order:

Server File Path, Title, Artist, Album, Year, Genre Duration, BPM, Loudness, Key, Scale, Chords Key, Chords Scale, Chords Changes Rate, Chords Number Rate, Danceability, Bassiness, Dynamic Complexity, Zero Crossing Rate, Intensity.

**Descriptions**

Server File Path:

The full file path to the file as located on file storage server. All songs will be in the format:

“/home/SongUpload/Songs/INSERT\_FILENAME\_HERE.mp3”

Title:

The title of the song

Artist:

The artist of the song

Album:

The album the song is associated with

Year:

The year the song was published

Genre:

The main genre classification of the song

Duration:

The length of the song (in seconds)

BPM:

The average beats per minute of the song. Also known as tempo.

Loudness:

The loudness of a signal, which is defined by Steven's power law as its energy raised to the power of 0.67. Can be thought of as how load the song is perceived to be.

Key:

The key the song is in. This values is a string containing a representation of one of the following keys:

A, A#/Bb, B, C, C#/Db, D, D#/Eb, E, F, F#/Gb, G, and G#/Ab.

(Note: the / means the two keys are the same)

Scale:

The scale the song is in. Either “major” or “minor”

Chords Key:

The most frequent chord of the progression.

See: <http://essentia.upf.edu/documentation/reference/std_ChordsDescriptors.html>

Chords Scale:

The scale of the most frequent chord of the progression (either 'major' or 'minor')

Chords Changes Rate:

The rate at which the chords change.

Chords Number Rate:

The ratio of different chords from the total number of chords in the progression

Danceability:

Yeah seriously. Normal values range from 0 to ~3. The higher, the more danceable.

Bassiness:

How much “bass” in the song. Sums the spectral energy contained in the bottom 4 bark bands. The higher the number, the more bass frequencies in the song.

Dynamic Complexity:

The dynamic complexity is the average absolute deviation from the global loudness level estimate on the dB scale. It is related to the dynamic range and to the amount of fluctuation in loudness present in a recording.

Zero Crossing Rate:

The number of sign changes between consecutive signal values divided by the total number of values. Noisy signals tend to have higher zero-crossing rate.

Intensity:

This algorithm is classified as not very reliable on the Essentia Website but we thought we would include it anyway. The output is the classification of the song as either relaxed (-1), moderate (0), or aggressive (1).

**Example output:**

/SongUpload/Songs/welcome-to-the-black-parade\_my-chemical-romance\_the-black-parade.mp3;; Welcome to the Black Parade;; My Chemical Romance;; The Black Parade;; 2006;; Alternative;; 320;; 150.083251953;; 3264.34155273;; G;; major;; G;; major;; 0.0686757490039;; 0.00231816875748;; 1.08423650265;; 23.0910757326;; -20.5461921692;; 0.0607157088816;; 0;;

All on one line of a text document though of course

**Algorithm Calculation**

Just an idea…

You could find the original values of the song the user picked and compare values of all other songs to it.

For string values, you could just compare if the strings match.

For number values, you could take the values of the user-picked song and calculate how much percent away from that value the other songs’ values are. For a low percentage, give a high number to that song (such as +10) and for a high percentage give a low value (such as +1). Then sum all of these to get a number for each song. These numbers would be from lowest to highest on how well they match.

Note: For songs the user likes, you could just add +5 and for songs the user dislikes you could add -5 or even -20