SOFTWARE REQUIREMENTS SPECIFICATION

Emertxe Information Technologies (P) Ltd

MP3 Tag Reader

VERSION: 0.1 REVISION DATE: 16-06-2014

Version	Date	Changed By	Modifications
0.1	16-06-2014	Emertxe	Initial Draft

Table of Contents

Introduction....3
Scope....3
Development Environment....3
User Interface Requirements....3
Functional Requirements....4
Design details....5
About (mp3) ID3 tag....5
  ID3v1 & ID3v1.1....5
  ID3v2 , ID3v2.3 & ID3v2.4....6
  ID3v2 header....6
  ID3v2.2 Frame header....6
  ID3v2.3 or ID3v2.4 Frame header....6
Before you start....7
Coding guidelines....7
Testing guidelines....7
References....7

Introduction

MP3 tag reader is a software which will read and display MP3 (ID3) tag information from MP3 files. The software will be desktop based and not web based.

This documents is the System Requirements Specification (SRS) for MP3 tag reader. The product will be useful to individuals who wish to view and collect mp3 tag data. This project can be extended to implement a tag editor, where-in users can modify mp3 tag information.

Scope

The purpose of this SRS is to completely capture all the requirements pertaining to mp3 tag reader. It will be used for determining the architecture, and preparing design document.

Development Environment

We will be using C programming language and command line interface tool to manupulate metadata of mp3 file.

User Interface Requirements

The Main UI has following components:

1. A help menu
    - This must display a help window showing all options.

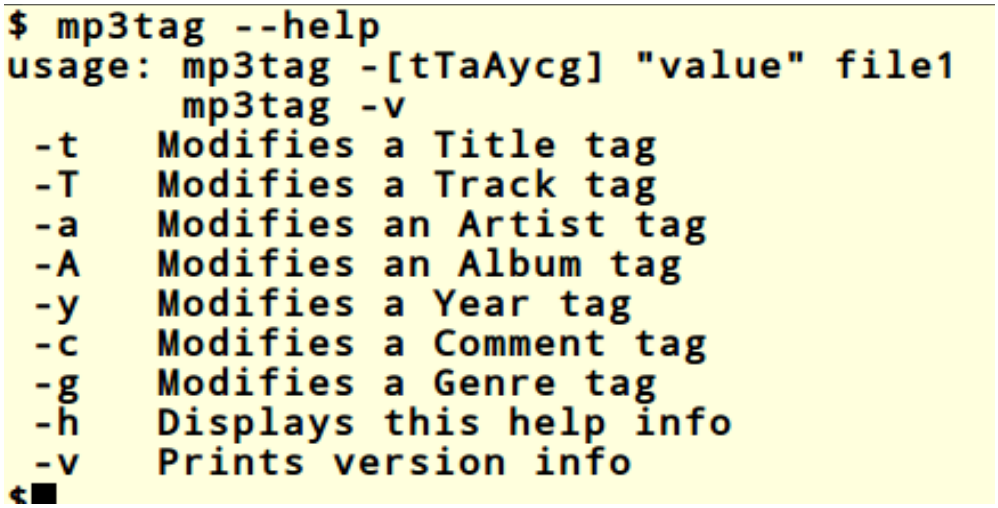


Illustration 1: Help menu

2. All information
    - This must display all metadata related to mp3 file.

```
$mp3tag sample.mp3

sample.mp3      [ 7.88 MB ]
--------------------------------------------------------------------------------
Time: 04:03     MPEG1, Layer III        [ ~271 kb/s @ 44100 Hz - Joint stereo ]
--------------------------------------------------------------------------------
ID3 v2.3:
title: Sunny Sunny              artist: Yo Yo Honey Singh
album: Yaariyan         year: 2013
boo
track:          genre: Pop (id 13)
Comment: [Description: Comment] [Lang: XXX]
Downloaded From Internet
```

Illustration 2: Sample output

Functional Requirements

This section list the functional requirements for the application.

Mandatory features:

1. This application, should able to handle all ID3 versions (Exept v2.4) tags.
2. Display which version of ID3 tag is used.
3. Display all the metadata information from ID3 tag.
4. User should able to change tags according to options given (Refer: illustration 1).
5. Should display a help screen on request (-h option).
6. If any image embedded in file, display details about image (Type, path and size).
7. If ID3 tag not found display proper error messages.

Additional features: (Get extra credit)

Implement tag editor

1. Add an option to extract the album art (image) from file .
2. Add an option to delete all tag datas from the file.
3. In corporate ID3v2.4 version .
4. It should be possible to delete a selected tag by options.

Design details

Create a simple design document before writing the code for the project. It is important to keep in mind that design should be object oriented and modular. Having modular code helps in allocating different modules/functionality to every team member. Thereby speeding up the implementation. Ensure that every team member make significant contribution to every phase of the project including design and implementation.

To re-emphasise, this is a team project and not an individual project. Please ensure good communication and team spirit. Choosing a team leader help in co-ordination and decision making.

About (mp3) ID3 tag

ID3 is a metadata container most often used in conjunction with the MP3 audio file format. It allows information such as the title, artist, album, track number, and other information about the file to be stored in the file itself.

There are two unrelated versions of ID3: ID3v1 and ID3v2. You will have to first detect the ID3 version before getting the tag details.

ID3v1 & ID3v1.1

Illustration 3: ID3v1.0

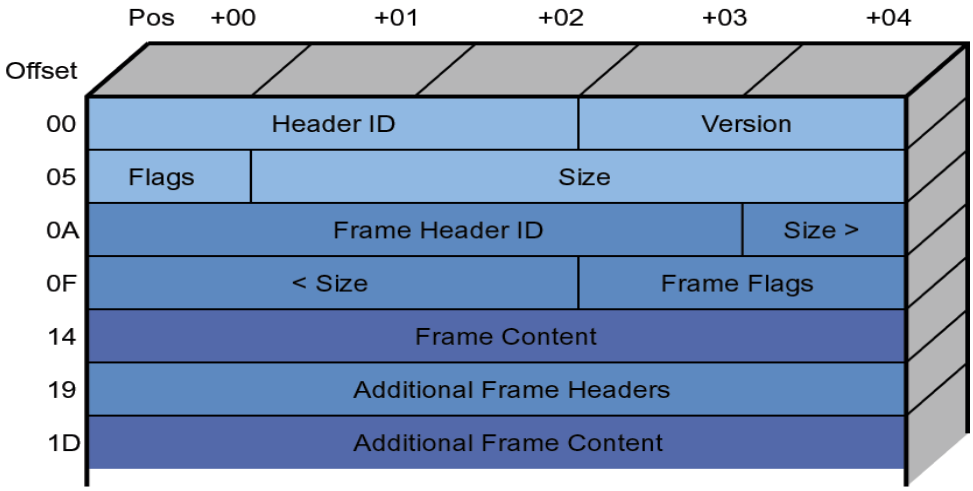
Illustration 4: ID3v1.1

Song title	30 characters
Artist	30 characters
Album	30 characters
Year	4 characters
Comment	30 characters
Genre	1 byte

Table 1: Id3v1.0

Total 128 bytes used for storing these metadata. If you one sum the the size of all these fields we see that 30+30+30+4+30+1 equals 125 bytes and not 128 bytes. The missing three bytes can be found at the very end of the tag, before the song title. These three bytes are always "TAG" and is the identification that this is indeed a ID3 tag. As all artists doesn't have a 30 character name it is said that if there is some bytes left after the information is entered in the field, those bytes should be fille with the binary value 0.

ID3v2 , ID3v2.3 & ID3v2.4



An ID3v2 tag starts with a tag header followed by one or more frames. Each frame in turn consists of a header and (usually) some kind of content.

The ID3v2 tag is more flexible and hence more difficult to work with. An ID3v2 tag has a signature code of "ID3x" where x is the sub-version number of the tag. Typically ID3v2 tags are found at the beginning of an MP3 file but this is not an absolute restriction. What happens next depends on the subversion of the ID3v2 tag. As far as we can tell, there have been three versions so far: 2,3 and 4.

ID3v2 header

ID3v2/file identifier "ID3"

ID3v2 version $03 00 (2, 3 or 4)

ID3v2 flags %abc00000

ID3v2 size 4 * %0xxxxxxx (Total size of ID3 tag)

ID3v2.2 Frame header

Frame ID $xx xx xx(Three characters)

eg:**TAL** Album/Movie/Show title

TOA Original artist(s)/performer(s)

Size $xx xx xx(Three characters)

ID3v2.3 or ID3v2.4 Frame header

Frame ID $xx xx xx xx (Four characters)

eg: **TALB** Album/Movie/Show title

TIT2 Title/songname/content description

Size $xx xx xx xx (Four characters)

Flags $xx xx

Before you start

- Understand the project specification – Please ensure that you understand the project specification. Clarify all doubts. Visualize what is the input and output.
- Do a feasibility study – Check if the project is feasible to implement with the allocated time and resources. If you feel certain features could not be implemented within the allocated time, talk to your instructor.
- Gain technology skills – If you need to learn specific technologies, please get prepared on the required topics.

Coding guidelines

Follow coding standards to ensure that code is more readable, re-usable and well commented.

- Use proper naming conventions for classes, variables, functions, data types, file names and directories.
- Do not hard code values.
- Arrange class definition into separate headers an functions in seperate .java files.
- Add block comments for files, and functions. Apart from this add comments whereever applicable.

Testing guidelines

In SDLC testing is an important phase of the project. Ensure that there are enough test cases listed. Applicable is of good quality only if it passes all the test cases.

References

1. Wikipedia article on ID3 tag - http://en.wikipedia.org/wiki/ID3
2. ID3 tag standard website - http://id3.org
3. ID3 tag version 2.3 standard - http://id3.org/id3v2.3.0