
mp3fm Documentation

Release 1.0.1

Akshit Agarwal

July 27, 2013

CONTENTS

1	Introduction to MP3fm	3
1.1	Features	3
2	Libraries Used and Install	5
2.1	Libraries Used	5
2.2	Installation	5
3	Working Screenshots	7
4	Usage & Download	13
4.1	Unix	13
4.2	Windows	13
5	Code	15
6	License	17
7	Indices and tables	19

MP3fm stands for “MP3 Folder Making app” which AUTOMATICALLY Pack Songs into folders according to user choice from TITLE/ARTIST/ALBUM/YEAR/DURATION/COMMENT.

I believe that for Music Lovers its a big problem to keep songs organized into folder, so **mp3fm** is a simple solution to that problem. Just run the app, select the folder which contain the songs and it will give you the option of Packing the Songs into folders corresponding to the properties choosen by you from Album(Movie)/Artist/Year/Comments/Title/Duration or Unpacking songs from already existing folders or Updating the Properties of songs present in that folder.

INTRODUCTION TO MP3FM

I believe that for Music Lovers its a big problem to keep songs organized into folder, so [mp3fm](#) is a simple solution to that problem.

Just run the app, select the folder which contain the songs and choose from the features/options it provides:

1.1 Features

- **PACK** : Move songs into folders based on property choosen by user from Album(Movie)/Artist/Year/Comments/Title/Duration.
- **UNPACK** : Move songs into choosen folder from already existing folders inside it.
- **UPDATE** : Update the Properties of all the songs(ID3 metadata) present in the choosen folder.

LIBRARIES USED AND INSTALL

2.1 Libraries Used

- **Mutagen** : Helps in reading and writing Song Properties(ID3 Metadata).
- **MusicBrainz-ngs** : Helps in accessing the online Music Brainz database.
- **EasyGui** : GUI building library.

2.2 Installation

```
$ pip install -U mp3fm
```


WORKING SCREENSHOTS



Figure 3.1: Songs present in a folder

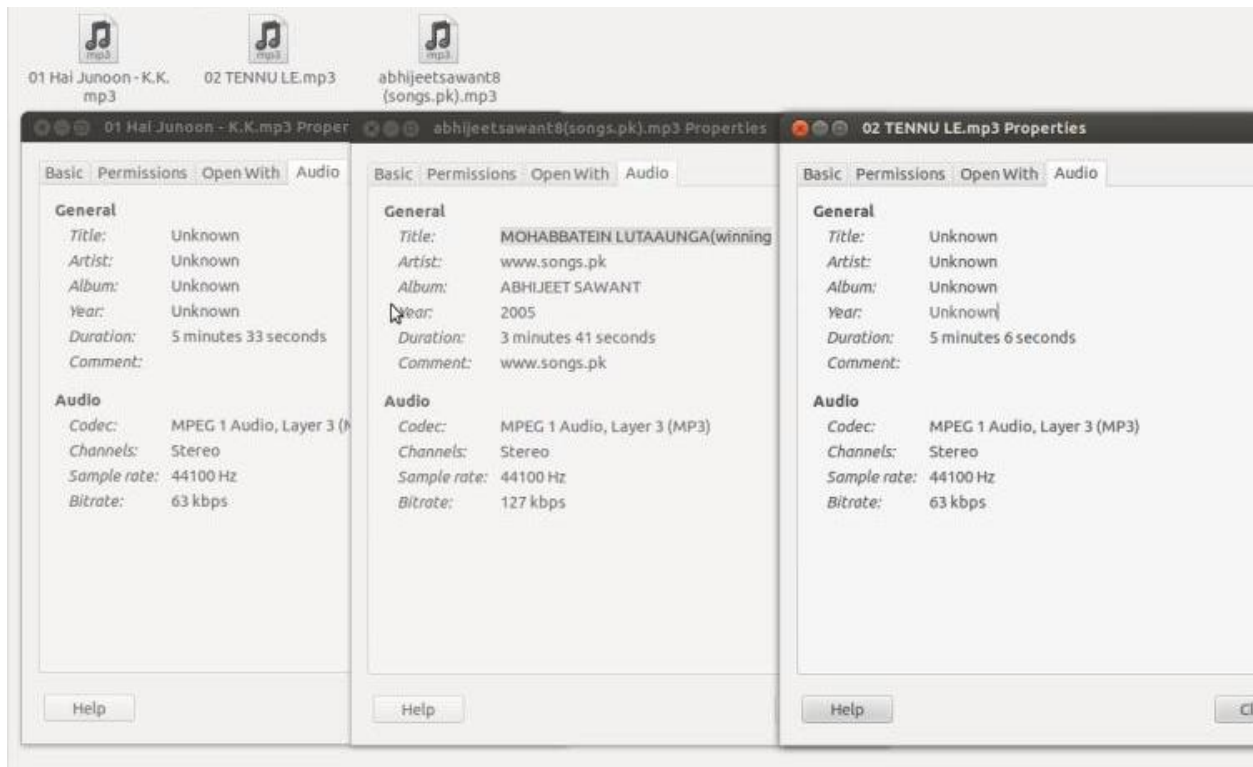


Figure 3.2: Songs properties before Update where most of the fields are Unknown or Wrong

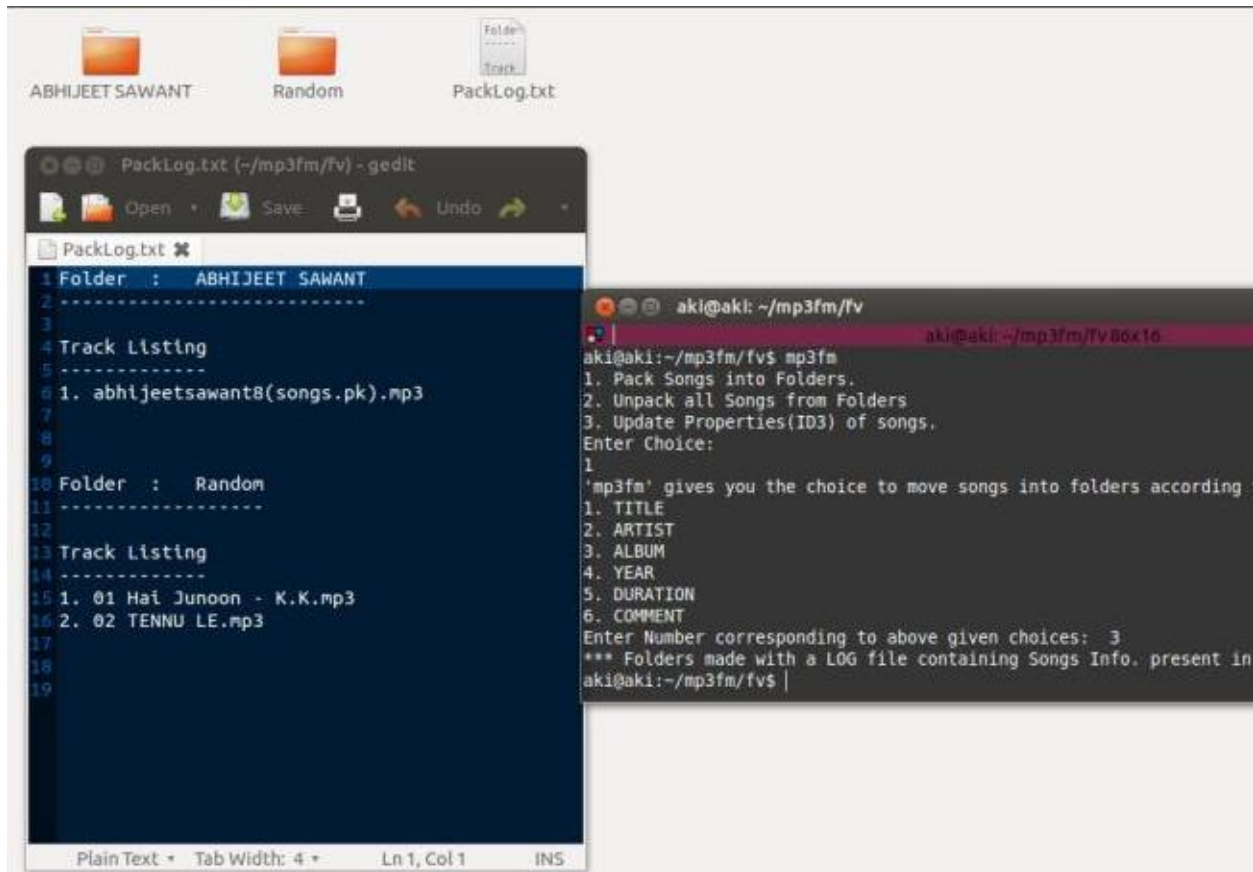


Figure 3.3: Packing Songs where most of the them move into Random folder as their properties are Broken or Wrong Pack Log file also shown

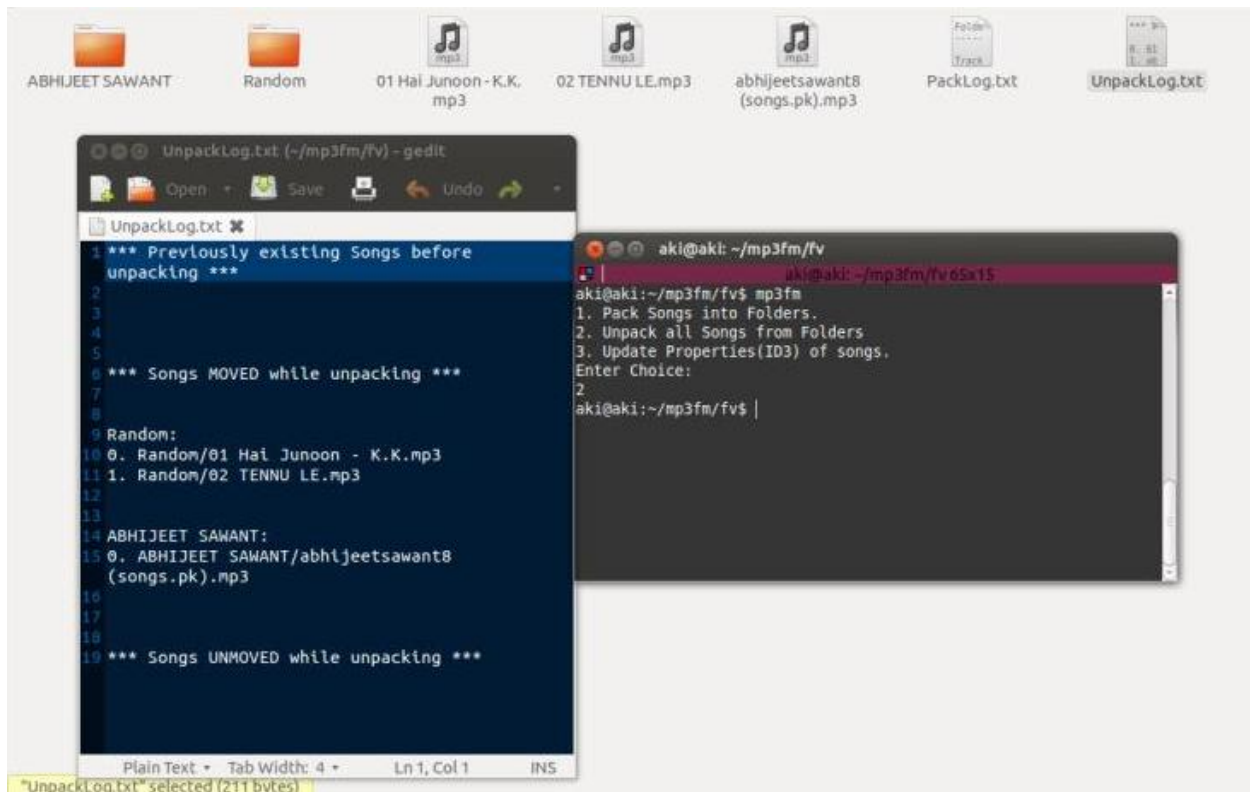


Figure 3.4: Unpacking Songs from folders Unpack Log file shown

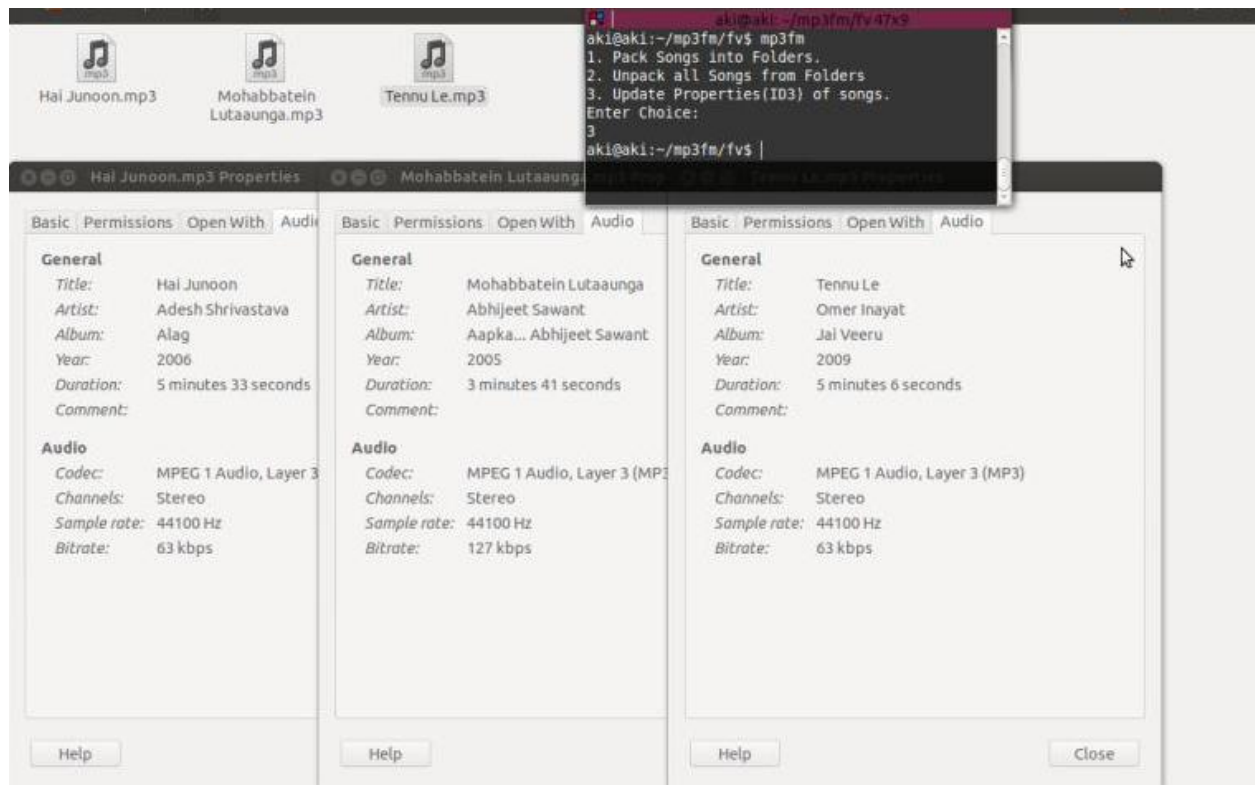


Figure 3.5: Songs properties repaired fully after Update so that iPod/Music Players can read songs properly and packing occurs efficiently

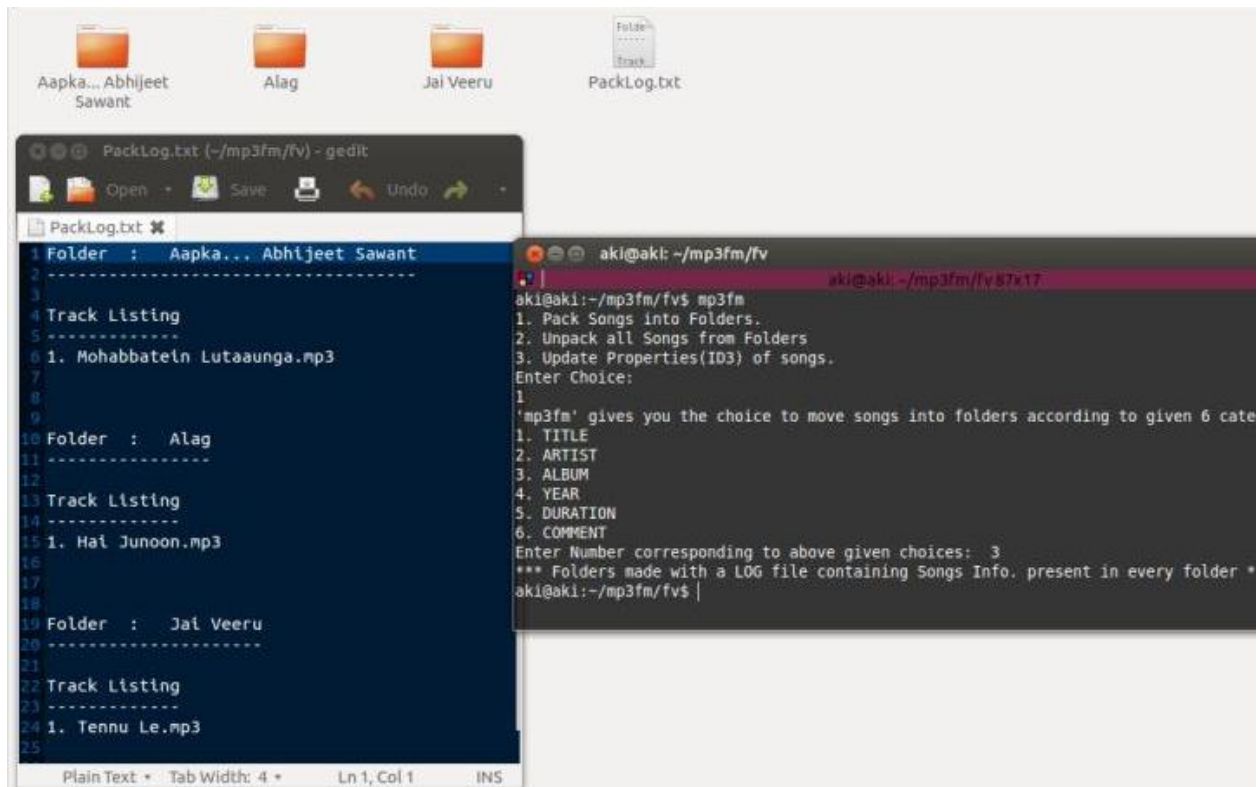


Figure 3.6: Songs packed much efficiently as compared to the packing done before Updating them

USAGE & DOWNLOAD

Just run the App and follow the instructions shown in it:

4.1 Unix

```
$ mp3fm
```

Download `mp3fm.tar.gz`

4.2 Windows

Open

```
mp3fm.exe
```

Download `mp3fm.zip`

CODE

```
class mp3fm.PackSongs (input_folder, tag='')
    It pack songs into folders according to property choosen by the user making easy to manage songs and keep
    them into folders

    change_cwd ()
        Changing current working directory to user input folder to access songs

    check_folder (folder_name)
        Checking if required folder exists otherwise create a new one

    find_info (song_name)
        Finding song info

    generate_log ()
        Generate a log file stating the songs inside each newly created directory

    list_mp3files (folder='')
        Storing list of all .mp3 files

    move_song (song, folder_name)
        Moving song to specific folder

    put_songs ()
        Putting all songs in folders according to user choice

class mp3fm.UnpackFolders (input_folder, tag='')
    It unpacks all the folders so that all songs comes out from folders and reside at one place

    generate_log ()
        Redefining the function which is Generating log according to different types

    list_folders ()
        Listing all folders

    move_song (folder_name, song)
        Redefining the function which is Moving song to specific folder

    unpack ()
        Running all other functions to unpack folders

class mp3fm.UpdateSongInfo (input_folder, tag='')
    It updates all the songs information using online Music Brainz database.

    authenticate ()
        Authenticate the client to query the Music Brainz Server
```

convert_to_unicode (*string*)

Converting string into unicode string using UTF-8 format

extract_info ()

Extracting information from result found in above function

save_info (*song_name*)

Saving new song info

search_musicbrainz (*song_name*)

Searching music brainz db for particular song

update_id3 ()

Function calling all other functions to update song info

`mp3fm.main` ()

Main function which is asking user for choices and making appropriate class objects and calling appropriate functions corresponding to them

LICENSE

MIT License Copyright 2013 Akshit Agarwal

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

INDICES AND TABLES

- *search*

PYTHON MODULE INDEX

m

mp3fm, ??