

Summary

From MP3Tunes.com "MP3tunes is a Music Service Provider (MSP) and the home of MP3Tunes Locker, the only secure, online music space to feature unlimited listening." MP3Tunes provides an "online music space" for its users to store their entire music collection with a web interface for playback and a python desktop client for syncing tracks up and down. They also provide an API so third parties can include the same functionality in their applications.

Currently the Amarok 2 trunk supports basic Locker functions, including browsing the locker, streaming tracks, and displaying album art. In my project I will enhance MP3Tunes and Amarok integration by adding support for downloading and uploading media content between an Amarok collection and the MP3Tunes locker. This will allow users to keep their remote and local collections in sync as well as provide a means for off site backup. From this initial implementation a service agnostic syncing framework can easily be extrapolated to enable syncing for any service with write capability.

Background

MP3Tunes has not yet provided the necessary documentation for 3rd party developers to utilize syncing actions. However, Nikolaj has been in contact with MP3Tunes developers and they have expressed significant interest in pushing this integration forward. They have promised to clean up their API and provide necessary documentation. Also, it should be noted that the LockerSync software (which fully implements the API) is licensed under the GPL.

I have been an active user on the MP3Tunes forums for several months now and as a result I am not only familiar with the MP3Tunes service, but have established good relationships with the developers/administrators.

Details

There is some significant work to be done outside src/servicebrowser before any syncing

framework can be used. First is the matter of copying tracks between collections using CollectionLocation framework. This needs to be fleshed out so copying can actually occur between collections. In the case of copying tracks to the local collection this will require prompting the user for a location to store tracks. Ideally the already existing (but unported?) "Organize to collection" feature can be utilized.

During this time I will be in constant contact with MP3Tunes developers to ensure the API documentation is produced in time for the next step. Once the groundwork is laid for copying and the API documentation is available the next step will be implementing the API method calls into the service architecture. Also, during this period a ServiceCollectionLocation class will be developed that defines the means by which tracks are copied (uploaded) to services. After the method calls are in place the corresponding UI widgets will be inserted so the API method calls can be invoked.

If time permits one or many of the following features will be included:

An Auto Synchronization feature would wrap this entire project into a tidy package for Amarok users. If something new is added or changed to a local collection the MP3Tunes service will detect this and upload it to the user's Locker automatically. Similarly, if a track is added to the Locker the service will download the track and add it to a collection of the user's choice. There are various design considerations that have to be researched before I can provide an accurate time line for this feature. Things such as, what constitutes change, handling said change, prompting the user before deleting/adding/editing tracks, and possibly allowing only a subset of a collection to be synced.

For the Amarok implementation of the MP3Tunes API to be feature complete it has to include Locker Search and Playlist Management functions. I will code support for searching the Locker collection via the searchWidget at the top of the collection panel. Using the Playlist management API features I will code 'add', 'delete', and 'edit' methods into the service and possibly create a new section in the playlist view where a user can perform those functions on stored playlists.

Deliverables

Mandatory:

1. Generic copy between Locations
2. Downloading of content to a local Collection from a MP3Tunes locker
3. Uploading of content to a MP3Tunes locker from a local Collection
4. Generic Syncing framework built into service architecture

Optional:

5. Automatic Synchronization
6. Locker Search
7. Locker Playlist Management

Timetable (Deliverable #)

April 14th – May 26th

- Become more familiar with the Amarok codebase
- Research synchronization options
- Stay in contact MP3Tunes staff

May 26th – June 8th

- Implement CollectionLocation functionality (1)
- Finish “Organize to Collection” port

June 9th – June 29th

- Implement downloading functionality (2)
- Implement uploading functionality (3)
- Add UI widgets

June 30th – August 10th

- Extrapolate syncing service (3)
- Depending on time allowances work on most realistic optional deliverables

Biography

I am a rising Junior Computer Science and Philosophy double major at Virginia Polytechnic Institute and State University (Virginia Tech) in Blacksburg Virginia. I have been using Linux since 2004 and KDE since 2005. Last fall I had a brief break in my studies and successfully implemented album covers into Amarok's MP3Tunes service, but due to time constraints

was unable to continue my foray into open source programming. This summer, however, is dedicated solely to hacking . I am in the process of applying for an Independent Study course in the Fall 2008 semester that would allow me to go on hacking after the summer. I am also an avid Gentoo user and getting involved with Gentoo's Kernel Security project.