xspf webmusicplayer

webmusicplayer

.short description

a flash-based web application that uses the xspf playlist format to play mp3 songs.

slim version

extended version

button version

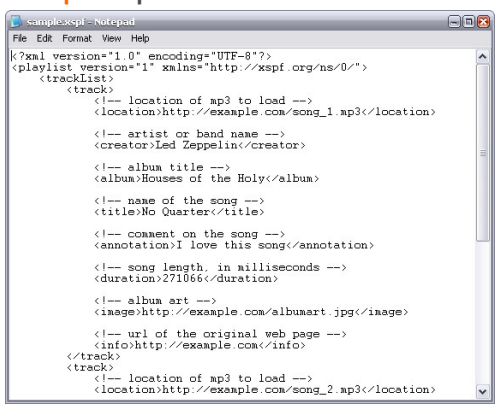
3 different versions

webmusicplayer

.about xspf

xml shareable playlist format (xspf) is an open xml-based format for sharing digital media playlists.

sample xspf document



tags for file path, track name, artist, album, e.t.c. defined

unlike m3u,
- xspf is xml

unlike smil,
- xspf is simple

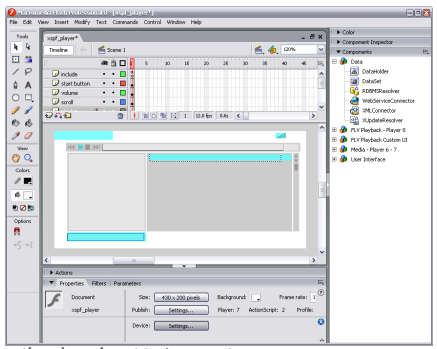
unlike asx,
- xpsf is open

webmusicplayer

.project insight

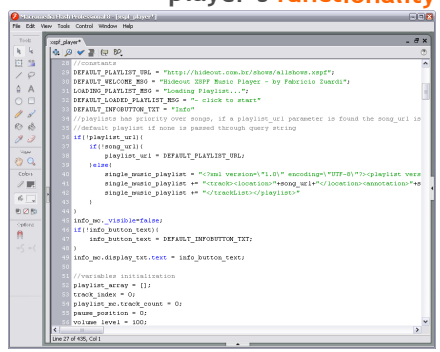
developed using macromedia flash, a full-featured authoring tool that allows you to create anything from a simple animation to a complex web application.

player's gui



the player's gui is drawn using Flash drawing tools

player's functionality



the player's functionality is coded using ActionScript

.project insight

upon compiling the source files (.fla + .as), what we get is a single file (.swf) which, in general, is called a flash movie.

.fla file holds player's interface

.as file holds actionscript code

flash's built-in complier

xpsf web music player

Basically, a .swf file is the compressed, non-editable, playable version of the .fla file

.system requirements

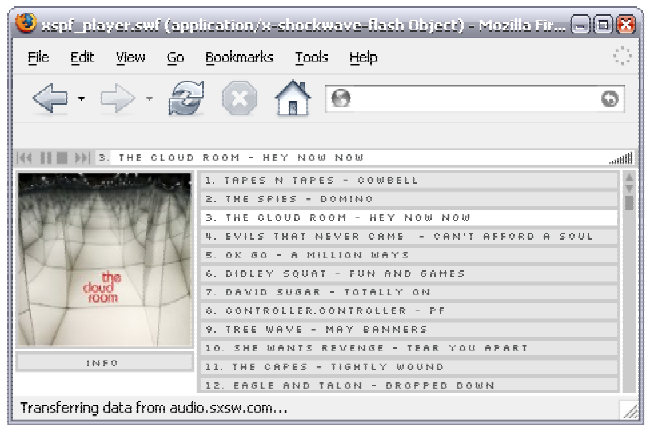
to play a flash movie, you'll need to have flash player installed. flash player comes in two modes:

- as a stand-alone application, and
- as a browser plug-in (flash player plug-in)

flash player (stand-alone)

the player is run directly from desktop

flash player (plug-in)



the player is run through a browser window

.code walkthrough

xspf web music player is mainly written following the procedural programming paradigm.

.main functions' signatures

```
loadPlaylist() //load xspf file
playlistLoaded(success:Boolean) //parse xpsf file's contents
loadTrack() //load track to be played
playTrack() //play track
volumeChange() //turns the volume up/down
```

.planned contribution

- extend player's functionality
  - suffle mode
  - track/playlist loop
  - seek feature
  - track's duration indicator
  - multiple playlists support

- enhance player's performance
  - slim/extented consolidation
  - multi-language support
  - better buffer implementation
  - xml settings file
  - minor fixes

.way to ‘glory’:-)

- write new functions,
  such as shuffleTracks(), loopTracks() e.t.c.
- modify existing functions,
  such as the loadPlaylist() function to make it support multiple playlists.
- create new classes,
  such as a Config class that will be responsible for applying player’s settings parsed from an external xml file.
- create visual objects,
  such as extra control buttons (which are in fact class instances).

.conclusion

- questions?