

**GUI Programming**  
Program #3 – **eXtreme Player**  
Due Thurs, Nov. 1  
100 Points

## **Overview**

You are to write a program using Visual Basic .NET that allows a user to organize and play sound files like the Windows Media Player and iTunes. The sound files can be viewed by artist, album, or title. Music can also be organized into play lists. A search option will allow the user to search for music based on artist, album, or title. A list of all sound files and play lists will be stored in XML format so a database is not necessary.

## **Requirements**

The following specifications must be met in order to receive 95 points:

1. The program should run in a non-standard Window using a shape that you define. There should be no standard menu.
2. The listing of all songs should be accessible from a tree view control. The control should allow songs to be selected by album, artist, and genre. All playlists should be accessible in the tree view as well.
3. Songs should display in a customized list view control that shows each song's title, artist, album, length, and genre. A song can be played by double-clicking the song in the data grid or right-clicking the song and selecting the play option from the pop-up menu.
4. The program should allow songs to be added to the listing of songs by launching an open dialog box and allowing the user to select a song. After selecting a song, the program should read the metadata stored in the music file (if available) and add the song to the tree view.
5. The program should allow the user to create playlists. Creating a new play list should be done with a dialog box that allows the user to enter the name of the playlist and select from a list of all songs to be added to the playlist. The user should also be able to add a song to a play list by right-clicking on a song and selecting the playlist from the "Add to Playlist" menu option. A playlist name can be changed by right-clicking the playlist in the tree view and selecting "Rename". A playlist can be deleted by right-clicking the playlist in the tree view and selecting "Delete".
6. The program should allow the user to enter a search string and press a button that will populate the song list with any song whose title, artist, album, or genre match any part of the search string.
7. The program should make the Media Player control available to play, pause, stop, advance, and change the volume.
8. The list of all song files and play lists should be stored in a file called music.xml. This file should reside in the same folder as the eXtreme Player's .exe. The file should be loaded when the player is executed and re-produced when the player terminates.
9. Songs can be removed from the list of All Music by right-clicking the song and selecting "Remove" from the pop-up menu. Songs can be removed from Playlists by selecting "Remove from Playlist" or from the library by selecting "Remove from Library".
10. Song information can be modified by right-clicking the song and selecting "Edit" from the pop-up menu. This should display a dialog box displays the title, artist, album, genre, filename, and length. The user should be able to edit the title, artist, album, or genre. An OK and Cancel button should be present on the dialog box.

## Additional Requirements

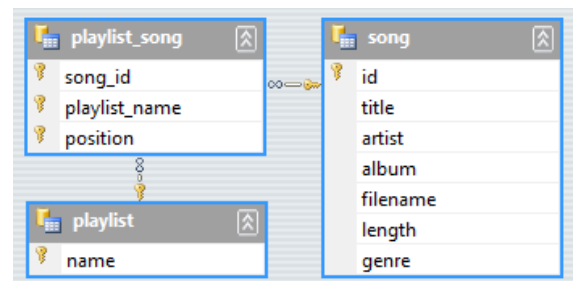
Each additional requirement is worth 5 points:

1. Allow the user to add songs to a playlist by dragging and dropping the playlist item from the song list to the playlist name in the tree view.
2. Add an icon to each option in the tree view, similar to the Windows Media Player.
3. Use a custom button control to implement the close button on the main form. The button should turn into a different graphic when the user puts the cursor over the button, and it should change back when the mouse is moved off of it.
4. Implement a Queue that plays multiple songs, one after the other. For example, if the user selects multiple songs and selects "Queue-It-Up" from the pop-up menu, all songs should be played in order. Also if a song is played in a playlist, all the songs in the playlist should be played.

## Music.xml

Your music.xml file should conform to the following example and use the XML schema file (music.xsd) given to you by the instructor:

```
<?xml version="1.0" standalone="yes"?>
<music xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <song>
    <id>1</id>
    <title>Ode To Camaraderie</title>
    <artist>Doc Burkes</artist>
    <album>Music That Puts You to Sleep</album>
    <filename>C:\mymusic\ode_to_camaraderie.mp3</filename>
    <length>02:56</length>
    <genre>Classical</genre>
  </song>
  ...
  <playlist_song>
    <song_id>1</song_id>
    <playlist_name>Cool stuff</playlist_name>
    <position>1</position>
  </playlist_song>
  <playlist_song>
    <song_id>3</song_id>
    <playlist_name>Cool stuff</playlist_name>
    <position>2</position>
  </playlist_song>
  <playlist_song>
    <song_id>2</song_id>
    <playlist_name>Quiet music</playlist_name>
    <position>1</position>
  </playlist_song>
</music>
```



## Grading

Your application should work similar to my eXtreme Player located at \\cs1\Classes\comp445\eXtremePlayer. I will replay your music.xml file with my own when testing your program. Note that my eXtreme Player is very ugly and does not have all the requirements implemented. It is just something to go by.

If you fulfill all the requirements, the most points you can receive on this project is 95 points. You will need to fulfill at least one of the additional requirements to get a higher score. You could get a 115 out of 100 points on this project.

Zip up your entire project (using WinZip or a similar program) and submit it to Easel *before* class on the due date.