

Documentation - Modular Music Kit

[1 - Information & Support](#)

[2 - Quick-Start-Guide](#)

[3 - Overview](#)

[4 - Examples](#)

[5 - Implementation](#)

[6 - Customization](#)

1 - Information & Support

Created by Robin Gerndt, published under the name bibCorp.

I created this kit to make sure that users have an easy way to make simple modifications to my soundtracks, but it can very easily be set up to play ANY music you have imported into Unity! If you have imported a soundtrack that is split up into separate audio channels (or if you buy one of my multichannel [Modular Music Packs from the asset store](#)) you can very easily make different variations that fit different moments in your game!

I want this asset to be very easy to use. I will continue to improve this documentation based on feedback. If you need any help or have any suggestions, contact me!

www.RobinGerndt.com/Support

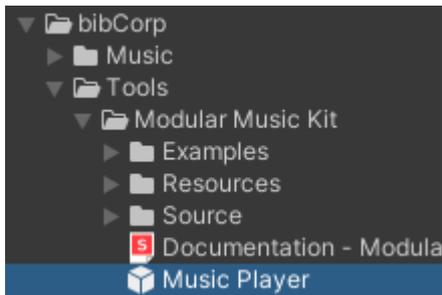
bibbis.inc@gmail.com

2 - Quick-Start-Guide

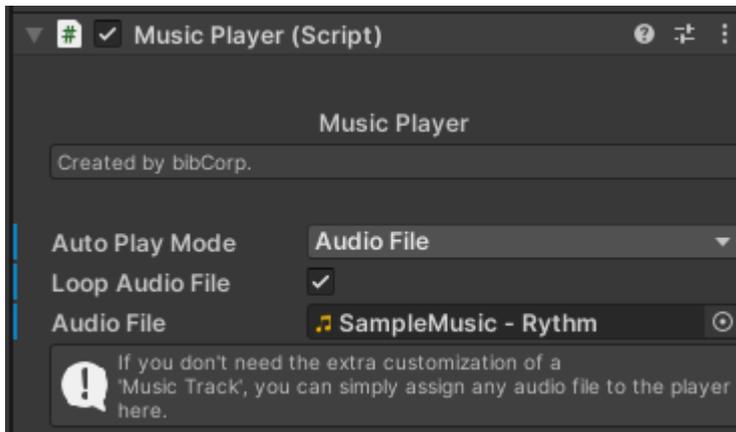
For more detailed information, please see the ['Implementation'](#) section.

To get started playing audio files immediately:

1. Navigate to: **'Assets\bibCorp\Tools\Modular Music Kit'** and drag the **'Music Player'** prefab into your scene.



2. In the inspector, set **'Auto Play Mode > Audio File'**



3. In the inspector, **assign an audio file from your assets folders.**

That's it. You are done.

Your music will now play in this scene, and will continue to play without interruption even if you change to a different scene. The music can be manually stopped by simply disabling the Music Player game object, or by using the code API. (i.e. `MusicPlayer.Stop()`)

3 - Overview

The three main components in this package are:

1. **The 'Music Player' prefab.**
 - Location: Assets\bibCorp\Tools\Music Player.
 - This prefab is coded to not be destroyed when you change scenes in your game, allowing for music to still play between levels and loading screens.
 - Can play any audio file imported into Unity, as well as 'Music Track' assets.
2. **The 'Music Track' asset.**
 - Created by right-clicking anywhere in the assets folder and selecting: Create > bibCorp > Music > Music Track.
 - Can store any music file you have imported to Unity.
 - One Music Track can contain multiple music channels with individual volumes, or just one single channel with a complete soundtrack.
(All Music Packs that I publish include these optional multi-channel Music Tracks that allow the user to quickly make simple modifications to my music)
3. **The 'Playlist' asset.**
 - Created by right-clicking anywhere in the assets folder and selecting: Create > bibCorp > Music > Playlist.
 - A simple playlist of Music Tracks that can be played in the Music Player instead of a single track, and can be set to loop.

4 - Examples

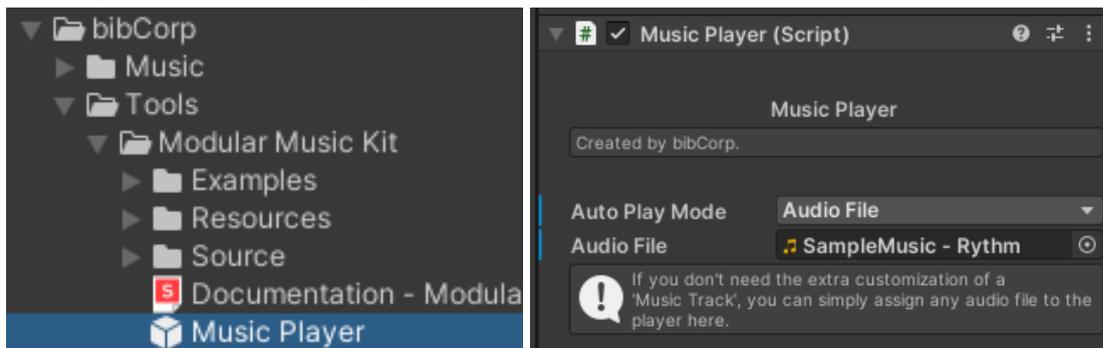
The 'Examples' folder: **Assets\bibCorp\Plugins\Music Player\Examples** includes sample music, example code and a scene with the example code hooked up to simple UI buttons.

5 - Implementation

There are two ways to implement the Music Player:

Codeless implementation:

1. Place the **'Music Player' prefab** into your scene and configure it in the inspector to autoplay a Music Track, Playlist or audio file.



2. It's possible to place multiple Music Player prefabs into the same scene, disabling them, and simply enabling them when it's time to play their music.
3. If there are multiple Music Players, the first player will become the master, and all the new Music Players will simply tell the existing one which Music Track or Playlist it should play, and then disable itself.

Code implementation:

1. Make sure to include the **'bibCorp'** namespace in your code.

```
using bibCorp;
```
2. Use these static functions to control the Music Player:
 - 2.1. `MusicPlayer.PlayMusicTrack()` - requires a `MusicTrack` asset reference.
 - 2.2. `MusicPlayer.PlayPlaylist()` - requires a `Playlist` asset reference.
 - 2.3. `MusicPlayer.Stop()` - simply stops any track or playlist.
3. A default Music Player prefab will automatically be instantiated when one of the `Play()` functions are used. It is also possible to manually instantiate Music Players if needed.

It is of course possible to mix and match how you implement the music into your game, just know that the system is set up using a singleton pattern, so there is always going to be one master Music Player and every time a new player is enabled or instantiated it will first check and see if a master exists, and if there is a master, the new player will ask the master to play the music and then disable itself.

6 - Customization

The soundtrack customization only works if the music has been exported into individual sound files by the creator. For example, the sample music included in this package has been imported into Unity as 3 individual sound files. Each sound file contains one instrument from the completed soundtrack. By adding all 3 sound files together in a 'Music Track' asset, all 3 channels can be re-combined into the completed soundtrack.

Music Tracks can be customized either directly in the inspector, or by using the included **Live Preview Scene** which lets you preview your modifications while you listen to the music. This scene is located in:

'Assets\bibCorp\Tools\Modular Music Kit\Resources'

Simply run the scene and use the on-screen controls to modify soundtracks.



1. Opens a list of all available 'Music Track' assets in your asset folders.
2. Un-check the box to disable a track.
3. Green indicator lights up when this channel is playing sound in the preview.
4. Click the title label to navigate to the track in your asset folders.

If you want to create a new variation without modifying the original, the easiest way is to make a copy of the original Music Track. Click the track title (4) to navigate to the track in your asset folders. Duplicate the asset (ctrl + D) and give the new variation a name. Run the live previewer scene and click the button to select track (1) and select the new variation you just created.