

# MUSIC PLAYER

## API REFERENCE

Freakshow Studio

---

## Table of Contents

---

<b>1</b>	<b>Namespace Documentation</b>	<b>1</b>
1.1	Package FreakLib	1
1.2	Package FreakLib.Music	2
<b>2</b>	<b>Class Documentation</b>	<b>3</b>
2.1	FreakLib.Music.MusicPlayer Class Reference	3
2.1.1	Detailed Description	6
2.1.2	Member Function Documentation	6
2.1.2.1	GetPlaylist	6
2.1.2.2	Next	6
2.1.2.3	Pause	7
2.1.2.4	PauseOrResume	7
2.1.2.5	Play	7
2.1.2.6	Play	7
2.1.2.7	Play	7
2.1.2.8	Previous	8
2.1.2.9	Stop	8
2.1.2.10	UnPause	8
2.1.3	Member Data Documentation	8
2.1.3.1	OnPause	8
2.1.3.2	OnPlay	8
2.1.3.3	OnPlaylistChange	8
2.1.3.4	OnStop	8
2.1.3.5	OnTrackChange	9
2.1.3.6	OnUnpause	9
2.1.4	Property Documentation	9
2.1.4.1	CurrentPlaylist	9
2.1.4.2	CurrentPlaylistName	9
2.1.4.3	CurrentTrack	9
2.1.4.4	CurrentTrackName	9
2.1.4.5	HistoryLength	10
2.1.4.6	IsPlaying	10
2.1.4.7	PlaylistNames	10
2.1.4.8	Playlists	10

2.1.4.9	PlayOnAwake	10
2.1.4.10	Playtime	10
2.1.4.11	PlaytimeNormalized	11
2.1.4.12	Volume	11
2.2	FreakLib.Music.MusicPlayer.MusicPlayerEvent Class Reference	12
2.2.1	Detailed Description	12
2.3	FreakLib.Music.Playlist Class Reference	13
2.3.1	Detailed Description	14
2.3.2	Constructor & Destructor Documentation	14
2.3.2.1	Playlist	14
2.3.2.2	Playlist	14
2.3.2.3	Playlist	14
2.3.3	Member Data Documentation	14
2.3.3.1	foldout	14
2.3.3.2	name	15
2.3.3.3	shuffle	15
2.3.3.4	tracks	15
2.3.4	Property Documentation	15
2.3.4.1	EnabledTrackIndices	15
2.4	FreakLib.Music.Track Class Reference	16
2.4.1	Detailed Description	17
2.4.2	Constructor & Destructor Documentation	17
2.4.2.1	Track	17
2.4.2.2	Track	17
2.4.3	Member Function Documentation	17
2.4.3.1	ToString	17
2.4.4	Member Data Documentation	18
2.4.4.1	clip	18
2.4.4.2	enabled	18
2.4.4.3	foldout	18
2.4.4.4	name	18
2.4.5	Property Documentation	18
2.4.5.1	Length	18
2.4.5.2	Plays	18

<b>Index</b>	<b>20</b>
--------------	-----------

# CHAPTER 1

---

## Namespace Documentation

---

### 1.1 Package FreakLib

#### Namespaces

- package [Music](#)

## 1.2 Package FreakLib.Music

### Classes

- class [MusicPlayer](#)

*This class implements a [MusicPlayer](#) for Unity.*

- class [Playlist](#)

*This class implements a [Playlist](#) for the [MusicPlayer](#).*

- class [Track](#)

*This class implements a single [Track](#) for the [MusicPlayer](#). Tracks are added to a [Playlist](#) that the [MusicPlayer](#) can then play.*

## CHAPTER 2

---

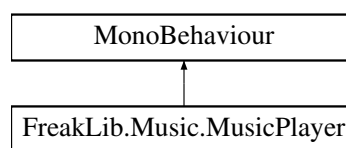
### Class Documentation

---

### 2.1 FreakLib.Music.MusicPlayer Class Reference

This class implements a [MusicPlayer](#) for Unity.

Inheritance diagram for FreakLib.Music.MusicPlayer:



#### Classes

- class [MusicPlayerEvent](#)

*A custom UnityEvent, which allows passing a [MusicPlayer](#) as an argument.*

#### Public Member Functions

- void [Play](#) ()

*Starts playing the current [Playlist](#), or the first defined [Playlist](#) if the current [Playlist](#) is not set.*

- void [Play](#) (string name)

*Starts playing the specified [Playlist](#).*

- void **Play** (**Playlist** playlist)

*Starts playing the specified **Playlist**.*

- void **PauseOrResume** ()

*Calling this will Pause the **MusicPlayer** if it is playing, or resume from pause if it is currently paused.*

- void **Pause** ()

*Pause the **MusicPlayer**.*

- void **UnPause** ()

*UnPause the **MusicPlayer** (resume from pause).*

- void **Stop** ()

*Stops the **MusicPlayer**.*

- void **Next** ()

*Skip to the next **Track**.*

- void **Previous** ()

*Skip to the previous **Track**.*

- **Playlist GetPlaylist** (string name)

*Get the **Playlist** with the given name.*

## Public Attributes

- **MusicPlayerEvent OnPlay**

*A UnityEvent that will fire when we start playing.*

- **MusicPlayerEvent OnStop**

*A UnityEvent that will fire when we stop playing.*

- **MusicPlayerEvent OnPause**

*A UnityEvent that will fire when we pause.*

- **MusicPlayerEvent OnUnpause**

*A UnityEvent that will fire when we un-pause.*

- [MusicPlayerEvent OnTrackChange](#)

*A UnityEvent that will fire when we change track.*

- [MusicPlayerEvent OnPlaylistChange](#)

*A UnityEvent that will fire when we change [Playlist](#).*

## Properties

- bool [PlayOnAwake](#) [get, set]

*Gets or sets a value indicating whether this [MusicPlayer](#) should play on awake.*

- float [Volume](#) [get, set]

*Gets or sets the volume.*

- int [HistoryLength](#) [get, set]

*Gets or sets the length of the history.*

- List< [Playlist](#) > [Playlists](#) [get, set]

*Gets the playlists.*

- List< string > [PlaylistNames](#) [get]

*Gets the playlist names.*

- [Playlist CurrentPlaylist](#) [get]

*Gets the current playlist.*

- string [CurrentPlaylistName](#) [get]

*Gets the name of the current playlist.*

- [Track CurrentTrack](#) [get]

*Gets the current track.*

- string [CurrentTrackName](#) [get]

*Gets the name of the current track.*



- bool [IsPlaying](#) [get]

*Gets a value indicating if we are currently playing.*

- float [Playtime](#) [get, set]

*Gets the current playtime, in seconds. Will be a value between 0 and the track length.*

- float [PlaytimeNormalized](#) [get, set]

*Gets the current playtime normalized. Will be a value between 0 and 1.*

### 2.1.1 Detailed Description

This class implements a [MusicPlayer](#) for Unity.

The [MusicPlayer](#) will play a [Playlist](#), either in order or shuffled, and allows pausing, skipping tracks forward and back, as well as seeking.

It will also send various UnityEvents when changing tracks, playlists and so on.

### 2.1.2 Member Function Documentation

#### 2.1.2.1 Playlist FreakLib.Music.MusicPlayer.GetPlaylist ( string *name* )

Get the [Playlist](#) with the given name.

Returns null if a playlist is not found.

Returns

The playlist.

Parameters

<i>name</i>	The name of the <a href="#">Playlist</a> .
-------------	--

#### 2.1.2.2 void FreakLib.Music.MusicPlayer.Next ( )

Skip to the next [Track](#).

### 2.1.2.3 void FreakLib.Music.MusicPlayer.Pause ( )

Pause the [MusicPlayer](#).

### 2.1.2.4 void FreakLib.Music.MusicPlayer.PauseOrResume ( )

Calling this will Pause the [MusicPlayer](#) if it is playing, or resume from pause if it is currently paused.

### 2.1.2.5 void FreakLib.Music.MusicPlayer.Play ( )

Starts playing the current [Playlist](#), or the first defined [Playlist](#) if the current [Playlist](#) is not set.

Calling Play will always start playing, regardless of the current state of the [MusicPlayer](#). It will also reset the [Playlist](#).

### 2.1.2.6 void FreakLib.Music.MusicPlayer.Play ( string *name* )

Starts playing the specified [Playlist](#).

The name should match a [Playlist](#) defined in this instance of the [MusicPlayer](#). If the specified playlist is not found, a `UnityException` is thrown.

Parameters

<i>name</i>	The name of the <a href="#">Playlist</a> to play.
-------------	---

### 2.1.2.7 void FreakLib.Music.MusicPlayer.Play ( [Playlist](#) *playlist* )

Starts playing the specified [Playlist](#).

The specified [Playlist](#) does not necessarily need to be defined in this instance of the [MusicPlayer](#), but can be generated and passed in as a separate instance.

Parameters

<i>playlist</i>	<a href="#">Playlist</a> .
-----------------	----------------------------

#### 2.1.2.8 void FreakLib.Music.MusicPlayer.Previous ( )

Skip to the previous [Track](#).

#### 2.1.2.9 void FreakLib.Music.MusicPlayer.Stop ( )

Stops the [MusicPlayer](#).

#### 2.1.2.10 void FreakLib.Music.MusicPlayer.UnPause ( )

UnPause the [MusicPlayer](#) (resume from pause).

### 2.1.3 Member Data Documentation

#### 2.1.3.1 MusicPlayerEvent FreakLib.Music.MusicPlayer.OnPause

A UnityEvent that will fire when we pause.

#### 2.1.3.2 MusicPlayerEvent FreakLib.Music.MusicPlayer.OnPlay

A UnityEvent that will fire when we start playing.

#### 2.1.3.3 MusicPlayerEvent FreakLib.Music.MusicPlayer.OnPlaylistChange

A UnityEvent that will fire when we change [Playlist](#).

#### 2.1.3.4 MusicPlayerEvent FreakLib.Music.MusicPlayer.OnStop

A UnityEvent that will fire when we stop playing.

#### **2.1.3.5 MusicPlayerEvent FreakLib.Music.MusicPlayer.OnTrackChange**

A UnityEvent that will fire when we change track.

#### **2.1.3.6 MusicPlayerEvent FreakLib.Music.MusicPlayer.OnUnpause**

A UnityEvent that will fire when we un-pause.

### **2.1.4 Property Documentation**

#### **2.1.4.1 Playlist FreakLib.Music.MusicPlayer.CurrentPlaylist [get]**

Gets the current playlist.

The current playlist.

#### **2.1.4.2 string FreakLib.Music.MusicPlayer.CurrentPlaylistName [get]**

Gets the name of the current playlist.

The name of the current playlist.

#### **2.1.4.3 Track FreakLib.Music.MusicPlayer.CurrentTrack [get]**

Gets the current track.

The current track.

#### **2.1.4.4 string FreakLib.Music.MusicPlayer.CurrentTrackName [get]**

Gets the name of the current track.

The name of the current track.

#### **2.1.4.5 int FreakLib.Music.MusicPlayer.HistoryLength [get], [set]**

Gets or sets the length of the history.

The length of the history.

#### **2.1.4.6 bool FreakLib.Music.MusicPlayer.IsPlaying [get]**

Gets a value indicating if we are currently playing.

true if playing; otherwise, false.

#### **2.1.4.7 List<string> FreakLib.Music.MusicPlayer.PlaylistNames [get]**

Gets the playlist names.

The playlist names.

#### **2.1.4.8 List<Playlist> FreakLib.Music.MusicPlayer.Playlists [get], [set]**

Gets the playlists.

The playlists.

#### **2.1.4.9 bool FreakLib.Music.MusicPlayer.PlayOnAwake [get], [set]**

Gets or sets a value indicating whether this [MusicPlayer](#) should play on awake.

true if play on awake; otherwise, false.

#### **2.1.4.10 float FreakLib.Music.MusicPlayer.Playtime [get], [set]**

Gets the current playtime, in seconds. Will be a value between 0 and the track length.

The playtime.

#### **2.1.4.11 float FreakLib.Music.MusicPlayer.PlaytimeNormalized** [get], [set]

Gets the current playtime normalized. Will be a value between 0 and 1.

The playtime normalized.

#### **2.1.4.12 float FreakLib.Music.MusicPlayer.Volume** [get], [set]

Gets or sets the volume.

The volume.

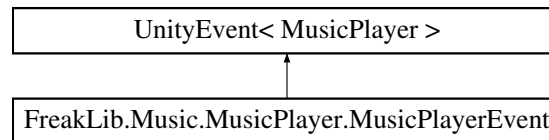
The documentation for this class was generated from the following file:

- Plugins/FreakshowStudio/MusicPlayer/Scripts/MusicPlayer.cs

## 2.2 FreakLib.Music.MusicPlayer.MusicPlayerEvent Class Reference

A custom UnityEvent, which allows passing a [MusicPlayer](#) as an argument.

Inheritance diagram for FreakLib.Music.MusicPlayer.MusicPlayerEvent:



### 2.2.1 Detailed Description

A custom UnityEvent, which allows passing a [MusicPlayer](#) as an argument.

The documentation for this class was generated from the following file:

- Plugins/FreakshowStudio/MusicPlayer/Scripts/MusicPlayer.cs

## 2.3 FreakLib.Music.Playlist Class Reference

This class implements a [Playlist](#) for the [MusicPlayer](#).

### Public Member Functions

- [Playlist](#) ()

*Initializes a new instance of the [FreakLib.Jukebox.Playlist](#) class.*

- [Playlist](#) (string aName)

*Initializes a new instance of the [FreakLib.Music.Playlist](#) class.*

- [Playlist](#) (string aName, List< [Track](#) > theTracks, bool shouldShuffle=false)

*Initializes a new instance of the [FreakLib.Music.Playlist](#) class.*

### Public Attributes

- string [name](#) = ""

*The name of the [Playlist](#).*

- List< [Track](#) > [tracks](#) = new List<[Track](#)>()

*The list of [FreakLib.Music.Tracks](#) in this [Playlist](#).*

- bool [shuffle](#) = false

*Should the [Playlist](#) be shuffled during play?*

- bool [foldout](#) = false

*If this [Playlist](#) should be expanded in the inspector.*

### Properties

- List< int > [EnabledTrackIndices](#) [get]

*Returns a list of indices to the tracks in this [Playlist](#) that are currently enabled.*



### 2.3.1 Detailed Description

This class implements a [Playlist](#) for the [MusicPlayer](#).

A [Playlist](#) is a group of [FreakLib.Music.Tracks](#) that the [MusicPlayer](#) can play in order, or randomly shuffled.

### 2.3.2 Constructor & Destructor Documentation

#### 2.3.2.1 [FreakLib.Music.Playlist.Playlist](#) ( )

Initializes a new instance of the [FreakLib.Jukebox.Playlist](#) class.

#### 2.3.2.2 [FreakLib.Music.Playlist.Playlist](#) ( string *aName* )

Initializes a new instance of the [FreakLib.Music.Playlist](#) class.

Parameters

<i>aName</i>	The <a href="#">Playlist</a> name.
--------------	------------------------------------

#### 2.3.2.3 [FreakLib.Music.Playlist.Playlist](#) ( string *aName*, List< [Track](#) > *theTracks*, bool *shouldShuffle* = *false* )

Initializes a new instance of the [FreakLib.Music.Playlist](#) class.

Parameters

<i>aName</i>	The <a href="#">Playlist</a> name.
<i>theTracks</i>	A list of tracks for this <a href="#">Playlist</a> .
<i>shouldShuffle</i>	If set to <code>true</code> the <a href="#">Playlist</a> will shuffle.

### 2.3.3 Member Data Documentation

#### 2.3.3.1 bool [FreakLib.Music.Playlist.foldout](#) = false

If this [Playlist](#) should be expanded in the inspector.

**2.3.3.2** `string FreakLib.Music.Playlist.name = ""`

The name of the [Playlist](#).

**2.3.3.3** `bool FreakLib.Music.Playlist.shuffle = false`

Should the [Playlist](#) be shuffled during play?

**2.3.3.4** `List<Track> FreakLib.Music.Playlist.tracks = new List<Track>()`

The list of [FreakLib.Music.Tracks](#) in this [Playlist](#).

## **2.3.4 Property Documentation**

**2.3.4.1** `List<int> FreakLib.Music.Playlist.EnabledTrackIndices` [get]

Returns a list of indices to the tracks in this [Playlist](#) that are currently enabled.

The enabled track indices.

The documentation for this class was generated from the following file:

- `Plugins/FreakshowStudio/MusicPlayer/Scripts/Playlist.cs`

## 2.4 FreakLib.Music.Track Class Reference

This class implements a single [Track](#) for the [MusicPlayer](#). Tracks are added to a [Playlist](#) that the [MusicPlayer](#) can then play.

### Public Member Functions

- [Track](#) ()

*Initializes a new instance of the [FreakLib.Music.Track](#) class.*

- [Track](#) (AudioClip aClip, string aName="")

*Initializes a new instance of the [FreakLib.Music.Track](#) class.*

- override string [ToString](#) ()

*Returns a [System.String](#) that represents the current [FreakLib.Music.Track](#).*

### Public Attributes

- string [name](#) = ""

*The [Track](#) name.*

- AudioClip [clip](#)

*The [AudioClip](#) for the [Track](#).*

- bool [enabled](#) = true

*If the [Track](#) is enabled. When disabled, the track will not be played.*

- bool [foldout](#) = false

*If this [Track](#) should be expanded in the inspector.*

### Properties

- float [Length](#) [get]

*The [Track](#) length, in seconds.*

- int [Plays](#) [get, set]

*The number of times this track has been played.*

### 2.4.1 Detailed Description

This class implements a single [Track](#) for the [MusicPlayer](#). Tracks are added to a [Playlist](#) that the [MusicPlayer](#) can then play.

### 2.4.2 Constructor & Destructor Documentation

#### 2.4.2.1 [FreakLib.Music.Track.Track](#) ( )

Initializes a new instance of the [FreakLib.Music.Track](#) class.

#### 2.4.2.2 [FreakLib.Music.Track.Track](#) ( [AudioClip](#) *aClip*, string *aName* = "" )

Initializes a new instance of the [FreakLib.Music.Track](#) class.

Parameters

<i>aClip</i>	The <a href="#">AudioClip</a> for this <a href="#">Track</a> .
<i>aName</i>	The name of this <a href="#">Track</a> .

### 2.4.3 Member Function Documentation

#### 2.4.3.1 override string [FreakLib.Music.Track.ToString](#) ( )

Returns a [System.String](#) that represents the current [FreakLib.Music.Track](#).

Returns

A System.String that represents the current [FreakLib.Music.Track](#).

## 2.4.4 Member Data Documentation

### 2.4.4.1 AudioClip FreakLib.Music.Track.clip

The AudioClip for the [Track](#).

### 2.4.4.2 bool FreakLib.Music.Track.enabled = true

If the [Track](#) is enabled. When disabled, the track will not be played.

### 2.4.4.3 bool FreakLib.Music.Track.foldout = false

If this [Track](#) should be expanded in the inspector.

### 2.4.4.4 string FreakLib.Music.Track.name = ""

The [Track](#) name.

## 2.4.5 Property Documentation

### 2.4.5.1 float FreakLib.Music.Track.Length [get]

The [Track](#) length, in seconds.

[Track](#) length.

### 2.4.5.2 int FreakLib.Music.Track.Plays [get], [set]

The number of times this track has been played.

This will be reset by the [MusicPlayer](#) whenever a new [Playlist](#) is started.

The number of plays.

The documentation for this class was generated from the following file:

- `Plugins/FreakshowStudio/MusicPlayer/Scripts/Track.cs`

- clip
  - FreakLib::Music::Track, [18](#)
- CurrentPlaylist
  - FreakLib::Music::MusicPlayer, [9](#)
- CurrentPlaylistName
  - FreakLib::Music::MusicPlayer, [9](#)
- CurrentTrack
  - FreakLib::Music::MusicPlayer, [9](#)
- CurrentTrackName
  - FreakLib::Music::MusicPlayer, [9](#)
- enabled
  - FreakLib::Music::Track, [18](#)
- EnabledTrackIndices
  - FreakLib::Music::Playlist, [15](#)
- foldout
  - FreakLib::Music::Playlist, [14](#)
  - FreakLib::Music::Track, [18](#)
- FreakLib, [1](#)
- FreakLib.Music, [2](#)
- FreakLib.Music.MusicPlayer, [3](#)
- FreakLib.Music.MusicPlayer.MusicPlayerEvent,  
[12](#)
- FreakLib.Music.Playlist, [13](#)
- FreakLib.Music.Track, [16](#)
- FreakLib::Music::MusicPlayer
  - CurrentPlaylist, [9](#)
  - CurrentPlaylistName, [9](#)
  - CurrentTrack, [9](#)
  - CurrentTrackName, [9](#)
  - GetPlaylist, [6](#)
  - HistoryLength, [9](#)
  - IsPlaying, [10](#)
  - Next, [6](#)
  - OnPause, [8](#)
  - OnPlay, [8](#)
  - OnPlaylistChange, [8](#)
  - OnStop, [8](#)
  - OnTrackChange, [8](#)
  - OnUnpause, [9](#)
  - Pause, [6](#)
  - PauseOrResume, [7](#)
  - Play, [7](#)
  - PlayOnAwake, [10](#)
  - PlaylistNames, [10](#)
  - Playlists, [10](#)
  - Playtime, [10](#)
  - PlaytimeNormalized, [10](#)
  - Previous, [8](#)
  - Stop, [8](#)
  - UnPause, [8](#)
  - Volume, [11](#)
- FreakLib::Music::Playlist
  - EnabledTrackIndices, [15](#)
  - foldout, [14](#)
  - name, [14](#)
  - Playlist, [14](#)
  - shuffle, [15](#)
  - tracks, [15](#)
- FreakLib::Music::Track
  - clip, [18](#)
  - enabled, [18](#)
  - foldout, [18](#)
  - Length, [18](#)
  - name, [18](#)
  - Plays, [18](#)
  - ToString, [17](#)
  - Track, [17](#)
- GetPlaylist
  - FreakLib::Music::MusicPlayer, [6](#)
- HistoryLength

- FreakLib::Music::MusicPlayer, [9](#)
- IsPlaying
  - FreakLib::Music::MusicPlayer, [10](#)
- Length
  - FreakLib::Music::Track, [18](#)
- name
  - FreakLib::Music::Playlist, [14](#)
  - FreakLib::Music::Track, [18](#)
- Next
  - FreakLib::Music::MusicPlayer, [6](#)
- OnPause
  - FreakLib::Music::MusicPlayer, [8](#)
- OnPlay
  - FreakLib::Music::MusicPlayer, [8](#)
- OnPlaylistChange
  - FreakLib::Music::MusicPlayer, [8](#)
- OnStop
  - FreakLib::Music::MusicPlayer, [8](#)
- OnTrackChange
  - FreakLib::Music::MusicPlayer, [8](#)
- OnUnpause
  - FreakLib::Music::MusicPlayer, [9](#)
- Pause
  - FreakLib::Music::MusicPlayer, [6](#)
- PauseOrResume
  - FreakLib::Music::MusicPlayer, [7](#)
- Play
  - FreakLib::Music::MusicPlayer, [7](#)
- PlayOnAwake
  - FreakLib::Music::MusicPlayer, [10](#)
- Playlist
  - FreakLib::Music::Playlist, [14](#)
- PlaylistNames
  - FreakLib::Music::MusicPlayer, [10](#)
- Playlists
  - FreakLib::Music::MusicPlayer, [10](#)
- Plays
  - FreakLib::Music::Track, [18](#)
- Playtime
  - FreakLib::Music::MusicPlayer, [10](#)
- PlaytimeNormalized
  - FreakLib::Music::MusicPlayer, [10](#)

- Previous
  - FreakLib::Music::MusicPlayer, [8](#)
- shuffle
  - FreakLib::Music::Playlist, [15](#)
- Stop
  - FreakLib::Music::MusicPlayer, [8](#)
- ToString
  - FreakLib::Music::Track, [17](#)
- Track
  - FreakLib::Music::Track, [17](#)
- tracks
  - FreakLib::Music::Playlist, [15](#)
- UnPause
  - FreakLib::Music::MusicPlayer, [8](#)
- Volume
  - FreakLib::Music::MusicPlayer, [11](#)