

# TriEngine2D

v0.0.8

Generated by Doxygen 1.8.3.1

Wed Feb 13 2013 01:42:55



# Contents

<b>1</b>	<b>Main Page</b>	<b>1</b>
<b>2</b>	<b>Namespace Index</b>	<b>3</b>
2.1	Namespace List . . . . .	3
<b>3</b>	<b>Hierarchical Index</b>	<b>5</b>
3.1	Class Hierarchy . . . . .	5
<b>4</b>	<b>Class Index</b>	<b>7</b>
4.1	Class List . . . . .	7
<b>5</b>	<b>File Index</b>	<b>9</b>
5.1	File List . . . . .	9
<b>6</b>	<b>Namespace Documentation</b>	<b>11</b>
6.1	Package TriDevs . . . . .	11
6.2	Package TriDevs.TriEngine2D . . . . .	11
6.3	Package TriDevs.TriEngine2D.Audio . . . . .	11
6.3.1	Enumeration Type Documentation . . . . .	12
6.3.1.1	AudioFormat . . . . .	12
6.4	Package TriDevs.TriEngine2D.Extensions . . . . .	12
6.5	Package TriDevs.TriEngine2D.Helpers . . . . .	12
6.6	Package TriDevs.TriEngine2D.Input . . . . .	13
6.7	Package TriDevs.TriEngine2D.Logging . . . . .	13
6.8	Package TriDevs.TriEngine2D.Native . . . . .	13
6.9	Package TriDevs.TriEngine2D.Serializing . . . . .	13
<b>7</b>	<b>Class Documentation</b>	<b>15</b>
7.1	TriDevs.TriEngine2D.Audio.AudioManager Class Reference . . . . .	15
7.1.1	Detailed Description . . . . .	16
7.1.2	Constructor & Destructor Documentation . . . . .	17
7.1.2.1	AudioManager . . . . .	17
7.1.3	Member Function Documentation . . . . .	17
7.1.3.1	Dispose . . . . .	17

7.1.3.2	GetSong	17
7.1.3.3	GetSound	17
7.1.3.4	HasSong	18
7.1.3.5	HasSound	18
7.1.3.6	LoadSong	18
7.1.3.7	LoadSound	19
7.1.3.8	StopAll	19
7.1.3.9	StopAllSongs	20
7.1.3.10	StopAllSounds	20
7.2	TriDevs.TriEngine2D.Extensions.EnumerationExtensions Class Reference	20
7.2.1	Detailed Description	20
7.2.2	Member Function Documentation	20
7.2.2.1	Has< T >	20
7.2.2.2	Include< T >	21
7.2.2.3	Missing< T >	21
7.2.2.4	Remove< T >	22
7.3	TriDevs.TriEngine2D.Native.Helpers Class Reference	22
7.3.1	Detailed Description	23
7.3.2	Member Function Documentation	23
7.3.2.1	GetErrorMessage	23
7.3.2.2	GetLastError	23
7.3.2.3	GetLastErrorInfo	23
7.3.2.4	GetLastErrorMessage	24
7.3.2.5	GetLastWin32Exception	24
7.3.2.6	GetWin32Exception	24
7.4	TriDevs.TriEngine2D.Audio.IAudioManager Interface Reference	25
7.4.1	Detailed Description	26
7.4.2	Member Function Documentation	26
7.4.2.1	GetSong	26
7.4.2.2	GetSound	26
7.4.2.3	HasSong	26
7.4.2.4	HasSound	27
7.4.2.5	LoadSong	27
7.4.2.6	LoadSound	27
7.4.2.7	StopAll	28
7.4.2.8	StopAllSongs	28
7.4.2.9	StopAllSounds	28
7.5	TriDevs.TriEngine2D.Input.IInputManager Interface Reference	28
7.5.1	Detailed Description	29
7.5.2	Member Function Documentation	29

7.5.2.1	ButtonDown	29
7.5.2.2	ButtonPressed	30
7.5.2.3	ButtonReleased	30
7.5.2.4	ButtonUp	30
7.5.2.5	KeyDown	30
7.5.2.6	KeyPressed	31
7.5.2.7	KeyReleased	31
7.5.2.8	KeyUp	31
7.5.2.9	Update	31
7.5.2.10	WheelChange	32
7.5.2.11	WheelChanged	32
7.5.2.12	WheelDown	32
7.5.2.13	WheelUp	32
7.5.3	Property Documentation	32
7.5.3.1	MousePosition	32
7.5.3.2	MouseWheelValue	32
7.5.3.3	MouseX	32
7.5.3.4	MouseY	33
7.5.3.5	this[Key key]	33
7.5.3.6	this[MouseButton button]	33
7.6	TriDevs.TriEngine2D.Input.InputManager Class Reference	33
7.6.1	Detailed Description	35
7.6.2	Constructor & Destructor Documentation	35
7.6.2.1	InputManager	35
7.6.3	Member Function Documentation	35
7.6.3.1	ButtonDown	35
7.6.3.2	ButtonPressed	35
7.6.3.3	ButtonReleased	36
7.6.3.4	ButtonUp	36
7.6.3.5	KeyDown	36
7.6.3.6	KeyPressed	37
7.6.3.7	KeyReleased	37
7.6.3.8	KeyUp	37
7.6.3.9	Update	38
7.6.3.10	WheelChange	38
7.6.3.11	WheelChanged	38
7.6.3.12	WheelDown	38
7.6.3.13	WheelUp	39
7.6.4	Property Documentation	39
7.6.4.1	MousePosition	39

7.6.4.2	<a href="#">MouseWheelValue</a>	39
7.6.4.3	<a href="#">MouseX</a>	39
7.6.4.4	<a href="#">MouseY</a>	39
7.6.4.5	<a href="#">this[Key key]</a>	39
7.6.4.6	<a href="#">this[MouseButton button]</a>	39
7.7	<a href="#">TriDevs.TriEngine2D.Helpers.IO Class Reference</a>	39
7.7.1	<a href="#">Detailed Description</a>	39
7.7.2	<a href="#">Member Function Documentation</a>	39
7.7.2.1	<a href="#">GetAbsolutePath</a>	39
7.8	<a href="#">TriDevs.TriEngine2D.Audio.ISong Interface Reference</a>	40
7.8.1	<a href="#">Detailed Description</a>	41
7.8.2	<a href="#">Member Function Documentation</a>	41
7.8.2.1	<a href="#">Pause</a>	41
7.8.2.2	<a href="#">Play</a>	41
7.8.2.3	<a href="#">Resume</a>	41
7.8.2.4	<a href="#">Stop</a>	41
7.8.3	<a href="#">Property Documentation</a>	42
7.8.3.1	<a href="#">File</a>	42
7.8.3.2	<a href="#">IsLooped</a>	42
7.8.3.3	<a href="#">Name</a>	42
7.8.3.4	<a href="#">Volume</a>	42
7.9	<a href="#">TriDevs.TriEngine2D.Audio.ISound Interface Reference</a>	42
7.9.1	<a href="#">Detailed Description</a>	43
7.9.2	<a href="#">Member Function Documentation</a>	43
7.9.2.1	<a href="#">Play</a>	43
7.9.2.2	<a href="#">Stop</a>	43
7.9.3	<a href="#">Property Documentation</a>	44
7.9.3.1	<a href="#">File</a>	44
7.9.3.2	<a href="#">Name</a>	44
7.10	<a href="#">TriDevs.TriEngine2D.Logging.LogManager Class Reference</a>	44
7.10.1	<a href="#">Detailed Description</a>	44
7.10.2	<a href="#">Member Function Documentation</a>	44
7.10.2.1	<a href="#">ClearOldLogs</a>	44
7.10.2.2	<a href="#">DestroyConsole</a>	45
7.10.2.3	<a href="#">GetLogger</a>	45
7.10.2.4	<a href="#">LoadConfig</a>	46
7.10.2.5	<a href="#">SetupConsole</a>	46
7.11	<a href="#">TriDevs.TriEngine2D.Helpers.Math Class Reference</a>	47
7.11.1	<a href="#">Detailed Description</a>	47
7.11.2	<a href="#">Member Function Documentation</a>	48

7.11.2.1	Clamp	48
7.11.2.2	Clamp	48
7.11.2.3	Clamp	48
7.11.2.4	Clamp	49
7.11.2.5	Clamp	49
7.11.2.6	Clamp	49
7.11.2.7	Clamp	49
7.11.2.8	Clamp	50
7.11.2.9	Clamp	50
7.12	TriDevs.TriEngine2D.Audio.NullAudioManager Class Reference	50
7.12.1	Detailed Description	52
7.12.2	Member Function Documentation	52
7.12.2.1	Dispose	52
7.12.2.2	GetSong	52
7.12.2.3	GetSound	52
7.12.2.4	HasSong	53
7.12.2.5	HasSound	53
7.12.2.6	LoadSong	53
7.12.2.7	LoadSound	54
7.12.2.8	StopAll	54
7.12.2.9	StopAllSongs	54
7.12.2.10	StopAllSounds	54
7.13	TriDevs.TriEngine2D.Input.NullInputManager Class Reference	55
7.13.1	Detailed Description	56
7.13.2	Member Function Documentation	56
7.13.2.1	ButtonDown	56
7.13.2.2	ButtonPressed	57
7.13.2.3	ButtonReleased	57
7.13.2.4	ButtonUp	57
7.13.2.5	KeyDown	58
7.13.2.6	KeyPressed	58
7.13.2.7	KeyReleased	58
7.13.2.8	KeyUp	59
7.13.2.9	Update	59
7.13.2.10	WheelChange	59
7.13.2.11	WheelChanged	59
7.13.2.12	WheelDown	60
7.13.2.13	WheelUp	60
7.13.3	Property Documentation	60
7.13.3.1	MousePosition	60

7.13.3.2	<a href="#">MouseWheelValue</a>	60
7.13.3.3	<a href="#">MouseX</a>	60
7.13.3.4	<a href="#">MouseY</a>	60
7.13.3.5	<a href="#">this[Key key]</a>	60
7.13.3.6	<a href="#">this[MouseButton button]</a>	60
7.14	<a href="#">TriDevs.TriEngine2D.Audio.NullSong Class Reference</a>	60
7.14.1	<a href="#">Detailed Description</a>	62
7.14.2	<a href="#">Member Function Documentation</a>	62
7.14.2.1	<a href="#">Dispose</a>	62
7.14.2.2	<a href="#">Pause</a>	62
7.14.2.3	<a href="#">Play</a>	62
7.14.2.4	<a href="#">Resume</a>	62
7.14.2.5	<a href="#">Stop</a>	62
7.14.3	<a href="#">Property Documentation</a>	63
7.14.3.1	<a href="#">File</a>	63
7.14.3.2	<a href="#">IsLooped</a>	63
7.14.3.3	<a href="#">Name</a>	63
7.14.3.4	<a href="#">Volume</a>	63
7.15	<a href="#">TriDevs.TriEngine2D.Audio.NullSound Class Reference</a>	63
7.15.1	<a href="#">Detailed Description</a>	64
7.15.2	<a href="#">Member Function Documentation</a>	64
7.15.2.1	<a href="#">Dispose</a>	64
7.15.2.2	<a href="#">Play</a>	64
7.15.2.3	<a href="#">Stop</a>	65
7.15.3	<a href="#">Property Documentation</a>	65
7.15.3.1	<a href="#">File</a>	65
7.15.3.2	<a href="#">Name</a>	65
7.16	<a href="#">TriDevs.TriEngine2D.Point Struct Reference</a>	65
7.16.1	<a href="#">Detailed Description</a>	65
7.16.2	<a href="#">Constructor &amp; Destructor Documentation</a>	65
7.16.2.1	<a href="#">Point</a>	65
7.16.3	<a href="#">Member Data Documentation</a>	66
7.16.3.1	<a href="#">X</a>	66
7.16.3.2	<a href="#">Y</a>	66
7.17	<a href="#">TriDevs.TriEngine2D.Serializing.Serializer Class Reference</a>	66
7.17.1	<a href="#">Detailed Description</a>	66
7.17.2	<a href="#">Member Function Documentation</a>	66
7.17.2.1	<a href="#">Deserialize&lt; T &gt;</a>	66
7.17.2.2	<a href="#">Serialize&lt; T &gt;</a>	67
7.17.2.3	<a href="#">Serialize&lt; T &gt;</a>	67



7.18 TriDevs.TriEngine2D.Services Class Reference . . . . .	68
7.18.1 Detailed Description . . . . .	68
7.18.2 Member Function Documentation . . . . .	68
7.18.2.1 Provide . . . . .	68
7.18.2.2 Provide . . . . .	69
7.18.2.3 Provide . . . . .	69
7.18.3 Property Documentation . . . . .	69
7.18.3.1 Audio . . . . .	69
7.18.3.2 Input . . . . .	69
7.19 TriDevs.TriEngine2D.Audio.Song Class Reference . . . . .	69
7.19.1 Detailed Description . . . . .	71
7.19.2 Member Function Documentation . . . . .	71
7.19.2.1 Dispose . . . . .	71
7.19.2.2 Pause . . . . .	71
7.19.2.3 Play . . . . .	71
7.19.2.4 Resume . . . . .	71
7.19.2.5 Stop . . . . .	72
7.19.3 Property Documentation . . . . .	72
7.19.3.1 File . . . . .	72
7.19.3.2 IsLooped . . . . .	72
7.19.3.3 Name . . . . .	72
7.19.3.4 Volume . . . . .	72
7.20 TriDevs.TriEngine2D.Audio.Sound Class Reference . . . . .	72
7.20.1 Detailed Description . . . . .	73
7.20.2 Member Function Documentation . . . . .	73
7.20.2.1 Dispose . . . . .	73
7.20.2.2 Play . . . . .	73
7.20.2.3 Stop . . . . .	74
7.20.3 Property Documentation . . . . .	74
7.20.3.1 File . . . . .	74
7.20.3.2 Name . . . . .	74
7.21 TriDevs.TriEngine2D.Extensions.StringExtensions Class Reference . . . . .	74
7.21.1 Detailed Description . . . . .	74
7.21.2 Member Function Documentation . . . . .	74
7.21.2.1 Replace . . . . .	74
7.21.2.2 Replace . . . . .	75
7.21.2.3 ReplaceFirst . . . . .	75
7.22 TriDevs.TriEngine2D.Helpers.Threading Class Reference . . . . .	76
7.22.1 Detailed Description . . . . .	76
7.22.2 Member Function Documentation . . . . .	76

7.22.2.1	<a href="#">SetCurrentThreadName</a>	76
7.23	<a href="#">TriDevs.TriEngine2D.Version Class Reference</a>	76
7.23.1	<a href="#">Detailed Description</a>	77
7.23.2	<a href="#">Member Data Documentation</a>	77
7.23.2.1	<a href="#">Major</a>	77
7.23.2.2	<a href="#">Minor</a>	77
7.23.2.3	<a href="#">Patch</a>	77
7.23.2.4	<a href="#">Suffix</a>	77
7.23.2.5	<a href="#">VersionStringFormat</a>	77
7.23.2.6	<a href="#">VersionStringFormatWithSuffix</a>	77
7.23.3	<a href="#">Property Documentation</a>	77
7.23.3.1	<a href="#">VersionString</a>	77
7.24	<a href="#">TriDevs.TriEngine2D.Native.WinAPI Class Reference</a>	78
7.24.1	<a href="#">Detailed Description</a>	78
7.24.2	<a href="#">Member Function Documentation</a>	78
7.24.2.1	<a href="#">AllocConsole</a>	78
7.24.2.2	<a href="#">FreeConsole</a>	78
7.24.2.3	<a href="#">GetStdHandle</a>	79
7.24.3	<a href="#">Member Data Documentation</a>	79
7.24.3.1	<a href="#">CODE_PAGE</a>	79
7.24.3.2	<a href="#">STD_OUTPUT_HANDLE</a>	79
<b>8</b>	<b><a href="#">File Documentation</a></b>	<b>81</b>
8.1	<a href="#">README.md File Reference</a>	81
8.2	<a href="#">TriDevs.TriEngine2D/Audio/AudioFormat.cs File Reference</a>	81
8.3	<a href="#">TriDevs.TriEngine2D/Audio/AudioManager.cs File Reference</a>	81
8.4	<a href="#">TriDevs.TriEngine2D/Audio/IAudioManager.cs File Reference</a>	81
8.5	<a href="#">TriDevs.TriEngine2D/Audio/ISong.cs File Reference</a>	82
8.6	<a href="#">TriDevs.TriEngine2D/Audio/ISound.cs File Reference</a>	82
8.7	<a href="#">TriDevs.TriEngine2D/Audio/NullAudioManager.cs File Reference</a>	82
8.8	<a href="#">TriDevs.TriEngine2D/Audio/NullSong.cs File Reference</a>	82
8.9	<a href="#">TriDevs.TriEngine2D/Audio/NullSound.cs File Reference</a>	83
8.10	<a href="#">TriDevs.TriEngine2D/Audio/Song.cs File Reference</a>	83
8.11	<a href="#">TriDevs.TriEngine2D/Audio/Sound.cs File Reference</a>	83
8.12	<a href="#">TriDevs.TriEngine2D/Extensions/EnumerationExtensions.cs File Reference</a>	83
8.13	<a href="#">TriDevs.TriEngine2D/Extensions/StringExtensions.cs File Reference</a>	84
8.14	<a href="#">TriDevs.TriEngine2D/Helpers/IO.cs File Reference</a>	84
8.15	<a href="#">TriDevs.TriEngine2D/Helpers/Math.cs File Reference</a>	84
8.16	<a href="#">TriDevs.TriEngine2D/Helpers/Threading.cs File Reference</a>	84
8.17	<a href="#">TriDevs.TriEngine2D/Input/IInputManager.cs File Reference</a>	85

8.18	<a href="#">TriDevs.TriEngine2D/Input/InputManager.cs File Reference</a>	85
8.19	<a href="#">TriDevs.TriEngine2D/Input/NullInputManager.cs File Reference</a>	85
8.20	<a href="#">TriDevs.TriEngine2D/Logging/LogManager.cs File Reference</a>	85
8.21	<a href="#">TriDevs.TriEngine2D/Native/Helpers.cs File Reference</a>	86
8.22	<a href="#">TriDevs.TriEngine2D/Native/WinAPI.cs File Reference</a>	86
8.23	<a href="#">TriDevs.TriEngine2D/Point.cs File Reference</a>	86
8.24	<a href="#">TriDevs.TriEngine2D/Properties/AssemblyInfo.cs File Reference</a>	86
8.25	<a href="#">TriDevs.TriEngine2D/Serializing/Serializer.cs File Reference</a>	86
8.26	<a href="#">TriDevs.TriEngine2D/Services.cs File Reference</a>	87
8.27	<a href="#">TriDevs.TriEngine2D/Version.cs File Reference</a>	87

## [Index](#)

87



# Chapter 1

## Main Page

2D general-purpose engine in C#/OpenGL

### IRC

[TriDevs](#) has an IRC channel, feel free to hop in if you have a question about anything: **Server:** irc.kottnet.net **Port:** 6667, 6697 (SSL) **Channel:** [TriDevs](#)

The channel topic contains further info.

### License

Copyright © 2013 by [Adam Hellberg](#), [Sijmen Schoon](#) and [Preston Shumway](#).

TriEngine2D is licensed under the [MIT License](#), more info can be found in the **LICENSE** file.

### Contributing

You are free to fork this project and make your own changes, as long as you follow the MIT License.

If you want to make a pull request, please do so to the [main project](#) and not any of the "official" forks.

For your pull request to be accepted, please follow our coding style:

- Indent with 4 spaces, not tabs.
- Curly braces placed on next line.
- All **public** methods, accessors and members must be properly documented.
- Use sensible variable names that describe what they are for.
- Method declarations written as:

```
“c# public void Hello(string world) “
```

- If your method accepts many parameters, it can be useful to put parameters on separate lines, as per this style:

```
“c# public void Hello(string world, bool print) “
```

- Please write tests for your code (not strictly required, but it's a plus)

By looking through the current source code, you should be able to get a good understanding of the formatting we use.

If you're using Visual Studio, you can change the indent behaviour by going to: **Tools -> Options -> Text Editor -> C# -> Tabs** and make sure "Insert spaces" is checked.

If you write tests for your code, please place these tests in their own project: "<strong>Namespace</strong> >.Tests", create said project if it does not exist (of type Class Library).

We use NUnit as test framework, feel free to use something else if you want to, but make sure you document what framework you are using and that it is freely available for anyone to obtain.

## Dependencies

TriEngine2D depends on [log4net](#), which is included in the **libs/log4net** folder.

TriEngine2D depends on [OpenTK](#), this is not included and you will have to build/install it yourself. OpenTK depends on OpenGL drivers being installed, they are usually in your normal video card drivers.

TriEngine2D depends on [Json.NET](#), this is not included, but is specified in the NuGet package config. If you [properly configure your NuGet settings](#), NuGet will automatically download Json.NET when building any projects that depend on it.

TriEngine2D depends on [NVorbis](#), which is included in the **libs/NVorbis** folder. [NVorbis is licensed under Ms-PL](#).

If you want to run the tests you will need to have [NUnit](#) installed.

## Chapter 2

# Namespace Index

### 2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">TriDevs</a>	11
<a href="#">TriDevs.TriEngine2D</a>	11
<a href="#">TriDevs.TriEngine2D.Audio</a>	11
<a href="#">TriDevs.TriEngine2D.Extensions</a>	12
<a href="#">TriDevs.TriEngine2D.Helpers</a>	12
<a href="#">TriDevs.TriEngine2D.Input</a>	13
<a href="#">TriDevs.TriEngine2D.Logging</a>	13
<a href="#">TriDevs.TriEngine2D.Native</a>	13
<a href="#">TriDevs.TriEngine2D.Serializing</a>	13





## Chapter 3

# Hierarchical Index

### 3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

TriDevs.TriEngine2D.Extensions.EnumerationExtensions . . . . .	20
TriDevs.TriEngine2D.Native.Helpers . . . . .	22
IDisposable	
TriDevs.TriEngine2D.Audio.IAudioManager . . . . .	25
TriDevs.TriEngine2D.Audio.AudioManager . . . . .	15
TriDevs.TriEngine2D.Audio.NullAudioManager . . . . .	50
TriDevs.TriEngine2D.Audio.ISong . . . . .	40
TriDevs.TriEngine2D.Audio.NullSong . . . . .	60
TriDevs.TriEngine2D.Audio.Song . . . . .	69
TriDevs.TriEngine2D.Audio.ISound . . . . .	42
TriDevs.TriEngine2D.Audio.NullSound . . . . .	63
TriDevs.TriEngine2D.Audio.Sound . . . . .	72
TriDevs.TriEngine2D.Input.IInputManager . . . . .	28
TriDevs.TriEngine2D.Input.InputManager . . . . .	33
TriDevs.TriEngine2D.Input.NullInputManager . . . . .	55
TriDevs.TriEngine2D.Helpers.IO . . . . .	39
TriDevs.TriEngine2D.Logging.LogManager . . . . .	44
TriDevs.TriEngine2D.Helpers.Math . . . . .	47
TriDevs.TriEngine2D.Point . . . . .	65
TriDevs.TriEngine2D.Serializing.Serializer . . . . .	66
TriDevs.TriEngine2D.Services . . . . .	68
TriDevs.TriEngine2D.Extensions.StringExtensions . . . . .	74
TriDevs.TriEngine2D.Helpers.Threading . . . . .	76
TriDevs.TriEngine2D.Version . . . . .	76
TriDevs.TriEngine2D.Native.WinAPI . . . . .	78



## Chapter 4

# Class Index

### 4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">TriDevs.TriEngine2D.Audio.AudioManager</a>	15
Class to manage engine audio. . . . .	
<a href="#">TriDevs.TriEngine2D.Extensions.EnumerationExtensions</a>	20
Extensions for System.Enum. . . . .	
<a href="#">TriDevs.TriEngine2D.Native.Helpers</a>	22
Helper class with various methods to help native coding and debugging. . . . .	
<a href="#">TriDevs.TriEngine2D.Audio.IAudioManager</a>	25
Provides various methods to manipulate audio. . . . .	
<a href="#">TriDevs.TriEngine2D.Input.IInputManager</a>	28
Provides various methods to query input devices like the keyboard. . . . .	
<a href="#">TriDevs.TriEngine2D.Input.InputManager</a>	33
Input manager interfacing with input methods provided by a GameWindow. . . . .	
<a href="#">TriDevs.TriEngine2D.Helpers.IO</a>	39
Provides various helper functions for doing IO operations. . . . .	
<a href="#">TriDevs.TriEngine2D.Audio.ISong</a>	40
A song that will be streamed in the audio player. . . . .	
<a href="#">TriDevs.TriEngine2D.Audio.ISound</a>	42
A sound file for use with the audio manager. . . . .	
<a href="#">TriDevs.TriEngine2D.Logging.LogManager</a>	44
Class to manage logging. ILog interfaces should be obtained from this class' methods, as opposed to calling default log4net methods. . . . .	
<a href="#">TriDevs.TriEngine2D.Helpers.Math</a>	47
Various helper methods for working with math. . . . .	
<a href="#">TriDevs.TriEngine2D.Audio.NullAudioManager</a>	50
Used as a fallback AudioManager object when the service locator fails to find one. . . . .	
<a href="#">TriDevs.TriEngine2D.Input.NullInputManager</a>	55
Used as a fallback InputManager object when the service locator fails to find one. . . . .	
<a href="#">TriDevs.TriEngine2D.Audio.NullSong</a>	60
Fallback song class used in NullAudioManager. . . . .	
<a href="#">TriDevs.TriEngine2D.Audio.NullSound</a>	63
Fallback sound class used in NullAudioManager. . . . .	
<a href="#">TriDevs.TriEngine2D.Point</a>	65
A struct representing an X/Y coordinate. . . . .	
<a href="#">TriDevs.TriEngine2D.Serializing.Serializer</a>	66
Provides serialization methods. . . . .	
<a href="#">TriDevs.TriEngine2D.Services</a>	68
Provides different game-related service interfaces. . . . .	

<a href="#">TriDevs.TriEngine2D.Audio.Song</a>	
<a href="#">Song</a> class that can be used with <a href="#">AudioManager</a> .	69
<a href="#">TriDevs.TriEngine2D.Audio.Sound</a>	
<a href="#">Sound</a> class that can be used with the <a href="#">AudioManager</a> .	72
<a href="#">TriDevs.TriEngine2D.Extensions.StringExtensions</a>	
<a href="#">Extensions</a> for <a href="#">System.String</a>	74
<a href="#">TriDevs.TriEngine2D.Helpers.Threading</a>	
Provides various helper functions for doing threading operations.	76
<a href="#">TriDevs.TriEngine2D.Version</a>	
<a href="#">Version</a> class specifying the version of this project.	76
<a href="#">TriDevs.TriEngine2D.Native.WinAPI</a>	
Holds various <a href="#">WinAPI</a> stuff.	78

## Chapter 5

# File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

TriDevs.TriEngine2D/ <a href="#">Point.cs</a> . . . . .	86
TriDevs.TriEngine2D/ <a href="#">Services.cs</a> . . . . .	87
TriDevs.TriEngine2D/ <a href="#">Version.cs</a> . . . . .	87
TriDevs.TriEngine2D/Audio/ <a href="#">AudioFormat.cs</a> . . . . .	81
TriDevs.TriEngine2D/Audio/ <a href="#">AudioManager.cs</a> . . . . .	81
TriDevs.TriEngine2D/Audio/ <a href="#">IAudioManager.cs</a> . . . . .	81
TriDevs.TriEngine2D/Audio/ <a href="#">ISong.cs</a> . . . . .	82
TriDevs.TriEngine2D/Audio/ <a href="#">ISound.cs</a> . . . . .	82
TriDevs.TriEngine2D/Audio/ <a href="#">NullAudioManager.cs</a> . . . . .	82
TriDevs.TriEngine2D/Audio/ <a href="#">NullSong.cs</a> . . . . .	82
TriDevs.TriEngine2D/Audio/ <a href="#">NullSound.cs</a> . . . . .	83
TriDevs.TriEngine2D/Audio/ <a href="#">Song.cs</a> . . . . .	83
TriDevs.TriEngine2D/Audio/ <a href="#">Sound.cs</a> . . . . .	83
TriDevs.TriEngine2D/Extensions/ <a href="#">EnumerationExtensions.cs</a> . . . . .	83
TriDevs.TriEngine2D/Extensions/ <a href="#">StringExtensions.cs</a> . . . . .	84
TriDevs.TriEngine2D/Helpers/ <a href="#">IO.cs</a> . . . . .	84
TriDevs.TriEngine2D/Helpers/ <a href="#">Math.cs</a> . . . . .	84
TriDevs.TriEngine2D/Helpers/ <a href="#">Threading.cs</a> . . . . .	84
TriDevs.TriEngine2D/Input/ <a href="#">IInputManager.cs</a> . . . . .	85
TriDevs.TriEngine2D/Input/ <a href="#">InputManager.cs</a> . . . . .	85
TriDevs.TriEngine2D/Input/ <a href="#">NullInputManager.cs</a> . . . . .	85
TriDevs.TriEngine2D/Logging/ <a href="#">LogManager.cs</a> . . . . .	85
TriDevs.TriEngine2D/Native/ <a href="#">Helpers.cs</a> . . . . .	86
TriDevs.TriEngine2D/Native/ <a href="#">WinAPI.cs</a> . . . . .	86
TriDevs.TriEngine2D/Properties/ <a href="#">AssemblyInfo.cs</a> . . . . .	86
TriDevs.TriEngine2D/Serializing/ <a href="#">Serializer.cs</a> . . . . .	86



## Chapter 6

# Namespace Documentation

### 6.1 Package TriDevs

#### Namespaces

- package [TriEngine2D](#)

### 6.2 Package TriDevs.TriEngine2D

#### Namespaces

- package [Audio](#)
- package [Extensions](#)
- package [Helpers](#)
- package [Input](#)
- package [Logging](#)
- package [Native](#)
- package [Serializing](#)

#### Classes

- struct [Point](#)  
*A struct representing an X/Y coordinate.*
- class [Services](#)  
*Provides different game-related service interfaces.*
- class [Version](#)  
*[Version](#) class specifying the version of this project.*

### 6.3 Package TriDevs.TriEngine2D.Audio

#### Classes

- class [AudioManager](#)  
*Class to manage engine audio.*
- interface [IAudioManager](#)  
*Provides various methods to manipulate audio.*

- interface [ISong](#)  
*A song that will be streamed in the audio player.*
- interface [ISound](#)  
*A sound file for use with the audio manager.*
- class [NullAudioManager](#)  
*Used as a fallback [AudioManager](#) object when the service locator fails to find one.*
- class [NullSong](#)  
*Fallback song class used in [NullAudioManager](#).*
- class [NullSound](#)  
*Fallback sound class used in [NullAudioManager](#).*
- class [Song](#)  
*[Song](#) class that can be used with [AudioManager](#).*
- class [Sound](#)  
*[Sound](#) class that can be used with the [AudioManager](#).*

## Enumerations

- enum [AudioFormat](#) { [Ogg](#), [Wav](#) }  
*Different audio formats usable with audio managers.*

### 6.3.1 Enumeration Type Documentation

#### 6.3.1.1 enum TriDevs.TriEngine2D.Audio.AudioFormat

Different audio formats usable with audio managers.

##### Enumerator

**Ogg** The Ogg Vorbis format.

**Wav** The WAVE format

```

30      {
34          Ogg,
35
39          Wav
40      }
```

## 6.4 Package TriDevs.TriEngine2D.Extensions

### Classes

- class [EnumerationExtensions](#)  
*[Extensions](#) for System.Enum.*
- class [StringExtensions](#)  
*[Extensions](#) for System.String*

## 6.5 Package TriDevs.TriEngine2D.Helpers

### Classes

- class [IO](#)



- Provides various helper functions for doing [IO](#) operations.*
- class [Math](#)  
*Various helper methods for working with math.*
- class [Threading](#)  
*Provides various helper functions for doing threading operations.*

## 6.6 Package TriDevs.TriEngine2D.Input

### Classes

- interface [IInputManager](#)  
*Provides various methods to query input devices like the keyboard.*
- class [InputManager](#)  
*[Input](#) manager interfacing with input methods provided by a [GameWindow](#).*
- class [NullInputManager](#)  
*Used as a fallback [InputManager](#) object when the service locator fails to find one.*

## 6.7 Package TriDevs.TriEngine2D.Logging

### Classes

- class [LogManager](#)  
*Class to manage logging. [ILog](#) interfaces should be obtained from this class' methods, as opposed to calling default [log4net](#) methods.*

## 6.8 Package TriDevs.TriEngine2D.Native

### Classes

- class [Helpers](#)  
*Helper class with various methods to help native coding and debugging.*
- class [WinAPI](#)  
*Holds various [WinAPI](#) stuff.*

## 6.9 Package TriDevs.TriEngine2D.Serializing

### Classes

- class [Serializer](#)  
*Provides serialization methods.*



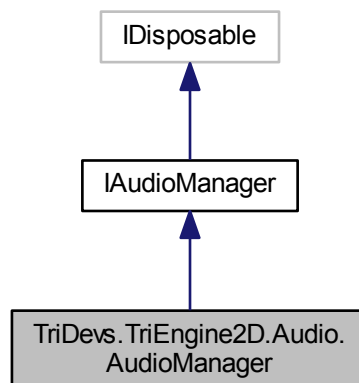
## Chapter 7

# Class Documentation

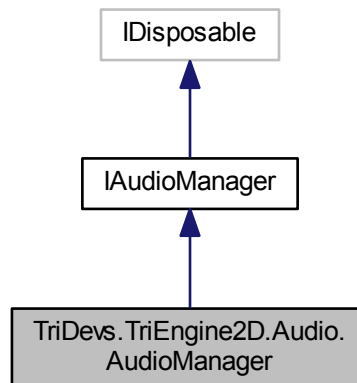
### 7.1 TriDevs.TriEngine2D.Audio.AudioManager Class Reference

Class to manage engine audio.

Inheritance diagram for TriDevs.TriEngine2D.Audio.AudioManager:



Collaboration diagram for TriDevs.TriEngine2D.Audio.AudioManager:



## Public Member Functions

- [AudioManager](#) ()  
*Creates a new instance of [AudioManager](#).*
- void [Dispose](#) ()
- void [StopAll](#) ()  
*Immediately stops all playback of sounds and songs.*
- [ISound LoadSound](#) (string name, string file, [AudioFormat](#) format=AudioFormat.Wav)  
*Loads a sound into the audio manager.*
- bool [HasSound](#) (string name)  
*Returns a value indicating whether a sound with the specified name has been loaded into the audio manager.*
- [ISound GetSound](#) (string name)  
*Gets the sound with the specified name.*
- void [StopAllSounds](#) ()  
*Immediately stops playback of all sounds.*
- [ISong LoadSong](#) (string name, string file, [AudioFormat](#) format=AudioFormat.Ogg)  
*Load a song into the audio manager.*
- bool [HasSong](#) (string name)  
*Returns a value indicating whether a song with the specified name has been loaded into the audio manager.*
- [ISong GetSong](#) (string name)  
*Gets the song with the specified name.*
- void [StopAllSongs](#) ()  
*Immediately stops playback of all songs.*

### 7.1.1 Detailed Description

Class to manage engine audio.

## 7.1.2 Constructor & Destructor Documentation

### 7.1.2.1 TriDevs.TriEngine2D.Audio.AudioManager.AudioManager ( )

Creates a new instance of [AudioManager](#).

```

47     {
48         _context = new AudioContext();
49         _oggStreamer = new OggStreamer();
50         _sounds = new List<ISound>();
51         _songs = new List<ISong>();
52     }

```

## 7.1.3 Member Function Documentation

### 7.1.3.1 void TriDevs.TriEngine2D.Audio.AudioManager.Dispose ( )

```

55     {
56         foreach (var sound in _sounds.Where(sound => sound != null))
57             sound.Dispose();
58
59         foreach (var song in _songs.Where(song => song != null))
60             song.Dispose();
61
62         if (_oggStreamer != null)
63         {
64             _oggStreamer.Dispose();
65             _oggStreamer = null;
66         }
67
68         if (_context != null)
69         {
70             _context.Dispose();
71             _context = null;
72         }
73     }

```

### 7.1.3.2 ISong TriDevs.TriEngine2D.Audio.AudioManager.GetSong ( string name )

Gets the song with the specified name.

#### Parameters

<i>name</i>	Name of the song to get.
-------------	--------------------------

#### Returns

An object implementing the [ISong](#) interface, or null if no song matched the requested name.

Implements [TriDevs.TriEngine2D.Audio.IAudioManager](#).

```

132     {
133         return _songs.FirstOrDefault(s => s.Name == name);
134     }

```

### 7.1.3.3 ISound TriDevs.TriEngine2D.Audio.AudioManager.GetSound ( string name )

Gets the sound with the specified name.

#### Parameters

<i>name</i>	Name of the sound to get.
-------------	---------------------------

**Returns**

An object implementing the [ISound](#) interface, or null if no sound matched the requested name.

Implements [TriDevs.TriEngine2D.Audio.IAudioManager](#).

```

102     {
103         return _sounds.FirstOrDefault(s => s.Name == name);
104     }

```

**7.1.3.4 bool TriDevs.TriEngine2D.Audio.AudioManager.HasSong ( string name )**

Returns a value indicating whether a song with the specified name has been loaded into the audio manager.

**Parameters**

<i>name</i>	Name to check for.
-------------	--------------------

**Returns**

True if the song has been loaded, false otherwise.

Implements [TriDevs.TriEngine2D.Audio.IAudioManager](#).

```

127     {
128         return _songs.Any(s => s.Name == name);
129     }

```

**7.1.3.5 bool TriDevs.TriEngine2D.Audio.AudioManager.HasSound ( string name )**

Returns a value indicating whether a sound with the specified name has been loaded into the audio manager.

**Parameters**

<i>name</i>	Name to check for.
-------------	--------------------

**Returns**

True if the sound has been loaded, false otherwise.

Implements [TriDevs.TriEngine2D.Audio.IAudioManager](#).

```

97     {
98         return _sounds.Any(s => s.Name == name);
99     }

```

**7.1.3.6 ISong TriDevs.TriEngine2D.Audio.AudioManager.LoadSong ( string name, string file, AudioFormat format = AudioFormat.Ogg )**

Load a song into the audio manager.

If a song with the same name has already been loaded, it will be returned instead.

**Parameters**

<i>name</i>	Name to assign to the new song object.
<i>file</i>	Path to the file that will be used.
<i>format</i>	Format of the file.

**Returns**

The resulting object implementing the [ISong](#) interface.

Implements [TriDevs.TriEngine2D.Audio.IAudioManager](#).

```

112     {
113         var existing = _songs.FirstOrDefault(s => s.Name == name);
114         if (existing != null)
115             return existing;
116
117         var loaded = _songs.FirstOrDefault(s => s.File == file) != null;
118         if (loaded)
119             throw new Exception("The song file \"" + file + "\" has already been loaded under a
different name.");
120
121         var song = new Song(name, file, format);
122         _songs.Add(song);
123         return song;
124     }

```

### 7.1.3.7 ISound TriDevs.TriEngine2D.Audio.AudioManager.LoadSound ( string name, string file, AudioFormat format = AudioFormat.Wav )

Loads a sound into the audio manager.

If a sound with the same name has already been loaded, it will be returned instead.

**Parameters**

<i>name</i>	Name to assign to the new sound object.
<i>file</i>	Path to the file that will be used.
<i>format</i>	The format of the file to load.

**Returns**

The resulting object implementing the [ISound](#) interface.

Implements [TriDevs.TriEngine2D.Audio.IAudioManager](#).

```

82     {
83         var existing = _sounds.FirstOrDefault(s => s.Name == name);
84         if (existing != null)
85             return existing;
86
87         var loaded = _sounds.FirstOrDefault(s => s.File == file) != null;
88         if (loaded)
89             throw new Exception("The sound file \"" + file + "\" has already been loaded under a
different name.");
90
91         var sound = new Sound(name, file, format);
92         _sounds.Add(sound);
93         return sound;
94     }

```

### 7.1.3.8 void TriDevs.TriEngine2D.Audio.AudioManager.StopAll ( )

Immediately stops all playback of sounds and songs.

Implements [TriDevs.TriEngine2D.Audio.IAudioManager](#).

```

76     {
77         StopAllSounds();
78         StopAllSongs();
79     }

```

### 7.1.3.9 void TriDevs.TriEngine2D.Audio.AudioManager.StopAllSongs ( )

Immediately stops playback of all songs.

Implements [TriDevs.TriEngine2D.Audio.IAudioManager](#).

```
137         {
138             _songs.ForEach(s => s.Stop());
139         }
```

### 7.1.3.10 void TriDevs.TriEngine2D.Audio.AudioManager.StopAllSounds ( )

Immediately stops playback of all sounds.

Implements [TriDevs.TriEngine2D.Audio.IAudioManager](#).

```
107         {
108             _sounds.ForEach(s => s.Stop());
109         }
```

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

- [TriDevs.TriEngine2D/Audio/AudioManager.cs](#)

## 7.2 TriDevs.TriEngine2D.Extensions.EnumerationExtensions Class Reference

[Extensions](#) for System.Enum.

### Static Public Member Functions

- static T [Include< T >](#) (this Enum value, T append)  
*Includes an enumerated type and returns the new value.*
- static T [Remove< T >](#) (this Enum value, T remove)  
*Removes an enumerated type and returns the new value.*
- static bool [Has< T >](#) (this Enum value, T check)  
*Checks if an enumerated type contains a value.*
- static bool [Missing< T >](#) (this Enum obj, T value)  
*Checks if an enumerated type is missing a value.*

### 7.2.1 Detailed Description

[Extensions](#) for System.Enum.

### 7.2.2 Member Function Documentation

#### 7.2.2.1 static bool TriDevs.TriEngine2D.Extensions.EnumerationExtensions.Has< T > ( this Enum value, T check ) [static]

Checks if an enumerated type contains a value.

#### Template Parameters

<i>T</i>	The enum type.
----------	----------------



## Parameters

<i>value</i>	The enum to check.
<i>check</i>	Value to check for.

## Returns

True if the enum has the value(s), false otherwise.

```

148     {
149         Type type = value.GetType();
150
151         //determine the values
152         var parsed = new _Value(check, type);
153         if (parsed.Signed.HasValue) //if (parsed.Signed is long)
154         {
155             return (Convert.ToInt64(value) & (long)parsed.Signed) == (long)parsed.Signed;
156         }
157         if (parsed.Unsigned.HasValue) //if (parsed.Unsigned is ulong)
158         {
159             return (Convert.ToUInt64(value) & (ulong)parsed.Unsigned) == (ulong)parsed.Unsigned;
160         }
161         return false;
162     }

```

**7.2.2.2** `static T TriDevs.TriEngine2D.Extensions.EnumerationExtensions.Include< T > ( this Enum value, T append )`  
`[static]`

Includes an enumerated type and returns the new value.

## Template Parameters

<i>T</i>	The enum type.
----------	----------------

## Parameters

<i>value</i>	The enum to append to.
<i>append</i>	Value to append.

## Returns

New enum T with the new values.

```

94     {
95         var type = value.GetType();
96
97         //determine the values
98         object result = value;
99         var parsed = new _Value(append, type);
100         if (parsed.Signed.HasValue) //if (parsed.Signed is long)
101         {
102             result = Convert.ToInt64(value) | (long)parsed.Signed;
103         }
104         else if (parsed.Unsigned.HasValue) //else if (parsed.Unsigned is ulong)
105         {
106             result = Convert.ToUInt64(value) | (ulong)parsed.Unsigned;
107         }
108
109         //return the final value
110         return (T)Enum.Parse(type, result.ToString());
111     }

```

**7.2.2.3** `static bool TriDevs.TriEngine2D.Extensions.EnumerationExtensions.Missing< T > ( this Enum obj, T value )`  
`[static]`

Checks if an enumerated type is missing a value.

## Template Parameters

<i>T</i>	The enum type.
----------	----------------

## Parameters

<i>obj</i>	The enum to check.
<i>value</i>	Value to check for.

## Returns

True if the enum is missing the value(s), false otherwise.

```

172     {
173         return !Has(obj, value);
174     }
```

**7.2.2.4** `static T TriDevs.TriEngine2D.Extensions.EnumerationExtensions.Remove< T > ( this Enum value, T remove )`  
`[static]`

Removes an enumerated type and returns the new value.

## Template Parameters

<i>T</i>	The enum type.
----------	----------------

## Parameters

<i>value</i>	The enum to remove from.
<i>remove</i>	Value to remove.

## Returns

New enum T with the value(s) removed.

```

121     {
122         Type type = value.GetType();
123
124         //determine the values
125         object result = value;
126         var parsed = new _Value(remove, type);
127         if (parsed.Signed.HasValue) //if (parsed.Signed is long)
128         {
129             result = Convert.ToInt64(value) & ~(long)parsed.Signed;
130         }
131         else if (parsed.Unsigned.HasValue) //else if (parsed.Unsigned is ulong)
132         {
133             result = Convert.ToUInt64(value) & ~(ulong)parsed.Unsigned;
134         }
135
136         //return the final value
137         return (T)Enum.Parse(type, result.ToString());
138     }
```

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

- TriDevs.TriEngine2D/Extensions/[EnumerationExtensions.cs](#)

## 7.3 TriDevs.TriEngine2D.Native.Helpers Class Reference

Helper class with various methods to help native coding and debugging.

## Static Public Member Functions

- static int [GetLastError](#) ()  
*Gets the last error that was thrown.*
- static int [GetLastErrorInfo](#) (out string message)  
*Gets information about the last error that was thrown.*
- static string [GetErrorMessage](#) (int err)  
*Gets the error message associated with an error code.*
- static string [GetLastErrorMessage](#) ()  
*Gets the error message associated with the last thrown error.*
- static Win32Exception [GetWin32Exception](#) (int err)  
*Gets the Win32Exception associated with the specified error code.*
- static Win32Exception [GetLastWin32Exception](#) ()  
*Gets the Win32Exception associated with the last thrown error.*

### 7.3.1 Detailed Description

Helper class with various methods to help native coding and debugging.

### 7.3.2 Member Function Documentation

#### 7.3.2.1 static string TriDevs.TriEngine2D.Native.Helpers.GetErrorMessage ( int *err* ) [static]

Gets the error message associated with an error code.

##### Parameters

<i>err</i>	The error code to get information about.
------------	--

##### Returns

The error message.

```

61         {
62             return GetWin32Exception(err).Message;
63         }
```

#### 7.3.2.2 static int TriDevs.TriEngine2D.Native.Helpers.GetLastError ( ) [static]

Gets the last error that was thrown.

##### Returns

The error code associated with the thrown error.

```

39         {
40             return Marshal.GetLastWin32Error();
41         }
```

#### 7.3.2.3 static int TriDevs.TriEngine2D.Native.Helpers.GetLastErrorInfo ( out string *message* ) [static]

Gets information about the last error that was thrown.

##### Parameters

<i>message</i>	Will be set to the error message.
----------------	-----------------------------------

**Returns**

The error code associated with the thrown error.

```

49         {
50             var err = GetLastError();
51             message = GetWin32Exception(err).Message;
52             return err;
53         }

```

**7.3.2.4 static string TriDevs.TriEngine2D.Native.Helpers.GetLastErrorMessage ( ) [static]**

Gets the error message associated with the last thrown error.

**Returns**

The error message.

```

70         {
71             return GetErrorMessage(GetLastError());
72         }

```

**7.3.2.5 static Win32Exception TriDevs.TriEngine2D.Native.Helpers.GetLastWin32Exception ( ) [static]**

Gets the Win32Exception associated with the last thrown error.

**Returns**

The Win32Exception.

```

89         {
90             return GetWin32Exception(GetLastError());
91         }

```

**7.3.2.6 static Win32Exception TriDevs.TriEngine2D.Native.Helpers.GetWin32Exception ( int err ) [static]**

Gets the Win32Exception associated with the specified error code.

**Parameters**

<i>err</i>	The error code.
------------	-----------------

**Returns**

The Win32Exception for the provided error code.

```

80         {
81             return new Win32Exception(err);
82         }

```

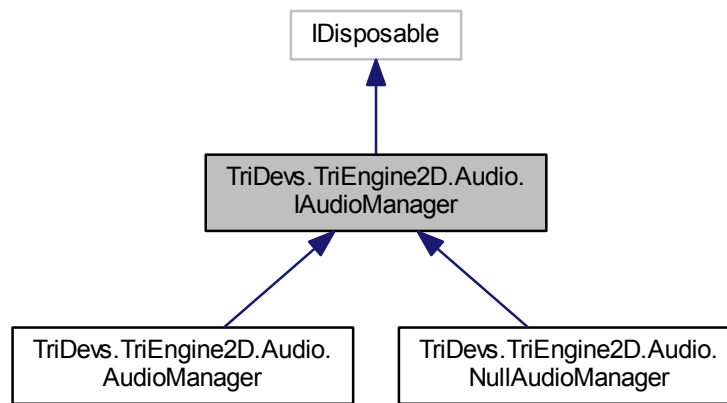
The documentation for this class was generated from the following file:

- TriDevs.TriEngine2D/Native/[Helpers.cs](#)

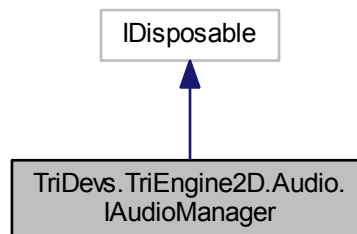
## 7.4 TriDevs.TriEngine2D.Audio.IAudioManager Interface Reference

Provides various methods to manipulate audio.

Inheritance diagram for TriDevs.TriEngine2D.Audio.IAudioManager:



Collaboration diagram for TriDevs.TriEngine2D.Audio.IAudioManager:



### Public Member Functions

- void `StopAll` ()  
*Immediately stops all playback of sounds and songs.*
- `ISound LoadSound` (string name, string file, `AudioFormat` format=`AudioFormat.Wav`)  
*Loads a sound into the audio manager.*
- bool `HasSound` (string name)  
*Returns a value indicating whether a sound with the specified name has been loaded into the audio manager.*
- `ISound GetSound` (string name)  
*Gets the sound with the specified name.*
- void `StopAllSounds` ()

*Immediately stops playback of all sounds.*

- [ISong LoadSong](#) (string name, string file, [AudioFormat](#) format=AudioFormat.Ogg)

*Load a song into the audio manager.*

- bool [HasSong](#) (string name)

*Returns a value indicating whether a song with the specified name has been loaded into the audio manager.*

- [ISong GetSong](#) (string name)

*Gets the song with the specified name.*

- void [StopAllSongs](#) ()

*Immediately stops playback of all songs.*

### 7.4.1 Detailed Description

Provides various methods to manipulate audio.

### 7.4.2 Member Function Documentation

#### 7.4.2.1 [ISong](#) [TriDevs.TriEngine2D.Audio.IAudioManager.GetSong](#) ( string *name* )

Gets the song with the specified name.

##### Parameters

<i>name</i>	Name of the song to get.
-------------	--------------------------

##### Returns

An object implementing the [ISong](#) interface, or null if no song matched the requested name.

Implemented in [TriDevs.TriEngine2D.Audio.AudioManager](#), and [TriDevs.TriEngine2D.Audio.NullAudioManager](#).

#### 7.4.2.2 [ISound](#) [TriDevs.TriEngine2D.Audio.IAudioManager.GetSound](#) ( string *name* )

Gets the sound with the specified name.

##### Parameters

<i>name</i>	Name of the sound to get.
-------------	---------------------------

##### Returns

An object implementing the [ISound](#) interface, or null if no sound matched the requested name.

Implemented in [TriDevs.TriEngine2D.Audio.AudioManager](#), and [TriDevs.TriEngine2D.Audio.NullAudioManager](#).

#### 7.4.2.3 bool [TriDevs.TriEngine2D.Audio.IAudioManager.HasSong](#) ( string *name* )

Returns a value indicating whether a song with the specified name has been loaded into the audio manager.

##### Parameters

<i>name</i>	Name to check for.
-------------	--------------------

**Returns**

True if the song has been loaded, false otherwise.

Implemented in [TriDevs.TriEngine2D.Audio.AudioManager](#), and [TriDevs.TriEngine2D.Audio.NullAudioManager](#).

**7.4.2.4 bool TriDevs.TriEngine2D.Audio.IAudioManager.HasSound ( string name )**

Returns a value indicating whether a sound with the specified name has been loaded into the audio manager.

**Parameters**

<i>name</i>	Name to check for.
-------------	--------------------

**Returns**

True if the sound has been loaded, false otherwise.

Implemented in [TriDevs.TriEngine2D.Audio.AudioManager](#), and [TriDevs.TriEngine2D.Audio.NullAudioManager](#).

**7.4.2.5 ISong TriDevs.TriEngine2D.Audio.IAudioManager.LoadSong ( string name, string file, AudioFormat format = AudioFormat.Ogg )**

Load a song into the audio manager.

If a song with the same name has already been loaded, it will be returned instead.

**Parameters**

<i>name</i>	Name to assign to the new song object.
<i>file</i>	Path to the file that will be used.
<i>format</i>	Format of the file.

**Returns**

The resulting object implementing the [ISong](#) interface.

Implemented in [TriDevs.TriEngine2D.Audio.AudioManager](#), and [TriDevs.TriEngine2D.Audio.NullAudioManager](#).

**7.4.2.6 ISound TriDevs.TriEngine2D.Audio.IAudioManager.LoadSound ( string name, string file, AudioFormat format = AudioFormat.Wav )**

Loads a sound into the audio manager.

If a sound with the same name has already been loaded, it will be returned instead.

**Parameters**

<i>name</i>	Name to assign to the new sound object.
<i>file</i>	Path to the file that will be used.
<i>format</i>	The format of the file to load.

**Returns**

The resulting object implementing the [ISound](#) interface.

Implemented in [TriDevs.TriEngine2D.Audio.AudioManager](#), and [TriDevs.TriEngine2D.Audio.NullAudioManager](#).

#### 7.4.2.7 void TriDevs.TriEngine2D.Audio.IAudioManager.StopAll ( )

Immediately stops all playback of sounds and songs.

Implemented in [TriDevs.TriEngine2D.Audio.AudioManager](#), and [TriDevs.TriEngine2D.Audio.NullAudioManager](#).

#### 7.4.2.8 void TriDevs.TriEngine2D.Audio.IAudioManager.StopAllSongs ( )

Immediately stops playback of all songs.

Implemented in [TriDevs.TriEngine2D.Audio.AudioManager](#), and [TriDevs.TriEngine2D.Audio.NullAudioManager](#).

#### 7.4.2.9 void TriDevs.TriEngine2D.Audio.IAudioManager.StopAllSounds ( )

Immediately stops playback of all sounds.

Implemented in [TriDevs.TriEngine2D.Audio.AudioManager](#), and [TriDevs.TriEngine2D.Audio.NullAudioManager](#).

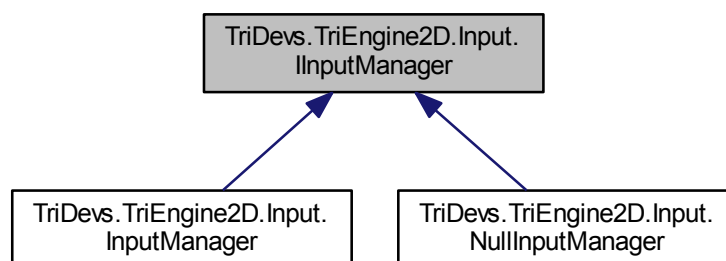
The documentation for this interface was generated from the following file:

- [TriDevs.TriEngine2D/Audio/IAudioManager.cs](#)

## 7.5 TriDevs.TriEngine2D.Input.IInputManager Interface Reference

Provides various methods to query input devices like the keyboard.

Inheritance diagram for TriDevs.TriEngine2D.Input.IInputManager:



### Public Member Functions

- void [Update](#) ( )  
*Updates the input manager, refreshing all current and previous states.*
- bool [KeyUp](#) (Key key)  
*Returns whether or not the specified key is currently unpressed.*
- bool [KeyDown](#) (Key key)  
*Returns whether or not the specified key is currently being pressed.*
- bool [KeyPressed](#) (Key key)  
*Returns whether or not the specified key has been pressed.*
- bool [KeyReleased](#) (Key key)



- Returns whether or not the specified key has been released.*
- bool [ButtonUp](#) (MouseButton button)  
*Returns whether or not the specified mouse button is currently unpressed.*
- bool [ButtonDown](#) (MouseButton button)  
*Returns whether or not the specified mouse button is currently being pressed.*
- bool [ButtonPressed](#) (MouseButton button)  
*Returns whether or not the specified mouse button has been pressed.*
- bool [ButtonReleased](#) (MouseButton button)  
*Returns whether or not the specified mouse button has been released.*
- bool [WheelUp](#) ()  
*Returns whether the mouse wheel was scrolled up.*
- bool [WheelDown](#) ()  
*Returns whether the mouse wheel was scrolled down.*
- bool [WheelChanged](#) ()  
*Returns whether the mouse wheel scrolled at all.*
- int [WheelChange](#) ()  
*Returns the mouse wheel's change in value.*

## Properties

- int [MouseX](#) [get]  
*Gets the absolute X position of the pointer, in window pixel coordinates.*
- int [MouseY](#) [get]  
*Gets the absolute Y position of the pointer, in window pixel coordinates.*
- [Point](#) [MousePosition](#) [get]  
*Gets a [Point](#) representing the position of the mouse pointer, in window pixel coordinates.*
- int [MouseWheelValue](#) [get]  
*Gets the current value of the mouse wheel.*
- bool [this\[Key key\]](#) [get]  
*Gets a boolean value indicating whether the specified OpenTK.Input.Key is pressed.*
- bool [this\[MouseButton button\]](#) [get]  
*Gets a boolean value indicating whether the specified OpenTK.Input.MouseButton is pressed.*

### 7.5.1 Detailed Description

Provides various methods to query input devices like the keyboard.

### 7.5.2 Member Function Documentation

#### 7.5.2.1 bool TriDevs.TriEngine2D.Input.IInputManager.ButtonDown ( MouseButton button )

Returns whether or not the specified mouse button is currently being pressed.

##### Parameters

<i>button</i>	The button to query for.
---------------	--------------------------

##### Returns

True if button is currently being pressed, false otherwise.

Implemented in [TriDevs.TriEngine2D.Input.InputManager](#), and [TriDevs.TriEngine2D.Input.NullInputManager](#).

#### 7.5.2.2 bool TriDevs.TriEngine2D.Input.IInputManager.ButtonPressed ( MouseButton *button* )

Returns whether or not the specified mouse button has been pressed.

Only returns true if the last state of the mouse button was not pressed.

##### Parameters

<i>button</i>	Button to query for.
---------------	----------------------

##### Returns

True if button was pressed, false otherwise.

Implemented in [TriDevs.TriEngine2D.Input.InputManager](#), and [TriDevs.TriEngine2D.Input.NullInputManager](#).

#### 7.5.2.3 bool TriDevs.TriEngine2D.Input.IInputManager.ButtonReleased ( MouseButton *button* )

Returns whether or not the specified mouse button has been released.

Only returns true if the last state of the button was pressed.

##### Parameters

<i>button</i>	The button to query for.
---------------	--------------------------

##### Returns

True if the button was released, false otherwise.

Implemented in [TriDevs.TriEngine2D.Input.InputManager](#), and [TriDevs.TriEngine2D.Input.NullInputManager](#).

#### 7.5.2.4 bool TriDevs.TriEngine2D.Input.IInputManager.ButtonUp ( MouseButton *button* )

Returns whether or not the specified mouse button is currently unpressed.

##### Parameters

<i>button</i>	Button to query for.
---------------	----------------------

##### Returns

True if the button is currently up (not pressed), false otherwise.

Implemented in [TriDevs.TriEngine2D.Input.InputManager](#), and [TriDevs.TriEngine2D.Input.NullInputManager](#).

#### 7.5.2.5 bool TriDevs.TriEngine2D.Input.IInputManager.KeyDown ( Key *key* )

Returns whether or not the specified key is currently being pressed.

##### Parameters

<i>key</i>	Key to query for.
------------	-------------------

**Returns**

True if key is currently being pressed, false otherwise.

Implemented in [TriDevs.TriEngine2D.Input.InputManager](#), and [TriDevs.TriEngine2D.Input.NullInputManager](#).

**7.5.2.6 bool TriDevs.TriEngine2D.Input.IInputManager.KeyPressed ( Key key )**

Returns whether or not the specified key has been pressed.

Only returns true if the last state of the key was not pressed.

**Parameters**

<i>key</i>	Key to query for.
------------	-------------------

**Returns**

True if key was pressed, false otherwise.

Implemented in [TriDevs.TriEngine2D.Input.InputManager](#), and [TriDevs.TriEngine2D.Input.NullInputManager](#).

**7.5.2.7 bool TriDevs.TriEngine2D.Input.IInputManager.KeyReleased ( Key key )**

Returns whether or not the specified key has been released.

Only returns true if the last state of the key was pressed.

**Parameters**

<i>key</i>	Key to query for.
------------	-------------------

**Returns**

True if key was released, false otherwise.

Implemented in [TriDevs.TriEngine2D.Input.InputManager](#), and [TriDevs.TriEngine2D.Input.NullInputManager](#).

**7.5.2.8 bool TriDevs.TriEngine2D.Input.IInputManager.KeyUp ( Key key )**

Returns whether or not the specified key is currently unpressed.

**Parameters**

<i>key</i>	Key to query for.
------------	-------------------

**Returns**

True if the key is currently up (not pressed), false otherwise.

Implemented in [TriDevs.TriEngine2D.Input.InputManager](#), and [TriDevs.TriEngine2D.Input.NullInputManager](#).

**7.5.2.9 void TriDevs.TriEngine2D.Input.IInputManager.Update ( )**

Updates the input manager, refreshing all current and previous states.

Implemented in [TriDevs.TriEngine2D.Input.InputManager](#), and [TriDevs.TriEngine2D.Input.NullInputManager](#).

#### 7.5.2.10 `int TriDevs.TriEngine2D.Input.IInputManager.WheelChange ( )`

Returns the mouse wheel's change in value.

##### Returns

Negative value if wheel scrolled down, positive value if scrolled up, zero if not scrolled.

Implemented in [TriDevs.TriEngine2D.Input.InputManager](#), and [TriDevs.TriEngine2D.Input.NullInputManager](#).

#### 7.5.2.11 `bool TriDevs.TriEngine2D.Input.IInputManager.WheelChanged ( )`

Returns whether the mouse wheel scrolled at all.

##### Returns

True if the mouse wheel scrolled, false otherwise.

Implemented in [TriDevs.TriEngine2D.Input.InputManager](#), and [TriDevs.TriEngine2D.Input.NullInputManager](#).

#### 7.5.2.12 `bool TriDevs.TriEngine2D.Input.IInputManager.WheelDown ( )`

Returns whether the mouse wheel was scrolled down.

##### Returns

True if mouse wheel was scrolled down, false otherwise.

Implemented in [TriDevs.TriEngine2D.Input.InputManager](#), and [TriDevs.TriEngine2D.Input.NullInputManager](#).

#### 7.5.2.13 `bool TriDevs.TriEngine2D.Input.IInputManager.WheelUp ( )`

Returns whether the mouse wheel was scrolled up.

##### Returns

True if mouse wheel was scrolled up, false otherwise.

Implemented in [TriDevs.TriEngine2D.Input.InputManager](#), and [TriDevs.TriEngine2D.Input.NullInputManager](#).

### 7.5.3 Property Documentation

#### 7.5.3.1 `Point TriDevs.TriEngine2D.Input.IInputManager.MousePosition` [get]

Gets a [Point](#) representing the position of the mouse pointer, in window pixel coordinates.

#### 7.5.3.2 `int TriDevs.TriEngine2D.Input.IInputManager.MouseWheelValue` [get]

Gets the current value of the mouse wheel.

#### 7.5.3.3 `int TriDevs.TriEngine2D.Input.IInputManager.MouseX` [get]

Gets the absolute X position of the pointer, in window pixel coordinates.

7.5.3.4 `int TriDevs.TriEngine2D.Input.IInputManager.MouseY` `[get]`

Gets the absolute Y position of the pointer, in window pixel coordinates.

7.5.3.5 `bool TriDevs.TriEngine2D.Input.IInputManager.this[Key key]` `[get]`

Gets a boolean value indicating whether the specified OpenTK.Input.Key is pressed.

## Parameters

<i>key</i>	The key to query.
------------	-------------------

## Returns

True if pressed, false otherwise.

7.5.3.6 `bool TriDevs.TriEngine2D.Input.IInputManager.this[MouseButton button]` `[get]`

Gets a boolean value indicating whether the specified OpenTK.Input.MouseButton is pressed.

## Parameters

<i>button</i>	The button to query.
---------------	----------------------

## Returns

True if pressed, false otherwise.

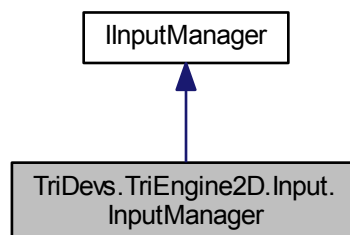
The documentation for this interface was generated from the following file:

- [TriDevs.TriEngine2D/Input/IInputManager.cs](#)

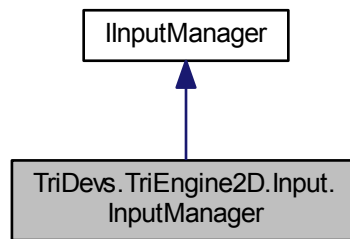
## 7.6 TriDevs.TriEngine2D.Input.InputManager Class Reference

[Input](#) manager interfacing with input methods provided by a `GameWindow`.

Inheritance diagram for `TriDevs.TriEngine2D.Input.InputManager`:



Collaboration diagram for TriDevs.TriEngine2D.Input.InputManager:



## Public Member Functions

- [InputManager](#) (GameWindow window)  
Creates a new [InputManager](#) associated with the specified GameWindow.
- void [Update](#) ()  
Updates the input manager, refreshing all current and previous states.
- bool [KeyUp](#) (Key key)  
Returns whether or not the specified key is currently unpressed.
- bool [KeyDown](#) (Key key)  
Returns whether or not the specified key is currently being pressed.
- bool [KeyPressed](#) (Key key)  
Returns whether or not the specified key has been pressed.
- bool [KeyReleased](#) (Key key)  
Returns whether or not the specified key has been released.
- bool [ButtonUp](#) (MouseButton button)  
Returns whether or not the specified mouse button is currently unpressed.
- bool [ButtonDown](#) (MouseButton button)  
Returns whether or not the specified mouse button is currently being pressed.
- bool [ButtonPressed](#) (MouseButton button)  
Returns whether or not the specified mouse button has been pressed.
- bool [ButtonReleased](#) (MouseButton button)  
Returns whether or not the specified mouse button has been released.
- bool [WheelUp](#) ()  
Returns whether the mouse wheel was scrolled up.
- bool [WheelDown](#) ()  
Returns whether the mouse wheel was scrolled down.
- bool [WheelChanged](#) ()  
Returns whether the mouse wheel scrolled at all.
- int [WheelChange](#) ()  
Returns the mouse wheel's change in value.

## Properties

- int [MouseX](#) [get]
- int [MouseY](#) [get]
- [Point](#) [MousePosition](#) [get]
- int [MouseWheelValue](#) [get]
- bool [this\[Key key\]](#) [get]
- bool [this\[MouseButton button\]](#) [get]

### 7.6.1 Detailed Description

[Input](#) manager interfacing with input methods provided by a [GameWindow](#).

### 7.6.2 Constructor & Destructor Documentation

#### 7.6.2.1 [TriDevs.TriEngine2D.Input.InputManager.InputManager](#) ( [GameWindow](#) *window* )

Creates a new [InputManager](#) associated with the specified [GameWindow](#).

##### Parameters

<i>window</i>	The <a href="#">GameWindow</a> this <a href="#">InputManager</a> will interface with.
---------------	---

```

62         {
63             _mouse = window.Mouse;
64         }
```

### 7.6.3 Member Function Documentation

#### 7.6.3.1 [bool](#) [TriDevs.TriEngine2D.Input.InputManager.ButtonDown](#) ( [MouseButton](#) *button* )

Returns whether or not the specified mouse button is currently being pressed.

##### Parameters

<i>button</i>	The button to query for.
---------------	--------------------------

##### Returns

True if button is currently being pressed, false otherwise.

Implements [TriDevs.TriEngine2D.Input.IInputManager](#).

```

101         {
102             return _mouseState[button];
103         }
```

#### 7.6.3.2 [bool](#) [TriDevs.TriEngine2D.Input.InputManager.ButtonPressed](#) ( [MouseButton](#) *button* )

Returns whether or not the specified mouse button has been pressed.

Only returns true if the last state of the mouse button was not pressed.

##### Parameters

<i>button</i>	Button to query for.
---------------	----------------------

**Returns**

True if button was pressed, false otherwise.

Implements [TriDevs.TriEngine2D.Input.IInputManager](#).

```

106         {
107             return _mouseState[button] && !_lastMouseState[button];
108         }

```

**7.6.3.3 bool TriDevs.TriEngine2D.Input.InputManager.ButtonReleased ( MouseButton *button* )**

Returns whether or not the specified mouse button has been released.

Only returns true if the last state of the button was pressed.

**Parameters**

<i>button</i>	The button to query for.
---------------	--------------------------

**Returns**

True if the button was released, false otherwise.

Implements [TriDevs.TriEngine2D.Input.IInputManager](#).

```

111         {
112             return !_mouseState[button] && _lastMouseState[button];
113         }

```

**7.6.3.4 bool TriDevs.TriEngine2D.Input.InputManager.ButtonUp ( MouseButton *button* )**

Returns whether or not the specified mouse button is currently unpressed.

**Parameters**

<i>button</i>	Button to query for.
---------------	----------------------

**Returns**

True if the button is currently up (not pressed), false otherwise.

Implements [TriDevs.TriEngine2D.Input.IInputManager](#).

```

96         {
97             return !_mouseState[button];
98         }

```

**7.6.3.5 bool TriDevs.TriEngine2D.Input.InputManager.KeyDown ( Key *key* )**

Returns whether or not the specified key is currently being pressed.

**Parameters**

<i>key</i>	Key to query for.
------------	-------------------



**Returns**

True if key is currently being pressed, false otherwise.

Implements [TriDevs.TriEngine2D.Input.IInputManager](#).

```
81         {
82             return _keyboardState[key];
83         }
```

**7.6.3.6 bool TriDevs.TriEngine2D.Input.InputManager.KeyPressed ( Key key )**

Returns whether or not the specified key has been pressed.

Only returns true if the last state of the key was not pressed.

**Parameters**

<i>key</i>	Key to query for.
------------	-------------------

**Returns**

True if key was pressed, false otherwise.

Implements [TriDevs.TriEngine2D.Input.IInputManager](#).

```
86         {
87             return _keyboardState[key] && !_lastKeyboardState[key];
88         }
```

**7.6.3.7 bool TriDevs.TriEngine2D.Input.InputManager.KeyReleased ( Key key )**

Returns whether or not the specified key has been released.

Only returns true if the last state of the key was pressed.

**Parameters**

<i>key</i>	Key to query for.
------------	-------------------

**Returns**

True if key was released, false otherwise.

Implements [TriDevs.TriEngine2D.Input.IInputManager](#).

```
91         {
92             return !_keyboardState[key] && _lastKeyboardState[key];
93         }
```

**7.6.3.8 bool TriDevs.TriEngine2D.Input.InputManager.KeyUp ( Key key )**

Returns whether or not the specified key is currently unpressed.

**Parameters**

<i>key</i>	Key to query for.
------------	-------------------

**Returns**

True if the key is currently up (not pressed), false otherwise.

Implements [TriDevs.TriEngine2D.Input.IInputManager](#).

```

76         {
77             return !_keyboardState[key];
78         }

```

**7.6.3.9 void TriDevs.TriEngine2D.Input.InputManager.Update ( )**

Updates the input manager, refreshing all current and previous states.

Implements [TriDevs.TriEngine2D.Input.IInputManager](#).

```

67         {
68             _lastKeyboardState = _keyboardState;
69             _keyboardState = Keyboard.GetState();
70
71             _lastMouseState = _mouseState;
72             _mouseState = Mouse.GetState();
73         }

```

**7.6.3.10 int TriDevs.TriEngine2D.Input.InputManager.WheelChange ( )**

Returns the mouse wheel's change in value.

**Returns**

Negative value if wheel scrolled down, positive value if scrolled up, zero if not scrolled.

Implements [TriDevs.TriEngine2D.Input.IInputManager](#).

```

131         {
132             return _mouseState.Wheel - _lastMouseState.Wheel;
133         }

```

**7.6.3.11 bool TriDevs.TriEngine2D.Input.InputManager.WheelChanged ( )**

Returns whether the mouse wheel scrolled at all.

**Returns**

True if the mouse wheel scrolled, false otherwise.

Implements [TriDevs.TriEngine2D.Input.IInputManager](#).

```

126         {
127             return _mouseState.Wheel != _lastMouseState.Wheel;
128         }

```

**7.6.3.12 bool TriDevs.TriEngine2D.Input.InputManager.WheelDown ( )**

Returns whether the mouse wheel was scrolled down.

**Returns**

True if mouse wheel was scrolled down, false otherwise.

Implements [TriDevs.TriEngine2D.Input.IInputManager](#).

```

121         {
122             return _mouseState.Wheel < _lastMouseState.Wheel;
123         }

```

## 7.6.3.13 bool TriDevs.TriEngine2D.Input.InputManager.WheelUp ( )

Returns whether the mouse wheel was scrolled up.

## Returns

True if mouse wheel was scrolled up, false otherwise.

Implements [TriDevs.TriEngine2D.Input.IInputManager](#).

```

116         {
117             return _mouseState.Wheel > _lastMouseState.Wheel;
118         }

```

## 7.6.4 Property Documentation

7.6.4.1 Point TriDevs.TriEngine2D.Input.InputManager.MousePosition [get]

7.6.4.2 int TriDevs.TriEngine2D.Input.InputManager.MouseWheelValue [get]

7.6.4.3 int TriDevs.TriEngine2D.Input.InputManager.MouseX [get]

7.6.4.4 int TriDevs.TriEngine2D.Input.InputManager.MouseY [get]

7.6.4.5 bool TriDevs.TriEngine2D.Input.InputManager.this[Key key] [get]

7.6.4.6 bool TriDevs.TriEngine2D.Input.InputManager.this[MouseButton button] [get]

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

- TriDevs.TriEngine2D/Input/[InputManager.cs](#)

## 7.7 TriDevs.TriEngine2D.Helpers.IO Class Reference

Provides various helper functions for doing [IO](#) operations.

## Static Public Member Functions

- static string [GetAbsolutePath](#) (string path)  
*Resolves the absolute path from a relative path.*

## 7.7.1 Detailed Description

Provides various helper functions for doing [IO](#) operations.

## 7.7.2 Member Function Documentation

7.7.2.1 static string TriDevs.TriEngine2D.Helpers.IO.GetAbsolutePath ( string path ) [static]

Resolves the absolute path from a relative path.

## Parameters

<i>path</i>	The relative path to resolve.
-------------	-------------------------------

### Returns

The absolute path to the item.

```
39      {  
40          return Path.Combine(Directory.GetCurrentDirectory(), path);  
41      }
```

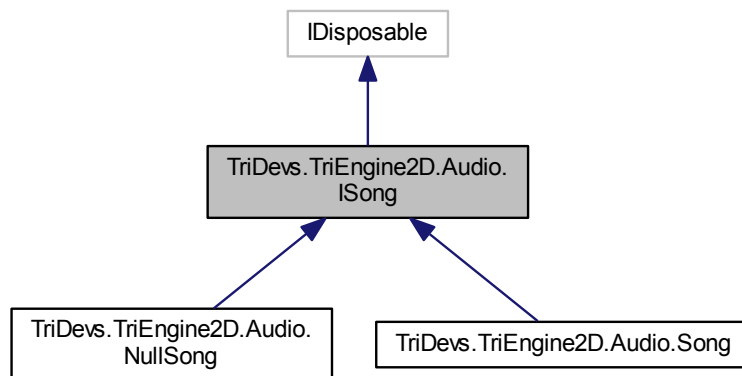
The documentation for this class was generated from the following file:

- [TriDevs.TriEngine2D/Helpers/IO.cs](#)

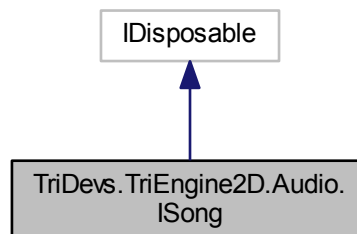
## 7.8 TriDevs.TriEngine2D.Audio.ISong Interface Reference

A song that will be streamed in the audio player.

Inheritance diagram for TriDevs.TriEngine2D.Audio.ISong:



Collaboration diagram for TriDevs.TriEngine2D.Audio.ISong:



### Public Member Functions

- void [Play](#) ()

- Starts playback of the song.*
- void [Stop](#) ()  
*Stops playback of the song.*
- void [Pause](#) ()  
*Pauses playback of the song.*
- void [Resume](#) ()  
*Resumes playback of a paused song.*

## Properties

- string [Name](#) [get]  
*Unique name of this song object.*
- string [File](#) [get]  
*Path to the file associated with this song object.*
- float [Volume](#) [get, set]  
*Gets or sets the song volume.*
- bool [IsLooped](#) [get, set]  
*Gets or sets a value indicating that the song should be looped once it reaches the end.*

### 7.8.1 Detailed Description

A song that will be streamed in the audio player.

### 7.8.2 Member Function Documentation

#### 7.8.2.1 void TriDevs.TriEngine2D.Audio.ISong.Pause ( )

Pauses playback of the song.

Call the [Resume](#) or [Play](#) method to resume playback.

Implemented in [TriDevs.TriEngine2D.Audio.Song](#), and [TriDevs.TriEngine2D.Audio.NullSong](#).

#### 7.8.2.2 void TriDevs.TriEngine2D.Audio.ISong.Play ( )

Starts playback of the song.

Implemented in [TriDevs.TriEngine2D.Audio.Song](#), and [TriDevs.TriEngine2D.Audio.NullSong](#).

#### 7.8.2.3 void TriDevs.TriEngine2D.Audio.ISong.Resume ( )

Resumes playback of a paused song.

Implemented in [TriDevs.TriEngine2D.Audio.Song](#), and [TriDevs.TriEngine2D.Audio.NullSong](#).

#### 7.8.2.4 void TriDevs.TriEngine2D.Audio.ISong.Stop ( )

Stops playback of the song.

Implemented in [TriDevs.TriEngine2D.Audio.Song](#), and [TriDevs.TriEngine2D.Audio.NullSong](#).

### 7.8.3 Property Documentation

7.8.3.1 `string TriDevs.TriEngine2D.Audio.ISong.File` `[get]`

Path to the file associated with this song object.

7.8.3.2 `bool TriDevs.TriEngine2D.Audio.ISong.IsLooped` `[get]`, `[set]`

Gets or sets a value indicating that the song should be looped once it reaches the end.

7.8.3.3 `string TriDevs.TriEngine2D.Audio.ISong.Name` `[get]`

Unique name of this song object.

7.8.3.4 `float TriDevs.TriEngine2D.Audio.ISong.Volume` `[get]`, `[set]`

Gets or sets the song volume.

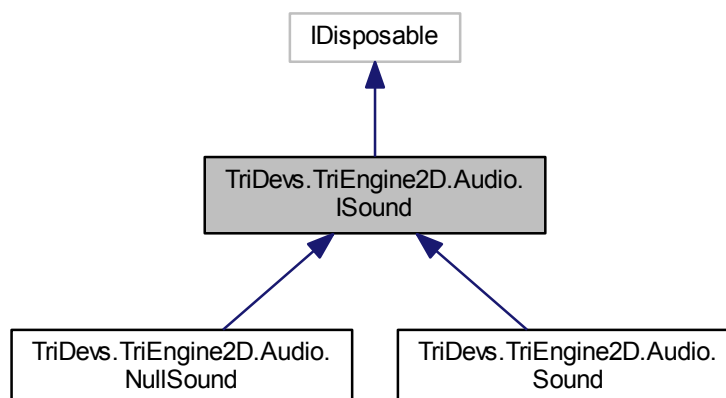
The documentation for this interface was generated from the following file:

- `TriDevs.TriEngine2D/Audio/ISong.cs`

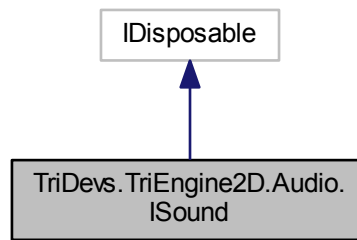
## 7.9 TriDevs.TriEngine2D.Audio.ISound Interface Reference

A sound file for use with the audio manager.

Inheritance diagram for `TriDevs.TriEngine2D.Audio.ISound`:



Collaboration diagram for TriDevs.TriEngine2D.Audio.ISound:



### Public Member Functions

- void [Play](#) ()  
*Plays the sound.*
- void [Stop](#) ()  
*Stops the sound.*

### Properties

- string [Name](#) [get]  
*Unique name of this sound object.*
- string [File](#) [get]  
*Path to the file this sound object is associated with.*

#### 7.9.1 Detailed Description

A sound file for use with the audio manager.

#### 7.9.2 Member Function Documentation

##### 7.9.2.1 void TriDevs.TriEngine2D.Audio.ISound.Play ( )

Plays the sound.

Implemented in [TriDevs.TriEngine2D.Audio.Sound](#), and [TriDevs.TriEngine2D.Audio.NullSound](#).

##### 7.9.2.2 void TriDevs.TriEngine2D.Audio.ISound.Stop ( )

Stops the sound.

Implemented in [TriDevs.TriEngine2D.Audio.Sound](#), and [TriDevs.TriEngine2D.Audio.NullSound](#).

### 7.9.3 Property Documentation

#### 7.9.3.1 `string TriDevs.TriEngine2D.Audio.ISound.File` `[get]`

Path to the file this sound object is associated with.

#### 7.9.3.2 `string TriDevs.TriEngine2D.Audio.ISound.Name` `[get]`

Unique name of this sound object.

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

- `TriDevs.TriEngine2D/Audio/ISound.cs`

## 7.10 TriDevs.TriEngine2D.Logging.LogManager Class Reference

Class to manage logging. `ILog` interfaces should be obtained from this class' methods, as opposed to calling default `log4net` methods.

### Static Public Member Functions

- static void `LoadConfig` (string file=null)  
*Load a config to use with log4net.*
- static `ILog` `GetLogger` (object sender)  
*Gets an `ILog` object for the specified object.*
- static void `SetupConsole` ()  
*Set up a new console for this process. Will not set up a console if a debugger is attached. This method does nothing if `DEBUG` is not #defined.*
- static void `DestroyConsole` ()  
*Destroys the console associated with the process, if loaded. This method does nothing if `DEBUG` is not #defined.*
- static void `ClearOldLogs` (int daysOld=7, string logsDir="logs")  
*Clear logs that are older than the specified amount of days.*

#### 7.10.1 Detailed Description

Class to manage logging. `ILog` interfaces should be obtained from this class' methods, as opposed to calling default `log4net` methods.

#### 7.10.2 Member Function Documentation

##### 7.10.2.1 `static void TriDevs.TriEngine2D.Logging.LogManager.ClearOldLogs ( int daysOld = 7, string logsDir = "logs" )` `[static]`

Clear logs that are older than the specified amount of days.

#### Parameters

<code>daysOld</code>	Logs older than this amount of days will be deleted.
<code>logsDir</code>	The directory to clear.

```

136     {
137         var log = GetLogger(typeof(LogManager));
138     }

```



```

139         log.InfoFormat(">> ClearOldLogs({0}, \"{1}\")", daysOld, logsDir);
140
141         if (!Directory.Exists(logsDir))
142         {
143             log.InfoFormat("Directory {0} not found, no logs to clear", logsDir);
144             log.Info("<< ClearOldLogs()");
145             return;
146         }
147
148         var now = DateTime.Now;
149         var max = new TimeSpan(daysOld, 0, 0, 0);
150         var count = 0;
151         foreach (var file in from file in Directory.GetFiles(logsDir)
152                             let modTime = File.GetLastAccessTime(file)
153                             let age = now.Subtract(modTime)
154                             where age > max
155                             select file)
156         {
157             try
158             {
159                 File.Delete(file);
160                 log.InfoFormat("Deleted old log file: {0}", file);
161                 count++;
162             }
163             catch (IOException ex)
164             {
165                 log.WarnFormat("Failed to delete log file: {0} ({1})", file, ex.Message);
166             }
167         }
168
169         log.InfoFormat("Done! Cleared {0} log files.", count);
170         log.Info("<< ClearOldLogs()");
171     }

```

#### 7.10.2.2 static void TriDevs.TriEngine2D.Logging.LogManager.DestroyConsole ( ) [static]

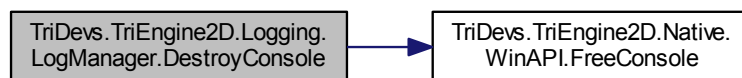
Destroys the console associated with the process, if loaded. This method does nothing if DEBUG is not #defined.

```

123     {
124     #if DEBUG
125         if (_consoleLoaded)
126             WinAPI.FreeConsole();
127     #endif
128     }

```

Here is the call graph for this function:



#### 7.10.2.3 static ILog TriDevs.TriEngine2D.Logging.LogManager.GetLogger ( object sender ) [static]

Gets an ILog object for the specified object.

To get the logger object for a static class, or from static context, call GetLogger(typeof(YourClass)).

##### Parameters

<i>sender</i>	The object or Type to get an ILog object for.
---------------	---

**Returns**

The ILog object.

```

89         {
90             if (!_loaded)
91                 LoadConfig();
92
93             return log4net.LogManager.GetLogger(sender.GetType().ToString() == "System.RuntimeType" ? (Type
94 )sender : sender.GetType());
95         }

```

**7.10.2.4 static void TriDevs.TriEngine2D.Logging.LogManager.LoadConfig ( string file = null ) [static]**

Load a config to use with log4net.

LoadConfig will first try to load the specified file, if not null. If it is unable to find the specified file, it will call itself again with file set to null. If no file is specified, it will attempt to load a config file following the pattern: "(Assembly-Name).config" If it is unable to load the config, it will default to BasicConfigurator.

**Parameters**

<i>file</i>	The config file to load, null if automatic loading is preferred.
-------------	--

```

57         {
58             if (file == null)
59             {
60                 if (File.Exists(AppDomain.CurrentDomain.FriendlyName + ".config"))
61                     XmlConfigurator.Configure();
62                 else
63                     BasicConfigurator.Configure();
64             }
65             else
66             {
67                 if (File.Exists(file))
68                     XmlConfigurator.Configure(new FileInfo(file));
69                 else
70                 {
71                     LoadConfig();
72                     return;
73                 }
74             }
75
76             _loaded = true;
77         }

```

**7.10.2.5 static void TriDevs.TriEngine2D.Logging.LogManager.SetupConsole ( ) [static]**

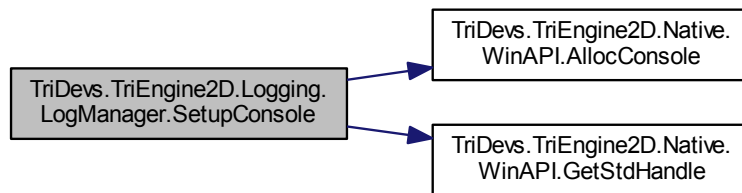
Set up a new console for this process. Will not set up a console if a debugger is attached. This method does nothing if DEBUG is not #defined.

```

102         {
103             #if DEBUG
104                 if (System.Diagnostics.Debugger.IsAttached)
105                     return;
106
107                 WinAPI.AllocConsole();
108                 var stdHandle = WinAPI.GetStdHandle(WinAPI.
STD_OUTPUT_HANDLE);
109                 var safeFileHandle = new SafeFileHandle(stdHandle, true);
110                 var fileStream = new FileStream(safeFileHandle, FileAccess.Write);
111                 var encoding = Encoding.GetEncoding(WinAPI.CODE_PAGE);
112                 var stdout = new StreamWriter(fileStream, encoding) { AutoFlush = true };
113                 Console.SetOut(stdout);
114                 _consoleLoaded = true;
115             #endif
116         }

```

Here is the call graph for this function:



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

- `TriDevs.TriEngine2D/Logging/LogManager.cs`

## 7.11 TriDevs.TriEngine2D.Helpers.Math Class Reference

Various helper methods for working with math.

### Static Public Member Functions

- static byte [Clamp](#) (byte value, byte min, byte max)  
*Clamps the specified byte between a minimum and maximum value.*
- static short [Clamp](#) (short value, short min, short max)  
*Clamps the specified short between a minimum and maximum value.*
- static ushort [Clamp](#) (ushort value, ushort min, ushort max)  
*Clamps the specified unsigned short between a minimum and maximum value.*
- static int [Clamp](#) (int value, int min, int max)  
*Clamps the specified integer between a minimum and maximum value.*
- static uint [Clamp](#) (uint value, uint min, uint max)  
*Clamps the specified unsigned integer between a minimum and maximum value.*
- static long [Clamp](#) (long value, long min, long max)  
*Clamps the specified 64-bit integer between a minimum and maximum value.*
- static ulong [Clamp](#) (ulong value, ulong min, ulong max)  
*Clamps the specified 64-bit unsigned integer between a minimum and maximum value.*
- static float [Clamp](#) (float value, float min, float max)  
*Clamps the specified float between a minimum and maximum value.*
- static double [Clamp](#) (double value, double min, double max)  
*Clamps the specified double between a minimum and maximum value.*

### 7.11.1 Detailed Description

Various helper methods for working with math.

## 7.11.2 Member Function Documentation

### 7.11.2.1 static byte TriDevs.TriEngine2D.Helpers.Math.Clamp ( byte *value*, byte *min*, byte *max* ) [static]

Clamps the specified byte between a minimum and maximum value.

#### Parameters

<i>value</i>	Value to clamp.
<i>min</i>	If the specified byte is below this value, then this will be returned.
<i>max</i>	If the specified byte is above this value, then this will be returned.

#### Returns

The clamped value of the byte.

```
39         {  
40             return value < min ? min : (value > max ? max : value);  
41         }
```

### 7.11.2.2 static short TriDevs.TriEngine2D.Helpers.Math.Clamp ( short *value*, short *min*, short *max* ) [static]

Clamps the specified short between a minimum and maximum value.

#### Parameters

<i>value</i>	Value to clamp.
<i>min</i>	If the specified short is below this value, then this will be returned.
<i>max</i>	If the specified short is above this value, then this will be returned.

#### Returns

The clamped value of the short.

```
51         {  
52             return value < min ? min : (value > max ? max : value);  
53         }
```

### 7.11.2.3 static ushort TriDevs.TriEngine2D.Helpers.Math.Clamp ( ushort *value*, ushort *min*, ushort *max* ) [static]

Clamps the specified unsigned short between a minimum and maximum value.

#### Parameters

<i>value</i>	Value to clamp.
<i>min</i>	If the specified unsigned short is below this value, then this will be returned.
<i>max</i>	If the specified unsigned short is above this value, then this will be returned.

#### Returns

The clamped value of the unsigned short.

```
63         {  
64             return value < min ? min : (value > max ? max : value);  
65         }
```

**7.11.2.4** `static int TriDevs.TriEngine2D.Helpers.Math.Clamp ( int value, int min, int max )` `[static]`

Clamps the specified integer between a minimum and maximum value.

**Parameters**

<i>value</i>	Value to clamp.
<i>min</i>	If the specified integer is below this value, then this will be returned.
<i>max</i>	If the specified integer is above this value, then this will be returned.

**Returns**

The clamped value of the integer.

```

75      {
76          return value < min ? min : (value > max ? max : value);
77      }
```

**7.11.2.5** `static uint TriDevs.TriEngine2D.Helpers.Math.Clamp ( uint value, uint min, uint max )` `[static]`

Clamps the specified unsigned integer between a minimum and maximum value.

**Parameters**

<i>value</i>	Value to clamp.
<i>min</i>	If the specified unsigned integer is below this value, then this will be returned.
<i>max</i>	If the specified unsigned integer is above this value, then this will be returned.

**Returns**

The clamped value of the unsigned integer.

```

87      {
88          return value < min ? min : (value > max ? max : value);
89      }
```

**7.11.2.6** `static long TriDevs.TriEngine2D.Helpers.Math.Clamp ( long value, long min, long max )` `[static]`

Clamps the specified 64-bit integer between a minimum and maximum value.

**Parameters**

<i>value</i>	Value to clamp.
<i>min</i>	If the specified 64-bit integer is below this value, then this will be returned.
<i>max</i>	If the specified 64-bit integer is above this value, then this will be returned.

**Returns**

The clamped value of the 64-bit integer.

```

99      {
100         return value < min ? min : (value > max ? max : value);
101     }
```

**7.11.2.7** `static ulong TriDevs.TriEngine2D.Helpers.Math.Clamp ( ulong value, ulong min, ulong max )` `[static]`

Clamps the specified 64-bit unsigned integer between a minimum and maximum value.

**Parameters**

<i>value</i>	Value to clamp.
<i>min</i>	If the specified 64-bit unsigned integer is below this value, then this will be returned.
<i>max</i>	If the specified 64-bit unsigned integer is above this value, then this will be returned.

**Returns**

The clamped value of the 64-bit unsigned integer.

```

111     {
112         return value < min ? min : (value > max ? max : value);
113     }
```

**7.11.2.8 static float TriDevs.TriEngine2D.Helpers.Math.Clamp ( float *value*, float *min*, float *max* ) [static]**

Clamps the specified float between a minimum and maximum value.

**Parameters**

<i>value</i>	Value to clamp.
<i>min</i>	If the specified float is below this value, then this will be returned.
<i>max</i>	If the specified float is above this value, then this will be returned.

**Returns**

The clamped value of the float.

```

123     {
124         return value < min ? min : (value > max ? max : value);
125     }
```

**7.11.2.9 static double TriDevs.TriEngine2D.Helpers.Math.Clamp ( double *value*, double *min*, double *max* ) [static]**

Clamps the specified double between a minimum and maximum value.

**Parameters**

<i>value</i>	Value to clamp.
<i>min</i>	If the specified double is below this value, then this will be returned.
<i>max</i>	If the specified double is above this value, then this will be returned.

**Returns**

The clamped value of the double.

```

135     {
136         return value < min ? min : (value > max ? max : value);
137     }
```

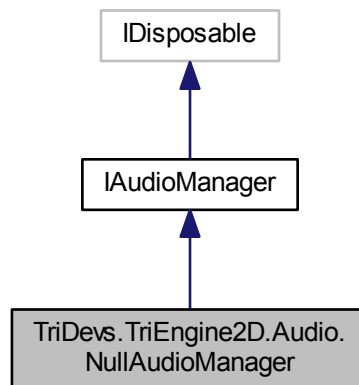
The documentation for this class was generated from the following file:

- TriDevs.TriEngine2D/Helpers/[Math.cs](#)

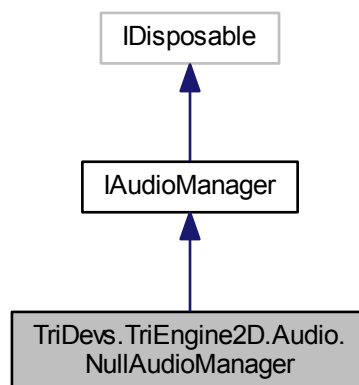
**7.12 TriDevs.TriEngine2D.Audio.NullAudioManager Class Reference**

Used as a fallback [AudioManager](#) object when the service locator fails to find one.

Inheritance diagram for TriDevs.TriEngine2D.Audio.NullAudioManager:



Collaboration diagram for TriDevs.TriEngine2D.Audio.NullAudioManager:



## Public Member Functions

- void [Dispose](#) ()
- void [StopAll](#) ()  
*Immediately stops all playback of sounds and songs.*
- [ISound LoadSound](#) (string name, string file, [AudioFormat](#) format=AudioFormat.Wav)  
*Loads a sound into the audio manager.*
- bool [HasSound](#) (string name)  
*Returns a value indicating whether a sound with the specified name has been loaded into the audio manager.*
- [ISound GetSound](#) (string name)  
*Gets the sound with the specified name.*

- void [StopAllSounds](#) ()  
*Immediately stops playback of all sounds.*
- [ISong LoadSong](#) (string name, string file, [AudioFormat](#) format=AudioFormat.Ogg)  
*Load a song into the audio manager.*
- bool [HasSong](#) (string name)  
*Returns a value indicating whether a song with the specified name has been loaded into the audio manager.*
- [ISong GetSong](#) (string name)  
*Gets the song with the specified name.*
- void [StopAllSongs](#) ()  
*Immediately stops playback of all songs.*

### 7.12.1 Detailed Description

Used as a fallback [AudioManager](#) object when the service locator fails to find one.

### 7.12.2 Member Function Documentation

#### 7.12.2.1 void [TriDevs.TriEngine2D.Audio.NullAudioManager.Dispose](#) ( )

```

35         {
36
37     }
```

#### 7.12.2.2 [ISong TriDevs.TriEngine2D.Audio.NullAudioManager.GetSong](#) ( string *name* )

Gets the song with the specified name.

##### Parameters

<i>name</i>	Name of the song to get.
-------------	--------------------------

##### Returns

An object implementing the [ISong](#) interface, or null if no song matched the requested name.

Implements [TriDevs.TriEngine2D.Audio.IAudioManager](#).

```

75         {
76             return Song;
77     }
```

#### 7.12.2.3 [ISound TriDevs.TriEngine2D.Audio.NullAudioManager.GetSound](#) ( string *name* )

Gets the sound with the specified name.

##### Parameters

<i>name</i>	Name of the sound to get.
-------------	---------------------------

##### Returns

An object implementing the [ISound](#) interface, or null if no sound matched the requested name.

Implements [TriDevs.TriEngine2D.Audio.IAudioManager](#).



```
55     {  
56         return Sound;  
57     }
```

#### 7.12.2.4 bool TriDevs.TriEngine2D.Audio.NullAudioManager.HasSong ( string *name* )

Returns a value indicating whether a song with the specified name has been loaded into the audio manager.

##### Parameters

<i>name</i>	Name to check for.
-------------	--------------------

##### Returns

True if the song has been loaded, false otherwise.

Implements [TriDevs.TriEngine2D.Audio.IAudioManager](#).

```
70     {  
71         return false;  
72     }
```

#### 7.12.2.5 bool TriDevs.TriEngine2D.Audio.NullAudioManager.HasSound ( string *name* )

Returns a value indicating whether a sound with the specified name has been loaded into the audio manager.

##### Parameters

<i>name</i>	Name to check for.
-------------	--------------------

##### Returns

True if the sound has been loaded, false otherwise.

Implements [TriDevs.TriEngine2D.Audio.IAudioManager](#).

```
50     {  
51         return false;  
52     }
```

#### 7.12.2.6 ISong TriDevs.TriEngine2D.Audio.NullAudioManager.LoadSong ( string *name*, string *file*, AudioFormat *format* = AudioFormat.Ogg )

Load a song into the audio manager.

If a song with the same name has already been loaded, it will be returned instead.

##### Parameters

<i>name</i>	Name to assign to the new song object.
<i>file</i>	Path to the file that will be used.
<i>format</i>	Format of the file.

**Returns**

The resulting object implementing the [ISong](#) interface.

Implements [TriDevs.TriEngine2D.Audio.IAudioManager](#).

```
65         {
66             return Song;
67         }
```

#### 7.12.2.7 **ISound** [TriDevs.TriEngine2D.Audio.NullAudioManager.LoadSound](#) ( string *name*, string *file*, **AudioFormat** *format* = **AudioFormat.Wav** )

Loads a sound into the audio manager.

If a sound with the same name has already been loaded, it will be returned instead.

**Parameters**

<i>name</i>	Name to assign to the new sound object.
<i>file</i>	Path to the file that will be used.
<i>format</i>	The format of the file to load.

**Returns**

The resulting object implementing the [ISound](#) interface.

Implements [TriDevs.TriEngine2D.Audio.IAudioManager](#).

```
45         {
46             return Sound;
47         }
```

#### 7.12.2.8 **void** [TriDevs.TriEngine2D.Audio.NullAudioManager.StopAll](#) ( )

Immediately stops all playback of sounds and songs.

Implements [TriDevs.TriEngine2D.Audio.IAudioManager](#).

```
40         {
41
42         }
```

#### 7.12.2.9 **void** [TriDevs.TriEngine2D.Audio.NullAudioManager.StopAllSongs](#) ( )

Immediately stops playback of all songs.

Implements [TriDevs.TriEngine2D.Audio.IAudioManager](#).

```
80         {
81
82         }
```

#### 7.12.2.10 **void** [TriDevs.TriEngine2D.Audio.NullAudioManager.StopAllSounds](#) ( )

Immediately stops playback of all sounds.

Implements [TriDevs.TriEngine2D.Audio.IAudioManager](#).

```
60     {  
61  
62     }
```

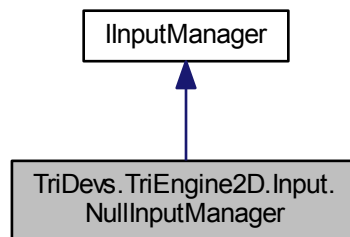
The documentation for this class was generated from the following file:

- TriDevs.TriEngine2D/Audio/[NullAudioManager.cs](#)

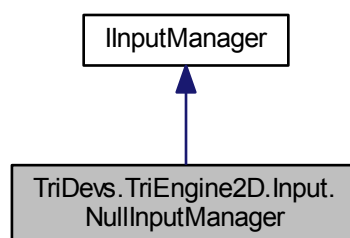
## 7.13 TriDevs.TriEngine2D.Input.NullInputManager Class Reference

Used as a fallback [InputManager](#) object when the service locator fails to find one.

Inheritance diagram for TriDevs.TriEngine2D.Input.NullInputManager:



Collaboration diagram for TriDevs.TriEngine2D.Input.NullInputManager:



### Public Member Functions

- void [Update](#) ()  
*Updates the input manager, refreshing all current and previous states.*
- bool [KeyUp](#) (Key key)  
*Returns whether or not the specified key is currently unpressed.*
- bool [KeyDown](#) (Key key)

- Returns whether or not the specified key is currently being pressed.*
- bool [KeyPressed](#) (Key key)  
*Returns whether or not the specified key has been pressed.*
- bool [KeyReleased](#) (Key key)  
*Returns whether or not the specified key has been released.*
- bool [ButtonUp](#) (MouseButton button)  
*Returns whether or not the specified mouse button is currently unpressed.*
- bool [ButtonDown](#) (MouseButton button)  
*Returns whether or not the specified mouse button is currently being pressed.*
- bool [ButtonPressed](#) (MouseButton button)  
*Returns whether or not the specified mouse button has been pressed.*
- bool [ButtonReleased](#) (MouseButton button)  
*Returns whether or not the specified mouse button has been released.*
- bool [WheelUp](#) ()  
*Returns whether the mouse wheel was scrolled up.*
- bool [WheelDown](#) ()  
*Returns whether the mouse wheel was scrolled down.*
- bool [WheelChanged](#) ()  
*Returns whether the mouse wheel scrolled at all.*
- int [WheelChange](#) ()  
*Returns the mouse wheel's change in value.*

## Properties

- int [MouseX](#) [get]
- int [MouseY](#) [get]
- [Point](#) [MousePosition](#) [get]
- int [MouseWheelValue](#) [get]
- bool [this\[Key key\]](#) [get]
- bool [this\[MouseButton button\]](#) [get]

### 7.13.1 Detailed Description

Used as a fallback [InputManager](#) object when the service locator fails to find one.

### 7.13.2 Member Function Documentation

#### 7.13.2.1 bool [TriDevs.TriEngine2D.Input.NullInputManager.ButtonDown](#) ( [MouseButton](#) *button* )

Returns whether or not the specified mouse button is currently being pressed.

#### Parameters

<i>button</i>	The button to query for.
---------------	--------------------------

#### Returns

True if button is currently being pressed, false otherwise.

Implements [TriDevs.TriEngine2D.Input.IInputManager](#).

```
80         return false;
81     }
```

#### 7.13.2.2 bool TriDevs.TriEngine2D.Input.NullInputManager.ButtonPressed ( MouseButton *button* )

Returns whether or not the specified mouse button has been pressed.

Only returns true if the last state of the mouse button was not pressed.

##### Parameters

<i>button</i>	Button to query for.
---------------	----------------------

##### Returns

True if button was pressed, false otherwise.

Implements [TriDevs.TriEngine2D.Input.IInputManager](#).

```
84     {
85         return false;
86     }
```

#### 7.13.2.3 bool TriDevs.TriEngine2D.Input.NullInputManager.ButtonReleased ( MouseButton *button* )

Returns whether or not the specified mouse button has been released.

Only returns true if the last state of the button was pressed.

##### Parameters

<i>button</i>	The button to query for.
---------------	--------------------------

##### Returns

True if the button was released, false otherwise.

Implements [TriDevs.TriEngine2D.Input.IInputManager](#).

```
89     {
90         return false;
91     }
```

#### 7.13.2.4 bool TriDevs.TriEngine2D.Input.NullInputManager.ButtonUp ( MouseButton *button* )

Returns whether or not the specified mouse button is currently unpressed.

##### Parameters

<i>button</i>	Button to query for.
---------------	----------------------

##### Returns

True if the button is currently up (not pressed), false otherwise.

Implements [TriDevs.TriEngine2D.Input.IInputManager](#).

```

74         {
75             return true;
76         }

```

#### 7.13.2.5 bool TriDevs.TriEngine2D.Input.NullInputManager.KeyDown ( Key key )

Returns whether or not the specified key is currently being pressed.

##### Parameters

<i>key</i>	Key to query for.
------------	-------------------

##### Returns

True if key is currently being pressed, false otherwise.

Implements [TriDevs.TriEngine2D.Input.IInputManager](#).

```

59         {
60             return false;
61         }

```

#### 7.13.2.6 bool TriDevs.TriEngine2D.Input.NullInputManager.KeyPressed ( Key key )

Returns whether or not the specified key has been pressed.

Only returns true if the last state of the key was not pressed.

##### Parameters

<i>key</i>	Key to query for.
------------	-------------------

##### Returns

True if key was pressed, false otherwise.

Implements [TriDevs.TriEngine2D.Input.IInputManager](#).

```

64         {
65             return false;
66         }

```

#### 7.13.2.7 bool TriDevs.TriEngine2D.Input.NullInputManager.KeyReleased ( Key key )

Returns whether or not the specified key has been released.

Only returns true if the last state of the key was pressed.

##### Parameters

<i>key</i>	Key to query for.
------------	-------------------

##### Returns

True if key was released, false otherwise.

Implements [TriDevs.TriEngine2D.Input.IInputManager](#).

```

69         {
70             return false;
71         }

```

#### 7.13.2.8 bool TriDevs.TriEngine2D.Input.NullInputManager.KeyUp ( Key key )

Returns whether or not the specified key is currently unpressed.

##### Parameters

<i>key</i>	Key to query for.
------------	-------------------

##### Returns

True if the key is currently up (not pressed), false otherwise.

Implements [TriDevs.TriEngine2D.Input.IInputManager](#).

```

54         {
55             return true;
56         }

```

#### 7.13.2.9 void TriDevs.TriEngine2D.Input.NullInputManager.Update ( )

Updates the input manager, refreshing all current and previous states.

Implements [TriDevs.TriEngine2D.Input.IInputManager](#).

```

49         {
50             // Do nothing
51         }

```

#### 7.13.2.10 int TriDevs.TriEngine2D.Input.NullInputManager.WheelChange ( )

Returns the mouse wheel's change in value.

##### Returns

Negative value if wheel scrolled down, positive value if scrolled up, zero if not scrolled.

Implements [TriDevs.TriEngine2D.Input.IInputManager](#).

```

109         {
110             return 0;
111         }

```

#### 7.13.2.11 bool TriDevs.TriEngine2D.Input.NullInputManager.WheelChanged ( )

Returns whether the mouse wheel scrolled at all.

##### Returns

True if the mouse wheel scrolled, false otherwise.

Implements [TriDevs.TriEngine2D.Input.IInputManager](#).

```

104         {
105             return false;
106         }

```

#### 7.13.2.12 bool TriDevs.TriEngine2D.Input.NullInputManager.WheelDown ( )

Returns whether the mouse wheel was scrolled down.

##### Returns

True if mouse wheel was scrolled down, false otherwise.

Implements [TriDevs.TriEngine2D.Input.IInputManager](#).

```
99      {
100          return false;
101      }
```

#### 7.13.2.13 bool TriDevs.TriEngine2D.Input.NullInputManager.WheelUp ( )

Returns whether the mouse wheel was scrolled up.

##### Returns

True if mouse wheel was scrolled up, false otherwise.

Implements [TriDevs.TriEngine2D.Input.IInputManager](#).

```
94      {
95          return false;
96      }
```

### 7.13.3 Property Documentation

7.13.3.1 Point TriDevs.TriEngine2D.Input.NullInputManager.MousePosition [get]

7.13.3.2 int TriDevs.TriEngine2D.Input.NullInputManager.MouseWheelValue [get]

7.13.3.3 int TriDevs.TriEngine2D.Input.NullInputManager.MouseX [get]

7.13.3.4 int TriDevs.TriEngine2D.Input.NullInputManager.MouseY [get]

7.13.3.5 bool TriDevs.TriEngine2D.Input.NullInputManager.this[Key key] [get]

7.13.3.6 bool TriDevs.TriEngine2D.Input.NullInputManager.this[MouseButton button] [get]

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

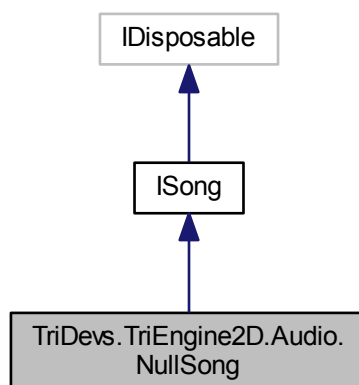
- [TriDevs.TriEngine2D/Input/NullInputManager.cs](#)

## 7.14 TriDevs.TriEngine2D.Audio.NullSong Class Reference

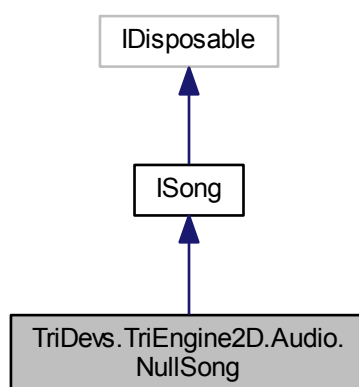
Fallback song class used in [NullAudioManager](#).



Inheritance diagram for TriDevs.TriEngine2D.Audio.NullSong:



Collaboration diagram for TriDevs.TriEngine2D.Audio.NullSong:



### Public Member Functions

- void [Dispose](#) ()
- void [Play](#) ()  
*Starts playback of the song.*
- void [Stop](#) ()  
*Stops playback of the song.*
- void [Pause](#) ()  
*Pauses playback of the song.*
- void [Resume](#) ()  
*Resumes playback of a paused song.*

## Properties

- string [Name](#) [get]
- string [File](#) [get]
- float [Volume](#) [get, set]
- bool [IsLooped](#) [get, set]

### 7.14.1 Detailed Description

Fallback song class used in [NullAudioManager](#).

### 7.14.2 Member Function Documentation

#### 7.14.2.1 void TriDevs.TriEngine2D.Audio.NullSong.Dispose ( )

```
37         {  
38  
39     }
```

#### 7.14.2.2 void TriDevs.TriEngine2D.Audio.NullSong.Pause ( )

Pauses playback of the song.

Call the [Resume](#) or [Play](#) method to resume playback.

Implements [TriDevs.TriEngine2D.Audio.ISong](#).

```
52         {  
53  
54     }
```

#### 7.14.2.3 void TriDevs.TriEngine2D.Audio.NullSong.Play ( )

Starts playback of the song.

Implements [TriDevs.TriEngine2D.Audio.ISong](#).

```
42         {  
43  
44     }
```

#### 7.14.2.4 void TriDevs.TriEngine2D.Audio.NullSong.Resume ( )

Resumes playback of a paused song.

Implements [TriDevs.TriEngine2D.Audio.ISong](#).

```
57         {  
58  
59     }
```

#### 7.14.2.5 void TriDevs.TriEngine2D.Audio.NullSong.Stop ( )

Stops playback of the song.

Implements [TriDevs.TriEngine2D.Audio.ISong](#).

```
47         {  
48  
49     }
```

### 7.14.3 Property Documentation

7.14.3.1 `string TriDevs.TriEngine2D.Audio.NullSong.File` `[get]`

7.14.3.2 `bool TriDevs.TriEngine2D.Audio.NullSong.IsLooped` `[get]`, `[set]`

7.14.3.3 `string TriDevs.TriEngine2D.Audio.NullSong.Name` `[get]`

7.14.3.4 `float TriDevs.TriEngine2D.Audio.NullSong.Volume` `[get]`, `[set]`

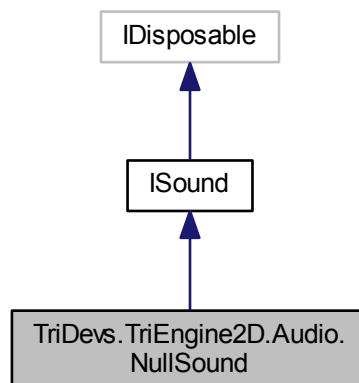
The documentation for this class was generated from the following file:

- `TriDevs.TriEngine2D/Audio/NullSong.cs`

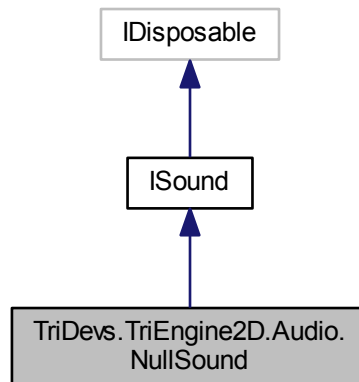
## 7.15 TriDevs.TriEngine2D.Audio.NullSound Class Reference

Fallback sound class used in [NullAudioManager](#).

Inheritance diagram for `TriDevs.TriEngine2D.Audio.NullSound`:



Collaboration diagram for TriDevs.TriEngine2D.Audio.NullSound:



## Public Member Functions

- void [Play](#) ()  
*Plays the sound.*
- void [Stop](#) ()  
*Stops the sound.*
- void [Dispose](#) ()

## Properties

- string [Name](#) [get]
- string [File](#) [get]

### 7.15.1 Detailed Description

Fallback sound class used in [NullAudioManager](#).

### 7.15.2 Member Function Documentation

#### 7.15.2.1 void TriDevs.TriEngine2D.Audio.NullSound.Dispose ( )

```
45     {  
46  
47     }
```

#### 7.15.2.2 void TriDevs.TriEngine2D.Audio.NullSound.Play ( )

Plays the sound.

Implements [TriDevs.TriEngine2D.Audio.ISound](#).

```

35         {
36
37     }

```

### 7.15.2.3 void TriDevs.TriEngine2D.Audio.NullSound.Stop ( )

Stops the sound.

Implements [TriDevs.TriEngine2D.Audio.ISound](#).

```

40         {
41
42     }

```

## 7.15.3 Property Documentation

**7.15.3.1** string TriDevs.TriEngine2D.Audio.NullSound.File [get]

**7.15.3.2** string TriDevs.TriEngine2D.Audio.NullSound.Name [get]

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

- TriDevs.TriEngine2D/Audio/[NullSound.cs](#)

## 7.16 TriDevs.TriEngine2D.Point Struct Reference

A struct representing an X/Y coordinate.

### Public Member Functions

- [Point](#) (int x, int y)  
*Creates a new [Point](#) with the specified X and Y values.*

### Public Attributes

- int [X](#)  
*The X value of the coordinate.*
- int [Y](#)  
*The Y value of the coordinate.*

### 7.16.1 Detailed Description

A struct representing an X/Y coordinate.

### 7.16.2 Constructor & Destructor Documentation

**7.16.2.1** TriDevs.TriEngine2D.Point.Point ( int x, int y )

Creates a new [Point](#) with the specified X and Y values.

#### Parameters

x	The X value.
y	The Y value.

```

47     {
48         X = x;
49         Y = y;
50     }

```

### 7.16.3 Member Data Documentation

#### 7.16.3.1 int TriDevs.TriEngine2D.Point.X

The X value of the coordinate.

#### 7.16.3.2 int TriDevs.TriEngine2D.Point.Y

The Y value of the coordinate.

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

- TriDevs.TriEngine2D/[Point.cs](#)

## 7.17 TriDevs.TriEngine2D.Serializing.Serializer Class Reference

Provides serialization methods.

### Static Public Member Functions

- static string [Serialize< T >](#) (T data)  
*Serialize an object to string.*
- static void [Serialize< T >](#) (T data, string file, Formatting formatting=Formatting.Indented)  
*Serializes an object to file.*
- static T [Deserialize< T >](#) (string file)  
*Deserialize a serialized object from file.*

#### 7.17.1 Detailed Description

Provides serialization methods.

#### 7.17.2 Member Function Documentation

##### 7.17.2.1 static T TriDevs.TriEngine2D.Serializing.Serializer.Deserialize< T > ( string file ) [static]

Deserialize a serialized object from file.

#### Template Parameters

T	Type of the object being deserialized.
---	--

## Parameters

<i>file</i>	File to read from.
-------------	--------------------

## Returns

The deserialized object.

```

84     {
85         T data;
86         // Create the StreamReader
87         using (var reader = new StreamReader(file))
88         {
89             // And the json reader
90             using (var jsonReader = new JsonTextReader(reader))
91             {
92                 // Now deserialize the file to the requested object...
93                 data = JsonSerializer.Value.Deserialize<T>(jsonReader);
94                 // ... and close the json reader.
95                 jsonReader.Close();
96             }
97             // Finally, close the file reader
98             reader.Close();
99         }
100         return data;
101     }

```

## 7.17.2.2 static string TriDevs.TriEngine2D.Serializing.Serializer.Serialize&lt; T &gt; ( T data ) [static]

Serialize an object to string.

## Template Parameters

<i>T</i>	Type of data.
----------	---------------

## Parameters

<i>data</i>	Data to serialize.
-------------	--------------------

## Returns

The serialized object in string format.

```

44     {
45         // Create a StringWriter to hold the serialized object
46         var writer = new StringWriter();
47         // Serialize the object into the writer
48         JsonSerializer.Value.Serialize(writer, data);
49         return writer.ToString(); // Return the data as string
50     }

```

## 7.17.2.3 static void TriDevs.TriEngine2D.Serializing.Serializer.Serialize&lt; T &gt; ( T data, string file, Formatting formatting = Formatting.Indented ) [static]

Serializes an object to file.

## Template Parameters

<i>T</i>	Type of the data.
----------	-------------------

## Parameters

<i>data</i>	Data to serialize.
<i>file</i>	File to serialize to.

<i>formatting</i>	The formatting to use for the JSON output.
-------------------	--

```

60     {
61         // Create the StreamWriter
62         using (var writer = new StreamWriter(file, false))
63         {
64             // Create the json writer
65             using (var jsonWriter = new JsonTextWriter(writer){Formatting = formatting})
66             {
67                 // Now serialize the object to the file...
68                 JsonSerializer.Value.Serialize(jsonWriter, data);
69                 // ... and close the json writer
70                 jsonWriter.Close();
71             }
72             // Finally, close the file writer
73             writer.Close();
74         }
75     }

```

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

- TriDevs.TriEngine2D/Serializing/[Serializer.cs](#)

## 7.18 TriDevs.TriEngine2D.Services Class Reference

Provides different game-related service interfaces.

### Static Public Member Functions

- static void [Provide](#) ([IInputManager](#) input)  
*Specifies an input manager service to provide.*
- static void [Provide](#) ([IAudioManager](#) audio)  
*Specifies an audio manager service to provide.*
- static void [Provide](#) ([IInputManager](#) input, [IAudioManager](#) audio)  
*Specifies what services to provide.*

### Properties

- static [IInputManager](#) [Input](#) [get]  
*The input manager service.*
- static [IAudioManager](#) [Audio](#) [get]  
*The audio manager service.*

#### 7.18.1 Detailed Description

Provides different game-related service interfaces.

Actual service providers must be supplied from external code. All Service properties are initialized with Null-type services that provide no real functionality.

#### 7.18.2 Member Function Documentation

##### 7.18.2.1 static void TriDevs.TriEngine2D.Services.Provide ( IInputManager input ) [static]

Specifies an input manager service to provide.



## Parameters

<i>input</i>	An object implementing the <code>IInputManager</code> interface.
--------------	--

```

57         {
58             _input = input;
59         }

```

### 7.18.2.2 static void TriDevs.TriEngine2D.Services.Provide ( `IAudioManager audio` ) [static]

Specifies an audio manager service to provide.

## Parameters

<i>audio</i>	An object implementing the <code>IAudioManager</code> interface.
--------------	--

```

66         {
67             _audio = audio;
68         }

```

### 7.18.2.3 static void TriDevs.TriEngine2D.Services.Provide ( `IInputManager input`, `IAudioManager audio` ) [static]

Specifies what services to provide.

## Parameters

<i>input</i>	The input service to provide.
<i>audio</i>	The audio service to provide.

```

76         {
77             Provide(input);
78             Provide(audio);
79         }

```

## 7.18.3 Property Documentation

### 7.18.3.1 `IAudioManager TriDevs.TriEngine2D.Services.Audio` [static],[get]

The audio manager service.

### 7.18.3.2 `IInputManager TriDevs.TriEngine2D.Services.Input` [static],[get]

The input manager service.

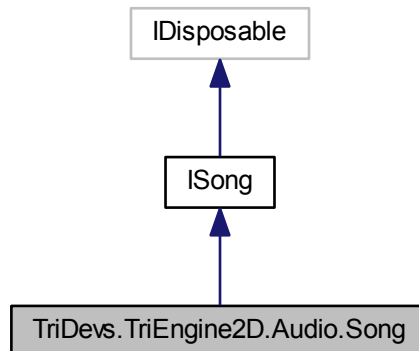
The documentation for this class was generated from the following file:

- TriDevs.TriEngine2D/[Services.cs](#)

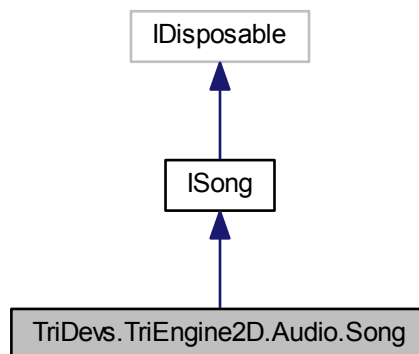
## 7.19 TriDevs.TriEngine2D.Audio.Song Class Reference

[Song](#) class that can be used with [AudioManager](#).

Inheritance diagram for TriDevs.TriEngine2D.Audio.Song:



Collaboration diagram for TriDevs.TriEngine2D.Audio.Song:



## Public Member Functions

- void [Dispose](#) ()
- void [Play](#) ()  
*Starts playback of the song.*
- void [Stop](#) ()  
*Stops playback of the song.*
- void [Pause](#) ()  
*Pauses playback of the song.*
- void [Resume](#) ()  
*Resumes playback of a paused song.*

## Properties

- string [Name](#) [get]
- string [File](#) [get]
- float [Volume](#) [get, set]
- bool [IsLooped](#) [get, set]

### 7.19.1 Detailed Description

[Song](#) class that can be used with [AudioManager](#).

The song class will stream from a specified file, instead of loading it into memory.

### 7.19.2 Member Function Documentation

#### 7.19.2.1 void TriDevs.TriEngine2D.Audio.Song.Dispose ( )

```
80         {
81             if (_stream == null)
82                 return;
83
84             Stop();
85             _stream.Dispose();
86             _stream = null;
87         }
```

#### 7.19.2.2 void TriDevs.TriEngine2D.Audio.Song.Pause ( )

Pauses playback of the song.

Call the [Resume](#) or [Play](#) method to resume playback.

Implements [TriDevs.TriEngine2D.Audio.ISong](#).

```
108         {
109             _stream.Pause();
110         }
```

#### 7.19.2.3 void TriDevs.TriEngine2D.Audio.Song.Play ( )

Starts playback of the song.

Implements [TriDevs.TriEngine2D.Audio.ISong](#).

```
90         {
91             try
92             {
93                 _stream.Play();
94             }
95             catch (InvalidDataException)
96             {
97                 Stop();
98                 Play();
99             }
100         }
```

#### 7.19.2.4 void TriDevs.TriEngine2D.Audio.Song.Resume ( )

Resumes playback of a paused song.

Implements [TriDevs.TriEngine2D.Audio.ISong](#).

```

113     {
114         _stream.Resume();
115     }

```

#### 7.19.2.5 void TriDevs.TriEngine2D.Audio.Song.Stop ( )

Stops playback of the song.

Implements [TriDevs.TriEngine2D.Audio.ISong](#).

```

103     {
104         _stream.Stop();
105     }

```

### 7.19.3 Property Documentation

7.19.3.1 string TriDevs.TriEngine2D.Audio.Song.File [get]

7.19.3.2 bool TriDevs.TriEngine2D.Audio.Song.IsLooped [get], [set]

7.19.3.3 string TriDevs.TriEngine2D.Audio.Song.Name [get]

7.19.3.4 float TriDevs.TriEngine2D.Audio.Song.Volume [get], [set]

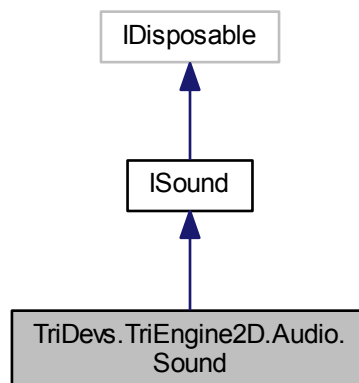
The documentation for this class was generated from the following file:

- TriDevs.TriEngine2D/Audio/[Song.cs](#)

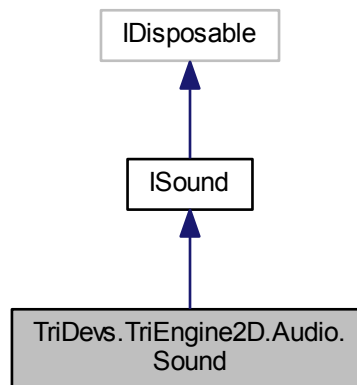
## 7.20 TriDevs.TriEngine2D.Audio.Sound Class Reference

[Sound](#) class that can be used with the [AudioManager](#).

Inheritance diagram for TriDevs.TriEngine2D.Audio.Sound:



Collaboration diagram for TriDevs.TriEngine2D.Audio.Sound:



## Public Member Functions

- void [Play](#) ()  
*Plays the sound.*
- void [Stop](#) ()  
*Stops the sound.*
- void [Dispose](#) ()

## Properties

- string [Name](#) [get]
- string [File](#) [get]

### 7.20.1 Detailed Description

[Sound](#) class that can be used with the [AudioManager](#).

### 7.20.2 Member Function Documentation

#### 7.20.2.1 void TriDevs.TriEngine2D.Audio.Sound.Dispose ( )

```

187         {
188             Stop();
189             AL.DeleteSources(_sources);
190             AL.DeleteBuffer(_buffer);
191         }
  
```

#### 7.20.2.2 void TriDevs.TriEngine2D.Audio.Sound.Play ( )

Plays the sound.

Implements [TriDevs.TriEngine2D.Audio.ISound](#).

```

169         {
170             for (var i = 0; i < SourceCount; i++)
171             {
172                 if (GetSourceState(_sources[i]) != ALSourceState.Playing)
173                 {
174                     AL.SourcePlay(_sources[i]);
175                     return;
176                 }
177             }
178         }

```

### 7.20.2.3 void TriDevs.TriEngine2D.Audio.Sound.Stop ( )

Stops the sound.

Implements [TriDevs.TriEngine2D.Audio.ISound](#).

```

181         {
182             for (var i = 0; i < SourceCount; i++)
183                 AL.SourceStop(_sources[i]);
184         }

```

## 7.20.3 Property Documentation

7.20.3.1 string TriDevs.TriEngine2D.Audio.Sound.File [get]

7.20.3.2 string TriDevs.TriEngine2D.Audio.Sound.Name [get]

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

- TriDevs.TriEngine2D/Audio/[Sound.cs](#)

## 7.21 TriDevs.TriEngine2D.Extensions.StringExtensions Class Reference

[Extensions](#) for System.String

### Static Public Member Functions

- static string [ReplaceFirst](#) (this string s, string search, string replace, bool caseInsensitive=false)  
*Returns a string in which the first occurrence of a specified string is replaced with another string.*
- static string [Replace](#) (this string s, string search, string replace, int count, bool caseInsensitive=false)  
*Returns a string in which the N first occurrences of a specified string are replaced with another string.*
- static string [Replace](#) (this string s, string search, string replace, bool caseInsensitive=false)  
*Returns a string in which all occurrences of a specified string are replaced with another string.*

### 7.21.1 Detailed Description

[Extensions](#) for System.String

### 7.21.2 Member Function Documentation

7.21.2.1 static string TriDevs.TriEngine2D.Extensions.StringExtensions.Replace ( this string s, string search, string replace, int count, bool caseInsensitive = false ) [static]

Returns a string in which the N first occurrences of a specified string are replaced with another string.

## Parameters

<i>s</i>	String to modify.
<i>search</i>	String to search for.
<i>replace</i>	String to replace the match(es) with.
<i>count</i>	Number of occurrences to replace.
<i>caseInsensitive</i>	True for case insensitive search, false for case sensitive.

## Returns

The supplied string with the N first occurrences of the specified string replaced with the other.

```

56      {
57          var re = caseInsensitive ? new Regex(search, RegexOptions.IgnoreCase) : new Regex(search);
58          return re.Replace(s, replace, count);
59      }

```

**7.21.2.2** `static string TriDevs.TriEngine2D.Extensions.StringExtensions.Replace ( this string s, string search, string replace, bool caseInsensitive = false ) [static]`

Returns a string in which all occurrences of a specified string are replaced with another string.

This extension method supports case insensitive searches.

## Parameters

<i>s</i>	String to modify.
<i>search</i>	String to search for.
<i>replace</i>	String to replace the match(es) with.
<i>caseInsensitive</i>	True for case insensitive search, false for case sensitive.

## Returns

The supplied string with all occurrences of the specified string replaced with the other.

```

73      {
74          var re = caseInsensitive ? new Regex(search, RegexOptions.IgnoreCase) : new Regex(search);
75          return re.Replace(s, replace);
76      }

```

**7.21.2.3** `static string TriDevs.TriEngine2D.Extensions.StringExtensions.ReplaceFirst ( this string s, string search, string replace, bool caseInsensitive = false ) [static]`

Returns a string in which the first occurrence of a specified string is replaced with another string.

## Parameters

<i>s</i>	String to modify.
<i>search</i>	String to search for.
<i>replace</i>	String to replace the match with.
<i>caseInsensitive</i>	True for case insensitive search, false for case sensitive.

## Returns

The supplied string with the first occurrence of the specified string replaced with the other.

```

42      {
43          return Replace(s, search, replace, 1, caseInsensitive);
44      }

```

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

- TriDevs.TriEngine2D/Extensions/[StringExtensions.cs](#)

## 7.22 TriDevs.TriEngine2D.Helpers.Threading Class Reference

Provides various helper functions for doing threading operations.

### Static Public Member Functions

- static void [SetCurrentThreadName](#) (string name)  
*Sets the name of the current thread, does nothing if the thread already has a name.*

#### 7.22.1 Detailed Description

Provides various helper functions for doing threading operations.

#### 7.22.2 Member Function Documentation

##### 7.22.2.1 static void TriDevs.TriEngine2D.Helpers.Threading.SetCurrentThreadName ( string name ) [static]

Sets the name of the current thread, does nothing if the thread already has a name.

#### Parameters

<i>name</i>	The new name for the current thread
-------------	-------------------------------------

```

39         {
40             // We can't set the name on a thread if it's already set, it would throw an exception
41             // So we have to check if the current name is null before trying to set a new one
42             if (string.IsNullOrEmpty(Thread.CurrentThread.Name))
43                 Thread.CurrentThread.Name = name;
44         }

```

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

- TriDevs.TriEngine2D/Helpers/[Threading.cs](#)

## 7.23 TriDevs.TriEngine2D.Version Class Reference

[Version](#) class specifying the version of this project.

### Public Attributes

- const int [Major](#) = 0  
*Major version of the project.*
- const int [Minor](#) = 0  
*Minor version of the project.*
- const int [Patch](#) = 8  
*Patch version of the project.*
- const string [Suffix](#) = ""  
*Optional suffix, empty if no suffix for this version.*



- const string [VersionStringFormat](#) = "{0}.{1}.{2}"  
*The format string used when formatting major, minor and patch version to their string representation.*
- const string [VersionStringFormatWithSuffix](#) = [VersionStringFormat](#) + "-{3}"  
*The format string used when formatting major, minor and patch version to their string representation (with suffix).*

## Properties

- static string [VersionString](#) [get]  
*String representation of the current project version.*

### 7.23.1 Detailed Description

[Version](#) class specifying the version of this project.

### 7.23.2 Member Data Documentation

#### 7.23.2.1 const int TriDevs.TriEngine2D.Version.Major = 0

Major version of the project.

#### 7.23.2.2 const int TriDevs.TriEngine2D.Version.Minor = 0

Minor version of the project.

#### 7.23.2.3 const int TriDevs.TriEngine2D.Version.Patch = 8

Patch version of the project.

#### 7.23.2.4 const string TriDevs.TriEngine2D.Version.Suffix = ""

Optional suffix, empty if no suffix for this version.

Example values could be "beta" and "alpha".

#### 7.23.2.5 const string TriDevs.TriEngine2D.Version.VersionStringFormat = "{0}.{1}.{2}"

The format string used when formatting major, minor and patch version to their string representation.

#### 7.23.2.6 const string TriDevs.TriEngine2D.Version.VersionStringFormatWithSuffix = VersionStringFormat + "-{3}"

The format string used when formatting major, minor and patch version to their string representation (with suffix).

### 7.23.3 Property Documentation

#### 7.23.3.1 string TriDevs.TriEngine2D.Version.VersionString [static], [get]

String representation of the current project version.

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

- TriDevs.TriEngine2D/[Version.cs](#)

## 7.24 TriDevs.TriEngine2D.Native.WinAPI Class Reference

Holds various [WinAPI](#) stuff.

### Public Member Functions

- static IntPtr [GetStdHandle](#) (int nStdHandle)  
*Retrieves a handle to the specified standard device (standard input, standard output, or standard error).*
- static bool [AllocConsole](#) ()  
*Allocates a new console for the calling process.*
- static int [FreeConsole](#) ()  
*Detaches the calling process from its console.*

### Public Attributes

- const int [STD\\_OUTPUT\\_HANDLE](#) = -11  
*The standard output device. Initially, this is the active console screen buffer, CONOUT\$.*
- const int [CODE\\_PAGE](#) = 437  
*The code page to use for the console.*

### 7.24.1 Detailed Description

Holds various [WinAPI](#) stuff.

### 7.24.2 Member Function Documentation

#### 7.24.2.1 static bool TriDevs.TriEngine2D.Native.WinAPI.AllocConsole ( )

Allocates a new console for the calling process.

#### Returns

If the function succeeds, the return value is nonzero. If the function fails, the return value is zero. To get extended error information, call GetLastError.

Here is the caller graph for this function:



#### 7.24.2.2 static int TriDevs.TriEngine2D.Native.WinAPI.FreeConsole ( )

Detaches the calling process from its console.

**Returns**

If the function succeeds, the return value is nonzero. If the function fails, the return value is zero. To get extended error information, call GetLastError.

Here is the caller graph for this function:

**7.24.2.3 static IntPtr TriDevs.TriEngine2D.Native.WinAPI.GetStdHandle ( int nStdHandle )**

Retrieves a handle to the specified standard device (standard input, standard output, or standard error).

**Parameters**

<i>nStdHandle</i>	The standard device.
-------------------	----------------------

**Returns**

If the function succeeds, the return value is a handle to the specified device, or a redirected handle set by a previous call to SetStdHandle. The handle has GENERIC\_READ and GENERIC\_WRITE access rights, unless the application has used SetStdHandle to set a standard handle with lesser access. If the function fails, the return value is INVALID\_HANDLE\_VALUE. To get extended error information, call GetLastError. If an application does not have associated standard handles, such as a service running on an interactive desktop, and has not redirected them, the return value is NULL.

Here is the caller graph for this function:

**7.24.3 Member Data Documentation****7.24.3.1 const int TriDevs.TriEngine2D.Native.WinAPI.CODE\_PAGE = 437**

The code page to use for the console.

**7.24.3.2 const int TriDevs.TriEngine2D.Native.WinAPI.STD\_OUTPUT\_HANDLE = -11**

The standard output device. Initially, this is the active console screen buffer, CONOUT\$.

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

- [TriDevs.TriEngine2D/Native/WinAPI.cs](#)

## Chapter 8

# File Documentation

### 8.1 README.md File Reference

### 8.2 TriDevs.TriEngine2D/Audio/AudioFormat.cs File Reference

#### Namespaces

- package [TriDevs.TriEngine2D.Audio](#)

#### Enumerations

- enum [TriDevs.TriEngine2D.Audio.AudioFormat](#) { [TriDevs.TriEngine2D.Audio.Ogg](#), [TriDevs.TriEngine2D.Audio.Wav](#) }

*Different audio formats usable with audio managers.*

### 8.3 TriDevs.TriEngine2D/Audio/AudioManager.cs File Reference

#### Classes

- class [TriDevs.TriEngine2D.Audio.AudioManager](#)

*Class to manage engine audio.*

#### Namespaces

- package [TriDevs.TriEngine2D.Audio](#)

### 8.4 TriDevs.TriEngine2D/Audio/IAudioManager.cs File Reference

#### Classes

- interface [TriDevs.TriEngine2D.Audio.IAudioManager](#)

*Provides various methods to manipulate audio.*

## Namespaces

- package [TriDevs.TriEngine2D.Audio](#)

## 8.5 TriDevs.TriEngine2D/Audio/ISong.cs File Reference

### Classes

- interface [TriDevs.TriEngine2D.Audio.ISong](#)  
*A song that will be streamed in the audio player.*

## Namespaces

- package [TriDevs.TriEngine2D.Audio](#)

## 8.6 TriDevs.TriEngine2D/Audio/ISound.cs File Reference

### Classes

- interface [TriDevs.TriEngine2D.Audio.ISound](#)  
*A sound file for use with the audio manager.*

## Namespaces

- package [TriDevs.TriEngine2D.Audio](#)

## 8.7 TriDevs.TriEngine2D/Audio/NullAudioManager.cs File Reference

### Classes

- class [TriDevs.TriEngine2D.Audio.NullAudioManager](#)  
*Used as a fallback [AudioManager](#) object when the service locator fails to find one.*

## Namespaces

- package [TriDevs.TriEngine2D.Audio](#)

## 8.8 TriDevs.TriEngine2D/Audio/NullSong.cs File Reference

### Classes

- class [TriDevs.TriEngine2D.Audio.NullSong](#)  
*Fallback song class used in [NullAudioManager](#).*

## Namespaces

- package [TriDevs.TriEngine2D.Audio](#)

## 8.9 TriDevs.TriEngine2D/Audio/NullSound.cs File Reference

### Classes

- class [TriDevs.TriEngine2D.Audio.NullSound](#)  
*Fallback sound class used in [NullAudioManager](#).*

### Namespaces

- package [TriDevs.TriEngine2D.Audio](#)

## 8.10 TriDevs.TriEngine2D/Audio/Song.cs File Reference

### Classes

- class [TriDevs.TriEngine2D.Audio.Song](#)  
*[Song](#) class that can be used with [AudioManager](#).*

### Namespaces

- package [TriDevs.TriEngine2D.Audio](#)

## 8.11 TriDevs.TriEngine2D/Audio/Sound.cs File Reference

### Classes

- class [TriDevs.TriEngine2D.Audio.Sound](#)  
*[Sound](#) class that can be used with the [AudioManager](#).*

### Namespaces

- package [TriDevs.TriEngine2D.Audio](#)

## 8.12 TriDevs.TriEngine2D/Extensions/EnumerationExtensions.cs File Reference

### Classes

- class [TriDevs.TriEngine2D.Extensions.EnumerationExtensions](#)  
*[Extensions](#) for [System.Enum](#).*

### Namespaces

- package [TriDevs.TriEngine2D.Extensions](#)

## 8.13 TriDevs.TriEngine2D/Extensions/StringExtensions.cs File Reference

### Classes

- class [TriDevs.TriEngine2D.Extensions.StringExtensions](#)  
*Extensions for System.String*

### Namespaces

- package [TriDevs.TriEngine2D.Extensions](#)

## 8.14 TriDevs.TriEngine2D/Helpers/IO.cs File Reference

### Classes

- class [TriDevs.TriEngine2D.Helpers.IO](#)  
*Provides various helper functions for doing IO operations.*

### Namespaces

- package [TriDevs.TriEngine2D.Helpers](#)

## 8.15 TriDevs.TriEngine2D/Helpers/Math.cs File Reference

### Classes

- class [TriDevs.TriEngine2D.Helpers.Math](#)  
*Various helper methods for working with math.*

### Namespaces

- package [TriDevs.TriEngine2D.Helpers](#)

## 8.16 TriDevs.TriEngine2D/Helpers/Threading.cs File Reference

### Classes

- class [TriDevs.TriEngine2D.Helpers.Threading](#)  
*Provides various helper functions for doing threading operations.*

### Namespaces

- package [TriDevs.TriEngine2D.Helpers](#)



## 8.17 TriDevs.TriEngine2D/Input/IInputManager.cs File Reference

### Classes

- interface [TriDevs.TriEngine2D.Input.IInputManager](#)  
*Provides various methods to query input devices like the keyboard.*

### Namespaces

- package [TriDevs.TriEngine2D.Input](#)

## 8.18 TriDevs.TriEngine2D/Input/InputManager.cs File Reference

### Classes

- class [TriDevs.TriEngine2D.Input.InputManager](#)  
*Input manager interfacing with input methods provided by a GameWindow.*

### Namespaces

- package [TriDevs.TriEngine2D.Input](#)

## 8.19 TriDevs.TriEngine2D/Input/NullInputManager.cs File Reference

### Classes

- class [TriDevs.TriEngine2D.Input.NullInputManager](#)  
*Used as a fallback [IInputManager](#) object when the service locator fails to find one.*

### Namespaces

- package [TriDevs.TriEngine2D.Input](#)

## 8.20 TriDevs.TriEngine2D/Logging/LogManager.cs File Reference

### Classes

- class [TriDevs.TriEngine2D.Logging.LogManager](#)  
*Class to manage logging. ILog interfaces should be obtained from this class' methods, as opposed to calling default log4net methods.*

### Namespaces

- package [TriDevs.TriEngine2D.Logging](#)

## 8.21 TriDevs.TriEngine2D/Native/Helpers.cs File Reference

### Classes

- class [TriDevs.TriEngine2D.Native.Helpers](#)  
*Helper class with various methods to help native coding and debugging.*

### Namespaces

- package [TriDevs.TriEngine2D.Native](#)

## 8.22 TriDevs.TriEngine2D/Native/WinAPI.cs File Reference

### Classes

- class [TriDevs.TriEngine2D.Native.WinAPI](#)  
*Holds various [WinAPI](#) stuff.*

### Namespaces

- package [TriDevs.TriEngine2D.Native](#)

## 8.23 TriDevs.TriEngine2D/Point.cs File Reference

### Classes

- struct [TriDevs.TriEngine2D.Point](#)  
*A struct representing an X/Y coordinate.*

### Namespaces

- package [TriDevs.TriEngine2D](#)

## 8.24 TriDevs.TriEngine2D/Properties/AssemblyInfo.cs File Reference

## 8.25 TriDevs.TriEngine2D/Serializing/Serializer.cs File Reference

### Classes

- class [TriDevs.TriEngine2D.Serializing.Serializer](#)  
*Provides serialization methods.*

### Namespaces

- package [TriDevs.TriEngine2D.Serializing](#)

## 8.26 TriDevs.TriEngine2D/Services.cs File Reference

### Classes

- class [TriDevs.TriEngine2D.Services](#)  
*Provides different game-related service interfaces.*

### Namespaces

- package [TriDevs.TriEngine2D](#)

## 8.27 TriDevs.TriEngine2D/Version.cs File Reference

### Classes

- class [TriDevs.TriEngine2D.Version](#)  
*[Version](#) class specifying the version of this project.*

### Namespaces

- package [TriDevs.TriEngine2D](#)

# Index

- AllocConsole
  - TriDevs::TriEngine2D::Native::WinAPI, [78](#)
- Audio
  - TriDevs::TriEngine2D::Services, [69](#)
- AudioFormat
  - TriDevs::TriEngine2D::Audio, [12](#)
- AudioManager
  - TriDevs::TriEngine2D::Audio::AudioManager, [17](#)
- ButtonDown
  - TriDevs::TriEngine2D::Input::IInputManager, [29](#)
  - TriDevs::TriEngine2D::Input::InputManager, [35](#)
  - TriDevs::TriEngine2D::Input::NullInputManager, [56](#)
- ButtonPressed
  - TriDevs::TriEngine2D::Input::IInputManager, [29](#)
  - TriDevs::TriEngine2D::Input::InputManager, [35](#)
  - TriDevs::TriEngine2D::Input::NullInputManager, [57](#)
- ButtonReleased
  - TriDevs::TriEngine2D::Input::IInputManager, [30](#)
  - TriDevs::TriEngine2D::Input::InputManager, [36](#)
  - TriDevs::TriEngine2D::Input::NullInputManager, [57](#)
- ButtonUp
  - TriDevs::TriEngine2D::Input::IInputManager, [30](#)
  - TriDevs::TriEngine2D::Input::InputManager, [36](#)
  - TriDevs::TriEngine2D::Input::NullInputManager, [57](#)
- CODE\_PAGE
  - TriDevs::TriEngine2D::Native::WinAPI, [79](#)
- Clamp
  - TriDevs::TriEngine2D::Helpers::Math, [48–50](#)
- ClearOldLogs
  - TriDevs::TriEngine2D::Logging::LogManager, [44](#)
- Deserialize< T >
  - TriDevs::TriEngine2D::Serializing::Serializer, [66](#)
- DestroyConsole
  - TriDevs::TriEngine2D::Logging::LogManager, [45](#)
- Dispose
  - TriDevs::TriEngine2D::Audio::AudioManager, [17](#)
  - TriDevs::TriEngine2D::Audio::NullAudioManager, [52](#)
  - TriDevs::TriEngine2D::Audio::NullSong, [62](#)
  - TriDevs::TriEngine2D::Audio::NullSound, [64](#)
  - TriDevs::TriEngine2D::Audio::Song, [71](#)
  - TriDevs::TriEngine2D::Audio::Sound, [73](#)
- File
  - TriDevs::TriEngine2D::Audio::ISong, [42](#)
  - TriDevs::TriEngine2D::Audio::ISound, [44](#)
  - TriDevs::TriEngine2D::Audio::NullSong, [63](#)
  - TriDevs::TriEngine2D::Audio::NullSound, [65](#)
  - TriDevs::TriEngine2D::Audio::Song, [72](#)
  - TriDevs::TriEngine2D::Audio::Sound, [74](#)
- FreeConsole
  - TriDevs::TriEngine2D::Native::WinAPI, [78](#)
- GetAbsolutePath
  - TriDevs::TriEngine2D::Helpers::IO, [39](#)
- GetErrorMessage
  - TriDevs::TriEngine2D::Native::Helpers, [23](#)
- GetLastError
  - TriDevs::TriEngine2D::Native::Helpers, [23](#)
- GetLastErrorInfo
  - TriDevs::TriEngine2D::Native::Helpers, [23](#)
- GetLastErrorMessage
  - TriDevs::TriEngine2D::Native::Helpers, [24](#)
- GetLastWin32Exception
  - TriDevs::TriEngine2D::Native::Helpers, [24](#)
- GetLogger
  - TriDevs::TriEngine2D::Logging::LogManager, [45](#)
- GetSong
  - TriDevs::TriEngine2D::Audio::AudioManager, [17](#)
  - TriDevs::TriEngine2D::Audio::IAudioManager, [26](#)
  - TriDevs::TriEngine2D::Audio::NullAudioManager, [52](#)
- GetSound
  - TriDevs::TriEngine2D::Audio::AudioManager, [17](#)
  - TriDevs::TriEngine2D::Audio::IAudioManager, [26](#)
  - TriDevs::TriEngine2D::Audio::NullAudioManager, [52](#)
- GetStdHandle
  - TriDevs::TriEngine2D::Native::WinAPI, [79](#)
- GetWin32Exception
  - TriDevs::TriEngine2D::Native::Helpers, [24](#)
- Has< T >
  - TriDevs::TriEngine2D::Extensions::Enumeration-Extensions, [20](#)
- HasSong
  - TriDevs::TriEngine2D::Audio::AudioManager, [18](#)
  - TriDevs::TriEngine2D::Audio::IAudioManager, [26](#)
  - TriDevs::TriEngine2D::Audio::NullAudioManager, [53](#)
- HasSound
  - TriDevs::TriEngine2D::Audio::AudioManager, [18](#)
  - TriDevs::TriEngine2D::Audio::IAudioManager, [27](#)
  - TriDevs::TriEngine2D::Audio::NullAudioManager, [53](#)
- Include< T >

- TriDevs::TriEngine2D::Extensions::Enumeration-Extensions, 21
- Input
  - TriDevs::TriEngine2D::Services, 69
- InputManager
  - TriDevs::TriEngine2D::Input::InputManager, 35
- IsLooped
  - TriDevs::TriEngine2D::Audio::ISong, 42
  - TriDevs::TriEngine2D::Audio::NullSong, 63
  - TriDevs::TriEngine2D::Audio::Song, 72
- KeyDown
  - TriDevs::TriEngine2D::Input::IInputManager, 30
  - TriDevs::TriEngine2D::Input::InputManager, 36
  - TriDevs::TriEngine2D::Input::NullInputManager, 58
- KeyPressed
  - TriDevs::TriEngine2D::Input::IInputManager, 31
  - TriDevs::TriEngine2D::Input::InputManager, 37
  - TriDevs::TriEngine2D::Input::NullInputManager, 58
- KeyReleased
  - TriDevs::TriEngine2D::Input::IInputManager, 31
  - TriDevs::TriEngine2D::Input::InputManager, 37
  - TriDevs::TriEngine2D::Input::NullInputManager, 58
- KeyUp
  - TriDevs::TriEngine2D::Input::IInputManager, 31
  - TriDevs::TriEngine2D::Input::InputManager, 37
  - TriDevs::TriEngine2D::Input::NullInputManager, 59
- LoadConfig
  - TriDevs::TriEngine2D::Logging::LogManager, 46
- LoadSong
  - TriDevs::TriEngine2D::Audio::AudioManager, 18
  - TriDevs::TriEngine2D::Audio::IAudioManager, 27
  - TriDevs::TriEngine2D::Audio::NullAudioManager, 53
- LoadSound
  - TriDevs::TriEngine2D::Audio::AudioManager, 19
  - TriDevs::TriEngine2D::Audio::IAudioManager, 27
  - TriDevs::TriEngine2D::Audio::NullAudioManager, 54
- Major
  - TriDevs::TriEngine2D::Version, 77
- Minor
  - TriDevs::TriEngine2D::Version, 77
- Missing< T >
  - TriDevs::TriEngine2D::Extensions::Enumeration-Extensions, 21
- MousePosition
  - TriDevs::TriEngine2D::Input::IInputManager, 32
  - TriDevs::TriEngine2D::Input::InputManager, 39
  - TriDevs::TriEngine2D::Input::NullInputManager, 60
- MouseWheelValue
  - TriDevs::TriEngine2D::Input::IInputManager, 32
  - TriDevs::TriEngine2D::Input::InputManager, 39
  - TriDevs::TriEngine2D::Input::NullInputManager, 60
- MouseX
  - TriDevs::TriEngine2D::Input::IInputManager, 32
  - TriDevs::TriEngine2D::Input::InputManager, 39
- TriDevs::TriEngine2D::Input::NullInputManager, 60
- MouseY
  - TriDevs::TriEngine2D::Input::IInputManager, 32
  - TriDevs::TriEngine2D::Input::InputManager, 39
  - TriDevs::TriEngine2D::Input::NullInputManager, 60
- Name
  - TriDevs::TriEngine2D::Audio::ISong, 42
  - TriDevs::TriEngine2D::Audio::ISound, 44
  - TriDevs::TriEngine2D::Audio::NullSong, 63
  - TriDevs::TriEngine2D::Audio::NullSound, 65
  - TriDevs::TriEngine2D::Audio::Song, 72
  - TriDevs::TriEngine2D::Audio::Sound, 74
- Ogg
  - TriDevs::TriEngine2D::Audio, 12
- Patch
  - TriDevs::TriEngine2D::Version, 77
- Pause
  - TriDevs::TriEngine2D::Audio::ISong, 41
  - TriDevs::TriEngine2D::Audio::NullSong, 62
  - TriDevs::TriEngine2D::Audio::Song, 71
- Play
  - TriDevs::TriEngine2D::Audio::ISong, 41
  - TriDevs::TriEngine2D::Audio::ISound, 43
  - TriDevs::TriEngine2D::Audio::NullSong, 62
  - TriDevs::TriEngine2D::Audio::NullSound, 64
  - TriDevs::TriEngine2D::Audio::Song, 71
  - TriDevs::TriEngine2D::Audio::Sound, 73
- Point
  - TriDevs::TriEngine2D::Point, 65
- Provide
  - TriDevs::TriEngine2D::Services, 68, 69
- README.md, 81
- Remove< T >
  - TriDevs::TriEngine2D::Extensions::Enumeration-Extensions, 22
- Replace
  - TriDevs::TriEngine2D::Extensions::StringExtensions, 74, 75
- ReplaceFirst
  - TriDevs::TriEngine2D::Extensions::StringExtensions, 75
- Resume
  - TriDevs::TriEngine2D::Audio::ISong, 41
  - TriDevs::TriEngine2D::Audio::NullSong, 62
  - TriDevs::TriEngine2D::Audio::Song, 71
- Serialize< T >
  - TriDevs::TriEngine2D::Serializing::Serializer, 67
- SetCurrentThreadName
  - TriDevs::TriEngine2D::Helpers::Threading, 76
- SetupConsole
  - TriDevs::TriEngine2D::Logging::LogManager, 46
- Stop
  - TriDevs::TriEngine2D::Audio::ISong, 41
  - TriDevs::TriEngine2D::Audio::ISound, 43

- TriDevs::TriEngine2D::Audio::NullSong, 62
- TriDevs::TriEngine2D::Audio::NullSound, 65
- TriDevs::TriEngine2D::Audio::Song, 72
- TriDevs::TriEngine2D::Audio::Sound, 74
- StopAll
  - TriDevs::TriEngine2D::Audio::AudioManager, 19
  - TriDevs::TriEngine2D::Audio::IAudioManager, 27
  - TriDevs::TriEngine2D::Audio::NullAudioManager, 54
- StopAllSongs
  - TriDevs::TriEngine2D::Audio::AudioManager, 19
  - TriDevs::TriEngine2D::Audio::IAudioManager, 28
  - TriDevs::TriEngine2D::Audio::NullAudioManager, 54
- StopAllSounds
  - TriDevs::TriEngine2D::Audio::AudioManager, 20
  - TriDevs::TriEngine2D::Audio::IAudioManager, 28
  - TriDevs::TriEngine2D::Audio::NullAudioManager, 54
- Suffix
  - TriDevs::TriEngine2D::Version, 77
- TriDevs::TriEngine2D::Audio
  - Ogg, 12
  - Wav, 12
- TriDevs, 11
- TriDevs.TriEngine2D, 11
- TriDevs.TriEngine2D.Audio, 11
- TriDevs.TriEngine2D.Audio.AudioManager, 15
- TriDevs.TriEngine2D.Audio.IAudioManager, 25
- TriDevs.TriEngine2D.Audio.ISong, 40
- TriDevs.TriEngine2D.Audio.ISound, 42
- TriDevs.TriEngine2D.Audio.NullAudioManager, 50
- TriDevs.TriEngine2D.Audio.NullSong, 60
- TriDevs.TriEngine2D.Audio.NullSound, 63
- TriDevs.TriEngine2D.Audio.Song, 69
- TriDevs.TriEngine2D.Audio.Sound, 72
- TriDevs.TriEngine2D.Extensions, 12
- TriDevs.TriEngine2D.Extensions.EnumerationExtensions, 20
- TriDevs.TriEngine2D.Extensions.StringExtensions, 74
- TriDevs.TriEngine2D.Helpers, 12
- TriDevs.TriEngine2D.Helpers.IO, 39
- TriDevs.TriEngine2D.Helpers.Math, 47
- TriDevs.TriEngine2D.Helpers.Threading, 76
- TriDevs.TriEngine2D.Input, 13
- TriDevs.TriEngine2D.Input.IInputManager, 28
- TriDevs.TriEngine2D.Input.InputManager, 33
- TriDevs.TriEngine2D.Input.NullInputManager, 55
- TriDevs.TriEngine2D.Logging, 13
- TriDevs.TriEngine2D.Logging.LogManager, 44
- TriDevs.TriEngine2D.Native, 13
- TriDevs.TriEngine2D.Native.Helpers, 22
- TriDevs.TriEngine2D.Native.WinAPI, 78
- TriDevs.TriEngine2D.Point, 65
- TriDevs.TriEngine2D.Serializing, 13
- TriDevs.TriEngine2D.Serializing.Serializer, 66
- TriDevs.TriEngine2D.Services, 68
- TriDevs.TriEngine2D.Version, 76
- TriDevs.TriEngine2D.Audio.AudioFormat.cs, 81
- TriDevs.TriEngine2D.Audio.AudioManager.cs, 81
- TriDevs.TriEngine2D.Audio.IAudioManager.cs, 81
- TriDevs.TriEngine2D.Audio.ISong.cs, 82
- TriDevs.TriEngine2D.Audio.ISound.cs, 82
- TriDevs.TriEngine2D.Audio.NullAudioManager.cs, 82
- TriDevs.TriEngine2D.Audio.NullSong.cs, 82
- TriDevs.TriEngine2D.Audio.NullSound.cs, 83
- TriDevs.TriEngine2D.Audio.Song.cs, 83
- TriDevs.TriEngine2D.Audio.Sound.cs, 83
- TriDevs.TriEngine2D.Extensions/EnumerationExtensions.-cs, 83
- TriDevs.TriEngine2D.Extensions/StringExtensions.cs, 84
- TriDevs.TriEngine2D/Helpers/IO.cs, 84
- TriDevs.TriEngine2D/Helpers/Math.cs, 84
- TriDevs.TriEngine2D/Helpers/Threading.cs, 84
- TriDevs.TriEngine2D/Input/IInputManager.cs, 85
- TriDevs.TriEngine2D/Input/InputManager.cs, 85
- TriDevs.TriEngine2D/Input/NullInputManager.cs, 85
- TriDevs.TriEngine2D/Logging/LogManager.cs, 85
- TriDevs.TriEngine2D/Native/Helpers.cs, 86
- TriDevs.TriEngine2D/Native/WinAPI.cs, 86
- TriDevs.TriEngine2D/Point.cs, 86
- TriDevs.TriEngine2D/Properties/AssemblyInfo.cs, 86
- TriDevs.TriEngine2D/Serializing/Serializer.cs, 86
- TriDevs.TriEngine2D/Services.cs, 87
- TriDevs.TriEngine2D/Version.cs, 87
- TriDevs::TriEngine2D::Audio
  - AudioFormat, 12
- TriDevs::TriEngine2D::Audio::AudioManager
  - AudioManager, 17
  - Dispose, 17
  - GetSong, 17
  - GetSound, 17
  - HasSong, 18
  - HasSound, 18
  - LoadSong, 18
  - LoadSound, 19
  - StopAll, 19
  - StopAllSongs, 19
  - StopAllSounds, 20
- TriDevs::TriEngine2D::Audio::IAudioManager
  - GetSong, 26
  - GetSound, 26
  - HasSong, 26
  - HasSound, 27
  - LoadSong, 27
  - LoadSound, 27
  - StopAll, 27
  - StopAllSongs, 28
  - StopAllSounds, 28
- TriDevs::TriEngine2D::Audio::ISong
  - File, 42
  - IsLooped, 42
  - Name, 42
  - Pause, 41
  - Play, 41

- Resume, [41](#)
- Stop, [41](#)
- Volume, [42](#)
- TriDevs::TriEngine2D::Audio::ISound
  - File, [44](#)
  - Name, [44](#)
  - Play, [43](#)
  - Stop, [43](#)
- TriDevs::TriEngine2D::Audio::NullAudioManager
  - Dispose, [52](#)
  - GetSong, [52](#)
  - GetSound, [52](#)
  - HasSong, [53](#)
  - HasSound, [53](#)
  - LoadSong, [53](#)
  - LoadSound, [54](#)
  - StopAll, [54](#)
  - StopAllSongs, [54](#)
  - StopAllSounds, [54](#)
- TriDevs::TriEngine2D::Audio::NullSong
  - Dispose, [62](#)
  - File, [63](#)
  - IsLooped, [63](#)
  - Name, [63](#)
  - Pause, [62](#)
  - Play, [62](#)
  - Resume, [62](#)
  - Stop, [62](#)
  - Volume, [63](#)
- TriDevs::TriEngine2D::Audio::NullSound
  - Dispose, [64](#)
  - File, [65](#)
  - Name, [65](#)
  - Play, [64](#)
  - Stop, [65](#)
- TriDevs::TriEngine2D::Audio::Song
  - Dispose, [71](#)
  - File, [72](#)
  - IsLooped, [72](#)
  - Name, [72](#)
  - Pause, [71](#)
  - Play, [71](#)
  - Resume, [71](#)
  - Stop, [72](#)
  - Volume, [72](#)
- TriDevs::TriEngine2D::Audio::Sound
  - Dispose, [73](#)
  - File, [74](#)
  - Name, [74](#)
  - Play, [73](#)
  - Stop, [74](#)
- TriDevs::TriEngine2D::Extensions::EnumerationExtensions
  - Has< T >, [20](#)
  - Include< T >, [21](#)
  - Missing< T >, [21](#)
  - Remove< T >, [22](#)
- TriDevs::TriEngine2D::Extensions::StringExtensions
  - Replace, [74](#), [75](#)
  - ReplaceFirst, [75](#)
- TriDevs::TriEngine2D::Helpers::IO
  - GetAbsolutePath, [39](#)
- TriDevs::TriEngine2D::Helpers::Math
  - Clamp, [48–50](#)
- TriDevs::TriEngine2D::Helpers::Threading
  - SetCurrentThreadName, [76](#)
- TriDevs::TriEngine2D::Input::InputManager
  - ButtonDown, [29](#)
  - ButtonPressed, [29](#)
  - ButtonReleased, [30](#)
  - ButtonUp, [30](#)
  - KeyDown, [30](#)
  - KeyPressed, [31](#)
  - KeyReleased, [31](#)
  - KeyUp, [31](#)
  - MousePosition, [32](#)
  - MouseWheelValue, [32](#)
  - MouseX, [32](#)
  - MouseY, [32](#)
  - Update, [31](#)
  - WheelChange, [31](#)
  - WheelChanged, [32](#)
  - WheelDown, [32](#)
  - WheelUp, [32](#)
- TriDevs::TriEngine2D::Input::InputManager
  - ButtonDown, [35](#)
  - ButtonPressed, [35](#)
  - ButtonReleased, [36](#)
  - ButtonUp, [36](#)
  - InputManager, [35](#)
  - KeyDown, [36](#)
  - KeyPressed, [37](#)
  - KeyReleased, [37](#)
  - KeyUp, [37](#)
  - MousePosition, [39](#)
  - MouseWheelValue, [39](#)
  - MouseX, [39](#)
  - MouseY, [39](#)
  - Update, [38](#)
  - WheelChange, [38](#)
  - WheelChanged, [38](#)
  - WheelDown, [38](#)
  - WheelUp, [38](#)
- TriDevs::TriEngine2D::Input::NullInputManager
  - ButtonDown, [56](#)
  - ButtonPressed, [57](#)
  - ButtonReleased, [57](#)
  - ButtonUp, [57](#)
  - KeyDown, [58](#)
  - KeyPressed, [58](#)
  - KeyReleased, [58](#)
  - KeyUp, [59](#)
  - MousePosition, [60](#)
  - MouseWheelValue, [60](#)
  - MouseX, [60](#)
  - MouseY, [60](#)
  - Update, [59](#)

- WheelChange, [59](#)
- WheelChanged, [59](#)
- WheelDown, [59](#)
- WheelUp, [60](#)
- TriDevs::TriEngine2D::Logging::LogManager
  - ClearOldLogs, [44](#)
  - DestroyConsole, [45](#)
  - GetLogger, [45](#)
  - LoadConfig, [46](#)
  - SetupConsole, [46](#)
- TriDevs::TriEngine2D::Native::Helpers
  - GetErrorMessage, [23](#)
  - GetLastError, [23](#)
  - GetLastErrorInfo, [23](#)
  - GetLastErrorMessage, [24](#)
  - GetLastWin32Exception, [24](#)
  - GetWin32Exception, [24](#)
- TriDevs::TriEngine2D::Native::WinAPI
  - AllocConsole, [78](#)
  - FreeConsole, [78](#)
  - GetStdHandle, [79](#)
- TriDevs::TriEngine2D::Point
  - Point, [65](#)
  - X, [66](#)
  - Y, [66](#)
- TriDevs::TriEngine2D::Serializing::Serializer
  - Deserialize< T >, [66](#)
  - Serialize< T >, [67](#)
- TriDevs::TriEngine2D::Services
  - Audio, [69](#)
  - Input, [69](#)
  - Provide, [68](#), [69](#)
- TriDevs::TriEngine2D::Version
  - Major, [77](#)
  - Minor, [77](#)
  - Patch, [77](#)
  - Suffix, [77](#)
  - VersionString, [77](#)
  - VersionStringFormat, [77](#)
  - VersionStringFormatWithSuffix, [77](#)
- Update
  - TriDevs::TriEngine2D::Input::IInputManager, [31](#)
  - TriDevs::TriEngine2D::Input::InputManager, [38](#)
  - TriDevs::TriEngine2D::Input::NullInputManager, [59](#)
- VersionString
  - TriDevs::TriEngine2D::Version, [77](#)
- VersionStringFormat
  - TriDevs::TriEngine2D::Version, [77](#)
- VersionStringFormatWithSuffix
  - TriDevs::TriEngine2D::Version, [77](#)
- Volume
  - TriDevs::TriEngine2D::Audio::ISong, [42](#)
  - TriDevs::TriEngine2D::Audio::NullSong, [63](#)
  - TriDevs::TriEngine2D::Audio::Song, [72](#)
- Wav
  - TriDevs::TriEngine2D::Audio, [12](#)
- WheelChange
  - TriDevs::TriEngine2D::Input::IInputManager, [31](#)
  - TriDevs::TriEngine2D::Input::InputManager, [38](#)
  - TriDevs::TriEngine2D::Input::NullInputManager, [59](#)
- WheelChanged
  - TriDevs::TriEngine2D::Input::IInputManager, [32](#)
  - TriDevs::TriEngine2D::Input::InputManager, [38](#)
  - TriDevs::TriEngine2D::Input::NullInputManager, [59](#)
- WheelDown
  - TriDevs::TriEngine2D::Input::IInputManager, [32](#)
  - TriDevs::TriEngine2D::Input::InputManager, [38](#)
  - TriDevs::TriEngine2D::Input::NullInputManager, [59](#)
- WheelUp
  - TriDevs::TriEngine2D::Input::IInputManager, [32](#)
  - TriDevs::TriEngine2D::Input::InputManager, [38](#)
  - TriDevs::TriEngine2D::Input::NullInputManager, [60](#)
- X
  - TriDevs::TriEngine2D::Point, [66](#)
- Y
  - TriDevs::TriEngine2D::Point, [66](#)