TriEngine2D

v0.0.8

Generated by Doxygen 1.8.3.1

Wed Feb 13 2013 01:42:55

Contents

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

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: "&lt; <strong>Namespace</strong> &gt;.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:

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

Chapter 4

Class Index

4.1 Class List

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

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

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 specifiying 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

```
{
    Ogg,

    Wav
}
```

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.

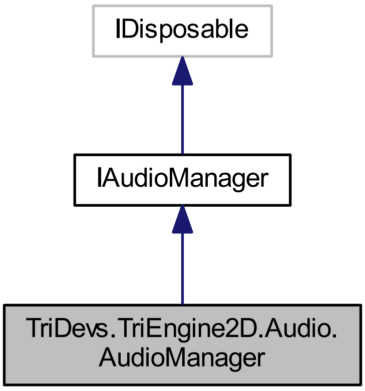
Chapter 7

Class Documentation

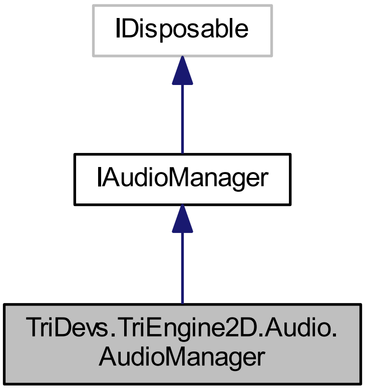
7.1 TriDevs.TriEngine2D.Audio.AudioManager Class Reference

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.

```
        {
            _context = new AudioContext();
            _oggStreamer = new OggStreamer();
            _sounds = new List<ISound>();
            _songs = new List<ISong>();
        }
```

7.1.3 Member Function Documentation

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

```
        {
            foreach (var sound in _sounds.Where(sound => sound != null))
                sound.Dispose();

            foreach (var song in _songs.Where(song => song != null))
                song.Dispose();

            if (_oggStreamer != null)
            {
                _oggStreamer.Dispose();
                _oggStreamer = null;
            }

            if (_context != null)
            {
                _context.Dispose();
                _context = null;
            }
        }
```

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

Gets the song with the specified name.

Parameters

name	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.

```
        {
            return _songs.FirstOrDefault(s => s.Name == name);
        }
```

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

Gets the sound with the specified name.

Parameters

name	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.

```
        {
            return _sounds.FirstOrDefault(s => s.Name == name);
        }
```

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

name	Name to check for.

Returns

True if the song has been loaded, false otherwise.

Implements TriDevs.TriEngine2D.Audio.IAudioManager.

```
        {
            return _songs.Any(s => s.Name == name);
        }
```

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

name	Name to check for.

Returns

True if the sound has been loaded, false otherwise.

Implements TriDevs.TriEngine2D.Audio.IAudioManager.

```
        {
            return _sounds.Any(s => s.Name == name);
        }
```

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

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

Returns

The resulting object implementing the ISong interface.

Implements TriDevs.TriEngine2D.Audio.IAudioManager.

```
        {
            var existing = _songs.FirstOrDefault(s => s.Name == name);
            if (existing != null)
                return existing;

            var loaded = _songs.FirstOrDefault(s => s.File == file) != null;
            if (loaded)
                throw new Exception("The song file \"" + file + "\" has already been loaded under a different name.");

            var song = new Song(name, file, format);
            _songs.Add(song);
            return song;
        }
```

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

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

Returns

The resulting object implementing the ISound interface.

Implements TriDevs.TriEngine2D.Audio.IAudioManager.

```
        {
            var existing = _sounds.FirstOrDefault(s => s.Name == name);
            if (existing != null)
                return existing;

            var loaded = _sounds.FirstOrDefault(s => s.File == file) != null;
            if (loaded)
                throw new Exception("The sound file \"" + file + "\" has already been loaded under a different name.");

            var sound = new Sound(name, file, format);
            _sounds.Add(sound);
            return sound;
        }
```

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

Immediately stops all playback of sounds and songs.

Implements TriDevs.TriEngine2D.Audio.IAudioManager.

```
        {
            StopAllSounds();
            StopAllSongs();
        }
```

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

Immediately stops playback of all songs.

Implements TriDevs.TriEngine2D.Audio.IAudioManager.

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

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

Immediately stops playback of all sounds.

Implements TriDevs.TriEngine2D.Audio.IAudioManager.

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

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

T	The enum type.

Parameters

value	The enum to check.
check	Value to check for.

Returns

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

```
        {
            Type type = value.GetType();

            //determine the values
            var parsed = new _Value(check, type);
            if (parsed.Signed.HasValue) //if (parsed.Signed is long)
            {
                return (Convert.ToInt64(value) & (long)parsed.Signed) == (long)parsed.Signed;
            }
            if (parsed.Unsigned.HasValue) //if (parsed.Unsigned is ulong)
            {
                return (Convert.ToUInt64(value) & (ulong)parsed.Unsigned) == (ulong)parsed.Unsigned;
            }
            return false;
        }
```

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

T	The enum type.

Parameters

value	The enum to append to.
append	Value to append.

Returns

New enum T with the new values.

```
        {
            var type = value.GetType();

            //determine the values
            object result = value;
            var parsed = new _Value(append, type);
            if (parsed.Signed.HasValue) //if (parsed.Signed is long)
            {
                result = Convert.ToInt64(value) | (long)parsed.Signed;
            }
            else if (parsed.Unsigned.HasValue) //else if (parsed.Unsigned is ulong)
            {
                result = Convert.ToUInt64(value) | (ulong)parsed.Unsigned;
            }

            //return the final value
            return (T)Enum.Parse(type, result.ToString());
        }
```

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

T	The enum type.

Parameters

obj	The enum to check.
value	Value to check for.

Returns

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

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

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

T	The enum type.

Parameters

value	The enum to remove from.
remove	Value to remove.

Returns

New enum T with the value(s) removed.

```
        {
            Type type = value.GetType();

            //determine the values
            object result = value;
            var parsed = new _Value(remove, type);
            if (parsed.Signed.HasValue) //if (parsed.Signed is long)
            {
                result = Convert.ToInt64(value) & ~(long)parsed.Signed;
            }
            else if (parsed.Unsigned.HasValue) //else if (parsed.Unsigned is ulong)
            {
                result = Convert.ToUInt64(value) & ~(ulong)parsed.Unsigned;
            }

            //return the final value
            return (T)Enum.Parse(type, result.ToString());
        }
```

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

err	The error code to get information about.

Returns

The error message.

```
        {
            return GetWin32Exception(err).Message;
        }
```

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.

```
        {
            return Marshal.GetLastWin32Error();
        }
```

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

message	Will be set to the error message.

Returns

The error code associated with the thrown error.

```
        {
            var err = GetLastError();
            message = GetWin32Exception(err).Message;
            return err;
        }
```

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.

```
        {
            return GetErrorMessage(GetLastError());
        }
```

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

Gets the Win32Exception associated with the last thrown error.

Returns

The Win32Exception.

```
        {
            return GetWin32Exception(GetLastError());
        }
```

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

Gets the Win32Exception associated with the specified error code.

Parameters

err	The error code.

Returns

The Win32Exception for the provided error code.

```
        {
            return new Win32Exception(err);
        }
```

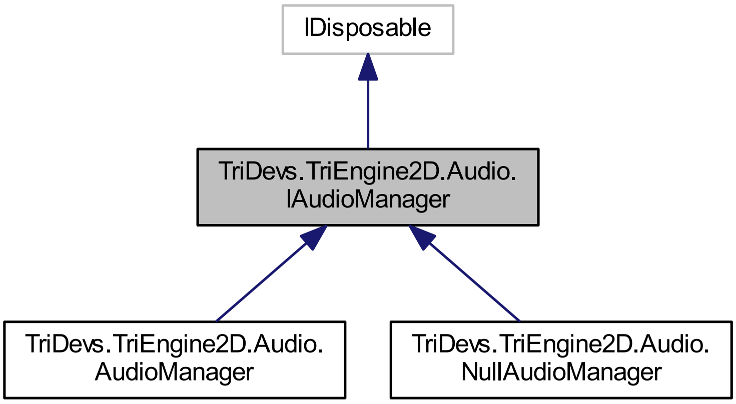
The documentation for this class was generated from the following file:

- TriDevs.TriEngine2D/Native/Helpers.cs

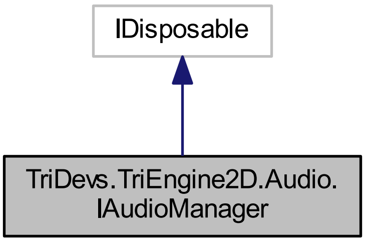
7.4 TriDevs.TriEngine2D.Audio.IAudioManager Interface Reference

Provides various methods to manipulate audio.

Inheritance diagram for TriDevs.TriEngine2D.Audio.IAudioManager:


IDisposable
TriDevs.TriEngine2D.Audio.IAudioManager
TriDevs.TriEngine2D.Audio.AudioManager
TriDevs.TriEngine2D.Audio.NullAudioManager

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

name	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

name	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

name	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

name	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

name	Name to assign to the new song object.
file	Path to the file that will be used.
format	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

name	Name to assign to the new sound object.
file	Path to the file that will be used.
format	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.

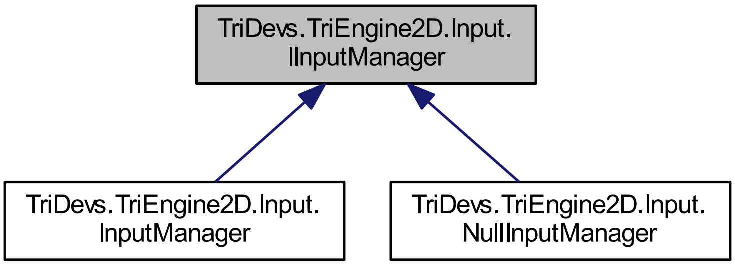
The documentation for this interface was generated from the following file:

- 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:


TriDevs.TriEngine2D.Input.
IInputManager
TriDevs.TriEngine2D.Input.
InputManager
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]`

  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

button	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

button	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

button	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

button	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

key	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

key	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

key	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

key	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

key	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

button	The button to query.

Returns

True if pressed, false otherwise.

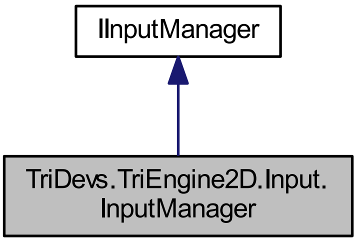
The documentation for this interface was generated from the following file:

- TriDevs.TriEngine2D/Input/IInputManager.cs

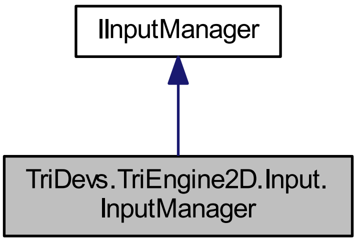
7.6 TriDevs.TriEngine2D.Input.InputManager Class Reference

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

window	The GameWindow this InputManager will interface with.

```
        {
            _mouse = window.Mouse;
        }
```

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

button	The button to query for.

Returns

True if button is currently being pressed, false otherwise.

Implements TriDevs.TriEngine2D.Input.IInputManager.

```
        {
            return _mouseState[button];
        }
```

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

button	Button to query for.

Returns

True if button was pressed, false otherwise.

Implements TriDevs.TriEngine2D.Input.IInputManager.

```
        {
            return _mouseState[button] && !_lastMouseState[button];
        }
```

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

button	The button to query for.

Returns

True if the button was released, false otherwise.

Implements TriDevs.TriEngine2D.Input.IInputManager.

```
        {
            return !_mouseState[button] && _lastMouseState[button];
        }
```

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

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

Parameters

button	Button to query for.

Returns

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

Implements TriDevs.TriEngine2D.Input.IInputManager.

```
        {
            return !_mouseState[button];
        }
```

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

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

Parameters

key	Key to query for.

Returns

True if key is currently being pressed, false otherwise.

Implements TriDevs.TriEngine2D.Input.IInputManager.

```
        {
            return _keyboardState[key];
        }
```

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

key	Key to query for.

Returns

True if key was pressed, false otherwise.

Implements TriDevs.TriEngine2D.Input.IInputManager.

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

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

key	Key to query for.

Returns

True if key was released, false otherwise.

Implements TriDevs.TriEngine2D.Input.IInputManager.

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

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

Returns whether or not the specified key is currently unpressed.

Parameters

key	Key to query for.

Returns

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

Implements TriDevs.TriEngine2D.Input.IInputManager.

```
        {
            return !_keyboardState[key];
        }
```

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.

```
        {
            _lastKeyboardState = _keyboardState;
            _keyboardState = Keyboard.GetState();

            _lastMouseState = _mouseState;
            _mouseState = Mouse.GetState();
        }
```

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.

```
        {
            return _mouseState.Wheel - _lastMouseState.Wheel;
        }
```

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.

```
        {
            return _mouseState.Wheel != _lastMouseState.Wheel;
        }
```

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.

```
        {
            return _mouseState.Wheel < _lastMouseState.Wheel;
        }
```

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.

```
        {
            return _mouseState.Wheel > _lastMouseState.Wheel;
        }
```

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

path	The relative path to resolve.

Returns

The absolute path to the item.

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

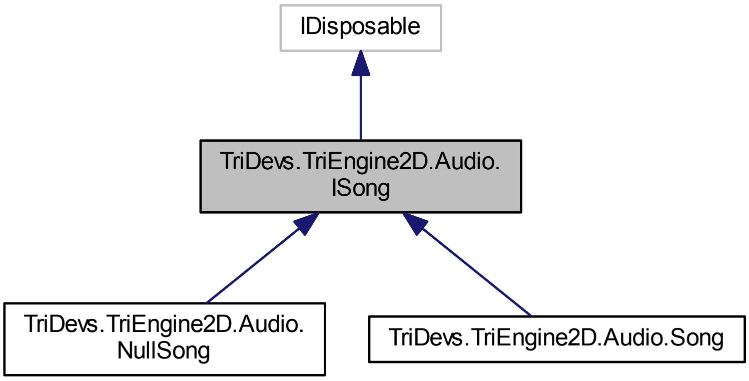
The documentation for this class was generated from the following file:

- TriDevs.TriEngine2D/Helpers/IO.cs

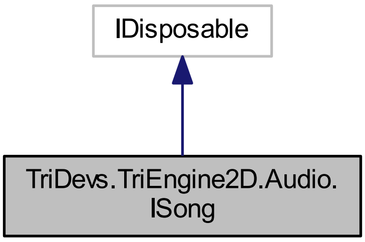
7.8 TriDevs.TriEngine2D.Audio.ISong Interface Reference

A song that will be streamed in the audio player.

Inheritance diagram for TriDevs.TriEngine2D.Audio.ISong:


IDisposable
TriDevs.TriEngine2D.Audio.ISong
TriDevs.TriEngine2D.Audio.NullSong
TriDevs.TriEngine2D.Audio.Song

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.

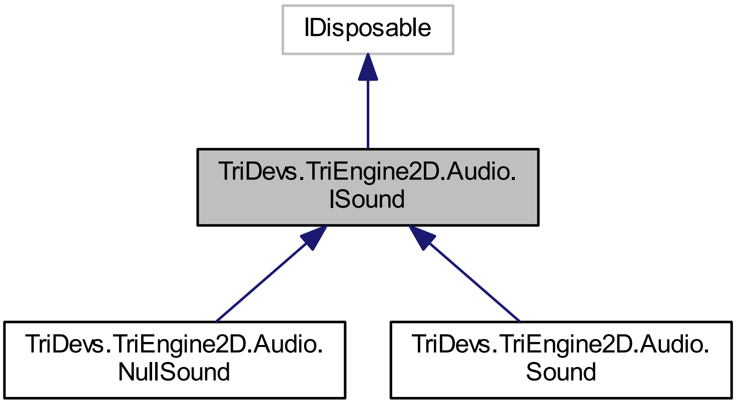
The documentation for this interface was generated from the following file:

- TriDevs.TriEngine2D/Audio/ISong.cs

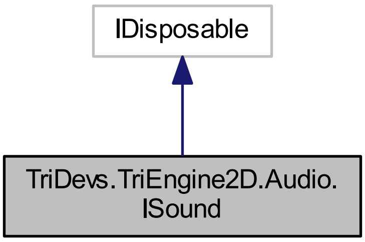
7.9 TriDevs.TriEngine2D.Audio.ISound Interface Reference

A sound file for use with the audio manager.

Inheritance diagram for TriDevs.TriEngine2D.Audio.ISound:


IDisposable
TriDevs.TriEngine2D.Audio.ISound
TriDevs.TriEngine2D.Audio.NullSound
TriDevs.TriEngine2D.Audio.Sound

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

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

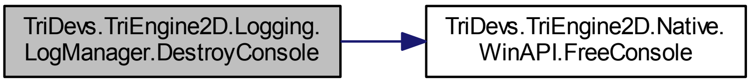
```
        {
            var log = GetLogger(typeof(LogManager));

            log.InfoFormat(">> ClearOldLogs({0}, \"{1}\")", daysOld, logsDir);

            if (!Directory.Exists(logsDir))
            {
                log.InfoFormat("Directory {0} not found, no logs to clear", logsDir);
                log.Info("<< ClearOldLogs()");
                return;
            }

            var now = DateTime.Now;
            var max = new TimeSpan(daysOld, 0, 0, 0);
            var count = 0;
            foreach (var file in from file in Directory.GetFiles(logsDir)
                                 let modTime = File.GetLastAccessTime(file)
                                 let age = now.Subtract(modTime)
                                 where age > max
                                 select file)
            {
                try
                {
                    File.Delete(file);
                    log.InfoFormat("Deleted old log file: {0}", file);
                    count++;
                }
                catch (IOException ex)
                {
                    log.WarnFormat("Failed to delete log file: {0} ({1})", file, ex.Message);
                }
            }

            log.InfoFormat("Done! Cleared {0} log files.", count);
            log.Info("<< ClearOldLogs()");
        }
```

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.

```
        {
#if DEBUG
            if (_consoleLoaded)
                WinAPI.FreeConsole();
#endif
        }
```

Here is the call graph for this function:


TriDevs.TriEngine2D.Logging.
LogManager.DestroyConsole
TriDevs.TriEngine2D.Native.
WinAPI.FreeConsole

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

sender	The object or Type to get an ILog object for.

Returns

The ILog object.

```
        {
            if (!_loaded)
                LoadConfig();

            return log4net.LogManager.GetLogger(sender.GetType().ToString() == "System.RuntimeType" ? (Type
   )sender : sender.GetType());
        }
```

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: "(AssemblyName).config" If it is unable to load the config, it will default to BasicConfigurator.

Parameters

file	The config file to load, null if automatic loading is preferred.

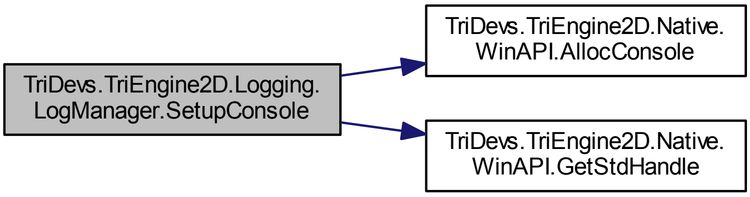
```
        {
            if (file == null)
            {
                if (File.Exists(AppDomain.CurrentDomain.FriendlyName + ".config"))
                    XmlConfigurator.Configure();
                else
                    BasicConfigurator.Configure();
            }
            else
            {
                if (File.Exists(file))
                    XmlConfigurator.Configure(new FileInfo(file));
                else
                {
                    LoadConfig();
                    return;
                }
            }

            _loaded = true;
        }
```

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.

```
        {
#if DEBUG
            if (System.Diagnostics.Debugger.IsAttached)
                return;

            WinAPI.AllocConsole();
            var stdHandle = WinAPI.GetStdHandle(WinAPI.
  STD_OUTPUT_HANDLE);
            var safeFileHandle = new SafeFileHandle(stdHandle, true);
            var fileStream = new FileStream(safeFileHandle, FileAccess.Write);
            var encoding = Encoding.GetEncoding(WinAPI.CODE_PAGE);
            var stdOut = new StreamWriter(fileStream, encoding) { AutoFlush = true };
            Console.SetOut(stdOut);
            _consoleLoaded = true;
#endif
        }
```

Here is the call graph for this function:


TriDevs.TriEngine2D.Logging.
LogManager.SetupConsole
TriDevs.TriEngine2D.Native.
WinAPI.AllocConsole
TriDevs.TriEngine2D.Native.
WinAPI.GetStdHandle

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

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

Returns

The clamped value of the byte.

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

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

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

Returns

The clamped value of the short.

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

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

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

Returns

The clamped value of the unsigned short.

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

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

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

Returns

The clamped value of the integer.

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

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

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

Returns

The clamped value of the unsigned integer.

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

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

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

Returns

The clamped value of the 64-bit integer.

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

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

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

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

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

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

Returns

The clamped value of the float.

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

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

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

Returns

The clamped value of the double.

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

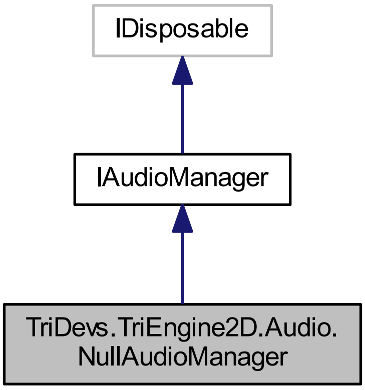
The documentation for this class was generated from the following file:

- TriDevs.TriEngine2D/Helpers/Math.cs

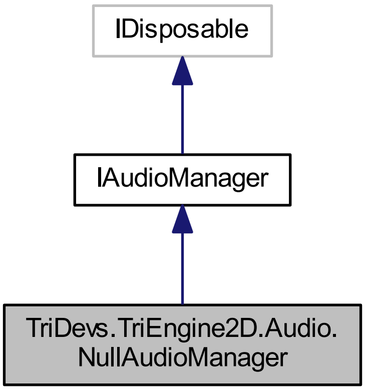
7.12 TriDevs.TriEngine2D.Audio.NullAudioManager Class Reference

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 ( )

```
        {

        }
```

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

Gets the song with the specified name.

Parameters

name	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.

```
        {
            return Song;
        }
```

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

Gets the sound with the specified name.

Parameters

name	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.

```
        {
            return Sound;
        }
```

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

name	Name to check for.

Returns

True if the song has been loaded, false otherwise.

Implements TriDevs.TriEngine2D.Audio.IAudioManager.

```
        {
            return false;
        }
```

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

name	Name to check for.

Returns

True if the sound has been loaded, false otherwise.

Implements TriDevs.TriEngine2D.Audio.IAudioManager.

```
        {
            return false;
        }
```

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

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

Returns

The resulting object implementing the ISong interface.

Implements TriDevs.TriEngine2D.Audio.IAudioManager.

```
        {
            return Song;
        }
```

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

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

Returns

The resulting object implementing the ISound interface.

Implements TriDevs.TriEngine2D.Audio.IAudioManager.

```
        {
            return Sound;
        }
```

7.12.2.8 void TriDevs.TriEngine2D.Audio.NullAudioManager.StopAll ( )

Immediately stops all playback of sounds and songs.

Implements TriDevs.TriEngine2D.Audio.IAudioManager.

```
        {

        }
```

7.12.2.9 void TriDevs.TriEngine2D.Audio.NullAudioManager.StopAllSongs ( )

Immediately stops playback of all songs.

Implements TriDevs.TriEngine2D.Audio.IAudioManager.

```
        {

        }
```

7.12.2.10 void TriDevs.TriEngine2D.Audio.NullAudioManager.StopAllSounds ( )

Immediately stops playback of all sounds.

Implements TriDevs.TriEngine2D.Audio.IAudioManager.

```
        {
        }
```

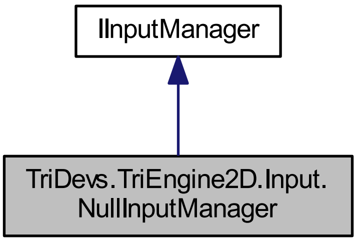
The documentation for this class was generated from the following file:

- TriDevs.TriEngine2D/Audio/NullAudioManager.cs

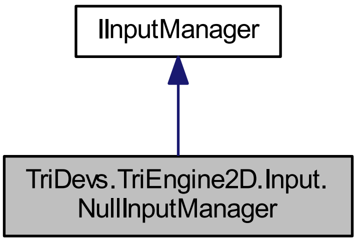
7.13 TriDevs.TriEngine2D.Input.NullInputManager Class Reference

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

button	The button to query for.

Returns

True if button is currently being pressed, false otherwise.

Implements TriDevs.TriEngine2D.Input.IInputManager.

```
79         {
```

```
            return false;
        }
```

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

button	Button to query for.

Returns

True if button was pressed, false otherwise.

Implements TriDevs.TriEngine2D.Input.IInputManager.

```
        {
            return false;
        }
```

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

button	The button to query for.

Returns

True if the button was released, false otherwise.

Implements TriDevs.TriEngine2D.Input.IInputManager.

```
        {
            return false;
        }
```

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

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

Parameters

button	Button to query for.

Returns

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

Implements TriDevs.TriEngine2D.Input.IInputManager.

```
        {
            return true;
        }
```

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

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

Parameters

key	Key to query for.

Returns

True if key is currently being pressed, false otherwise.

Implements TriDevs.TriEngine2D.Input.IInputManager.

```
        {
            return false;
        }
```

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

key	Key to query for.

Returns

True if key was pressed, false otherwise.

Implements TriDevs.TriEngine2D.Input.IInputManager.

```
        {
            return false;
        }
```

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

key	Key to query for.

Returns

True if key was released, false otherwise.

Implements TriDevs.TriEngine2D.Input.IInputManager.

```
        {
            return false;
        }
```

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

Returns whether or not the specified key is currently unpressed.

Parameters

key	Key to query for.

Returns

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

Implements TriDevs.TriEngine2D.Input.IInputManager.

```
        {
            return true;
        }
```

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.

```
        {
            // Do nothing
        }
```

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.

```
        {
            return 0;
        }
```

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.

```
        {
            return false;
        }
```

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.

```
        {
            return false;
         }
```

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.

```
        {
            return false;
        }
```

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]

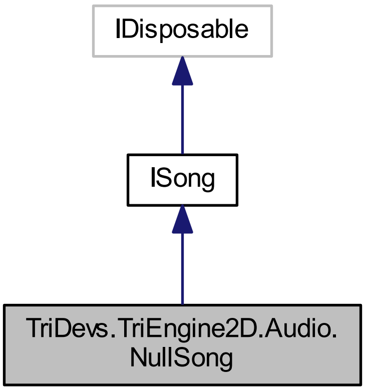
The documentation for this class was generated from the following file:

- TriDevs.TriEngine2D/Input/NullInputManager.cs

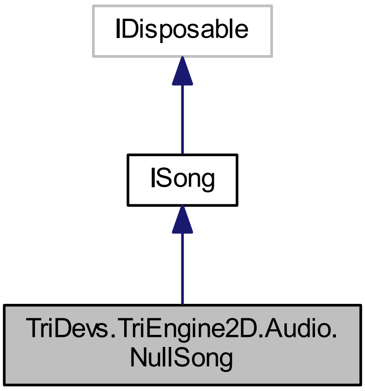
7.14 TriDevs.TriEngine2D.Audio.NullSong Class Reference

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 ( )

```
        {

        }
```

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.

```
        {

        }
```

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

Starts playback of the song.

Implements TriDevs.TriEngine2D.Audio.ISong.

```
        {

        }
```

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

Resumes playback of a paused song.

Implements TriDevs.TriEngine2D.Audio.ISong.

```
        {

        }
```

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

Stops playback of the song.

Implements TriDevs.TriEngine2D.Audio.ISong.

```
        {

        }
```

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]

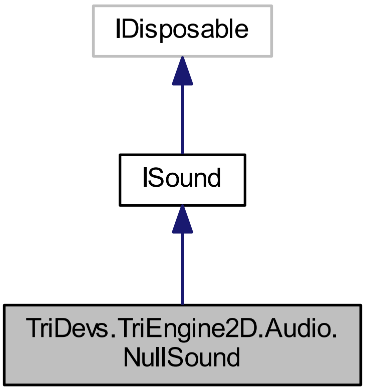
The documentation for this class was generated from the following file:

- TriDevs.TriEngine2D/Audio/NullSong.cs

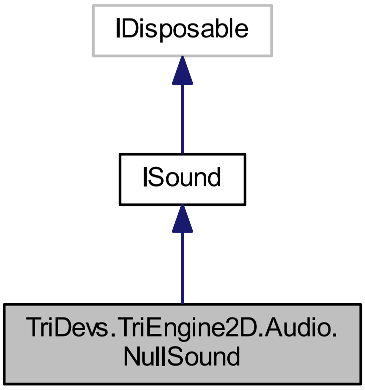
7.15 TriDevs.TriEngine2D.Audio.NullSound Class Reference

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 ( )

```
        {

        }
```

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

Plays the sound.

Implements TriDevs.TriEngine2D.Audio.ISound.

```
        {
        }
```

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

Stops the sound.

Implements TriDevs.TriEngine2D.Audio.ISound.

```
        {
        }
```

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.

```
        {
            X = x;
            Y = y;
        }
```

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

file	File to read from.

Returns

The deserialized object.

```
        {
            T data;
            // Create the StreamReader
            using (var reader = new StreamReader(file))
            {
                // And the json reader
                using (var jsonReader = new JsonTextReader(reader))
                {
                    // Now deserialize the file to the requested object...
                    data = JsonSerializer.Value.Deserialize<T>(jsonReader);
                    // ... and close the json reader.
                    jsonReader.Close();
                }
                // Finally, close the file reader
                reader.Close();
            }
            return data;
        }
```

7.17.2.2 static string TriDevs.TriEngine2D.Serializing.Serializer.Serialize< T > (T *data*) [static]

Serialize an object to string.

Template Parameters

T	Type of data.

Parameters

data	Data to serialize.

Returns

The serialized object in string format.

```
        {
            // Create a StringWriter to hold the serialized object
            var writer = new StringWriter();
            // Serialize the object into the writer
            JsonSerializer.Value.Serialize(writer, data);
            return writer.ToString(); // Return the data as string
        }
```

7.17.2.3 static void TriDevs.TriEngine2D.Serializing.Serializer.Serialize< T > (T *data*, string *file*, Formatting *formatting* = Formatting.Indented) [static]

Serializes an object to file.

Template Parameters

T	Type of the data.

Parameters

data	Data to serialize.
file	File to serialize to.

formatting	The formatting to use for the JSON output.

```
        {
            // Create the StreamWriter
            using (var writer = new StreamWriter(file, false))
            {
                // Create the json writer
                using (var jsonWriter = new JsonTextWriter(writer){Formatting = formatting})
                {
                    // Now serialize the object to the file...
                    JsonSerializer.Value.Serialize(jsonWriter, data);
                    // ... and close the json writer
                    jsonWriter.Close();
                }
                // Finally, close the file writer
                writer.Close();
            }
        }
```

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 intialized 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

input	An object implementing the IInputManager interface.

```
        {
            _input = input;
        }
```

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

Specifies an audio manager service to provide.

Parameters

audio	An object implementing the IAudioManager interface.

```
        {
            _audio = audio;
        }
```

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

Specifies what services to provide.

Parameters

input	The input service to provide.
audio	The audio service to provide.

```
        {
            Provide(input);
            Provide(audio);
        }
```

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.

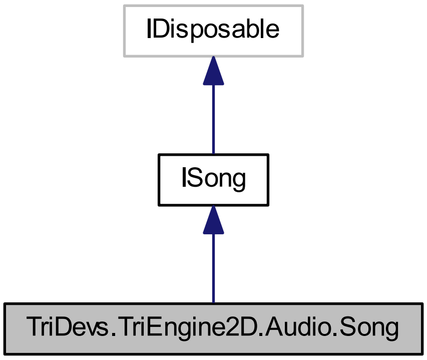
The documentation for this class was generated from the following file:

- TriDevs.TriEngine2D/Services.cs

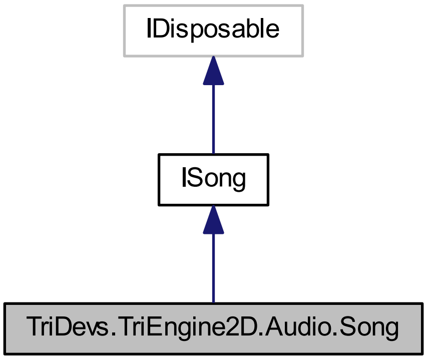
7.19 TriDevs.TriEngine2D.Audio.Song Class Reference

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 ( )

```
        {
            if (_stream == null)
                return;

            Stop();
            _stream.Dispose();
            _stream = null;
        }
```

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.

```
        {
            _stream.Pause();
        }
```

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

Starts playback of the song.

Implements TriDevs.TriEngine2D.Audio.ISong.

```
        {
            try
            {
                _stream.Play();
            }
            catch (InvalidDataException)
            {
                Stop();
                Play();
            }
        }
```

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

Resumes playback of a paused song.

Implements TriDevs.TriEngine2D.Audio.ISong.

```
        {
            _stream.Resume();
        }
```

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

Stops playback of the song.

Implements TriDevs.TriEngine2D.Audio.ISong.

```
        {
            _stream.Stop();
        }
```

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]

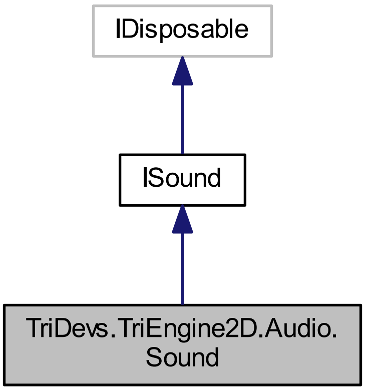
The documentation for this class was generated from the following file:

- TriDevs.TriEngine2D/Audio/Song.cs

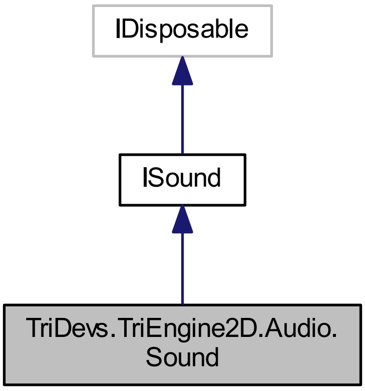
7.20 TriDevs.TriEngine2D.Audio.Sound Class Reference

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 ( )

```
        {
            Stop();
            AL.DeleteSources(_sources);
            AL.DeleteBuffer(_buffer);
        }
```

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

Plays the sound.

Implements TriDevs.TriEngine2D.Audio.ISound.

```
        {
            for (var i = 0; i < SourceCount; i++)
            {
                if (GetSourceState(_sources[i]) != ALSourceState.Playing)
                {
                    AL.SourcePlay(_sources[i]);
                    return;
                }
            }
        }
```

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

Stops the sound.

Implements TriDevs.TriEngine2D.Audio.ISound.

```
        {
            for (var i = 0; i < SourceCount; i++)
                AL.SourceStop(_sources[i]);
        }
```

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

s	String to modify.
search	String to search for.
replace	String to replace the match(es) with.
count	Number of occurrences to replace.
caseInsensitive	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.

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

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

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

Returns

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

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

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

s	String to modify.
search	String to search for.
replace	String to replace the match with.
caseInsensitive	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.

```
        {
            return Replace(s, search, replace, 1, caseInsensitive);
        }
```

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

name	The new name for the current thread

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

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 specifiying 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 specifiying 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

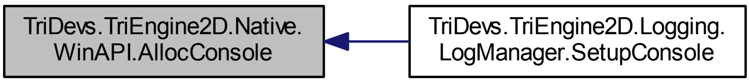
7.24.2.1 static bool TriDevs.TriEngine2D.Native.WinAPI.AllocConsole ( )

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:



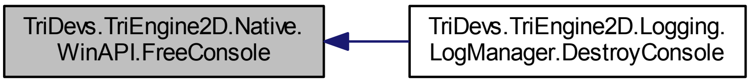
7.24.2.2 static int TriDevs.TriEngine2D.Native.WinAPI.FreeConsole ( )

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:


TriDevs.TriEngine2D.Native.
WinAPI.FreeConsole
TriDevs.TriEngine2D.Logging.
LogManager.DestroyConsole

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).

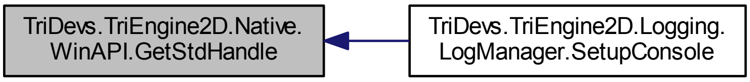
Parameters

nStdHandle	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 InputManager 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 specifiying the version of this project.

Namespaces

- package TriDevs.TriEngine2D

Index