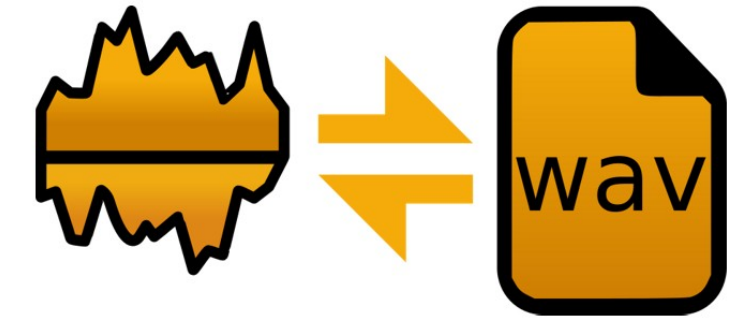
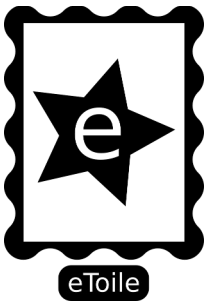


Open WAV Parser

OpenWavParser Class

Cross-platform AudioClip to PCM WAV file codec.



(eToile 2023) V: 1.6

Index

Introduction

Thanks for downloading OpenWavParser, this class is designed to be simple and lightweight, so you will only need to learn how to use a few methods.
This product is just that: a static class, so you can add it to your project without any risks.
You also can access the full source code.
The sample scene allows testing most of the OpenWavParser functions, giving also an example of how to record audio from the microphone.

Class Description

OpenWavParser is a static class. That means that you don't have to create/instantiate an OpenWavParser object, just write OpenWavParser dot (.) followd by the method that you need.

There are not any special considerations when exporting to other platforms, neither any special considerations when uploading to digital markets.
Just switch platform from "Build settings" dialog on Unity editor.

Class Integration

To integrate this class to your project you must include the main file only:

- "OpenWavParser.cs".

This file contains the OpenWavParser class.

Everything else contained in this package can be deleted without risk.

OpenWavParser Public Interfaces

This is the complete definition of OpenWavParser public interfaces.

ByteArrayToAudioClip()

```
public static AudioClip ByteArrayToAudioClip(byte[] wavFile, string name = "",
bool stream = false)
```

The wavFile argument must contain a WAV file to be converted into an AudioClip (contained into a byte[] array).
The name and stream arguments are parameters assigned to the AudioClip at the moment of its creation.

This example reads the sample WAV file:

```
string filePath = Application.persistentDataPath + "/MyFile.wav";
byte[] wavFile = File.ReadAllBytes(filePath);
gameObject.GetComponent<AudioSource>().clip = OpenWavParser.ByteArrayToAudioClip(wavFile);
```

AudioClipToByteArray()

```
public static byte[] AudioClipToByteArray(AudioClip clip, Resolution res = Resolution._16bit)
```

The clip argument requires an AudioClip with some valid data to be converted to a WAV file.
The converted WAV file gets the AudioClip properties to assign the audio format (channels, frequency, etc.).
NOTE: A streamed AudioClip may not be saved correctly. Make sure that the clip is not streamed before converting.

This example saves an AudioClip to a WAV file:

```
AudioSource source = gameObject.GetComponent<AudioSource>();
byte[] wavFile = OpenWavParser.AudioClipToByteArray(source.clip);
File.WriteAllBytes(Application.persistentDataPath + "/MyFile.wav", wavFile);
```

IsWAVFile()

```
public static v IsWAVFile(byte[] wavFile)
```

This method returns true if the provided wavFile (contained into a byte[] array) is PCM WAV compatible.

This example checks the validity of a WAV file:

```
if(IsWAVFile(wavFile))
  print("This is a valid PCM WAV file!!");
else
  print("This is not a PCM WAV file.");
```

Combine()

```
public static AudioClip Combine(AudioClip[] clips)
```

This method picks an array of clips and combines them into one single AudioClip.
It doesn't performs any conversions, so it's preferable to use clips with similar parameters to avoid unwanted results.

StereoToMono()

```
public static AudioClip StereoToMono(AudioClip stereoClip, bool stream = false)
```

This method returns an AudioClip resampled from stereo to mono performing an average mix between both channels.

The method returns a new AudioClip:

```
AudioClip monoClip = OpenWavParser.StereoToMono(stereoClip);
```

MonoToStereo()

```
public static AudioClip MonoToStereo(AudioClip monoClip, bool stream = false)
```

This method returns an AudioClip resampled from mono to stereo duplicating the channel.

The method returns a new AudioClip:

```
AudioClip stereoClip = OpenWavParser.MonoToStereo(monoClip);
```

Known Issues

- The sample scene is compatible with iOS, Android and standalone platforms. To extend the compatibility please purchase FileManagement from the AssetStore.

- Every AudioClip "stream" enabled or "compressed in memory" can't be converted.

- The run-time AudioClip effects will be not saved (they are not present into the clip content).

- In WebGL, AudioClip can't be converted to WAV until Unity version 2021.3 and above.

Contact

If you need some support or if you find some errors in this documentation or the application, don't hesitate on sending me an email to: jmonsuarez@gmail.com

Please, once you have tested this product, take a minute of your time to write a good review in the Unity Asset Store, so you will help to improve this product:

https://assetstore.unity.com/packages/slug/90832

Thanks.