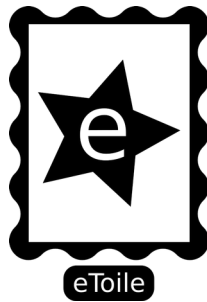


Open WAV Parser

OpenWavParser Class

Cross-platform AudioClip to PCM WAV file codec.



(eToile 2019) V: 1.1

Index

Introduction.....	3
Class Description.....	3
Class Integration.....	3
OpenWavParser Public Interfaces.....	4
ByteArrayToAudioClip().....	4
AudioClipToByteArray().....	4
Known Issues.....	5
Contact.....	5

Introduction

Thanks for downloading `OpenWavParser`, this class is designed to be simple and lightweight, so you will not need to learn how to use a few methods.

This product is just that: a 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 all 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 (.) the interface you need.

There are no 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.

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`.

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)
```

The `clip` argument requires an `AudioClip` with some valid data to be converted to a WAV file (contained into a `byte[]` array).

The converted WAV file gets the `AudioClip` properties to assign the audio format (channels, frequency, etc.).

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);
```

IsCompatible()

```
public static bool IsCompatible(byte[] wavFile)
```

This method returns true if the provided `wavFile` (loaded from disk as `byte[]` array) is a PCM WAV compatible audio file.

This example checks the validity of a WAV file:

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

Known Issues

- The sample scene is compatible with iOS, Android and standalone platforms. To extend the compatibility please buy [FileManagement](#) from the AssetStore.

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

If you find that the test version is not the same version that you downloaded from the AssetStore, please send me your invoice number and I will send you back the last FileManagement version (new version normally is into approval process).

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://www.assetstore.unity3d.com/#!/content/90832>

Thanks.