AUDIO MANAGER

SCRIPTING API

Summary

Asset Version	3.1.x or newer
Unity Version	2020.3.x or newer
Price	FREE
Revision	1
Last Updated (Y/M/D)	2025/04/19

For asset usage, please refer to the documentation pdf.

Contents

Scripting API

Assemblies

If you are using assemblies for your code base, you'll need to reference the audio manager assemblies to access the API of the asset.

Editor > CarterGames.AudioManager.Editor
Runtime > CarterGames.AudioManager.Runtime

The asset also has some shared libraries between assets. If you need to access these, you can do so from these assemblies:

Shared Editor > CarterGames.Shared.Editor.AudioManager
Shared Runtime > CarterGames.Shared.Runtime.AudioManager

Namespace

The main namespace for the asset is CarterGames.Assets.AudioManager

What is Evt?

Evt is a custom class that just wraps an System.Action into a nicer API for me personally. It also handles avoiding over-subscription from a single subscriber.

+= > Add()

-= > Remove()

?.Invoke() > Raise()

API example:

ClassName.ItemName

Description	A summary of what it does.
Type	The type the API is Property/Method etc.
Returns	What if anything the API returns

Parameters / Parameter Variants:

Any parameters or parameter groupings that are required or optional for the API to function. Usually in groups when for example a method has several different overloads for the same method.

Parameter Summaries:

Parameter Name	A description of what the parameter is for.

AudioManager.cs

The main API you'll interact with is the Audio Manager class. This is split into partial classes purely for maintainability. Functionally it'll play no differently to if the class was all one file.

AudioManager.ChangePlayState

Description	Changes the play state of the Audio Manager at runtime.
Type	Method
Returns	void

Parameters:

PlayState playstate

Parameter Summaries:

Playstate	The playstate to set to.

AudioManager.Prepare

Description	Prepares an audio clip player for use, but doesn't call it to play.
Type	Method
Returns	AudioPlayer

Parameters:

```
string request,
params IEditModule[] edits
```

Parameter Summaries:

Request	The audio clip id to play from the audio library.
Edits	Adds any additional edits you want to make to the player before it plays via the edit modules system.

AudioManager.Play

Description	Plays the audio clip requested
Type	Method
Returns	AudioPlayer

Parameter Variants:

```
string request,
params IEditModule[] edits
```

```
string request,
float? volume = 1f,
float? pitch = 1f,
params IEditModule[] edits
```

Parameter Summaries:

Request	The audio clip id to play from the audio library.
Volume	Sets the volume of the player to the defined value.
Pitch	Sets the pitch of the player to the defined value.
Edits	Adds any additional edits you want to make to the player before it plays via the edit modules system.

AudioManager.PlayFromTime

Description	Plays the audio clip requested at the specified start time.
Type	Method
Returns	AudioPlayer

Parameter Variants:

```
string request,
float startTime,
params IEditModule[] edits
```

```
string request,
float startTime,
float? volume = 1f,
float? pitch = 1f,
params IEditModule[] edits
```

Parameter Summaries:

Request	The audio clip id to play from the audio library.
Start Time	The time in the clip length the player show play from. This will override any dynamic start time setup that would otherwise be used.
Volume	Sets the volume of the player to the defined value.
Pitch	Sets the pitch of the player to the defined value.
Edits	Adds any additional edits you want to make to the player before it plays via the edit modules system.

AudioManager.PlayWithDelay

Description	Plays the audio clip requested with a delay to the start of the clip.
Type	Method
Returns	AudioPlayer

Parameter Variants:

```
string request,
float delay,
params IEditModule[] edits
```

```
string request,
float delay,
float? volume = 1f,
float? pitch = 1f,
params IEditModule[] edits
```

Parameter Summaries:

Request	The audio clip id to play from the audio library.
Delay	The delay to before the clip plays after you call for it to play.
Volume	Sets the volume of the player to the defined value.
Pitch	Sets the pitch of the player to the defined value.
Edits	Adds any additional edits you want to make to the player before it plays via the edit modules system.

AudioManager.PlayAtLocation

Description	Plays an audio clip at the requested location.
Type	Method
Returns	AudioPlayer

Parameter Variants:

```
string request,
Vector2 position,
params IEditModule[] edits
```

```
string request,
Vector3 position,
params IEditModule[] edits
```

```
string request,
Transform position,
bool useLocalPosition,
params IEditModule[] edits
```

```
string request,
Vector2 position,
float? volume = 1f,
float? pitch = 1f,
params IEditModule[] edits
```

```
string request,
Vector3 position,
float? volume = 1f,
float? pitch = 1f,
params IEditModule[] edits
```

```
string request,
Transform position,
bool useLocalPosition,
float? volume = 1f,
float? pitch = 1f,
params IEditModule[] edits
```

Continued on next page…

Parameter Summaries:

Request	The audio clip id to play from the audio library.
Position	The position to place the audio player in the scene. **Note:** Position is relative to the parent of the player.
Use Local Position	Defines if the transform input uses local position instead of world position for the value it reads from.
Volume	Sets the volume of the player to the defined value.
Pitch	Sets the pitch of the player to the defined value.
Edits	Adds any additional edits you want to make to the player before it plays via the edit modules system.

AudioManager.PrepareGroup

Description	Prepares a player to play from a group defined in the audio library or an array of clip ids. It doesn't call to play the clips, just prepares the player for use.
Type	Method
Returns	AudioPlayer

Parameter Variants:

```
string request,
params IEditModule[] edits
```

```
string[] request,
GroupPlayMode playMode,
params IEditModule[] edits
```

Parameter Summaries:

Request	The group id or array of audio clip ids to play from the audio library.
PlayMode	The group play method to use. Default is a random clip from the group.
Edits	Adds any additional edits you want to make to the player before it plays via the edit modules system.

AudioManager.PlayGroup

Description	Play a group defined in the audio library or an array of clip ids.
Type	Method
Returns	AudioPlayer

Parameter Variants:

```
string request,
params IEditModule[] edits
```

```
string request,
float? volume = 1f,
float? pitch = 1f,
params IEditModule[] edits
```

```
string[] request,
GroupPlayMode playMode,
params IEditModule[] edits
```

```
string[] request,
GroupPlayMode playMode,
float? volume = 1f,
float? pitch = 1f,
params IEditModule[] edits
```

Parameter Summaries:

Request	The group id or array of audio clip ids to play from the audio library.
PlayMode	The group play method to use. Default is a random clip from the group.
Volume	Sets the volume of the player to the defined value.
Pitch	Sets the pitch of the player to the defined value.
Edits	Adds any additional edits you want to make to the player before it plays via the edit modules system.

AudioManager.PlayGroupWithDelay

Description	Play a group defined in the audio library or an array of clip ids with a delay before the group plays.
Type	Method
Returns	AudioPlayer

Parameter Variants:

```
string request,
float delay,
params IEditModule[] edits
```

```
string request,
float delay,
float? volume = 1f,
float? pitch = 1f,
params IEditModule[] edits
```

```
string[] request,
GroupPlayMode playMode,
float delay,
params IEditModule[] edits
```

```
string[] request,
GroupPlayMode playMode,
float delay,
float? volume = 1f,
float? pitch = 1f,
params IEditModule[] edits
```

Continued on next page…

Parameter Summaries:

Request	The group id or array of audio clip ids to play from the audio library.
PlayMode	The group play method to use. Default is a random clip from the group.
Delay	The delay to before the clip plays after you call for it to play.
Volume	Sets the volume of the player to the defined value.
Pitch	Sets the pitch of the player to the defined value.
Edits	Adds any additional edits you want to make to the player before it plays via the edit modules system.

AudioManager.PlayGroupAtLocation

Description	Play a group defined in the audio library or an array of clip ids, at the specified location.
Type	Method
Returns	AudioPlayer

Parameter Variants:

```
string request,
Vector2 position,
params IEditModule[] edits
```

```
string request,
Vector3 position,
params IEditModule[] edits
```

```
string request,
Transform position,
bool useLocalPosition,
params IEditModule[] edits
```

```
string request,
Vector2 position,
float? volume = 1f,
float? pitch = 1f,
params IEditModule[] edits
```

```
string request,
Vector3 position,
float? volume = 1f,
float? pitch = 1f,
params IEditModule[] edits
```

```
string request,
Transform position,
bool useLocalPosition,
float? volume = 1f,
float? pitch = 1f,
params IEditModule[] edits
```

Continued on next page.

```
string[] request,
GroupPlayMode playMode,
Vector2 position,
params IEditModule[] edits
```

```
string[] request,
GroupPlayMode playMode,
Vector3 position,
params IEditModule[] edits
```

```
string[] request,
GroupPlayMode playMode,
Transform position,
bool useLocalPosition,
params IEditModule[] edits
```

```
string[] request,
GroupPlayMode playMode,
Vector2 position,
float? volume = 1f,
float? pitch = 1f,
params IEditModule[] edits
```

```
string[] request,
GroupPlayMode playMode,
Vector3 position,
float? volume = 1f,
float? pitch = 1f,
params IEditModule[] edits
```

```
string[] request,
GroupPlayMode playMode,
Transform position,
bool useLocalPosition,
float? volume = 1f,
float? pitch = 1f,
params IEditModule[] edits
```

Continued on next page…

Parameter Summaries:

Request	The group id or array of audio clip ids to play from the audio library.
PlayMode	The group play method to use. Default is a random clip from the group.
Position	The position to place the audio player in the scene. **Note:** Position is relative to the parent of the player.
Use Local Position	Defines if the transform input uses local position instead of world position for the value it reads from.
Volume	Sets the volume of the player to the defined value.
Pitch	Sets the pitch of the player to the defined value.
Edits	Adds any additional edits you want to make to the player before it plays via the edit modules system.

AudioManager.GetClipIdsWithTag

Description	Returns a list of all the clip ids that have the entered tag assigned to them in the audio library.
Type	Method
Returns	List<string>

Parameters:

string tag

Parameter Summaries:

tag	The tag to look for on all clips in the library.

AudioManager.GetAudioDataWithTag

Description	Returns a list of all the clip audio data that have the entered tag assigned to them in the audio library.
Type	Method
Returns	List<AudioData>

Parameters:

string tag

Parameter Summaries:

tag	The tag to look for on all clips in the library.

AudioPlayer.cs

The audio player class is a base class for any audio player from the audio manager. Other classes inherit this to play audio in specific setups. The API is the same regardless of which play method you are using.

AudioPlayer.Source

Description	Returns the audio source the player is attached to.
Type	Property
Returns	AudioSourceInstance

AudioPlayer.AdditionalSources

Description	Returns any additional audio sources that the player is attached to.
Type	Property
Returns	List<AudioSourceInstance>

AudioPlayer.AllSources

Description	Returns the audio source the player is attached to.
Type	Property
Returns	AudioSourceInstance

AudioPlayer.IsPlaying

Description	Returns if the audio player is currently playing audio or not.
Type	Property
Returns	bool

AudioPlayer.Started

Description	Is raised when the audio player has started playing audio.
Type	Evt (Use Add() to sub & Remove() to un-sub)
Returns	Evt

AudioPlayer.Looped

Description	Is raised when the audio player has completed a loop.
Type	Evt (Use Add() to sub & Remove() to un-sub)
Returns	Evt

AudioPlayer.Completed

Description	Is raised when the audio player has completed playing audio.
Type	Evt (Use Add() to sub & Remove() to un-sub)
Returns	Evt

AudioPlayer.Stopped

Description	Is raised when the audio player has stopped playing audio.
Type	Evt (Use Add() to sub & Remove() to un-sub)
Returns	Evt

AudioPlayer.Paused

Description	Is raised when the audio player has stopped playing audio.
Type	Evt (Use Add() to sub & Remove() to un-sub)
Returns	Evt

AudioPlayer.Resumed

Description	Is raised when the audio player has stopped playing audio.
Type	Evt (Use Add() to sub & Remove() to un-sub)
Returns	Evt

AudioPlayer.Play

Description	Plays the audio from the player when called.
Type	Method
Returns	void

AudioPlayer.Pause

Description	Pauses the audio from the player when called.
Type	Method
Returns	void

AudioPlayer.Resume

Description	Resumes the audio from the player when called.
Type	Method
Returns	void

AudioPlayer.Stop

Description	Stops the audio from the player when called.
Type	Method
Returns	void

IEditModule.cs

Implement this interface to create your own edit modules to use with the audio manager.

IEditModule.ProcessOnLoop

Description	Defines if the edit should reapply when the clip loops.
Type	Property
Returns	bool

IEditModule.Process

Description	Processes the edit onto the source.
Type	Method
Returns	void

Parameters:

AudioSourceInstance source

Parameter Summaries:

Source	The audio source instance the edit should apply to. Use to make your edits to the source.

IEditModule.Revert

Description	Reverts the edit on the source.
Type	Method
Returns	void

Parameters:

AudioSourceInstance source

Parameter Summaries:

Source	The audio source instance the edit should apply to. Use to make your edits to the source.

Example IEditModule Implementation

```
public sealed class MuteEdit : IEditModule
{
   private bool isMuted;


   /// <summary>
   /// Gets if the edits should process when looping
   /// </summary>
   public bool ProcessOnLoop => false;


   /// <summary>
   /// Processes the edit when called.
   /// </summary>
   /// <param name="source">The AudioSource to edit.</param>
   public void Process(AudioSourceInstance source)
   {
       source.Source.mute = isMuted;
   }


   /// <summary>
   /// Revers the edit to default when called.
   /// </summary>
   /// <param name="source">The AudioSource to edit.</param>
   public void Revert(AudioSourceInstance source)
   {
       source.Source.mute = UtilRuntime.SettingAudioPlayState !=
PlayState.PlayMuted
;
   }

   /// <summary>
   /// Makes a new mute edit with the setting entered.
   /// </summary>
   /// <param name="value">The value to set to.</param>
   public MuteEdit(bool value)
   {
       isMuted = value;
   }
}
```