Audio Manager 3.x Offline Docs

This is an offline copy with the bare bones you need to use the asset. For full docs, please see the online copy. These can be found here:

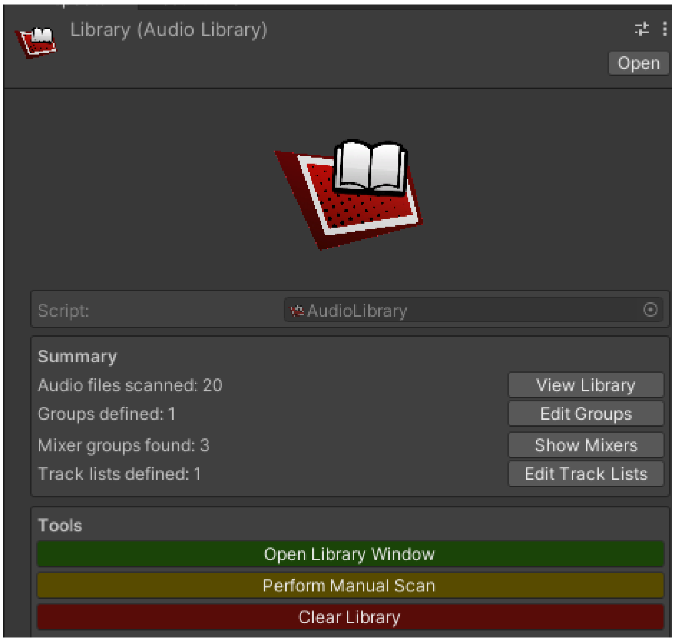
Audio Manager - Carter Games

🎮 https://carter.games/audiomanager/

🌟 Thanks

Thank you for deciding to use my asset for your project. If you like my asset, feel free to leave a ⭐⭐⭐⭐⭐ review! If you find that our asset is not up to scratch or find and issue, please do let me know either via our email: **hello@carter.games** and I will do my best to help you with the issues you are facing. I can't read minds, so if you don't speak up, it won't get fixed 😆

Library Management



Script: AudioLibrary

Summary

Audio files scanned: 20

Groups defined: 1

Mixer groups found: 3

Track lists defined: 1

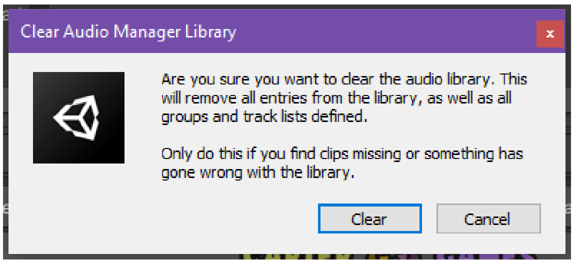
Tools

When you import or delete any audio clip from your project, the audio manager will automatically process it into the library for you. Once in the library you can find the clip in editor window for the audio library. If you have just installed the asset for the first time a dialogue window will appear asking you to perform the first scan for audio. Afterwards any other changes will be automatic.

If the automated process doesn't work for some reason, the library can be manually scanned by an option in the navtab for the asset or via the library object inspector:

> Tools/Carter Games/Audio Manager/Perform Manual Scan

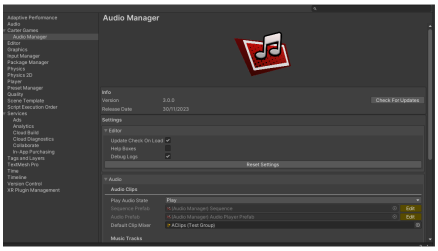
If however everything has gone wrong or nothing is been updated, even with a manual scan, you may need to do a clean library scan. You can access this from the library object inspector. Doing this will open a dialogue that asks you if you want to clear the library and will also clear groups and track lists, so be sure to make a backup before doing this.



Are you sure you want to clear the audio library. This will remove all entries from the library, as well as all groups and track lists defined.

Only do this if you find clips missing or something has gone wrong with the library.

Asset Settings



Info
Version 3.0.0
Release Date 30/11/2023

Settings
Editor
Update Check On Load ☑
Help Boxes □
Debug Logs ☑

Audio
Audio Clips
Play Audio State: Play
Sequence Prefab: (Audio Manager) Sequence
Audio Prefab: (Audio Manager) Audio Player Prefab
Default Clip Mixer: AClips (Test Group)
Music Tracks

The audio manager has all of its runtime and editor settings shown in the settings provider for the asset. This can be found in `Project Settings → Carter Games → Audio Manager`

Info Section

The info section displays the version of the asset your are currently using and the date it was released on. If you want to make sure you have the latest version of the asset, you can do so by pressing the `Check For Updates` button on the far right hand side of this section. A dialogue will appear briefly after you press this button letting you know if you have the latest version or not. If you don't have the latest, a link will be provided to get the latest update from GitHub.

Note: The check for updates button requires an internet connection to work.

Settings Explained

Editor Settings

All of the editor settings are hidden from view as they are settings that should be applied on a per user basis. These can be reset from the navbar menu or from here should you wish. However a few are exposed here you you can toggle them with ease:

Setting	Description	Default Value
Version Check On Load	Checks for update on the asset's GitHub repository when you open the Unity project with the asset in to. You'll see a dialogue if there is an update.	true
Help Boxes	Defines if additional help boxes appear in the editor of the audio library to help explain how it works.	false
Debug Logs	Defines if the asset throws any intentional logs to the user. Disable to clear up the logs a tad.	true

Audio Settings

Setting	Description	Default Value
Default Clip Mixer	Defines a mixer that is applied to clip play calls when any clip is called to play and is not overridden by a mixer edit module.	null
Default Music Mixer	Defines a mixer that is applied to music play calls when any clip is called to play and is not overridden by a	null

Setting	Description	Default Value
	mixer edit module.	
Audio Play State	Defines the state for audio clips to play in. Can be set to “Play”, “Play Muted” & “Disabled” ”Play” - Plays all clips as normal. ”Play Muted” - Plays all clips, but has them muted. ”Disabled” - No clips are played regardless of calls to do so from the manager.	Play
Music Play State	Defines the state for music tracks to play in. Can be set to “Play”, “Play Muted” & “Disabled” ”Play” - Plays all clips as normal. ”Play Muted” - Plays all clips, but has them muted. ”Disabled” - No clips are played regardless of calls to do so from the manager.	Play
Audio Pool Init Size	Defines the size of the audio clip pool when first initialized. It does auto expand so you should never run out of elements.	5
Global Variance Volume Variance	Defines the global offset from the volume used in an clip. If global variance is turned off for the clip play call this does nothing.	0.1
Global Variance Pitch Variance	Defines the global offset from the pitch used in an clip. If global variance is turned off for the clip play call this does nothing.	0.1
Dynamic Time Dynamic Start Offset	The amount of offset applied to the automatic start time detection from where it detects the start time. Only applies to automatic detection, if manual is used this is ignored.	50

Library Editor Window

The audio manager library window let you see and manage all of the audio in your project. You can group clips together, setup music tracks & playlists and edit the start time and keys of clips used by the manager. The window can be accessed through the following path:

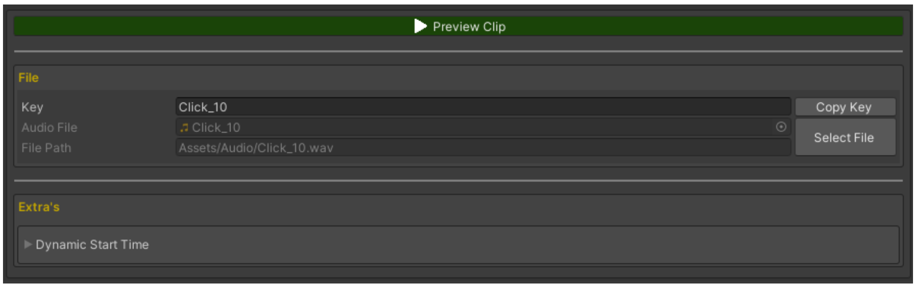
`Top Nav Bar → Tools → Carter Games → Audio Manager → Library Editor`

Library Tab

The library tab lets you browse all the audio in your project that the audio manager has scanned. A list of all the clips with buttons for each entry will be on the left side, if you have a lot of clips you can search the library to select a particular clip by name. Likewise you can just press the button for the entry to view it.

Clip Details

Once selected you can view all the details about that clip in the library. This includes the key that is used for the clip helper struct, the audio file reference & path as well as a way to preview the clip.



Key: Click_10
Audio File: Click_10
File Path: Assets/Audio/Click_10.wav

You can customize the key if you wish to be anything you like. Note that the clip struct helper class doesn't update automictically, so you'll need to update it manually if you change the key of a clip.

Extra's

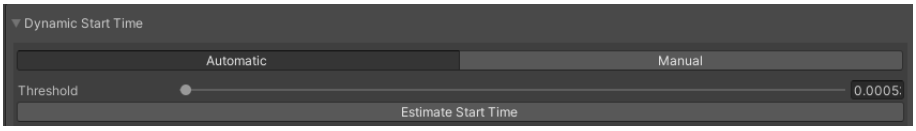
Dynamic Time

Time module lets you edit the start time of the clip when you call it to play through the Audio Manager to play at a time near or on the moment the audio actually starts. This helps when your clips have dead space at the start of the clip and make the audio more responsive. You can choose to toggle the use of dynamic time on or off in globally as well as per call by adding a dynamic time edit to the audio play call.

There are two modes for this setup, automatic & manual. This is pretty self explanatory in itself:

Automatic Mode

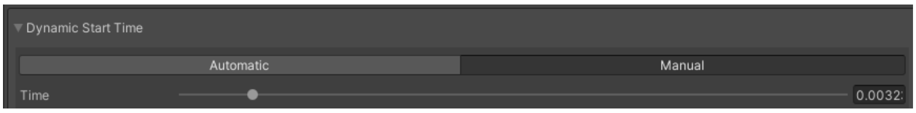
The automatic setup tries to work out where the clip starts and places the start point a little before that. It can be inaccurate so you may need to adjust the threshold slider and re-estimate the start time to get it on the dot.



Threshold: 0.0005:

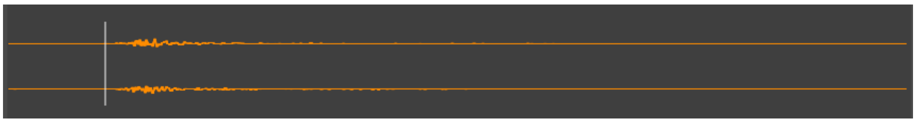
Manual Mode

If the automatic setup isn't working out for you, the manual option lets you just drag a slider along which will set the start time to wherever you set it to.



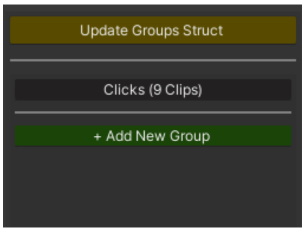
Time: 0.0032:

Both of these setups have a preview of the clips waveform underneath it where you can visually see where the start time is on the clip.



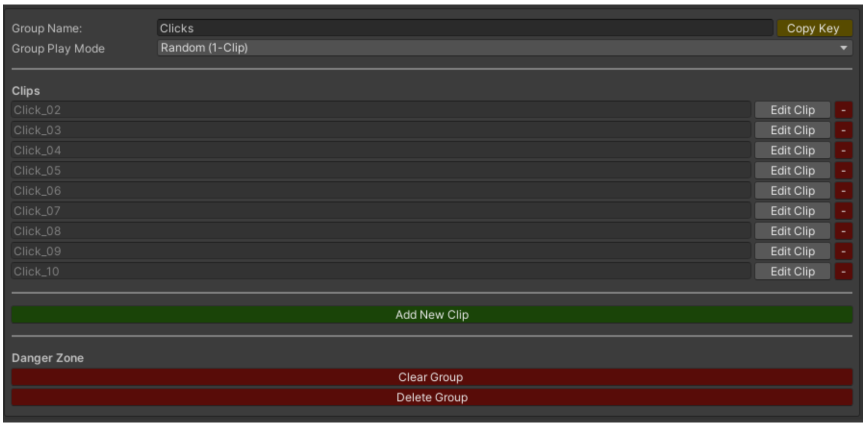
Group Tab

The group tab lets you define collections of clips from the library into a container that can be called by the audio manager. This can be used to play a random clip from a group, like footsteps or clicks, a sequence of clips one after the other or a group of clips at the same time.



The left side holds all the groups you have defined already with the option to make a new one should you wish.

When a group is selected you'll be able to see all its properties on the right side



Group Name: Clicks
Group Play Mode Random (1-Clip)

Clips
Click_02
Click_03
Click_04
Click_05
Click_06
Click_07
Click_08
Click_09
Click_10

Danger Zone

Name

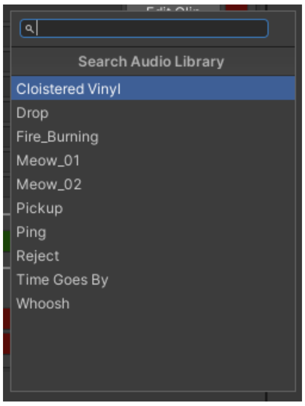
⚠️ Groups **MUST** have unique names.

Defines the name that the group is referred to in the audio manager when calling for the group to play. Can be used in its pure string form or via the group helper struct.

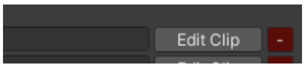
Playmodes

Random (1-Clip)	Plays a single random clip from the group when the group is called to play.
Sequential (All-Clips)	Plays all the clips in the group one after the other when the group is called to play.
Combined (All-Clips)	Plays all the clips at the same time when the group is called to play.

Clip Selecting



You can select a clip by just pressing the add clip button and search for the clip by name. Once you've found the clip you want, just select it to apply it to the group.



If you need to remove a entry or change the clip for an entry from the group, you can do so with the edit clip or - buttons on that entry.

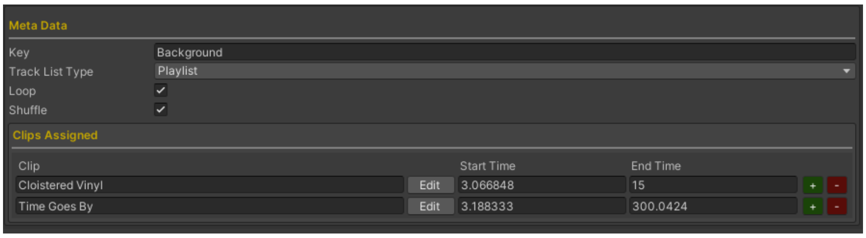
Danger Zone

The danger zone lets you delete or clear the group data. These options have an extra dialogue prompt to avoid mis-clicks.

Music Tab

The music tab lets you define clips in your library as music to the music management side of the asset. Tracks are defined like groups but with some extra options to change the start and end time of each track, overriding the start and end times on the normal clip play.

Example



Meta Data

Key: Background

Track List Type: Playlist

Loop: ☑

Shuffle: ☑

Clips Assigned

Clip		Start Time	End Time		
Cloistered Vinyl	Edit	3.066848	15	+	-
Time Goes By	Edit	3.188333	300.0424	+	-

Key

Defines the name the track list is referred to in the music manager. This is also used for the track helper struct.

Track list modes

Single	Defines a single track playlist. Ideal for looping tracks.
Playlist	Defines a playlist of tracks that play either in order or randomly dependant on your setup.

> 💡 Note: the single track list mode will ignore any additional tracks other than the first one assigned.

The other options, loop & shuffle are self explanatory. If you turn on loop, it will loop the track list when it has ended instead of stopping. If shuffle is on it will play the tracks in a random order instead of as listed.

For each clip you can also customize the start and end times. When a clip is used as music it ignores the dynamic start time setup and uses these values instead. You can see the start and end points by clicking the view waveform button on each entry of a track list.

Audio Manager - Play

Declaration

```
public static AudioSequence Play(string request, params IEditModule[] edits)
public static AudioSequence Play(string request, float? volume = 1f, float? pitch = 1f, params IEditModule[] edi
```

Parameters

`string` request	The key for the group to be played.

params `IEditModule[]` edits	An array of edits to apply to the clip. Is defines as params, so the elements do not needs to be in an array when defined.

`string` request	The key for the group to be played.
`float` volume	The volume to set the clip to.
`float` pitch	The pitch to set the clip to.
params `IEditModule[]` edits	An array of edits to apply to the clip. Is defines as params, so the elements do not needs to be in an array when defined.

Returns

`AudioSequence`

The audio sequence that the clip is being played on.

Description

Sets up an `AudioSequence` to play with all the options provided and plays it when ready. Returning the `AudioSequence` its playing from for you to use should you need the reference.

Example

```
private void OnEnable()
{
    // Plays the clip with no user edits.
    AudioManager.Play("MyClip");

    // Plays the clip with edits to volume.
    AudioManager.Play("MyClip", .5f);

    // Plays the clip with edits to volume via edit modules.
    AudioManager.Play("MyClip", new VolumeEdit(.5f));
}
```

Music Manager - Play

Declaration

```
public static void Play(string id, float volume = 1f)
public static void Play(MusicTrackList trackList, volume = 1f)
```

Parameters

`string` id	The track list id to play.
`float` volume	The volume to play at.

Description

Sets up an `IMusicPlayer` with the track list desired and plays it.

Example

```
private void OnEnable()
{
    // Plays the track list when called.
    MusicManager.Play("MyTrackList");
}
```

Audio Edit Modules

Behind the scenes, each edit to a setting for a clip play call uses an edit module to apply and revert the change. You can use these modules to modify the play calls further than any pre-defined call setup in the manager. All pre-defined calls use this same setup but are slightly nicer on the API for the end user.

To apply an edit you just need to pass through its constructor at the end of the play method call. All play methods support editor modules as the last parameter. Below are all the modules, what parameters they take and what they do:

Volume Edit

Description

Sets the volume of the clip within a range of 0 → 1 on the audio source the clip plays on.

Parameters

Standard

Name	Type	Description	Default
volume	`float`	The value to set the volume as.	

Range

Name	Type	Description	Default
minVolumeValue	`float`	The min value the volume can be set to.	
maxVolumeValue	`float`	The max value the volume can be set to.	

Variance

Name	Type	Description	Default
variance	`Variance`	The variance to set the volume from.	

Pitch Edit

Description

Sets the pitch of the clip within a range of -3 → 3 on the audio source the clip plays on.

Parameters

Standard

Name	Type	Description	Default
pitch	`float`	The value to set the pitch as.	

Range

Name	Type	Description	Default
minPitchValue	`float`	The min value the pitch can be set to.	
maxPitchValue	`float`	The max value the pitch can be set to.	

Variance

Name	Type	Description	Default
variance	`Variance`	The variance to set the pitch from.	

Mixer Edit

Description

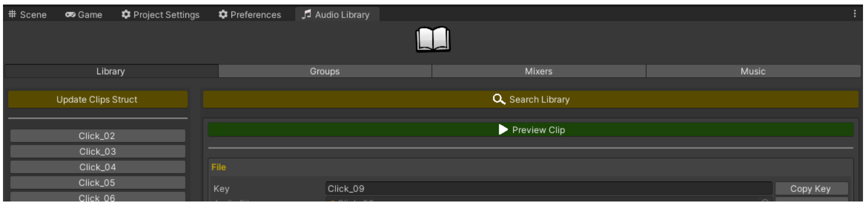
Sets the mixer group that is used for the audio source the clip is using for your request.

Parameters

Name	Type	Description	Default
mixerRef	`string`	The mixer name as defined in the library. Either as a raw string or via the Mixer.??? helper struct class.	

Name	Type	Description	Default
mixerGroup	`AudioMixerGroup`	The mixer to set.	

Struct Helper Classes



Key: Click_09

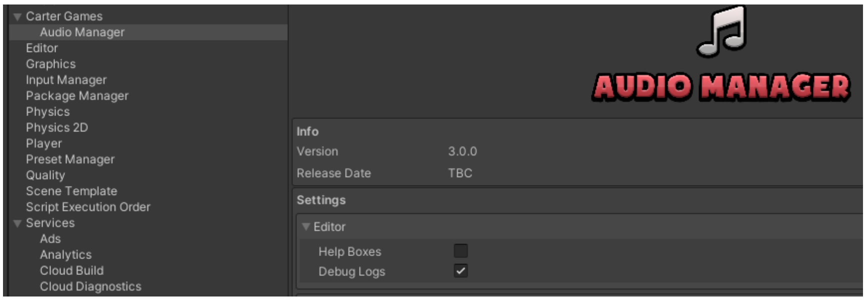
On every tab of the audio library editor there is an option to generate a helper struct. This struct when generated makes a static entry for each element in that tab that you can use in your code instead of a string reference to help avoid typo's and give you some useful autofill. The value that these structs pass in will vary per tab, some are the matching name while others will be id's.

When pressing the update struct button for a helper struct, the class will be totally re-generated. This may take a while and scripts will be reloaded once it is done. The generation code should catch most failures and log them under the assets logs as a warning so you know if any entries failed to add and why.

Class Name	Description
Clip	Holds the names of all the clips in the audio library.
Group	Holds the ids of all the groups the user has defined in the audio library.
Mixer	Holds the names of all the mixers in the project the audio library has found.
Track	Holds a name reference for all the track lists stored in the audio library.

Logs

The asset has its own logging setup using the normal `Debug.Log` in the backend. All logs from that asset that are intentional can be disabled from the settings provided in the Audio Manager's settings provider. This can be found in Project Settings → Carter Games → Audio Manager under the editor dropdown.



Info

Version 3.0.0

Release Date TBC

Settings

Editor

Help Boxes □

Debug Logs ☑

The asset will use all types of logging dependant on the severity of the issue. Any errors will come with a unique error code which will be displayed in the log message if such an issue arises. You can see all these codes on the ▲ Error Codes page.