



## MASTER AUDIO PLUGIN V3.1 - By Dark Tonic, Inc. (c) 2013

### Table of Contents

1. Solutions! Master Audio solves the following problems	2
2. Quick Start: How to set up your scene to utilize Master Audio	2
3. Triggering the Audio - "no code" methods	5
4. Triggering the Audio - code methods	9
5. Controlling the Audio - code methods	11
6. Audio Groups: Fine-tuning through the Inspector panel	11
7. Master Audio: Advanced Options	15
8. Audio FX Filters: Reverb, chorus etc.	17
9. Playlist Controller and Music Playlists	17
10. Dynamic Creation and Modification of Sound Groups	20
11. Playmaker integration	22
12. 2D Toolkit integration	23
13. Final Words	24

This code was written to be the be-all end-all for video game audio management! We are always open to hearing your ideas for improvements, suggestions and problems. Email us any time at [support@darktonic.com](mailto:support@darktonic.com)

Demo video at: <http://bit.ly/17saxEf> (under five minutes and moves briskly).

Full undo support was added with V3.0. It uses the Unity 4.3 new Undo API. Master Audio still supports Unity 3.5.7, but you do not get full undo support with that version.

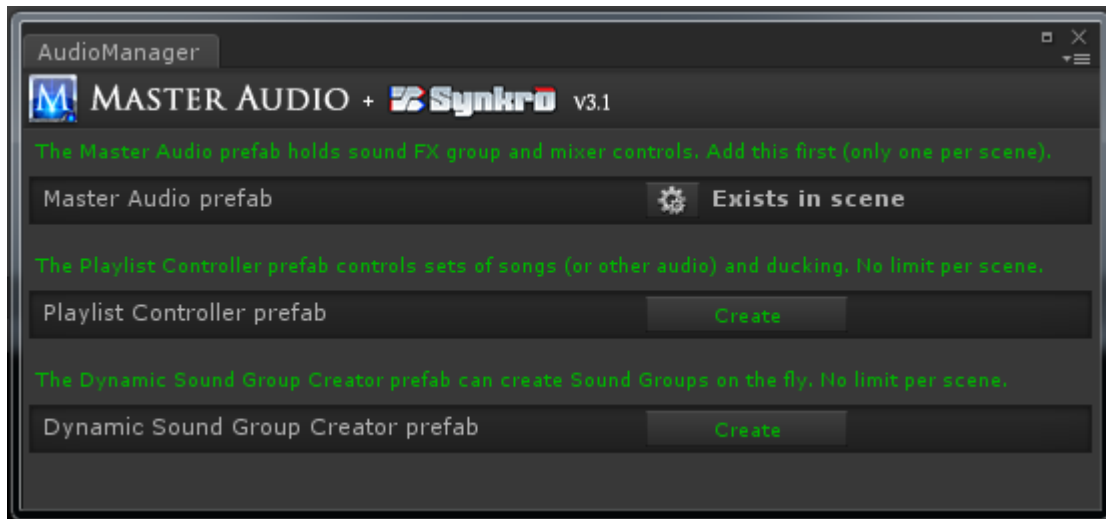
Most of the code options are listed in various parts of this document. The **entire** Master Audio API documentation can be found here: <http://bit.ly/1bkiRei>

## 1. Solutions! Master Audio solves the following problems (and a lot more)

- 1) Too many instances of the same audio clip playing simultaneously or near-simultaneously. For example, if an enemy has a death scream sound, you may kill 30 of them with a single blow. 30 audio clips playing simultaneously is not only unnecessary but it drags the CPU down, especially on mobile devices. Master Audio lets you specify the maximum number of each sound that can be playing at a single time.
- 2) The ability to randomize a certain sound to actually play from a pool of weighted variation sounds. This goes hand in hand with setting up the maximum number of each sound that can be played. All X sounds in a group can be the same, or variations. You have complete control over that.
- 3) Having to write code to trigger each sound. Master Audio eliminates the need for this in most cases by letting you specify sounds to play when certain events occur. No coding is needed to do this.
- 4) Not being able to play sounds when the AudioSource is attached to a game object that is being despawned or destroyed. What happens normally is that you hear a brief blip of the sound and then when the game object is gone, the sound abruptly stops. Master Audio keeps all its Audio Sources in a central location separate from your prefabs so that this doesn't happen.
- 5) Loading an audio clip from a Resource file, playing it, and then unloading it from memory so you don't have the memory taken up until the whole time. All with no coding!
- 6) Being able to stop all currently playing instances of a certain sound. Since Master Audio knows where all its Audio Sources are at all times, it can trivially do this with one simple line of code.
- 7) Adjusting the volume of categories of sound effects with a slider. Unity does not have anything natively, we have included the concept of a pro mixer with buses. More on this later!
- 8) Music ducking. You can configure Master Audio to have the music "duck" (meaning get quieter and ramp back up) for whichever sound(s) you like with no coding needed. The amount of ducking is also configurable.
- 9) Music cross-fading. You can set up multiple music Playlists that picks tracks that can shuffle, cross-fade and auto advance. You can have any number of Playlist Controllers, each playing a Playlist and with cross-fading!
- 10) Not being able to play music during a scene change. Simple to do with Master Audio!

## 2. Quick Start: How to set up your scene to utilize Master Audio

- 1) From the Unity "Window" menu, select "Audio Manager". This is a small window that will help explain and set up what you need for Master Audio. It looks like this:



- 2) First, create the Master Audio prefab with the "Create" button. The position of the MasterAudio prefab will be used for all triggered 2D sounds (if you have any), so put it somewhat close to the AudioListener in your scene if you have any 2D sounds planned.
- 3) To use music Playlists and cross-fading, you can create 1 or more Playlist Controller prefabs with its "Create" button as well.
- 4) Dynamic Sound Group Creator is covered in its own section (Section 10) later in this document. It is used for per-scene audio and lets you specify them easily.
- 5) Configuring your first Sound Group.
  - a. Click on the MasterAudio prefab in the Hierarchy. Your Inspector should now look something like this:



- b. We're going to use the Group Mixer section to create our first Sound Group.
  - i. Drag your Scream audio clip into the colored rectangle area that says "Drag Audio clips...". It will automatically create a Sound Group for you that has controls below that.
- c. Now your Inspector should look something like this:



- d. Notice the MasterAudio logo up top in most Inspectors. You can click on it to navigate to the Master Audio prefab in the Scene as a shortcut. Also notice the speaker and stop icons. These will appear in many places in Master Audio and allow you to preview the audio clip, as well as stop previewing.
- e. Notice that "Scream" now shows up under the last section. If you expand the MasterAudio prefab, you will see it now has a child prefab called Scream, and if you expand that, you will see a child prefab for each variation (only one this time).
- f. Each of the variations has an Audio Source component where you can individually tweak the pitch / pan / etc to create different variations. You can also add effects such as Reverb or Distortion to individual variations. Or you can drag entirely different Audio Clips in there for variations as well.

**Note:** the prefab template for variations can be found in the MasterAudio/Sources/Prefabs folder and is called GroupVariation. If you change any properties on the Audio Source in this prefab, it will carry over into all Master Audio variations created afterward. So change it there first after you get the Doppler settings (and any others) working to your liking.

- g. The Sound Groups listed in the "Show Group Mixer" section have a couple buttons for each group.
  - i. Go – clicking this will select the Sound Group in the Hierarchy so you can make additional changes.
  - ii. Delete icon – clicking this will delete the Sound Group. Sometimes you might want to delete it and recreate it with a different number of children. It's just faster that way.
  - iii. "S" for Solo. If any Sound Groups are soloed, only the soloed groups will be heard.
  - iv. "M" for Mute. This will mute the Sound Group. It will produce no audio while muted.

## 6) Configuring your additional Sound Groups.

- a. Drag your "blast" sound clip into the Audio Clip field. It will automatically create the "Blast" Sound Group.

- b. Click the gear (settings) icon on the mixer row for Blast. It will now take you to the Group settings.
- c. Change the Weight field to 6. This means that 6 Blast sounds will be able to be played simultaneously. More on this later.
- d. Click back to the Master Audio prefab. Note the dropdowns above the mixer drag area. The first one is labeled "Bulk Creation Mode". This is for speed Sound Group and Variation creation. You can click the lock icon at the very top right of the Inspector to enable selection of multiple clips from Project view without losing focus on the Inspector to use bulk mode. There are two choices in the dropdown:
  - i. One Group Per Clip (the default) . Each clip will create a new Sound Group with 1 variation, which is the clip.
  - ii. One Group With Variations. This will create a single Sound Group and each clip will become a Variation of the clip.
- e. Bulk Variation Mode - this is the 2nd dropdown and lets you choose between Resource File and Clip.
  - i. Resource File - if you choose this, all Variations will be created with the Audio only available as you need it, not created when the Scene starts. It will also unload from memory after it is no longer being played.  
  
**Note:** when Resource File is chosen, you can drag a clip into a drag area that appears underneath to use its file name. This drag area appears everywhere that Resource Files can be used in Master Audio.
  - ii. Clip - if you choose this, the clip will be loaded immediately when the Scene starts.

### 3. Triggering the Audio - "no code" methods

Now that you have sounds set up, let's see how to trigger them automatically. There are a few scripts to help out on this:

1. **ButtonClicker.cs** is a script that works with NGUI only. It triggers two MasterAudio Sound Groups. One for when you click the mouse button down, and one for when you release. If you have an NGUI button with a collider, go ahead and attach this script to it. You will notice that the Inspector has two dropdowns in the ButtonClicker section. They each contain the list of Sound Groups in MasterAudio.cs. Go ahead and choose Scream and Blast for the two sounds

and try it out by clicking the play button. Now it automatically triggers the sounds for you on these events.

2. **EventSounds.cs** is a powerful and flexible script that you can attach to prefabs to trigger MasterAudio sounds (and manipulate Playlists, Sound Groups and Buses) for certain MonoBehavior, PoolManager and other events. So even without Playmaker, you have a wide variety of "no code" actions you can do with Master Audio. There are events for:
  - a. OnStart
  - b. OnBecameVisible
  - c. OnBecameInvisible
  - d. OnEnable
  - e. OnDisable
  - f. OnCollision2D (this and the next two are only available on Unity 4.3+)
  - g. OnTriggerEnter2D
  - h. OnTriggerExit2D
  - i. OnCollisionEnter
  - j. OnTriggerEnter
  - k. OnTriggerExit
  - l. OnMouseEnter
  - m. OnMouseDown (Mouse Click is what it's called).
  - n. For PoolManager users, we also have OnSpawned and OnDespawned. To get these to show up, you must check the "PoolManager Events" checkbox.

**Note:** OnBecameVisible (and OnBecameInvisible) will only work inside a prefab that has a Renderer component inside it. In cases where batching will reassign or not use the Renderer (NGUI / 2D Toolkit, etc), you may opt to use the OnEnable / OnDisable / OnStart events instead (or the Pool Manager events). They don't provide exactly the same functionality but it will work for most purposes.

**Minimal Mode** - All events are shown at once, but if you want to only show events you're using, check the Minimal Mode checkbox and add events from the "Event to Activate" dropdown.

Event Sounds looks something like this:



You will see a section for each event type in the Inspector if you attach EventSounds to one of your prefabs. Each section has:



Action Type - with 4 choices.

- i. Play Sound -the default. Play a sound, 2d or 3d, according to your Sound Spawn Mode setting.
  - a. Sound Group dropdown (as before).
  - b. Volume control to make the sound quieter
  - c. Delay Sound (seconds) - this allows you to schedule a sound to be played X seconds from now. **Now available on Unity V3.5.7 as well!**
- ii. Group Control - choosing this will reveal a menu of Group Commands to perform various Sound Group actions, some of which have an additional field or two. There is also a checkbox "Do For Every Group?", which if checked will perform the command on every Sound Group.
  - a. Fade To Volume
  - b. Fade Out All Of Sound
  - c. Mute
  - d. Pause
  - e. Solo
  - f. Stop All of Sound
  - g. Unmute
  - h. Unpause
  - i. Unsolo
- iii. Bus Control - choosing this will reveal you a menu of Bus Command to perform various Bus actions, some of which have an additional field or two. There is also a checkbox "Do For Every Bus?", which if checked will perform the command on every Bus.
  - a. Fade to Volume
  - b. Mute
  - c. Pause
  - d. Solo

- e. Unmute
  - f. Unpause
  - g. Unsolo
- iv. Playlist Control - choosing this will reveal a menu of Playlist Commands to perform various Playlist actions, some of which have an additional field or two. There is also a checkbox "All Playlist Controllers?", which if checked will perform the command on every Playlist Controller. Not every command has this option.
  - a. Change Playlist (by name)
  - b. Fade to Volume
  - c. Play Clip
  - d. Play Random Song
  - e. Play Next Song
  - f. Pause
  - g. Resume
  - h. Stop

**Note:** if you have more than one Playlist Controller, you will need to select a Playlist Controller from the dropdown or select the "All Playlist Controllers" checkbox so Master Audio knows what to do.

If you have a Shuriken particle system attached to this object, you can emit particles as well with the other two properties there.

Additionally, the Trigger and Collision events have layer and tag filters. If you enable these, you can specify which layer(s) and / or tag(s) the object you're colliding with must be to trigger the sound.

At the top are the following Group Controls:

- i. Sound Spawn Mode - 3 possible settings.
  - a. Master Audio Location: The sound will emanate from MasterAudio's position.
  - b. Caller Location: This will trigger the sound in 3D from the prefab's position.

- c. **Attach To Caller:** This will not actually reparent the variation prefab, but it will follow the location of the prefab that has the Event script. This way sounds won't get cut off or Variation objects destroyed when things despawn or get destroyed by Scene changes.
  - ii. **Disable Sounds:** Checking this will disable all event sounds on this prefab.
3. **EventCalcSounds.cs** is a script just like **EventSounds.cs**, with slightly more CPU-intensive operations. It has one event type:
- i. **AudioSourceEnded** - This is only usable when you have an AudioSource component with a sound/music on your prefab. If you do, this can trigger a MasterAudio sound every time the AudioSource finishes playing. If your AudioSource is looped, this will keep happening every time the sound loops again.

#### 4. Triggering the Audio - code methods

If you need to trigger any Sound Groups during times other than those provided by the included scripts, you can use the following single lines of code:

*MasterAudio.PlaySound(string soundGroupName, float volumePercentage, float? pitch, float delaySoundTime, string variationName);*

This plays the sound from the position of MasterAudio. All parameters after the 1st are optional. Volume percentage lets you play a lower volume version (0-1 is the range). Pitch, if specified, let you override the chosen Variation's pitch and random pitch and use the pitch parameter instead. Variation Name is optional and lets you play a specific variation (or its clones created from Weight >1) by name. DelaySoundTime lets you schedule a sound to be played X seconds from now (Unity 4.X+ only).

*MasterAudio.PlaySound3D(string soundGroupName, Transform trans, bool attachToSource, float volumePercentage, float? pitch, float delaySoundTime, string variationName);*

This is the same as "PlaySound", but you are passing in the Transform object as well so that the sound will trigger from its position. If you also pass in true for the last parameter, the Sound Group's variation will become a child of the "calling" GameObject so that the sound can "follow" the caller.

*MasterAudio.PlaySound3D(string soundGroupName, Vector3 sourcePosition, float volumePercentage = 1f, float? pitch = null, float delaySoundTime = 0f, string variationName = null)*

The third variant lets you specify a location to play the sound from. This is useful for 2D games where the Z of the object making the sound might not want to be used. You can alter it and use this.

All PlaySound methods return a PlaySoundResult object with the following properties:

1. SoundPlayed (boolean)
2. SoundScheduled (boolean) - false unless you scheduled a sound with the delaySoundTime field.
3. ActingVariation (SoundGroupVariation) - this will give you access to the actual variation used, if a sound was played.

**Note:** You can use ActingVariation.audio to access the properties of the Audio Source for the Variation used. You should *\*never\** set volume this way though, as it will not take into account all the other MasterAudio calculations for volume (Group / Bus / Variation / Mixer volume). If you do, it will appear that Master Audio is not working correctly.

You can also use the PlaySoundResult to be notified of when a sound is finished playing like this:

```
var result = MasterAudio.PlaySound("Scream");  
  
if (result.SoundPlayed) { // note: if you played the sound with a delay, use result.SoundScheduled  
  
    result.ActingVariation.SoundFinished += YourMethodToCall;  
  
}
```

Then simply add the following method to the same class to receive the Message Sent:

```
void YourMethodToCall() {  
  
    // do something, like play an animation!  
  
}
```

**Note:** You do not need to worry about unsubscribing to the Event as all subscribers are cleared out every time the Variation is played.

The PlaySoundResult can also be used to fade a clip out early.

```
var result = MasterAudio.PlaySound("Scream");
```

```
result.FadeOutNow(float fadeTime);
```

If you do not specify the fadeTime parameter, it will use the variation's fade out time value from the Inspector.

```
MasterAudio.StopAllOfSound(string soundGroupName);
```

This will stop all sounds of the given type instantly.

```
MasterAudio.FadeOutAllOfSound(string soundGroupName, float fadeTime);
```

This will fade out all variations of the specified sound.

-You can always check the **MasterAudio.SoundsReady** property through code (returns true or false) if you want to check if MasterAudio has finished initializing. This is only needed in rare startup cases during Awake on objects that are present during Scene load. All the other methods check this anyway to make sure it is true. Generally try not to trigger sounds during Scene Awake as MasterAudio initializes itself then.

## 5. Controlling the Audio - code methods

There are several methods you can call to modify the volume levels and mute/solo switches.

1) *MasterAudio.MasterVolumeLevel* - can be read or set. Value between 0 and 1.

2) *MasterAudio.GetGroupVolume(string soundType)* - returns a float.

3) *MasterAudio.SetGroupVolume(string soundType, float volume)*

4) *MasterAudio.MuteGroup(string soundType)*

5) *MasterAudio.UnmuteGroup(string soundType)*

6) *MasterAudio.SoloGroup(string soundType)* - also unmutes the group.

7) *MasterAudio.UnsoloGroup(string soundType)*

8) *MasterAudio.GrabGroup(string soundType)* - in case you want to read or manipulate other properties of the group such as limit mode or "neverInterrupt" settings, grab the object here.

9) *MasterAudio.SetBusVolumeByName(float volume, string busName)* - this can change a bus volume.

10) *MasterAudio.GrabBusByName(string busName)* - you can grab the Bus to read or change its properties.

11) *MasterAudio.GrabPlaylist(string playlistName)* - grabs a Playlist by name.

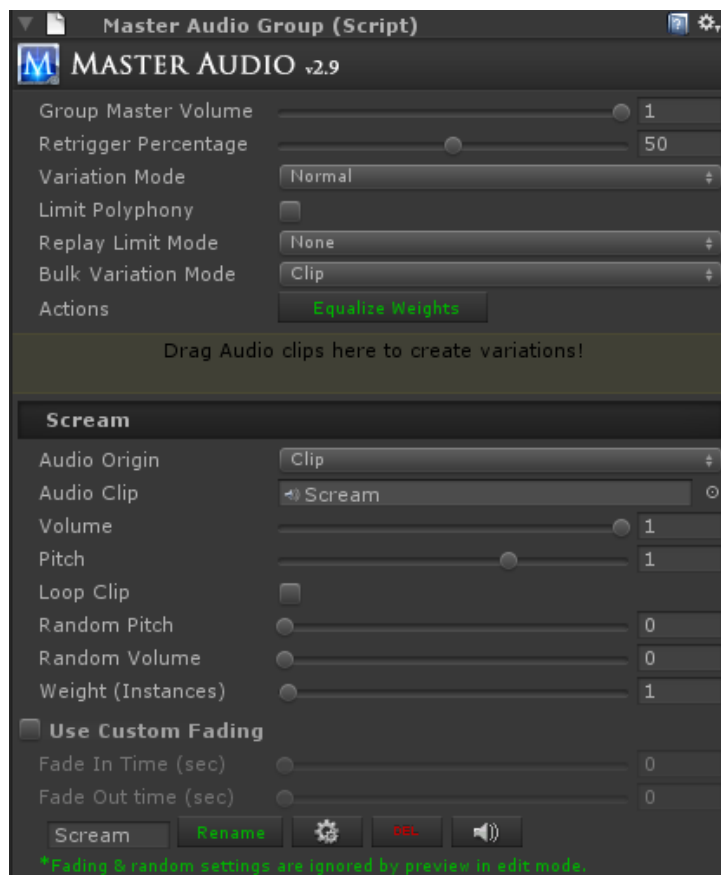
12) *PlaylistController.InstanceByName(string playlistControllerName)* - grabs a Playlist Controller by name.

There also are methods to pause or fade to a specified volume (over X seconds) a Bus, Group, Variation or the Playlist. Consult the Master Audio API document for details: <http://bit.ly/1bkiRei>. Playmaker Custom Actions exist for these as well.

## 6. Audio Groups: Fine-tuning through the Inspector panel

Additional settings are on each Sound Group (under then MasterAudio prefab). It allows you to quickly change the variation clips and volumes.

1) The Inspector for a Sound Group looks like this:



The controls are described below.

- a. Group Master Volume. This allows you to make all clips under this Sound Group quieter without having to go into each clip and adjust. Its default setting of 1 is full volume. This is the one that shows up on the group mixer.
- b. Retrigger Percentage - this setting will control the percentage of each clip in this Group that must be played before re-using the Variation is allowed. Anything lower than this percentage and the Variation is considered "busy" and cannot be used. 50% is the default.

An example: If you set this to 100%, clips can never be interrupted.

- c. Variation Mode - has two choices:
  - i. Normal - the default. Just think of it as "not looped chain". All variations can be played simultaneously.
  - ii. Looped Chain - with a looped chain, the Sound Group becomes something of a mini-Playlist. When you play a Variation, when it reaches the end of the clip, another Variation will be randomly played. It will continue to play random Variations until you stop the Sound Group. In this mode, only one Variation can be played at a time. Any attempt to play a Variation during the same time will stop the other chain that's already going so it can safely start a new chain.
- d. Limit Polyphony - this is an optional setting to limit the number of simultaneous variations that can be played in this Sound Group to a number smaller than the number of variations. If you check this setting, the next setting will appear.
- e. Polyphony Voice Limit - this is only visible and active if Limit Polyphony is checked. This limits the number of simultaneously playing variations in the Sound Group. i.e. you can set this to 3 even though you have 10 variations, and only 3 can play at the same time. This number cannot be set higher than the total weight of all variations in this Sound Group.
- f. Replay Limit Mode. This can be used to limit the amount of retriggers of this Sound Group, either by time or frames since the last trigger by MasterAudio. It has 3 modes.
  - i. 'None' is the default, which does nothing to limit retriggers.
  - ii. Frame Based will let you choose the number of frames to wait before retriggering is allowed.
  - iii. Time Based will let you set the amount of time to wait before retriggering is allowed.
- g. Bulk Variation Mode - this is the same as the one on the Master Audio prefab. Resource File or Clip are the choices.

- h. Equalize Weights button. This will set the Weight of all Variations in this Sound Group to one (equal weight). Weights control how often each variation will be triggered in relation to the other variations. More on this below.
- i. Create New Variations – this section is used to create a new variation. You drag 1 or more Audio clips into the colored rectangle to add variations to the Sound Group. This is optional and the variations will be played randomly from a pool.
- j. Variation Settings (Clip1 / Clip2 / etc). Here you can quickly fine tune your variations without going into each Variation prefab.

- i. Audio Origin - you choose either Clip (the default) or Resource File. If you choose Resource File, you will type or paste the name of the file in Resources in the Resource Filename field.

**Note:** if you use Resource files, you can save on memory usage. Audio Clips from Resource fields are loaded when told to Play, simultaneously into all Variations referencing that Clip. Whenever one finishes playing or stops for another reason, if zero are playing, the Clip is unloaded from memory.

- ii. Audio Clip – you can change the Audio Clip of the variation by dragging and dropping here. Only visible for Audio Origin of Clip.
    - iii. Resource filename - only visible for Audio Origin of Resource Filename. Do not put the file extension here. i.e. for King.mp3 enter "King". You can also drag the file from a Resource folder into the drag area above this. It will populate the folder and filename automatically.
    - iv. Volume / pitch / loop - properties of the Audio clip.
    - v. Random Pitch - Here you can specify the max pitch to randomly vary by each time the clip is played, based on the original clip pitch. It will fluctuate up OR down by a number no higher than the value you specify here.
    - vi. Random Volume - same as random pitch, for volume instead.
    - vii. Weight – you can make each variation trigger more or less often than the other variations by changing this value. For example, if you have 2 variations, and variation A has a weight of 4 and variation B has a weight of 1, then variation A will be triggered 4 times as often. This saves you from having to create more variations than you would need otherwise for duplicates. A weight of zero can be specified to not use the variation but not delete it either.



- viii. Custom fading section - if you enable this, whenever this Variation plays, it will use any fade in and fade out time you've specified. Fade out time is applied at the very end of the clip.
  - ix. Rename – type the new name in the text box and click rename. Remember to not have any duplicate variation names in a single Sound Group.
  - x. Delete button – deletes the variation.
  - xi. Go – takes you to the variation prefab in the Hierarchy, so you can tweak additional settings. The same clip settings can be found on the clip prefab, however if you wish to modify the Audio Source properties itself, this is a shortcut.
  - xii. Preview (speaker icon) - this will play the audio of the variation for preview purposes. In edit mode, this will ignore fading and random settings. In play mode, it will not.
- k. MasterAudio will automatically play the clips under each Sound Group in random order. It will make sure, taking the variation weights into account, that the “random pool” plays all weighted variations before refilling the random pool. This will mean that you get an even distribution of your weighted sounds over time regardless of application.

#### **A few words on weights vs. variations.**

MasterAudio will create additional variation children for each weight greater than one once you press Play.

- 1) if you need the ability to play up to 5 of the SAME sound in a polyphonic manner, use a single variation with a weight of 5.
- 2) If you need the ability to play different sounds in the same group, use more than one variation. Each variation can have its own weight, which will also have clones created at runtime for polyphonic purposes.

**Setting up a Group for a max number of voices:** Master Audio never uses AudioSource.PlayOneShot. This is because that method allows multiple overlapping samples to play on the same AudioSource. This would make the voice-limiting (polyphony) and the Retrigger Percentage code not work. However, the combination of weight + retrigger percentage gives you a "controllable" PlayOneShot in effect. By default a single variation in a Group will have a weight of 1. Therefore you can only play one of that sound at the same time. If you want to allow 5 simultaneously, up the weight of that variation to 5. Four additional clones of the variation will be created at runtime. If you have 3 different variations of the sound, but only want to allow 2 to play at the same time, keep the weight of all 3 at just 1, then set the

Polyphony Voice Limit to 2. Master Audio will play each variation randomly until all have been played, then "refill the pool" and start over the next time that Group is requested. When all variation are "busy" per the Retrigger Percentage, nothing will be heard for this trigger.

## 7. Master Audio: Advanced Options

Some additional settings on MasterAudio (top-level prefab) are:

- 1) Master Mixer Volume: this will control the volume of all sounds coming out of MasterAudio. The calculation is:

*clipVolume \* groupVolume \* busVolume \* masterAudioVolume.*

*Buses are explained shortly.*

You can also mute ALL Sound Groups with the mute button next to the slider!

- 2) Master Playlist Volume: this is the master volume for all Playlists. A Playlist clip volume is:

*clipVolume \* playlistVolume \* masterPlaylistVolume.*

- 3) Cross-fade time - this is the amount of time songs will cross-fade when you change to a new song.

- 4) Persist Across Scenes: Checking this will make it so the Master Audio prefab (and all Playlist Controller prefabs if you have them) not be destroyed when loading new scenes. If you are going to use this option, please use a "bootstrapper" scene that only ever occurs once at the beginning. Otherwise you could end up with more than one Master Audio prefab in a scene, which is not allowed.

**Note:** If you use this option, you will likely want to use DynamicSoundGroupCreators to create temporary Sound Groups & buses for each scene. That way, only the Sound Groups that are used in all scenes would go in the Master Audio prefab and memory usage is not wasted. There is a section devoted to this topic later.

- 5) Log All Sounds: This will output things to console about which random child has been played, whether there were none available to play, and a lot more. Turn this on for debugging only.
- 6) To configure the sounds that cause Music ducking, click on MasterAudio in the Hierarchy. There is a section labeled "Show Music Ducking ". Expand that. To add a sound, click the "Add Duck Group" icon, then choose the MasterAudio Sound Groups from the dropdown list that appear below. That's it!

- a. There's also a setting for "Begin Unduck" for each Duck Group. This controls when the music volume ramping back up starts. It defaults to 50. That means that after 50% of the clip that caused the ducking has been played, then volume will start ramping back up over the remaining duration of the clip.
  - b. Duck Vol Multiplier - this controls the ratio of volume during the beginning of a duck. If you set this to .5 for example, the music will duck to half volume initially. The range is from 0 to 1.
- 7) When testing, pay attention to the music getting quieter during those sounds. Note, it may be hard to notice the effect on very quick sounds. Try it on longer sounds. To remove the last sound in this section, click the minus icon.
- 8) Pro Audio Mixer controls: this section in the MasterAudio Inspector is a mixer for your Sound Groups. It looks about like this:



- a. The slider is to control the master volume of that Sound Group.
- b. There's a colored LED strip that lights up for awhile and animates each time a Variation on that Sound Group starts playing. This is just for your information to see what's playing without having to go into Debug log mode. It does not give any indication of real volume.
- c. "S" is a solo switch.
- d. "M" is a mute switch.
- e. The gears icon (settings) will select the Sound Group in the hierarchy so you can tweak variations and additional controls.
- f. "Del" will delete the Sound Group and all variations in it.
- g. When "playing" in the Unity editor, the Delete button is replaced by a speaker icon. You can click that to audition the Sound Group whenever you like.

- h. Quick buttons for deselecting all mute / solo switches and another for settings all group volumes to 1 (the maximum).
- i. For those of you who have not used a mixing board before, here's an explanation of solo and mute switches.
  - i. If you have zero soloed groups, all groups will produce sound except the ones that are muted.
  - ii. If at least one group is soloed, you will only hear the soloed groups – all non-soloed groups will not be heard.
  - iii. Selecting solo will deselect mute, and vice versa.
- j. Buses! The blue dropdown is for assigning Sound Groups to buses. This allows you to control the volume of several Sound Groups at once. In essence, it's a sound router.
  - i. By default there are no buses. The text "[NO BUS]" means the Sound Group does not go to a bus.
  - ii. To create a bus, select "[NEW BUS]" from the dropdown. A new Bus Control section will show up under the Group Control section. It's still part of the Group mixer section of MasterAudio. You can type into the text field that says "[BUS NAME]" to change the name of the bus.
  - iii. Each bus has a volume, solo and mute switches, and a delete button. These work as expected, except that the solo and mute switches actually solo or mute all Sound Groups assigned to the bus (to make things less confusing).

**Note:** if you mute or solo a bus, all Groups with that bus cannot have their mute and solo buttons pressed. This preserves the "bus mute" and "bus solo" status.

  - iv. When you have at least one bus created, there will be a checkbox at the top of the Sound Group section to "group by bus". This is very helpful when you have a lot of sounds! It defaults to "on".
  - v. Note - all mixer and bus controls now work in real time during Editor play!

## 8. Audio FX Filters

All Unity FX Filters are available for use in Master Audio, and at the time of writing this is a Master Audio exclusive. They are included in each Variation Inspector, turned off by default. They can also be added individually from the Component -> Audio menu. The filters are:

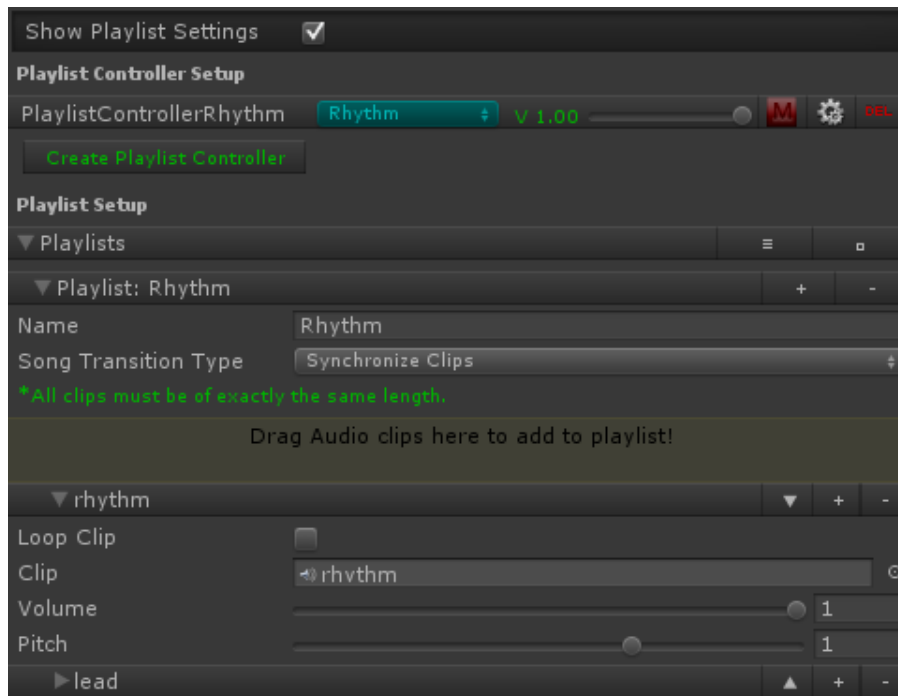
- 1) Audio Low-pass filter
- 2) Audio High-pass filter
- 3) Audio Reverb filter
- 4) Audio Chorus filter
- 5) Audio Distortion filter
- 6) Audio Echo filter

**Note:** Filters are only available on Unity Pro. There are convenience lazy load properties in the SoundGroupVariation class you can use to grab each filter for manipulations.

## 9. Playlist Controller and Music Playlists

**Music Playlists** - the last section in the MasterAudio prefab's Inspector. Here you can set up multiple Playlists of music tracks that MasterAudio uses for your soundtrack.

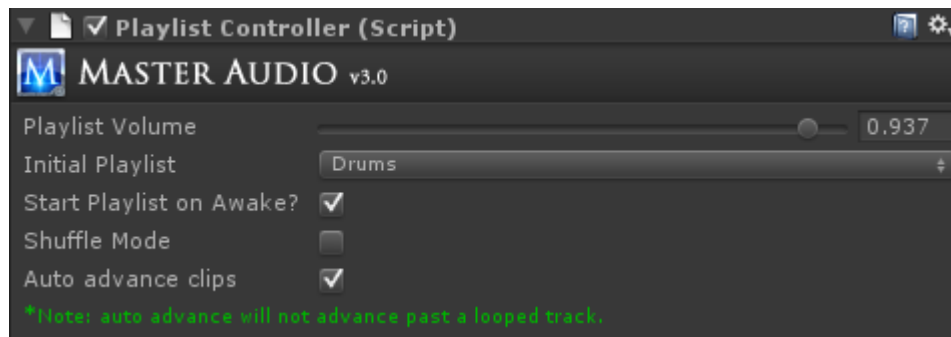
**Note:** In order to use the Playlist and/or ducking features, you will need to put a PlaylistController prefab in your scene. There is a button here to do that named "Create PlaylistController". Or use the Audio Manager window to add it.



The settings are as follows:

### Playlist Controller Setup

This section lists your Playlist Controllers and lets you create more, assign the initial Playlist (blue dropdown), set the volume for the Controller, or mute it. If you click the settings icon (gear), you will see the Playlist Controller settings, shown below. Each Playlist Controller has its own settings like this.



- Playlist Volume - think of this as the volume of the Playlist Controller itself. This is a way to balance volumes between multiple Playlist Controllers' Playlists.
- Start Playlist on Awake - pretty self-explanatory. This will play the first clip in the selected Playlist as soon as the scene begins. If you have Shuffle mode turned on as well, it will play a random clip instead.

- c. Shuffle mode - if you turn this on, tracks in the current Playlist will be played in random order. All will be played before the random pool refills. If this is not turned on, the tracks will be played in order top to bottom.
- d. Auto advance clips - if you turn this on, as soon as a track ends, another will start playing. It will be the next track if you are not using shuffle, or a random track if you are.

### Playlist Controller Setup

- e. The Music Playlists - you can add any number of clips here. They will play from top to bottom if you have not enabled shuffle mode. If you have more than one clip, up and down arrows will appear to change sequence of the clips. The settings for each Playlist are:
  - a. Playlist name (for reference by name if you want to play a specific Playlist).
  - b. Song Transition Type. Choices are as follows:
    - i. New Song From Beginning (the default). When playing the next or random song, the new song will start from the beginning.
    - ii. Synchronize Clips - with this setting, playing the next or random song will start the new clip at the same position (time) the previous clip was at. Very cool for cross-fading between alternate version of the same track.

**Note:** if you use Synchronize Clips, auto-advance is disabled in any Playlist Controller using this Playlist and all clips in the Playlist will loop.
- f. To create additional Playlists, click the plus icon in Playlist one.
- g. The minus icon on a Playlist row deletes the Playlist.
- h. You can move the Playlist order (if you have more than one Playlist) by clicking the up and down arrow icons.
- i. You can add songs to the Playlist by dragging one or more clips into the colored rectangle.
- j. Code options - to control the Playlist from code, you have the following options to call:

*MasterAudio.TriggerPlaylistClip(string clipName);*

*MasterAudio.ChangePlaylistByName(string playlistName, bool playFirstClip);*

*MasterAudio.ChangePlaylistByIndex(int playlistIndex, bool playFirstClip);*

*MasterAudio.StopPlaylist(); // stops playing the current song and fades out to silence.*

**Note:** The above methods work if you only have one Playlist Controller. If you have more than one, there are overloaded methods that take the Playlist Controller Name as a parameter as well. Some have an "all Playlist Controllers" method as well, so you can "pause all" etc.

### Events to subscribe to

1. You can subscribe to the SongChanged event in the PlaylistController class to be notified when the song changes. That code looks like this:

```
var controller = PlaylistController.InstanceByName("PlaylistControllerBass");  
  
controller.SongChanged += SongChanged; // the name of your listener method  
  
private void SongChanged(string newSongName) {  
  
    Debug.Log("Song changed to: " + newSongName);  
  
}
```

2. You can also execute a method when a gradual PlaylistFade you asked for is completed as well. That code is:

```
PlaylistController.InstanceByName("PlaylistControllerDrums").FadeToVolume(.5f, 2f, delegate {  
  
    Debug.Log("done");  
  
});
```

**New Jukebox!** In Master Audio 3.0, we have a new Jukebox section that shows up when you press play in the editor. It is shown below - there is one shows for each Playlist Controller. It shows you the current Playlist, active song and fading song, plus time remaining on both. Also, there are controls for changing Playlist, stopping / pausing the song, going to next song and going to random song. You can change to a different clip in the Playlist by picking it from the dropdown. You also can adjust the Playlist volume and

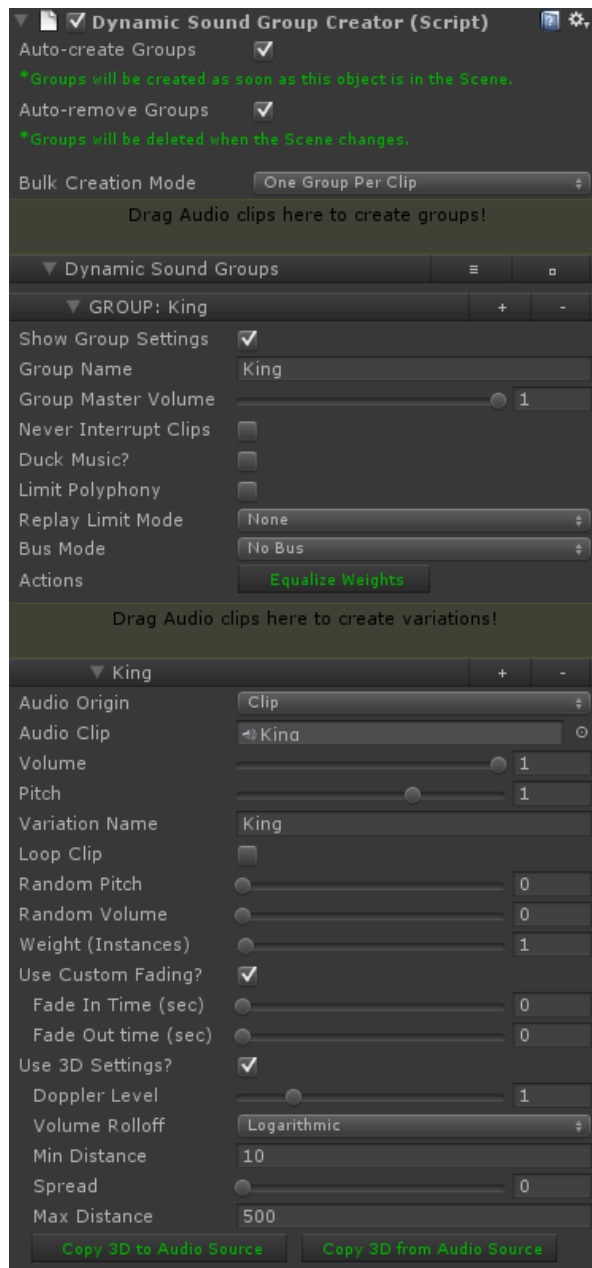


jump to a specific part of the song with the slider in the bottom row. Lastly, you can mute the playlist or jump to a specific part of the song by moving the lower slider.



## 10. Dynamic Creation and Modification of Sound Groups

There is a prefab called `DynamicSoundGroupCreator`, which can be created from the Audio Manager window. You can attach this to dynamic content that you want to create Sound Groups based on its settings. These Sound Group clips can come from Resource files or other locations. It looks like this:



The settings are explained here.

- 1) Auto-create Groups - if you check this, the Sound Groups specified in the lower section will be created in the OnStart method of the script. In other words, as soon as this prefab is Instantiated, it will create the groups. If you do not check this box, you will need to call the CreateGroups method yourself from some script.
- 2) Auto-remove Groups - If you check this box, the Sound Groups you create are temporary and will be removed when this object is destroyed (normally this will be when the Scene changes).

- 3) Bulk Creation Mode: Same as this setting in the Master Audio prefab. Controls whether multiple clips dragged in at the same time create 1 or multiple Sound Groups.
- 4) Variation Mode - this is the same as Bulk Variation Mode in Master Audio, only it applies to non-bulk items added here as well as bulk items.
- 5) Dynamic Sound Groups section. Here you specify settings for any number of Sound Groups to create. You also have full variation support. This section looks and functions exactly the same as the Sound Group Variation Inspector, with a few extras:
  - a) Show Group Settings - you can uncheck this to collapse all the Group settings temporarily. Variations are still collapsible individually.
  - b) Duck Music - if you check this checkbox, when the Sound Group is created it will also be added to the list of Groups that cause music ducking.
  - c) Bus Mode - No Bus / Use Existing / Create New. This setting can be useful to assign the new Sound Group to a particular Bus, including creating a new Bus for it!
  - d) Alpha Sort Variations button - this will sort the variations by "Variation Name" in alphabetical order.
- 6) Variations each have a "Use 3D Settings" section you can toggle on to set five key 3D Audio settings that will be used in the Variation that's created at runtime.
  - a) For visualization purposes, you can click "Copy 3D to Audio Source" to copy the variation settings into the single Audio Source located in the Dynamic Sound Group Creator prefab. Then you can scroll up and use the default Audio Source widgets to visually tweak your 3d settings. When finished, go back down and click "Copy 3D from Audio Source".

There are methods to create new Sound Groups on the fly.

1. *MasterAudio.CreateNewSoundGroup (string soundGroupName, AudioClip clip, int variationCount = 1, float pitch = 1f, float volume = 1f, float randomPitch = 0f, float randomVolume = 1f);*
2. *MasterAudio.CreateNewSoundGroupFromResources (string soundGroupName, string resourceFileName, int variationCount = 1, float pitch = 1f, float volume = 1f, float randomPitch = 0f, float randomVolume = 1f);*

All the parameters with values assigned are optional to pass.

## 11. Playmaker integration

I have included the optional "PlaymakerCustomActions" package so that you don't have to write any code to integrate with Playmaker. There are 37 custom actions included, under the Audio category. These should cover every method you would call manually. This is a list of the custom actions.

1. Master Audio Bus Fade
2. Master Audio Bus Mute
3. Master Audio Bus Pause
4. Master Audio Bus Set Volume
5. Master Audio Bus Solo
6. Master Audio Bus Unmute
7. Master Audio Bus Unpause
8. Master Audio Bus Unsolo
9. Master Audio Ducking Add Group
10. Master Audio Ducking Remove Group
11. Master Audio Ducking Toggle
12. Master Audio Fade Out All Of Sound
13. Master Audio Group Fade
14. Master Audio Group Mute
15. Master Audio Group Pause
16. Master Audio Group Set Volume
17. Master Audio Group Solo
18. Master Audio Group Toggle Mute
19. Master Audio Group Toggle Solo
20. Master Audio Group Unmute
21. Master Audio Group Unpause
22. Master Audio Group Unsolo

23. Master Audio Playlist Clip By Name
24. Master Audio Playlist Clip Next
25. Master Audio Playlist Clip Random
26. Master Audio Playlist Fade
27. Master Audio Playlist Pause
28. Master Audio Playlist Set Volume
29. Master Audio Playlist Start By Name
30. Master Audio Playlist Stop
31. Master Audio Playlist Unpause
32. Master Audio Play Sound
33. Master Audio Set Master Volume
34. Master Audio Stop All Of Sound
35. Master Audio Variation Change Pitch

## 12. 2D Toolkit Integration

To make 2D Toolkit use Master Audio instead of its own tk2dUIAudioManager, simply install the optional package "MA\_Tk2d". It will overwrite these 3 files:

- `_TK2DROOT/tk2dUI/Code/Controls/tk2dUISoundItem.cs`
- `_TK2DROOT/tk2dUI/Code/Core/tk2dUIAudioManager.cs`
- `_TK2DROOT/tk2dUI/Editor/Controls/tk2dUISoundItemEditor.cs`

This means that for a UISoundItem, the Inspector will allow you to select or type a Master Audio Sound Group from a dropdown. Note that if you have moved 2D Toolkit folders after importing it, you will need to move the Master Audio replacement files to the new location. Every time you upgrade 2D Toolkit, you will need to open the Master Audio 2D Toolkit package again to overwrite the new changes.

### 13. Final Words

Support is available by emailing [support@darktonic.com](mailto:support@darktonic.com) or in the Unity forum thread [here](#). Again, the Master Audio API documentation can be found here: <http://bit.ly/1bkiRei>

Make sure to check out our other plugins such as Killer Waves at <http://www.darktonic.com/p/developer.html>. Thank you!

-All at Dark Tonic