MASTER AUDIO PLUGIN V3.4 - By Dark Tonic, Inc. (c) 2013-2014

Table of Contents

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

Demo videos at: http://bit.ly/17MNI2f (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.

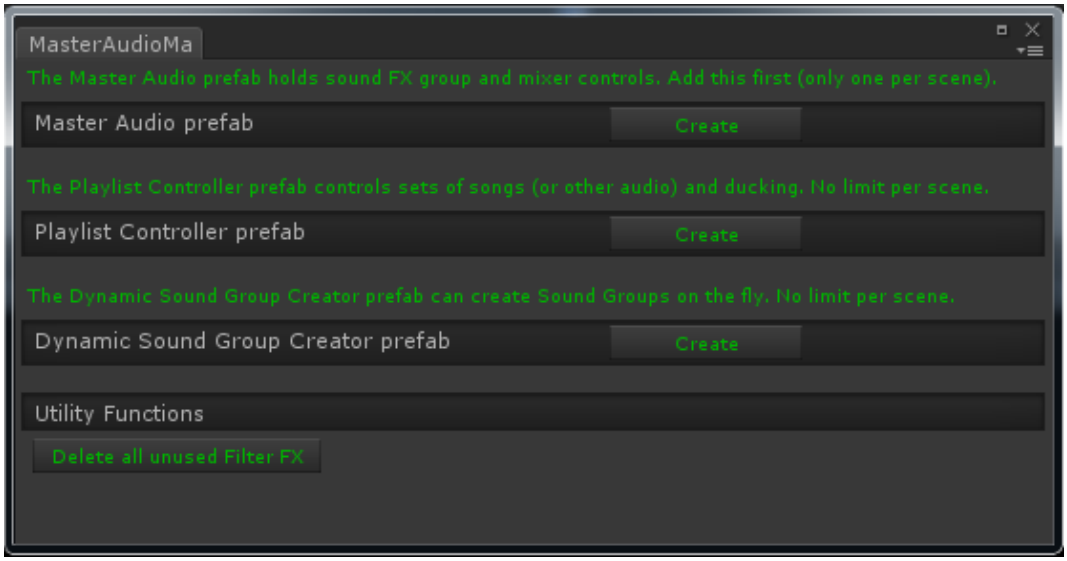
Most of the code options are listed in various parts of this document. The **entire** Master Audio API documentation can be found here: http://bit.ly/1bkiRei

1. Solutions! Master Audio solves the following problems (and tons 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, including custom events you can define. No coding is needed to do this in most cases.
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

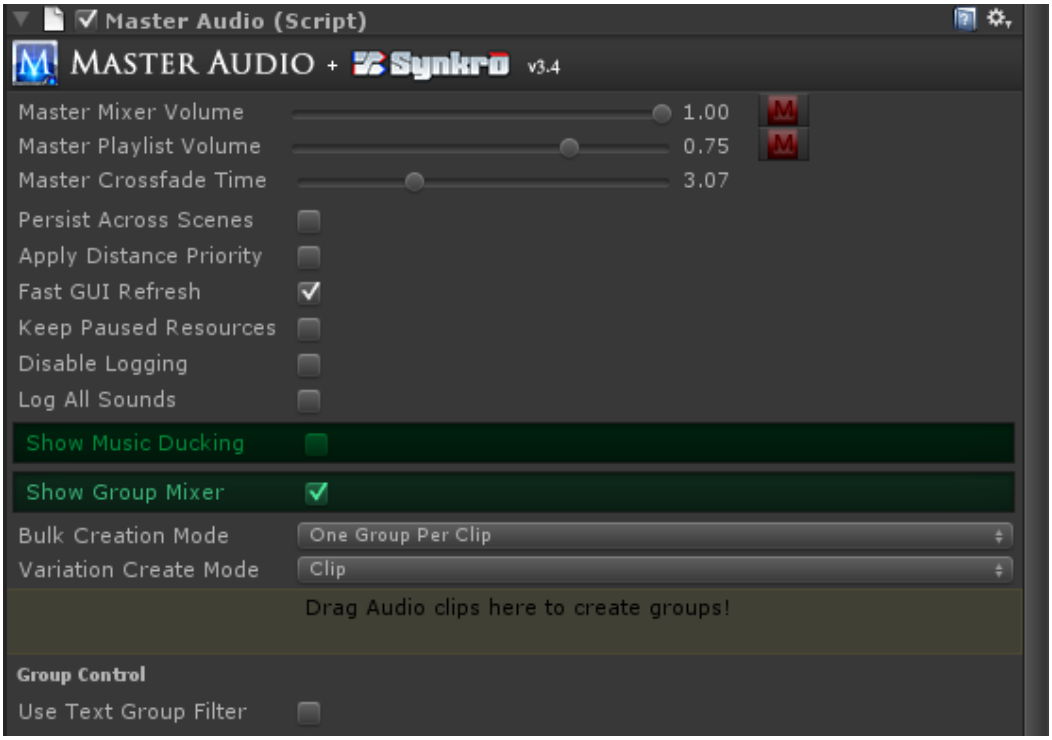
1) From the Unity "Window" menu, select "Master Audio Manager". This is a small window that will help explain and set up what you need for Master Audio. It looks like this:



2) First, create the Master Audio prefab with the "Create" button. The position of the MasterAudio prefab will be used for any sounds that you don't provide a position to (usually only used with 2D sounds).

Note: Do not reparent the Master Audio prefab. Leave it as top-level. Also do not move any of the sub-prefabs that get created under it anywhere else. They need to stay where they are for Master Audio to work properly.

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. Variation Create 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 coding" 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. See NGUI Integration section. Add this to a prefab from the menu here: Component -> Dark Tonic -> Master Audio -> ButtonClicker.

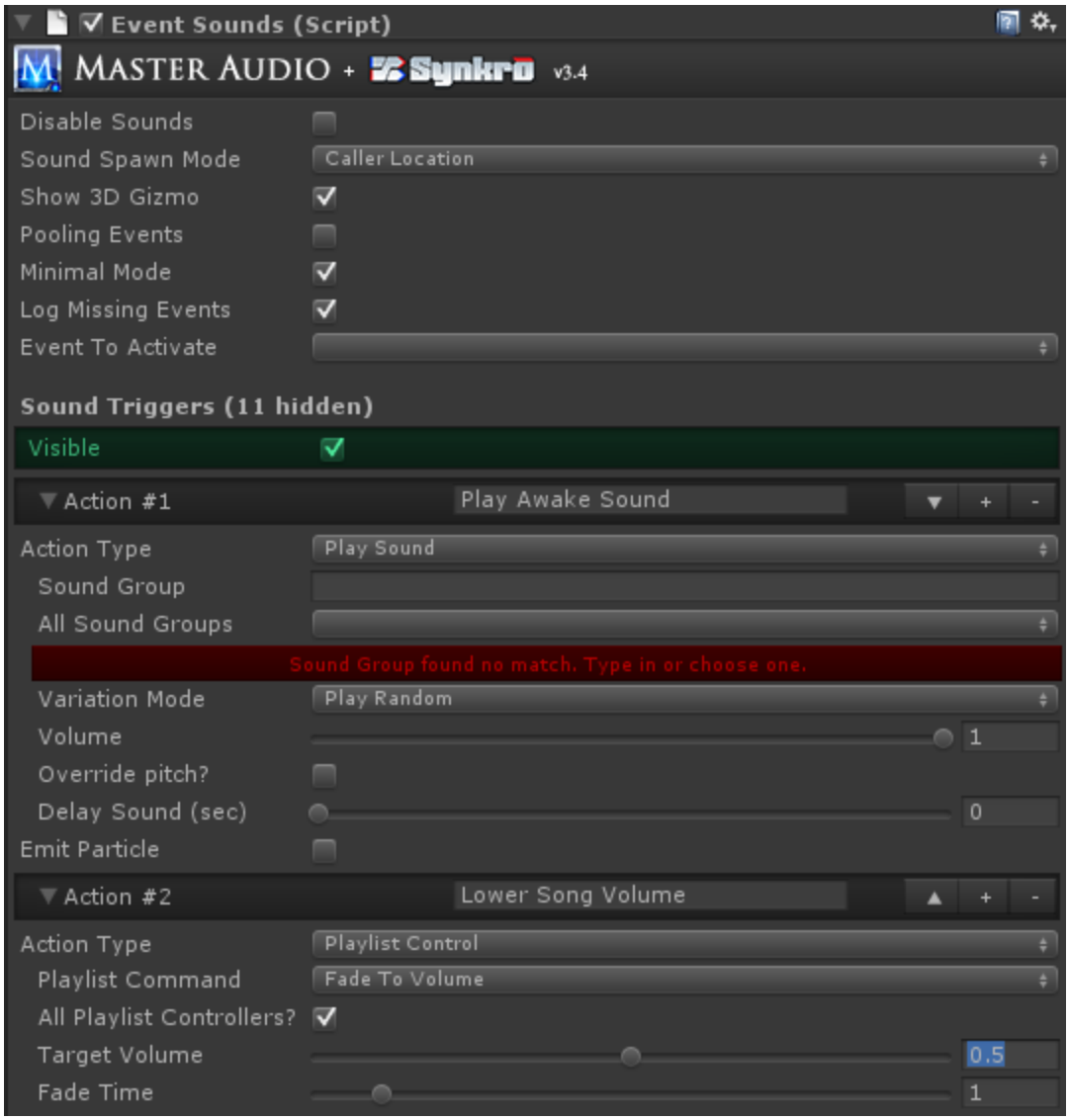
2. **EventSounds.cs** is a powerful and flexible script that you can attach to prefabs to trigger MasterAudio sounds (and manipulate Playlists, Sound Groups and Buses) for certain MonoBehavior, PoolManager / PoolBoss (KillerWaves) and other events. So even without Playmaker, you have a wide variety of "no code" actions you can do with Master Audio. Add this script to a prefab from the menu here: Component -> Dark Tonic -> Master Audio -> EventSounds. 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. OnParticleCollision
    m. OnMouseEnter
    n. OnMouseDown (Mouse Click is what it's called).
    o. For PoolManager and Killer Waves users, we also have OnSpawned and OnDespawned. To get these to show up, you must check the "Pooling Events" checkbox.
    p. Custom Event - you can add any number of Custom Event Receivers to Event Sounds. For each one, you specify which event you're receiving.

    Note: OnBecameVisible (and OnBecameVisible) 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 Pooling events). They don't provide exactly the same functionality but it will work for most purposes.

    Minimal Mode - Defaults to on. In minimal mode, add events from the "Event to Activate" dropdown. Note that any Custom Events you have added will show up even when deactivated. If you turn off minimal mode, all event sections will be shown whether you're using them or not (unused ones collapsed). When not in minimal mode, you check the checkbox for the event to activate it for usage.

    Event Sounds looks something like this:



You can add a section for each event type in the Inspector. Each section can have any number of actions (starts with 1). You can add more actions using the plus icon. Each action has a name you can edit -this is for your use only. All actions are executed from top to bottom when the event occurs. You can move the actions up and down using the arrow icons. The minus icon deletes an action. Each action has the following settings:

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. Stop
    f. Unmute
    g. Unpause
    h. 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. Playlist Controller Name is optional if you have only one of them.
    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
v. Custom Event Control - choosing this will reveal a menu of Custom Event Commands.
    a. Fire Event - this will fire the event you specify, and all Custom Event Receivers in the Scene that are configured to receive the event will respond.

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.
iii. Log Missing Events: defaults to on. If you create Custom Events at runtime (not configured in Master Audio prefab, turn this off to avoid false warnings in a "Type In" event in this script.

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).

There are many new methods for PlaySound and PlaySound3D in V3.3.4, listed below. The old PlaySound3D (not listed here) is considered deprecated and may be removed in the future. Please use

one of the more specific methods instead. In fact, you cannot compile if you are using PlaySound3D. Consult the API Website for full details (link above on page 2).

1. *PlaySound*
2. *PlaySoundAndForget*
3. *PlaySound3DAtVector3*
4. *PlaySound3DAtVector3AndForget*
5. *PlaySound3DAtTransform*
6. *PlaySound3DAtTransformAndForget*
7. *PlaySound3DFollowTransform*
8. *PlaySound3DFollowTransformAndForget*
9. *PlaySound3DAndForget (used only by Playmaker Custom Action).*

These are the same as "PlaySound", but you are passing in the Transform object as well (or Vector3) so that the sound will trigger from its position. If you use the FollowTransform methods, the Sound Group's variation will "follow" the caller.

Using the "AtVector3" methods is useful for 2D games where the Z of the object making the sound might not want to be used. You can alter is and use this.

The "AndForget" methods do not return a PlaySoundResult, and the rest do. If you don't need it, it is better for performance to not generate it. A PlaySoundResult object has 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 (make sure to check if it's null first!:

```
var result = MasterAudio.PlaySound("Scream");
if (result != null && 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.

Note: for your own code classes you may write, if you're on Unity 4, you can use the Custom Property Drawer "SoundGroupAttribute", so you can decorate public strings that are Sound Groups in your Inspectors like this:

```
[SoundGroupAttribute] public string laserSound;
```

This will show you the familiar Sound Group dropdowns used on all the Master Audio Inspectors!

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.

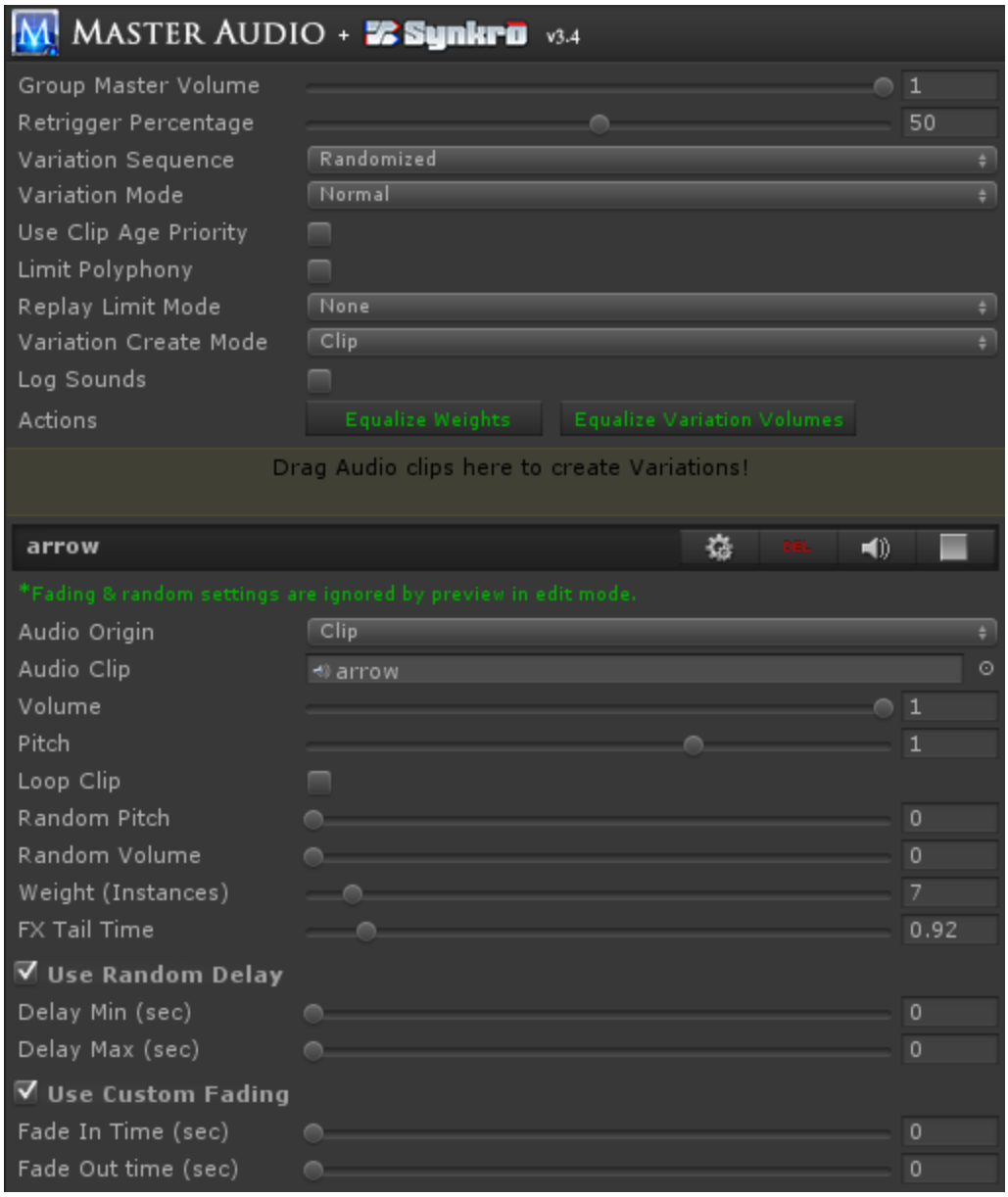
12)PlaylistController.InstanceByName(string playlistControllerName) - grabs a Playlist Controller by name.

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. Variation Sequence - two choices.

   i. Random (the default) - Variations are played randomly from a pool and refilled after all have been played.

   ii. Top To Bottom - if you choose this, the Variations are played in alphabetical order and refilled after all have been played. If you have chosen this option, more fields appear.

      1. Refill Variation Pool After Inactive Time - if you check this box, the next field will be used to refill the pool after X seconds of not playing a sound from

this Group. Note that the time is measure from the time a sound starts to play, not when it ends.

2. Inactive Time (sec) - this is how long the pool will wait without any sound being played before it automatically refills the pool (and starts at the top Variation again next time it's played).

c. Variation Mode - has three choices:
    i. Normal - the default. All variations can be played simultaneously. There are additional voice and time limiting controls in this mode, described below.
    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 another Variation during the same time will stop the other chain that's already going so it can safely start a new chain. All Variation clips will automatically turn off their loop setting in this mode. This is useful for random ambience sound sequences.
    iii. Dialog - this setting will also stop all other Variations in its Group every time you play one, but with no other restrictions or side effects. Useful for a single character's dialog, to make sure he never says more than one thing at a time without you having to call StopAllOfSound every time you play something from the Group.
d. If your Variation Mode is Looped Chain, you will see these fields.
    i. Loop Mode - Endless (default) or Number of Loops.
    ii. Number of Loops - If you choose Number of Loops for your Loop Mode, you will see this field. Specify the number of times the entire group shall be played. The Group will stop playing after this number of loops.
    iii. Clip Change Delay Min / Clip Change Delay Max - if you change these to non-zero, they will specify a range of random pausing before each subsequent chained clip is played.
e. If your Variation Mode is Normal, you will see these fields.
    i. 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.
    ii. 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.
    iii. 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.

iv. 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.
    1. 'None' is the default, which does nothing to limit retriggers.
    2. Frame Based will let you choose the number of frames to wait before retriggering is allowed.
v. Time Based will let you set the amount of time to wait before retriggering is allowed.

f. If your Variation Mode is Dialog, you will see these fields.
    i. Dialog Custom Fade? Checking box will show and use the next field.
    ii. Custom Fade Out Time - the amount of seconds to fade out over when another Variation is played.

g. Use Clip Age Priority - This is the same as the setting of the same name under Master Audio, but this is for this Sound Group alone. Turn this on if you wish to periodically update the priority of a 3D sound effect based on its "oldness" as well.
h. Variation Create Mode - this is the same as the one on the Master Audio prefab. Resource File or Clip are the choices.
i. 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.
j. Equalize Variation Volumes - clicking this button will mathematically even the volume of all Variations in the Sound Group based on their average volume. It will move the volume sliders of the Variations to do this. No alternation of the sound clips is done.
   Note: This function does not work on streaming, mp3 or compressed (OGGVORBIS) files. You will actually show an error in the console for these types and those files will be omitted from the volume leveling. The error is not trappable unfortunately.

k. 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.
l. 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. FX Tail Time - this only shows up if you have one of more Unity Filter FX components active (checked) on the Variation. If you do, you can specify a longer FX Tail Time here so that the FX tails (such as a reverb tail) don't get cut off when the sound is done playing.
ix. Random delay section - if you enable this, whenever this Variation plays, it will delay for an additional X seconds, X is random between the min and max you specify. This is in addition to any delay you specified in the MasterAudio.PlaySound method parameters, EventSounds or ChainedLoop random delay.
x. 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.
xi. 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.
xii. Delete button – deletes the variation.
xiii. 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.
xiv. 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.

m. 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. When the first random variation is busy, the rest will be tried from least recently played to most recently played. This way usually the 2nd choice will succeed.

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) Master Crossfade Time - this is the amount of time songs will cross-fade when you change to a new song. You can override this per Playlist if you want to.
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. The Example Scenes "BootstrapperScene" and "GameScene" show this working with per-Scene sounds. Play the Bootstrapper Scene in the editor for instructions on setup.

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) Apply Distance Priority - defaults to off. If you check this, Master Audio will automatically calculate a priority to assign to each 3D AudioSource you play based on its distance from the AudioListener in the Scene. Further away objects will get a lower priority. All sounds played in a 2D manner will get a high priority, and all music in PlaylistControllers will get a high priority. The reason for this setting is that when there are more than 32 AudioSources playing, Unity likes to mute AudioSources that aren't in the highest 32 priorities. This gives you some control over which sounds will be muted.
    a. Reprioritize Time Gap - this defaults to 0.1 seconds and can range from 0.1 to 1.0 seconds. This value controls how often Master Audio will re-evaluate the distance of playing Variations and assign a new priority. This is only used for Variations that are following an object. Increasing this value will result in better performance, so play with it.
    b. Use Clip Age Priority - if you turn this on, even non-following sounds will have their priority recalculated every X seconds (using the setting above: Reprioritize Time Gap), taking into account the amount of time since the sound was started. Newer clips will get a higher priority. Old clips will be more likely to be muted. Turning this on will cost extra performance, so use with care on mobile and make sure you aren't using a faster Time Gap than you need! This is the global setting, and will turn this on for all Sound Groups. There's also a per-Sound Group setting found in the Sound Group's Inspector, with the same name.
6) Fast GUI Refresh. This is checked by default and provides the fastest (constant) UI refresh. You can uncheck it to use less CPU. This only affects CPU when you have the Master Audio prefab selected.
7) Keep Paused Resources: these defaults to off. If you check this, you will be able to resume paused sound clips that are loaded from Resource folders. However, be advised that until you call Stop on Resource sounds that you pause, memory will not be released.
8) Disable Logging - turn this on when you do a release and you're satisfied that you don't need to read logs any more. It will override Log All Sounds and any per-Group log settings.
9) 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. Messages are "info" level, so they're white.
10) 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. Global settings:

i. Default Begin Unduck - this will set the default for the previous Begin Unduck control for all new Groups added.
ii. Ducked 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.

11) 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.
12) 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. The yellow bracketed number at the left of each Group & Bus row is the number of active voices playing. Good to visualize how close you are coming to any limits you may have set up, or the Unity max (32). The number is red if you have reached a Group's polyphony limit or a Bus' Voice limit.
d. “S” is a solo switch.
e. “M” is a mute switch.

f. The gears icon (settings) will select the Sound Group in the hierarchy so you can tweak variations and additional controls.
g. “Del“ will delete the Sound Group and all variations in it.
h. 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.
i. Quick buttons for deselecting all mute / solo switches and another for settings all group volumes to 1 (the maximum).
j. 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.
k. 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. Bus voice limit - this defaults to unlimited, but you can pick between 1-32 voices to limit the bus to. For example, if you have 10 character Sound Groups assigned to a "dialogue" bus, you could limit the bus voices to 5 so that in total only 5 Variations among those 10 Groups could be played at the same time. This helps you avoid hardware-based voice limits for mobile devices, etc. Some devices can only play 20-some odd voices for example.
    iv. 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.

    v. When you have a lot of Sound Groups, it can be time-consuming to even locate a Group, so use the bus filter (below), or turn on "Text Group Filter", which allows you to type a few characters and filter out all non-matches, wildcard style (not case sensitive). For instance if you type "AR" you would match arrow, car, and warship.
    vi. When you have at least one bus created, there will be additional controls above the mixer.

1. 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".
2. A Bus filter dropdown. You can choose which Sound Groups appear in the mixer by selecting the bus they belong to. All buses is the first choice and the default.

vii. 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 can be added in each Variation Inspector with the dropdown labeled "Add Filter Effect". 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

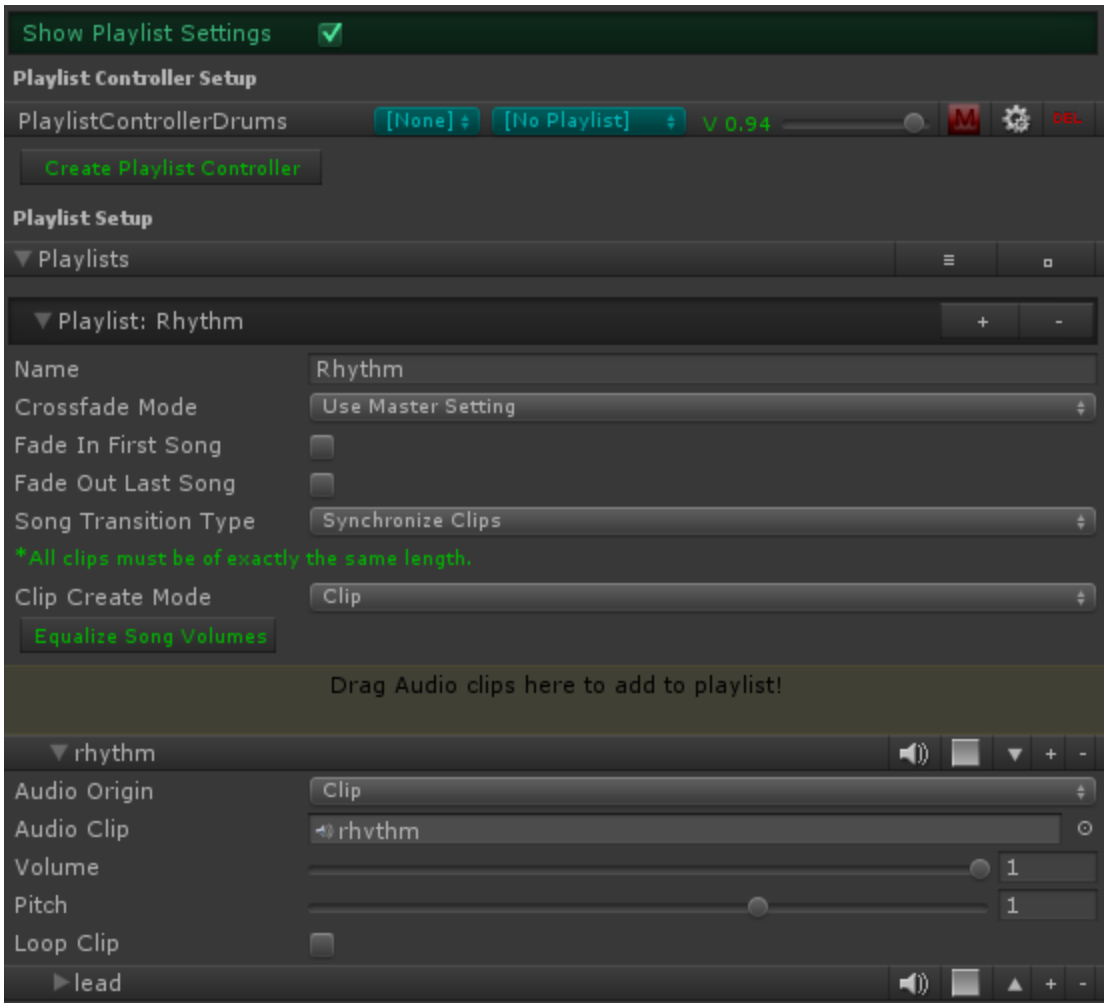
Note: Filters are only available on Unity Pro. There are convenience lazy load properties in the SoundGroupVariation class you can use to grab each filter for manipulations.

Note: Even unused and disabled filters take up audio memory, so only add them if you'll use them. There's a button to "Delete all unused Filter FX" in the Master Audio Manager window.

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.

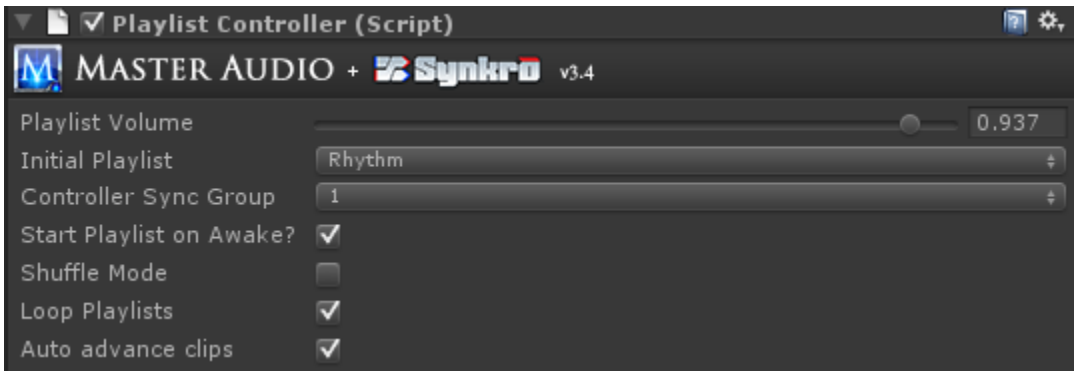
Note: In order to use the Playlist and/or ducking features, you will need to put a PlaylistController prefab in your scene. There is a button here to do that named "Create PlaylistController". Or use the Audio Manager window to add it.



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.



a. 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.
b. Controller Sync Group - this is off by default. It's optional and is a number 1-4. If you assign a number, then whenever a new song is played, it's time will be set to the first matching Playlist Controller's (with the same Sync Group and a clip playing) time marker. This will also help align layered Resource clips more precisely.
c. Initial Playlist - here you select a Playlist to start with.
d. 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.
e. 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.
f. Loop Playlists - this defaults to on. This means that when there are no tracks left to play (with either shuffle mode or not), the song pool will refill and repeat. If you turn this off, the last song will fade out using your crossfade timing.
g. 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

a. Equalize Song Volumes - clicking this button will mathematically even the volume of all songs in a Playlist based on their average volume. It will move the volume sliders of the songs to do this. No alternation of the sound clips is done.
b. 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. Crossfade Mode - by default it uses the Master Setting (Master Crossfade Time). You can also choose Override and specify this Playlist's crossfade time with the next field.
        i. Crossfade time - the crossfade time to use when hosting this Playlist.
    c. Fade In First Song - check this box to fade in the first song (meaning when you play a song when no song is already playing).
    d. Fade Out Last Song - check this box to fade out the last song (meaning the Playlist is not looped and no song will play after this).
    e. Song Transition Type. Choices are as follows:
        i. New Clip From Beginning (the default). When playing the next or random song, the new song will start from the beginning.
        ii. New Clip From Last Known Position. This will let each song always resume from the last position it was at before cross-fading to another song. If no previous play of the song, it will start from the beginning.

iii. 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.

c. To create additional Playlists, click the plus icon in Playlist one.
d. The minus icon on a Playlist row deletes the Playlist.
e. You can move the Playlist order (if you have more than one Playlist) by clicking the up and down arrow icons.
f. You can add songs to the Playlist by dragging one or more clips into the colored rectangle.
g. 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.
MasterAudio.QueuePlaylistClip(string clipName); // will play a song after the current song.Requires auto-advance to be on and it turns looping off for the current song.
```

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);
}
```

   There's now also a SongEnded event that you can hook up to with the same type of code. You can use the DelayBetweenSongs script if you want to have a fixed or random pause between songs. You do have to turn off auto-advance for the Playlist Controller you're using this on. Just fill out the 3 properties in the Inspector and it works!

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");
});
```

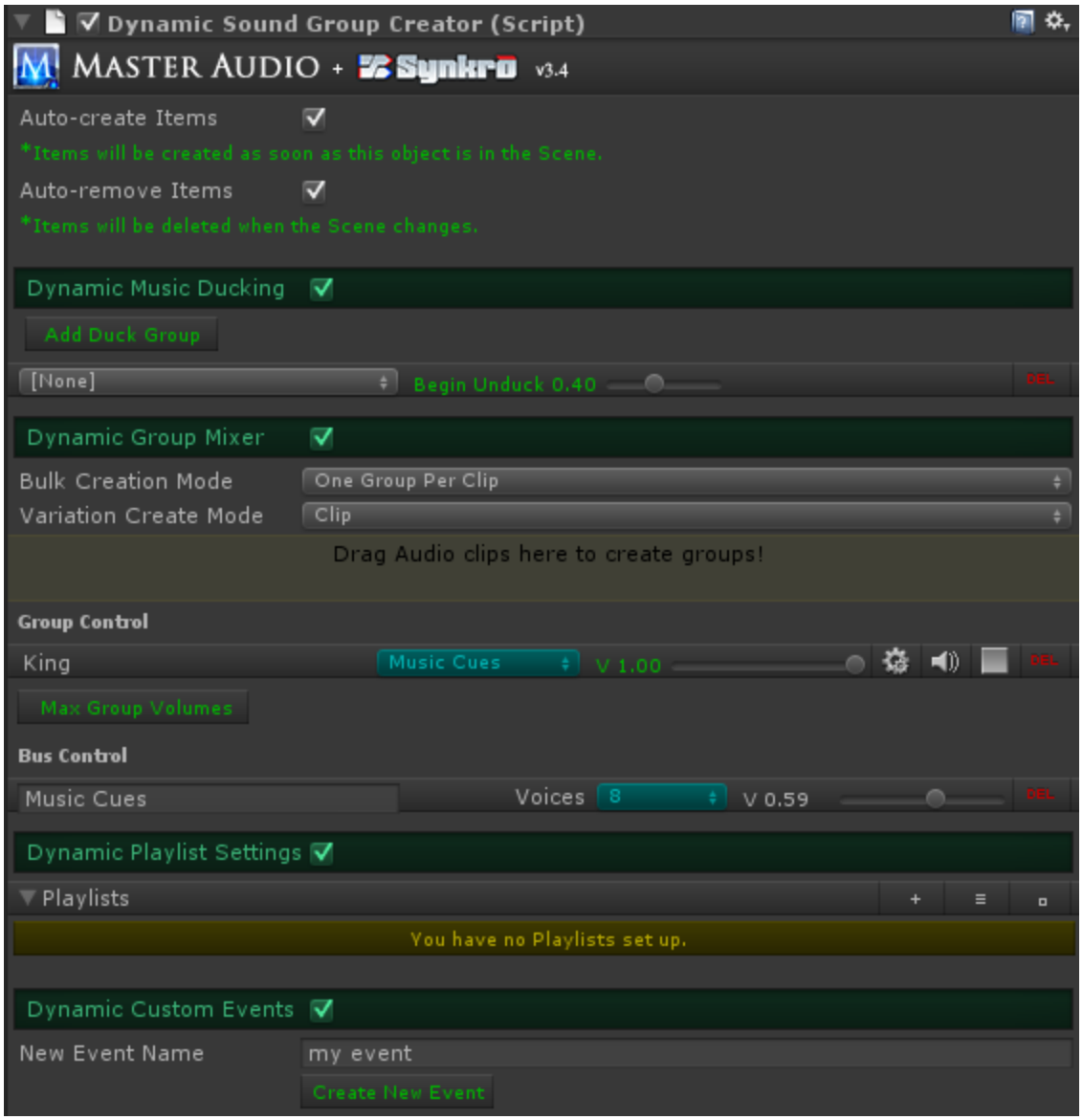
New Jukebox! In Master Audio 3.0, we have a new Jukebox section that shows up when you press play in the editor. It is shown below - there is one shows for each Playlist Controller. It shows you the current Playlist, active song and fading song, plus time remaining on both. Also, there are controls for 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 & Playlists

There is a prefab called DynamicSoundGroupCreator, which can be created from the Audio Manager window. You can use these to create Sound Groups, Playlists and Custom Events based on its settings. This is usually used to set up per-Scene sound effects that will delete themselves from the MasterAudio prefab when the Scene changes when in persist mode.

A second scenario these come in handy is when you're not sure which audio clips you will use before runtime - like if you wanted to change to a "audio skin" of different fire, jump, and run sounds based on user input selection for example. Prefabs with Dynamic SGC's could then be spawned to put these sounds into Master Audio at runtime.

The Inspector looks like a stripped down version of the Master Audio prefab, and is basically a mini-MasterAudio prefab:



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 a script.
2) Auto-remove Groups - If you check this box, the Sound Groups you create are temporary and will removed when this object is destroyed (normally this will be when the Scene changes). You can manually remove the groups by calling the RemoveItems method yourself from a script.
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 Create 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 Music Ducking - this is for specifying ducking settings for the Groups you create in this prefab.
6) 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.
    a) Buses have an additional choice in the dropdown labeled "Existing Bus" which allows you to use a bus that's already created in the Master Audio prefab. You must type the name in though.
    b) You can click the gears icon (settings) to edit the Dynamic Groups. Then you can click the gear icon there to edit the Dynamic Variations of the Group. Full Filter FX controls are under the variation. To save on memory usage, filter FX components are deleted from the Dynamic Sound Group Creator's prefabs as soon as they are created under Master Audio at runtime.
7) Dynamic Playlist Settings - exactly the same as the Playlist settings in the Master Audio prefab, but here you only create temporary dynamic Playlists.

There is a method to create new Sound Groups on the fly.
1. MasterAudio.CreateNewSoundGroup (DynamicSoundGroup aGroup, string creatorObjectName);

All the parameters with values assigned are optional to pass.

8) You can also create per-Scene Custom Events (which are explained in the next section) in this prefab. The controls are identical to the controls in the Master Audio prefab's Custom Event section, except that events you configure here can be auto-deleted when the Scene changes if you set it up that way.

11. Custom Events

In Master Audio, you can define custom events. You create them by giving them a name. Each custom event can have any number of custom event receivers. Master Audio automatically notifies all receivers when the event is fired so they can perform an action if they are configured to respond to that exact event. You can use EventSounds to fire a custom event (when another built-in event such as onInvisible happens), and EventSounds can also receive the event if you want to do a normal MasterAudio function such as Play a sound, change song in a Playlist, fade a bus, etc. The last section in the Master Audio Inspector is Custom Events. Here you create the events available to fire. You can also create per-Scene events in Dynamic Sound Group Creator.

The interface - all Custom Event Receivers (including EventSounds) must implement the ICustomEventReceiver interface. That way MasterAudio will keep track of the receiver and be able to automatically notify it when events are fired. There is a sample class that implements the interface called "MA_SampleICustomEventReceiver" in the Example Scene. It's attached to the Main Camera prefab. You can look at that if you wish to respond to multiple events in a single receiver or perform more customized behavior for events.

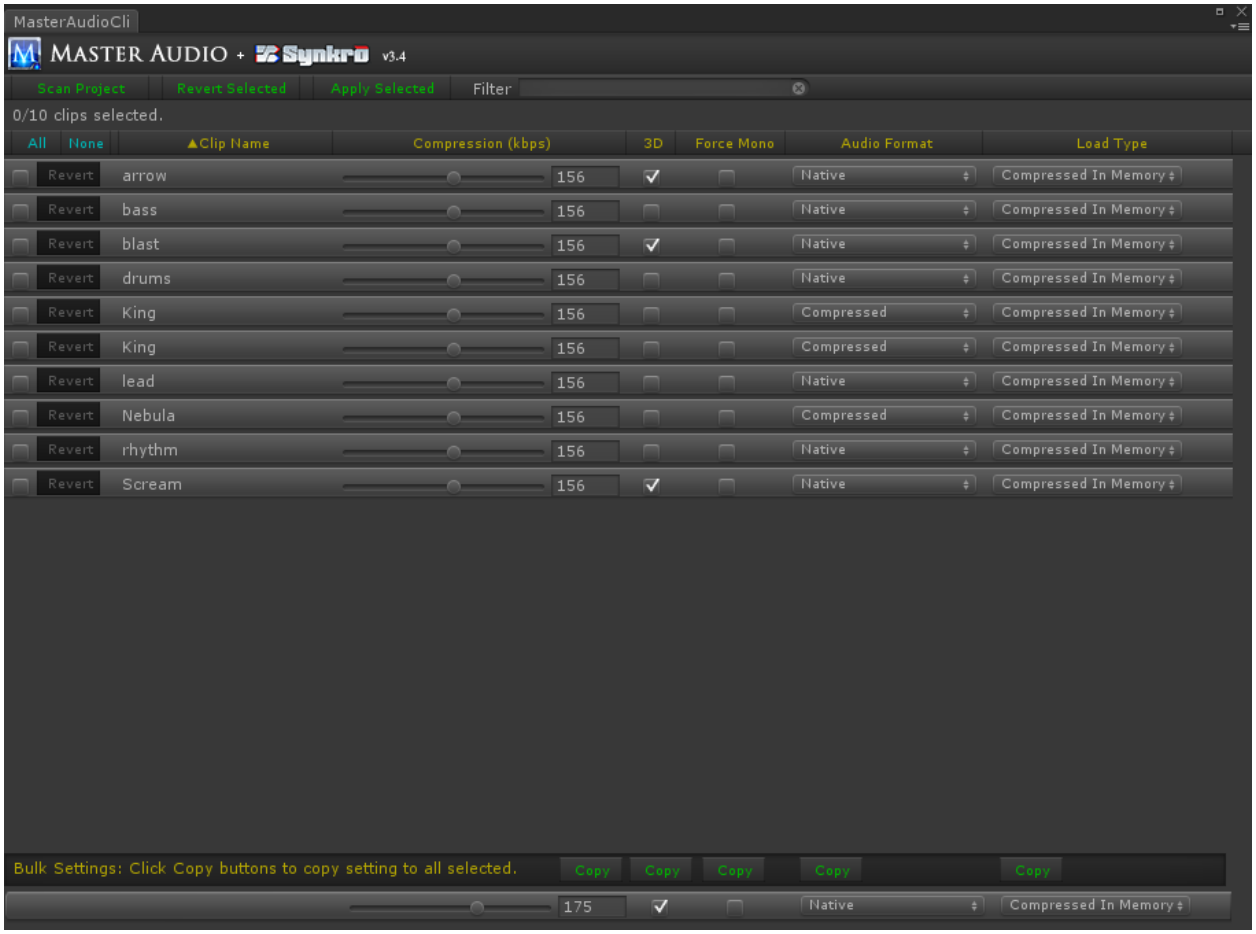
Typical event setup:

1. Create the event in either MasterAudio prefab or Dynamic Sound Group Creator prefab (the latter only if you want the event to be per-Scene instead of permanent.
2. Set up the Custom Event Receiver. Either use EventSounds script or create your own class implementing the ICustomEventReceiver class.
3. Be able to fire the event. You can use EventSounds for this or you can call:

   MasterAudio.FireCustomEvent(string customEventName);

12. Master Audio Clip Manager

There is a second window called Master Audio Clip Manager under the Window menu. It looks like this:



This window helps you to do bulk edits on your Audio files, such as changing the compression bitrate of 10 or all clips at once or setting all music to streaming. It also is useful because you can sort by any of the yellow column headers, and filter by name by wildcard as well.

If you change any field on an Audio Clip, the row will become selected, indicated by highlighting it. Also each field that has been changed will display in green background with yellow text. When you want to apply changes to all selected rows, click the "Apply Selected" button. Or you can Revert Selected as well.

Click on the yellow column headers to sort. Click again to alternate between ascending and descending sort, indicated by the yellow arrow in the column header.

Typing in the Filter textbox will hide all clips whose name doesn't contain the text. This is case-insensitve.

The main section will scroll if there are more than about 16 clips. At the bottom is the bulk settings section. There is a Copy button for each field. That way you can copy just the compression bitrate to all selected clips. When you've copied all the fields you want from the bulk section, remember to click "Apply Selected" to actually make the changes take effect. This may take awhile as Unity processes everything.

Whenever you add or delete clip while Unity is open, if you want to see them in this window, you will need to click "Scan Project" again to grab them.

13. Playmaker integration

We have included the optional "MA_PlaymakerActionsAndScene" package so that you don't have to write any code to integrate with Playmaker. There are 40 custom actions included, under the Audio category. These should cover every method you would call manually. Also included is a *very* simple scene with a PlaySound FSM set up. 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 Stop
7. Master Audio Bus Unmute
8. Master Audio Bus Unpause
9. Master Audio Bus Unsolo
10. Master Audio Ducking Add Group
11. Master Audio Ducking Remove Group
12. Master Audio Ducking Toggle
13. Master Audio Fade Out All Of Sound
14. Master Audio Group Fade
15. Master Audio Group Mute
16. Master Audio Group Pause
17. Master Audio Group Set Volume
18. Master Audio Group Solo
19. Master Audio Group Toggle Mute
20. Master Audio Group Toggle Solo

21. Master Audio Group Unmute
22. Master Audio Group Unpause
23. Master Audio Group Unsolo
24. Master Audio Playlist Clip By Name
25. Master Audio Playlist Clip Next
26. Master Audio Playlist Clip Random
27. Master Audio Playlist Fade
28. MasterAudio Playlist Get Current Clip Name
29. Master Audio Playlist Mute
30. Master Audio Playlist Pause
31. Master Audio Playlist Set Volume
32. Master Audio Playlist Start By Name
33. Master Audio Playlist Stop
34. Master Audio Playlist Toggle Mute
35. Master Audio Playlist Unmute
36. Master Audio Playlist Unpause
37. Master Audio Play Sound
38. Master Audio Set Master Volume
39. Master Audio Stop All Of Sound
40. Master Audio Variation Change Pitch

14. 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.

15. NGUI Integration

ButtonClicker script - this can trigger up to five MasterAudio Sound Groups based on built in NGUI events.

1) MouseDown
2) MouseUp

3) MouseClick
4) MouseHover Start
5) MouseHover End

If you have an NGUI button with a collider, go ahead and attach this script to it. You will notice that the Inspector has dropdowns in the ButtonClicker section. They each contain the list of Sound Groups in MasterAudio.cs.

16. Installation Folder

Installation folder path - by default, it is Assets/DarkTonic/MasterAudio. If you move this folder and still want the Master Audio Manager window to work properly, you will need to open MasterAudioManager.cs and change that path in the variable at the top of that file. Here it is shown:

```
public const string MasterAudioFolderPath = "Assets/DarkTonic/MasterAudio";
```

17. Audio Memory Usage & Tips!

Here are some tips and facts on Master Audio memory usage and optimization.

Unless you are using Resource files (see below for more info), then the amount of memory used by Master Audio will remain constant per Scene. It's not dynamic. All Sound Group Variation audio clips (that aren't Resource files) are loaded into memory when the Scene starts. Extra clones of Variations take up zero memory because they're the same audio clip. Playing a sound does not make the audio memory change (unless it's a resource file).

Typically, to optimize memory usage, we do the following:

1. If the target platform is mobile, we resample all audio clips to 22050Hz so they're smaller. We can't tell the difference in sound. For pure speech sounds, you can get away with an even lower sample rate. We use .wav files for all sound effects, although some users report success with .ogg files as well. We don't like to use compressed audio for sound effects because there are memory and performance implications. Keep it raw!
2. Set all music clips to compressed and streaming which basically take up zero memory (although the profiler doesn't show this correctly when running inside Unity). But don't stream more than one audio clip at a time, that's terrible on performance.
3. Set up infrequently used sounds in a Resources folder and configure the Audio Origin as "Resource File" so they will be loaded only when played and promptly unloaded after done playing.
4. If you only need certain sounds in certain Scenes, set up a separate Master Audio per Scene only with needed sounds, or use the Dynamic Sound Group Creator prefab to populate those sounds in each Scene if you have a "persist across Scenes" Master Audio.

18. Final Words

Support is available by emailing 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 Core GameKit at http://www.darktonic.com/p/developer.html. Thank you!

-All at Dark Tonic