Tazman-Audio

Fabric Manual

FABRIC

Table of contents

Fabric v2.2.0

Fabric v2.2.0

NOTE: The User manual provides an overview of Fabric's features. For a detailed explanation of each property please refer to the Reference Manual.

Introduction

Fabric extends Unity's existing audio functionality and provides an extensive set of high level audio components that allows the creation of complex and rich audio behaviours.

Key benefits of Fabric are:

- **It allows the design of many types of audio behaviours quickly and easily.**

- **All game audio assets are located under one hierarchy making it easy to manage and locate.**

- **Quick integration with the game using an easy and intuitive event base system. Trigger audio in the editor or from code.**

- **Custom user interfaces allow to quickly iterate and focus on the audio functionality that is important.**

- **Multiple instances of the same audio behaviour can be triggered from different Game Objects.**

- **Written entirely with Unity's scripting language, doesn't require external native plug-ins and can be used on any platform Unity supports.**

Fabric v2.2.0

Installation

To install Fabric you simple need to import the Fabric_*verX_unityX*.unitypackage into your project.

Where verX refers to the current version of Fabric and unityX refers to the Unity version that is compatible with.

NOTE: Development versions of Fabric have a letter at the end of the verX number.

The package comes with a number of DLL assemblies, documentation, extensions and set of tutorials.

Fabric audio hierarchy

Fabric uses Unity's object based hierarchical structure in order to build complex audio behaviours.

Each game object must have a Fabric component within it in order to be able to expand the hierarchy.

Fabric v2.2.0
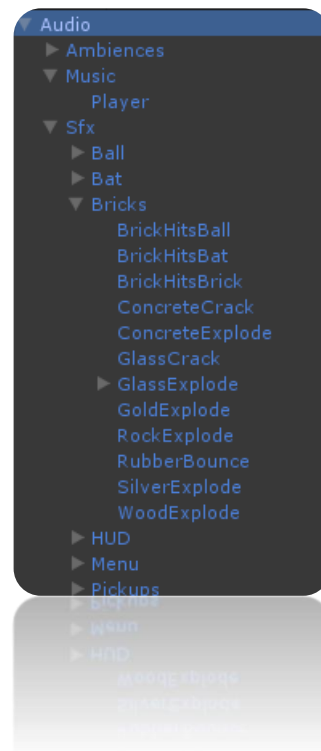
FabricManager

Fabric manager component must always be at the top of the hierarchy. Its main responsibility is to manage the component hierarchy.

A scene MUST have only ONE Fabric manager present.

Fabric v2.2.0

Audio Source Pool

The manager also provides a pool of audio sources that are used by the audio components instead of having to create one for each instance. For example if there is an audio component that has a very big number 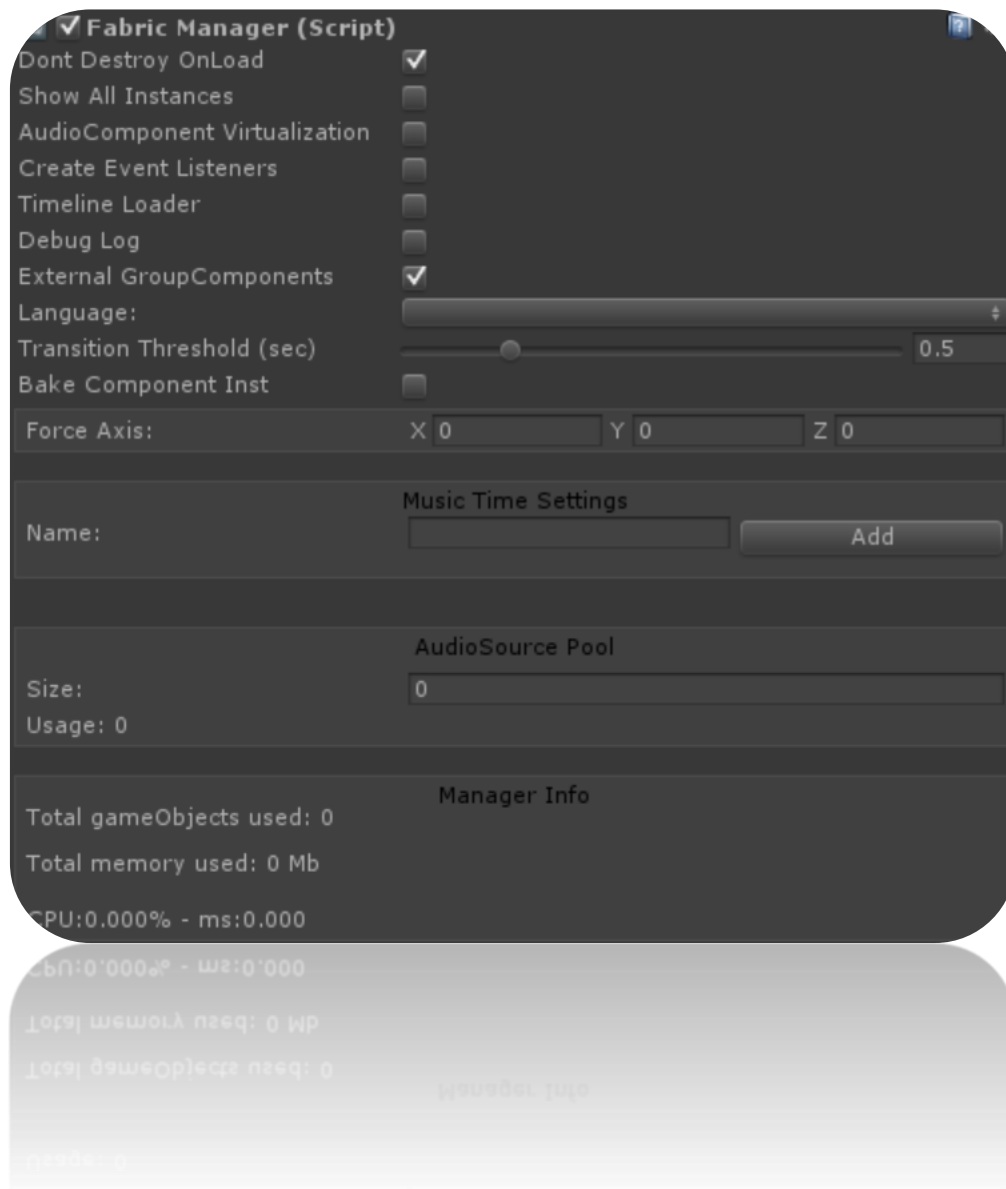Fabric's default behaviour is to create an audio source for each instance. However, using the audio pool the number of audio sources is pre-allocated.

NOTE: At the moment if the maximum number of audio sources are used the pool will fail to allocate a new one.

Usage value shows the number of audio sources that are in use.

Fabric Spring Board

Fabric spring board is a set of scripts (not part of the Fabric DLL) that improve the workflow when working in multiple scenes by ensuring that the Fabric manager will automatically get loaded when is required but also destroyed when the scene is saved so you don't end up with multiple managers in the game causing issues.
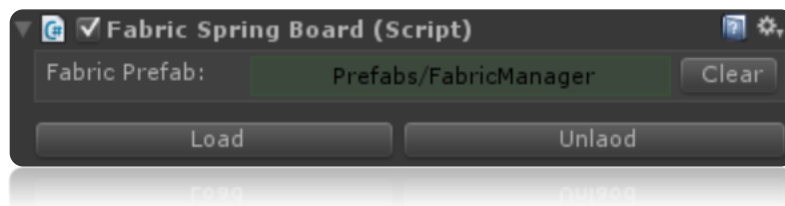
To setup the Fabric Spring Board you need to do the following:

- Create a FabricSpringBoard on every scene that requires audio.

- Place the Fabric manager in a prefab

- Drag and drop the fabric manager prefab into the FabricSpringBoard

It is possible to manually Load/Unload the fabric manager prefab in the scene.

Fabric v2.2.0

Events

Fabric's event based system allows events to be used by the game in order to drive the audio. Each event can be set to perform an action chosen from a list such as: play, stop, set volume, set pitch, set parameters. Events can be implemented very easily in a game either in the editor or by code even before any audio authoring is done. This method allows creating and re-iterating on the audio content of the game quickly and efficiently.

### EventManager

The event manager component is responsible for managing the list of events as well as the flow of events between the game and the fabric components.

NOTE: EventManager component needs to be located in the same game object as the Fabric Manager component. Fabric will automatically add it at runtime if one is not available but it will not be possible to store the event list.
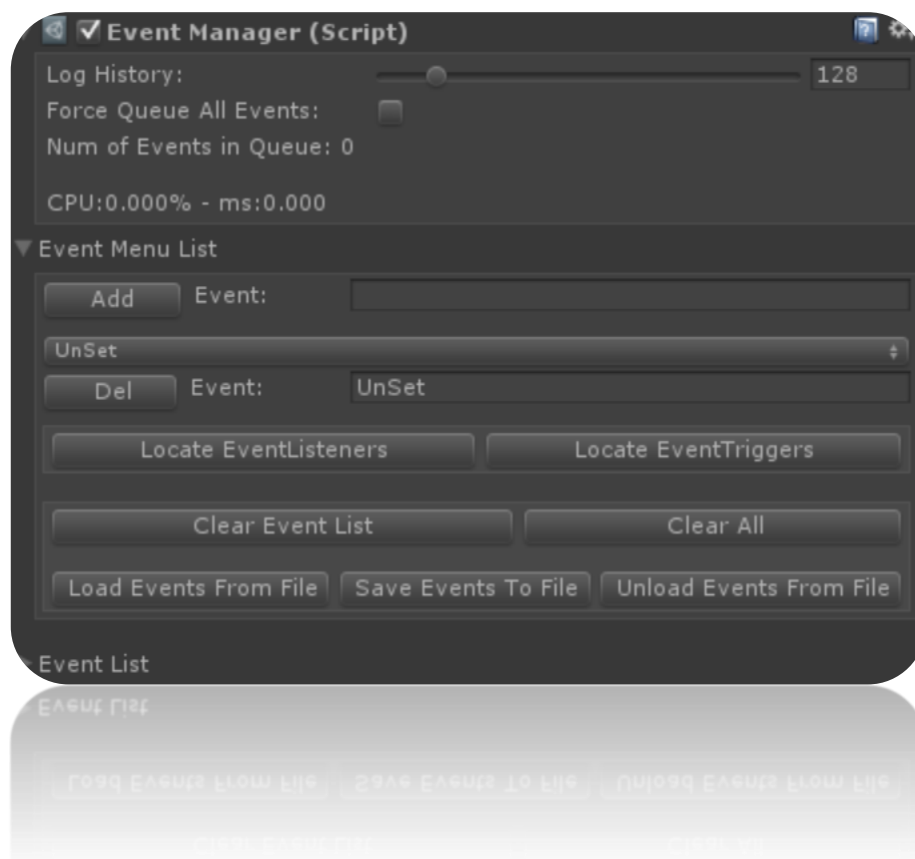
The EventManager allows the creation of event sub-menu hierarchy by using the "/" when typing an event path. So for example the event with the name "Character/Footsteps/Gravel" will be shown as a sub-menu for both the event listener and trigger menu options.
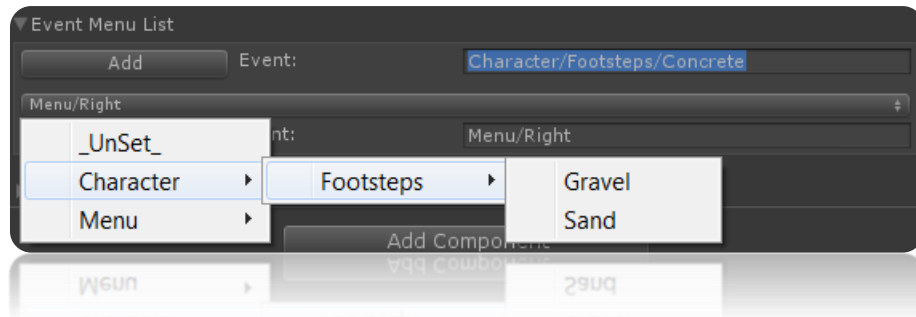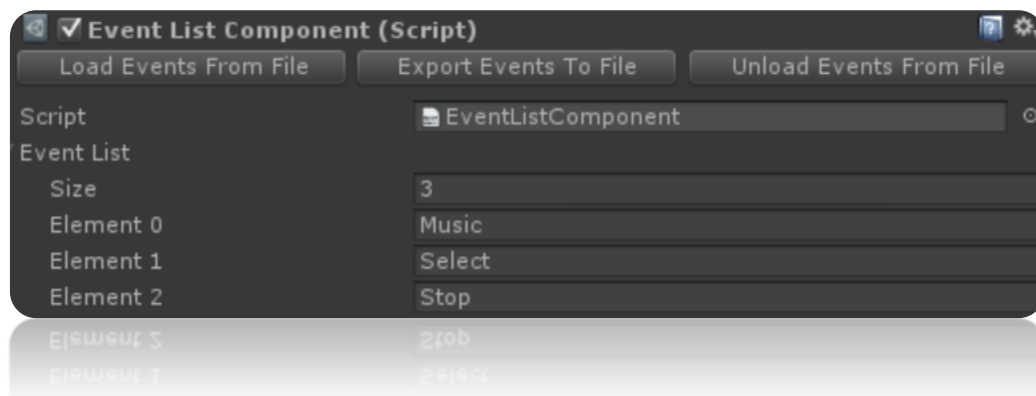
It is also possible to load/save events from/to text files as well as unloading a subset of events that are located in the event list.
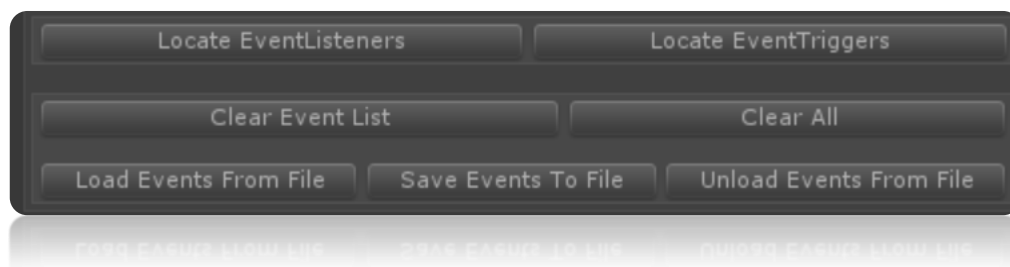
The code API also provides the functions LoadEventsFromFile & UnloadEventsFromFile that can be called at any time to populate the event list.

Fabric v2.2.0

Event List Component

The event list component makes it possible to store a list of events outside of the event manager but make them available when the component is instantiated. This allows them to be bundled together with Fabric hierarchies and audio assets.

Listener

The listener simply waits and listens for a specific event name and only when it receives that name it will respond and perform an action. Furthermore, a listener can accept and set parameters if they are supported by the component.

It is possible to override any or a specific event action from the incoming event and replace it with a new one. This can be very useful since it is possible with a single event to perform multiple actions.

Fabric v2.2.0

Trigger

The Event trigger component is responsible for sending an event into Fabric with a specific action (i.e. Play, Stop, Set Volume/Pitch) and parameter values. Every component that is listening for that event within the Fabric hierarchy will respond accordingly.
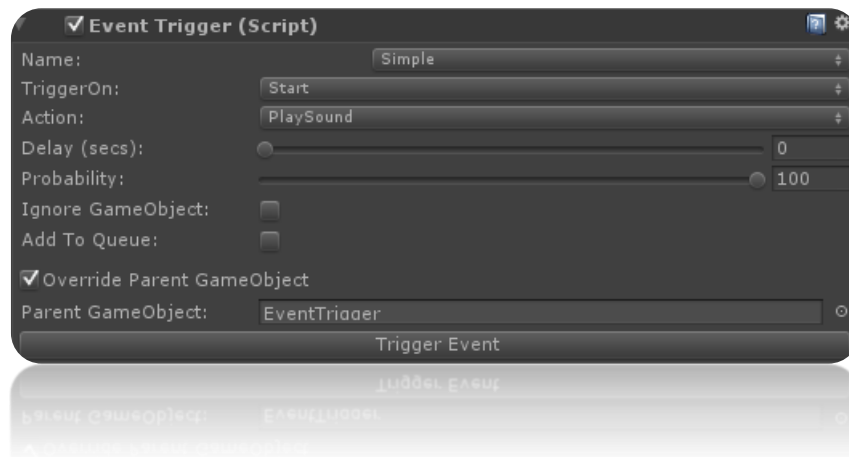

An even trigger can be set to trigger when certain game object function is called or by code through the event manager.

NOTE: It is very important to remember that when a game object is used when posting an event it is associated with a component instance. The same game object **MUST** be used in order to control the instance.

Fabric v2.2.0

Event name persistence

Both event trigger and listener components store the event name even if the event manager or event list component are not present in the scene. This will ensure that the event name will not reset to the default, "_UnSet_", state unless the Clear button is selected.
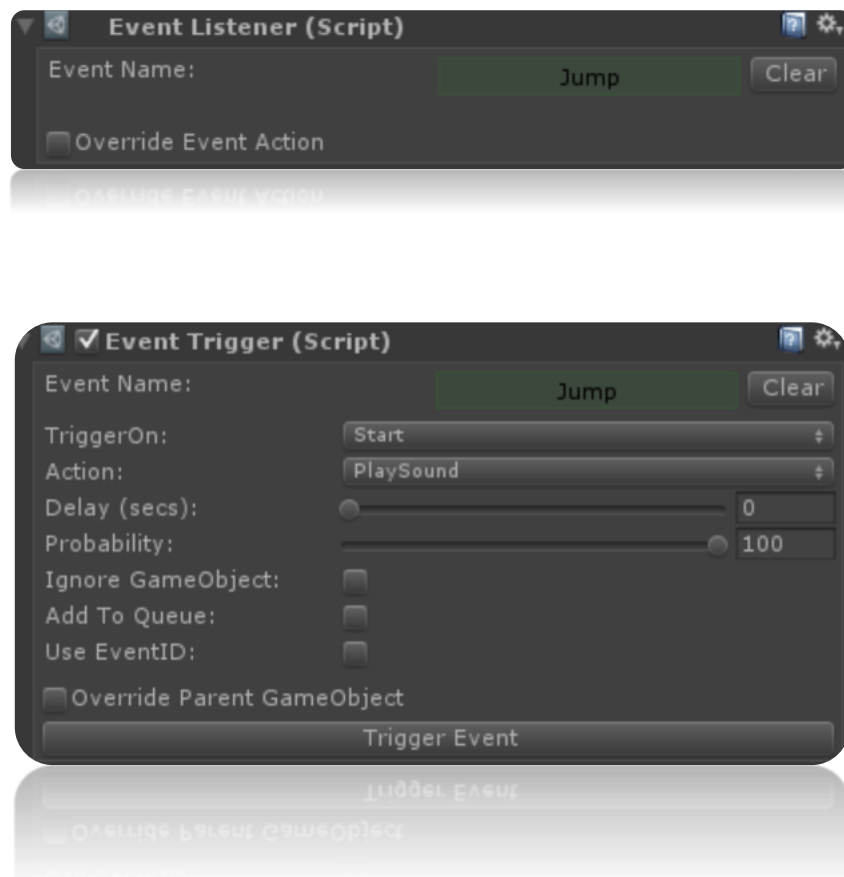
Components

Fabric components are the building blocks that allow you to create sophisticated audio designs using any combination of the following available components:

Fabric provides a menu option that allows to automatically create a new component node in the Fabric hiearchy saving time.
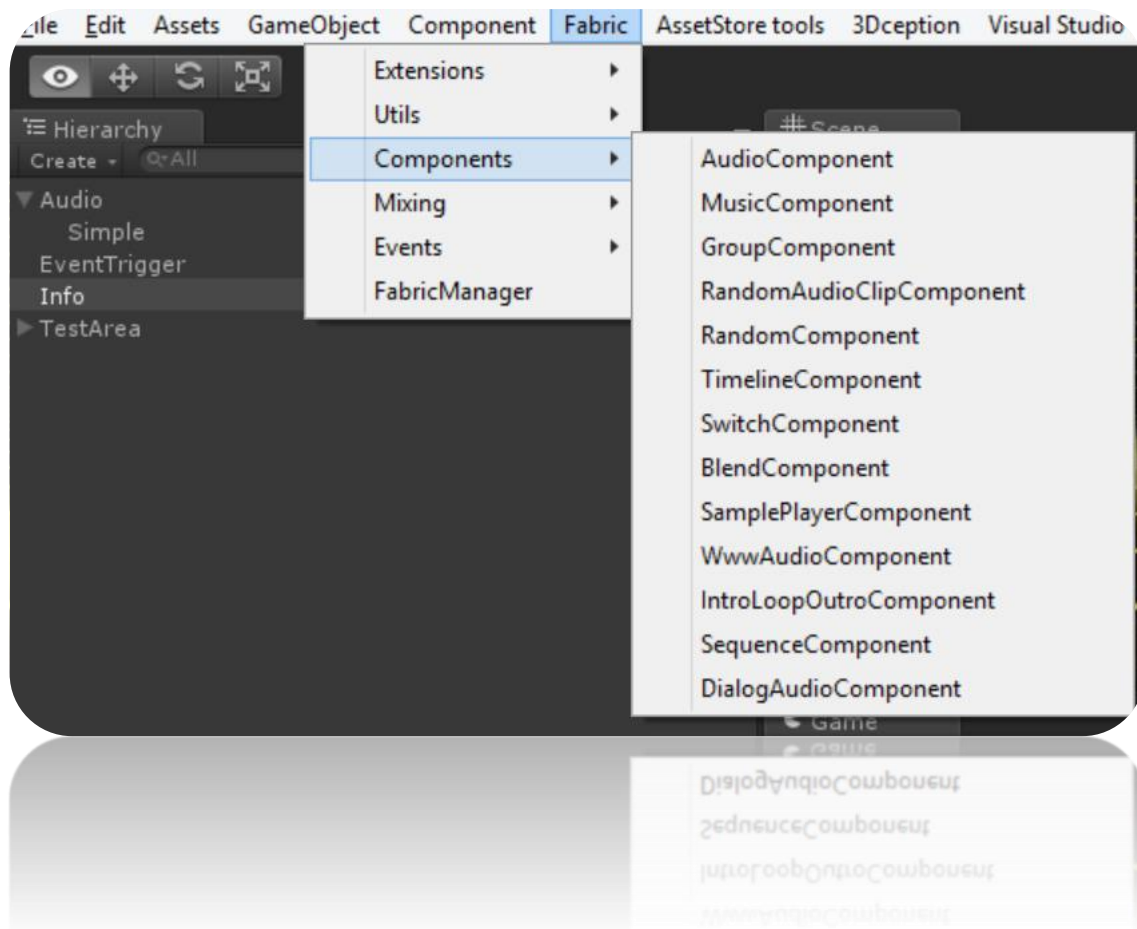
Component properties

All components come with a custom inspector UI that allows modifying their properties according to their position in the hierarchy.

Fabric v2.2.0

Fabric v2.2.0

A component displays different properties according to their position in the hierarchy or if they have a listener attached.

The node info provides an overview of the component internal state and properties such as: the number of instances used/exist, volume and pitch with their randomisation offsets as well as CPU current/max usage.

Propagation of component properties

All of the component properties are propagated through their hierarchy and are either added or multiplied with their parent properties. However, it is possible to override some of the parent properties and use their own values instead.

Volume / Pitch properties:

The component volume and pitch properties by default are always multiplied with its parent volume and pitch properties unless the override tick box is selected which then uses the node's properties.

3D/2D Properties:

The default behaviour of these properties is to passed down from their parent unmodified. When the override tick box is selected for either property then they become the properties that are going to be propagated down the hierarchy.

Fabric v2.2.0

Fabric Manual

Top node:

All components that are just below the FabricManager are "top nodes" so they don't provide any override options.

Child node:

When a component is located as a child of another component it allows to override most of the parent properties.


Fabric v2.2.0

Max instances and virtualization

Any component that has an event listener attached will allow to set the number of maximum instances that can be played along with the priority and stealing behaviour that will be used if all instances are playing and a play sound event has been requested.

The virtualization option allows the component to store all the PlaySound event requests and track their distance from the listener. When an event is within the max distance property the component will allocate and play an instance. If the event moves outside the max distance value it will stop playing it release the instance but keep tracking it. The event will be released when an EventAction.StopSound has been received. This is a great method to start a large number of ambient sources (i.e. torches, forests, birds) each having their own unique audio position but only use a small number of audio resources.

Import Selected Audio Clips

Fabric provides a method to quickly import a selection of audio clips as audio components or dialog audio components. This is particularly useful when it is necessary to import a large number of audio such as dialog files into a random component etc.
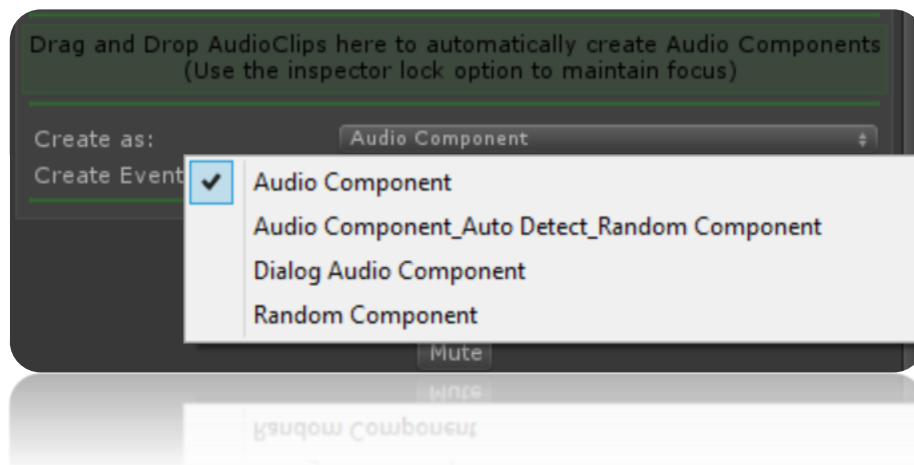
In order to import multiple audio clips first the inspector UI needs to be locked by pressing the small lock icon located on the top right hand corner (red circle).

Fabric v2.2.0
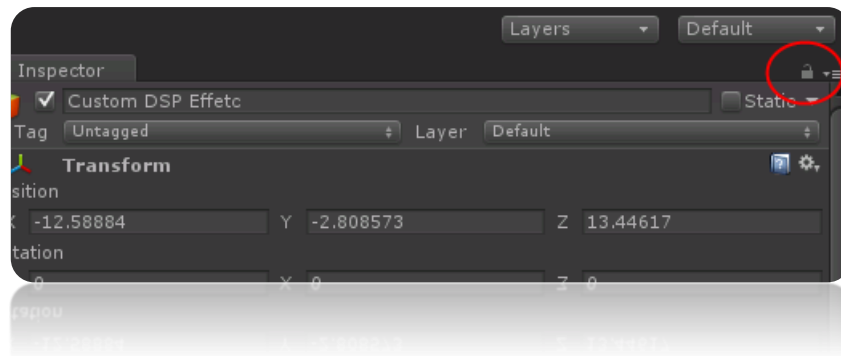
This allows the selection of multiple audio clips without the inspector view loosing focus.

The audio clips are selected in the project tab view can be simply dragged and dropped in the drop box area and for each clip an audio component will be created according to the type selected.

Audio Component

This is the lowest component in a hierarchy and extends Unity's existing audio source component functionality. An audio component cannot have any children components attached.
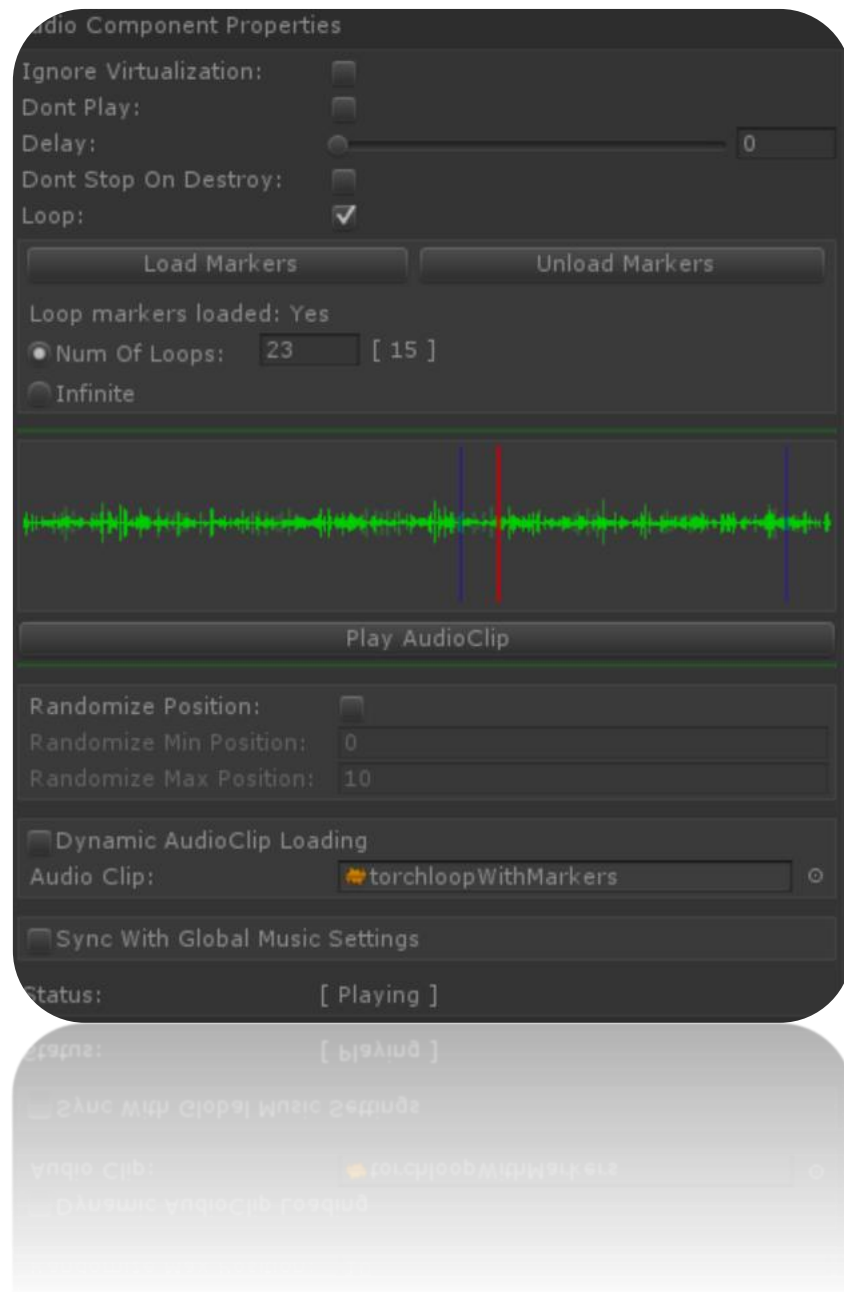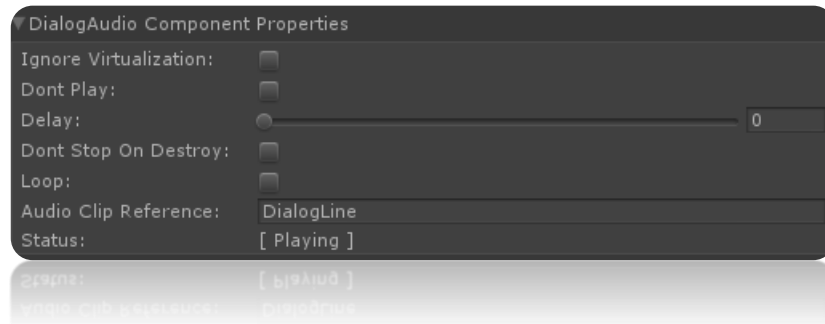
In the audio component you can:

- Set the audio clip that will be used to play audio, delay it
- Disable it from the audio component virtualization (if enabled in the Fabric manager)
- Keep the audio playing even if the parent game object has been destroyed
- Play audio clip
- Read loop markers from wavfile, they have to be two markers in the wafile for Fabric to read them and create a loop region.
- Loop
    o Define num of loops
    o Loop infinitely until stopped
- Randomise position
- Dynamic load the audio clip
    o In Unity 4.5.3> the Resourcesw.LoadAsync is used, in previous version WWW is used
- Sync with global music selection set in the component properties
    o Delay the start of the audio components by a number of beats

Fabric v2.2.0

Dialog Audio Component

Dialog Audio component provides the same properties as the standard Audio Component and its using the Language properties in order to load and play an audio clip on demand. This allows to localise large number of VO in games for multiple languages easily and without having to duplicate Fabric hierarchy etc.

The audio clip will get loaded before the component is about to play and will get unloaded when is finished playing saving audio memory usage.

WWW Audio Component

The WWW audio component utilises Unity's WWW class and in particular its audio support. Using this component it's possible to play audio clips from a http location or a file that is located in any folder (even outside Resources)

Fabric v2.2.0
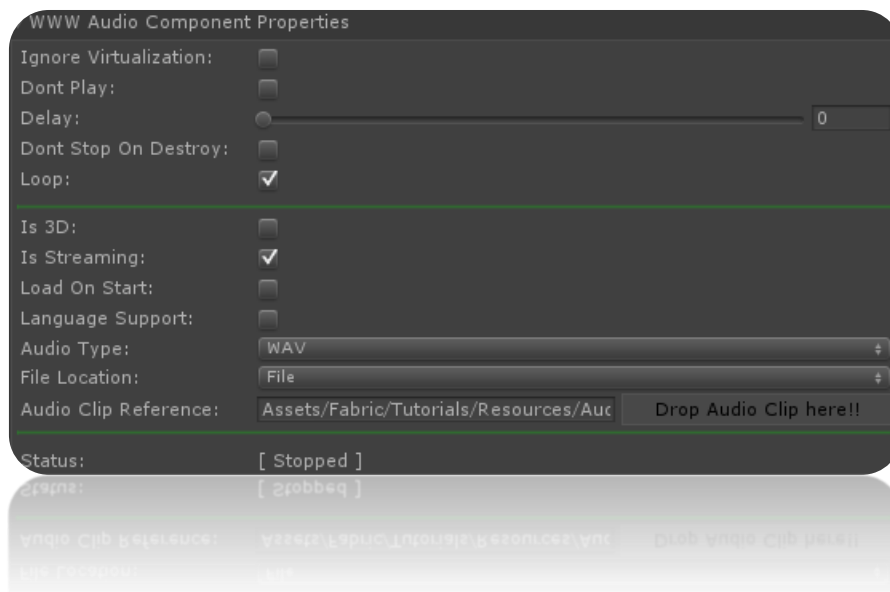
Sample Player Component

The sample player component provides a powerful way to control audio samples in Unity even further. It makes it possible to control the gain of each channel in a multichannel audio file independently. It also makes it possible to read loop region markers from a wavfile or to set them manually from the inspector UI.
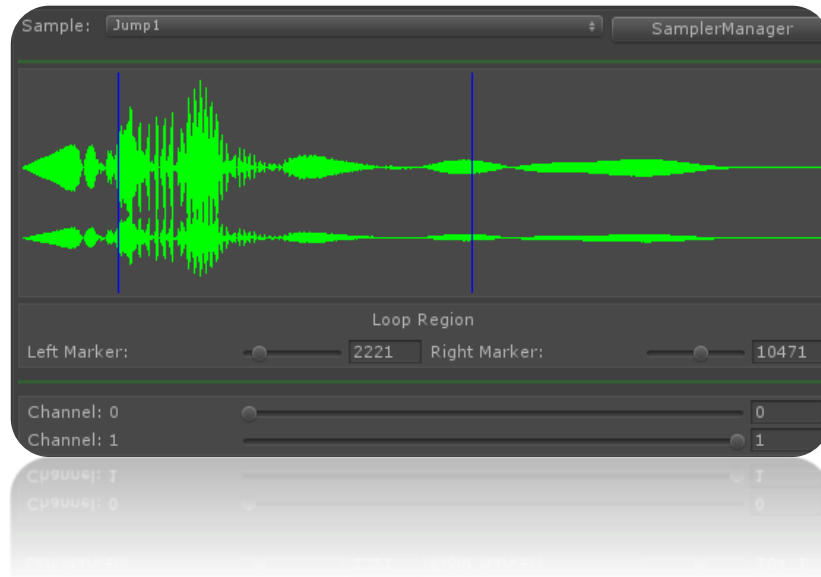
The sampler manager is responsible for managing and creating sample instances that are only visible to the sample player component.

Fabric v2.2.0

To create a new sample file first an empty slot needs to be created (i.e. Add New SampleFile) which allows to drag and drop an audio clip into the red region. Doing this will make the audio clip available in the sample player drop down menu.
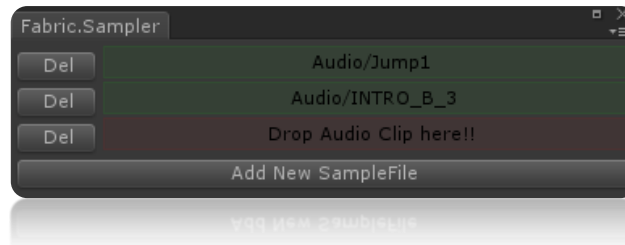
NOTE: This component does have a few limitations with the way it handles audio clips.

- If the audio clip is decompressed it will always use the "Decompress on load" load type
- No streaming is supported
- Loop region markers can only be detected on a wavfile

The sample player also has its own DSP effect panner which exposes the volume properties for each channel in the event system as well as the component runtime parameter support.

Fabric v2.2.0
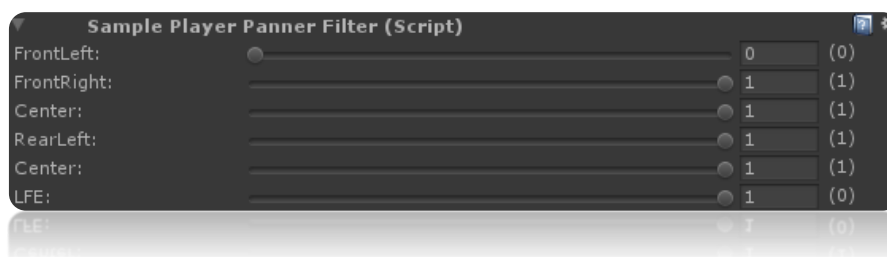
Timeline component

Timeline component provides a powerful multi-track editor functionality that allows layering and crossfading a number of sounds together. It also provides parameters driven by the game that can be linked with the timeline and volume/pitch curves.

A more detailed overview of the timeline features and how it works can be found in the timeline window section.

Random Component

This component triggers a series of child components in a random order or in a way that avoids repetition.
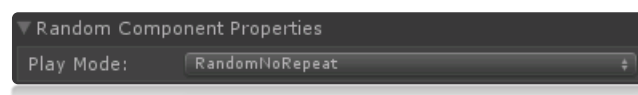
In random mode it is possible to set the weights for each child component:

In RandomNoRepeat mode there is Looped property that will make the random component to behave like a randomised sequence. It is also possible to add a delay and delay randomization properties.

Fabric v2.2.0
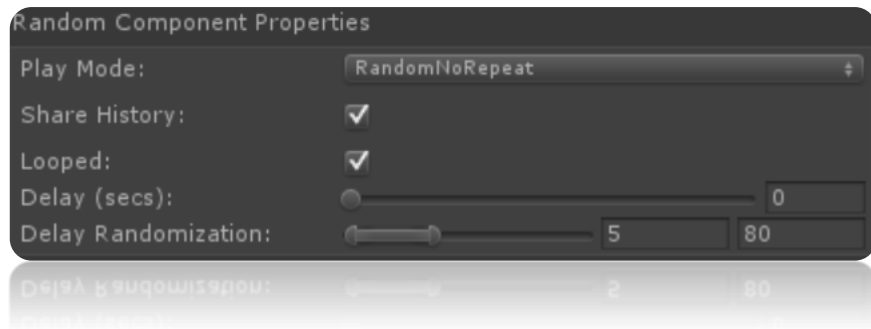
Another useful feature is the option for the random components instances to share the same random no repeat history. This will ensure that you will never hear the same sounds playing in succession if multiple instances are played.

RandomAudioClip Component

Fabric v2.2.0
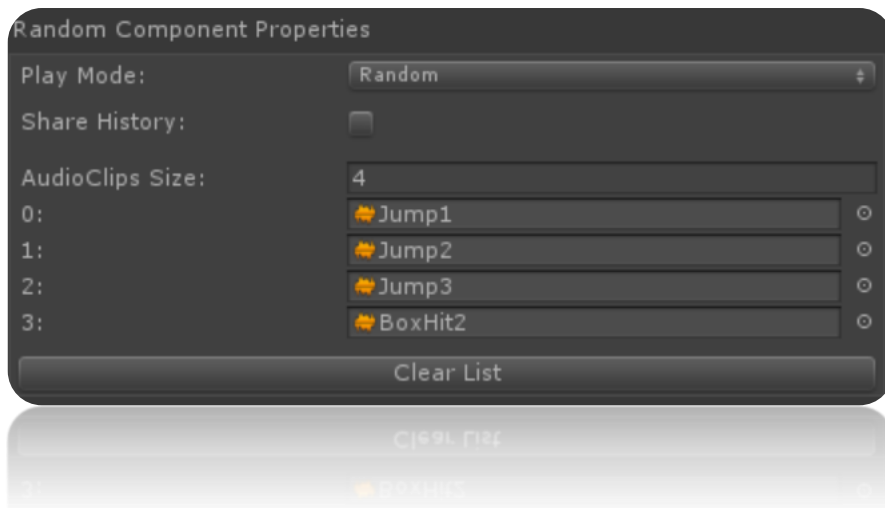
Sequence Component

The sequence component triggers a series of child components defined by a playlist. It continuously advances (PlayContinuous) on the playlist in a sample-accurate way and depending on the play mode selected it will either loops back to the beginning or stop.

Furthermore in advance mode it is possible to configure the sequence to advance to the next entry when an "AdvanceSequence" event action is received (PlayAdvance) or on a PlaySound event action.
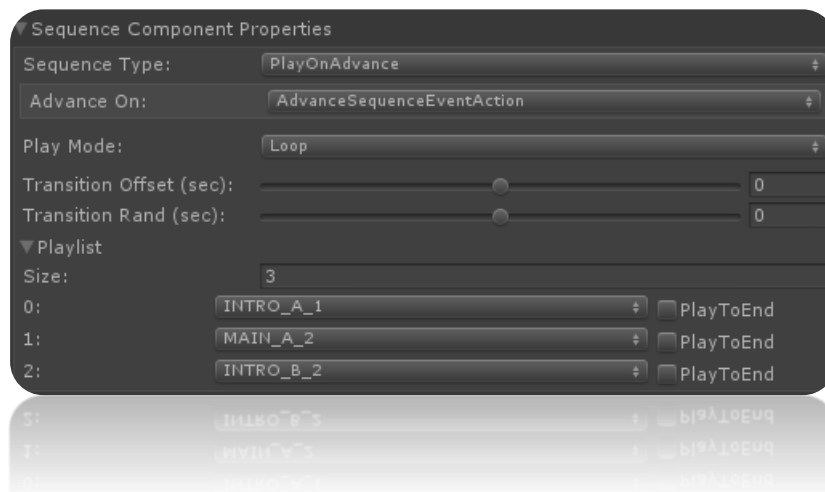
Finally, it is possible to set and randomize the Transition Offset to occur either before or after the end of the current track. This allows you to overlap two entries or introduce delays etc

IntroLoopOutro Component

IntroLoopOutro component provides common sequence behaviour. The Intro will play first followed by the loop. When the component is stopped it will play the Outro section before is stopped

Fabric v2.2.0

The component provides the same transition properties as the Sequence component. An offset will either add (if >1) a delay between each segment or play them early (if < 0). The transition rand parameter adds a randomisation value on top of the offset.

Switch Component

The switch component selects which child component to trigger from an option that is set by the game. The option is **ALWAYS** the name of the component that is required to trigger.
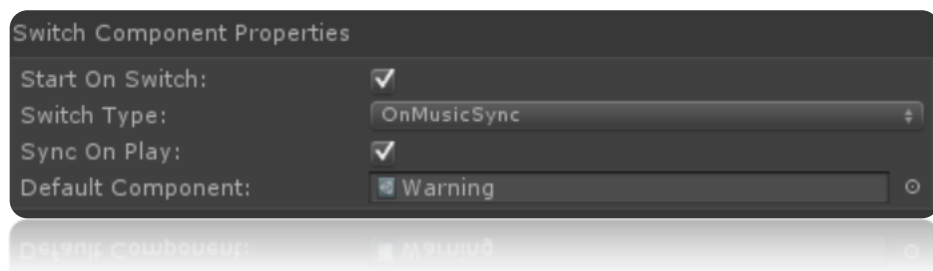
Blend Component

The Blend component is very similar to the group component, it triggers all its child components at the same time but it doesn't appear on the mixer view. Also through its inspector UI it is possible to set the children volumes, saving time from having to go to each node.

Fabric v2.2.0
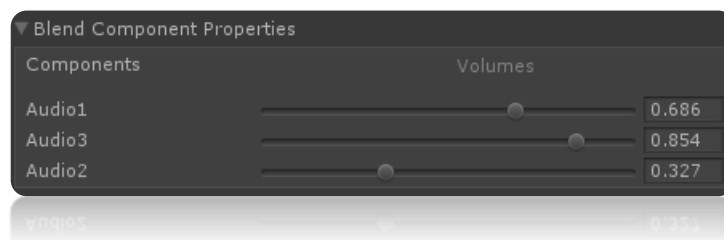
Group Component

The group component controls the volume and pitch properties of all the sounds in its hierarchy.
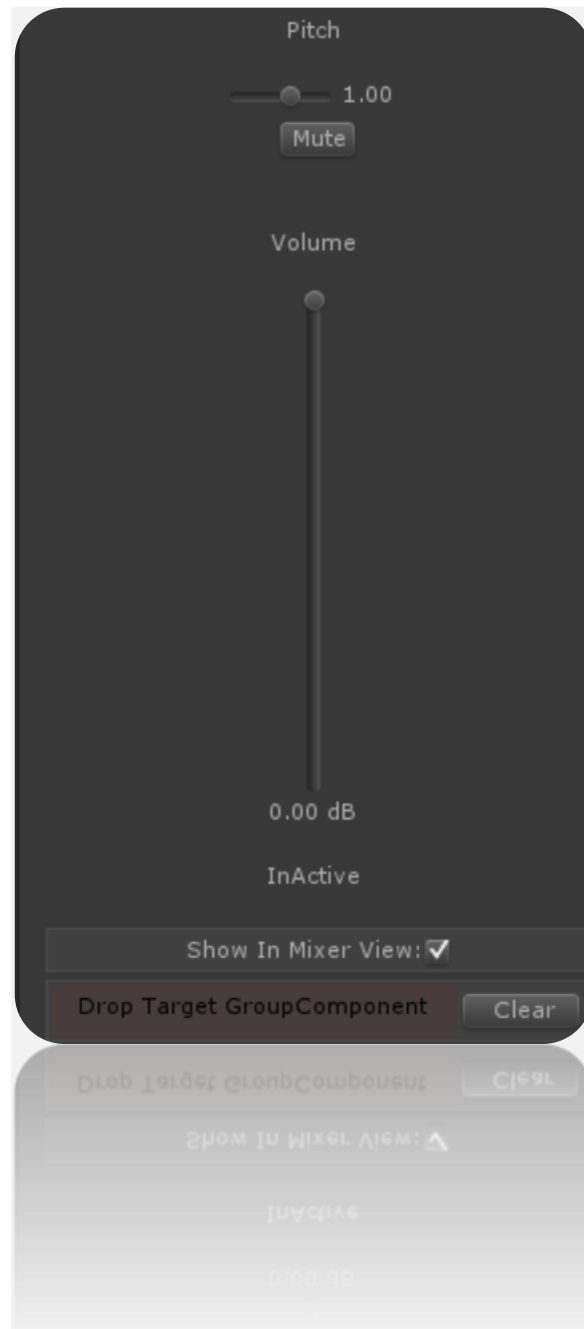
When a group component is placed outside the main Fabric hierarchy and the External Group Component option in the Fabric manager is set it will be automatically registered in the hierarchy

.
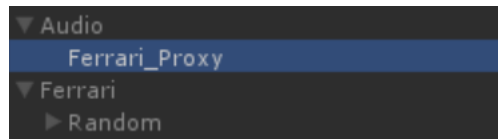
The proxy component will display the group component that is associated with,

Volume Meter Component

The volume meter component displays the audio levels of the node in which the component is added. It will try and collect a mixed down version of ALL audio sources that exists below the node.

Fabric v2.2.0
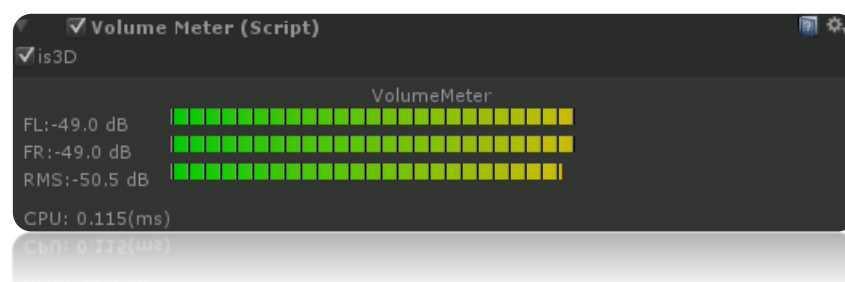
Loudness Meter Component

Loudness meter offers EBU R 128-compliant loudness metering which measures the loudness levels that listeners will perceive.

There are three standard types of loudness metering available:

- Momentary (M) loudness with an integration time of 400ms.
- Short-term (S) loudness with an integration time of 3s.
- Integrated (I) loudness which is a long-term metering with a gating function.

Fabric v2.2.0

Side chain Component

This component by default will use the volume meter's RMS property in order to reduce the volume of the component in which is added.
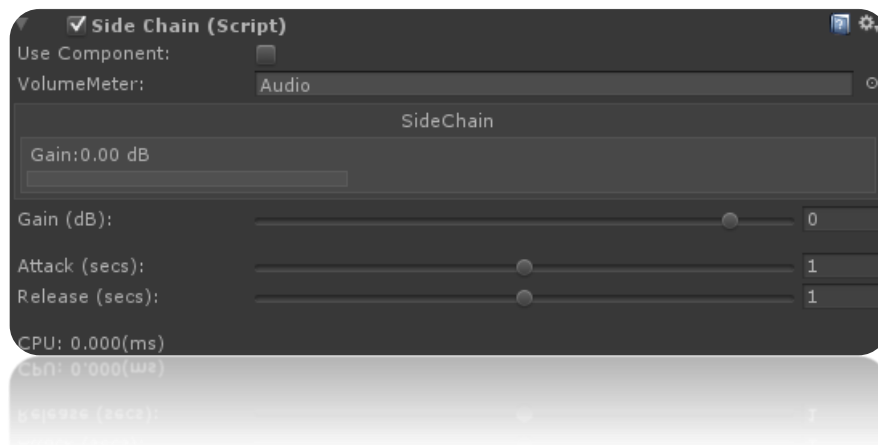
However if the "Use Component" property is checked it is possible to link any Fabric component and the side chain will get activated for as long as the component is active. This provides a less accurate but very low CPU usage side chain control.

A typical method using a side chain is reducing the music levels during speech.

Fabric v2.2.0

DSP Components

Fabric provides a number of custom DSP effects on top of the DSP effects that Unity supports. A DSP component can be inserted anywhere in the Fabric hierarchy. At runtime it will collect all audio sources underneath that node and will add the effect on each source. This way it's possible to set an effect property at the same time for all instances either through the component inspector UI, Fabric's event system or in code using the API.

Fabric based audio effects

Stereo Spreader

This effect attempts to re-create a stereo signal from a mono source.
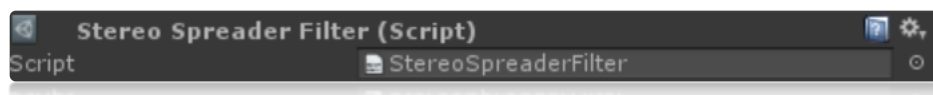
Audio Panner

The audio panner effect allows to control the volume of each channel.
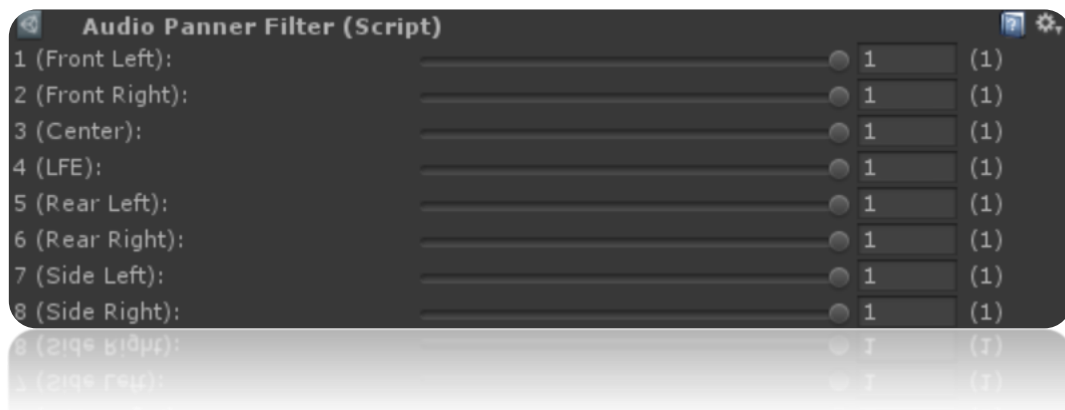
NOTE: In order to control a multi-channel audio the Default Speaker Mode option in Edit->Project Settings->Audio menu option must be set to the appropriate output.

Fabric v2.2.0

Sample Player Panner Filter

The sample panner works in the sample player component and allows to control the gain of each channel on the audio clip.
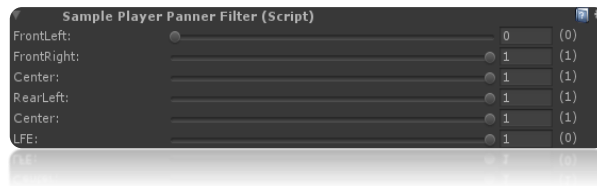
Audio Capture

The audio capture effect can be inserted on the audio listener or audio component and it will capture the audio playing ready to be exported into a wav file.
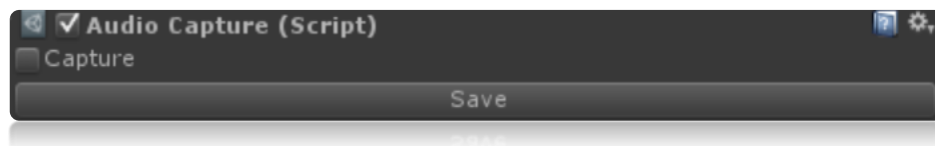
Unity based audio effects

Fabric provides DSP component for all the unity audio effects. For explanation of each DSP effect parameter refer into the audio section of Unity's documentation and Fabric reference manual.
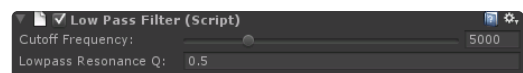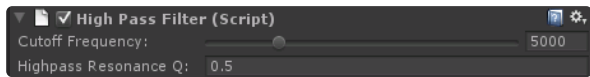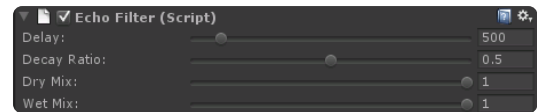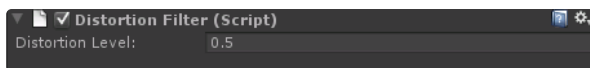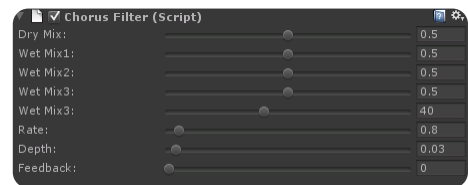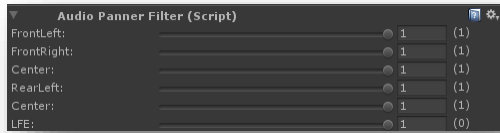
It is possible to set a DSP parameter from the event trigger component by setting the EventAction.SetDSPParameter. The event trigger editor will show all the DSP effects with properties.
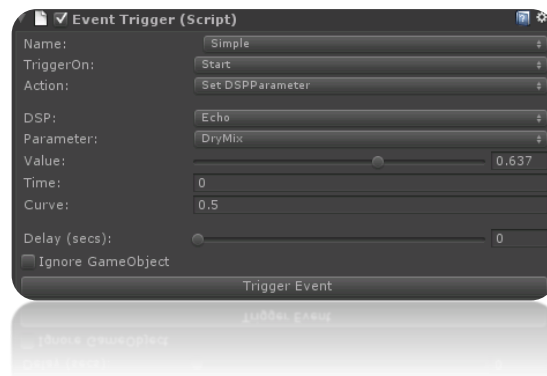
Dynamic Mixer

The dynamic mixer component changes the audio balance of the game at runtime. Its strength lies in the ability to add multiple presets together, each changing the volume or pitch properties of a group component. The properties from each active preset are added together to produce the final volume and pitch values that are then passed to the group components. It is also possible to switch from one preset to another thereby giving a nice, smooth transition of the parameters.

In the dynamic mixer inspector it is possible to create or delete presets that can be called up at any time either from the event trigger component or programmatically using the API.
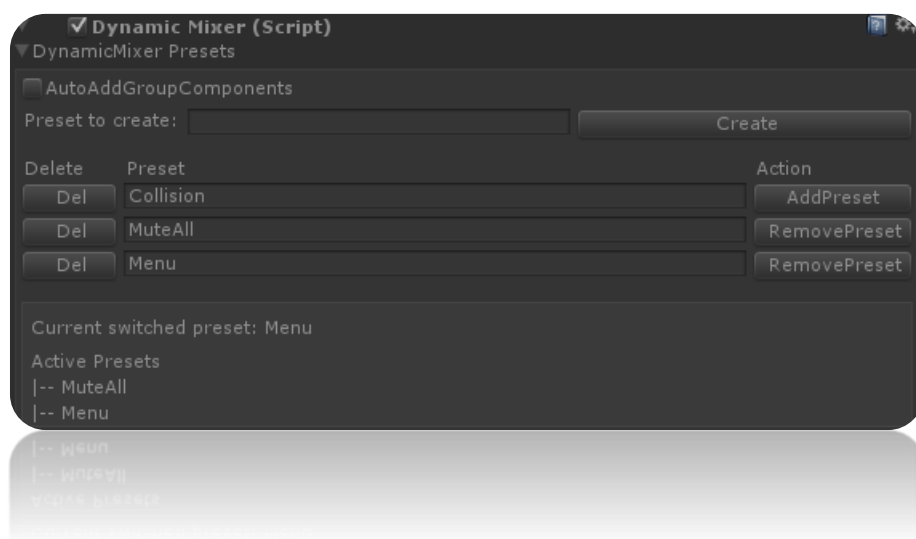
Preset

Presets is simply a collection of group presets with a unique name. Presets can be activated at any moment either from the event trigger component or through the API.
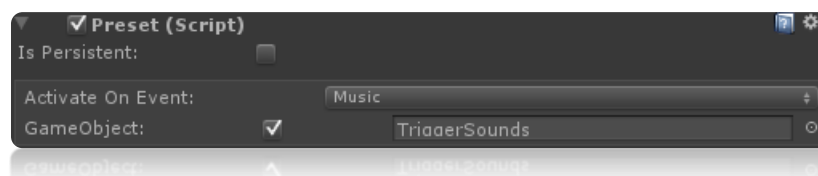
A preset can be persistent which allows it to remain active even when the dynamic mixer receives a reset event action. It can also be activated automatically when an event is posted and remain active for as long as the event is playing. When the event is no longer active the preset will be deactivated.

Fabric v2.2.0

This is a low CPU usage ducking technique which could be used to replace side-chain when performance is an issue.
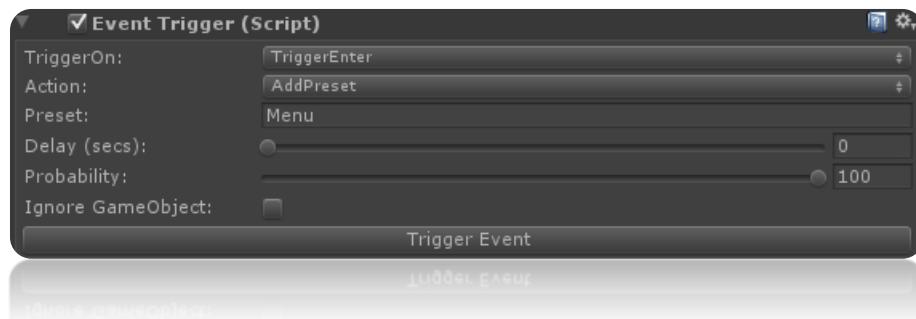
Group preset

A group preset defines properties that will make a change to an assigned group component. It is possible to change the volume, pitch and specify the duration and curve type (i.e. linear, log, exp) for each property change.
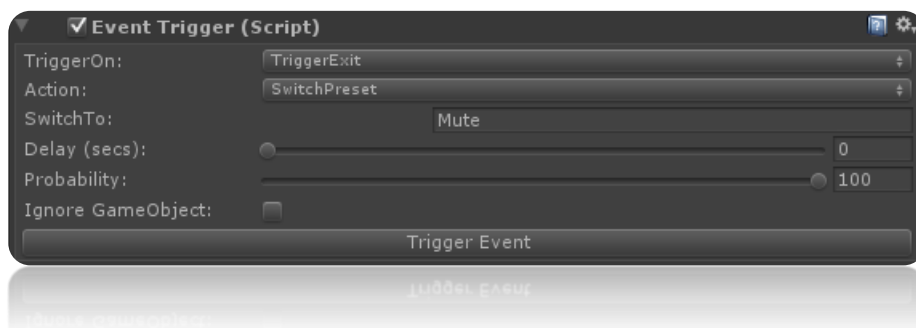
Fabric Manual
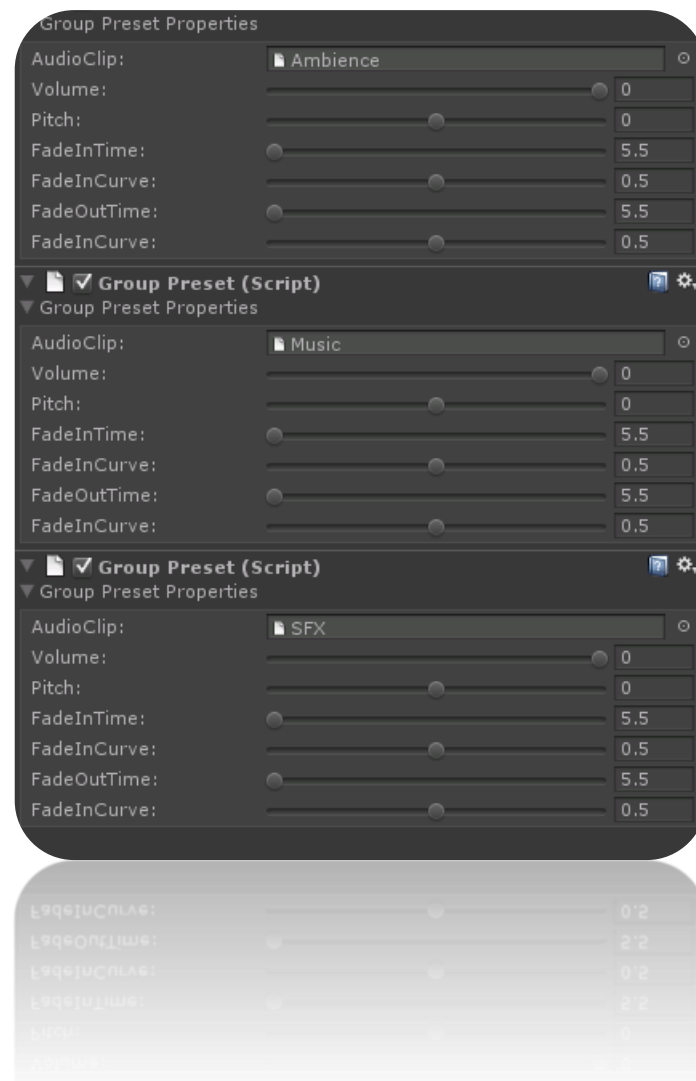
Adding presets at runtime

Adding a preset at runtime could be done either through the event trigger component or the API.

Switch preset at runtime

Sending a "SwitchPreset" event action will cause the mixer to switch from the currently active preset to the target preset.

Fabric v2.2.0

Music support

When a component is set to music sync it will display a Music Time Setting property in the common properties section. In there it is possible to select the music tempo that the component will respond to.
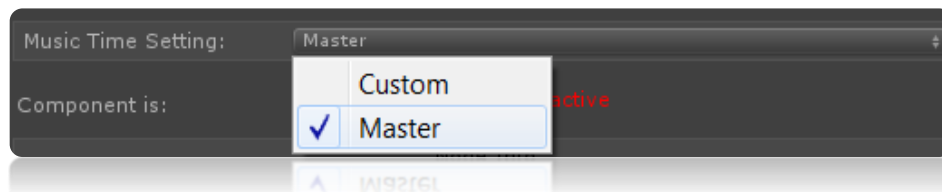
Music Component

This new music component behaves in the same way as a switch component, In that it allows to switch between components, but with the addition of allowing to play a transition segment in between.
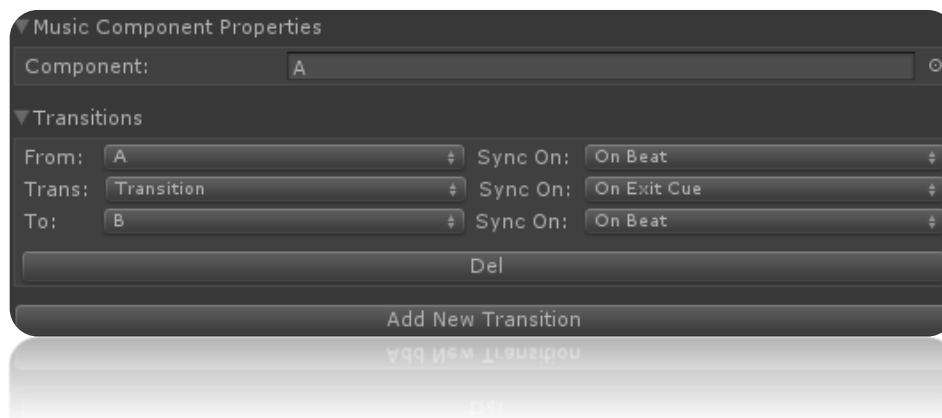
This is possible by adding/removing transition behaviours that apart from the transition segment they also allow to set the type of sync (i.e On Beat, On Bar or On Exit Cue).

Fabric v2.2.0

Any component that exists underneath a music component will allow to override the music settings. This allows to support music tracks that have different tempo sections.

NOTE: The override music properties option is only available if the component is under a music component.

Audio Component

Any audio component can be synced with music by enabling the "Sync With Global Music" option and choosing from the available music settings.

NOTE: This option is disabled if the audio component is in a hierarchy that already syncs to music.

Fabric v2.2.0
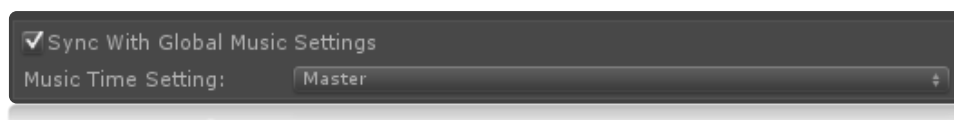
Sequence Component

Sequence component now supports the music sync feautres and allows to advance to the next entry on the playlist according to the sync type set on the music timing settings selected.
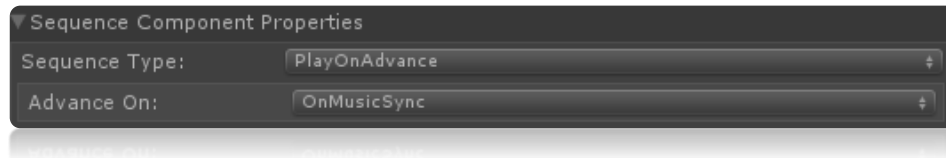
Switch Component

Switch component also allows to select the switch type to be on a music sync making it possible to switch to a child component on a beat/bar.

Fabric v2.2.0
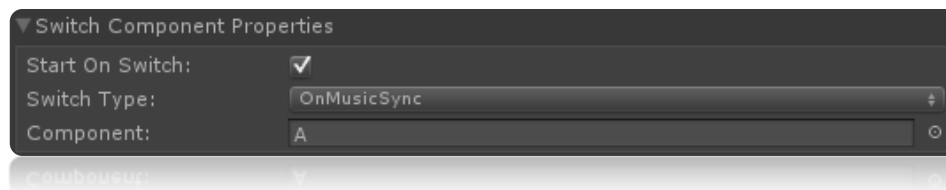
Utilities

Debug log

DebugLog provides the option to choose which type of messages will be displayed in the console output therefore reduces the amount of information displayed.

NOTE: The DebugLog component needs to be located in the same game object as the Fabric Manager component. Fabric will automatically add it at runtime if one is not available but it will not be possible to store it's properties.
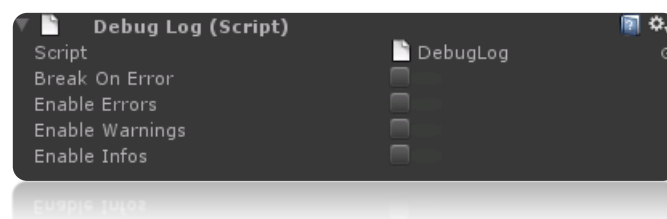
AssetLoader

AssetLoader is a component that loads and unloads prefab assets that contain Fabric components. It is possible to define when the asset will load the list of its prefabs such as: Start/Destroy, Enable/Disable, Trigger Enter/Exit, Collision Enter/Exit.
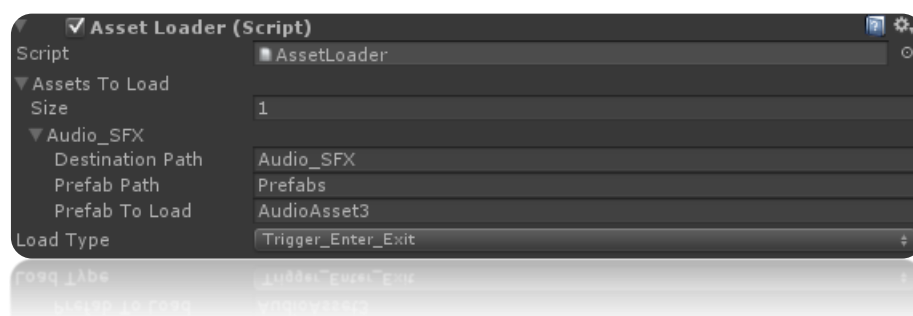
NOTE: On the destination path each node name MUST be separated with the "_" character.

Fabric v2.2.0

Fabric windows

Fabric provides a number of custom windows that allows quickly iterating and focusing on the audio functionality that is important.

Timeline

The timeline component is a powerful multi-track component that allows layering and crossfading a number of sounds together, linking the timeline and volume/pitch curves with game parameters.

A timeline component can be added in the hierarchy by selecting the Fabric->Components->Timeline menu option.
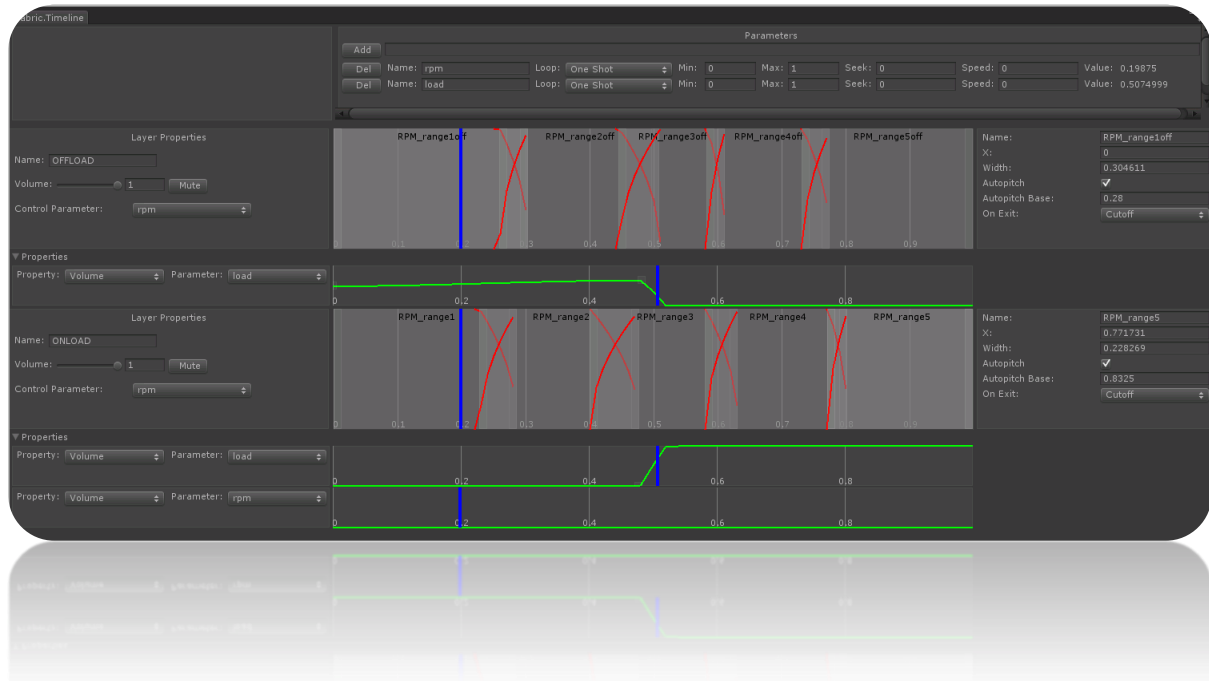
Parameters

A timelime parameter provides the link between the game and the timeline internal functionality. With a parameter it is possible to trigger regions that are placed in a layer as well as control property/parameter graphs.

A timeline parameter can be controlled using the event trigger component or through the API.

When a timeline parameter is set to Reset On Play it means that every time the timeline plays again the parameter resets back to its last position.

When a layer has a controlled parameter selected it displays a blue vertical bar in the timeline view. The length of the timeline is mapped to the range of the parameter.

It is possible to select the parameter bar and drag it left or right or use the slider displayed in the timeline inspector view.

Fabric v2.2.0
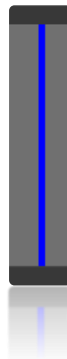
Layer

A layer is added into the timeline by right clicking into an empty region and selecting the "Add Layer" option.
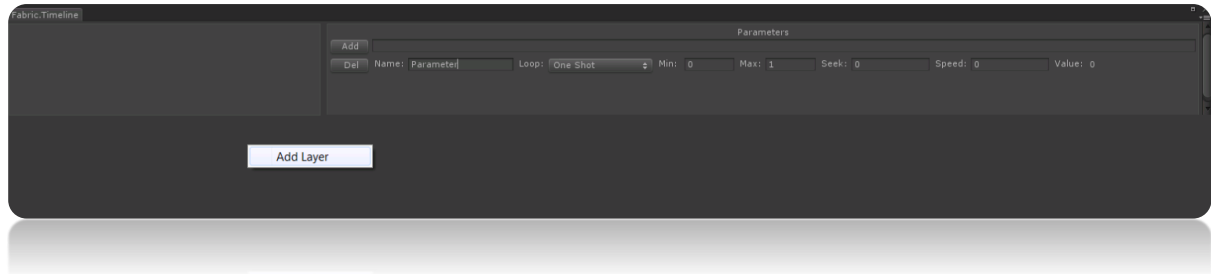
A layer provides a timeline in which multiple regions can be inserted. When two or more regions overlap a crossfade is applied.
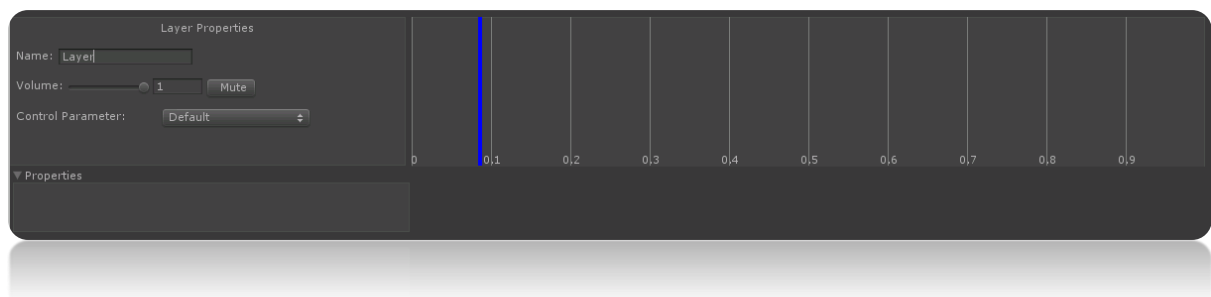
Region

A region defines an area in the timeline in which any component will be activated when the parameter cursor is entered, and stopped, according to the "On Exit" behaviour, when is exited.

A region can be added by selecting anywhere in the layer timeline and right clicking.

Fabric v2.2.0

A region provides a number of properties. For a detailed description of each property please refer to the reference manual.
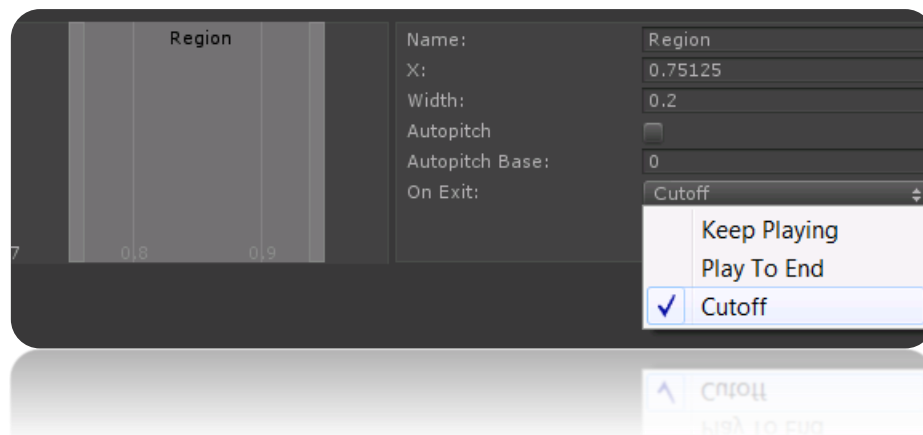
When a region is created it is possible to right click on it and add any of the Fabric components shown in the context menu.
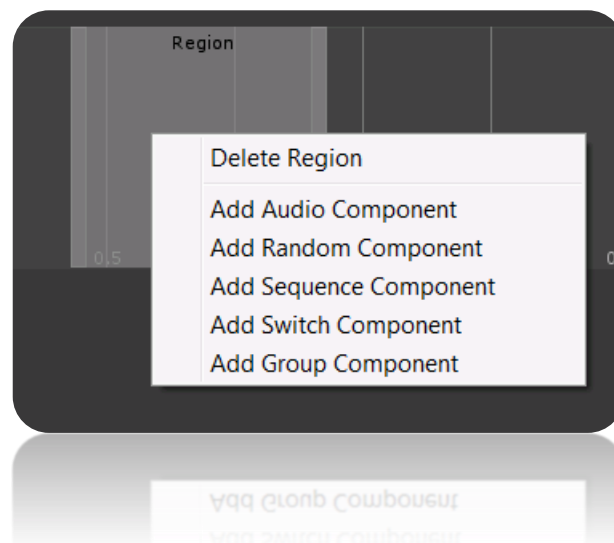
On each layer it is possible to add property/parameter graphs that allow to link and control a layer property from a timeline parameter such as volume and pitch.

Fabric v2.2.0
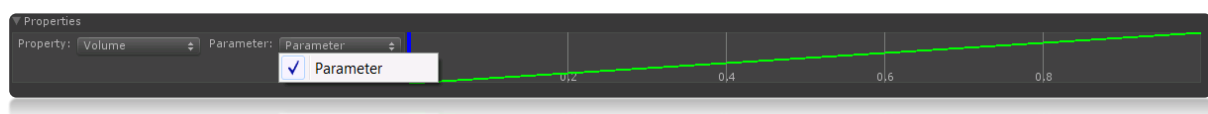
It is possible to add points in the graph by right click on the line. It is also possible to define the type of curve for the line between two points.
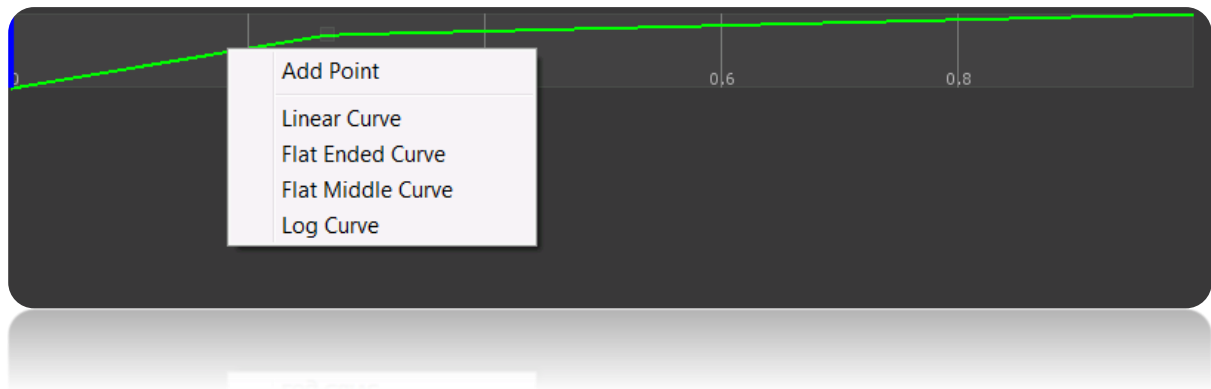
A point can be deleted by selecting on it and pressing the mouse right click.

Fabric v2.2.0
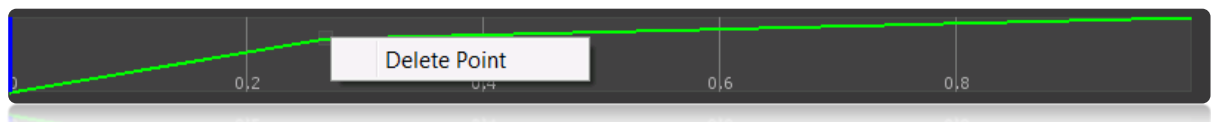
Runtime parameter (RTP) Window

The runtime parameter (RTP) window allows you to link parameters from the game with internal component properties.
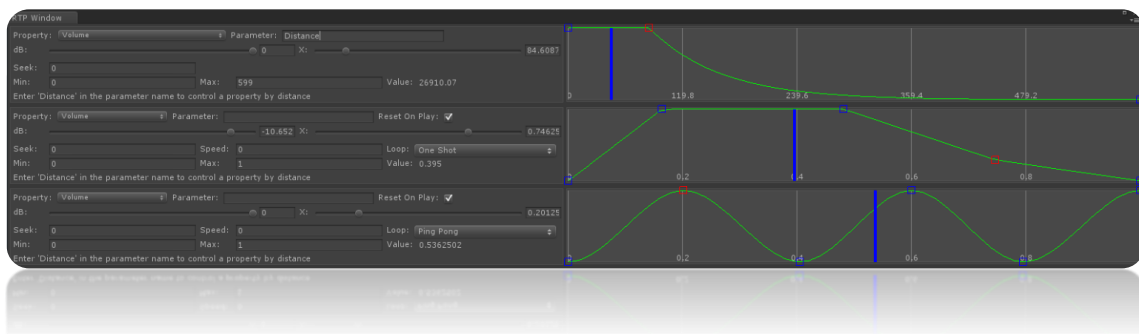
The RTP parameter area allows setting which property a parameter will control. It is possible for a parameter to have a seek value that defines the amount of time it will take to reach the target value, speed which moves it continuously in one direction and loop type that determines if the parameter that has speed will loop when it reaches the end.
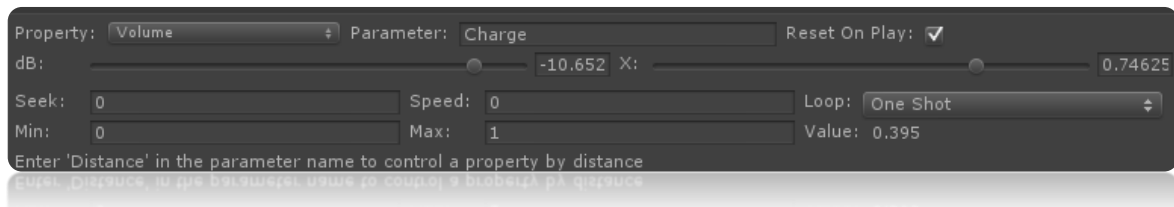
It is possible to get access to the distance value between the component and the camera if you type "Distance" in the parameter name. This way it is possible to control any of the component properties using the distance parameters.

Fabric v2.2.0
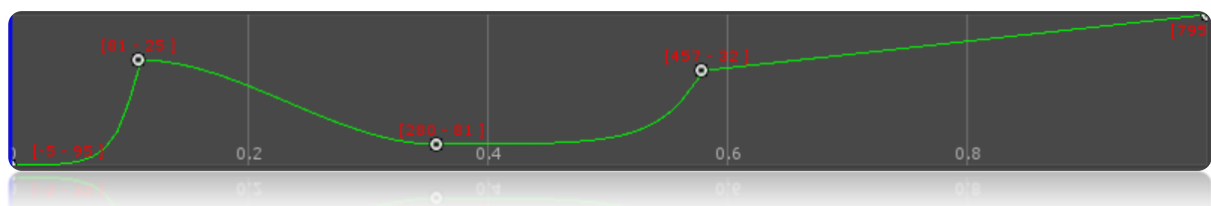
The graph defines the mapping between the input parameter and the property that will be changed. Just like the timeline it is possible to add/delete points in a graph, change the curve line between points (i.e. linear, log, exp etc)
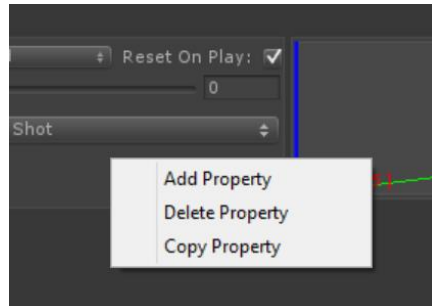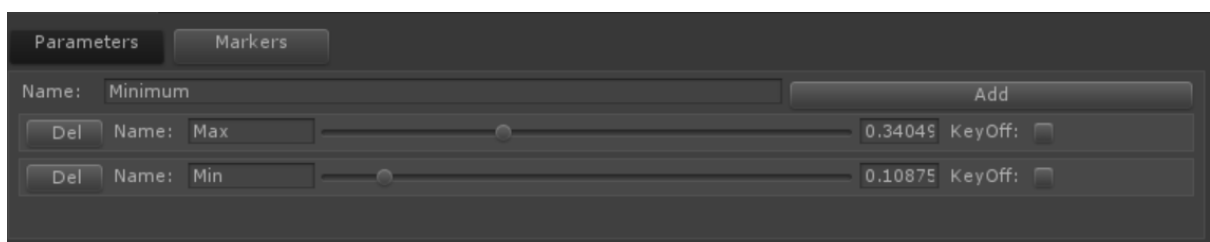

It is also possible to add/delete or copy/paste RTP parameters between components

Markers support
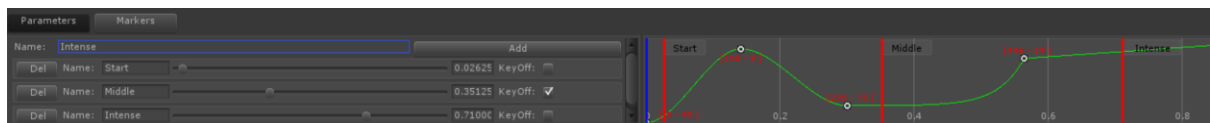
In this version it is now possible to add markers on Runtime Parameters and timelines making it possible to set a parameter using the marker name.

To add a marker you enter a text and click on the add button. It is also possible to delete, rename, set the position of the marker (normalised for now) and enable its KeyOff support. When a maker is set in KeyOff mode it will stop the parameter cursor moving past (assuming it has speed) unless the EventAction.KeyOff is posted.

Fabric v2.2.0

Runtime Parameter types
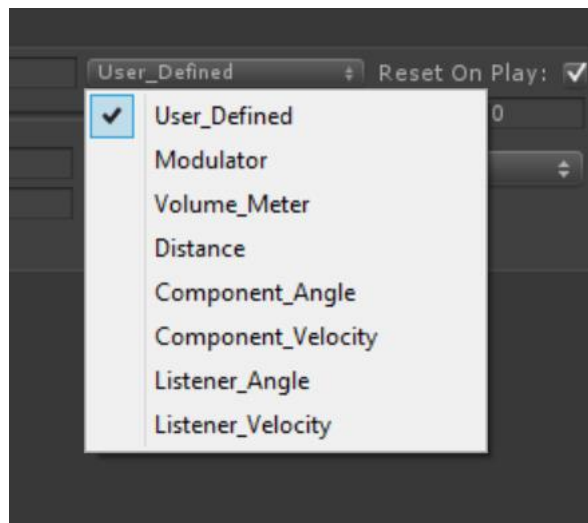
In this version of Fabric it is possible to control a parameter in different ways:

- User_Defined: Parameter has a name that the game can use to set a value
- Modulator: Parameter can be modulated using the new RTPModulator effect
- Volume_Meter: Any volume meter can link its RMS output with the RTP
- Distance: The distance between the listener and parent game object
- Component_Angle: The angle between the component and the listener or camera
- Component_Velocity: The velocity of the event game object
- Listener_Angle: The angle between listener and the component. Ideal for create simple culling processes that focus the audio to what the camera sees.
- Listener_Velocity: The velocity of the listener

Fabric v2.2.0
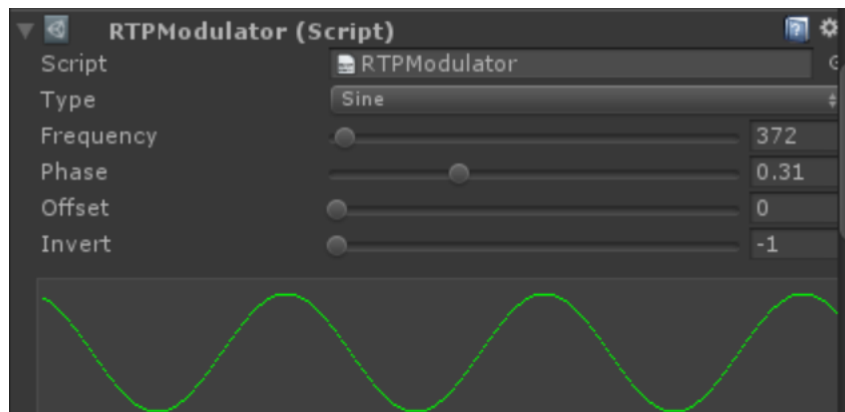
RTM Modulators

The Runtime Parameter modulator is a basic signal generator that can be used to drive runtime parameters. There are a number of options and type of signals available, such as:

Type: The types of different signals	Sine, Square, Sawtooth, Triangle
Frequency: The signal frequency	[0-20000]
Phase: Shifts the phase of the signal	[0 – 1]
Offset: Offsets the signal amplitude	[0 – 1]
Invert: Inverts the signal	[0 – 1]

Fabric v2.2.0

Languages

It is possible to provide localization for hundreds of dialog files by creating languages that are used by the Dialog Audio Component. The component will automatically add at the start of the audio clip reference name the currently active language folder and append at the end the language prefix.
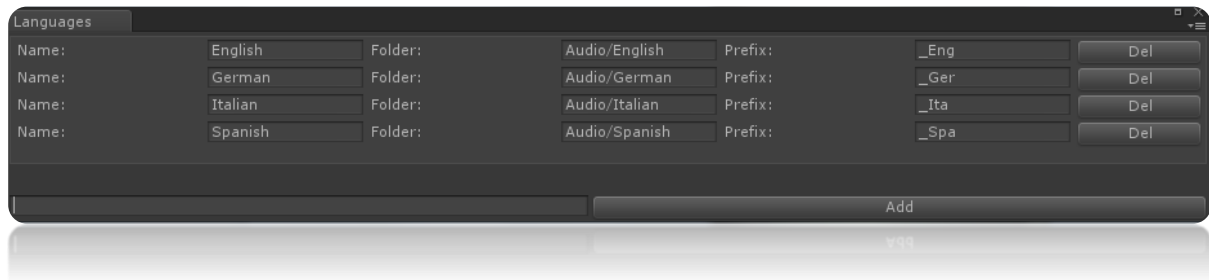
The language currently active can be set in the Fabric manager or by code.

Fabric v2.2.0

Graph View

Provides a flat view of the whole component hierarchy with runtime information of their state.
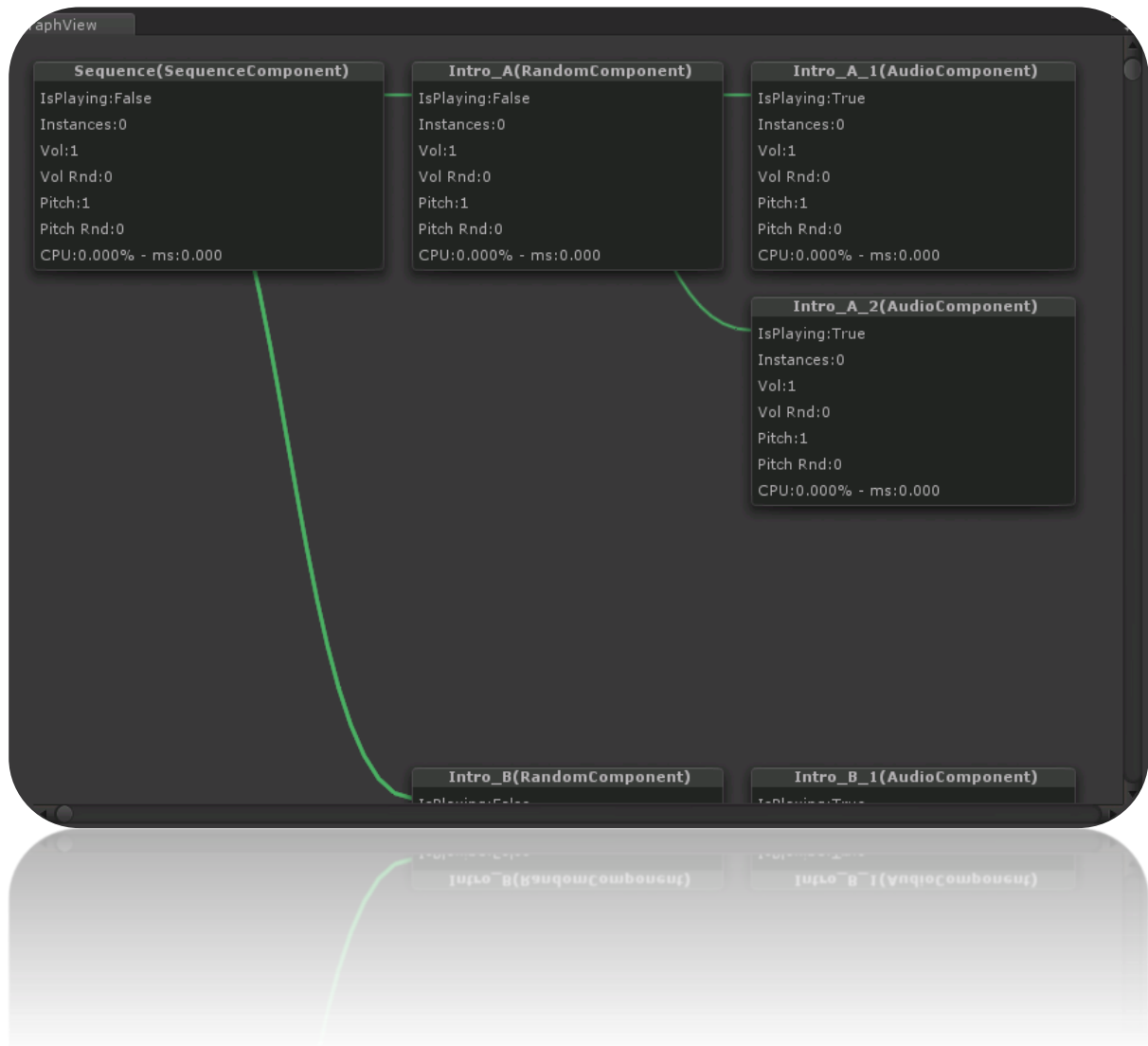
Event Log

Event log view allows monitoring the flow of events and identifying potential problems or missing events.

The event log view allows pausing or clearing the event log list, selecting to follow the current flow of events as well as filter event according to their description name.

The event filter can be used to select which event action types will be shown in the event log.

Event Monitor

The event monitor displays all the active events along with useful info such as: position, distance, volume, pitch, pa2D, instances, status.
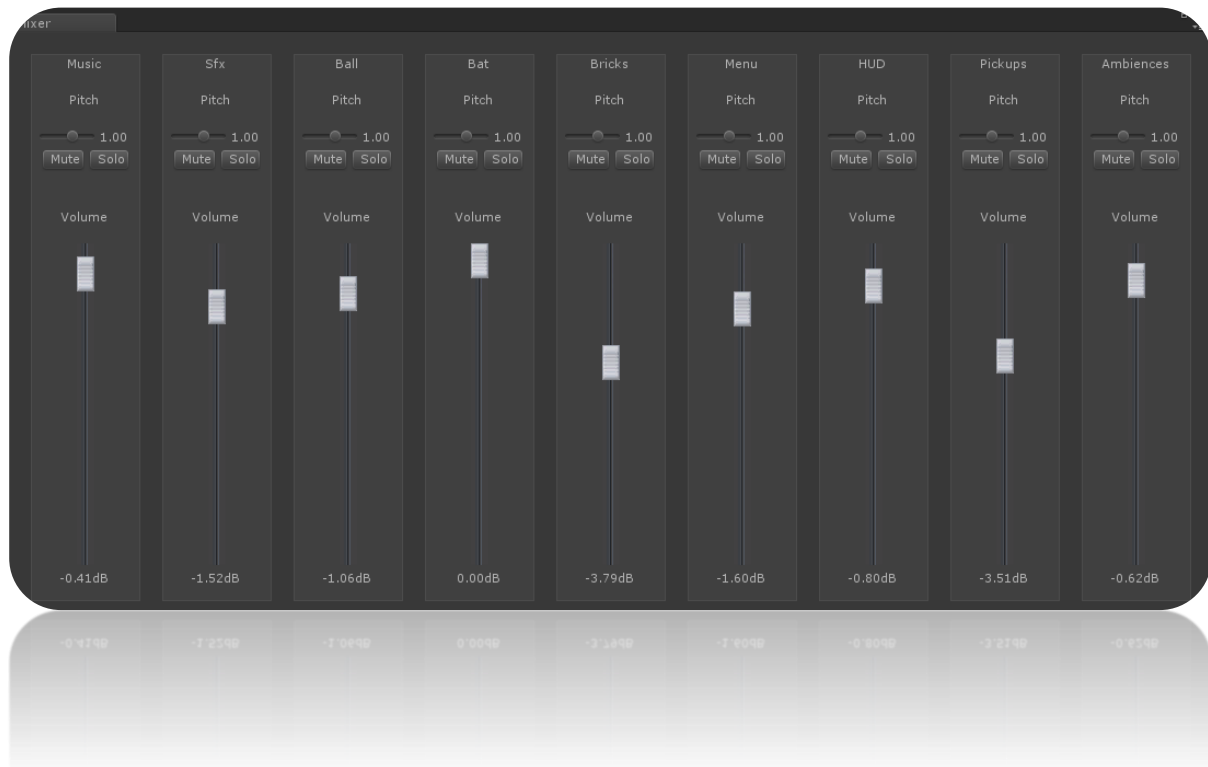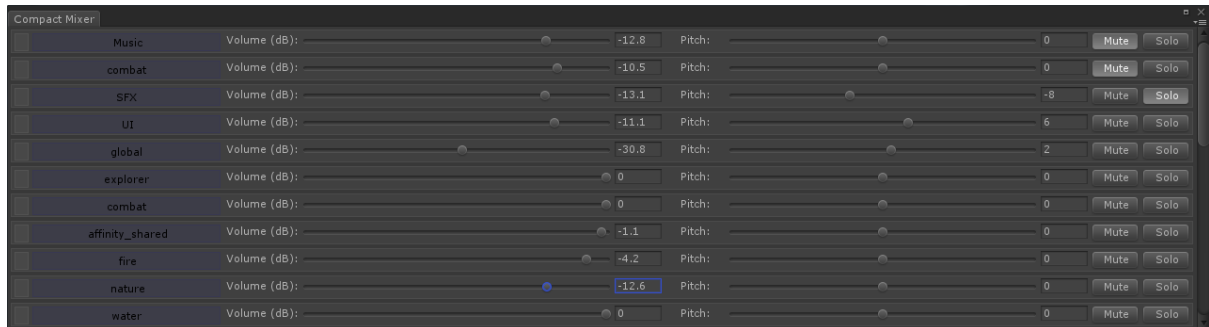
Mixer

The mixer window displays the group components in the hierarchy. From this view it is possible to change the volume or pitch properties of individual components.

The mixers Mute and Solo functionality allows to easily to mix and balance the audio levels in a game.

Fabric v2.2.0

Compact Mixer

A new compact mixer has been implemented making it much easier to manage a large number of group components.

The mixer provides the following functionality:

- A green icon indicating when a group component is active

- When clicking on the name it highlights the group component game object in the hierarchy view

- Volume is shown in dB and pitch in semitones

- Mute/Solo group components

Multi edit Audio
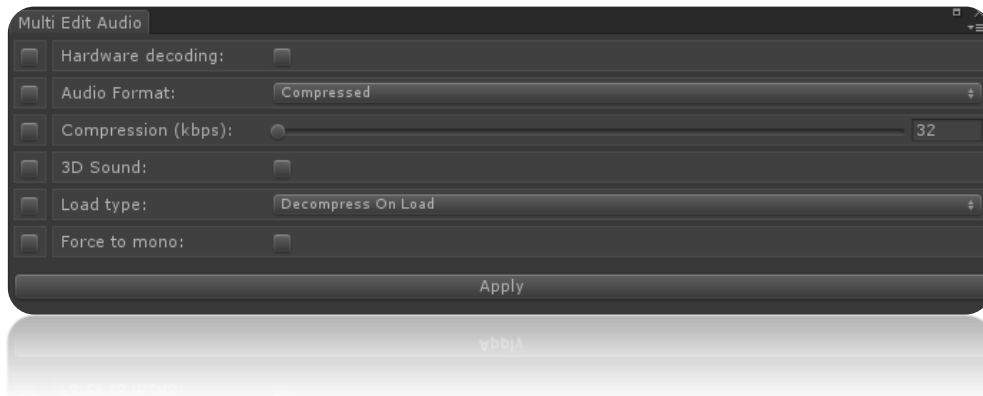
The multi edit audio window allows selecting any number of audio clips together and editing their properties collective with a touch of a button.

Previewer

The previewer allows to play Fabric components, as long as they have an event listener attached, in the editor.

Sessions can be created allowing to drag and drop components in their view area so that they can be previewed easier against other components without having to find them in the hierarchy.

Fabric v2.2.0
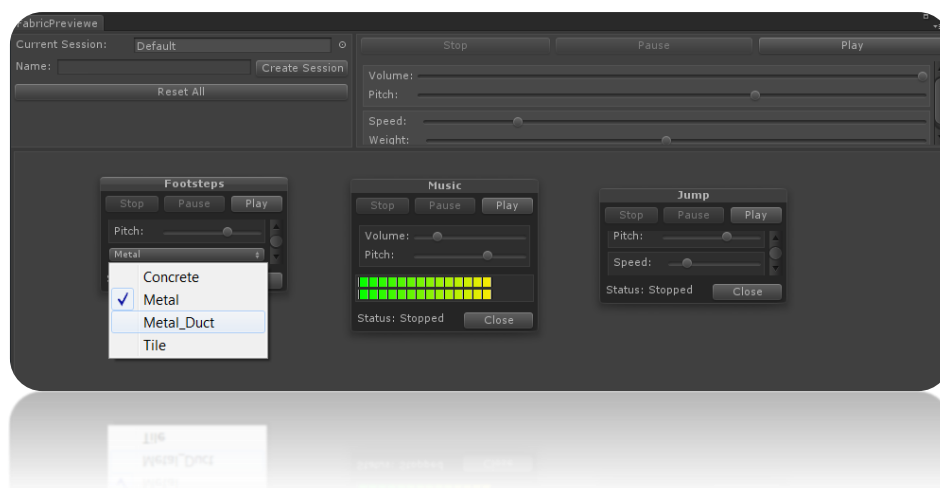
Furthermore each component will expose some basic properties (i.e. volume, pitch) as well as parameters that are specific to the selected component (i.e. switch, RTPC etc.)

Utils

Fabric provides a number of specialised utility scripts through the Window/Fabric/Utils menu option.

ConvertSamplesToSec

This utility script is used to convert all component properties that have been set as samples in projects created before Unity 4.1 version. Running this script once will be enough to convert the values to seconds.

Enable/Disable Dynamic Audio Clip Loading

Utility script that enables or disables all the audio components that are present underneath the selected game object in the Fabric hierarchy.

Copy/Paste Audio Clip References

Helper script that allows to copy all audio clip references from the audio components and paste them to another hierarchy that has a similar structure.

PrefixStringToDialogComponent

Helper script that opens a dialog box that allows to enter a string that will be added at the start of every dialog component name that is underneath the selected components.

Fabric v2.2.0

Fabric Extensions

Fabric comes with a number of extension that provide useful audio functionality.

Audio Spline Extension

This extension allows the creation of splines that have an event trigger attached that gets positioned at the nearest point on the spline to the listener.

An audio spline can be created from the menu option Fabric/Extensions/AudioSpline and it will be added underneath the currently selected game object in the hierarchy tab.

The spline already creates a game object that has an event trigger component that can be used to trigger a specific sound (i.e. river). This is the game object that will be continuously (and smoothly) be moved to the nearest point in the spline from the listener.
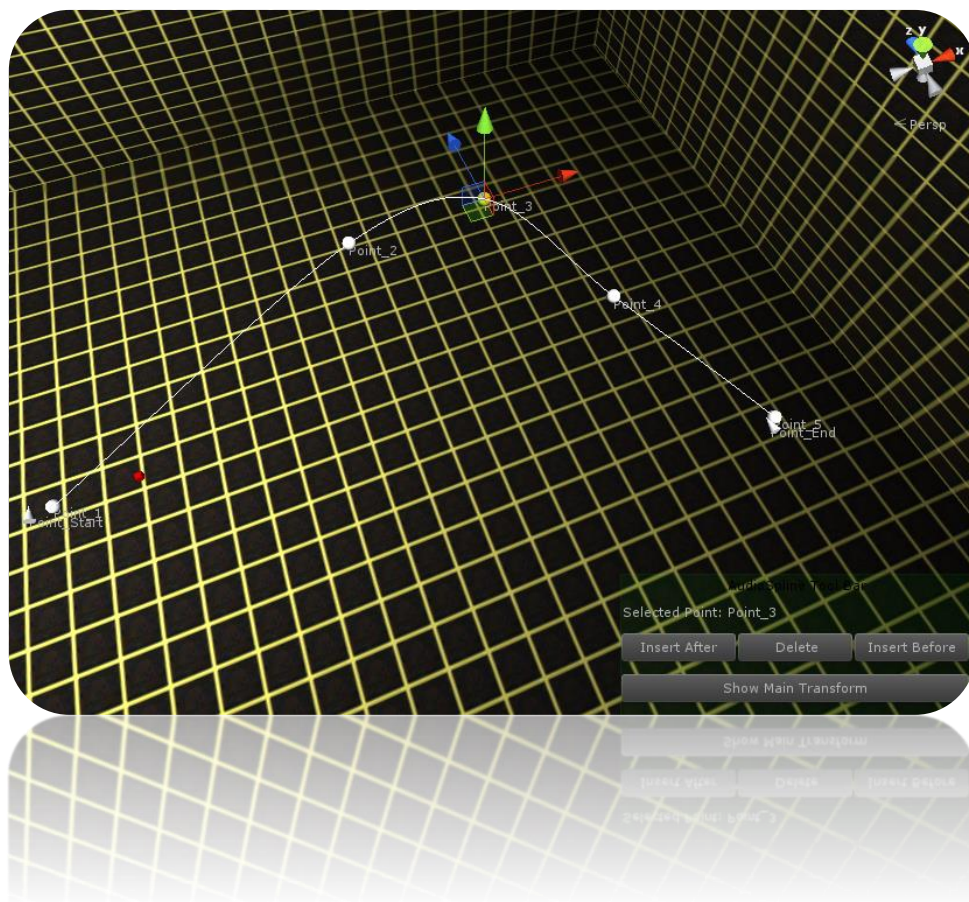
The AudioSpline provides UI controls in the Scene window that allows to add points on either side of the currently selected point or delete it. If the whole spline is required to be moved then the Show Main Transform button needs to be pressed.
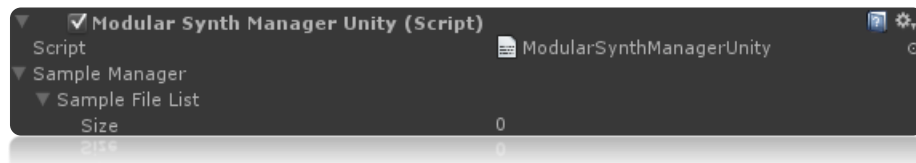
Fabric v2.2.0

Modular Synth Extension

Fabric provides a modular synth which can link audio modules together in order to create complex audio effects or generators.
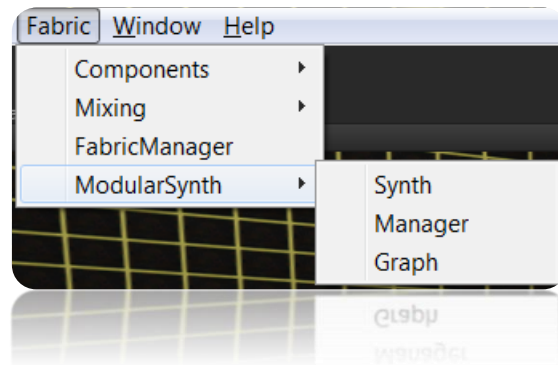
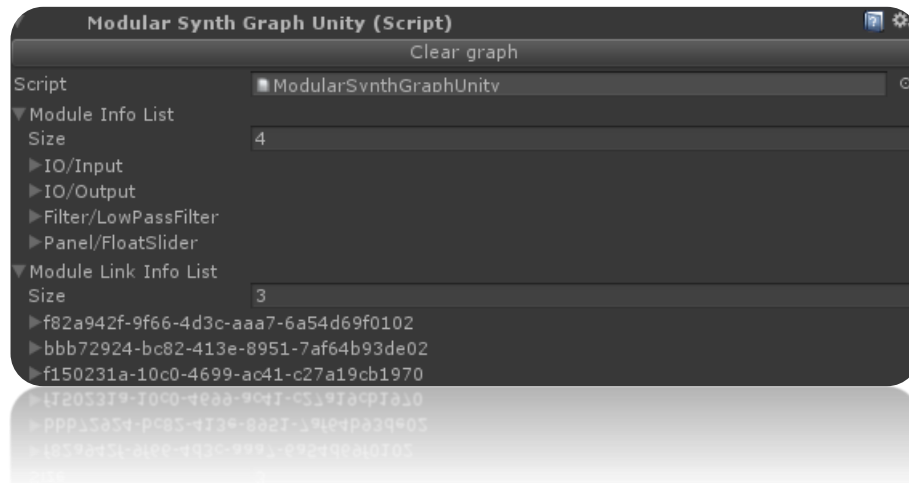There are three key modular synth components

Synth manager

The manager is responsible for managing all the modular synths and also the samples used by the sample player module.
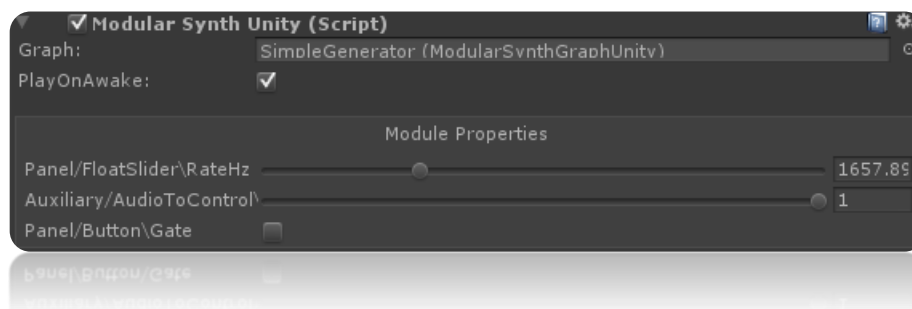
Synth Graph

The synth graph stores all the modules and their connections

Synth

The synth component can accept any modular synth graph for processing. It also displays any panel module properties.
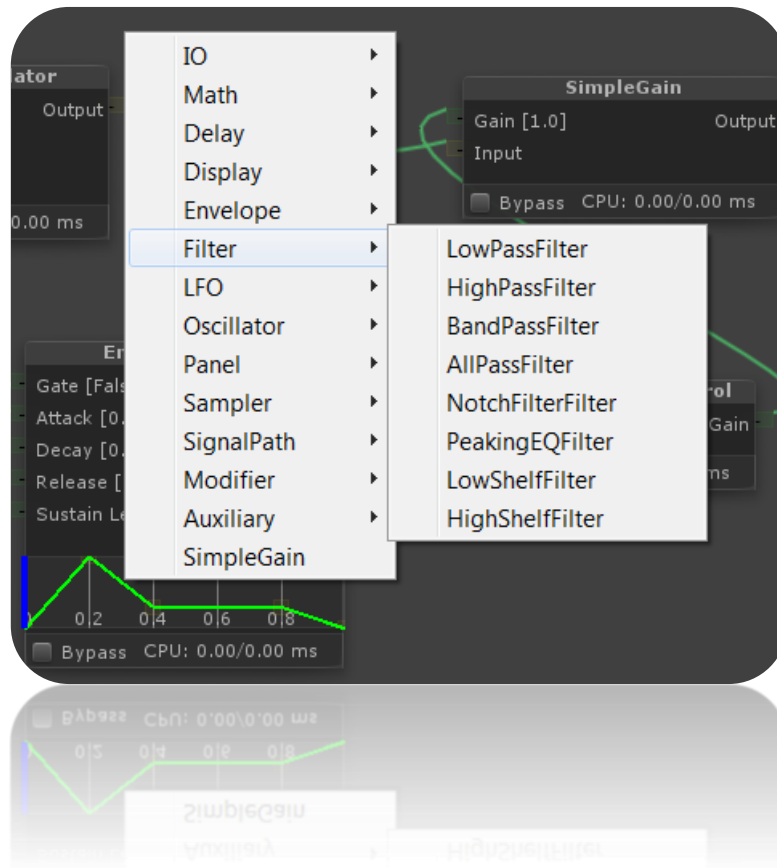

Fabric v2.2.0

Synth window

This is the main window that displays all the modules in a graph as well as allowing the addition of new ones, removing or linking them together

Adding modules to the graph

It is possible to add a new module by right clicking on the module window, this will show a drop down menu with the different modules available.

Fabric v2.2.0

Linking module pins

Modules can be linked together by selecting an output pin and holding the left mouse button down; this will show a green line that can be dragged around into an input pin of a similar type on another module.
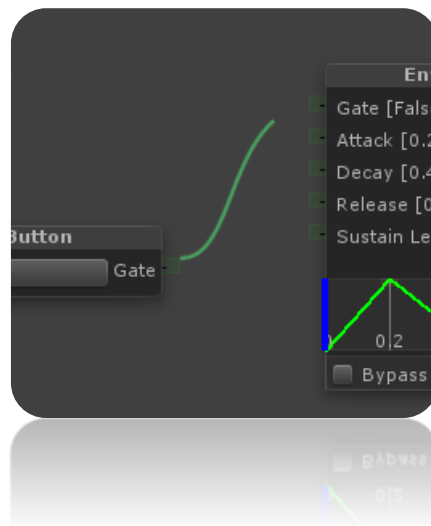
There are two different types of input/output pins:

Pins	Description
Audio input	Receives audio data from audio output pin
Audio output	Outputs audio data from module
Control input	Receives control input data (float, int, bool, string, list)
Control output	Outputs control data (float, int, bool, string, list)

Fabric v2.2.0

Deleting module pins

Links can be deleted by hovering on top of a link line and clicking on the right mouse button.
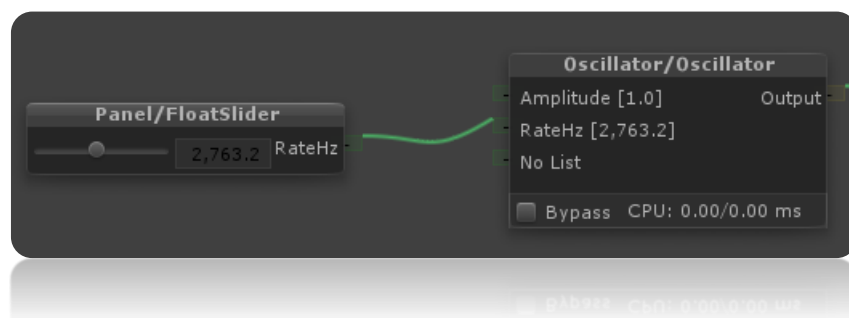

Modules

Modular synth provides two different types of module:

Audio processing modules

These are the processing modules that usually have input and/or output audio or control pins.

Panel modules:

These types of modules only have an output pin and are responsible for outputting parameter data such as float, integer, etc. They are automatically assigned a name and range when they are connected with a similar type input pin on another module.


Fabric v2.2.0

The list panel module by default is an empty list but when it is linked with an input pin it will automatically populate and provide the available options of that input pin. Below is an example of the panel list connected with the Oscillator input type.
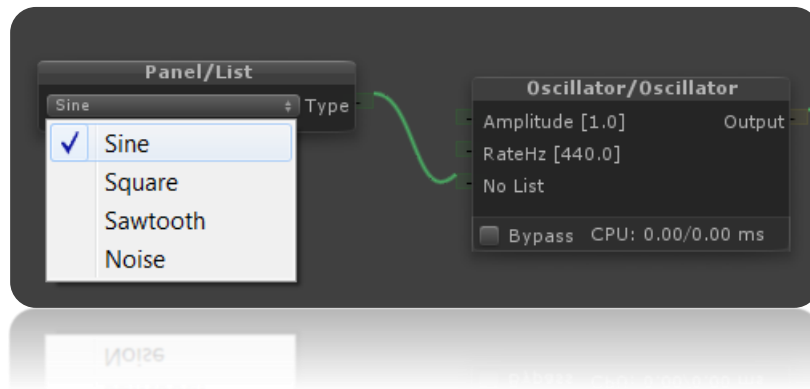
Sample manager

SamplerPlayer module imports audio clips into the modular synth.

NOTE: The sample player can load wavfiles with markers.

Fabric v2.2.0
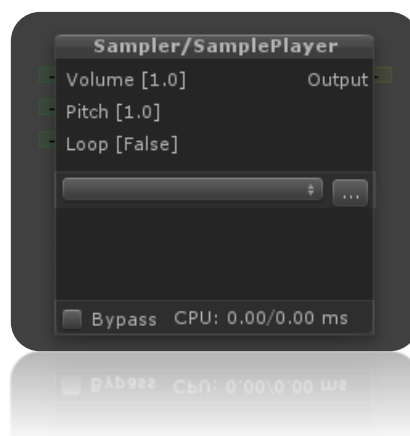
In order for a sample player to see the audio clips they have to be added first to the sample manager. By selecting on the "…" button the manager view window will be displayed allowing addition or removal of audio clips.

Once added to the SamplerManager, they will be available in all the SamplePlayer dropdown menu options.


Fabric v2.2.0
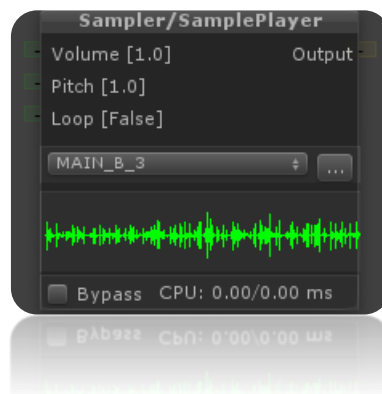
Quick walkthrough

This is a quick walkthrough explaining how to create and trigger a simple audio component in Fabric.

Setting up the scene

First we are going to start with a new empty Unity project in which we are going to place our audio assets, scene and Fabric hierarchy.

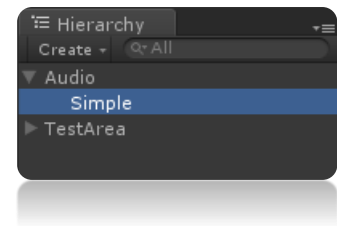## Creating the audio hierarchy

- Create a game object which is going to be the root of Fabric and give it a meaningful name (i.e. Audio, GameAudio etc.)
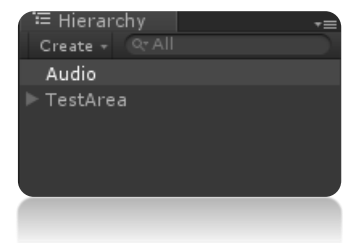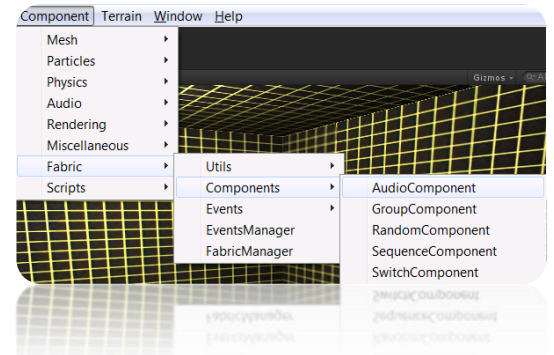

- Highlight the game object and from the Components->Fabric menu add FabricManager and EventManager components.
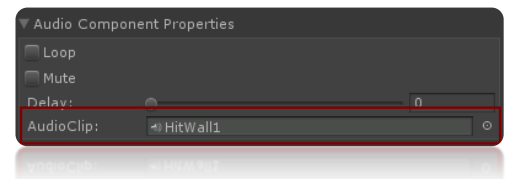
Fabric v2.2.0
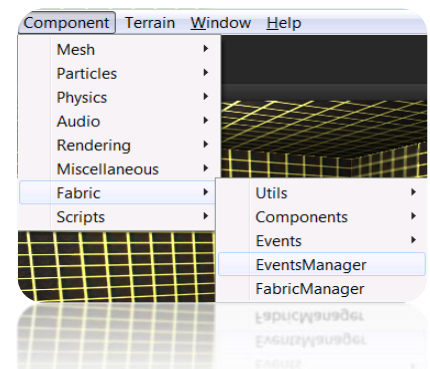
- Create another game object below the root and call it Simple.

- From the Components->Fabric->Components menu selection add the AudioComponent component.

- Next step is to assign the audio clip property of the audio, this could be done either by dragging and dropping the audio file in the AudioClip property or by clicking on it.

Creating the events

Each component needs to have a listener with a specific event name that it will listen to and respond accordingly.
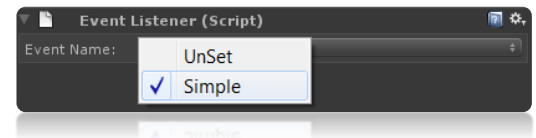
- In the EventManager enter the name of the event and click on the Add button to add it into the list of available events.
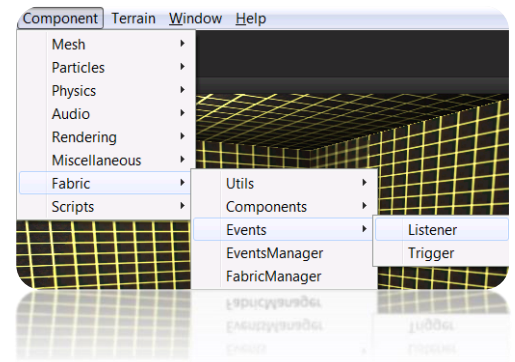

- Highlight the Simple game object and from the Components->Fabric->Events add the EventListener component.
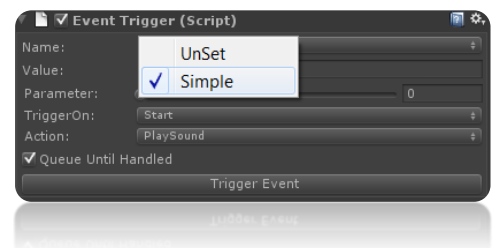

Fabric v2.2.0
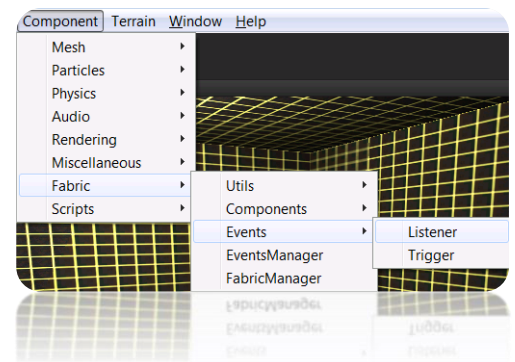
- Next step is to select the event name from the drop down menu.

- Now create another empty game object, outside of the audio hierarchy, and from the Components->Fabric->Events menu selection add an EventTrigger.

- In the event name drop down menu select the same event name as the listener.

- And that's it, when you run the game it will automatically trigger the audio component and you should be able to hear a sound. It is also possible to trigger the audio again by clicking on the trigger button.