

midi2event Unity instructions v2

Reach out to Cadence if you have questions/feature requests!

The Concept:

The midi2event system is a lightweight tool for charting C# events using a MIDI file. This allows you to trigger any code you would like in time with the chart/music.

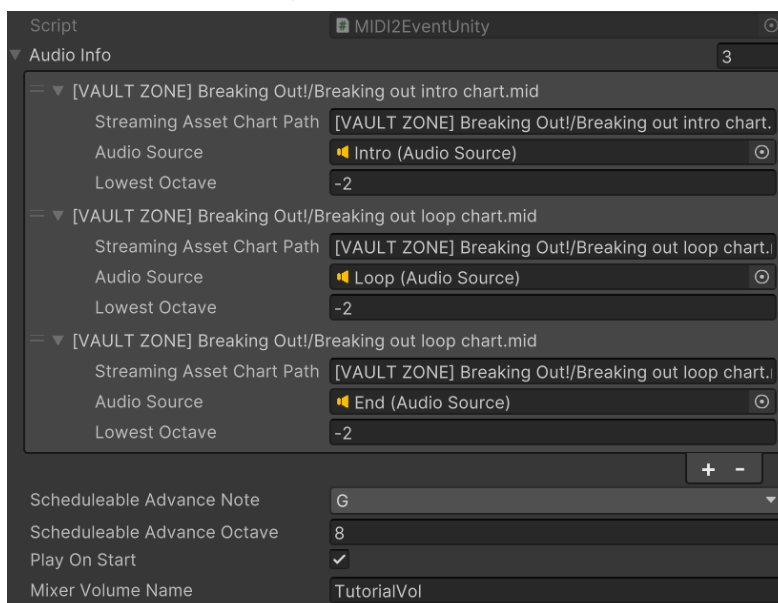
The Prefab:

The midi2event prefab contains a parent object containing the midi2event Unity script as well as several child objects containing audio sources.

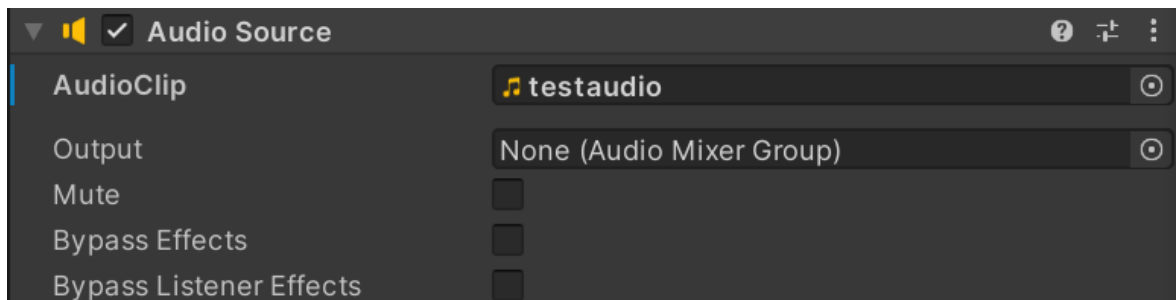


To set up the prefab, do the following:

- Drag the prefab into the scene hierarchy.
- Add/Remove elements from the
- Right click the midi chart you want to use in the Unity project directory (which must be contained within the StreamingAssets folder to build properly) , copy the path, and paste it into the Chart Path field of the MIDI2Event game object. Then, remove "Assets/StreamingAssets/" from the start.



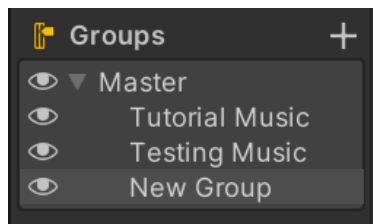
- If you have a note in looping sections that indicate where the loop should be exited, mark it appropriately in the Schedule Advance fields.
- The mixer volume name is important for scenes with multiple songs. More info on how to set this up in the
- Drag the audio asset you would like to use with the chart into the corresponding child's Audio Source's AudioClip field. If you want a certain part of the song to loop until some other event, enable the Loop checkbox.



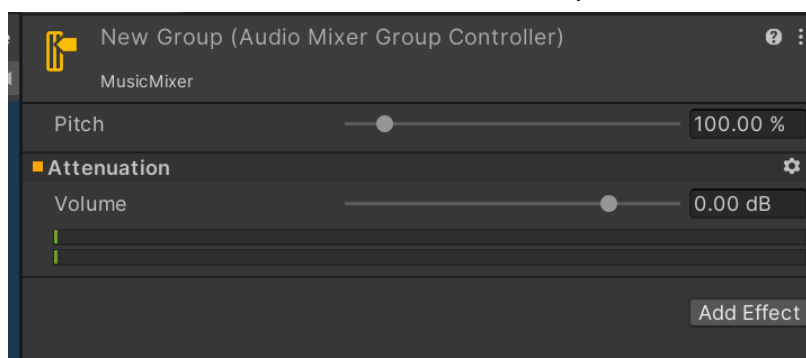
The AudioSystem:

The AudioSystem handles transitions between separate MIDI2Event systems within the same scene using an Audio Mixer Controller asset. To set it up:

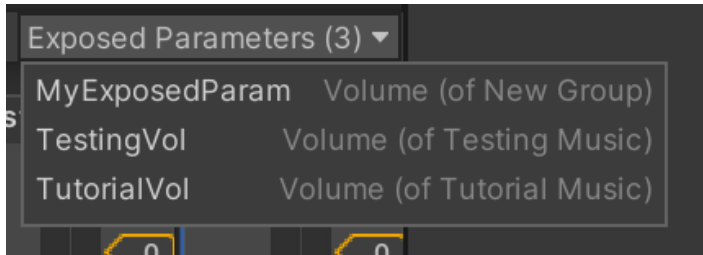
- Open the Music Mixer asset in the Audio Folder.
- Add a new group under master.



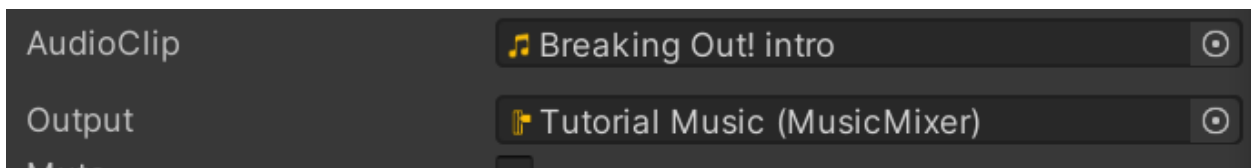
- Name the group, select it, and in the inspector, right click the word volume, then click "Expose Volume to script"



- In the Exposed Parameters dropdown of the Audio Mixer tab, set a name for your created parameter.



- Enter that name into the Mixer Volume name of the Corresponding MIDI2EventUnity object, and drag the mixer asset into each of its children's AudioSource.



Using midi2event:

Subscribing:

- To subscribe code to the midi2event system, you must use the Subscribe method. This method has one required and three optional parameters.
 - Action action - The code you want the midi2event system to trigger, in C# Action form.
 - Midi2Event.Notes note - The enum value corresponding to the note pitch you want to subscribe your action to.
 - int octave - The octave number containing the note you want to subscribe your action to.
 - Midi2Event.SubType type - What kind of event you want to subscribe your action to. Right now, you can subscribe to the start of a note, the end of a note, or the end of the chart.
- The Subscribe method returns an Action which you can invoke to unsubscribe the action from the system.

- You should subscribe in OnEnable and store the return values to unsubscribe in OnDisable.

```
Unity Message | 0 references
private void OnEnable()
{
    unsub = handler.Subscribe(Max, note, octave);
}

Unity Message | 0 references
private void OnDisable()
{
    unsub.Invoke();
}
```

Playing/Stopping:

- You can start playing a chart and its associated music by using the Play method.
- You can stop a chart and its associated music from playing, while resetting them both, by using the Stop method.
- Or, do both with Reset!

Known Issues/Quirks:

- Right now, the system only supports type 0 MIDI files that use ticks per quarter note. To my knowledge, this should be the most common format. You can read more [here](#).

Feature Requests:

- As this is a system I am hoping to release at some point, I would like to be the one who makes code modifications to it. If you would like to see a feature that does not already exist that relies on modifying the MIDI2Event class or lower, let me know and I can work on it!