Wwise (Audio Modding)

Instructions on how to use Wwise for audio modding in idStudio.

- **Audio Modding (Legacy)**
- **Creating Custom Music - idStudio**
- **Fixing Custom Audio - idStudio**

Audio Modding (Legacy)

Below is a link to the legacy Audio Modding chapter.
Details in this chapter are still relevant for Wwise/idStudio audio modding.

https://wiki.eternalmods.com/books/eternal-audio-modding

Creating Custom Music - idStudio

> Official idStudio guide for reference - **https://idstudio.idsoftware.com/audio/audio-file-modding-by-wwise**

> Note: Packaging custom music in your mod is currently bugged (as of the patch from May 19, 2026) and will disable vanilla music & SFX. Wait for a **future patch** or **follow this guide** before uploading a mod with custom music.

Requirements

- Windows 10/11 Operating System for Wwise. Linux may work, but it has not been tested and may require workarounds.
- Download the latest version of idStudio on Steam - **https://store.steampowered.com/app/2545650/DOOM_Eternal_idStudio/**
- Download the Audiokinetic launcher for Wwise - **https://www.audiokinetic.com/en/download/**
  - Create an audiokinetic account with a free trial. Free trial accounts do not expire and are sufficient for modding Doom Eternal's audio.
- From the Audiokinetic Launcher:
  - Wwise -> INSTALL A NEW VERSION -> ALL -> Major: 2019.2 -> Version: 2019.15.7667
  - Install Wwise (2019.2.15.7667) and accept default plugins (~100 MBytes)
  - After Wwise is installed, select the wrench icon next to it -> Set Environment Variables... -> Yes

Using Wwise

- Navigate to your DOOMEternal\idStudio folder in File Explorer:
  - ex: `C:\Program Files (x86)\Steam\steamapps\common\DOOMEternal\idStudio`
- Extract `wwise_mods_music_project.zip` to your WwiseProjects folder:

- ex: `C:\Users\USERNAME\Documents\WwiseProjects`

> **IMPORTANT**
> - Navigate to the "WwiseProjects\Wwise Music Mod\ModWavs" folder and open "test.wsources" with any text editor.
> - In line 2, change the value:
> Root="S:\ghost\game\base\sound\ghost"
> to:
> Root="."
> - The root value has to be a period "." so the PCK build script can run properly.

- Open the Audiokinetic Launcher -> Wwise -> Launch Wwise (2019.2.15.7667)
- Project Launcher: Open Other... -> Navigate to: `Wwise Music Mod\ModsWwise\music_mods.wproj` -> Open
  - In the future, you can find `ghost_mods` in the Audiokinetic Launcher and select: "Open in Wwise 2019.2.15.7667"
  - The project also be listed as `music_mods` if you Launch Wwise (2019.2.15.7667) from the Wwise tab.

Modding Music Flow in Wwise

> Make sure you are using the "Interactive Music" layout to see everything you need (default keybind: F10)

Getting Accustomed to Wwise

On the top left panel, you will see the **Project Explorer** and **Audio** tab. This is where you will be viewing the Interactive Music Hierarchy and selecting its "children":

- **Work Unit**
  - Default Work Unit -> music_ghost
- **Music Switch Container**
  - music_ghost_switch -> `mod1_music, mod2_music`
- **Music Playlist Container**
  - `ambient, heavy, light`
- **Music Segment** (control over individual music files)
  - `mod1_ambient_01, mod1_ambient_02, etc`

- **Music Track** (the actual music files themselves)
  - `mod1_ambient_general_01, mod1_ambient_general_02, etc`

- Note that the `mod1_stinger` and `mod2_stinger` **Music Playlist Containers** are present in the `mod1_music` & `mod2_music` **Music Switch Container**. They are not necessary, but can be used as additional music states. In this guide, they will be ignored. You can delete the **Music Segment** under them if they will not be used.

On the left-center panel, you will see the **Contents Editor**.

- This panel can be ignored.

On the bottom-left panel, you will see the **Transport Control**.

- This is where you will play, pause, and stop music. The space-bar key will start/pause music. If music is currently paused in Wwise, the "square" stop button will light up. You need to select it before playing different music, otherwise space-bar will resume the current music.

On the center-top panel, you will see the **Property Editor**.

- **Music Playlist Container** - This is where music volume is controlled under the **General Settings** tab. It is recommended to change the volume of each **Music Playlist Container** so all ambient, light, & heavy music can share the same volume, but this can be done at higher or lower levels in the Interactive Music Hierarchy.

- **Music Switch Container** - This is where transition rules are created under the **Transitions** tab. Under `mod1_music` & `mod2_music`, heavy or light outro **Music Segments** can be added here.

On the center-bottom panel, you will see the **Association / Music Playlist / Music Segment Editors**, depending on what you have selected in the Project Explorer.

- **Music Switch Container - Association Editor**: This is where the **Music Playlist Container** is mapped to the appropriate music_ghost_state.

- **Music Playlist Container - Music Playlist Editor**: This is where the **Music Segments** flow. Drag and Drop individual **Music Segments** from the Project Explorer to add them.

- **Music Segment / Music Track - Music Segment Editor**: This is where you define the length, start position, and stop position of a **Music Track**. The green line is the start position and the red line is the end position. These define when the previous **Music Track** ends and when the new one starts. Hold CTRL to move the green line without moving the start of the **Music Track**.

On the left-panel, you will see the **Master Audio Bus**.

- This will let you know how "quiet" or "loud" music is while playing.

Editing Music Flow in Wwise

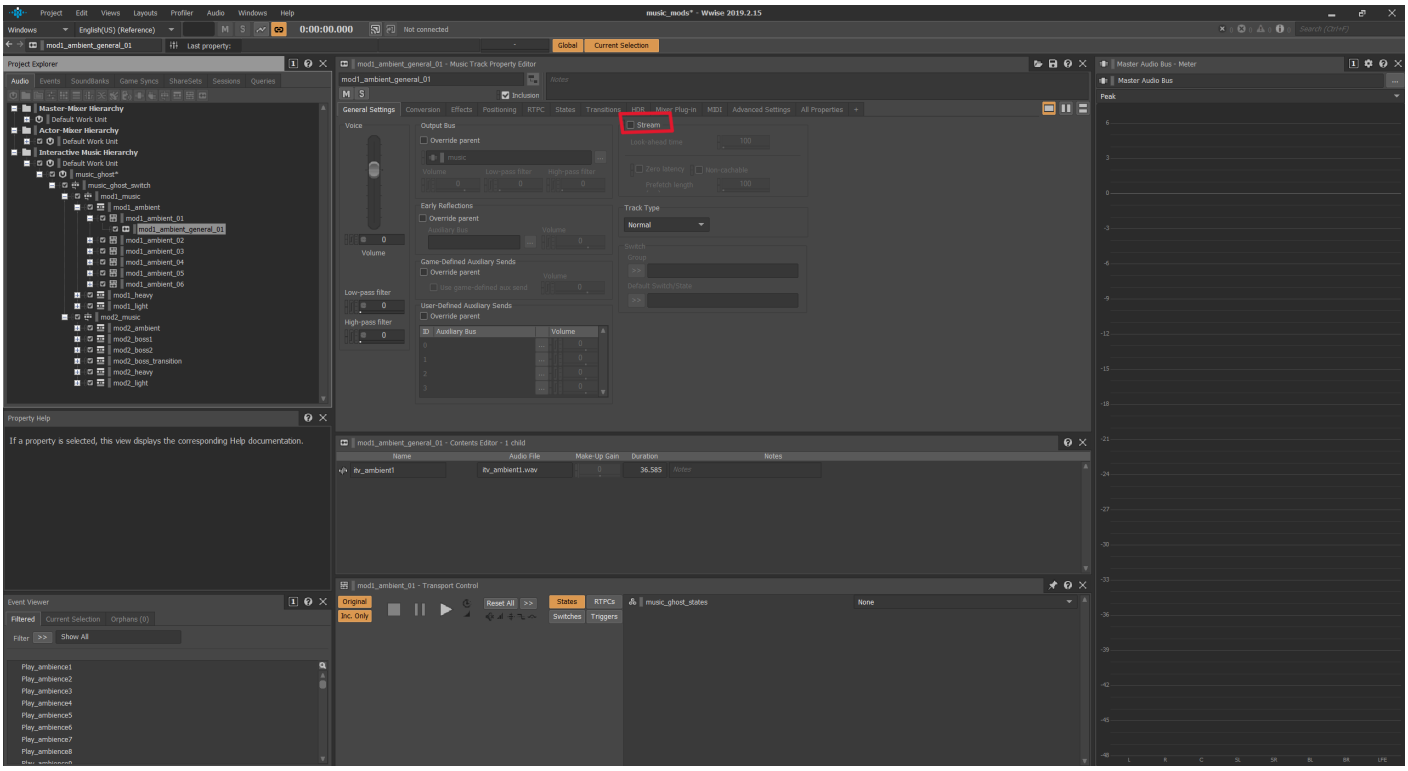
For a standard music mod, you will want control over which music tracks (stems) play in what order and when they should play. This guide will show you how to add and mix the following music states:

- **Ambient**
- **Heavy Combat**
  - **Heavy intro** -> when heavy music starts, these always play first and don't repeat
  - **Heavy outro** -> when transitioning from heavy to ambient music
- **Light Combat**
  - **Light intro** -> when light music starts, these always play first and don't repeat
  - **Light outro** -> when transitioning from light to ambient music
- Other music states are available and can be added if desired.

Adding Stems

- First, count how many ambient, heavy, & light stems (including intros/outros) you will be adding.
- Delete all **Music Tracks** under the existing **Music Segments**. Add the appropriate number of **Music Segments** under each **Music Playlist Container** by copy/pasting one of the existing **Music Segments**.
- Add a stem by right-clicking a **Music Segment** -> select: Import Audio Files... (Shift + I) -> Add Files... -> select WAV file -> Open -> Import
- Set the appropriate start and stop positions in the **Music Segment Editor**.

> If you plan on using "**mod2_music**", be sure to uncheck "**Stream**" under each individual Music Track in the **Property Editor - General Settings** tab so they won't be excluded from the **music_mods.pck** file. Music tracks under "mod1_music" already have this option unchecked.



Intro/Outro Stems

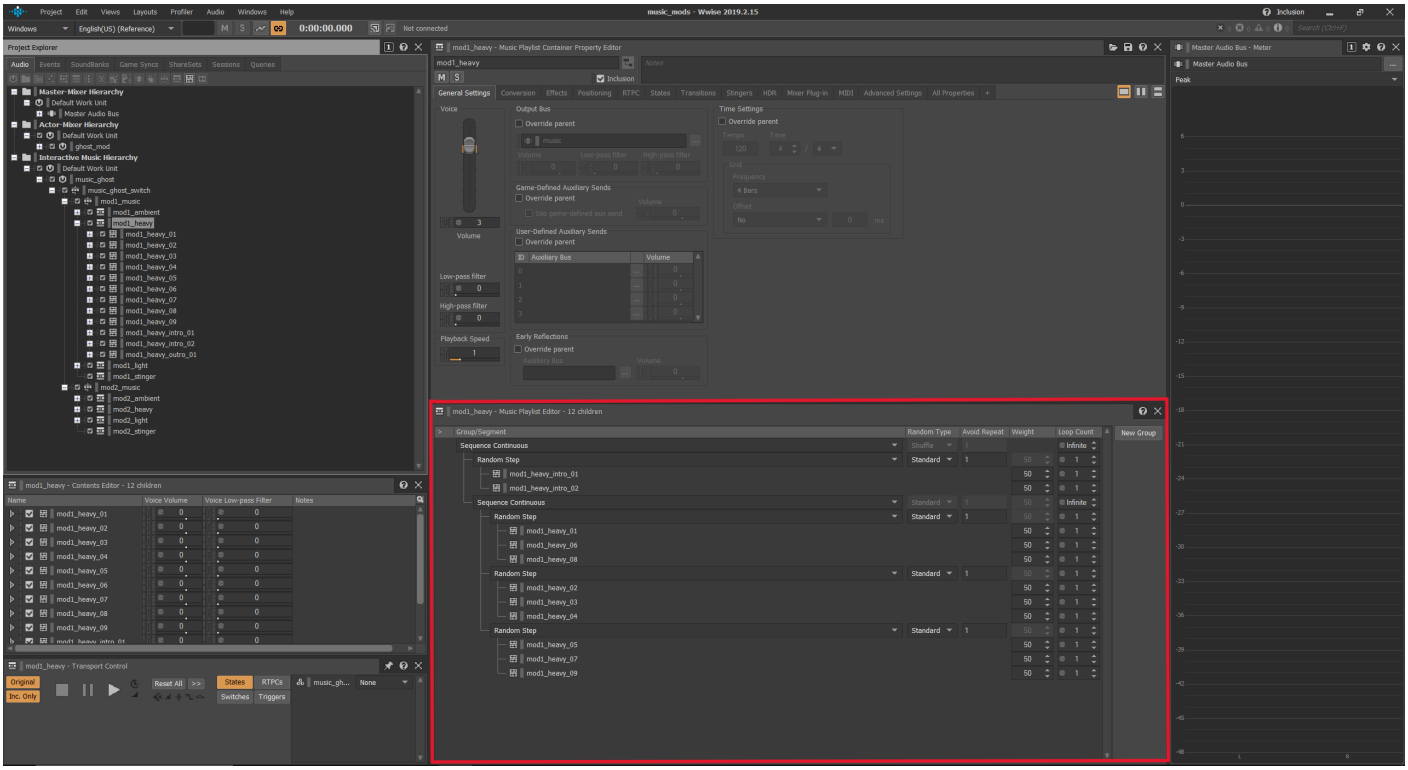
- Add intro stems the same way as normal.
- For outro stems, you need ONE **Music Segment** and ONE **Music Track**. If you have more than 1 outro stem (from the same **Music Playlist Container**) that you would like to play, then following to proceeding steps. Otherwise, just add the outro stem as normal.
  - In the **Music Segment Editor** -> right click the gray box below the Voice Volume field -> select: Random Step
  - This will add another Music Track field below the first. Add your second outro stem there. Both **Music Tracks** will share the same start & stop position, so be sure they are moved if needed. If you have 3 or more outro stems, add them here as well.
  - If you do not want each outro to share the same start & stop positions, then you can use the `mod1_stinger` or `mod2_stinger` **Music Playlist Containers** and add the outros as separate Music Segments.

Mixing Stems

- Navigate to a **Music Playlist Container** and view the **Music Playlist Editor**.
- Drag & drop all your **Music Segments** (except for the outros) to the **Music Playlist Editor** from their respected **Music Playlist Container**. Configure the mixing to your liking.
  - `mod1_ambient, mod1_heavy, mod1_light, etc`
- Below is an example of a heavy music flow state. There are 2 intro stems that play first and do not repeat. Only one is played and the other is ignored at random. Following the tree,

there are 3 groups with 3 tracks under them. Each group only plays one of the tracks at random then moves to the 2nd group. After the 3rd group plays a track, it will move back to group 1 and repeat.

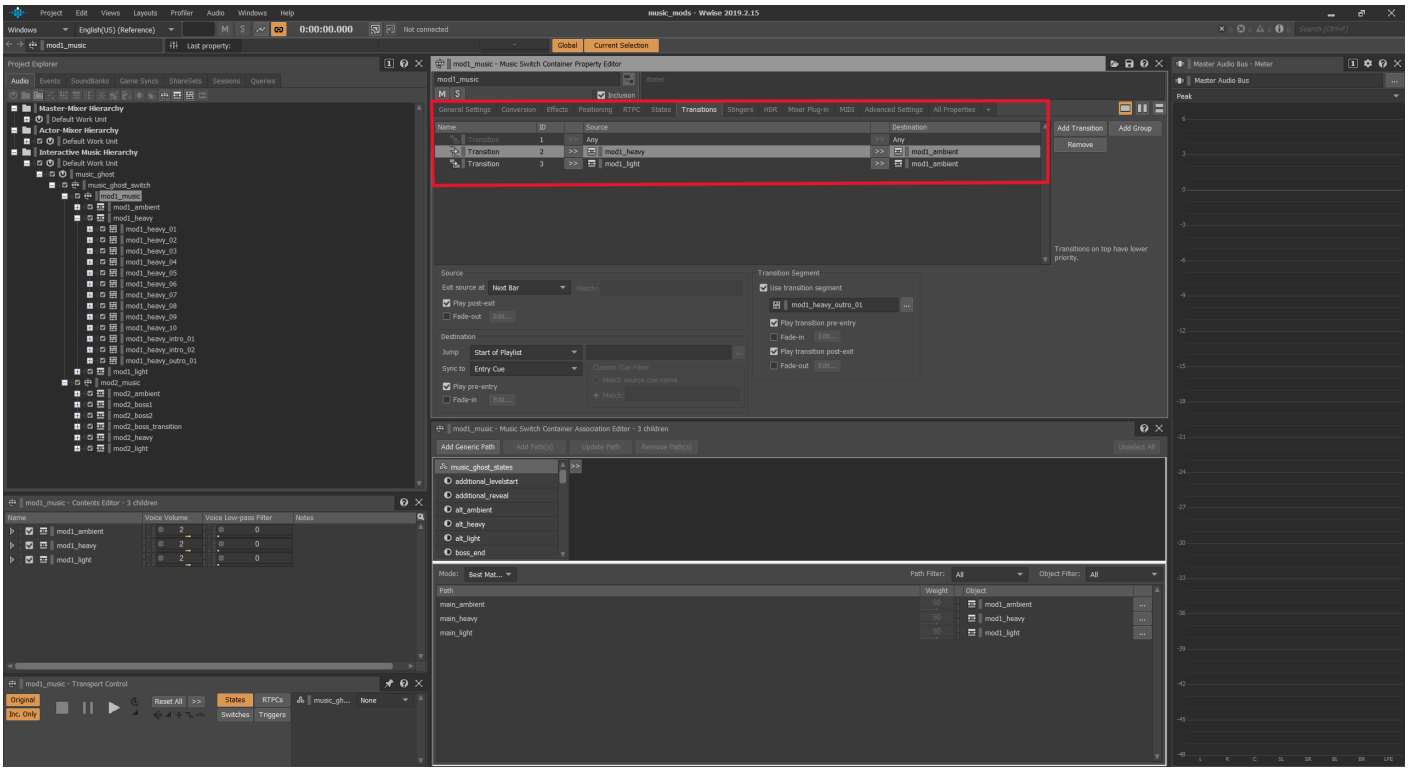
- If no intros are used, then don't add the additional **Sequence Continuous** -> **Random Step** tree.
- If you want ALL **Music Tracks** to flow randomly with no order, then just create the **Sequence Continuous** -> **Random Step** group and add all **Music Segments** in the one group.
- Press space-bar to start the music playlist and visualize how the **Music Segments** flow.



- To add outros, navigate to the `mod1_music` **Music Switch Container** -> **Property Editor** window -> **Transitions** tab -> select: Add Transition
  - Source: ">>" -> Browse... -> mod1_heavy
  - Destination: ">>" -> Browse... -> mod1_ambient
  - Source -> Exit source at "Next Bar"
  - Transition Segment -> check: Use transition segment -> ... -> select outro **Music Segment** -> OK

- Do the same for `mod1_light, mod2_heavy, & mod2_light` if applicable. If you did not add any outro stems, ignore this step.

- Select the grayed-out Transition at the top of the list. Uncheck any Fade-out and Fade-in boxes. Do the same for all transitions unless you explicitly want a fade between **Music Playlist Containers**.



Testing Custom Music

In Wwise

- Navigate to the `mod1_music` or `mod2_music` **Music Switch Container** -> Association Editor
  - Select: `main_ambient, main_heavy, or main_light` then press the space-bar to start playing one of them.
  - Select the other ambient, heavy, & light paths to listen their transitions. If you are satisfied, proceed to the next steps.
    - There may be some audio clipping while listening here, but it will not carry over when testing in idStudio & Sandbox.

- Select: Project -> Convert All Audio Files... -> verify that "Windows" is checked -> OK

In File Explorer

- Navigate to your WwiseProjects folder -> Wwise Music Mod -> ModsWwise -> Scripts
  - Path: `C:\Users\USERNAME\Documents\WwiseProjects\Wwise Music Mod\ModsWwise\Scripts`

- Run the `build.bat` script by either double clicking it or executing it in the Windows Command Prompt.

    - It is recommended to run with the Command Prompt so you can see the log output and verify if there were any errors. If there were errors, verify that:
        - Wwise version (2019.2.15.7667) is installed, environment variables are set, and the Wwise Project is opened with that version
        - In 2 line of the "WwiseProjects\Wwise Music Mod\ModWavs\test.wsources" file, Root="."

- Navigate to the Wwise Music Mod folder -> soundbanks -> pc
    - Path `C:\Users\USERNAME\Documents\WwiseProjects\Wwise Music Mod\soundbanks\pc`

- Verify that BOTH `mods.pck` and `music_mods.pck` were last modified when you ran the `build.bat` script.
    - `music_mods.pck` will be noticeably larger in file size since the converted WEM files are inside it.
    - Up one level in the soundbanks folder, you can find the `SoundbanksInfo.xml` file, which contains a list of all your converted music files. This file does not need to be packaged for your mod.

- Copy BOTH the `mods.pck` and `music_mods.pck` files into your idStudio mod's `\base\sound\soundbanks\pc` folder and replace them. Even if you did not add any custom SFX, both files are necessary.

- In idStudio, verify that **mod1_music** (and **mod2_music** if you added music to it) play in your map and if the transitions flow as intended.

Packaging Custom Music

Package the following in idStudio:

- `sound/soundbanks/pc/mods.pck`
- `sound/soundbanks/pc/music_mods.pck`
- `sound/soundbanks/pc/soundmetadata.bin`
- Any additional WEM files that you may have added, but are not in the `music_mods.pck` file.
    - Does not include the WEM files that were added when your first created your mod. Those can be deleted.

- All other relevant decl, image, model, and map files.

Fixing Custom Audio - idStudio

> For creating custom music, follow this guide before proceeding - **https://wiki.eternalmods.com/books/eternal-idstudio/page/creating-custom-music-idstudio**

> Note: Packaging custom music in your mod is currently bugged (as of the patch from May 19, 2026) and will disable vanilla music & SFX. Wait for a **future patch** or **follow this guide** before uploading a mod with custom music.

> Note: This is an advanced guide and assumes the viewer has adept knowledge of **Wwise** and **FusionTools**.

Context (What's Broken?)

After creating custom music and/or SFX, you will notice that all vanilla music & TAG2/Horde SFX are silent. Vanilla music is broken due to ID conflicts between the `mus.pck` & `music_mods.pck` files. TAG2/Horde SFX is broken due to a missing **containerMask** in the `mods.pck` file.

Fixing SFX

- Download the fixed `soundmetadata.bin` file modified by FlavorfulGecko5 - **https://drive.google.com/file/d/13UzqXSEA9_NWH8LAXVhIlWnnMYMixApy**
- Move the fixed `soundmetadata.bin` file in your mod's `\base\sound\soundbanks\pc` folder and overwrite the existing one
  - Creating a new mod in idStudio will generate a new (broken) `\base\sound\soundbanks\pc\soundmetadata.bin` file in case you need it

All vanilla SFX are now fixed. Some SFX will still be muted in idStudio, but will play without error when packaged and tested in the Sandbox.

Fixing Music

- Download **FusionTools** version 2.0.1 or later -

  https://www.nexusmods.com/site/mods/1812
  - May prompt to install Microsoft .NET
- Download the fixed **MusicSwitchContainer** HIRC object modified by Velser -

  https://drive.google.com/file/d/1Shuo0EaI16S6stjp8HEz5tKv2Ha4Gb6l
- Create a backup copy of your `\base\sound\soundbanks\pc\music_mods.pck` file in case you encounter issues
- Launch **FusionTools.exe** -> File -> Open -> open the `music_mods.pck` file
- Find the HIRC object with the ID **483143099**
  - You can quickly search for the ID in the **Filter...** text input box
- Right click the **MusicSwitchContainer [1CCC2DBB]** HIRC object -> Import HIRC from XML -> open the `483143099.xml` fix so it will overwrite the object
- Go to File -> Save As -> Save and overwrite the `\base\sound\soundbanks\pc\music_mods.pck` file

All vanilla music are now fixed. Some menu music will be muted in **idStudio**, but will play without error when packaged and tested in the **Sandbox**.

Fixing TAG Menu Music

> There are some additional ID conflicts for menu music.
> This step may not be necessary, but is helpful for cleanup.

- In **FusionTools**, right click the parent **HIRC** tree -> Export HIRC to XML -> save anywhere
- Open the `HIRC.xml` file with a raw text editor
- Scroll to the bottom of the file and delete the following fields

```
<EventAction ID="178021616" Name="178021616" Type="3">
    <field Name="ActionType" Type="UInt16">1027</field>
    <field Name="ObjectID" Type="UInt32">483143099</field>
    <field Name="Unk00" Type="Byte">0</field>
    <List Name="Props" Type="Prop" />
    <List Name="Modifiers" Type="RangedModifier" />
    <Array Name="ActionData" Type="Byte">
        04 B6 0B 2C 59
```

```
        </Array>
    </EventAction>
    <Event ID="1699343283" Name="Play_music_menu" Type="4">
        <List Name="Actions" Type="UInt32">
            <field Type="UInt32">178021616</field>
        </List>
    </Event>
```

- Save, return to **FusionTools**, right click the parent **HIRC** tree -> Import HIRC to XML -> open the **HIRC.xml** file

These HIRC objects were deleted, they had ID conflicts with the `mus.pck` file and were missing some data.

  - EventAction [0A9C64F0]
  - Event - Play_music_menu [6549EBB3]

TAG menu music may still be muted in idStudio, but will now work in the Sandbox.