

Wwise engine description and best practices:

Wwise version 2017.2.5.6619

Document version 1.1 @ 2019-06-23

Credits for information of wwise in relation to WoT and recommended settings go to Budyx69

Disclaimer: This is compiled information and is accurate to the best of our knowledge. It should not be treated as golden.

Resources:

- [Wwise Tutorial 20 - Profiler Overview](#)
- [Wwise 101-15 Memory Optimization](#)
- [WWISE Tutorial: Streaming](#)
- [Streaming Manager Tips](#)
- [Streaming Manager Settings](#)
- [Optimizing Memory Pools](#)

Introduction: The purpose of this document

The purpose of this document is to give the reader a basic understanding of the wwise sound engine, in context to memory management. Memory management is important because memory in your computer is a finite resource, and must be allocated as efficiently as possible. Take this situations in WoT as an example:

- You are playing in a battle, and all of a sudden, your crew sounds stop working. Why?
- You try to load into a battle, or load the game, and it crashes to desktop, with the python log crying about wwise memory usage. What happened?
- (In context to Relhax Modpack), why was I told to use this “expand memory” dependency for this sound mod?

The WWISE engine

This all has to do with the memory management system in Wwise, and how it affects WoT, and your computer. The following illustration shows what is meant by “memory is a finite resource”:

Example drawing, not to scale



World of Tanks is allocated a specific amount of memory (in simple terms). From there, the game allocates a specified amount of its RAM to the wwise sound engine.

When we load custom soundbanks into the game, it causes the required amount of memory to increase. If the mod soundbanks require too much memory, at some point the engine will dump them from it's allocated memory, to preserve the stability of the engine and the game. At that point, some events stop playing. This can be seen in the python log, as the engine can't find sound events (Error code 15).

One solution to this is to increase the amount of memory that WoT allocates to wwise, however this is more complex then just raising the value. WoT itself only has so much memory to allocate to WWISE, and if you tell it to allocate more then it physically can, then the game will complain in the python log and crash. Not good. To avoid this, we must be able to find out how much memory we can allow WoT to allocate, and how much it actually needs. But how do we do that? Before we can, we must understand some properties of the sound banks, and how they can be adjusted in WoT

WWISE engine and sound bank properties

The wwise engine has what is called “pools” of memory. These pools are statically allocated on game start from engine_config.xml, and therefore can only hold a specific amount of sounds. There are 3 pools that we need to focus on:

- Default pool: This holds the sounds and events that are loaded when the game engine calls them at specific times. For example, battle gui sounds are loaded from the gui soundbank when you are loading into a battle. **This is the pool that needs to be increased most of the time.**
- Low engine pool: This is the pool allocated to the low-level engine processing pipeline. Here is where any “on the fly” dsp effects are applied before being sent to the audio device. For example, decompression, equalization, and other audio effects like reverb and pitch shift are applied here.
- IO pool: This pool is used for streaming and caching audio stream from disk. **This is the pool that sometimes need to be increased.** More on this later.

Sound banks can be dynamically loaded and unloaded into WoT (loading music for a battle, for example). WoT has 3 areas that allow developers to add soundbanks to be loaded. They are in different configuration areas and have different behaviors as to how and when they are loaded into the game:

- Engine_config.xml
 - sfx_loadonce: Loaded once in the battle
 - voice: Loaded once in the battle, in the garage, everytime you change the crew sounds in settings.
- Audio_mods.xml
 - loadBanks: **unknown when loaded, at least loaded on game start**

Sound banks can be played by two different styles: default and stream

- Default: The entire soundbank is loaded into ram. This will take some of the allocated default pool memory in the game, but will reduce latency for when the audio is played. This is great for sounds that should happen immediately and are small, like crew voices and gui battle sounds.
- Stream: The soundbank is loaded in reference only into the engine. This means that when sounds are needed, the engine knows on the disk where the file is, and get it to play when the engine requests it. This is great to save memory on the default pool, but can cause delays when the sound is requested. **This does not mean that the game lags**, it only means that the sound may play a few (40-400) milliseconds behind from when the event was triggered. Streaming is excellent for sounds that are large, and don’t need immediate event firing, like music, tracks, engines, guns, and other background atmosphere sounds.

“But Willster”, I hear you say, “sounds like gun and tracks are used a lot, shouldn’t they be put into the default pool?” Normally yes, but in cases like this, there is a concept called [caching](#). This means that the sound can be loaded into the IO pool, and if the game finds that it is (or predicts it will be) used a lot, then it will keep it in the memory pool allocated to the IO. Additionally, you can, from the wwise project builder, set the “zero latency” checkbox. This will cause a small amount of the file to be pre-cached into the IO pool, therefore while the engine is getting the next segment from disk, it can immediately play what currently is in the pool.

There is a way to tell in WoT if a sound bank has stream events and stream audio files with it. The rule of thumb is: if there is a filename matching the “.bnk”, called the “.pck”, then there are sounds that are streamed. The pck extension is for the wwise packed sound files.

WoT sound engine properties

The wwise sound engine, in context of memory, can be configured by specifying settings in the engine_config.xml file. These are the default memory manager settings from WoT:

```

<Wwise_adv_profile>
...
<memoryManager>
...
<defaultPool>>13</defaultPool>
...
<lowEnginePool>>10</lowEnginePool>
...
<streamingPool>>2</streamingPool>
...
<IOPoolSize>>4</IOPoolSize>
...
<IOGranularity>>16</IOGranularity>
...
</memoryManager>

<memoryLimit>>160</memoryLimit>

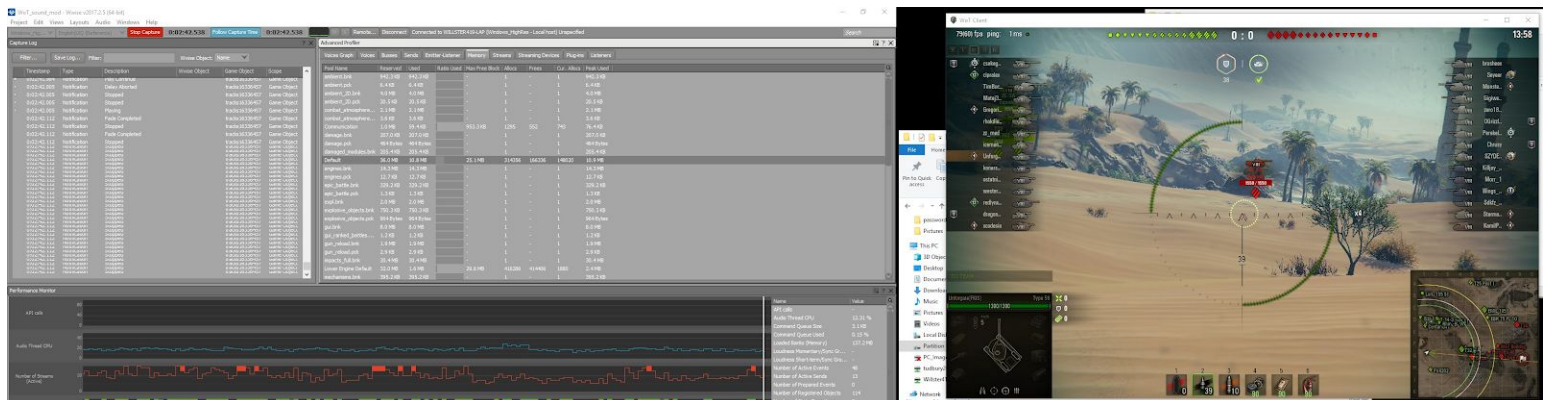
```

Here are the summarized definitions of what each setting is, and what Budyx69 uses for his sound mod.

- **memoryLimit** - This is the maximum amount of memory, in MB, that WoT is told that it *should* allocate to the wwise sound engine. There is more then what is specified here for memory allocated to the wwise engine. If this number is exceeded, then the game will complain in the form of spamming your python log. It can be treated as a warning, but that is all. Budyx69 recommends that the value is not set above 250MB.
- **defaultPool** - This is the amount of memory, in MB, given to the default pool for loading and playing sounds in RAM. Set this too low, and it will cause the engine to stop working during a battle. Set this too high, and the game will crash because it can't allocate enough memory to it. Budyx69 recommends the value of 36MB.
- **lowEnginePool** - This is the amount of memory, in MB, given to the low engine pool. If this is set too low, the wwise engine will completely crash for that battle. Budyx69 recommends the value of 16.
- **streamingPool** - This is the amount of memory, in MB, that the game allocates as it is streaming from disk. This is not the same as the IO pool. A value too low will cause the streaming sound to stop suddenly. This value should be changed when the mods being added are of high quality, or multiple sound files can be streamed at once. Budyx69 recommends the value of 24. It should be noted here that we may not need to increase the value to this much, as Budy has many sounds streaming from disk.
- **IOPoolSize** - This is the amount of memory, in MB, that the IO pool can use for caching. Same effects as defaultPool. Budyx69 uses a value of 8.
- **IOGranularity** - This is a value that is the standard request size, in KB when sent to IO for getting parts of the file to stream. Increasing this value will improve performance of the stream buffers, but there will then be less stream buffers available to get sound streams. This number generally does not need to be adjusted, especially if the sound mod uses lots of streaming sounds. Budyx69 does not change this value.

How to monitor in WoT the sound usage

The game allows you to be able to monitor the wwise sound engine usage. This allows you in real time to see how the game is managing the wwise sound engine, and how the engine is managing loading of the sounds:



There are two ways to do this:

XVM - Install and in sounds.xc, set the following to true:

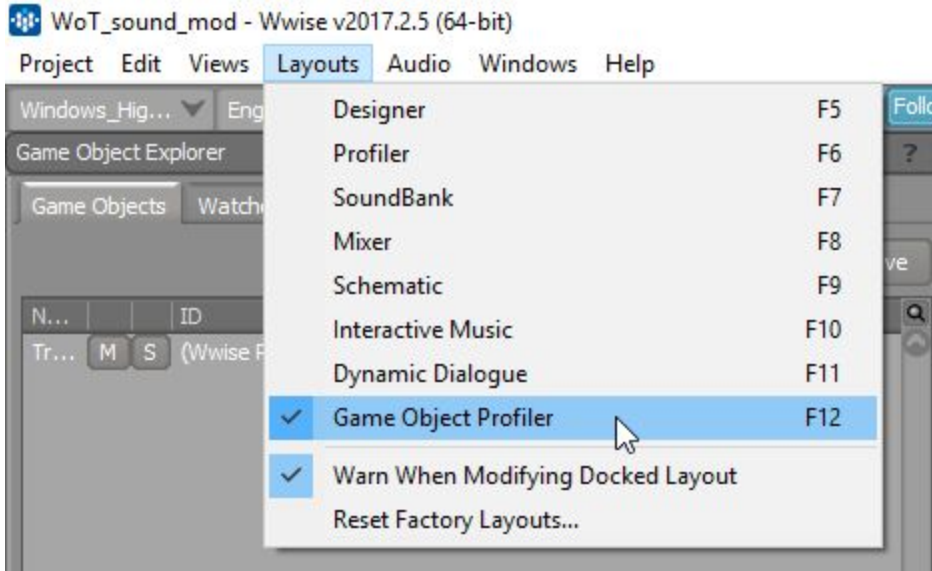
```
// true to enable remote communication between WoT client and WWISE Authoring Tools.  
// установите в true для включения коммуникации между клиентом WoT и WWISE Authoring Tools.  
"remote_communication": false,
```

Or, you can do it from engine_config.xml, in setting the value to true:

```
<communication>true</communication>
```

Open WoT with a replay.

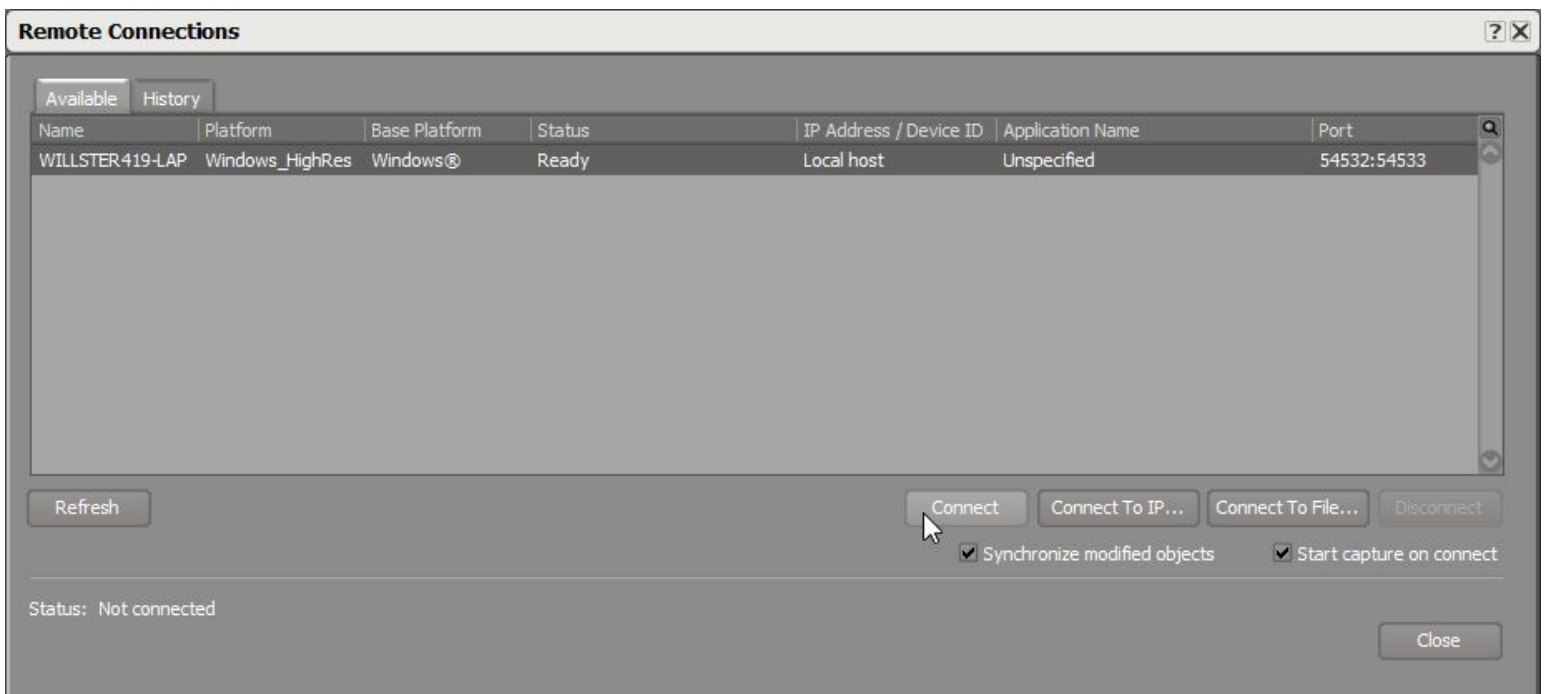
Then, assuming you have a wwise project already setup, on the menu bar, change the layout to “Game Object Profiler”:



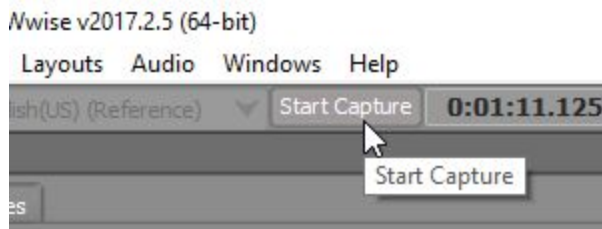
Then, click “remote”, to connect to the WoT client:



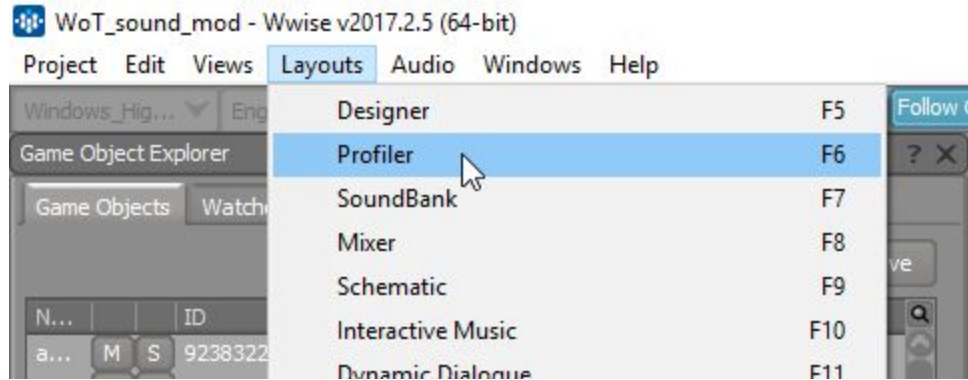
Select the WoT client and click connect:



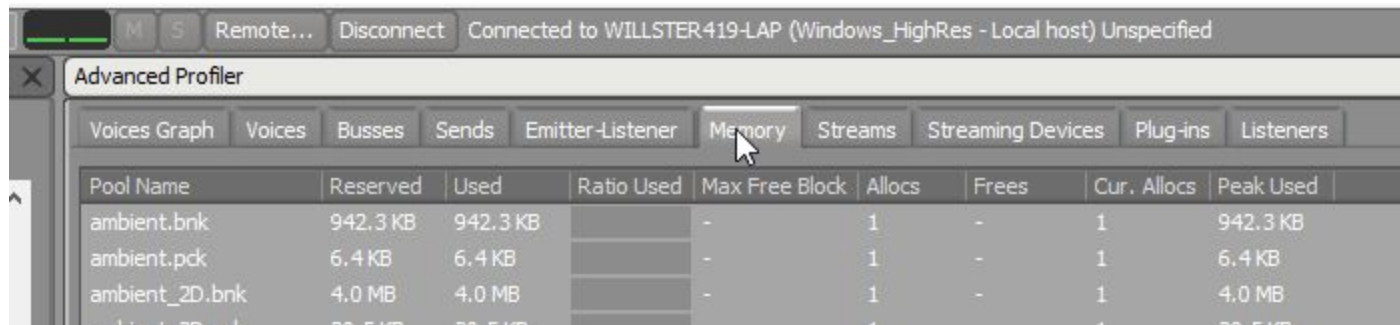
If capture does not start automatically, then press “start capture”:



At this point, you can monitor the memory usage by changing the layout to the profiler:



Click the memory tab, and you can start monitoring the memory management:



You will notice that it shows bnk and pck files. These show the reserved and used amount, and always being completely used. This can be ignored, as this is just the size of the files on the disk. The important pool names to monitor are:

Stream I/O	32.0 MB	32.0 MB	-	1	-	1	32.0 MB
Stream Manager	48.0 MB	17.5 KB	48.0 MB	7486	7293	193	36.9 KB
Default	36.0 MB	10.8 MB	25.1 MB	350927	202892	148035	10.9 MB
Lower Engine Default	32.0 MB	1.1 MB	29.6 MB	535335	533820	1515	2.4 MB

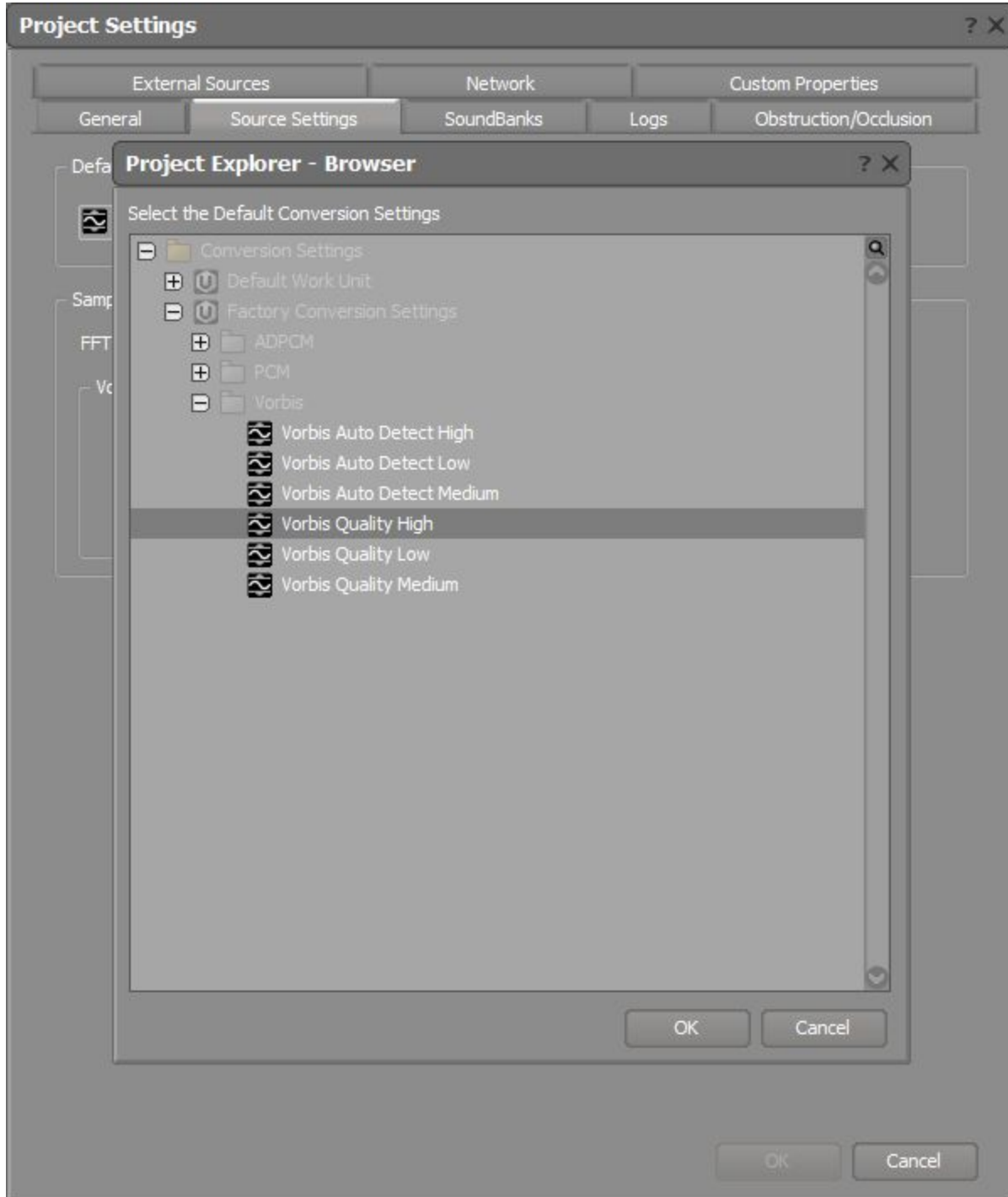
You should notice that the “reserved” values are exactly specified in engine_config, when this was loaded:

```
<memoryManager>
  <defaultPool>36</defaultPool>
  <lowEnginePool>32</lowEnginePool>
  <streamingPool>48</streamingPool>
  <IOPoolSize>32</IOPoolSize>
  <IOGranularity>64</IOGranularity>
```

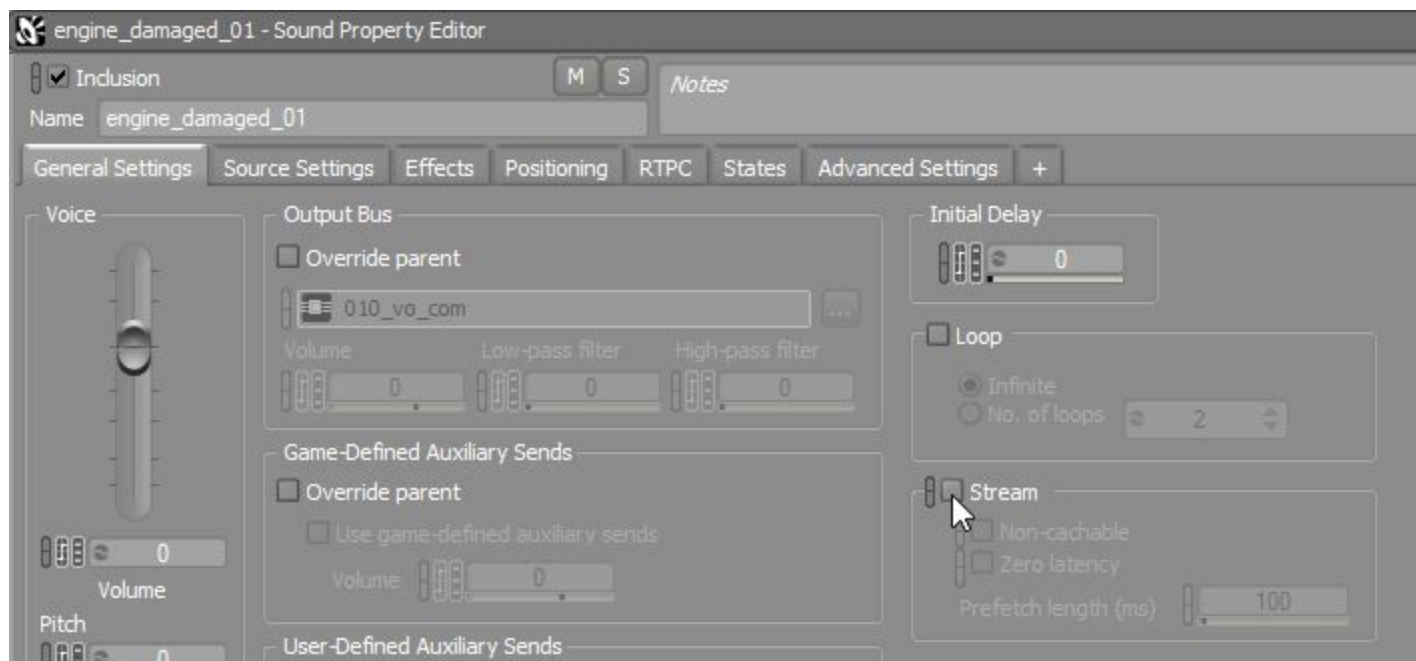
- defaultPool -> Default
- lowEnginePool -> Low Engine Default
- streamingPool -> Stream Manager
- IOPoolSize -> Stream I/O

Audio tips

Another way to reduce the amount of memory needed, as an audio developer, (besides allowing streaming), is to use a compression algorithm when encoding your sounds and building your project. When you add your files into wwise, they are uncompressed wav files. This is great for development, but a stereo channel sound uses about 1400kb/s for encoding ,and that can add up fast. You can tell wwise to compress your audio files into smaller sounds by using the **vorbis encoding** audio. This can be done by specifying the conversion settings in your wwise project. The files will be compressed significantly, (around 70-90% of their original size), and will sound almost exactly like their original versions. For a videogame, this is good enough in the audio quality department. From the project settings window, you can specify the default conversion settings:



Not every sound event/file in a bank needs to be loaded into memory. For example, if you have a sound that is long playing, large, and does not depend on latency, you can allow it to be streamed:



How to configure the audio system in WoT via Relhax patches

The values can be increased in the form of patches in Relhax modpack, by specifying that the patch will only change the value if the original is smaller then then the new value:

```
<patch>
..<type>xml</type>
..<mode>edit</mode>
..<patchPath>app</patchPath>
..<file>\res_mods\versiondir\engine_config.xml</file>
..<path>//engine_config.xml/soundMgr/WWISE_adv_profile/memoryManager/defaultPool[text()&lt;=36]</path>
..<search></search>
..<replace>36</replace>
</patch>
```

The *path* value will only return the defaultPool object to patch if: **original_value < 36**. This means that if the value was patched earlier to 48, then $48 < 36$ would be false, and the patch would not happen. This is to make sure it always takes the larger value of any patch applied to it.

Other notes relevant to Relhax Modpack

Gnomefather has high sound requirements and needs to fix this.

Since there are 4 values that should be increased, the following patches will be used:

Dependency_Sounds_Increase_Sound_Memory_default - Increase the sound memory limits, with emphasis placed on the default (RAM) pools.

Dependency_Sounds_Increase_Sound_Memory_larger_IO_stream - same RAM pool settings as above, but significantly increases the IO pool sizes. Useful for large mods, music, ambient, gun, engine.

You may notice that another value, **max_voices**, get increased. This is the number of sound elements that can be playing at the same time. This is increased a little as some mods add sound events.