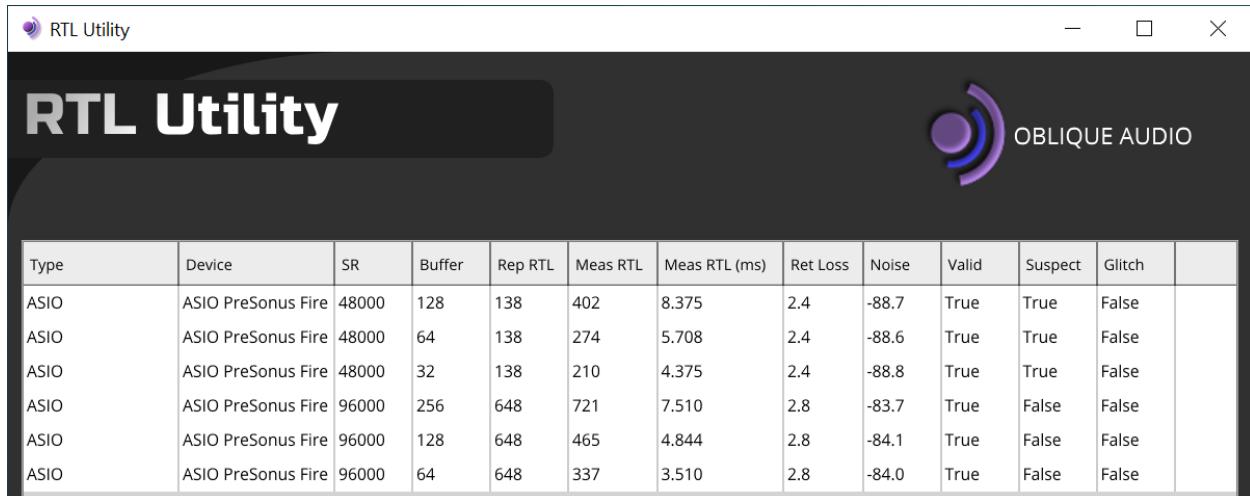# Introduction

Long story short: After tweaking, my roundtrip latency is down a lot

- Initially: A mediocre and just passable ~8.4ms, still with some tiny crackling every ~10 minutes
- Afterwards: An excellent ~3.5ms without any crackling/stuttering

Type	Device	SR	Buffer	Rep RTL	Meas RTL	Meas RTL (ms)	Ret Loss	Noise	Valid	Suspect	Glitch
ASIO	ASIO PreSonus Fire	48000	128	138	402	8.375	2.4	-88.7	True	True	False
ASIO	ASIO PreSonus Fire	48000	64	138	274	5.708	2.4	-88.6	True	True	False
ASIO	ASIO PreSonus Fire	48000	32	138	210	4.375	2.4	-88.8	True	True	False
ASIO	ASIO PreSonus Fire	96000	256	648	721	7.510	2.8	-83.7	True	False	False
ASIO	ASIO PreSonus Fire	96000	128	648	465	4.844	2.8	-84.1	True	False	False
ASIO	ASIO PreSonus Fire	96000	64	648	337	3.510	2.8	-84.0	True	False	False

This is on a soundcard that came out on the market in 2009 (Presonus Firestudio Mobile), i.e. not a new and very expensive RME or Antelope product. However, to get to this point I needed quite a bit of tweaking, so I want to share my process. I did a total of 5 steps:

- A workaround for what I consider an unfixable hardware bug in the Intel integrated graphics card (iGPU) is made in Step #4, and is the main contribution of this guide
- Step #3 is the time-consuming step, where you need to know a bit about your computer hardware. The pay-off is however massive

Step 1. Turn off CPU C-states greater than C1

Here are three alternatives on how to do this in my preferred order:

- In BIOS
- Using Throttlestop
- Using Bitsum ParkControl

Step 2. Debloat Windows

Follow the instructions available here. Personally I ran all options in the GUI script (Windows10DebloaterGUI.ps1):

- Remove all bloatware
- Disable Cortana
- Stop Edge PDF takeover
- OneDrive
- Disable Telemetry

- Unpin tiles from start menu
- Remove bloatware keys

Step 3. Overall low latency optimization

Although the target audience is low latency gamers in this great YouTube video, almost all applies for low latency audio as well. This will take some time, but was absolutely crucial in getting my final results.

2:18 to 4:27

Since we have already run the debloater script in step 2, then it is not necessary to run "Disable Unnecessary Services.cmd". All others are fine based on my own experience.

4:27 to 8:20

Affinity policy: On my 4 core/4 thread CPU (Intel 8350k) I made the following allocation:

- CPU0: Nothing dedicated
- CPU1: iGPU, Ethernet and Wireless
- CPU2: 1394 FireWire controller and the related PCI-to-PCI bus controller
    - Works great using a Texas Instrument XIO 2213 controller
- CPU3: USB3 xHCI

MSI Utility

- High priority: iGPU, 1394 FireWire controller and the related PCI-to-PCI bus controller, USB3 xHCI. I just left the SATA controller at high also
- Normal priority: Ethernet, Wireless, Built-in Realtek audio (High Definition Audio Controller)
- Low priority: Management Engine Interface

8:20 to 8:57

I did not apply the mouse registry optimizations, they would be completely unnecessary for audio.

8:57 to 9:20

I only added the two registry optimizations, and skipped the IRQ priority parts. I might re-visit the IRQ priority parts in the future.

fast forward to 14:54

14:54 to 15:56

Add the power plan as shown, and with that this step is done.

Step 4. Workaround for Intel iGPU hardware bug

I still had some glitches at this point, and it turned out that my Intel UHD 630 iGPU was the root cause. Hence this step is vital for modern Intel CPUs with an iGPU, see this post from the RME forums for more detailed information. When even RME interfaces have problems, you know that it is bad. I already had dual channel DDR4 RAM, so this was not the issue for me. Some more information here and here. Notes:

- Skip this if you have a dedicated Nvidia/AMD graphics card

- If you have a hybrid setup like for instance a gaming laptop with Nvidia Optimus, I am not sure if this will work well either. My guess is that you first should try to set your BIOS to only use dGPU for best performance. But in the end you might need to do the same thing as I did
- AMD Ryzen APUs: I honestly have no idea if there is a need to do anything. Hopefully they have not taken the design decisions Intel made
- The "_ForceGdiBackend" option did not work for me, and even if it did, it is only applicable to Ableton. Since I also use other low latency audio applications like for instance Jamulus, I needed a universal solution

Approach 1
- Change the driver from "Intel(R) UHD Graphics 630" to "Microsoft Basic Display Adapter"
- Restart your DAW (I am using Ableton Live 10)
- Manually minimize all open programs (can be done by pressing "Windows + M")

This was quick and easy to do, and magically solved all crackling!!

Approach 2
Another approach to this problem is to use the "devcon" command in .bat files to dynamically enable/disable the iGPU. Since I sometimes watch music videos on this computer, this gives more flexibility. Essentially Windows will install its basic GPU driver when the GPU is disabled.

- Disable all GPUs: Save this as a .bat file, and run as Administrator
- Enable all GPUs: Save this as a .bat file, and run as Administrator
- Flip the current GPU state(s): Save this as a .bat file, and run as Administrator

Warning: These .bat files will indeed operate on all GPUs in the system (*SUBSYS_D0001458*)

Approach 3
My final approach since I normally run my music computer without a screen is a .bat file that I run on every startup. The file does the following:

- Disables the iGPU
- Waits 10s, then starts Ableton Live 10 suite
- Minimizes all open programs every 10s (effectively presses Windows + M)
  - First download and install AutoHotKey
  - Save this text as "Minimize.ahk"
    - Optional step: Right click and select "Run Script" to verify that it works
  - Right click and select "Compile Script" to generate an executable file called "Minimize.exe", and copy it where you want it. In my case, I put it in the Ableton folder

If you think that this is too much work with AutoHotKey and/or blindly trust executable files from the internet, here is a .zip file with everything in Step 4 included.

Step 5. Other general tweaks
I found a generally useful Focusrite guide, which is applicable for most cases. In particular the steps under "Additional Optimization" was effective and not something I found in any other guides

I have found that avoiding excessive disk read/writes, can help significantly with low latency performance. So [disable swap file and hibernation](). Just make sure you have enough RAM.

Generally speaking, turning off system sounds and selecting the built-in audio interface (typically some Realtek device) as default for both Playback and recording also helps.