

Avatar Optimization

- [Avatar Optimization](#)

Avatar Optimization

What does "optimization" even mean?

"optimization" regarding avatars in VRChat refers to the process of preparing and adjusting assets to achieve the best balance between visual quality and performance efficiency. Be aware that optimization can be highly specific on the hardware you're wanting to optimize for. Hence all optimization guides are only general things that work for most people's hardware. However there are multiple key factors when it comes to optimization:

Compressed Size

The amount of data (bytes) that need to be downloaded from the VRChat CDN onto your system. This also directly affects the time it takes for your avatar to be loaded, however it also depends on the internet speed of the other person downloading your avatar and on the load of the VRChat CDN servers.

Uncompressed Size

The amount of data (bytes) after decompressing the downloaded data. This is usually split up into:

RAM Footprint

How much data needs to be stored in RAM (Random Access Memory, usually modules attached to your motherboard, but sometimes also embedded especially on laptops)

VRAM Footprint

How much data needs to be stored in VRAM (Video Random Access Memory, usually embedded into your graphics card)

Initialization Time

The time your avatar needs to be uncompressed and loaded into memory initially after it was downloaded.

Render Time

The time your avatar needs to be processed and rendered every frame.

Compressed Size

All assets used in an avatar contribute to its compressed and uncompressed size. Obviously **less data = smaller size**, so if you are able and willing to reduce the amount and size of assets, it will be the best way to reduce both compressed and uncompressed size.

However even if you don't want to/can't reduce the original size of your assets, some of them can be compressed. Compression refers to the process of reducing the size of data by encoding the data using algorithms that remove redundancy or simplify representation while allowing for retrieval of the original data (lossless) or appears acceptably similar (lossy).

Textures can be compressed via [Crunch Compression](#) (lossy)

(EDITORS NOTE: DXT, "Format", Compression Setting

<https://docs.unity3d.com/550/Documentation/Manual/class-TextureImporter.html>)

Audio clips can be compressed using the appropriate compression format (and when applicable lowering the quality slider).

For more details see [here](#)

Meshes can be compressed via [Mesh Compression](#) (?)

Animations can be compressed via [Animation Compression](#) (?)

For more details see [here](#)

Keep in mind that all forms of decompression-on-load negatively effect initialization time, because it takes time for your CPU to decompress the asset. It can also incur a **temporary** increase in RAM footprint during decompression

Additionally lossy compression can and will alter the end result of your avatar so be careful and test your avatar when using it

Uncompressed Size

Unfortunately compression only works to reduce the amount of data we need to download into RAM or send to VRAM.

However Unity offers a few ways to make decreasing asset sizes smaller (usually reducing the fidelity/quality)

(EDITORS NOTE: Texture Max Size)

Initialization Time

If an avatar has a very long initialization time, this can be observed as a noticable "hitch". This is because Unity runs synchronously on a single thread (with very few exceptions). Which means that anything that blocks this many thread will temporarily freeze VRChat. And that happens when an avatar is initialized, so the longer this takes, the bigger the hitch.

One of the biggest offenders here is loading a lot of textures/meshes into VRAM.

(EDITORS NOTE: UV Tile Discard => Immediate Load, Individual Meshes => Deferred Loading)