3DXChat World File Optimization Guide by Amber Marlow

Introduction:

Optimization is a very important part of building in the 3DXChat World Editor. Optimization is a multi-faceted and iterative process that can sometimes help to reduce lag experienced by players and builders. In this guide we'll cover everything in detail regarding optimization of world files. By the end of this guide you will have a complete and thorough understanding of the subtle nuances of how world files can be optimized. This guide is how I personally approach my optimizations and things may be done differently depending on the person and their goals.

Before You Build:

Optimization starts before you've even started building the world. Optimization starts at the conceptual level when thinking of ideas and visualizing in your mind how you want the finished product to look and feel. My general rule which is more of a guideline when considering optimizing performance is to attempt to build your idea in a 1 MB file size or less. For some people this may seem like a very small size compared to some worlds that are in excess of 5 MB. This is simply a target to aim for. You don't have to hit it. Giving yourself this constraint will help to improve your building efficiency. Doing this will force you to find new and creative ways to get more details with less objects.

A world file that is 1 MB or less will 99% of the time never lag even on 20 year old hardware and you should be able to get at least a consistent 30 fps. Many people have trouble loading worlds that are larger than 5 MB and usually require more powerful gaming computers to run smoothly. It's a fine line between making your vision come to life, while also allowing it to be accessible to a wider variety of players. Many times you won't be able to fully flesh out your vision without exceeding 1 MB depending on the type of room you're building, however it's best practice to keep that 1 MB target in mind through the entire building process.

Building Efficiently:

Building efficiently can be quite comprehensive but I'll do my best to simplify the process. This guide is not meant to teach you how to build, only to adjust some of your building techniques to produce more efficient worlds with fewer objects.

There are many ways to build and many ways to approach solving problems in world editor and sometimes one method may be better than another. Without getting too technical, the best way to keep your world efficient is to build and place the biggest objects in the room first, working your way down to the smaller objects. For example, when building a beach, start with your sand or island, then place any trees or rocks and determine the layout of the world. The landscaping aspect will be the most time consuming when doing it properly and there really are no shortcuts when doing it right.

A general rule is to build your world based on player perspective. If the players will only be walking around in specific areas, those are the places that should contain the most detail. Features that are further away should be scaled larger to fill in the visual space. This way you can use less objects for the same visual result. Try to build one thing at a time and build in layers. Terrain, then trees, then rocks, then grass and plant details, all while keeping in mind the paths players will take when walking.

Scaling distant objects to be larger will greatly improve world file size and performance. When placing things like trees and grass, do your best to avoid dense vegetation and foliage. Place grass and trees only around key areas in your world that will blend hard lines in the landscape, essentially using it as a visual transition between textures.

3DXChat World File Optimization Guide by Amber Marlow

Dynamic Lighting & Saturation:

Static lighting generally has a relatively low impact on performance when building a world. Most of the time you're able to get away with a large number of light sources, potentially hundreds, with minimal performance impact. However, dynamic lighting doesn't behave the same way. By dynamic lighting, I mean any light source that has motion or is subject to variability. Most of the time, lag happens because of CPU constraints, not GPU constraints. By default your CPU will work much harder than your GPU to render the world mostly because the CPU is not designed for rendering tasks. Too many dynamic lights in one area can cause lag even with a world file that is less than 1 MB in size due to the CPU being overloaded. This can also cause a low-end CPU to overheat, which can cause damage to the CPU or components surrounding it if cooling is insufficient. Best practice is to avoid too many dynamic lights in one area.

Global Lighting Modifications & Skybox:

Changing the skybox to something brighter with more uniform lighting can help to increase performance. Bright rooms that make heavy use of skybox lighting instead of a dark room using static or dynamic light objects, generally perform much better. Darker rooms with more custom static lighting is much more computationally demanding. Sometimes simply changing your skybox can be the difference between 10 fps and 30 fps.

Segmentation & Instances:

Further optimization of your world can be achieved by segmenting areas of the world into separate instances. This is very similar to how games like World of Warcraft handle dungeons and raids. They are called instances. In this case, instance is a very loose term but is still very much applicable. Segmentation is a much better term for this technique.

In world editor, objects have a maximum render distance from the camera. If an object is too far away from the camera it will not be rendered and become invisible until it is approached my the camera. This is a built-in performance feature of world editor, and we're simply exploiting it.

Designing your room in a modular way early on can make segmentation much easier in the later stages of building. Design entire areas of your world as a separate collection of grouped objects, either terrain or structures and then move them out of render distance, connecting the locations using a teleport.

The render distance is a spherical bubble around the camera and moves with the camera so you can move the object in any direction out of render distance and it will be the same distance on all sides. It's not usually human nature to look up or down, so when moving things out of render distance to increase performance, it's best practice to move objects either above or below because player names are not affected by this same render distance. Names will always render past the cameras render distance. By moving things above or below, it's much less likely someone will see names during normal camera operation.

Essentially what you're doing here is allowing your camera to not have to render as much at any given time. Segmentation is a great way to improve performance especially with worlds that exceed 5 MB in size.

3DXChat World File Optimization Guide by Amber Marlow

Removing Duplicate Objects:

Assuming the above steps have been implemented and taken into consideration, at this point your world file should already be heavily optimized. Generally you'd want to wait until your world is fully completed and you have a final version you're happy with before attempting to remove duplicate objects, however there is no harm in doing this at any point in the building process.

Removing duplicate objects is good practice because it's very common for world editor to do undesirable things under normal operation. For example, sometimes when duplicating objects, either as a result of user error or world editor glitches, you may end up making too many unwanted duplicates of objects that you wouldn't be aware of because they do not have clipping and occupy the same space. This can happen especially when using the same texture on multiple objects. It's also very difficult to select these duplicates because they occupy the same space. It's not uncommon to do a lot of work only to find that now you have hundreds of objects with duplicate counterparts. Manually removing them is very time consuming and wasteful.

There are many free tools available to remove duplicate objects. Find one that works best for you. I would always recommend using the tool I've created for this which can be found in my section of The Builder's Corner. It will remove all duplicates from a world file and show you the file size before and after, with clear representation of how much space in KB was saved. My tool also has an option to tell you exactly which objects were removed in an organized list.

Final Thoughts & Conclusions:

You should now have a solid understanding of how to increase the performance and efficiency of your world files when building. I'm certain there are many more things that you could do to help with lag however I'm confident the steps outlined in this guide carry the biggest weight factor in performance impact based on my 8 years of building experience. The bottom line outside of these considerations and steps really just comes down to hardware. If you've done all of the above and you're still lagging, it may be time to consider upgrading your hardware. Optimization will only get you so far and does have limitations.

Additional Resources:

If you've read through this guide and you struggle with any of the concepts outlined, you're more than welcome to PM me on discord and I'll do my best to have some more personalized time with you to see how I can help further. Thank you for taking the time to read this guide and I sincerely hope it helps you.

Much love,
Amber Marlow

Revision: 1.0
Revised Date: Feb 07, 2025