Level Streaming Volumes Reference

Guide to using volumes to control the streaming of Levels based on the player's viewpoint.

Level Streaming Volumes are used to aid in the Level Streaming process. They provide a simple way to encapsulate a level, as well as control when it streams in and out of memory, based on when a Player enters or exits the Volume.

You can adjust how a Level Streaming Volume handles level streaming by adjusting the properties from the **Details** panel, pictured below.

Property	Description
Streaming Levels	Displays the levels affected by the volume.
Editor Pre Vis Only	Determines if the streaming volume should only be used for editor streaming level previs.
Disabled	If true, the streaming volume is ignored by the streaming volume code. Also used to either disable a Level Streaming Volume without disassociating it from the level, or to toggle the control of a level's streaming between Blueprints and Volume Streaming.
Streaming Usage	Determines what the volume is used for, e.g. whether to control loading, loading and visibility, or just visibility (blocking on load).

Associating Streaming Volumes With Levels

Volume-based level streaming works as follows: each streaming Level can have associated with it a set of Level Streaming Volumes. Each frame, the engine iterates over each Level and checks to see if the player's viewpoint is inside any of the Level Streaming Volumes associated with that Level. If the viewpoint is inside at least one Level Streaming Volume, a request is issued to begin loading that Level. If the viewpoint is outside all Level Streaming Volumes, the Level is marked for unloading.

- Level Streaming using Volumes

Important Details

- All Level Streaming Volumes must exist in the persistent Level. Level Streaming Volumes that live in other Levels cannot be used for level streaming, and will generate warnings when the map is checked for errors.
- If a Level has any streaming volumes associated with it, other methods of streaming the Level will not behave correctly.
- A single Level Streaming Volume can affect multiple Levels. Similarly, a single Level can be affected by multiple Level Streaming Volumes.
- Volume-based streaming works for split screen. The viewpoints of all local players are considered before any loading/unloading requests are issued.

Testing Your Streaming Volume Setup

It is critical that volume-based level streaming be tested in game on the target platform. Streaming in **Play in Editor** mode (PIE) will show where the loads/unloads will happen, but streaming in PIE is not representative of real, in-game loading-unloading. This is because in PIE, the Levels are already in memory, and so "loading" a Level is simply a matter of unhiding it instantly.

Running the Level in the standalone game on the target platform is critical in making sure your streaming setups work. Note that on some platforms, it can take several seconds to stream in a Level. Size your Level Streaming Volumes appropriately, so that the Level is loaded by the time the player can reach it. Level loading behavior can be modified by resizing the Level Streaming Volumes associated with a Level. Growing a volume causes associated Levels to load sooner and unload later, while shrinking a volume causes later loads and earlier unloads.

Level Streaming Volumes for Previs

A Level Streaming Volume can be marked for editor previsualization only by setting the **Editor Pre Vis Only** flag on the Level Streaming Volume. In this manner, volume-based level streaming can still be used for editor previs while at the same time using another streaming method for in-game streaming.

Cost of Level Streaming Volumes

Each frame, `UWorld::ProcessLevelStreamingVolumes` iterates over each streaming Level and for each Level, that Level will begin loading if any local player is within any of the volumes associated with that Level. Likewise, the Level will be begin unloading if all local players are outside all volumes.

`UWorld::ProcessLevelStreamingVolumes` exploits coherency in the following manner: for each Level, the volume that most recently contained a player is cached. This cached volume is checked first, so that Levels a player is in or returns to are quickly accepted.

Volumes of any shape are fine, although obviously the fewer the better. An upper bound on the cost of Level Streaming Volumes can be approximated by the sum of the number of Level Streaming Volumes associated with unloaded Levels.

Two stats exist under the "Streaming" stats group for monitoring level streaming performance. The "Streaming Volumes" stat tracks the number of Level Streaming Volumes tested against player viewpoints per frame, while the "Volume Streaming Tick" stat tracks the amount of time spent in `UWorld::ProcessLevelStreamingVolumes` per frame.

Adding Hysteresis to Unloading Requests

A player moving back and forth across a Level Streaming Volume boundary causes unneeded load/unload requests to be issued. To address this, hysteresis has been added to unloading requests. No hysteresis exists for loading requests, because if a Level needs to be loaded, we always want it loaded as soon as possible.

The amount of unloading hysteresis can be adjusted by modifying the **Min Time Between Volume Unload Requests** property of the streaming Level in the Levels window. The default unloading hysteresis is 2.0 seconds.

Disabling Level Streaming Volumes

There is a property on Level Streaming Volumes called **bDisabled**. When set to true, the volume will be ignored by the streaming volume code, both in-game and in the editor. **bDisabled** can be used to disable a Level Streaming Volume without dissociating it from the Level.

As an example of where the **bDisabled** flag is useful, imagine a door leading to a Level whose streaming is controlled with streaming volumes. The streaming volume extends out past the door, so that the Level will be streamed in by the time the player can reach the door and open it. Initially, however, the door is locked, and will become unlocked when the player achieves an objective in another part of the Level. So, even though the streaming volume extends past the door, we don't want the Level on the other side of the door to stream in unless the door is actually unlocked ("openable").

Blueprints and C++ could both be used to toggle the disabled state of **Level Streaming Volumes**.