

# Unloaded Chunks

## Unloaded Chunks

Unloaded Chunks or Cached Chunks (but officially called unloaded chunks) are chunks that have most but not all functionality disabled within them. These chunks are not simulated and cannot process things like block updates and pending ticks. There are three main natural instances of unloaded chunks in a Minecraft world, [Spawn Chunks](#), Chunks in your render distance, and the "perma-cache" phenomenon. These chunks do process some events however and knowing what does and does not work will help you be able to interact with these chunks in much more confidence.

## What Works In Them

The things we know to work in unloaded chunks:

- Redstone circuit structure
- Capacitors (E.g, Redstone Torches, Repeaters, Comparators)
- Redstone Dust
- Global Entities (Player Owned Projectiles, Ender Dragon, fireworks, lightning)
- Block state changes from a wind charge
- Sculk Spread
- Block Creation
- Tripwire
- Target Block
- Pressure Plate
- Damaging/Killing Entities (Non-Global Entities will not be fully dead/deleted until simulated)
- Artificial Entity Spawning
- Breeze Projectile Redirection
- Pending Tick Scheduling (Not execution)
- Wind Charging effect.
- Weaving Effect

Although Observers are a "Pulse Capacitor" they cannot fire in unloaded chunks due to their reliance on pending ticks.

Since tridents are player-owned projectiles, they can fall through the world if they spawn in an unloaded chunk before the chunk's block collision map is initialized especially when the world is running slowly. Grindstones are used to combat this.

## Perma-Caching

Perma-Caching is an unexplained phenomenon that happens when a player spends enough time in a given part of their world. The game will generate a seemingly random arrangement of chunks around the area that will not become unprocessed and stay in the unloaded state. This phenomenon does not seem to go away with relog or waiting.

## Why "Unloaded Chunks"

We call them unloaded chunks because, in game, functions like these refer to them as unloaded chunks. This allows us to keep consistency with what the game developers at Mojang refer to them as internally. We do this because it helps us know what to look for when learning about something related to them.

```
void Dimension::transferEntityToUnloadedChunk(Actor& actor, LevelChunk* oldChunk)
```

These chunks are represented by the hex values 0x0000 through 0x0008 but each value is a different state of an unloaded chunk.

---

Revision #9

Created 2025-09-14 18:01:51 UTC by Nytreon

Updated 2025-11-15 19:50:40 UTC by Nytreon