PaperDocs Documentation

Release 1.12.2

PaperMC Team

Jan 12, 2019

Contents

> **Warning:** This documentation is a work in progress. Some things are still missing. If you'd like to see it completed faster, you can contribute to the documentation here.

Introduction

Welcome to PaperDocs, the official documentation source for the PaperMC project.

Paper is a high performance fork of the Spigot Minecraft Server that aims to fix gameplay and mechanics inconsistencies as well as to improve performance.

Paper contains numerous features, bug fixes, exploit preventions and major performance improvements not found in Spigot.

Not sure what you're looking for? Try our *About the PaperMC Project* section, which contains a short *Introduction*, our *Frequently Asked Questions* and an article about *The Structure of PaperMC*.

Server owners should read our tips and instructions in *Running A Server* on how to install, run and maintain a Paper based server.

The PaperMC project has several main components. Learn more about the project structure *here*.

Running A Server

Running a Paper server is easy. This section will cover common tasks such as configuring it, maintaining plugins, and general best practices.

Running A Server

Running a Paper server is easy.

Click the links below for more information.

Contents

Getting Started

- *Requirements*
- *Migrating From Vanilla*
- *Migrating From CraftBukkit/Spigot*
- *Getting A Server Jar*
- *Running The Server*
- *Updating The Server*

Requirements

Paper requires **Java 8** or newer to run. It should run fine on newer versions as well, such as Java 10, but certain plugins may have issues.

Migrating From Vanilla

Migrating from Vanilla is easy, but there are some differences, namely in world saves. Paper (and CraftBukkit and Spigot) seperate out each dimension of a world (nether, the end, etc) into separate world folders.

Paper will handle this conversion for you automatically.

Migrating From CraftBukkit/Spigot

Paper is a drop in replacement for both CraftBukkit and Spigot, you don't need to make any changes.

Getting A Server Jar

Paper provides runnable server jars directly from our continuous integration server here.

Download one of the artifacts (either paperclip.jar or paperclip-####.jar, they are the same) and save it to a new directory somewhere on your file system.

Running The Server

To run the server, simply start it up like any other Java application.

Open your terminal, navigate to the saved location, and then run `java -jar paperclip.jar`

For more advanced Java tuning, see this page

To configure your server, see the *Configuration* page.

Updating The Server

To update the server, first stop it safely by executing the `stop` command and then replace the old paperclip jar with a new one. That's it.

Configuration

This page details the various configuration settings exposed by Paper.

- *Global Settings*
- *World Settings*

If you want information on settings in spigot.yml and bukkit.yml you should see their respective documentation pages.

- Bukkit Configuration (bukkit.yml)
- Spigot Configuration (spigot.yml)

> **Warning:** Configuration values change frequently at times. It is possible for the information here to be incomplete. If you cannot find what you're looking for or think something may be wrong, *Contact Us*
>
> Last updated April 5th 2018 for MC 1.12.2

Global Settings

Global settings affect all worlds on the server as well as the core server functionality itself.

verbose

- **default**: false
- **description**: Sets whether the server should dump all configuration values to the server log on startup.

enable-player-collisions

- **default**: true
- **description**: Sets whether the server should allow players to collide with one another.
- **warning**: This setting can be broken by plugins interacting with the scoreboard, double check plugins when troubleshooting this value.

player-auto-save-rate

- **default**: -1
- **description**: Sets the tick delay between automatic player data saves. Default inherits from world autosave.

max-player-auto-save-per-tick

- **default**: -1
- **description**: Controls how many players may have their data autosaved per tick. The default is to automatically use Paper's recommended value, currently 10, subject to change as needed.

load-permissions-yml-before-plugins

- **default**: true

- **description**: Loads bukkit's permission.yml file before plugins, allowing them to check permissions immediately on enable.

bungee-online-mode

- **default**: true

- **description**: Instructs the server how to behave when behind bungee Set to match your proxy's online-mode setting.

sleep-between-chunk-saves

- **default**: false

- **description**: Instructs the server to sleep the chunk save thread after every chunk save.

- **warning**: Setting this value to true may lead to increased memory usage and slower chunk saving.

region-file-cache-size

- **default**: 256

- **description**: Sets the maximum size of the region file cache.

save-empty-scoreboard-teams

- **default**: false

- **description**: Some scoreboard plugins leave hundreds of empty scoreboard teams around, dramatically slowing down login times. This sets whether the server should remove those empty teams automatically.

incoming-packet-spam-threshold

- **default**: 300

- **description**: Sets the threshold at which the server will consider incoming packets as spam and ignore them.

remove-invalid-statistics

- **default**: false

- **description**: Instructs the server to remove invalid statistics from the world save data as it's loaded.

min-chunk-load-threads

- **default**: 2

- **description**: Sets the minimum number of threads to use for async chunk loading File I/O.

- **warning**: This value only affects File I/O threads; increasing it without reason is unlikely to improve chunk load speeds.

suggest-player-names-when-null-tab-completions

- **default**: true

- **description**: Instructs the server to return a list of Players when tab-completing if the plugin has no tab completions of its own.

timings

- **enabled**

    - **default**: true

    - **description**: Controls the global enable state of the Timings platform.

- **verbose**

    - **default**: true

    - **description**: Instructs Timings to provide more specific information in its reports. For example, specific entity types causing lag rather than just "entities".

- **server-name-privacy**

    - **default**: false

    - **description**: Instructs Timings to hide server name information in reports.

- **hidden-config-entries**

    - **default**: { database, settings.bungeecord-addresses }

    - **description**: Configuration entries to hide in Timings reports.

- **history-interval**

    - **default**: 300

    - **description**: The interval in seconds between individual points in the Timings report

- **history-length**

    - **default**: 3600

    - **description**: The total amount of data to keep for a single report.

    - **warning**: This value is validated server side, massive reports will be rejected by the report site.

messages

- **kick**

    - **authentication-servers-down**

        * **default**: ' ' (empty string)

        * **note**: The default value instructs the server to send the vanilla translateable kick message.

        * **description**: Message to kick a player with when they are disconnected because the Mojang authentication servers are down.

    - **flying-player**

        * **default**: Flying is not enabled on this server

        * **description**: Message to use when kicking a player for flying.

    - **flying-vehicle**

        * **default**: Flying is not enabled on this server

        * **description**: Message to use when kicking a player's vehicle for flying.

World Settings

World settings are configured on a per-world basis. The child-node *default* is used for all worlds that do not have their own specific settings.

keep-spawn-loaded-range

- **default**: 8
- **description**: The number of chunks around spawn to keep loaded.

auto-save-interval

- **default**: -1
- **note**: Default value instructs the world to use Bukkit's default.
- **description**: Instructs this world to use a specific value for auto-save instead of bukkit's global value.

parrots-are-unaffected-by-player-movement

- **default**: false
- **description**: Makes parrots "sticky" so they do not fall off a player's shoulder when they move. Use crouch to shake them off.

game-mechanics

- **disable-player-crits**

    - **default**: false

    - **description**: Instructs the server to disable critical hits in PvP, instead treating them as normal hits.

- **disable-chest-cat-detection**

    - **default**: false

    - **description**: Allows you to open chests even if they have a cat sitting on top of them.

- **disable-end-credits**

    - **default**: false

    - **description**: Instructs the server to never send the end game credits when leaving the end.

disable-explosion-knockback

- **default**: false

- **description**: Instructs the server to completely block any knockback that occurs as a result of an explosion.

elytra-hit-wall-damage

- **default**: true

- **description**: Instructs the server to calculate and do player damage when a player flies into a wall while using an elytra.

grass-spread-tick-rate

- **default**: 1

- **description**: Sets the delay, in ticks, at which the server attempts to spread grass. Higher values will result in slower spread.

allow-leashing-undead-horse

- **default**: false

- **description**: Instructs the server to allow players to leash undead horse types.

frosted-ice

- **enabled**

    - **default**: true

    - **description**: Instructs the server to enable (and tick) frosted ice blocks

- **delay**

    - **min**

* **default**: 20

* **description**: minimum RNG value to apply frosted-ice effects at.

– **max**

* **default**: 40

* **description**: maximum RNG value to apply frosted-ice effects at.

hopper

- **push-based**

  – **default**: false

  – **description**: Instructs the server to use an alternative hopper system in which items push themselves into hoppers.

  – **warning**: This feature improves performance but is buggy and known to cause problems.

- **cooldown-when-full**

  – **default**: true

  – **description**: Instructs the server to apply a short cooldown when the hopper is full, instead of constantly trying to pull new items.

- **disable-move-event**

  – **default**: false

  – **description**: Completely disables the *InventoryMoveItemEvent* for hoppers. Dramatically improves hopper performance but will break protection plugins and any others that depend on this event.

baby-zombie-movement-speed

- **default**: 0.5
- **note**: For reference, the vanilla player moves with a speed of 0.1.
- **description**: Controls the speed baby zombies move at.

keep-spawn-loaded

- **default**: true
- **description**: Instructs the server to keep the spawn chunks loaded at all times.

enable-treasure-maps

- **default**: true
- **description**: Allows villagers to trade treasure maps.
- **note**: Disabling this may help keep available map IDs higher.

treasure-maps-return-already-discovered

- **default**: false
- **description**: Instructs the server to target the first treasure location found, rather than the first undiscovered one. Vanilla mechanics normally find the first undiscovered location, which may lead to structures that were not fully looted, and can also fail with a world border set. Enabling this will make the map simply find the closest target structure, regardless if it has been loaded or not already.

nether-ceiling-void-damage

- **default**: false
- **description**: Instructs the server to do void damage when an entity is on top of the nether ceiling. Use to limit nether ceiling travel in a vanilla-lore friendly way.

allow-non-player-entities-on-scoreboards

- **default**: false
- **description**: Instructs the server to treat non-player entities as if they are never on a scoreboard.
- **note**: Enabling this value may increase the amount of time the server spends calculating entity collisions.

container-update-tick-rate

- **default**: 1
- **description**: The rate, in ticks, at which the server updates containers and inventories.

use-alternate-fallingblock-onGround-detection

- **default**: false
- **description**: Uses an alternative detection system to better handle falling blocks getting stuck on objects.

prevent-tnt-from-moving-in-water

- **default**: false
- **description**: Instructs the server to keep Primed TNT entities from moving in flowing water.

non-player-arrow-despawn-rate

- **default**: -1
- **note**: The default value instructs the server to use the same default arrow despawn rate from spigot.yml that is used for all arrows.
- **description**: The rate, in ticks, at which arrows shot from non-player entities are despawned.

anti-xray

- **enabled**

    - **default**: false

    - **description**: Controls the on/off state for the Anti-Xray system.

- **engine-mode**

    - **default**: 1

    - **description**: Sets the Anti-Xray engine mode. Where 1 is to replace hidden blocks with stone and 2 is to replace all blocks with random block data.

- **chunk-edge-mode**

    - **default**: 3

    - **description**: Sets how the engine handles chunk edges. Where 1 is not

    to obfuscate the edges of chunks with unloaded neighbors, 2 is to not send the chunk until its neighbors are present (similar to a x-1 view distance), and 3 is to load the neighbor so it can properly obfuscate the current edge (similar to an x+1 view distance).

- **max-chunk-section-index**

    - **default**: 3

    - **description**: Controls to what Y value (height) the engine should operate to, expressed in chunk sections.

    - **note**: To determine the total height, use this formula: ($index + 1) * 16. Therefore, the default value of 3 will result in the engine functioning up to Y: 64.

- **hidden-blocks**

    - **default**: { gold_ore, iron_ore, coal_ore, lapis_ore, mossy_cobblestone, obsidian, chest, diamond_ore, redstone_ore, lit_redstone_ore, clay, emerald_ore, ender_chest }

    - **description**: List of blocks to be hidden in engine mode 1.

    - **note**: This list is using Mojang server names *not* bukkit names.

- **replacement-blocks:**

    - **default**: { stone, planks }

    - **description**: List of blocks that should be replaced by hidden-blocks in engine mode 2.

    - **note**: This list is using Mojang server names *not* bukkit names.

experience-merge-max-value

- **default**: -1

- **description**: Instructs the server put a maximum value on experience orbs, preventing them all from merging down into 1 single orb.

- **note**: The default value instructs the server to use no max value, allowing them to merge down into a single orb.

armor-stands-do-collision-entity-lookups

- **default**: true
- **description**: Instructs armor stand entities to do entity collision checks.

queue-light-updates

- **default**: false
- **description**: Instructs the server to queue up lighting updates to the end of the tick, then run them as it has free time, instead of running them as they're scheduled.

optimize-explosions

- **default**: false
- **description**: Instructs the server to cache entity lookups during an explosion, rather than recalculating through-out the process.

use-chunk-inhabited-timer

- **default**: true
- **description**: Instructs the server to factor the chunk inhabited timer into various calculations.
- **note**: The timer is increased when chunks are kept loaded because of player activity.

use-vanilla-world-scoreboard-name-coloring

- **default**: false
- **description**: Instructs the server to use the vanilla scoreboard for player nickname coloring.
- **note**: Useful when playing on adventure maps made for the vanilla server and client.

delay-chunk-unloads-by

- **default**: 10s
- **description**: Instructs the server to delay chunk unloads by this value, in seconds, to prevent chunk load/unload thrashing.

max-auto-save-chunks-per-tick

- **default**: 24
- **description**: The maximum number of chunks the auto-save system will save in a single tick.

save-queue-limit-for-auto-save

- **default**: 50
- **description**: Instructs the server to skip adding more chunks to the auto save queue until it's below this value.

remove-corrupt-tile-entities

- **default**: false
- **description**: Instructs the server to automatically remove tile entities it detects as broken and cannot fix.

max-chunk-sends-per-tick

- **default**: 81
- **description**: The maximum number of chunks the server will send out to clients per-tick.

max-chunk-gens-per-tick

- **default**: 10
- **description**: The maximum number of chunks the server will generate in a single tick.

max-growth-height

- **cactus**
    - **default**: 3
    - **description**: Maximum height cactus blocks will naturally grow to.
- **reeds**
    - **default**: 3
    - **description**: Maximum height reed blocsk will naturally grow to.

fishing-time-range

- **MinimumTicks**
    - **default**: 100
    - **description**: The minimum number of RNG ticks needed to catch a fish.
- **MaximumTicks**
    - **default**: 600
    - **description**: The maximum number of RNG ticks before catching a fish.

despawn-ranges

- **soft**

    - **default**: 32

    - **description**: The number of blocks away from a player in which entities will be randomly selected to be despawned.

- **hard**

    - **default** 128

    - **description**: The number of blocks away from a player in which entities will be forcibly despawned.

falling-block-height-nerf

- **default**: 0

- **note**: Values less than 1, will disable this feature.

- **description**: The height at which falling blocks will be removed from the server.

tnt-entity-height-nerf

- **default**: 0

- **note**: Values less than 1, will disable this feature.

- **description**: The height at which Primed TNT entities will be removed from the server.

water-over-lava-flow-speed

- **default**: 5

- **description**: Sets the speed at which water flows while over lava.

fast-drain

- **lava**

    - **default**: false

    - **description**: Controls whether lava blocks should drain more quickly when their source blocks are removed.

- **water**

    - **default**: false

    - **description**: Controls whether water blocks should drain more quickly when their source blocks are removed.

lava-flow-speed

- **normal**

    - **default**: 30

    - **description**: Sets the speed at which lava flows in the overworld.

- **nether**

    - **default**: 10

    - **description**: Sets the speed at which lava flows in the nether.

lootables

- **auto-replenish**

    - **default**: false

    - **description**: Instructs the server to automatically replenish lootable containers.

    - **note**: This feature is useful for long-term worlds in which players are not expected to constantly explore to generate new chunks.

- **restrict-player-reloot**

    - **default**: true

    - **description**: Prevents the same players from coming back and re-looting the same containers over and over.

- **reset-seed-on-fill**

    - **default**: true

    - **description**: Resets the loot seed each time the lootable is refilled. Effectively randomizing the new loot items on each refill.

- **max-refills**

    - **default**: -1

    - **description**: Sets the maximum number of times a lootable may be refilled.

    - **note**: The default value will allow a lootable to refilled an infinite number of times.

- **refresh-min**

    - **default**: 12h

    - **description**: The minimum amount of time that must pass before a lootable will be eligible to be refilled.

    - **note**: This field uses time-based values. 12s = 12 seconds, 3h = 3 hours, 4d = 4 days.

- **refresh-max**

    - **default**: 2d

    - **description**: The maximum amount of time that can pass before a lootable is refilled.

    - **note**: This field uses time-based values. 12s = 12 seconds, 3h = 3 hours, 4d = 4 days.

filter-nbt-data-from-spawn-eggs-and-related

- **default**: true
- **description**: Instructs the server to remove certain NBT data from spawn-eggs, falling-blocks, and other often abused items in creative mode.
- **note**: Some adventure maps may require this be turned off to function correctly, but we do not recommend turning it off on a public server.

max-entity-collisions

- **default**: 8
- **description**: Instructs the server to stop processing collisions after this value is reached.

disable-creeper-lingering-effect

- **default**: false
- **description**: Disables creepers randomly leaving behind a lingering area effect cloud.

disable-thunder

- **default**: false
- **description**: Disables thunderstorms

skeleton-horse-thunder-spawn-chance

- **default**: 0.01
- **description**: Sets the chance that a "Skeleton Trap" (4 skeleton horsemen) will spawn in a thunderstorm.

disable-ice-and-snow

- **default**: false
- **description**: Disables ice and snow formation.

fire-physics-event-for-redstone

- **default**: false
- **description**: Instructs the server to fire the *BlockPhysicsEvent* when redstone is activated.

mob-spawner-tick-rate

- **default**: 1
- **description**: How often mob spawners should tick to calculate available spawn areas and spawn new entities into the world.

all-chunks-are-slime-chunks

- **default**: false

- **description**: Instructs the server to treat all chunks like slime chunks, allowing them to spawn in any chunk.

- **note**: This may actually decrease your chances of running into a Slime as they now have a much larger potential spawn area.

squid-spawn-height

- **maximum**

    - **default**: 0.0

    - **description**: The maximum height at which squids will spawn.

    - **note**: The default value defers to Minecraft's default setting, which as of 1.12 is the sea-level of the world (usually Y: 64).

disable-teleportation-suffocation-check

- **default**: false

- **description**: Disables the suffocation check the server performs before teleporting a player.

- **note**: While useful to keep your players out of walls, leaving this feature on may enable players to teleport through solid materials by logging out in specific locations.

portal-search-radius

- **default**: 128

- **description**: The maximum range the server will use to look for an existing nether portal. If it can't find one in that range, it will generate a new one.

generator-settings

- **canyon**

    - **default**: true

    - **description**: Instructs the server to generate canyons.

- **caves**

    - **default**: true

    - **description**: Instructs the server to generate caves.

- **dungeon**

    - **default**: true

    - **description**: Instructs the server to generate dungeons.

- **fortress**

    - **default**: true

- **description**: Instructs the server to generate fortresses.

- **mineshaft**

    - **default**: true

    - **description**: Instructs the server to generate mineshafts.

- **monument**

    - **default**: true

    - **description**: Instructs the server to generate monuments.

- **stronghold**

    - **default**: true

    - **description**: Instructs the server to generate strongholds.

- **temple**

    - **default**: true

    - **description**: Instructs the server to generate temples.

- **village**

    - **default**: true

    - **description**: Instructs the server to generate villages.

- **flat-bedrock**

    - **default**: false

    - **description**: Instructs the server to generate bedrock as a single, flat, layer.

- **disable-extreme-hills-emeralds**

    - **default**: false

    - **description**: Prevents the server from generating emeralds in extreme hills biomes.

- **disable-extreme-hills-monster-eggs**

    - **default**: false

    - **description**: Prevents the server from generating monster egg blocks in extreme hills biomes.

- **disable-mesa-additional-gold**

    - **default**: false

    - **description**: Prevents the server from generating extra gold in mesa biomes.

Contributing to Paper

It's great to see new contributors to the project! Here are some documents which provide further information to get you started:

Contributing

This page has yet to be written.

It will contain information about contributing to the PaperMC project.

For now see our main repository's contributing information.

About Paper

Paper is a fork of Spigot with certain goals in mind. The following links explain more about the purpose, rationale and structure of the PaperMC project.

About the PaperMC Project

This section provides basic information about the project, as well as covering asset information, frequently asked questions, and contact information.

Contents

Introduction

What is Paper?

Paper, and the PaperMC project behind it, exist to expand the platforms upon which it is built. Paper has a relatively small team, despite that, we have received a large amount of attention and a great showing of support from both server administrators and fellow developers.

As such, we focus on a few things:

- A performant server is paramount.
- Giving back to the community matters.
- Mojang's decisions and assumptions are not necessarily the best for everyone.

Paper is largely composed of two projects:

- **Paper-API**, An enhanced version of the Bukkit API.
- **Paper-Server**, usually just referred to as *Paper*, Paper-Server is an enhanced implementation of the server that goes hand-in-hand with our enhanced API.

There are also other, more technical, projects that support the above two projects. For more information on that see *The Structure of PaperMC*

Where can I download Paper?

A complete list of available builds can be obtained from our continuous integration server

Frequently Asked Questions

General

What is Paper?

Paper is a fork of the Spigot server implementation (which is itself a fork of CraftBukkit). Paper strives to bring improved performance, more features, and more APIs for developers to build awesome plugins with.

What do I need to run it?

Paper requires the Java Runtime Environment to run. Specifically, it requires at least Java version 8. Once that is installed you're all good to go! If you don't already have a Java 8 Runtime, it's easy to download and install.

See our docs on starting out: *Getting Started*

Where do I get it?

Builds of Paper are already available from our continuous integration server.

Alternatively, you can always get the latest build by clicking this link.

Server Administrators

What can I expect from switching to Paper?

When migrating your CraftBukkit or Spigot server to Paper, it is not uncommon to see a noticeable performance improvement.

Note: Though you may see an improvement, Paper is not a silver bullet. Ultimately, you are responsible for the performance of your server, good or bad, on any platform. Tailoring your server to best fit your players and gamemodes is ultimately the key to great performance.

Your plugins and worlds will not be changed and both should work just fine after the change.

Will players be able to tell?

That depends. Your players may see a benefit to gameplay because of the performance improvement, assuming you see one. On a properly maintained server, your players may not even be able to tell the difference.

Can I run Bukkit plugins on Paper?

Yep! You absolutely can. Paper takes care to maintain compatibility with Bukkit plugins made by the community.

Can I run Spigot plugins on Paper too?

Yes you can! We don't like to break things most of the time. Sometimes there are plugin authors who do, but we can usually make things work.

Is there anywhere to get plugins for Paper?

Many plugins that work with, and are made for, Paper are available on the AquiferMC resource section. Sometimes you'll also see them elsewhere, you just have to keep your eyes open.

Does Paper support Forge Mods?

No, Paper does not support Forge mods of any kind. While there have been attempts to merge the Forge and Bukkit platforms in the past, it has never been a wonderful experience for developers or administrators.

If this is something you're after, we'd point you towards the Sponge Project instead.

Developers

What can I do with Paper?

Paper provides additional APIs ontop of Bukkit, exposing new vanilla elements and even some of its own for you to play with.

TODO: Link to developer docs here :D

Does Paper make any breaking changes to the API?

Fortunately, Paper does not make breaking API changes so it can maintain plugin compatibility with upstream Spigot and CraftBukkit. At the same time, this means we are also sometimes limited with what we can do and how we can do it.

It's a double-edged sword.

The Structure of PaperMC

Project	Description
Paper	Shared API & Server repository
PaperDocs	All documentation for the project
Paperclip	A wrapper around Paper-Server to make it easy to distribute and easy to use
PaperTestServer	A skeleton test server we use for testing plugin compatibility

Paper

This is our main repository. Changes to Spigot-API and Spigot-Server are maintained as a (large) set of individual patch files in this repository. This repository contains only bash scripts and patch files. The bash scripts present in this repo will then take the patches to Spigot-API and to Spigot-Server and apply them.

From this process comes a proper, buildable, Paper-API and Paper-Server that we can then build as a standard maven project.

Paper-API is then compiled and sent to up to our maven server for developers to build their plugins against.

Paper-Server is compiled, then handed off to the Paperclip launcher tool. More on that in a second. From there, we distribute it to you!

Paperclip

Paperclip is a binary patch distribution system for Paper-Server. We give it a normal, compiled, Paper-Server jar file. From there, Paperclip generates a binary patch of the difference between that Paper-Server jar and the vanilla Minecraft server. It then takes that binary patch and includes it in a separate jar file along with some information about the version of Minecraft it came from, stored as a JSON file, and some minimal code to launch it.

When an end-user runs Paperclip for the first time a few things happen.

1. Paperclip checks the JSON file it stores internally to see what version of Minecraft it was built against.

2. Paperclip downloads a vanilla minecraft jar for that particular version directly from Mojang.

3. Paperclip checks to see that the vanilla jar it just downloaded is the same as the one it was built against using a SHA-256 hash.

4. Paperclip applies the included binary patch of Paper changes to the vanilla jar.

5. Paperclip verifies that the final patched version of the server is the same as the one we originally built using a SHA-256 hash.

6. Paperclip does some java classloader magic and starts the server right then and there.

It only does this on the first run, after this first run, it will skip parts of that process as it deems necessary.

For example, when Paper pushes a new build of Paper for a specific Minecraft version, Paperclip will not re-download the vanilla jar, it'll simply re-patch it and start up.

If there is a Minecraft version change (like 1.9 to 1.10), only then will Paperclip re-download the vanilla Minecraft jar.

Art Assets

This page provides the official PaperMC logomark and the terms under which you may use it.

> **Attention:** The PaperMC logo is subject to its own seperate licensing terms and does not inherit any from the projects it represents.

You may:

- Use the PaperMC logomark to represent the project in blogposts and other places in order to bring attention to the project.
- Use the PaperMC logomark to represent Paper-Server in downloads, server selectors, and similar places.
- Crop out extra transparent canvas space behind the PaperMC logomark so it fits better next to other content.

You may not:

- Alter any of the colors used in the PaperMC logomark.
- Change the dimensions of the PaperMC logomark.
- Create modified versions of the PaperMC logomark or derivative works of it.
- Add your own project images or branding to the PaperMC logomark.
- Claim the logomark as your own work or use it as a representation of your own projects.
- Sell the PaperMC logomark on its own or as part of other products without explicit permission.
- Alter the transparency of any of the elements in the PaperMC logomark

> **Note:** The logomark is available in higher resolution and vector formats. If you need a higher resolution version please contact us.

The official PaperMC logomark

Contact Us

- *IRC*
- *Discord*
- *Twitter*

IRC

The Paper project handles most of our communication via IRC.

You can join us on either of these two servers:

- #paper on irc.spi.gt

- #paper on irc.esper.net

Discord

Our IRC presence is also bridged to a Discord server, you can use the following link to join it.

- https://discord.gg/jETyjUw

Twitter

We often tweet out version release notes, update notices, and other information via our Twitter page.

- @PaperPowered

You should not DM or @ this account for support. It is not checked as regularly as the above locations.

Useful Links

- 1.12 Latest Downloads
- 1.13 Latest Downloads
- Discord
- IRC
- Docs
- GitHub
- Forums
- Javadocs
- Issue Tracker
- Organization GitHub

CHAPTER 3

Acknowledgements