

Commands

Overview

As a staff member, you have access to **many of Minecraft's wide variety of commands**. Commands are powerful tools that staff members of any Minecraft server should become familiar with, as they make up a lot of your administrative work in game. This page will discuss many vital parts of **how to write commands, describe some of the most important commands to know on WOFTNW, and provide some examples on how to write them**.

Usage Notation


The command usage is notated like so:

```
/command subcommand <required_argument> [optional_argument] [greedy_optional_argument]...
```

- `command` is the name of the command.
- `subcommand` is the name of the subcommand if relevant.
- `<required_argument>` is an argument that must be included to execute the command.
- `[optional_argument]` is an argument that may optionally be included.
- `<greedy_required_argument>...` or `[greedy_optional_argument]...` are arguments which permit spaces. If present, they will always be at the end of a command.

Vanilla Commands

Minecraft's vanilla commands are plentiful, but **most are either not important or are simple to understand**. The following is a complete list of the vanilla commands for our server version; **you do not have to be familiar with every command listed!** If you are curious about any specific command, please visit the wiki via the link in the description of the command.

Command	Description
<code>/advancement</code>	   Grants or revokes an advancement. Wiki
<code>/attribute</code>	  Queries, adds, removes or sets an entity attribute. Wiki

Command	Description
<code>/ban</code>	<code><player> <reason></code> Adds a player to the banlist. Wiki
<code>/banlist</code>	<code><limit></code> Displays the banlist. Wiki
<code>/bossbar</code>	<code><name> <color> <display> <max health> <progress></code> Creates and modifies bossbars. Wiki
<code>/clear</code>	<code><player> <item> <count></code> Clears items from player inventory. Wiki
<code>/clone</code>	<code><from x y z> <to x y z> <mode></code> Copies blocks from one place to another. Wiki
<code>/damage</code>	<code><player> <amount></code> Applies damage to specified entities. Wiki
<code>/data</code>	<code><player> <block> <entity> <storage></code> Manipulates block, entity, and storage NBT data. Wiki
<code>/datapack</code>	<code><enable> <disable> <reload></code> Controls the data packs interface. Wiki
<code>/debug</code>	<code><start> <stop></code> Starts or stops a debugging session. Wiki
<code>/defaultgamemode</code>	<code><gamemode></code> Sets the default game mode. Wiki
<code>/deop</code>	<code><player></code> Revokes operator status from a player. Wiki
<code>/difficulty</code>	<code><player> <difficulty></code> Sets the difficulty level. Wiki
<code>/effect</code>	<code><player> <effect> <duration> <amplifier></code> Adds or removes status effects. Wiki
<code>/enchant</code>	<code><player> <item> <enchantment> <level></code> Adds an enchantment to a player's selected item. Wiki
<code>/execute</code>	<code><player> <command></code> Executes another command. Wiki
<code>/experience, /xp</code>	<code><player> <amount></code> Adds or removes player experience. Wiki
<code>/fill</code>	<code><from x y z> <to x y z> <block></code> Fills a region with a specific block. Wiki
<code>/fillbiome</code>	<code><from x y z> <to x y z> <biome></code> Sets a region with a specific biome. Wiki
<code>/forceload</code>	<code><player> <chunk x z></code> Forces chunks to constantly be loaded or not. Wiki
<code>/function</code>	<code><player> <function name></code> Runs a function. Wiki
<code>/gamemode</code>	<code><player> <gamemode></code> Sets a player's game mode. Wiki
<code>/gamerule</code>	<code><player> <rule> <value></code> Sets or queries a game rule value. Wiki
<code>/give</code>	<code><player> <item> <count></code> Gives an item to a player. Wiki
<code>/help</code>	<code><command></code> Provides a list of commands or help for a command. Wiki
<code>/item</code>	<code><player> <item> <count></code> Manipulates items in inventories. Wiki

Command	Description
<code>/jfr</code>	<code>start</code> <code>stop</code> Start or stop a JFR profiler. Wiki
<code>/kick</code>	<code>player</code> Kicks a player off a server. Wiki
<code>/kill</code>	<code>entity</code> Kills entities (players, mobs, items, etc.). Wiki
<code>/list</code>	<code>players</code> Lists players on the server. Wiki
<code>/locate</code>	<code>structure</code> <code>biome</code> <code>poi</code> Locates closest structure, biome, or point of interest. Wiki
<code>/loot</code>	<code>table</code> Give a loot table to a selector or spawn a loot table. Wiki
<code>/me</code>	<code>message</code> Displays a message about the sender. Wiki
<code>/msg</code> , <code>/tell</code> , <code>/w</code>	<code>player</code> Displays a private message to other players. Wiki
<code>/op</code>	<code>player</code> Grants operator status to a player. Wiki
<code>/pardon</code>	<code>player</code> Removes players from the banlist. Wiki
<code>/pardon-ip</code>	<code>ip</code> Removes an IP address from the banlist. Wiki
<code>/particle</code>	<code>particle</code> <code>x</code> <code>y</code> <code>z</code> Generates particles. Wiki
<code>/perf</code>	<code>start</code> Captures info and metrics about the game for 10 seconds. Wiki
<code>/place</code>	<code>feature</code> <code>jigsaw</code> <code>template</code> Used to place a configured feature, jigsaw, template, or structure at a given location. Wiki
<code>/playsound</code>	<code>entity</code> Plays a sound. Wiki
<code>/publish</code>	<code>world</code> Opens a single-player world to a local network. Wiki
<code>/recipe</code>	<code>player</code> Gives or takes player recipes. Wiki
<code>/reload</code>	<code>loot</code> <code>advancements</code> <code>functions</code> Reloads loot tables, advancements, and functions from disk. Wiki
<code>/return</code>	<code>function</code> <code>label</code> <code>goto</code> Control execution flow inside functions. Wiki
<code>/ride</code>	<code>entity</code> Used to make entities ride other entities, stop entities from riding, make rides evict their riders, or summon rides or riders. Wiki
<code>/save-all</code>	<code>server</code> Saves the server to disk. Wiki
<code>/save-off</code>	<code>server</code> Disables automatic server saves. Wiki
<code>/save-on</code>	<code>server</code> Enables automatic server saves. Wiki

Command	Description
<code>/say</code>	<code><entity></code> Displays a message to multiple players. Wiki
<code>/schedule</code>	<code><entity></code> <code><function></code> <code><seconds></code> Schedule the execution of a function. Wiki
<code>/scoreboard</code>	<code><entity></code> <code><objective></code> Manages scoreboard objectives and players. Wiki
<code>/seed</code>	<code><entity></code> <code><world></code> Displays the world seed. Wiki
<code>/setblock</code>	<code><entity></code> <code><block></code> Changes a block to another block. Wiki
<code>/setidletimeout</code>	<code><entity></code> <code><seconds></code> Sets the time before idle players are kicked. Wiki
<code>/setworldspawn</code>	<code><entity></code> <code><world></code> Sets the world spawn. Wiki
<code>/spawnpoint</code>	<code><entity></code> <code><world></code> <code><location></code> Sets the spawn point for a player. Wiki
<code>/spectate</code>	<code><entity></code> <code><targetEntity></code> Force a spectating player to spectate an entity. Wiki
<code>/spreadplayers</code>	<code><entity></code> <code><radius></code> <code><count></code> Teleports entities to random locations. Wiki
<code>/stop</code>	<code><entity></code> <code><reason></code> Stops the server. Wiki
<code>/stopsound</code>	<code><entity></code> <code><soundName></code> Stops a sound currently playing. Wiki
<code>/summon</code>	<code><entity></code> <code><entityName></code> <code><location></code> Summons an entity. Wiki
<code>/tag</code>	<code><entity></code> <code><tag></code> Manages entity tags. Wiki
<code>/team</code>	<code><entity></code> <code><teamName></code> Controls teams. Wiki
<code>/teammsg, /tm</code>	<code><entity></code> Sends a message to the player's team. Wiki
<code>/teleport, /tp</code>	<code><entity></code> <code><targetEntity></code> <code><location></code> Teleports entities. Wiki
<code>/tellraw</code>	<code><entity></code> <code><message></code> Displays a JSON-formatted message to players. Wiki
<code>/time</code>	<code><entity></code> <code><time></code> Changes or queries the world's game time. Wiki
<code>/title</code>	<code><entity></code> <code><titleName></code> Manages screen titles. Wiki
<code>/trigger</code>	<code><entity></code> Manages trigger scoreboards. Wiki
<code>/weather</code>	<code><entity></code> <code><weatherType></code> Sets the weather. Wiki
<code>/whitelist</code>	<code><entity></code> <code><playerName></code> <code><action></code> Manages the server whitelist. Wiki
<code>/worldborder</code>	<code><entity></code> <code><radius></code> Manages the world border. Wiki

Targeting Entities

Target Selectors

Overview

Target selectors are used in many commands in place of usernames or UUIDs to **target multiple entities that match a broad criteria**.

Target Selector	Entities Selected
@p	Nearest player
@r	One random player
@a	All players, living and dead
@e	All loaded, living entities
@s	Entity executing the current command

Tips

- **Use the most specific selector available.** For example, if you want to target the nearest player, use `@p` instead of `@a[sort=nearest, limit=1]`. This is because some selectors such as `@e` will create a massive list of entities, which can unintentionally cause lag.

Selector Arguments

Overview

Selector arguments can be used to filter entities from the selection even further. These allow you to narrow your selection based on various attributes. There are a wide variety of arguments, but those detailed below are the most important.

Target Selector Argument	Filter By...
x, y, z	<u>coordinate</u>
distance	<u>distance</u>
dx, dy, dz	<u>volume dimensions</u>
x_rotation	<u>vertical rotation (pitch)</u>

Target Selector Argument	Filter By...
y_rotation	<u>horizontal rotation (yaw)</u>
scores	<u>scores</u>
tag	<u>tag</u>
team	<u>team</u>
name	<u>entity name</u>
type	<u>entity type</u>
predicate	<u>predicate</u>
nbt	<u>nbt</u>
level	<u>experience level</u>
gamemode	<u>gamemode</u>
advancements	<u>advancements</u>
limit, sort	<u>selector limiting</u>

distance=

`distance` allows you to target entities within a specified *spherical range* in blocks. The entity will only be targeted if the origin of their hitbox (bottom center) is within the distance. This selector also limits the selection to the current dimension.

@e[distance=1.5]	Target all entities exactly 1.5 blocks away.
@a[distance=8..16]	Target all players 8 to 16 blocks away, inclusive .
@p[distance=..40]	Target the nearest player up to 40 blocks away, inclusive . Note that if the first number is excluded, the command interprets it as 0.
@r[distance=12..]	Target a random player at least 12 blocks away, inclusive . Note that if the second number is excluded, the command will interpret it as infinity, which will include all entities in the dimension.

dx=, dy=, dz=

`dx`, `dy`, and `dz` select by rectangular volume from the command origin. These specify the distance from the origin rather than a global coordinate and extend asymmetrically. Unlike `distance`, volume selection targets entities if any part of their hitbox is inside the selection.

@a[dx=3, dy=5, dz=2]	Target all players whose hitbox lies within the volume extending 4 blocks east, 6 blocks up, and 3 blocks south of the execution spot.
----------------------	---

scores=

`scores` selects by matching scoreboard objectives. Targeted entities must have all specified scores on all specified objectives. You also can specify an integer range (inclusive) similar to the `distance` selector.

@e[scores={myScore1=4}]	Target all entities with a score of exactly 4 in myScore1 .
@p[scores={myScore2=..5, myScore3=10..}]	Target the nearest player with at most a score of 5 in myScore2 and at least a score of 10 in myScore3 .

tag=

`tag` targets entities with a specified tag. These tags can be applied with the `/tag` command.

@a[tag=myTag1]	Target all players with the myTag1 tag .
@p[tag=!myTag2]	Target the nearest player who does not have the myTag2 tag .
@e[tag=myTag3, tag=myTag4, tag=!myTag5]	Target all entities who have the myTag3 tag, and the myTag4 tag, and do not have the myTag5 tag .
@r[tag=]	Target a random player that has no tags .
@s[tag=!]	Target the current executing entity if they have at least one tag .

team=

`team` targets entities who are on a specified team, and works in nearly the exact same way as `tag`. Note that using this argument will only target living entities.

@a[team=myTeam1]	Target all players who are in myTeam1 . Note that, in this instance, this argument can only be stated once since entities can only belong to one team.
@p[team=!myTeam2, team=!myTeam3]	Target all players who are not in myTeam2 or myTeam3 . Note that this argument <i>can</i> be used more than once.
@r[team=]	Target a random player who is not on a team .
@a[team=!]	Target all players who are on any team .

nbt=

`nbt` targets entities with specified NBT data, and as such is the most resource-intensive arguments. **Avoid using NBT unless ABSOLUTELY necessary as misusing it has a high potential to cause lag.**

<code>@a[nbt={Inventory: [{id: "minecraft:iron_sword"}]}]</code>	Target all players who have an iron sword in their inventory .
<code>@e[nbt={DeathLootTable: "myLootTable"}]</code>	Target all entities who have the myLootTable death loot table .

limit=

`limit` can be used to limit the number of targeted entities.

Note that `limit` **and** `sort` **are most commonly used together**, as they can significantly improve the performance of selectors.

<code>@e[limit=15]</code>	Target at most 15 entities .
<code>@a[limit=4]</code>	Target at most 4 players .

sort=

Finally, `sort` can be used to sort through the currently selected entities. `sort` has four possible values: `arbitrary`, `nearest`, `furthest`, and `random`. `arbitrary` is the default, selecting sorting entities based off of their time of creation. `nearest` and `furthest` sort by increasing and decreasing distance respectively, and `random` is... well, random.

Note that `limit` **and** `sort` **are most commonly used together**, as they can significantly improve the performance of selectors.

<code>@e[sort=nearest]</code>	Target all entities, selecting the nearest entities first .
<code>@a[limit=4, sort=random]</code>	Target at most 4 random players .

Tips

- **Selector arguments — or lack thereof — can make commands dangerous!** When you run certain commands, always remember that selectors are made to target *everything* that matches the criteria. Ensure your arguments are specific and correct, and run them by a more experienced staff member if you are unsure.

- **x, y, z can almost always be replaced by** `execute positioned <x> <y> <z>`. You will rarely run into an instance where the former is necessary.
- **Use selector arguments to optimize your selector. Order matters!** The following list is the recommended order of selector arguments. **At the very least, specify type and limit**, but use all that apply:

- `sort=`
- `limit=`
- `type=`
- `gamemode=`
- `gamemode=!`
- `distance=`
- `x_rotation=` / `y_rotation=`
- `x, y, z` / `dx, dy, dz`
- `level=`
- `team=`
- `team=!`
- `type=!`
- `tag=`
- `tag=!`
- `name=`
- `name=!`
- `predicate=`
- `predicate=!`
- `scores=`
- `advancements=`
- `nbt=`
- `nbt=!`

Important Commands

The following commands listed are important to know during your time as a staff member. Here is a quick guide to understanding the command layout:

- `text` - Text that should be written how it is shown.
- `<keyword>` - A specific option from a list of valid entries.
- `[keyword]` - An optional entry.
- `(option1 | option2)` - A required "or", meaning you *must* use one field or another.
- `[option1 | option2]` - An optional "or", meaning you *may* use one field or another if you wish.

`/ban, ban-ip`

Bans a player or ip from the server. The player does not have to be online to be banned. In addition to being added to the banlist, they will be kicked automatically.

<code>/ban ban-ip <player> [reason]</code>	Syntax for the command.
<code>ban jjj_North</code>	Kick and ban jjj_North.
<code>ban-ip 7pencilsharpen3r Hacking</code>	Kick and ban 7pencilsharpen3r's IP for hacking, preventing any account from joining from that IP.

/clear

Clears an item or all items from a player or valid selector entities.

<code>/clear <selector> [item] [amount]</code>	Syntax for the command.
<code>clear mmaurice196</code>	Clear every item from mmaurice196.
<code>clear @a[team=SkyWing] cooked_beef 12</code>	Clear 12 steak from every player on the SkyWing team.
<code>clear CalamityLegendz diamond_sword{Enchantments: [{id: "minecraft:sharpness", lvl: 1000s}]}</code>	Clear all diamond swords with Sharpness 1000 from CalamityLegendz.
<code>clear @s iron_sword{titanium: 1b} 0</code>	<p>Check if the executing entity has an iron sword with the NBT "titanium:1b".</p> <p><i>By setting the quantity to 0, we can instead check if the selected entities have the item declared. This is mainly only useful in data packs, but you may find certain use cases for this feature.</i></p>

/datapack

Enable or disable a data pack or list all data packs.

<code>/datapack <(disable enable) <name> list></code>	Syntax for the command.
<code>datapack disable "file/Ender Dragon"</code>	Disable the "Ender Dragon" data pack.
<code>datapack enable "file/Example Datapack"</code>	Load the "Example Datapack" data pack.
<code>datapack list</code>	List all data packs.

/execute

Execute is an extremely powerful command that allows you to execute multiple subcommands to modify given parameters, check for different variables, and fork the command execution. While you do not need to know every in and out of this command, there may be cases in which you will utilise it, so it is important to know the basics!

- `align <axes>` - aligns the position of the command execution with the block position along the given axes. This allows you to align to the edges of a block.
- `anchored <anchor>` - sets the position of the command execution to either the eyes or feet of the executor.
- `as <targets>` - sets the executor without changing the location. You must use `at` to set the location. If multiple targets are selected, the following subcommands will apply to all targets
- `at <targets>` - sets the location of the execution without changing the executor. You must use `as` to set the executor.
- `facing <pos | <targets> <anchor>>` - sets the execution rotation.
- `in <dimension>` - sets the execution dimension. This will account for distance scaling in the nether and custom dimensions.
- `on <attacker | controller | leasher | origin | owner | passengers | target | vehicle>` - sets the executor based on relation to the current executor.
- `positioned <pos | as | over>` - sets the execution position.
- `rotated <rot | as <targets>>` - sets the execution rotation.
- `summon <entity>` - summons a new entity at the execution position and sets it as the executor.
- `(if|unless) biome <pos> <biome>` - checks for the given biome at a given position.
- `(if|unless) block <pos> <block>` - checks for the given block at a given position.
- `(if|unless) blocks <start> <end> <destination> (all|masked)` - compares blocks in two volumes.
- `(if|unless) data <block | entity | storage> <source> <path>` - checks whether the targeted block, entity or storage has any data tag for a given path.
- `(if|unless) dimension <dimension>` - checks if the execution is in a matching dimension.
- `(if|unless) function <function>` - checks a function or function tag and matches the return value(s). If a tag is given, all functions run regardless of the results of prior functions.
- `(if|unless) entity <entities>` - checks whether one or more entities exist.
- `(if|unless) loaded <pos>` - checks if chunks at a given position is fully loaded.
- `(if|unless) predicate <predicate>` - checks whether the predicate successes. This is primarily a data pack feature.
- `(if|unless) score <target> <targetObjective> <operation | matches <target> <targetObjective>>` - checks whether a score has the specific relation to another score, or whether it is in a given range.
- `store (result|success) <block | bossbar | entity | score | storage>` - stores either the result (i.e. save the number of items in a chest to a scoreboard objective) or the success (i.e. checking if a chest has a specific item in it) of the following command.

Minecraft will process subcommands in the order they are arranged in the command. Note that `execute ... run execute ... run ...` has no effect as it is essentially the same as `execute ... run`.

<code>/execute <subcommands> [run <command>]</code>	Syntax for the command.
<code>execute run say Hello, world!</code>	The same thing as simply using <code>/say Hello, world!</code> Not very practical...

<code>execute as @e[type=pig] run say I'm a pig!</code>	Change the executing entity to all pigs... ..then force them to use <code>"/say I'm a pig!"</code> . Nice!
<code>execute in the_end if block 0 64 0 beacon as @e[type=armor_stand, distance=0..] at @s facing 0 64 0 run tp ^ ^ ^1</code>	Change the command location to The End... ..then check if there is a beacon at 0 64 0... (if there is no beacon at 0 64 0, the command will stop here entirely!) ...then change the executing entity to all armor stands in the current dimension... ..then change the location of the command execution to the executing entity — all armor stands in the current dimension... ..then change the direction the command is running to face 0 64 0... ..then teleport the executing entities forwards one block. Phew.

/give

Gives the targeted selectors an item with optional NBT attached.

<code>/give <selector> <item> [nbt]</code>	Syntax for the command.
<code>give BogTheMudWing potato</code>	Give BogTheMudWing 1 potato.
<code>give @a[limit=10, sort=random] diamond{display: {Name: "Actual Coal"}}</code>	Give 10 random players 1 diamond named "Actual Coal".

/kick

Terminates a player's connection to the server.

<code>/kick <player> [reason]</code>	Syntax for the command.
<code>kick FlawLessII</code>	Kick FlawLessII from the server.
<code>kick s0meb0dysupply Profanity</code>	Kick s0meb0dysupply from the server for profanity.

/pardon, pardon-ip

Remove a player or an IP from the banlist, regranting said player or IP access to the server.

<code>/pardon pardon-ip <player></code>	Syntax for the command.
<code>pardon jjj_North</code>	Pardon jjj_North.
<code>pardon-ip 7pencilsharpen3r</code>	Pardon 7pencilsharpen3r's IP.

/scoreboard

Add objectives, get player scores, manipulate scoreboard objectives, and much more. This command is **very** extensive and powerful!

<code>/scoreboard <objective players> ...</code>	Syntax for the command.
<code>scoreboard objectives add player_health health</code>	Create a scoreboard objective called "player_health" that tracks... the players' health.
<code>scoreboard objectives modify player_health displayname {"text": "Player Health", "color": "red"}</code>	Modify the "player_health" scoreboard to appear as "Player Health" in red.
<code>scoreboard players get Julibeu WhiteKills</code>	Get Julibeu's score in the objective WhiteKills.

/summon

Summons an entity.

<code>/summon <entity> [<posX posY posZ> [NBT]]</code>	Syntax for the command.
<code>summon ender_dragon</code>	Summons an Ender Dragon.
<code>summon iron_golem ~ ~ ~ {NoAI: 1b}</code>	Summons an iron golem at the current position with no AI.
<code>summon cow ^ ^ ^10 {active_effects: [{id: "minecraft:levitation", amplifier: 10, duration: 200}], DeathLootTable: "chests/abandoned_mineshaft"}</code>	Summons a cow 10 blocks forwards from the player's current direction, with Levitation 11 for 10 seconds, that will drop loot from an abandoned mineshaft... wow.

/tellraw

Announce a highly-customizable message to a selector using JSON.

<code>/tellraw <selector> [string or JSON]</code>	Syntax for the command.
<code>tellraw FrazzledDingo "Welcome to the server!"</code>	Show the message "Welcome to the server!" to FrazzledDingo.
<code>tellraw @a[team=Green] {"text": "You are on the Green team!", "color": "dark_green"}</code>	Show the message "Welcome to the server!" to all players on the Green team.
<code>tellraw @a[distance=..30] [{"selector": "@r[distance=..30]", "color": "red"}, {"text": " won the ", "color": "gray"}, {"text": "Grand Raffle", "color": "gold"}, {"text": "!", "color": "gray"}]</code>	Show the message "<random player up to 30 blocks away> won the Grand Raffle!" to all players up to 30 blocks away.

/whitelist

Add or remove a player from the whitelist, enable or disable the whitelist, or reload the whitelist.

<code>/whitelist <add on off reload remove> <player></code>	Syntax for the command.
---	-------------------------

<code>whitelist add RaspberryConflict</code>	Adds RaspberryConflict to the whitelist.
<code>whitelist remove known__clams</code>	Removes known__clams from the whitelist.
<code>whitelist on</code>	Activates the whitelist, allowing only those on the whitelist to join the server.
<code>whitelist off</code>	Deactivates the whitelist, allowing any player to join the server.
<code>whitelist reload</code>	Reloads the whitelist.

Revision #6

Created 29 January 2025 01:21:20 by Bog

Updated 25 February 2025 00:26:48 by Bog