Fuji Manual

v5.0.0-f1ea2d98ba-mc1.21.2

sakurawald

October 30, 2024

Contents

List of commands

Chapter 1

Concept

1.1 Configuration File

1.1.1 Definition

All `files` inside `config/fuji` directory are named `configuration file`

1.1.2 Types

Note: The types of configuration files

1. **Control File** A `control file` is used to control behaviours.
    - 1.1. **Main-Control File** The **main-control file** refers to the **config/fuji/config.json** file, which is used to enable/disable a module.
    - 1.2. **Module-Control File** Some modules will have their own control file, which is used to control the behaviour of the module.
2. **User-Data File** User-data files are used to store the data generated by the user.

1.2 Module

1.2.1 Definition

A `module` is used to provide a specific purpose.

Example: The purpose of modules

1. **ChatModule** provides chat-format customization.
2. **TpaModule** provides `/tpa` command.

1.2.2 Properties

The properties of a module are as follows:

1. **Can be disabled** You can disable a module completely in `main-control file` by setting the value of its `enable` key to `false`.
2. **Can work standalone** The code of a module is self-contained, there is no symbol reference to other modules.

1.2.3 Module Path

A `module` is identified by a unique `module path`.

Example: What a module-path looks like?

The module path of the module `tpa` is `tpa`.
The module path of the module `history` whose parent module is `chat`, is `chat.history`.
You will see a list of `enabled modules` identified by their `module path` at the server-startup process.

A `module` can have `sub-module`.

The relationship between `parent-module` and `sub-module` is relative, and there is nothing special about `sub-module`.

1.2.4 How to enable/disable a module?

You can enable/disable a module in `config/fuji/config.json` by setting the value of its `enable` key to `true`/`false`.

A `module` will be enabled if the following conditions are met:

1. The value of `common.debug.disable_all_modules` is `false`.
2. The required dependency mods are installed.
3. Its `parnet-module` is `enabled`.
4. The value of its `enable` key is `true`.

Example: How to enable a sub-module?

To make the module `chat.display` enabled, you need to enable `chat` module first.

1.3 Job

1.3.1 Definition

A `job` is some things will be done `repeatedly`.

1.3.2 Cron Expression as Trigger Rule

A language named `cron language` is used to `define` when a `job` should be triggered.

Tip: Don't write cron expression by hand. Use generator!

A `cron expression` looks like `0 * * ? * *`, means `trigger the job every minute`.
You can use the generator to generate a `cron expression`: https://www.freeformatter.com/cron-expression-generator-quartz.html

1.4 Regex

1.4.1 Definition

Regex is a language used to define the pattern of strings.

Some modules that use `regex`:
See command_warmup
See command_rewrite
See command_cooldown
See teleport_warmup
See cleaner

1.4.2 Reference

1. https://regexr.com/
2. https://regex101.com/

1.5 Placeholder

1.5.1 Definition

A `placeholder` is a string which will be replaced based on context.

Tip: What is the placeholder api in fabric platform?

There is a plugin named `PlaceholderAPI` in bukkit platform.
Also, there is a mod named `Text Placeholder API` in fabric platform.
They are different projects, but provides the same purpose.

1.5.2 Example

Example: Replace player name by context

The placeholder `%player:name%` will be replaced by the name of the contextual player.

1.5.3 Reference

1. https://placeholders.pb4.eu/user/default-placeholders/
2. https://placeholders.pb4.eu/user/mod-placeholders/

1.6 Identifier

1.6.1 Definition

An `identifier` is a `string` to name an `object` in `minecraft`.

1.6.2 Example

Example: The identifier of items

The identifier of apple is `"minecraft:apple"`.
The identifier of diamond is `"minecraft:diamond"`.

Example: The identifier of entities

The identifier of zombie is `"minecraft:zombie"`.

1.6.3 How to query all identifiers of a type

Example: Query all identifiers of entity

```
/summon ...
```

Example: Query all identifiers of block

```
/setblock ...
```

Example: Query all identifiers of item

```
/give ...
```

Chapter 2

Permission

2.1 Definition

A `permission` is used to decide whether `a player` can do something or not.

2.2 Types

To make the discussion clearer, we define the types of `permission` as follows:

1. **level permission** A permission level is a non-negative number used in vanilla minecraft. The higher number means the higher authority.
2. **string permission** Usually, a `string permission` is introduced by a `permission plugin`, such as `luckperms`.

2.3 What is the permission system used by fuji?

2.3.1 Explanation

Fuji use the mojang's `vanilla permission system`, which is based on `level permission`.

As a convention, most of the commands registered by fuji, requires `level permission` to be `0` to use. Only a few of the commands require the `level permission` to be `4` to use.

Tip: Modify the default requirement of all commands registered by fuji

See Configuration

2.3.2 Set a string permission for a command

By default, fuji only use the `level permission` as `the requirement of` a command. However, if you want to use `string permission` for a `command`, you can use command_permission module, which is used to `override` the requirement of an existing command.

Example: Allow players to use /seed command

The command `/seed` provided by mojang requires `level permission` to be `3` to use. If you want to allow players to use `/seed` command, but you don't want to grant `op` for them. Then in this situation, you can grant the `string permission` for them: `/lp group default permission set fuji.permission.seed true`, which means that: set the requirement of command `/seed` to string permission `fuji.permission.seed`.

Example: Dis-allow players to use /list command

The command `/list` provided by mojang required `level permission` to be `0` to use. If you want to dis-allow players to use `/list` command, but because this command requires no `string permission` to use, so it's impossible to ban it via `luckperms`. In this situation, you can grant a `string permission`: `/lp group default permission set fuji.permission.list false` for them, which means that: set the requirement of command `/list` to string permission `fuji.permission.list`.

Example: Unset the override of requirement of the command

To undo the operation in `the first example`, just issue `/lp group default permission unset fuji.permission.seed`

2.4 Advanced Usage

2.4.1 The string permission is calculated with a context

The luckperms mod provides the `context` to calculate a string permission. A context is a predicate which accepts a player, and output a boolean value according to its conditions. Luckperms pre-define some useful context like:

1. **world context** : the player should in a specified world, so that this string permission is valid.
2. **temporal context** : current time should not exceed the expiration time, so that this string permission is valid.

2.5 Reference

1. https://minecraft.fandom.com/wiki/Permission_level

Chapter 3

Meta

3.1 Definition

A `meta` is a `key-value pair`.

Note:

Note that `meta` is introduced by `luckperms` mod, there is no `meta` in vanilla minecraft.

3.2 Example

Example: Set a meta for a gruop.

The `home module` supports the meta `fuji.home.home_limit`, which controls how many homes a player can create. To set the max homes limits to `3`: `/lp group default meta set fuji.home.home_limit 3`

Example: Query all metas for a group.

```
/lp group default info
```

Chapter 4

Configuration

4.1 Main-Control File

4.1.1 Configuration

core The `core` options inside `config/fuji/config.json` will influence `all modules`.

debug

disable_all_modules Used to test the compatibility between fuji and other mods.

log_debug_messages Whether to log the debug level messages into the console. Use this only for debug purpose, or it will cause console spam.

backup Fuji will back up the `config/fuji` directory automatically before it loads any module.

max_slots How many `backup` should we keep?

skip The list of `path resolver` to skip in backup.
Insert `modules/head` means skip the folder `config/fuji/modules/head`.

language

default_language The default language to use.

> **Tip: Enable multi-language support for fuji**
>
> See language

permission

all_commands_require_level_4_permission_to_use_by_default By default, most of commands registered by fuji requires level 0 permission to use. (e.g. /afk, /home, /warp). Some commands requires level 4 to use. (e.g. /fly, /god, /more). Enable this option will cause all commands registered by fuji require level 4 permission to use.

quartz Fuji use `quartz` library as scheduler, all the Job are managed by quartz. Quartz library use a language called `cron language` to define when to trigger a job.

logger_level The logger level for `quartz`. The logger level from high to low are: OFF, FATAL, ERROR, WARN, INFO, DEBUG, TRACE, ALL.

Example: Enable all logs for quartz

Set the value to `ALL` to display all the messages from quartz. It's recommended to set at least `WARN` level, to avoid `console spam`.

4.2 Module-Control File

You can read more about `module-control file` for each `module` in Module

Tip: Use `/fuji reload` command to hot reload the configuration files.

The `/fuji reload` command is provided by `fuji module`. See fuji

Chapter 5

Module

5.1 afk

5.1.1 Purpose

This module provides afk detection, afk event, afk name customization and afk effects.

5.1.2 How it works?

For each player, we have an `input counter` to track the last input time.
An input is as allows: mine blocks, movement.
The `afk checker` defined by cron will run and compare 2 consecutive value of the `input counter` associated with the player, if it's the same, then the player is considered as in afk.
In other words, if the `afk checker` is defined to run every 5 minutes, a player flagged as in afk is actually no input for 5--10 minutes.

5.1.3 Command

- `/afk`

5.1.4 Configuration

format The **tab list name** format when a player is afk

afk_checker

cron The cron to define how the afk_checker is triggered.

afk_event Execute commands on afk events.

Example: Kick a player if he enters afk state

```
"on_enter_afk": [
"send-broadcast <gold>Player %player:name% is now afk",
"kick %player:name% You are kicked beacause of afk."
]
```

event Afk effects are applied if a player enters afk state.

invulnerable Immune to all damage?

targetable Can be targeted by a hostile entity?

moveable Can be moved if in afk state?

5.2 anti_build

5.2.1 Purpose

This module allows you to ban the interaction with some item/block/entity.

5.2.2 Anti Types

The types supported by this module are as follows:

1. **break_block**
2. **place_block**
3. **interact_item**
4. **interact_block**
5. **interact_entity**

Tip: Query identifiers

See How to query all identifiers of a type

5.2.3 Exapmle

Example: Ban TNT

add minecraft:tnt into place_block list

Example: Ban TNT but allow a specific player to use

```
/lp user <player> permission set fuji.anti_build.place_block.bypass.minecraft:tnt
```

5.3 back

5.3.1 Purpose

This module allows player to teleport back to **last teleport point** or **death point**.

5.3.2 Command

- `/back`

5.3.3 Configuration

ignore_distance If the player's teleportation destination is close enough, we ignore this teleportation.

5.4 chat

5.4.1 Purpose

This module provides chat related customization.

Note: The compatibility with styled-chat mod and other chat-related mods

All submodules except the `chat.style` module are designed to work with `styled chat mod` and other `chat-related mods`. You are free to use these modules with them!

5.4.2 Reference

1. Text Placeholder API - default placeholders
2. Luckperms - prefix, suffix and meta

5.4.3 Sub-Module

5.4.3.1 style

5.4.3.1.1 Purpose

Customize global chat style, and also allow players to use `/chat style` to set their per-player chat style.

5.4.3.1.2 Feature

1. You can use `mini-message language` to define complex format.
2. Besides the `server chat format`, each player can also set their `per-player chat format`.
3. This module doesn't **break** the vanilla chat events, so it can work with other chat related mods.

Tip: Write complex style using mini-language

You can use `mini-language` to write complex text.

See more:

1. https://docs.advntr.dev/minimessage/format.html
2. https://placeholders.pb4.eu/user/quicktext

5.4.3.1.3 Configuration

`style` The style for chat message.

5.4.3.1.4 Command

- `/chat style`

Example: Set prefix and suffix for players

Luckperms is required to set `prefix` and `suffix`.

After you installed `luckperms` mod, just issue `/lp group default meta setprefix <yellow>[awesome]` to assign prefix.
Don't forget to insert `%fuji:player_prefix%` and `%fuji:player_suffix%` in `sender` option in configuration file, and issue `/fuji reload`

5.4.3.1.5 Example

Example: Set per-player chat style

```
/chat style set prefix + %message% + suffix
```

Example: Reset per-player chat style

```
/chat style reset
```

Example: Use chat stripe module to control the style tags usage

By default, the `chat style` module allows any player to use any `style tags` in chat message, including `"<click>"` tag.
See stripe

5.4.3.2 display

5.4.3.2.1 Purpose

Allow players to share their item in hand, items in inventory and items in ender chest.

5.4.3.2.2 Placeholder

1. `%fuji:item%` display player's main-hand item.
2. `%fuji:inv%` display player's inventory.
3. `%fuji:ender%` display player's ender-chest.

5.4.3.2.3 Configuration

expiration_duration_s For each display data, how long should we save in the memory. Note that if a player shares its inventory items, then fuji will save a copy of his inventory data in the memory.

replace_pattern Define the `patterns` to be replaced in chat message. By default, you can insert `"[item]"`, `"[inv]"` and `"[ender]"` to create displays.

5.4.3.2.4 Example

Example: Use chat rewrite module to create a shortcut

See rewrite

5.4.3.3 mention

5.4.3.3.1 Purpose

Insert the player name in the chat message, and the target player will be mentioned and sound notified.

5.4.3.3.2 Configuration

mention_player If you insert **Steve** in chat message, then the player named Steve will get audio mention.

sound The type of **sound** used to notify the mentioned player.

> **Tip: Query all identifiers of sound**
>
> `/playsound ...`

volume

pitch

repeat_count The sound repeat count.

interval_ms The interval between each repeat.

5.4.3.4 rewrite

5.4.3.4.1 Purpose

Allow you to define `regex` to replace `chat message` sent by players.

5.4.3.4.2 Example

Example: Rewrite the "item" into "[item]" as a shortcut.

```
{
    "regex": "(?<=^|\\s)item(?=\\s|$)",
    "replacement": "[item]"
}
```

Example: Rewrite badwords as a message filter.

```
{
    "regex": "(bad-word)|(bad-word-2)|(bad-word-3)",
    "replacement": "<st>***</st>"
}
```

5.4.3.5 replace

5.4.3.5.1 Purpose

Allow you to define `regex` to replace `chat text` with a given text before it's sent to the player.

Note: The difference with chat rewrite module

The `chat rewrite` module operates on `string level`, while `chat replace` module operates on `text level`. The chat rewrite module rewrites the chat string received from the client, and pass this string to the chat string processor, the chat string processor will yield the final chat text, which is ready to sent to the receiver player. This module will replace the final chat text using the defined rules.

5.4.3.5.2 Example

Example: Define a shortcut to evaluate a placeholder

```
{
    "regex": "(?<=^|\\s)uuid(?=\\s|$)",
    "replacement": "<green>my uuid is %player:uuid%</green>"
}
```

5.4.3.6 stripe

5.4.3.6.1 Purpose

Stripe the `style tags` in `chat message` sent by players based on permissions.

5.4.3.6.2 How it works?

This module will `strile` all `style tags` in chat message sent by a player, and only allows to use a specified style tag if the player has corresponding permission for the tag.

5.4.3.6.3 Example

Example: Allow players to use "<blue>" tag

```
/lp group default permission set fuji.style.chat.blue
```

Example: Allow players to use "<b>" tag

```
/lp group default permission set fuji.style.chat.b
```

Example: Allow players to use all tags

All tags also including dangerous tags like "<click>" tag which can run commands on clicked!

```
/lp group default permission set fuji.style.chat.*
```

5.4.3.7 spy

5.4.3.7.1 Purpose

Spy on specified `chat types`.

5.4.3.7.2 Command

- `/chat spy`

5.4.3.8 history

5.4.3.8.1 Purpose

Store `chat messages` as a history, and send them to a player joined the server.

5.4.3.8.2 Configuration

buffer_size How many chat messages should we save, so that we can send for a new-joined player.

5.5 cleaner

5.5.1 Purpose

This module provides the entity cleaner to remove specified entities automatically.

Note: Only use this module to clean some edge-case entity

Since the vanilla minecraft also has a cleaner to remove the item stack in the ground, so it's recommended to only use this module to clean some weak-loading entities, like: the sand item stack ...

5.5.2 Command

- `/cleaner clean`

Note: The cleaner will keep silent if cleans nothing

If the cleaner cleans nothing, then it will keep silent.(Which means you will not see any message in console, or in-game chat)

Tip: See what is cleaned in cleaner broadcast.

Hover your mouse on the cleaner broadcast, you can see what has been removed.

5.5.3 Configuration

cron The cron used to define the job to trigger `/cleaner clean`.

key2age The key is translatable key, which you can query in en_us.json language file in minecraft 1.21.
The translatable key of entity starts with entity.minecraft.
The translatable key of item starts with item.minecraft and block.minecraft.
The age is the existence time of the entity, the unit of age is game tick, which means 20 age = 20 ticks = 1 second.
The cleaner will only remove the entities whose translatable key equals key, and age greater equals the defined age, and the entity must not in the ignore list.

Example: Clean the sand-block entity lives longer than 60sec

```
"block.minecraft.sand": 1200
```

ignore Entities match the ignore list will not be cleaned.

ignore_item_entity Should we ignore all item entity.

ignore_living_entity Should we ignore all living entity?
If you want the cleaner to remove monster or animals, you should enable this option.

ignore_named_entity Should we ignore named entity.(With name tag, or name changed by anvil.)

ignore_entity_with_vehicle Like entity riding in some other entity, e.g. minecraft, pig or spider

ignore_entity_with_passengers

ignore_glowing_entity

ignore_leashed_entity

Note: The built-in safety rule

The cleaner will **always ignore** the following types:

1. player
2. any block attached entity (e.g. leash_knot)
3. any vehicle entity (e.g. minecart, boat ...)

5.6 color

5.6.1 Purpose

This module provides colorize for things.

5.6.2 Sub-Module

5.6.2.1 sign

5.6.2.1.1 Purpose

Colorize the sign.

5.6.2.1.2 Configuration

requires_corresponding_permission_to_use_style_tag By default, any player can use all style tags. Enable this options requires the player to has corresponding permission to use style tag. e.g. `"fuji.style.sign.<style-tag>"`

5.6.2.1.3 Example

Example: Write colorful text in sign

<red>This is the first line
<rb>The second line
<bold>The third line
<i>The last line

5.6.2.2 anvil

5.6.2.2.1 Purpose

Colorize the anvil.

5.6.2.2.2 Configuration

requires_corresponding_permission_to_use_style_tag By default, any player can use all style tags. Enable this options requires the player to has corresponding permission to use style tag. e.g. `"fuji.style.anvil.<style-tag>"`

5.7 command_alias

5.7.1 Purpose

This module allows you to define command alias, which redirect to the existing command node.

Note: A command node is identified by path

See also: https://minecraft.fandom.com/wiki/Commands

5.7.2 Example

Example: Shorten a existing command

The configuration create a command alias from `/r` to `/reply`

```
{
    "from": [
    "r"
    ],
    "to": [
    "reply"
    ]
}
```

Example: Shorten a existing command

The configuration create a command alias from `/sudo` to `/run as fake-op`

```
{
    "from": [
    "sudo"
    ],
    "to": [
    "run",
    "as",
    "fake-op"
    ]
}
```

5.7.3 What's more?

Tip: How can I define complex commands?

As you can see, the command alias module only support to `redirect` a simple command into another simple command. In other words, you can only use this module to create `alias` to a `existing command node`. Also, it's not allow to define `variable` inside a `command alias`.

If you want to define complex commands, use command_bundle module.

5.8 command_attachment

5.8.1 Purpose

This module allows you to attach commands into itemstack.

5.8.2 Command

- `/command-attachment`

Note: The same item-stack shares the-same instance

If you hold stick * 64 in your main-hand, then all the sticks share the same attached commands.

5.8.3 Example

Example: Make a magic-stick which heals the player on clicked.

Hold a stick item in your main hand.

```
/command-attachment attach-item-one heal
```

Example: Make a magic-stick which gives one dimaond on left clicked with use limit 3 and gives one gold_ingot on right clicked with use limit 5.

Hold a stick item in your main hand.

```
/command-attachment attach-item-one --maxUseTimes 3 --interactType LEFT give
%player:name% minecraft:diamond 1
/command-attachment attach-item-one --maxUseTimes 5 --interactType RIGHT
give %player:name% minecraft:gold_ingot 1
```

Example: Make a magic-stick which gives one apple on clicked with use limit 3 without destroying the item.

Hold a stick item in your main hand.

```
/command-attachment attach-item-one --maxUseTimes 3 --destroyItem false give
%player:name% minecraft:apple 1
```

Example: Query the attached commands in the mainhand item.

```
/command-attachment query-item
```

Example: Let an entity says hello on right click

```
/command-attachment attach-entity-one <entity-id> say hello %player:name%
```

Example: Say hello if you steppd on a specific block

```
/command-attachment attach-block-one 0 0 0 --interactType STEP_ON say hello
%player:name%
```

Example: Make a portal on a specific block

```
/command-attachment attach-block-one 0 0 0 --interactType STEP_ON tppos
--targetPlayer %player:name% --dimension minecraft:the_end --x 0 --y 66 --z
0 %player:name%
```

5.9 command_bundle

5.9.1 Purpose

This module allows you to create bundle commands, input one command, output many commands.

5.9.2 Command

- `/command-bundle`
  The root command of this module.

5.9.3 Feature

1. a user-friendly DSL to create new custom commands easily, with the interoperation of `variables`, `placeholders` and `selectors`.
2. support complex `command argument type`: `required argument`, `literal argument` and even the `optional argument` with a default value.
3. a powerful type-system to ensure the `type-safe` input, with fully command suggestion.

   Tip: To query all type strings

   `/fuji inspect argument-types`

4. register and un-register commands on the fly, without the server restart!

   Example: Reload the bundle commands

   Each time you modify the configuration file, you should issue `/fuji reload`, this will unregister all bundle commands in the server, and register the bundle commands defined in the file into the server.

   Also, you can use the `/command-bundle un-register` and `/command-bundle register` manually.

5.9.4 Example

Example: Combine commands into one command

```
{
    "requirement": {
    "level": 0,
    "string": null
},
    "pattern": "composite-heal",
    "bundle": [
    "say before heal %player:name%",
    "run as fake-op %player:name% particle minecraft:heart ~ ~2 ~"
    "run as player %player:name% heal",
    "say after heal %player:name%"
    ]
}
```

Example: Transform a command

```
{
    "requirement": {
    "level": 4,
    "string": null
},
    "pattern": "warn <player player-arg> <greedy greedy-arg>",
    "bundle": [
    "run as player %player:name% send-message $player-arg <red>You are
    ↪  warned: $greedy-arg"
    ]
}
```

Tip: Assign a string permission for a bundle command

See Permission

5.9.5 Reference

1. https://www.gamergeeks.net/apps/minecraft/particle-command-generator
2. https://learn.microsoft.com/en-us/minecraft/creator/documents/particleeffects?view=minecraft-bedrock-stable

5.10 command_cooldown

5.10.1 Purpose

This module provides:

1. **unnamed cooldown** : per command cooldown after the `command execution`
2. **named cooldown** : support to associate a named cooldown with commands.

Tip: How to write regex language?

See: Regex

5.10.2 Command

- `/command-cooldown`

5.10.3 Placeholder

1. `%fuji:command_cooldown_left_time <named-cooldown>%` the left time for specified named cooldown
2. `%fuji:command_cooldown_left_usage <named-cooldown>%` the left usage for specified named cooldown

5.10.4 Example

Example: Create a named cooldown

```
/command-cooldown create example 3000
```

Example: Test a named cooldown

```
/command-cooldown test example <player> --onFailed "say false
%fuji:command_cooldown_left_time 1%/%fuji:command_cooldown_left_usage 1%"
say true
```

Example: Reset a named cooldown for a player

Note that this will only reset the `timestamp` associated with the player, the `usage` associated with the player will not be reset.

```
/command-cooldown reset example <player>
```

Example: Create a named cooldown with 3 max usage and 15 sec cooldown

```
/command-cooldown create example 15000 --maxUsage 3
```

Example: Create a named global cooldown for all players

A `named global cooldown` means that, all players shares the same cooldown, instead of `per-player`.

```
/command-cooldown create example 3000 --global true
```

Example: Create a non-persistent named cooldown

A `non-persistent named cooldown` means that, the `timestamp` associated with a player will not be persisted into the storage. That's to say, a server restart will forget all `timestamp`, but the `usage` associated with a player will always be persisted.

```
/command-cooldown create example 999999999999 --persistent false
```

Taken this example, it means that each time the server restarted, the cooldown will be available only once.

5.11 command_event

5.11.1 Purpose

Execute commands on specific events.

5.11.2 How it works?

When an event occurs, this module will `execute` commands `as console` with the `contextual player`. The `contextual player` will be used to parse placeholders.

5.11.3 Example

Example: Welcome the new-bie player

```
"on_player_first_joined": {
    "command_list": [
    "send-broadcast <light_purple>Welcome new player %player:name% to join
    ↪  us!",
    "kit give %player:name% <kit-name>",
    "run as fake-op %player:name% rtp",
    "delay 10 spawnpoint %player:name%"
    ]
},
```

5.12 command_interactive

5.12.1 Purpose

This module allows you to write commands in **sign block**.

5.12.2 How it works?

When a player `right click` a `sign block`, this module will check if the sign block contains the character `"/"`. If contains, then we `treat as` the player issue the command followed by the character.

5.12.3 Definition

A `sign block` that contains the character `"/"` is named `an interactive sign block`.

Note: How can i edit an interactive sign block?

You need to press `shift + right click` to edit an interactive sign.

5.12.4 Example

Example: Basic usage

/say hi `%player:name%`
line 2 empty
line 3 empty
line 4 empty

Example: Add a prefix description text

prefix /say first
/say the second
/say hi `%player:name%`
/say the last command

Example: Concat commands between lines

prefix /say this is
the first /say and the
second
line 4 empty

5.13 command_meta

5.13.1 Purpose

This module provides commands to operate on commands.

5.13.2 Sub-Module

5.13.2.1 run

5.13.2.1.1 Purpose

This module provides `/run` command, which can run a command with context.

5.13.2.1.2 Command

- `/run`

5.13.2.1.3 Example

Example: Give random diamonds to online players

```
/run as console give @a minecraft:diamond %fuji:random 8 32%
```

Example: Give online players random diamonds

```
/run as console foreach give %fuji:escape player:name% minecraft:diamond
%fuji:escape fuji:random 8 32 1%
```

Example: Execute a command as a player

```
/run as player <player> back
```

Example: Execute a command as fake-op

```
/run as fake-op <player> give %player:name% minecraft:apple 1
```

5.13.2.2 for_each

5.13.2.2.1 Purpose

This module provides /foreach command. If a command is only targeted for single player, you can use `/foreach` to apply it for each player online.

5.13.2.2.2 Command

- `/foreach`

5.13.2.2.3 Example

Example: Say hello to online players

```
/foreach say hello %player:name%
```

Tip: Escape the placeholder properly

If you use foreach in scheduler module, then you should escape (Write `%fuji:escape player:name%` instead of `%player:name%`) the placeholder.
It's because the command-scheduler module will try to parse the placeholder, and you need to escape the placeholder, so that the placeholder can be parsed by `/foreach` command.
Here is an example about escape the foreach command in scheduler command list: `/foreach give %fuji:escape player:name% minecraft:diamond 16`

5.13.2.3 chain

5.13.2.3.1 Purpose

A chain command allows you to run another 2 commands, the first is any command, and the second is the chain command.

Note: The return value of a command

In vanilla minecraft, the return value of command, is an integer:

1. **<0** failed
2. **=0** passed
3. **>0** success

5.13.2.3.2 Command

- `/chain`

5.13.2.3.3 Example

Example: A nested chain.

```
/chain say 1 chain say 2 chain say 3
```

Example: A breakable chain.

```
/chain bad command here chain say 2
```

Example: Use chain command with predicate command

```
/run as player <player> chain test-level-perm %player:name% 4 chain say
value is true
```

5.13.2.4 delay

5.13.2.4.1 Purpose

Delay command allows you to execute a command in specified seconds.

Note: Delay command is only used for short term job

The command `/delay` is only used to perform short-term job, and will not be persisted on server shutdown. If you want to define long-term job, using command_scheduler

5.13.2.4.2 Command

- `/delay`

5.13.2.4.3 Example

Example: A basic usage

```
/delay 3 say three seconds passed
```

Example: A nested delay

```
/delay 1 delay 2 delay 3 say 6 seconds passed.
```

5.13.2.5 json

5.13.2.5.1 Purpose

Provides a unified json editor.

5.13.2.5.2 Command

- `/json`

5.13.2.5.3 Exaxmple

Example: Read a key

```
/json read "config/fuji/config.json" "$.core.quartz.logger_level"
```

Example: List keys

```
/json read "config/fuji/config.json" "$.modules.keys()"
```

Example: Set a key

```
/json write "config/fuji/config.json" "$.core.quartz.logger_level" STRING
INFO"
```

5.13.2.5.4 Reference

1. https://goessner.net/articles/JsonPath/

5.13.2.6 attachment

5.13.2.6.1 Purpose

Provides a unified attachment facility, which can attach any data to any object.

5.13.2.6.2 Command

- `/attachment`

5.13.2.6.3 Example

Example: Set a attachment

```
/attachment set news today hello world
```

Example: Get a attachment

```
/attachment get news today
```

5.13.2.7 shell

5.13.2.7.1 Purpose

This module provides `/shell` command, which executes the command line in the host shell.

Danger: This is a dangerous module

This module is a powerful and dangerous module, not recommended to enable it.

5.13.2.7.2 Configuration

enable_warning A precautionary option to prevent this module is enabled.

security The security options for this module.

- **only_allow_console** Only allow the console to execute `/shell` command.
- **allowed_player_names** Only allow the specified players to execute `/shell` command.

5.13.2.7.3 Example

Example: Create a file using placeholder

```
/shell touch %player:name%.dangerous
```

Example: Execute a program in the host os

```
/shell emacs
```

Example: Backup the data of your server

You can use `shell module` with `command scheduler module` as a combo: define a job to execute the shell command in os to execute a program to backup the data of your server. See more: https://rdiff-backup.net/

Example: Possible to download a virus from Internet and execute it!

```
/shell ...
```

5.14 command_permission

5.14.1 Purpose

This module provides the customization of **the requirement of all commands**.

5.14.2 Command

- `/command-permission`

5.14.3 How it works?

The vanilla minecraft use a command system named brigadier.
All the commands are registered, parsed and executed by brigadier.
In this system, all commands are build into **a tree structure**, that is to say, all commands are a direct or in-direct child of the **root command node**.

Example: What is the path of a specific command node?

For example, the command `/gamemode creative Steve` is composed by 3 command node:

1. `"gamemode"` = `a literal whose name is "gamemode"`
2. `"creative"` = **an argument whose type is gamemode, its name is** `"gamemode"` **and its value is** `"creative"`
3. `"Steve"` = **an argument whose type is player, its name is** `"target"`**, and its value is** `"Steve"`

We say that the command path of `/gamemode creative Steve`, is `["gamemode", "gamemode", "target"]`.

Tip: How to query the name of an argument

You can issue `/help gamemode` which will display the name of arguments. Or you can issue `/fuji inspect server-commands` to query the `command path` of all commands registered in the server.

Also, each **command node** has its **requirement**, which is a **predicate** to check if the **command source** can use the command node.

Tip: Query the command path of a command.

`/lp group default permission set fuji.permission...`
or `/command-permission`
or `/fuji inspect server-commands`

Warning: Understand the parent permission node and its children node

It's recommend to modify the `apply-sponge-implicit-wildcard` option to `false` in `luckperms.conf`, to control the `permission node` more fine-grained.

5.14.4 Example

Example: Allow everyone to use `/gamemode` command

```
/lp group default permission set fuji.permission.gamemode true
```

Tip: Allow the client-side to use gamemode switcher menu

After you assign the `/gamemode` command permission for players, the client-side also requires to install a mod to bypass the client-side permission checking: https://modrinth.com/mod/switcher

Example: Allow everyone to use `/gamemode` command except the player Alice

```
/lp group default permission set fuji.permission.gamemode true
/lp user Alice permission set fuji.permission.gamemode false
```

Example: Only allow everyone to use `/gamemode spectator`

It's impossible to assign a single gamemode, since the command path of `/gamemode creative` and `/gamemode spectator` are both `"gamemode.gamemode"`.
Notice that the first `"gamemode"` in the command path, means the literal argument `"gamemode"`.
The second `"gamemode"` in the command path, means an argument, whose type if `gamemode`. This gamemode argument contains all the 4 gamemodes: adventure, creative, spectator and survival. That's the real reason why we can't assign a single gamemode for the command `/gamemode`.

If you really want to assign only 1 single gamemode for everyone, you can use command_bundle to create a new command, which only switch the gamemode of player into spectator.

Example: More examples

See Permission

5.15 command_rewrite

5.15.1 Purpose

This module allows you to use regex language to rewrite the command line a player issued.

5.16 command_scheduler

5.16.1 Purpose

Define `jobs` using `cron expression` to execute commands.

5.16.2 How it works?

A Job is defined by crons, with conditions like `left times`.
A job will be `triggered` if current time meets any of its cron expressions.
A `commands` will be chosen randomly from the `commands list` of the job and executed if the following conditions are meet:

1. The `enable` of the job is `true`
2. The `left times` of the job $>= 0$

The commands will be executed as console.

5.16.3 Command

- `/command-scheduler trigger`
  Immediately trigger a job.
- `/command-scheduler list`
  List all defined jobs.

5.16.4 Example

Example: Define a job that executes commands 32 times on every minute

```
{
    "name": "example_job",
    "enable": true,
    "left_times": 32,
    "crons": [
    "0 * * ? * * *"
    ],
    "commands_list": [
    [
    "say 1 minute passed"
    ],
    [
    "say 60 seconds passed"
    ]
    ]
}
```

5.17 command_spy

5.17.1 Purpose

Log command execution event into the console.

5.17.2 Configuration

ignore Ignore the command that matches regex.

spy_on_console should we spy the command execution of the console?

5.18 command_toolbox

5.18.1 Purpose

This module provides some simple commands. (misc commands)

5.18.2 Sub-Module

5.18.2.1 bed

- `/bed`

5.18.2.2 extinguish

- `/extinguish`

5.18.2.3 feed

- `/feed`

5.18.2.4 fly

- `/fly`

5.18.2.5 god

- `/god`

5.18.2.6 hat

- `/hat`

5.18.2.7 sit

- `/sit`

5.18.2.8 heal

- `/heal`

5.18.2.9 lore

- `/lore`

Example: Set lore for item in mainhand

```
/lore set <rainbow>the first line<newline><bold><green>the second
```

5.18.2.10 more

- `/more`

5.18.2.11 ping

- /ping

5.18.2.12 realname

- /realname

5.18.2.13 nickname

- /nickname

5.18.2.14 repair

- /repair

5.18.2.15 reply

- /reply

5.18.2.15.1 How it works?

When a player use the command /msg or /tell, we track the usage, and allow the target player to use /reply to send message to the player recently meg he.

Tip: Create a command alias like /r

You can use command_alias module to create a command alias from /r to /reply.

5.18.2.16 seen

- /seen

5.18.2.17 suicide

- /suicide

5.18.2.18 top

- /top

5.18.2.19 trashcan

- /trashcan

5.18.2.20 tppos

- /tppos

5.18.2.20.1 Example

Example: Teleport to offline location of a player

```
/tppos offline <player>
```

Example: Specify another player as the target player

```
/tppos --targetPlayer <player>
```

Example: Use fixed teleport mode

When any of the following args is specified, the /tppos command will do a `fixed teleport`.

```
/tppos --x 32 --y 64 --z 128 --yaw 60 --pitch 90 --dimension
minecraft:overworld
```

Example: Use random teleport mode

The /tppos command will do a `random teleport` if not doing a `fixed teleport`.

```
/tppos --centerX 0 --centerZ 0 --circle true --minRange 512 --maxRange 2048
--minY 32 --maxY 128 --maxTryTimes 8 --dimension minecraft:overworld
```

5.18.2.21 warp

- /warp

5.18.2.21.1 Example

Example: Set a display name for a warp

```
/warp set-name <warp> <blue>This is the display name
```

Example: Set a lore for a warp

```
/warp set-lore <warp> <blue>This is the first line|<red>This is the second
line
```

5.18.2.22 burn

- /burn

5.18.2.23 help-op

- /help-op

5.18.2.24 near

- /near

5.18.2.25 jump

- /jump

5.18.2.26 compass

- /compass

5.18.2.27 glow

- /glow

5.18.2.28 freeze

- /freeze

5.19 command_warmup

5.19.1 Purpose

This module adds a cooldown before command execution.

Note: The difference between command warmup and command cooldown

command warmup is before command execution, while command cooldown is after that.

5.19.2 Example

Example: Set warmup for all commands

```
{
    "command": {
    "regex": ".+",
    "ms": 3000
},
    "interruptible": {
    "enable": true,
    "interrupt_distance": 3.0,
    "interrupt_on_damaged": true,
    "interrupt_in_combat": true
}
}
```

Example: Set warmup for all commands except the `/back` command with a special setup

Since the rules are matched from up to down in order, you can just put the special case above.

```
[
        {
        "command": {
            "regex": "back",
            "ms": 10000
        },
        "interruptible": {
            "enable": true,
            "interrupt_distance": 3.0,
            "interrupt_on_damaged": true,
            "interrupt_in_combat": true
        }
    }
,
        {
        "command": {
            "regex": ".+",
            "ms": 3000
        },
        "interruptible": {
            "enable": true,
            "interrupt_distance": 3.0,
            "interrupt_on_damaged": true,
            "interrupt_in_combat": true
        }
    }
]
```

Example: Set warmup for all commands except the /back command

Use `negative lookahead` to exclude a string that starts with "back".

```
{
    "command": {
    "regex": "(?!back).+",
    "ms": 3000
},
    "interruptible": {
    "enable": true,
    "interrupt_distance": 3.0,
    "interrupt_on_damaged": true,
    "interrupt_in_combat": true
}
}
```

Example: Share the same setup for multiple commands

Use the `or clause` to share the same setup.

```
{
    "command": {
    "regex": "(back)|(heal)|(feed)",
    "ms": 3000
},
    "interruptible": {
    "enable": true,
    "interrupt_distance": 3.0,
    "interrupt_on_damaged": true,
    "interrupt_in_combat": true
}
}
    ,
```

5.20 deathlog

5.20.1 Purpose

This module logs the inventory on player death.

5.20.2 Command

- `/deathlog`

5.20.3 Example

Example: Query the logs for a player

```
/deathlog view <player>
```

Tip: The death log index number is clickable

You can click the number to quickly restore the logged inventory to your inventory.

Example: Restore a death log from a player for a player

```
/deathlog restore <player> 0 <player>
```

5.21 disabler

5.21.1 Purpose

This module provides **disablers** to disable checkers inside **vanilla minecraft**.

5.21.2 Sub-Module

5.21.2.1 chat_speed_disabler

Disable **Kicked for spamming**.

5.21.2.2 move_speed_disabler

Disable **player moved too quickly** and **vehicle moved too quickly**.

5.21.2.3 move_wronlgy_disabler

Disable **player moved wrongly**.

Warning: The movement anti-cheat inside vanilla minecraft is bad

Inside the vanilla minecraft server, there is a checker used to check if the player moves correctly. However, this checker usually makes wrong detection, and force setback the player, which makes the client-side gameplay feel lagged.

5.21.2.4 max_player_disabler

Disable the max players limit of the server.

5.22 echo

This module provides commands to send echo to players.

Note: Many other modules require this moudle

Other modules may generate a `default configuration` including the commands provided by `echo` module. If the `echo module` is disabled, then these `echo commands` will not exist, causing a `command syntax error` while executing these commands.

5.22.1 Sub-Module

5.22.1.1 send-message

- `/send-message`

Example: Say hello to a player

```
/send-message <player> Hello %player:name%
```

5.22.1.2 send-broadcast

- `/send-broadcast`

Example: Say hello to all players

```
/send-broadcast Hello %player:name%
```

5.22.1.3 send-actionbar

- `/send-actionbar`

Example: Say hello to a player

```
/send-actionbar <player> Hello %player:name%
```

5.22.1.4 send-title

- `/send-title`

Example: Send title to a player

```
/send-title <player> --mainTitle "<rainbow>Hello" --subTitle "<blue>World"
--fadeInTicks 60 --stayTicks 60 --fadeOutTicks 60
```

Example: Send title to online players

```
/foreach send-title %player:name% --mainTitle "<rainbow>Hello %player:name%"
```

5.22.1.5 send-toast

- /send-toast

Example: Send toast to a player

```
/send-toast <player> --icon minecraft:golden_carrot <rb>eat this carrot
```

5.22.1.6 send-chat

- /send-chat

Example: Send chat as a player

```
/send-chat Steve i am steve.
```

Example: Send chat as a player for online players

```
/foreach send-chat %player:name% i am %player:name%
```

5.22.1.7 send-bossbar

- /send-bossbar

Example: A simple exapmle

```
/send-bossbar <player> Hello World
```

Example: All in one exapmle

```
/send-bossbar <player> --stepType BACKWARD --totalMs 5000 --color PURPLE
--style NOTCHED_6 --notifyMeOnComplete true --commandList "say the
player %player:name% is healed|heal %player:name%" <rb>Healing is coming
[elapsed_time]/[total_time]/[left_time]
```

5.22.1.8 send-custom

- /send-custom

Example: Create a custom text

Create a plain text file named "guide" in **config/fuji/modules/echo/send_custom/custom-text/guide** with content:

```
<blue>===== custom text =====
Hello <orange>%player:name%</orange>, you are in
↪ <orange>%world:name%</orange> now.

<hover:show_text:'you see me!'>Hover me</hover>

<click:run_command:'/back'>click me to run `/back` command</click>

<u><i><click:change_page:'3'>click me to the third page (this only works
↪ inside a book)</click></i></u>

<newpage><blue>This is the second page!

<click:suggest_command:'/back'>click me to suggest /back command (This
↪ doesn't work inside a book)</click>

<insert:'hello'>shift + click me to insert "hello" (This doesn't work
↪ inside a book)</insert>

<click:open_url:'https://placeholders.pb4.eu/user/text-format/'>click me
↪ to open the url</click>

<newpage>This is the third page!

<bold><click:change_page:'1'>click me to the first page</click></bold>

<orange>You can press `<keybind:'key.jump'>` key to jump!</orange>

<gradient:red:green:blue>This is gradient text.</gradient>

<rb>The rainbow text</rb>

<newpage>The end.
```

Example: Send custom text as a book

```
/send-custom as-book <player> guide --author "alice" --title "<rb>The Guide"
--giveBook true --openBook true
```

Example: Send custom text as a message

```
/send-custom as-message <player> guide
```

5.23 fuji

5.23.1 Purpose

This module provides the command `/fuji`, which includes some operations on fuji itself.

5.23.2 Command

- `/fuji reload`
  Reload all `configuration files` and all `modules`.

Note: Module itself can't be hot reloaded

After you `enable` or `disable` a module, you must `restart` the server.

- `/fuji about`
  Open a gui to display the about, including the mod version and contributor list.
- `/fuji inspect modules`
  Inspect all the enabled/disabled modules.
- `/fuji inspect server-commands`
  Inspect all the registered commands in the server.
- `/fuji inspect fuji-commands`
  Inspect all the commands registered by fuji mod.

Note: This will not show the requirement override from command permission module

The `requied level permision` and `required string permission` are the `default` value set by fuji. If you are using command_permission module, then this gui will not show the `overrided requirement` of a command.

- `/fuji inspect argument-types`
  Inspect all the registered argument types.
- `/fuji inspect configurations`
  Inspect all the loaded configuration files.
- `/fuji inspect registry`
  Inspect the registries in server.

5.24 functional

5.24.1 Purpose

This module allows players to open a virtual gui of functional-block.

5.24.2 Sub-Module

5.24.2.1 workbench

- `/workbench`

5.24.2.2 enchantment

- `/enchantment`

5.24.2.3 grindstone

- `/grindstone`

5.24.2.4 stonecutter

- `/stonecutter`

5.24.2.5 anvil

- `/anvil`

5.24.2.6 cartography

- `/cartography`

5.24.2.7 enderchest

- `/enderchest`

5.24.2.8 smithing

- `/smithing`

5.24.2.9 loom

- `/loom`

5.25 gameplay

5.25.1 Sub-Module

5.25.1.1 multi_obsidian_platform

5.25.1.1.1 Purpose

This module makes every **ender portal frame** generates its own **obsidian platform** (Up to 128 in survival-mode).
You can even use creative-mode to build more ender portal frame and more obsidian platform.

Note: All the obsidian platforms are vanilla-respect

All the extra obsidian platforms have the same behaviour as the vanilla one which locates in (100,50,0).

5.25.1.1.2 Configuration

factor The coordination-conversion factor between overworld and the_end. In vanilla minecraft, the factor between overworld and the_nether is 8.

5.25.1.2 carpet

5.25.1.2.1 Sub-Module

5.25.1.2.1.1 fake_player_manager

Purpose Enable this module requires carpet-fabric mod installed. This module provides some management for fake-player.

Command

- `/player who`
- `/player renew`

Configuration

caps_limit_rule How many fake-player can each player spawn (in different time)?

The tuple means (day_of_week, minutes_of_the_day, max_fake_player_per_player).
The range of day_of_week is [1,7].
The range of minutes_of_the_day is [0, 1440].

For example: (1, 0, 2) means if the days_of_week >= 1, and minutes_of_the_day >= 0, then the max_fake_player_per_player now is 2.
Besides, you can add multi rules, the rules are checked from up to down.
The first rule that matches current time will be used to decide the max_fake_player_per_player.
You can issue `/player who` to see the owner of the fake-player.
Only the owner can operates the fake-player. (Op can bypass this limit)

renew_duration_ms How long should we renew when a player issue `/player renew` The command /player renew allows the player to manually renew all of his fake-player. If a fake-player don't gets renew, then it will expired and get killed. Use-case: to avoid some long-term alive fake-player.

transform_name The rule to transform the name of fake-player. Use-case: add prefix or suffix for fake-player.

5.25.1.2.1.2 better_info

Purpose Add nbt query for /info block command. Add the command /info entity.

5.26 head

5.26.1 Purpose

This module allows players to buy decorative heads from a head-database.

5.26.2 Command

- `/head`

5.26.3 Configuration

economy_type Can be `ITEM` or `FREE`

cost_type This option is used when economy type is ITEM, to specify which item as the currency.

cost_amount This option is used when economy type is ITEM, to specify the amount of currency to buy a head.

5.27 home

5.27.1 Purpose

This module allows players to set a teleportation point as their home.

5.27.2 Command

- `/home`

5.27.3 Meta

1. fuji.home.home_limit The home number per player limit.

5.28 kit

5.28.1 Purpose

This module allows you to make kits.

5.28.2 Concept

A **kit** is a set of itemstack.

5.28.3 Command

- `/kit editor`
  Open the kit editor gui.
- `/kit give`
  Give a kit to a player.

5.28.4 Example

Example: Create a kit

Use `/kit editor` to create a kit.

Note: The item slot position will keep

The item put inside the kit will keep its original position, so you can put armors in the right position.

Example: Give a kit to a player

`/kit give <player> <kit-name>`

Note: The giving function behaviour

1. try to insert the item in the specified slot
2. try to insert the item in any slot
3. drop the item in the ground with the player as its thrower

Example: Associate a cooldown to a kit

To associate a cooldown with commands: see command_cooldown

Create a `named cooldown` with 60sec cooldown and infinite usage: `/command-cooldown create example-kit-cooldown 60000`

Test the named cooldown, giving the kit to the player if the test is success:`/command-cooldown test example-kit-cooldown <player> --onFailed "send-message %player:name% wait a moment" kit give %player:name% example-kit|send-message %player:name% kit received.`

Now, create a new command using command_bundle module to execute the test command

Example: The config to create a new command for cooldown test command to give a kit if the test is success

```
{
    "requirement": {
    "level": 0,
    "string": null
},
    "pattern": "claim-example-kit",
    "bundle": [
    "command-cooldown test example-kit-cooldown %player:name%
    ↪ --onFailed 'send-message %player:name% wait a moment
    ↪ (%fuji:command_cooldown_left_time example-kit-cooldown%
    ↪ ms)' kit give %player:name% example-kit|send-message
    ↪ %player:name% kit received."
    ]
}
```

Example: Claim a specific kit automatically for online players

You can use command_scheduler module to execute the `/claim-example-kit` for online players automatically every minute.

Example: Give a kit to newbie player automatically

See command_event

5.29 language

5.29.1 Purpose

This module provides client-side multi-language support.

Note: What is client-side language?

When the client joins a server, it will send its client options, including the client-side language value. The server can send messages in language used by the client later.

5.29.2 Feature

1. Client-Side Respect The client-side language will be respected if possible.

2. Lazy-load Only load the necessary language into the memory.

5.29.3 Difference

Disabled: All the players use the `default_language`.
Enabled: Fuji will **try** to respect the player's client-side language, if the server-side supports.

5.30 motd

5.30.1 Purpose

This module provides motd customization.

5.30.2 Example

Example: Configure server icons

You can put 64x64 pixels `.PNG images` into the directory `config/fuji/modules/motd/icon`, this module will pick up a random image as the icon of the server to respond the server status request.

Example: Configure random motd text

You can write multiple `motd text` in the configuration, the `motd text` will be chosen randomly.

5.30.3 Reference

1. https://colorize.fun/en/minecraft

5.31 multiplier

5.31.1 Purpose

This module provides some **numeric multiplier**.

5.31.2 Supported Numeric Types

1. damage damage to player

2. experience experience a player gained

5.31.3 Example

Example: Double the damage from zombie to a player

```
/lp group default meta set fuji.multiplier.damage.minecraft:zombie 2
```

Example: Cancel the fall damage

```
/lp group default meta set fuji.multiplier.damage.minecraft:fall 0
```

Example: Double all damage to a player

```
/lp group default meta set fuji.multiplier.damage.all 2
```

Example: Double all experience a player gained

```
/lp group default meta set fuji.multiplier.experience.all 2
```

Example: Half all damage to a player

```
/lp group default meta set fuji.multiplier.damage.all 0.5
```

5.32 nametag

5.32.1 Purpose

This module provides nametag customization.

5.32.2 Configuration

update_cron The cron used for the job to **update** the properties of **display entity**.

style Define the style for the nametag **display entity**.

render Define the nametag render behaviour.

Note: The explanation of each field in style and render

You can refer to the minecraft wiki about **display entity**: https://minecraft.wiki/w/Display

5.32.3 Example

Example: Set background to blue color

```
"background": -16776961
```

Example: Set half transparency

```
"text_opacity": 128
```

Example: Scale the size of text into double

```
"scale": {
    "x": 2.0,
    "y": 2.0,
    "z": 2.0
},
```

5.33 placeholder

5.33.1 Purpose

This module provides more `placeholder` for `Text Placeholder API` mod.

5.33.2 Command

- `/placeholder`

5.33.3 Placeholder

1. `%fuji:player_mined%` sum of mined blocks of a player
2. `%fuji:server_mined%` sum of mined blocks of a server
3. `%fuji:player_placed%` sum of placed blocks of a player
4. `%fuji:server_placed%` sum of placed blocks of a server
5. `%fuji:player_killed%` sum of killed entities of a player
6. `%fuji:server_killed%` sum of killed entities of a server
7. `%fuji:player_moved%` sum of moved distance of a player
8. `%fuji:server_moved%` sum of moved distance of a server
9. `%fuji:player_playtime%` sum of playtime of a player
10. `%fuji:server_playtime%` sum of playtime of a server
11. `%fuji:health_bar%` the health bar of a player
12. `%fuji:rotate hello%` rotate the string **hello**
13. `%fuji:has_permission%` check luckperms permission
14. `%fuji:has_meta%` get luckperms meta
15. `%fuji:random_player%` get a random online player
16. `%fuji:random 1 5%` get a random number from 1 to 5
17. `%fuji:escape%` escape a placeholder from the parser. An optional number argument is used as the levels to escape.
18. `%fuji:protect%` protect a string from the parser.
19. `%fuji:date%` get current date.

    An optional string argument is used to set the **date formatter**, for example, `%fuji:date HH:MM%`.
    See also: https://docs.oracle.com/javase/8/docs/api/java/text/SimpleDateFormat.html
20. `%fuji:player_prefix%` player luckperms prefix
21. `%fuji:player_suffix%` player luckperms suffix
22. `%fuji:pos%` player current position

5.33.4 What's more?

Tip: Use placeholder in language file

It's allowed to write placeholders in language file.

Note: Some other mods that provide more placeholders

https://placeholders.pb4.eu/user/mod-placeholders/

5.34 predicate

5.34.1 Purpose

This module provides `predicate` commands.

5.34.2 Definition

A command that suffixes with the character `"?"` is named **`predicate command`**.
e.g. The `/has-perm?` command.

5.34.3 How it works?

The **`return value`** is represented in an **`integer`** provided by vanilla minecraft command return layer.
This command will return `1` if test is success, and `-1` if test is failed.

See more: https://minecraft.fandom.com/wiki/Commands/return

5.34.4 Example

Example: Test a condition and then run a command

See more in chain

```
/run as player <player> chain has-perm?  %player:name% 4 chain say value is
true
```

5.35 profiler

5.35.1 Purpose

This module shows server health status, such as os, vm, cpu, ram, tps, mspt and gc.

Warning:

Enable this module requires spark mod installed.

5.35.2 Command

- `/profiler`

5.36 pvp

5.36.1 Purpose

This module provide pvp state toggle.

5.36.2 Command

- /pvp

5.37 rtp

5.37.1 Purpose

Provides random teleportation.

5.37.2 Feature

1. Per dimension configurable.
2. Ignore fluid blocks, such as water and lava.
3. Ignore powered snow.

5.37.3 Command

- `/rtp`

5.37.4 Configuration

setup Teleport setup per dimension. Dimensions that are not in the list will be disabled for rtp.

5.37.5 What's more?

Tip: Improve the performance on rtp

It's highly recommended to pre-gen the world chunks. To gen a new chunk during rtp requires about 2 to 10 seconds. If a chunk is pre-gen, then it will be fast.

5.38 skin

5.38.1 Purpose

This module provides skin customization.

5.38.2 Command

- `/skin`

5.38.3 Configuration

default_skins The default skin used for player who has no skin set.

5.38.4 Example

Example: Set a skin from mojang

```
/skin set mojang dream
```

Example: Set a skin from url

```
/skin set web slim "https://s.namemc.com/i/bd53d152d0cd91d0.png"
```

Example: Use default skins

```
/skin use-default-skins
```

Example: Use online skin

```
/skin use-online-skin
```

5.39 system_message

5.39.1 Purpose

This module provides **system message** customization.

5.39.2 How it works?

The server will send `translatable text` to the client, by default, the client will display the translatable text in the language used in client-side. This module will hijack the pre-define `translatable text` in the server-side, and sends the client a `customized text` directly. As a loss of this approach, the client will always receive the customized text defined in the server, instead of the `translatable text` defined in the client.

Note: What is translatable text?

For example:

1. player join and leave server message
2. player advancement message
3. player death message
4. player command feedback
5. player whitelist message

Note: How to query the translatable text?

See mojang's official language file, all keys inside the file are for `translatable text`.
Note that, some keys in the file are client-side only translatable text, which means that the server will never send this type of translatable text to the client.

Warning: Possible conflicting with StyledChat

If you are using this module with **StyledChat** mod installed, you need to remove the conflicting **language keys** in **key2value** configuration.
See also: https://github.com/sakurawald/fuji/issues/65
See also: https://github.com/sakurawald/fuji/issues/203

5.39.3 Example

Example: Custom the player join and leave message.

```
"system_message": {
    "enable": true,
    "key2value": {
        "multiplayer.player.joined": "<rainbow>+ %s",
        "multiplayer.player.left": "<dark_gray>%s leeeeeeeeft the game"
    }
},
```

Example: Custom a specific death message.

```
"system_message": {
    "enable": true,
    "key2value": {
        "death.attack.explosion.player": "<rainbow>%1$s booooooom because
        ↪  of %2$s"
    }
},
```

Example: Custom messages used in screen.

```
"system_message": {
    "enable": true,
    "key2value": {
        "multiplayer.disconnect.server_shutdown": "Server closeeeeeeeed",
        "multiplayer.disconnect.not_whitelisted": "<rainbow>Please apply a
        ↪  whitelist first!"
    }
},
```

Example: Custom messages in a container

```
"key2value": {
    "container.chest": "<rb>I see you opening the chest!"
}
```

Example: Custom the vanilla command feedback.

```
"system_message": {
    "enable": true,
    "key2value": {
        "commands.seed.success": "<rainbow> Seeeeeeeeeeed: %s"
    }
},
```

Example: Custom the player disconnect message

```
"system_message": {
    "enable": true,
    "key2value": {
        "multiplayer.player.left": "<dark_gray>%s leeeeeeeeft the game"
    }
},
```

Example: Cancel the player disconnect message

```
"system_message": {
    "enable": true,
    "key2value": {
        "multiplayer.player.left": null
    }
},
```

5.40 tab_list

5.40.1 Purpose

This module provides tab list customization.

5.40.2 Configuration

update_cron The cron used for the job to update the tab list.

style The style for tab list.

5.40.3 Sub-Module

5.40.3.1 sort

5.40.3.1.1 Purpose

If enable this module, the `player names` in `tab list` will be sorted by `weight`.
The default weight is 0, the range of weight is [0, 675], which means you can set at most 676 sort groups.

5.40.3.1.2 How it works

The tab list sort method is client-side decided. So the workaround is to send virtual-player entry to the client-side, and hide the real player in client-side's tablist.
In this case, the client-side will find that, all command target selector will display the virtual-player. And you can see the virtual-player in client-side's Player Reporting UI.

Note: The virtaul player has no perforamnce issue

The virtual-player is just an entry listed in tab list, when the client ask the server tab list, the server lie with the virtual-player list.
There is not a real player entity in the server side, so no extra performance problem.
The sync method is event-based, and cached, so the performance is good.

5.40.3.1.3 Example

Example: Set a weight in a group

`Issue the command` `/lp group default meta set fuji.tab_list.sort.weight 1`
After you set a new weight, you should issue `/fuji reload` or re-connect to refresh the tab-list.

5.40.3.1.4 Sub-Module

5.40.3.1.4.1 sync_game_profile

Whether to copy the game profile from real-player to virtual-player.

5.40.3.2 faker

5.40.3.2.1 Purpose

This module is used to send random fake data to client.

5.41 teleport_warmup

5.41.1 Purpose

This module adds a warmup cooldown before player-teleportation.

5.41.2 Configuration

warmup_second The second to wait before the teleportation.

interrupile Should we interrupt this teleporation if some conditions meet?

dimension Per dimension configuration.

blacklist Only apply teleport warmup in the following dimensions.

> **Warning: Dimensions that created by other mods may have special behaviour**
>
> Some other mods will add extra dimension (like, the mod the-bumblezone-fabric). Their dimension portal may work in a different way, so this module may not be compatible with these mods.
> In the default options, we only allow this module works in the vanilla minecraft dimensions.

5.42 temp_ban

5.42.1 Purpose

This module provides temp-ban.

5.42.2 Command

- `/temp-ban`

5.42.3 Example

Example:

```
/temp-ban player <player> 1s2m3h4d5w6M7y bad boy
```

5.43 tester

5.43.1 Purpose

This module is only used for development. If you are a developer, you can register new commands into this module for test-purpose.

5.43.2 Command

- `/tester`

Warning:

You should not use this module at the production-environment, because it almost does nothing useful.

5.44 top_chunks

5.44.1 Purpose

This module compute a **laggy score** for all loaded chunks, and return the topN lagged chunks. Higher score means more lagged.

5.44.2 Command

- `/chunks`

5.44.3 Configuration

top The top chunks to show in `/chunks` command

nearest_distance For a chunk, how much the radius used to search the nearest player around the chunk.

hide_location Should we hide the chunk-position for a lagged-chunk? Hide chunk location to avoid grief or privacy purpose.

type2score The dict to define how lagged a type(entity/entity_block) should be.
For example:

Example: What is the meaning of type2score field?

The configuration means that if there are 15 zombies inside a chunk, then the chunk gets score 15 * 4 = 60. Any other types not specified in type2score will use the score defined for type **default**.

```
"type2score": {
    "entity.minecraft.zombie": 4,
    "default": 1
}
```

5.45 tpa

5.45.1 Purpose

This module provides teleport request for players.

5.45.2 Command

- `/tpa`
- `/tpahere`
- `/tpaacept`
- `/tpadeny`
- `/tpacancel`

5.45.3 Configuration

timeout Tpa request expiration duration, unit is second.

mention_player See 5.4.3.3.2

5.46 view

5.46.1 Purpose

This module provides the player slot editor.

5.46.2 Command

- `/view inv`
- `/view ender`

5.47 whitelist

5.47.1 Purpose

This module makes the mojang vanilla whitelist system only compares the **username** and **ignore UUID**.

Warning: Only enable this module in offline-mode server

If you are hosting a online-mode server, you will never need to enable this module.

5.48 works

5.48.1 Purpose

This module provides a bill-board gui for players to show their **works**.

5.48.2 Concept

A **work** is a **teleporation point**.

The types of work are as follows:

1. Non-Production-Work the project don't produce any resource (e.g. bone, string, coal).

2. Production-Work the project produce some resource.

Note: The main difference between non-production work and production-work

For a production-work, fuji provides the `production sample` to count the `hopper` and `minecart-hopper`

Tip: About the production counter

1. You can use the production counter provided by production work to sample the output.
2. This module works with carpet-fabric's hopper counter. You can use both of them at the same time.
3. The hopper counter provided by this module will not destroy the item.

5.48.3 Command

- `/works`

5.48.4 Configuration

sample_time_ms For a production-work, how long should we sample it?

sample_distance_limit For a production-work, how large the radius should we considered as the work's production.

sample_counter_top_n For a production-work, we only display the topN output items.

5.49 world

5.49.1 Purpose

Provides a unified world management.

5.49.2 Command

- `/world`

5.49.3 Concept

Note: What is the difference between world, dimension and dimension type?

Well, in the early stage of minecraft, a **world** only support **single-dimension**, which means 1 world only contains 1 dimension.
But now, 1 world supports multi dimension. Sometimes, you will see **world** and **dimension** means the same thing.
But clearer, we say: 1 world can contain 1 or more dimension, and each dimension has its **dimension type**.

Usually, you can say a mod adds extra dimension type and **creates an extra dimension** with that dimension type instead of **creating extra world**.
See also: https://minecraft.wiki/w/Dimension_definition
See also: https://minecraft.wiki/w/Dimension_type

Note: The dimension and dimension types in vanilla minecraft

In vanilla minecraft, 1 world contains 3 dimensions:

1. minecraft:overworld
2. minecraft:the_nether
3. minecraft:the_end

You can see the dimensions of a world in **world/level.dat** file.

A dimension type is used to create dimensions, the vanilla minecraft has the following dimension type:

1. minecraft:overworld
2. minecraft:overworld_caves
3. minecraft:the_nether
4. minecraft:the_end

The file **server.properties** is used for **the only and default world**.

5.49.4 Configuration

blacklist The dimensions in the blacklist will not be operated by this module. Use blacklist to avoid mis-operation.

5.49.5 Example

Example: Create an extra the_nether dimension

```
/world create my_nether minecraft:the_nether
```

Example: Delete the extra dimension

```
/world delete fuji:my_nether
```

Example: Reset the extra dimension with random seed

```
/world reset fuji:my_nether
```

Example: Specify a seed for an extra dimension.

```
/world create my_nether --seed 1234567890 minecraft:the_nether
/world reset --useTheSameSeed true fuji:my_nether
```

Tip: Make a resource-world that automatically reset every day

You can use command_scheduler module to execute `/world reset` command automatically.

5.50 world_downloader

5.50.1 Purpose

This module allows a player to download nearby chunks.

5.50.2 Command

- `/download`

5.50.3 Configuration

url_format The url format used to broadcast.

port The port used for downloader http-service.

bytes_per_second_limit Max download speed for each connection.

context_cache_size Max download request saved in the memory.

Chapter 6

Q&A

6.1 Where is the configuration files?

As a convention, all the files are placed in `config/fuji/` directory.

6.2 What is .json file?

A json file is a text file, whose name normally ends with **.json**.

6.3 How can I edit a configuration file?

To ensure the `readable` and `transparent`, most of the files are saved as `pure text format`. You can open them with a `text editor`.

Tip: Use a moderen text editor.

Some files may have a large number of lines, so it's highly recommended to use a `modern` text editor, which can highlight symbols and reveal the structure of the file, such as:

1. Visual Studio Code
2. Visual Studio Code - Web Online Editor
3. Vim
4. Emacs
5. Sublime Text

6.4 What is .dat file?

The file whose name ends with **.dat** are the `vanilla minecraft NBT format file`. To open such a file, you need to use a `NBT Editor`, such as `NBTExplorer`.

6.5 How to update fuji to a new version?

6.5.1 Backup the data

Back up the `config/fuji` directory.

6.5.2 Test the new version in your test-environment

Put the new version of fuji into `mods/` directory, start the server, and adjust the configuration to what you want.

Warning: Don't test new changes directly in your production-environment

It's highly recommended to setup a test-environment for a network-maintainer, so that you can test and tweak installed mods into what you want, and avoid un-expected situations.

6.5.3 Apply the changes to production-environment

Now, it's ready to apply the changes to your production-environment.

6.6 Fuji conflits with one of my mods.

You can `disable` the `conflicting module`. If possible, create an issue at fuji issue page, so that we can solve this later.

6.7 Can I ask the forge or neoforge support?

We have no plan for forge or neoforge platform.However, you can try running this mod via **sinytra-connector** mod. It's tested the mod works in the following environment (105/105 modules works):

```
Loader: NeoForge v21.1.73:server for Minecraft 1.21.1
Mods:
- connector-2.0.0-beta.3+1.21.1-full.jar
- forgified-fabric-api-0.104.0+2.0.15+1.21.1.jar
- fuji-4.3.0-5518b05201-mc1.21.jar
```

6.8 How can I report bugs or suggest new features?

You can create an issue at fuji issue page

Chapter 7

Transformer

7.1 Command Transformer

The following commands can be transformed:

1. /blockcycling = /give <player> minecraft:debug_stick
2. /blockinfo = use the command /info block provided by carpet mod.
3. /blocknbt = /data get block
4. /entityinfo = /data get entity
5. /entitynbt = /data get entity
6. /customrecipe = recipe editor
7. /exp = /experience
8. /gm, /gms, /gms = /gamemode
9. /findbiome = /locate biome
10. /tempfly = use luckperms temporal permission
11. /flyspeed = /attribute <player> minecraft:generic.flying_speed
12. /walkspeed = /attribute <player> minecraft:movement_speed
13. /maxhealth = /attribute <player> minecraft:generic.max_health
14. /groundclean = /kill @e[type=...]
15. /spawner = interact spawner block with spawn egg
16. /spawnmob = /summon
17. /shoot = /summon with motion
18. /smite, /thunder = /summon minecraft:lighting_bolt

19. /stats, /statsedit = all stats files are located in `world/stats`
20. /tree = use `tree brush` from `world-edit mod`
21. /unbreakable = /enchant `<player>` minecraft:unbreaking
22. /item_editor = /give with command generator, or use some client-side mod to edit items with commands.
23. /replaceblock = /setblock
24. chunks loading/unloading commands = These commands will not bring performance improvements.
25. /note = /attachment get note
26. /... all = /foreach ...
27. /cuff = use `anti-build module` with `string permission`
28. /cheque = use `command-attachment module` to make `magic item`

Tip: Use command bundle module to create transformed commands

See command_bundle

7.2 Command Generator

You can generate complex and powerful commands using:

1. https://www.digminecraft.com/generators/
2. https://minecraft.tools/en/public-gallery.php

Chapter 8

Development

8.1 Setup the development environment

1. Clone the source:

```
git clone https://github.com/sakurawald/fuji.git
```

2. Change the working-directory:

```
cd fuji
```

3. Compile the source:

```
./gradlew build
```

Chapter 9

Suggestion

9.1 Suggestion on server-side mods

9.1.1 Explanation

Here are some mods that existing and recommended to use, which can make your life easier in fabric server-side crafting.

Note that fuji doesn't require these mods installed to work, and some of these mods have the same functionality as fuji.

If you want to taste something different, and if some of the mods provides a better experience than fuji, then you can just disable the module in fuji, and go use it.

Tip: Decide according to your situation

The suggestion is personal, you should make decision according to your preference.

9.1.2 Server-side mode list

1. blue map
2. anti xray
3. melius commands
4. vanilla permission
5. vanish
6. villager config
7. world-edit
8. essential commands
9. missions
10. luckperms
11. tab
12. armor stand editor
13. ban hammer
14. image2map
15. polydecorations
16. sleep warp
17. styled chat
18. styled nickname
19. styled player list
20. styled sidebar
21. universal graves
22. universal shop
23. goml
24. polydex
25. polymania
26. head index
27. inv view
28. ledger

29. falling tree
30. double doors
31. skin restorer
32. husk homes
33. yet another world protector
34. krypton
35. sit
36. stack deobf
37. lithium
38. let me despawn
39. carpet
40. mod viewer
41. simple voice chat
42. mini motd
43. spark
44. polyfactory
45. ouch
46. chunky
47. afk plus
48. taterzens