IISE Transactions LaTeX Template

John Doe [a] and Jane Roe [b]
[a] Department, University, City, Country
[b] Department, University, City, Country

Abstract

This document provides a LaTeX template for *IISE Transactions*. Your paper should be compiled in the following order: title; abstract; keywords; main text, including an introduction and a conclusion or summary; acknowledgments; declaration of interest statement; references; appendices (as appropriate). Figures and tables should be inserted into the text as close to first mention as possible (NOT appended to the end of the manuscript). In-text citations and the reference list must follow *IISE Transactions* guidelines. Use 11 point font, 1 inch margins, and double-spacing for the manuscript. A typical paper for this journal should be no more than 30 pages in manuscript format, counting from the title page to references. Appendices should be included as supplemental online materials. Do not use footnotes. *IISE Transactions* uses a double-blind review process. Please make sure that you submit the **blind version** of your manuscript, which does not contain any information identifying the authors. This includes removing the authors information on the title page as well as the information that may be identifying in the Acknowledgment section.

We strongly encourage authors to address the following three questions in their **abstract**, preferably following the order shown: (1) Research problem statement: what is the research problem to be addressed? (2) Methods and results: how do the authors address the research problem and what are the main results? (3) Insights and implications: What have the authors learned (as opposed to what they did, which is covered in point (2)) from conducting this research? What is the knowledge gained and why does it matter? The abstract should be written in **a single paragraph**..

We thank you for your attention to these details.

Keywords: IISE Transactions; LaTeX; Manuscript format; Taylor & Francis.

Contents

1 Documentation conventions **7**

2 Introduction **8**

2.1 Destiny . . . 8

2.2 Response . . . 8

2.3 Operation . . . 9

2.4 Arguments . . . 9

2.4.1 Character . . . 9

2.4.2 Integer . . . 9

2.4.3 Boolean . . . 9

2.4.4 Float . . . 10

2.4.5 String . . . 10

2.4.6 Array . . . 10

2.4.7 File . . . 11

2.4.8 Server type . . . 11

2.4.9 Block . . . 11

2.4.10 Item . . . 12

3 Server manager petition **13**

3.1 Start server operation . . . 13

3.1.1 Maps . . . 15

3.1.2 Plugins . . . 16

3.1.3 Server version . . . 17

3.1.4 Config files . . . 17

3.2 Server started notification . . . 18

3.3 Error notification . . . 18

4 Server petition **19**

4.1 Server petition group . . . 19

4.2 Server petition operation . . . 20

4.3 Base operations . . . 20

4.3.1 Server stop operation 20

4.3.2 Server stopped notification 21

4.3.3 Server started notification 21

4.3.4 Whitelist player operation 21

4.3.5 OP player operation 22

4.3.6 Error notification 22

4.4 Performance operations 23

4.5 WorldGuard operations 23

4.6 Residence operations 23

5 ? petition 24

5.1 *Subsection heading 3.1* 24

5.1.1 *Sub-subsection heading 3.1.1* 24

5.2 *Subsection heading 3.2* 24

5.3 *Subsection heading 3.3* 24

6 Revision history 25

A Blocks 26

A.1 Material modifiers 26

A.1.1 Unused modifiers 26

A.1.2 Age 26

A.1.3 Attachment 27

A.1.4 Axis 27

A.1.5 Berries 27

A.1.6 Bites 28

A.1.7 Candles 28

A.1.8 Charges 28

A.1.9 Conditional 28

A.1.10 Delay 28

A.1.11 Down 29

A.1.12 North, South, East and West 29

A.1.13 Up 30

A.1.14 Eggs 30

A.1.15 Waterlogged 38

A.2 Material modifiers concatenation 41

References **41**

Figures

2.1 Packet structure . . . 8
2.2 True packet with the LSB at 1 . . . 10
2.3 True packet with all bits at 1 . . . 10
2.4 Structure of a String . . . 10
2.5 Example of a string array . . . 11
2.6 File structure . . . 12
3.1 Server manager petition structure . . . 13
3.2 Start server petition structure . . . 14
3.3 Start server response structure . . . 15
3.4 Start server error response structure . . . 15
3.5 Usual plugin structure . . . 17
3.6 File plugin structure . . . 17
3.7 Server started notification structure . . . 18
3.8 Error notification structure . . . 18
4.1 Server petition structure . . . 19
4.2 Implemented group response structure . . . 20
4.3 Stop server operation structure . . . 21
4.4 Server stopped response structure . . . 21
4.5 Server started response structure . . . 22
4.6 Whitelist player operation structure . . . 22
4.7 OP player operation structure . . . 22
A.1 Modifier concatenation . . . 42

Tables

2.1 DST bits meaning 8
4.1 Extended types 20
6.1 Revision history 25
A.1 Unused Spigot BlockData's modifiers 42
A.2 Unused Spigot BlockData's modifiers on certain blocks 43

1 Documentation conventions

...

abbreviations

2 Introduction

explicar los distintos protocolos que se hablaran a continuacion

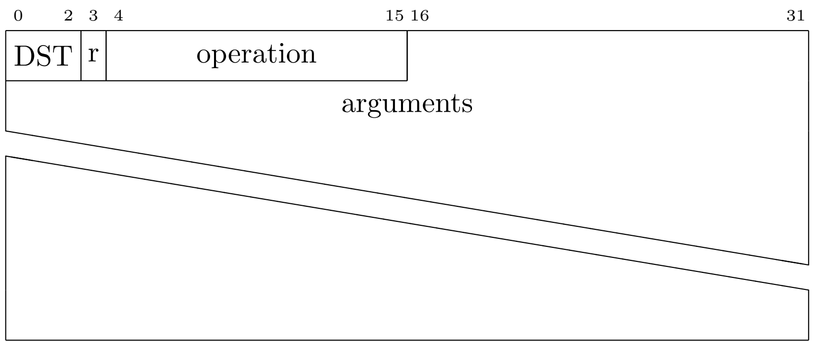

Figure 2.1: Packet structure

2.1 Destiny

explain

reference to the interconnected blocks

DST[2]	DST[1]	DST[0]	Destination
0	0	0	ServerManagerPetition
0	0	1	ServerPetition
0	1	0	ClientConnectorPetition
0	1	1	ClientPetition
1	X	X	*Reserved*

Table 2.1: DST bits meaning

2.2 Response

Some of the petitions have return objects. Those petitions will return to the sender (Tester-Connector) with the same code, but with a '1' on the Response parameter. In that case, the parameter Destiny now means 'Origin'.

Some petitions have async "returns" (for example: examples). Those will be sent using petitions without return's operations (so, petitions without a mirror petition with a '1' as

Response), marked as responses (Response bit at '1').

2.3 Operation

The Operation parameter specifies the desired request. Those change according to the Destiny, so they will be discussed in more detail in their respective sections.

The only exception is the all-zeroes operation (0b000000000000) which represents a NOP request. That way, if you need to perform a long test, you won't be explain the 'kicked by inactivity' concept kicked by inactivity if you send this request every few minutes.

2.4 Arguments

The Arguments parameter specifies the arguments (if any) to the *Operation* request. Those change according to the Destiny, so the amount of arguments, and their types and order will be discussed in more detail in their respective sections.

Now there will be discussed the most common data types, so they will be independent of any programming language.

2.4.1 Character

Characters are sent as a 1-byte integer, representing its ASCII ref? value.

2.4.2 Integer

Integers are signed 4-bytes integers.

2.4.3 Boolean

Booleans are 1-bit element that represents *true* (0b1), or *false* (0b0).

For alignment define? reasons, booleans will be sent as 1-byte element. To avoid misunderstandings, let's define *false* as 0x00, and *true* as '*not* define? *false*'. That way, this two packets are valid *true* elements:

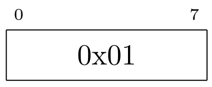

Figure 2.2: True packet with the LSB at 1

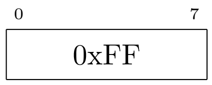

Figure 2.3: True packet with all bits at 1

2.4.4 Float

Floats are 4-bytes floating-point numbers. They are represented following the IEEE 754[1].

2.4.5 String

Strings are arrays of characters. Refer to the respective subsections for more information.

2.4.6 Array

Arrays are a set of n elements of the same type.

The structure is a 2-byte first (0..7) MSB, then (8..15) LSB integer (representing the number of elements, n), followed by n elements of the same type. As a note here, by representing the size with a 2-byte integer the maximum number of elements per array is 65,535.

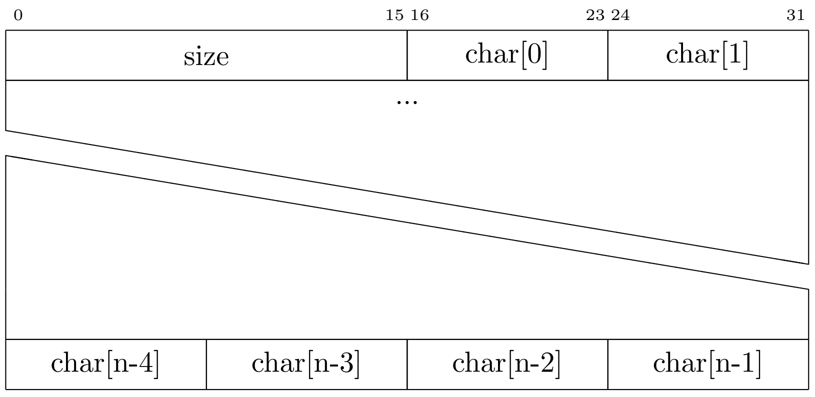

Figure 2.4: Structure of a String

[1]This standard should be used by C, Java and Python. cite?

Arrays can be multidimensional, holding n arrays of the same type. It's worth mentioning that they don't have to be arrays of the same length, as can be seen in Figure 2.5, Example of a string array.

0	15	16 23	24 31
2 [number of arrays]		5 [str[0]'s length]	
h	e	l	l
o	6 [str[1]'s length]		w
o	r	l	d
!	next type		

Figure 2.5: Example of a string array

2.4.7 File

Similar to the Array, a File is a name (String), followed by a group of bytes.

The problem here is that if we stick with the Array structure, the maximum size of a file will be around 8kB. To solve this, the File structure implements some kind of 'extended array', that extends the 'size' parameter to 32 bits. That way, the file size restriction by protocol definition[2] is 4GB.

2.4.8 Server type

The Server type specifies the Minecraft server.

As a standard, we only support Spigot (*Spigot* (n.d.)) and Paper (*PaperMC* (n.d.)), but for major compatibility this parameter is a String specifying the server type.

2.4.9 Block

...

[2]Besides defining here what's allowed, remember that this packet will be inside a TCP payload definition?. This means that the maximum file size will be probably redefined by the machine's TCP firewalls.

[3]The path must be relative, and you can't go outside the Server directory (using '../'). Both '' and './' means the root of the Server directory.

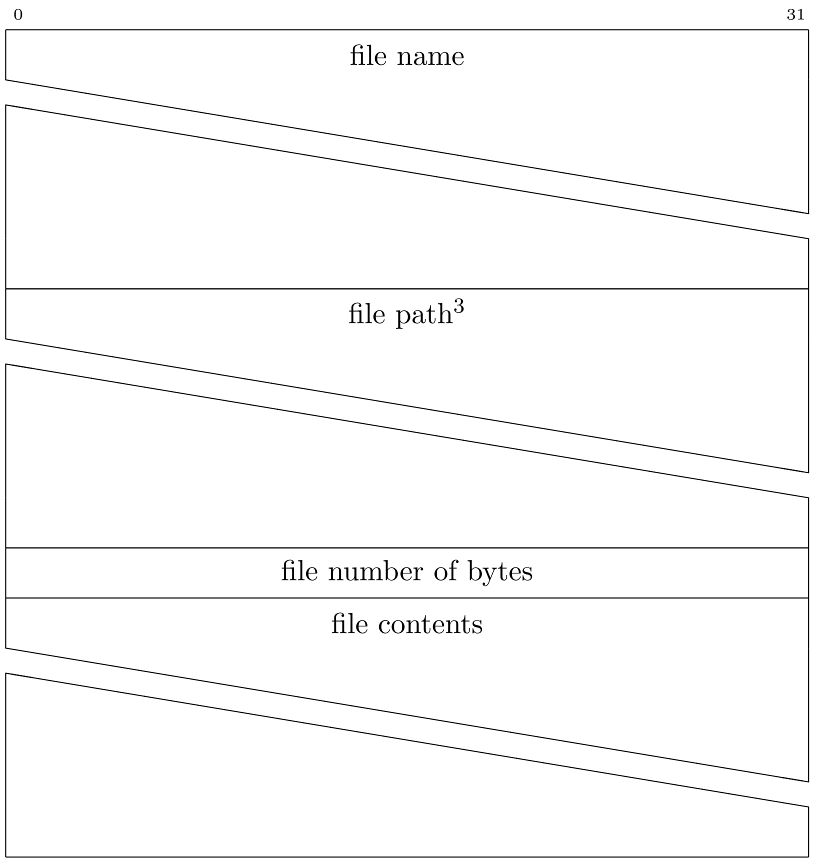

Figure 2.6: File structure

2.4.10 Item

...

3 Server manager petition

...

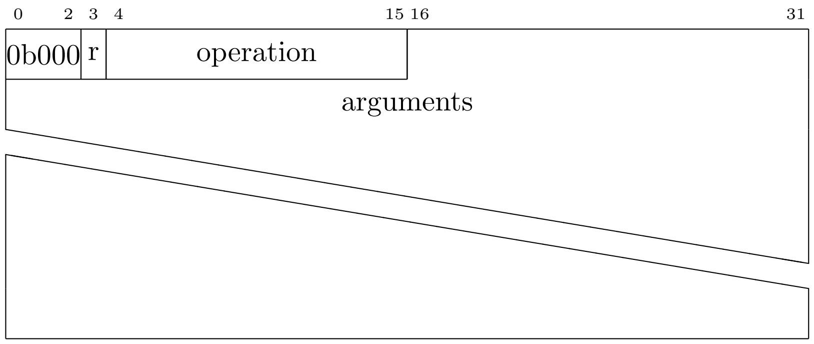

Figure 3.1: Server manager petition structure

Table of operations

You don't have to implement the NOP operation in this destiny block because the timeout happens inside the Server petition block. That is, if you don't call operations (or send NOPs) to the Server petition for a long time, the server will stop, and because the server stopped the Server manager will close the established connection.

3.1 Start server operation

...

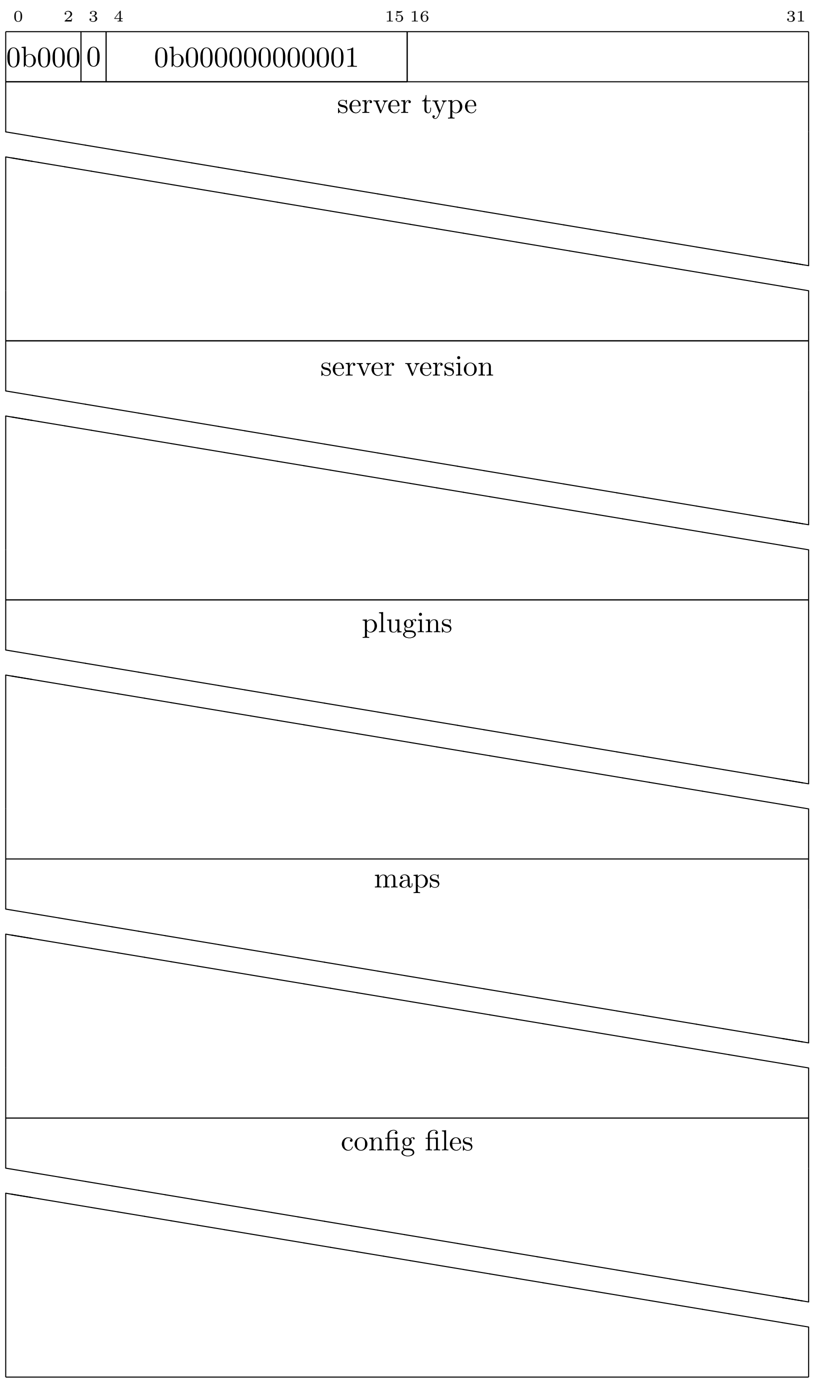

Figure 3.2: Start server petition structure

Once a 'start server' request is received the program should create a server with the specified arguments, and return its IP:Port (for example, '127.0.0.1:25565', a 15-characters string; see Figure 3.3, Start server response structure). The IP to send the Server Petitions is the same, but the next port (IP:<port+1>).

If it's not possible to create it (for example: one argument is invalid, the user sent a plugin when it's specified that only Usual Plugins are allowed explain, or there's no free servers of that type), then an empty IP is returned (see Figure 3.4, Start server error response structure).

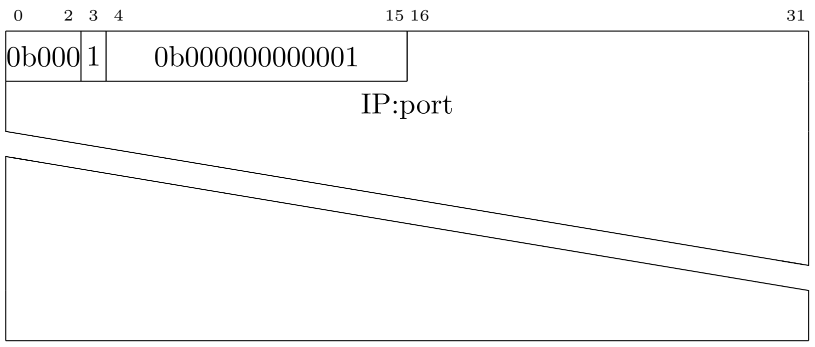

Figure 3.3: Start server response structure

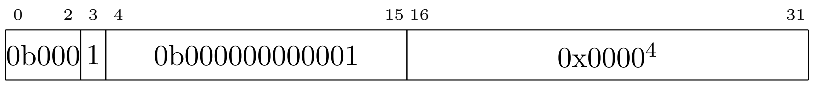

Figure 3.4: Start server error response structure

3.1.1 Maps

Array of maps (worlds; Map[]). To have more information about arrays check the subsection 2.4.6, Array.

About the Map type, Minecraft is divided on different worlds (*World - Minecraft Wiki* (n.d.)). By default there's only three, but with some plugins this number can increase.

In order to properly test some plugins, there may be needed some kind of known place. To avoid overusing the Set block operation link you can send using this argument your(s) world(s).

[4]Being the argument an array, the first 2 bytes specifies its size. As we must return an empty array, the argument should be exactly 16 zeroes.

Map in more detail

3.1.2 Plugins

Array of plugins (Plugin[]). To have more information check the subsection 2.4.6, Array.

About the Plugin type, there's three types of plugins:

1. Usual plugins

   The Usual plugins are plugins that you expect everyone to have for being extremely common, like WorldGuard (*WorldGuard* (n.d.)), or to allow the user to test plugins with Premium plugins[5] dependencies. This allows both security and performance.

   Something to highlight is the fact that, as mentioned in the operation Allows non usual plugins reference, some ServerManager will only allow plugins that are already in the machine.

   As can be seen in the Figure 3.5, Usual plugin structure, the first argument (that specifies the Plugin type) is 0x00.

   The plugin version is optional, and can't be specified in the parameter *name*. If no version is provided (an empty string) then the Server Manager will pick the plugin with the highest version that is compatible with the desired server version.

2. Uploaded plugins

   The Uploaded plugins are plugins available in some website, thus can be sent through an URL.

   structure?

3. File plugins

   File plugins are plugins that are non-usual and aren't uploaded in any website, so they must be sent as a file.

   As can be seen in the Figure 3.6, File plugin structure, the first argument (that specifies the Plugin type) is 0x02.

[5]Premium plugins are paid plugins. For that reason, only the purchaser can download them (so you can't send a link to the plugin), and sending them through the internet via file upload may not be legal, so the plugin must be already downloaded in the machine.

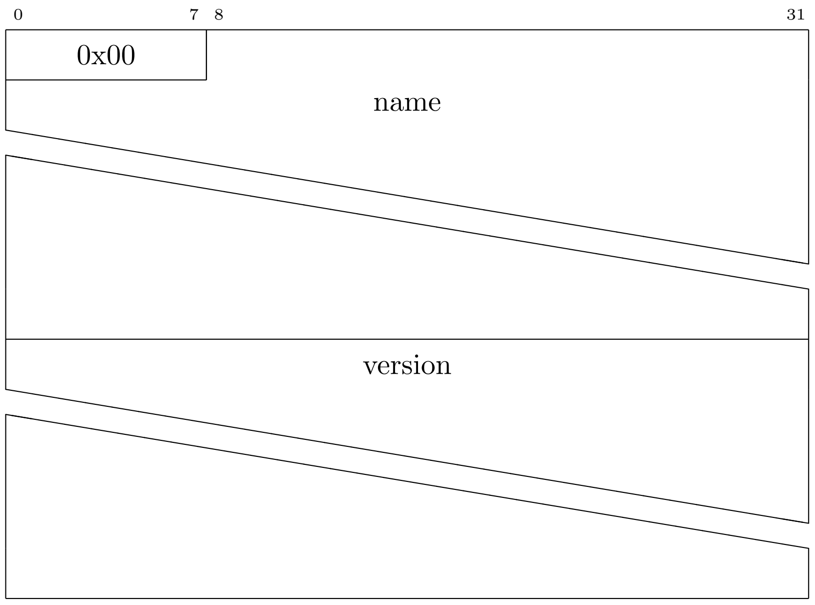

Figure 3.5: Usual plugin structure

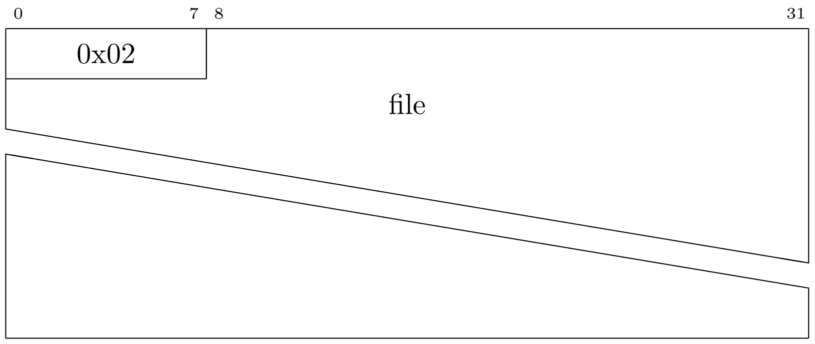

Figure 3.6: File plugin structure

mixed plugin types example?

3.1.3 Server version

String specifying the server type's version. For example, '1.12.2'.

3.1.4 Config files

...

3.2 Server started notification

After a Start server operation the server will start. Due to the unpredictable amount of time that the server takes to start up you'll receive a Server started notification once the server socket is available.

You may notice that there's another Server started notification under the Server petition section. That notification goes to the ServerManager ref?, while this goes to the Tester ref?. Also, the Server one have a token that is only shared between Server and the ServerManager, and the Tester doesn't have to know it too.

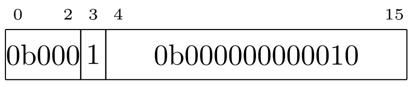

Figure 3.7: Server started notification structure

3.3 Error notification

...

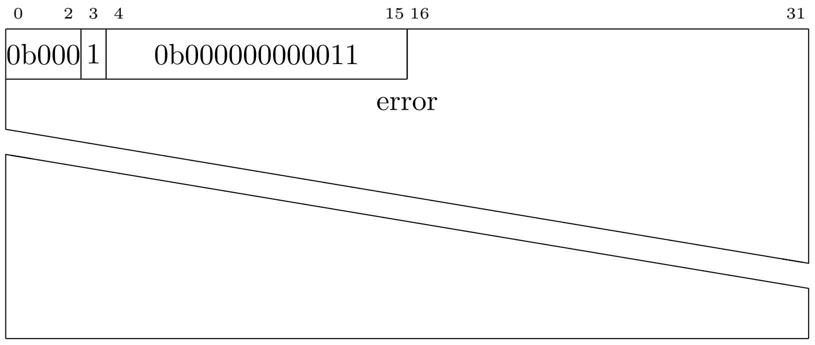

Figure 3.8: Error notification structure

4 Server petition

...

The server petitions are a bit different from the rest. The server petitions are designed in a way that everyone have some common operations, and then you can add some others optionally (and even non-standard ones). We'll define this 'set of operations' as groups.

For that reason, the operation field (defined on the Figure 2.1, Packet structure) becomes the group, and then the operation is defined on the next 2 bytes, as shown in the Figure 4.1, Server petition structure.

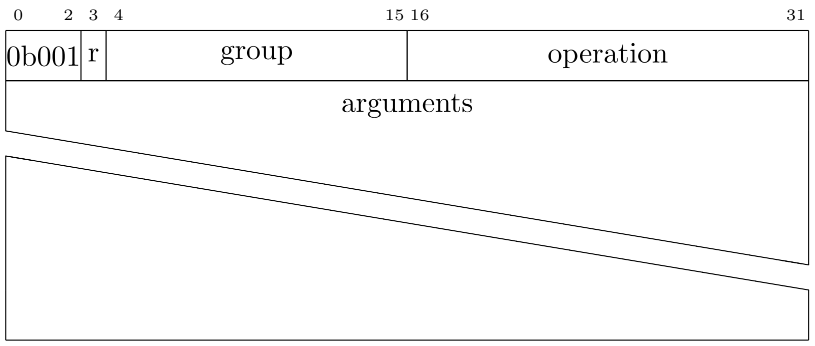

Figure 4.1: Server petition structure

4.1 Server petition group

The group tells which kind of petitions we're talking about.

The MSB abbreviation? tells if the group is one of the standards, thus must be followed by specification, or if it's non-standard, so the petition can be whatever the user want it to be. This is useful if you want to implement a petition not followed by the standard, or if the petition only makes sense in your personal environment.

The 0b000000000001 group represents the 'base group'. This group implements some basic operations, and must be implemented. All the others are optional.

If you've implemented an extended type and you believe that it makes sense to be part of the standard contact contacto@rogermiranda1000.com to reserve one of the addresses.

[6] As stated on the subsection 2.3, Operation, the all-zeroes operation represents a NOP request.

type[15]	type[14..4]	Extended type
0	0b00000000000	NOP[6]
0	0b00000000001	Base operations
0	0b00000000010	Performance operations
0	0b00000000011	WorldGuard operations
0	0b00000000100	Residence operations
1	XXXXXXXXXXX	Reserved for internal use

Table 4.1: Extended types

4.2 Server petition operation

Like the parameter Operation, it specifies the desired request. For more information, refer to the subsection 2.3, Operation.

The only reserved operation is the all-zeroes operation (0x0000). It represents the question 'is this extended petition implemented?'. The server must response (with the response bit at 1) with *true* (group implemented on this machine) or *false* (unknown/unimplemented group), as it can be seen in Figure 4.2, Implemented group response structure.

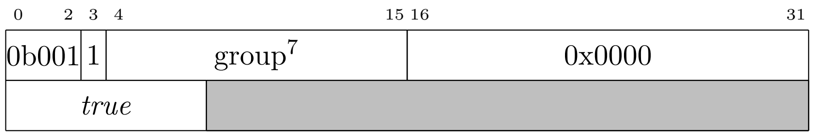

Figure 4.2: Implemented group response structure

4.3 Base operations

...

'is implemented' (all zeroes) optional

4.3.1 Server stop operation

...

[7]except for groups 0b000000000000 and 0b000000000001

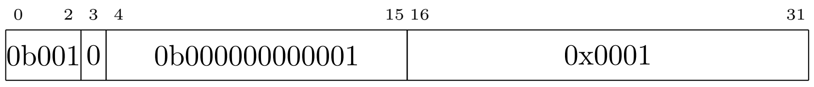

Figure 4.3: Stop server operation structure

4.3.2 Server stopped notification

... response to...

To have more information about the *server id* parameter check the Subsection 4.3.3, Server started notification.

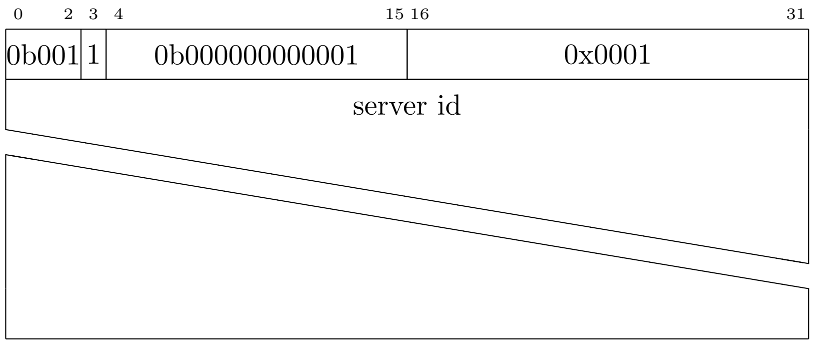

Figure 4.4: Server stopped response structure

4.3.3 Server started notification

This notification is sent to the Server Manager ref?, as a response for the Start server operation, thus not really a response of a Server's operation.

As one IP can have multiple servers, a string that identifies the server must be sent with the response. This argument can be whatever you want (for example, <server ip>:<server port> will be unique), but must be shared between both the Server Manager and the Server. For security reasons cite IP spoofing or similar (because the Tester ref? also knows the server's IP and port) a hash function is encouraged to be used.

4.3.4 Whitelist player operation

...

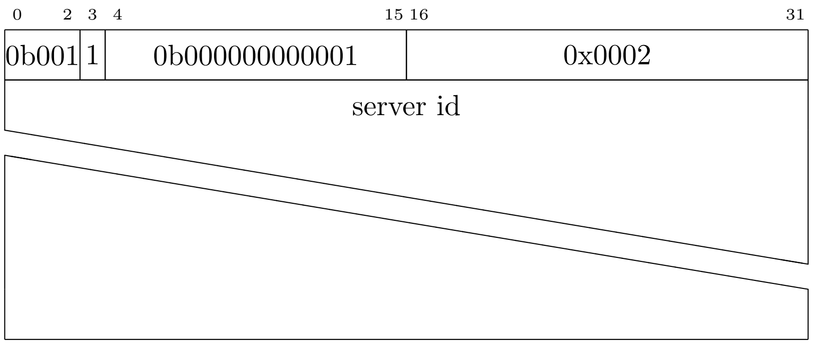

Figure 4.5: Server started response structure

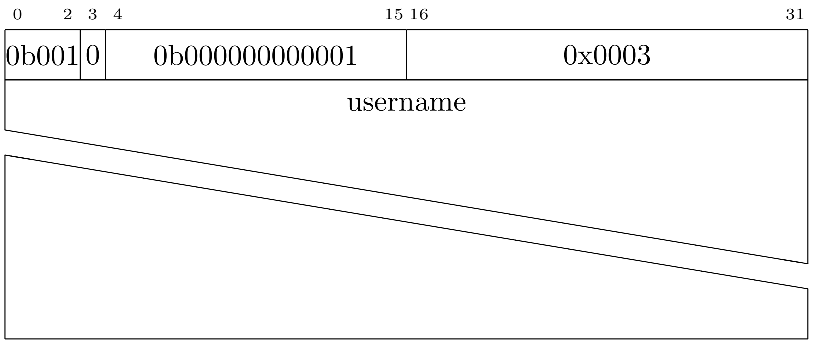

Figure 4.6: Whitelist player operation structure

4.3.5 OP player operation

...

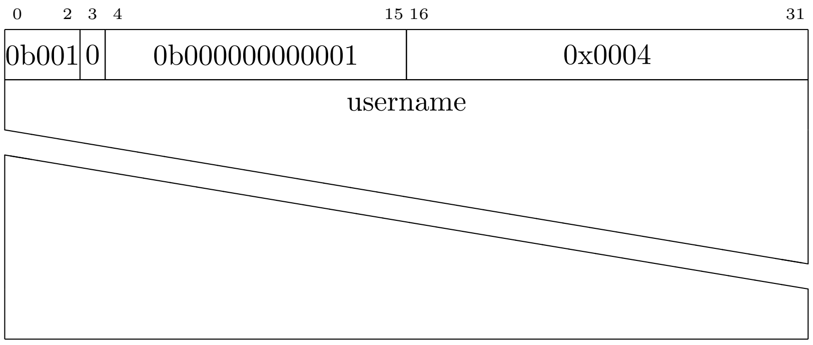

Figure 4.7: OP player operation structure

4.3.6 Error notification

...

4.4 Performance operations

...

4.5 WorldGuard operations

...

4.6 Residence operations

...

5 ? petition

First-level headings should be in bold.

5.1 ***Subsection heading 3.1***

Second-level headings should be in bold italics.

5.1.1 *Sub-subsection heading 3.1.1*

Third-level headings should be in italics.

5.2 ***Subsection heading 3.2***

5.3 ***Subsection heading 3.3***

6 Revision history

Date	Revision	Changes
date	1	Initial release.

Table 6.1: Revision history

A Blocks

For generating the blocks enum Spigot 1.19 was used. That means that all the block names *should* be the exact same as *Spigot - Enum Material* (n.d.).

A.1 Material modifiers

There's one downside on using Spigot's Material: it doesn't describes perfectly the block. In some aspects it will, for example, distinguish between wood types, but it won't differentiate between a wooden stair and a wooden stair with water.

That's why there's some prefixes and suffixes (that will be discussed in the following subsections) surrounding the original Spigot name, to make every possible Minecraft block combination appear in the block enum. Just to clarify, this has also been extracted from Spigot (all *Spigot - Interface BlockData* (n.d.)'s subinterfaces in Spigot 1.19).

A.1.1 Unused modifiers

There's some Spigot modifiers that beside existing it won't be imported because there aren't a distinguished block in their own. You can find those in Figure A.1, Unused Spigot BlockData's modifiers.

In addition to this, some modifiers applied to certain blocks doesn't change the block itself. Those are mentioned in Figure A.2, Unused Spigot BlockData's modifiers on certain blocks.

A.1.2 Age

Represents the different growth stages that a crop-like block can go through.

Defaults to 0.

BEETROOTS 0-3 BAMBOO 0-1 CARROTS 0-7 CHORUS_FLOWER 0/5 0 =¿ 0; 5 =¿ 1 (the other stages are the same block) COCOA 0-2 FROSTED_ICE 0-3 MELON_STEM 0-7 NETHER_WART 0-3 POTATOES 0-7 PUMPKIN_STEM 0-7 SWEET_BERRY_BUSH 0-3 WHEAT 0-7

A.1.3 Attachment

Denotes how the bell is attached to its block.

Defaults to floor.

BELL ceiling/double_wall/floor/single_wall

A.1.4 Axis

Represents the axis along whilst this block is oriented.

Except for NETHER_PORTAL (which defaults to x), it defaults to y.

NETHER_PORTAL x/z ACACIA_LOG x/y/z ACACIA_WOOD x/y/z BASALT x/y/z BIRCH_LOG x/y/z BIRCH_WOOD x/y/z BONE_BLOCK x/y/z CHAIN x/y/z CRIMSON_HYPHAE x/y/z CRIMSON_STEM x/y/z DARK_OAK_LOG x/y/z DARK_OAK_WOOD x/y/z DEEPSLATE x/y/z HAY_BLOCK x/y/z INFESTED_DEEPSLATE x/y/z JUNGLE_LOG x/y/z JUNGLE_WOOD x/y/z MANGROVE_LOG x/y/z MANGROVE_WOOD x/y/z MUDDY_MANGROVE_ROOTS x/y/z OAK_LOG x/y/z OAK_WOOD x/y/z OCHRE_FROGLIGHT x/y/z PEARLESCENT_FROGLIGHT x/y/z POLISHED_BASALT x/y/z PURPUR_PILLAR x/y/z QUARTZ_PILLAR x/y/z SPRUCE_LOG x/y/z SPRUCE_WOOD x/y/z STRIPPED_ACACIA_LOG x/y/z STRIPPED_ACACIA_WOOD x/y/z STRIPPED_BIRCH_LOG x/y/z STRIPPED_BIRCH_WOOD x/y/z STRIPPED_CRIMSON_HYPHAE x/y/z STRIPPED_CRIMSON_STEM x/y/z STRIPPED_DARK_OA
x/y/z STRIPPED_DARK_OAK_WOOD x/y/z STRIPPED_JUNGLE_LOG x/y/z STRIPPED_JUNGLE_WOO
x/y/z STRIPPED_MANGROVE_LOG x/y/z STRIPPED_MANGROVE_WOOD x/y/z STRIPPED_OAK_LO
x/y/z STRIPPED_OAK_WOOD x/y/z STRIPPED_SPRUCE_LOG x/y/z STRIPPED_SPRUCE_WOOD x/y/z STRIPPED_WARPED_HYPHAE x/y/z STRIPPED_WARPED_STEM x/y/z VERDANT_FROGLIGHT x/y/z WARPED_HYPHAE x/y/z WARPED_STEM x/y/z

A.1.5 Berries

Indicates whether the block has berries.

Defaults to false.

CAVE_VINES true/false CAVE_VINES_PLANT true/false

A.1.6 Bites

Represents the amount of bites which have been taken from this slice of cake.

Defaults to 0.

CAKE 0-6

A.1.7 Candles

Represents the number of candles which are present.

Defaults to 1.

BLACK_CANDLE 1-4 BLUE_CANDLE 1-4 BROWN_CANDLE 1-4 CANDLE 1-4 CYAN_CANDLE 1-4 GRAY_CANDLE 1-4 GREEN_CANDLE 1-4 LIGHT_BLUE_CANDLE 1-4 LIGHT_GRAY_CANDLE 1-4 LIME_CANDLE 1-4 MAGENTA_CANDLE 1-4 ORANGE_CANDLE 1-4 PINK_CANDLE 1-4 PURPLE_CANDLE 1-4 RED_CANDLE 1-4 WHITE_CANDLE 1-4 YELLOW_CANDLE 1-4

A.1.8 Charges

Represents the amount of times the anchor may still be used.

Defaults to 0.

RESPAWN_ANCHOR 0-4

A.1.9 Conditional

Denotes whether this command block is conditional or not.

Defaults to false.

CHAIN_COMMAND_BLOCK true/false COMMAND_BLOCK true/false REPEATING_COMMAND_BLC true/false

A.1.10 Delay

Propagation delay of a repeater.

Defaults to 1.

REPEATER 1-4

A.1.11 Down

Set which faces of the block textures are displayed on.

Except for BROWN_MUSHROOM_BLOCK, MUSHROOM_STEM and RED_MUSHROOM_BLOCK (which defaults to true), it defaults to false.

CHORUS_PLANT true/false GLOW_LICHEN true/false SCULK_VEIN true/false BROWN_MUSHROOM
true/false MUSHROOM_STEM true/false RED_MUSHROOM_BLOCK true/false

A.1.12 North, South, East and West

Set which faces of the block textures are displayed on.

east=false (ACACIA$_FENCE)east = false(BIRCH_FENCE)east = false(BLACK_STAINED_GLASS_P$
$false(BLUE_STAINED_GLASS_PANE)east = false(BROWN_STAINED_GLASS_PANE)east =$
$false(CHORUS_PLANT)east = false(CRIMSON_FENCE)east = false(CYAN_STAINED_GLASS_PANE$
$false(DARK_OAK_FENCE)east = false(FIRE)east = false(GLASS_PANE)east = false(GLOW_LICHEN$
$false(GRAY_STAINED_GLASS_PANE)east = false(GREEN_STAINED_GLASS_PANE)east =$
$false(IRON_BARS)east = false(JUNGLE_FENCE)east = false(LIGHT_BLUE_STAINED_GLASS_PANE)$
$false(LIGHT_GRAY_STAINED_GLASS_PANE)east = false(LIME_STAINED_GLASS_PANE)east =$
$false(MAGENTA_STAINED_GLASS_PANE)east = false(MANGROVE_FENCE)east =$
$false(NETHER_BRICK_FENCE)east = false(OAK_FENCE)east = false(ORANGE_STAINED_GLASS_P$
$false(PINK_STAINED_GLASS_PANE)east = false(PURPLE_STAINED_GLASS_PANE)east =$
$false(RED_STAINED_GLASS_PANE)east = false(SCULK_VEIN)east = false(SPRUCE_FENCE)east =$
$false(TRIPWIRE)east = false(VINE)east = false(WARPED_FENCE)east = false(WHITE_STAINE$
$false(YELLOW_STAINED_GLASS_PANE)east = none(ANDESITE_WALL)east = none(BLACKSTONE$
$none(BRICK_WALL)east = none(COBBLED_DEEPSLATE_WALL)east = none(COBBLESTONE_WALL$
$none(DEEPSLATE_BRICK_WALL)east = none(DEEPSLATE_TILE_WALL)east = none(DIORITE_WAL$
$none(END_STONE_BRICK_WALL)east = none(GRANITE_WALL)east = none(MOSSY_COBBLESTONE$
$none(MOSSY_STONE_BRICK_WALL)east = none(MUD_BRICK_WALL)east = none(NETHER_BRICK_WA$
$none(POLISHED_BLACKSTONE_BRICK_WALL)east = none(POLISHED_BLACKSTONE_WALL)east$
$none(POLISHED_DEEPSLATE_WALL)east = none(PRISMARINE_WALL)east = none(REDSTONE_W$
$none(RED_NETHER_BRICK_WALL)east = none(RED_SANDSTONE_WALL)east = none(SANDSTONE$
$none(STONE_BRICK_WALL)east = true(BROWN_MUSHROOM_BLOCK)east = true(MUSHROOM_STE$

$true(RED_MUSHROOM_BLOCK)$

A.1.13 Up

Set which faces of the block textures are displayed on.

Except for CHORUS_PLANT, FIRE, GLOW_LICHEN, SCULK_VEIN and VINE (which defaults to false), it defaults to true.

up=false (CHORUS$_PLANT)up = false(FIRE)up = false(GLOW_LICHEN)up =$
$false(SCULK_VEIN)up = false(VINE)up = true(ANDESITE_WALL)up = true(BLACKSTONE_WALL$
$true(BRICK_WALL)up = true(BROWN_MUSHROOM_BLOCK)up = true(COBBLED_DEEPSLATE_WAI$
$true(COBBLESTONE_WALL)up = true(DEEPSLATE_BRICK_WALL)up = true(DEEPSLATE_TILE_WA$
$true(DIORITE_WALL)up = true(END_STONE_BRICK_WALL)up = true(GRANITE_WALL)up =$
$true(MOSSY_COBBLESTONE_WALL)up = true(MOSSY_STONE_BRICK_WALL)up =$
$true(MUD_BRICK_WALL)up = true(MUSHROOM_STEM)up = true(NETHER_BRICK_WALL)up =$
$true(POLISHED_BLACKSTONE_BRICK_WALL)up = true(POLISHED_BLACKSTONE_WALL)up =$
$true(POLISHED_DEEPSLATE_WALL)up = true(PRISMARINE_WALL)up = true(RED_MUSHROOM_E$
$true(RED_NETHER_BRICK_WALL)up = true(RED_SANDSTONE_WALL)up = true(SANDSTONE_WAL$
$true(STONE_BRICK_WALL)$

A.1.14 Eggs

eggs=1 (TURTLE$_EGG)$

extended=false (PISTON) extended=false (STICKY$_PISTON)eye = false(END_PORTAL_FRAME)face$
$wall(ACACIA_BUTTON)face = wall(BIRCH_BUTTON)face = wall(CRIMSON_BUTTON)face =$
$wall(DARK_OAK_BUTTON)face = wall(GRINDSTONE)face = wall(JUNGLE_BUTTON)face =$
$wall(LEVER)face = wall(MANGROVE_BUTTON)face = wall(OAK_BUTTON)face =$
$wall(POLISHED_BLACKSTONE_BUTTON)face = wall(SPRUCE_BUTTON)face =$
$wall(STONE_BUTTON)face = wall(WARPED_BUTTON)facing = down(HOPPER)facing =$
$north(ACACIA_BUTTON)facing = north(ACACIA_DOOR)facing = north(ACACIA_FENCE_GATE)faci$
$north(ACACIA_STAIRS)facing = north(ACACIA_TRAPDOOR)facing = north(ACACIA_WALL_SIGN)f$
$north(ANDESITE_STAIRS)facing = north(ANVIL)facing = north(ATTACHED_MELON_STEM)facin$
$north(ATTACHED_PUMPKIN_STEM)facing = north(BARREL)facing = north(BEEHIVE)facing =$
$north(BEE_NEST)facing = north(BELL)facing = north(BIG_DRIPLEAF)facing =$

$north(BIG_DRIPLEAF_STEM)facing = north(BIRCH_BUTTON)facing = north(BIRCH_DOOR)facing$

$north(BIRCH_FENCE_GATE)facing = north(BIRCH_STAIRS)facing = north(BIRCH_TRAPDOOR)fa$

$north(BIRCH_WALL_SIGN)facing = north(BLACKSTONE_STAIRS)facing = north(BLACK_BED)faci$

$north(BLACK_GLAZED_TERRACOTTA)facing = north(BLACK_WALL_BANNER)facing =$

$north(BLAST_FURNACE)facing = north(BLUE_BED)facing = north(BLUE_GLAZED_TERRACOTTA)$

$north(BLUE_WALL_BANNER)facing = north(BRAIN_CORAL_WALL_FAN)facing = north(BRICK_STAI$

$north(BROWN_BED)facing = north(BROWN_GLAZED_TERRACOTTA)facing = north(BROWN_WALL$

$north(BUBBLE_CORAL_WALL_FAN)facing = north(CAMPFIRE)facing = north(CARVED_PUMPKI$

$north(CHAIN_COMMAND_BLOCK)facing = north(CHEST)facing = north(CHIPPED_ANVIL)facing$

$north(COBBLED_DEEPSLATE_STAIRS)facing = north(COBBLESTONE_STAIRS)facing =$

$north(COCOA)facing = north(COMMAND_BLOCK)facing = north(COMPARATOR)facing =$

$north(CREEPER_WALL_HEAD)facing = north(CRIMSON_BUTTON)facing = north(CRIMSON_DOO$

$north(CRIMSON_FENCE_GATE)facing = north(CRIMSON_STAIRS)facing = north(CRIMSON_TRAP$

$north(CRIMSON_WALL_SIGN)facing = north(CUT_COPPER_STAIRS)facing = north(CYAN_BED)faci$

$north(CYAN_GLAZED_TERRACOTTA)facing = north(CYAN_WALL_BANNER)facing =$

$north(DAMAGED_ANVIL)facing = north(DARK_OAK_BUTTON)facing = north(DARK_OAK_DOOR)fa$

$north(DARK_OAK_FENCE_GATE)facing = north(DARK_OAK_STAIRS)facing = north(DARK_OAK_TRA$

$north(DARK_OAK_WALL_SIGN)facing = north(DARK_PRISMARINE_STAIRS)facing =$

$north(DEAD_BRAIN_CORAL_WALL_FAN)facing = north(DEAD_BUBBLE_CORAL_WALL_FAN)facing =$

$north(DEAD_FIRE_CORAL_WALL_FAN)facing = north(DEAD_HORN_CORAL_WALL_FAN)facing =$

$north(DEAD_TUBE_CORAL_WALL_FAN)facing = north(DEEPSLATE_BRICK_STAIRS)facing =$

$north(DEEPSLATE_TILE_STAIRS)facing = north(DIORITE_STAIRS)facing = north(DISPENSER)f$

$north(DRAGON_WALL_HEAD)facing = north(DROPPER)facing = north(ENDER_CHEST)facing =$

$north(END_PORTAL_FRAME)facing = north(END_STONE_BRICK_STAIRS)facing =$

$north(EXPOSED_CUT_COPPER_STAIRS)facing = north(FIRE_CORAL_WALL_FAN)facing =$

$north(FURNACE)facing = north(GRANITE_STAIRS)facing = north(GRAY_BED)facing =$

$north(GRAY_GLAZED_TERRACOTTA)facing = north(GRAY_WALL_BANNER)facing =$

$north(GREEN_BED)facing = north(GREEN_GLAZED_TERRACOTTA)facing = north(GREEN_WALL_B$

$north(GRINDSTONE)facing = north(HORN_CORAL_WALL_FAN)facing = north(IRON_DOOR)facing$

$north(IRON_TRAPDOOR)facing = north(JACK_OLANTERN)facing = north(JUNGLE_BUTTON)faci$

$north(JUNGLE_DOOR)facing = north(JUNGLE_FENCE_GATE)facing = north(JUNGLE_STAIRS)faci$

$north(JUNGLE_TRAPDOOR)facing = north(JUNGLE_WALL_SIGN)facing = north(LADDER)facing =$

$north(LECTERN)facing = north(LEVER)facing = north(LIGHT_BLUE_BED)facing =$

$north(LIGHT_BLUE_GLAZED_TERRACOTTA)facing = north(LIGHT_BLUE_WALL_BANNER)facing =$

$north(LIGHT_GRAY_BED)facing = north(LIGHT_GRAY_GLAZED_TERRACOTTA)facing =$

$north(LIGHT_GRAY_WALL_BANNER)facing = north(LIME_BED)facing = north(LIME_GLAZED_TERR$

$north(LIME_WALL_BANNER)facing = north(LOOM)facing = north(MAGENTA_BED)facing =$

$north(MAGENTA_GLAZED_TERRACOTTA)facing = north(MAGENTA_WALL_BANNER)facing =$

$north(MANGROVE_BUTTON)facing = north(MANGROVE_DOOR)facing = north(MANGROVE_FEN$

$north(MANGROVE_STAIRS)facing = north(MANGROVE_TRAPDOOR)facing = north(MANGROV$

$north(MOSSY_COBBLESTONE_STAIRS)facing = north(MOSSY_STONE_BRICK_STAIRS)facing =$

$north(MOVING_PISTON)facing = north(MUD_BRICK_STAIRS)facing = north(NETHER_BRICK_STA$

$north(OAK_BUTTON)facing = north(OAK_DOOR)facing = north(OAK_FENCE_GATE)facing =$

$north(OAK_STAIRS)facing = north(OAK_TRAPDOOR)facing = north(OAK_WALL_SIGN)facing =$

$north(ORANGE_BED)facing = north(ORANGE_GLAZED_TERRACOTTA)facing =$

$north(ORANGE_WALL_BANNER)facing = north(OXIDIZED_CUT_COPPER_STAIRS)facing =$

$north(PINK_BED)facing = north(PINK_GLAZED_TERRACOTTA)facing = north(PINK_WALL_BANN$

$north(PISTON)facing = north(PISTON_HEAD)facing = north(PLAYER_WALL_HEAD)facing =$

$north(POLISHED_ANDESITE_STAIRS)facing = north(POLISHED_BLACKSTONE_BRICK_STAIRS)$

$north(POLISHED_BLACKSTONE_BUTTON)facing = north(POLISHED_BLACKSTONE_STAIRS)fa$

$north(POLISHED_DEEPSLATE_STAIRS)facing = north(POLISHED_DIORITE_STAIRS)facing =$

$north(POLISHED_GRANITE_STAIRS)facing = north(PRISMARINE_BRICK_STAIRS)facing =$

$north(PRISMARINE_STAIRS)facing = north(PURPLE_BED)facing = north(PURPLE_GLAZED_TER$

$north(PURPLE_WALL_BANNER)facing = north(PURPUR_STAIRS)facing = north(QUARTZ_STAIRS)$

$north(REDSTONE_WALL_TORCH)facing = north(RED_BED)facing = north(RED_GLAZED_TERRACO$

$north(RED_NETHER_BRICK_STAIRS)facing = north(RED_SANDSTONE_STAIRS)facing =$

$north(RED_WALL_BANNER)facing = north(REPEATER)facing = north(REPEATING_COMMAND_B$

$north(SANDSTONE_STAIRS)facing = north(SKELETON_WALL_SKULL)facing =$

$north(SMALL_DRIPLEAF)facing = north(SMOKER)facing = north(SMOOTH_QUARTZ_STAIRS)fac$

$north(SMOOTH_RED_SANDSTONE_STAIRS)facing = north(SMOOTH_SANDSTONE_STAIRS)facing$

$north(SOUL_CAMPFIRE)facing = north(SOUL_WALL_TORCH)facing = north(SPRUCE_BUTTON)fac$

$north(SPRUCE_DOOR)facing = north(SPRUCE_FENCE_GATE)facing = north(SPRUCE_STAIRS)faci$

$north(SPRUCE_TRAPDOOR)facing = north(SPRUCE_WALL_SIGN)facing = north(STICKY_PISTON)$

$north(STONECUTTER)facing = north(STONE_BRICK_STAIRS)facing = north(STONE_BUTTON)fa$

$north(STONE_STAIRS)facing = north(TRAPPED_CHEST)facing = north(TRIPWIRE_HOOK)facing$

$north(TUBE_CORAL_WALL_FAN)facing = north(WALL_TORCH)facing = north(WARPED_BUTTON)f$

$north(WARPED_DOOR)facing = north(WARPED_FENCE_GATE)facing = north(WARPED_STAIRS)f$

$north(WARPED_TRAPDOOR)facing = north(WARPED_WALL_SIGN)facing = north(WAXED_CUT_CO$

$north(WAXED_EXPOSED_CUT_COPPER_STAIRS)facing = north(WAXED_OXIDIZED_CUT_COPPER$

$north(WAXED_WEATHERED_CUT_COPPER_STAIRS)facing = north(WEATHERED_CUT_COPPER_ST$

$north(WHITE_BED)facing = north(WHITE_GLAZED_TERRACOTTA)facing = north(WHITE_WALL_B$

$north(WITHER_SKELETON_WALL_SKULL)facing = north(YELLOW_BED)facing =$

$north(YELLOW_GLAZED_TERRACOTTA)facing = north(YELLOW_WALL_BANNER)facing =$

$north(ZOMBIE_WALL_HEAD)facing = south(OBSERVER)facing = up(AMETHYST_CLUSTER)facin$

$up(BLACK_SHULKER_BOX)facing = up(BLUE_SHULKER_BOX)facing = up(BROWN_SHULKER_BO$

$up(CYAN_SHULKER_BOX)facing = up(END_ROD)facing = up(GRAY_SHULKER_BOX)facing =$

$up(GREEN_SHULKER_BOX)facing = up(LARGE_AMETHYST_BUD)facing = up(LIGHTNING_ROD)f$

$up(LIGHT_BLUE_SHULKER_BOX)facing = up(LIGHT_GRAY_SHULKER_BOX)facing =$

$up(LIME_SHULKER_BOX)facing = up(MAGENTA_SHULKER_BOX)facing = up(MEDIUM_AMETHY$

$up(ORANGE_SHULKER_BOX)facing = up(PINK_SHULKER_BOX)facing = up(PURPLE_SHULKER_B$

$up(RED_SHULKER_BOX)facing = up(SHULKER_BOX)facing = up(SMALL_AMETHYST_BUD)facing$

$up(WHITE_SHULKER_BOX)facing = up(YELLOW_SHULKER_BOX)half = bottom(ACACIA_STAIRS)$

$bottom(ACACIA_TRAPDOOR)half = bottom(ANDESITE_STAIRS)half = bottom(BIRCH_STAIRS)hal$

$bottom(BIRCH_TRAPDOOR)half = bottom(BLACKSTONE_STAIRS)half = bottom(BRICK_STAIRS)h$

$bottom(COBBLED_DEEPSLATE_STAIRS)half = bottom(COBBLESTONE_STAIRS)half =$

$bottom(CRIMSON_STAIRS)half = bottom(CRIMSON_TRAPDOOR)half = bottom(CUT_COPPER_STAI$

$bottom(DARK_OAK_STAIRS)half = bottom(DARK_OAK_TRAPDOOR)half = bottom(DARK_PRISMARI$

$bottom(DEEPSLATE_BRICK_STAIRS)half = bottom(DEEPSLATE_TILE_STAIRS)half =$

$bottom(DIORITE_STAIRS)half = bottom(END_STONE_BRICK_STAIRS)half = bottom(EXPOSED_CUT$

$bottom(GRANITE_STAIRS)half = bottom(IRON_TRAPDOOR)half = bottom(JUNGLE_STAIRS)half =$

$bottom(JUNGLE_TRAPDOOR)half = bottom(MANGROVE_STAIRS)half = bottom(MANGROVE_TRA$

$bottom(MOSSY_COBBLESTONE_STAIRS)half = bottom(MOSSY_STONE_BRICK_STAIRS)half =$

$bottom(MUD_BRICK_STAIRS)half = bottom(NETHER_BRICK_STAIRS)half = bottom(OAK_STAIRS)h$

$bottom(OAK_TRAPDOOR)half = bottom(OXIDIZED_CUT_COPPER_STAIRS)half =$

$bottom(POLISHED_ANDESITE_STAIRS)half = bottom(POLISHED_BLACKSTONE_BRICK_STAIRS)$

$bottom(POLISHED_BLACKSTONE_STAIRS)half = bottom(POLISHED_DEEPSLATE_STAIRS)half =$

$bottom(POLISHED_DIORITE_STAIRS)half = bottom(POLISHED_GRANITE_STAIRS)half =$

$bottom(PRISMARINE_BRICK_STAIRS)half = bottom(PRISMARINE_STAIRS)half =$

$bottom(PURPUR_STAIRS)half = bottom(QUARTZ_STAIRS)half = bottom(RED_NETHER_BRICK_STA$

$bottom(RED_SANDSTONE_STAIRS)half = bottom(SANDSTONE_STAIRS)half = bottom(SMOOTH_Q$

$bottom(SMOOTH_RED_SANDSTONE_STAIRS)half = bottom(SMOOTH_SANDSTONE_STAIRS)half =$

$bottom(SPRUCE_STAIRS)half = bottom(SPRUCE_TRAPDOOR)half = bottom(STONE_BRICK_STAIRS$

$bottom(STONE_STAIRS)half = bottom(WARPED_STAIRS)half = bottom(WARPED_TRAPDOOR)half$

$bottom(WAXED_CUT_COPPER_STAIRS)half = bottom(WAXED_EXPOSED_CUT_COPPER_STAIRS)hal$

$bottom(WAXED_OXIDIZED_CUT_COPPER_STAIRS)half = bottom(WAXED_WEATHERED_CUT_COPP$

$bottom(WEATHERED_CUT_COPPER_STAIRS)half = lower(ACACIA_DOOR)half =$

$lower(BIRCH_DOOR)half = lower(CRIMSON_DOOR)half = lower(DARK_OAK_DOOR)half =$

$lower(IRON_DOOR)half = lower(JUNGLE_DOOR)half = lower(LARGE_FERN)half =$

$lower(LILAC)half = lower(MANGROVE_DOOR)half = lower(OAK_DOOR)half =$

$lower(PEONY)half = lower(ROSE_BUSH)half = lower(SMALL_DRIPLEAF)half =$

$lower(SPRUCE_DOOR)half = lower(SUNFLOWER)half = lower(TALL_GRASS)half =$

$lower(TALL_SEAGRASS)half = lower(WARPED_DOOR)hanging = false(LANTERN)hanging =$

$false(MANGROVE_PROPAGULE)hanging = false(SOUL_LANTERN)has_book = false(LECTERN)hat$

$0(TURTLE_EGG)hinge = left(ACACIA_DOOR)hinge = left(BIRCH_DOOR)hinge =$

$left(CRIMSON_DOOR)hinge = left(DARK_OAK_DOOR)hinge = left(IRON_DOOR)hinge =$

$left(JUNGLE_DOOR)hinge = left(MANGROVE_DOOR)hinge = left(OAK_DOOR)hinge =$

$left(SPRUCE_DOOR)hinge = left(WARPED_DOOR)honey_level = 0(BEEHIVE)honey_level =$

$0(BEE_NEST)in_wall = false(ACACIA_FENCE_GATE)in_wall = false(BIRCH_FENCE_GATE)in_wall =$

$false(CRIMSON_FENCE_GATE)in_wall = false(DARK_OAK_FENCE_GATE)in_wall =$

$false(JUNGLE_FENCE_GATE)in_wall = false(MANGROVE_FENCE_GATE)in_wall =$

$false(OAK_FENCE_GATE)in_wall = false(SPRUCE_FENCE_GATE)in_wall = false(WARPED_FENCE_GA$

$false(DAYLIGHT_DETECTOR)layers = 1(SNOW)leaves = none(BAMBOO)level =$

$0(COMPOSTER)level = 0(LAVA)level = 0(WATER)level = 1(POWDER_SNOW_CAULDRON)level =$

$1(WATER_CAULDRON)lit = false(BLACK_CANDLE)lit = false(BLACK_CANDLE_CAKE)lit =$

$false(BLAST_FURNACE)lit = false(BLUE_CANDLE)lit = false(BLUE_CANDLE_CAKE)lit =$

$false(BROWN_CANDLE)lit = false(BROWN_CANDLE_CAKE)lit = false(CANDLE)lit =$

$false(CANDLE_CAKE)lit = false(CYAN_CANDLE)lit = false(CYAN_CANDLE_CAKE)lit =$

$false(DEEPSLATE_REDSTONE_ORE)lit = false(FURNACE)lit = false(GRAY_CANDLE)lit =$

$false(GRAY_CANDLE_CAKE)lit = false(GREEN_CANDLE)lit = false(GREEN_CANDLE_CAKE)lit =$

$false(LIGHT_BLUE_CANDLE)lit = false(LIGHT_BLUE_CANDLE_CAKE)lit = false(LIGHT_GRAY_CAND$

$false(LIGHT_GRAY_CANDLE_CAKE)lit = false(LIME_CANDLE)lit = false(LIME_CANDLE_CAKE)lit$

$false(MAGENTA_CANDLE)lit = false(MAGENTA_CANDLE_CAKE)lit = false(ORANGE_CANDLE)l$

$false(ORANGE_CANDLE_CAKE)lit = false(PINK_CANDLE)lit = false(PINK_CANDLE_CAKE)lit =$

$false(PURPLE_CANDLE)lit = false(PURPLE_CANDLE_CAKE)lit = false(REDSTONE_LAMP)lit =$

$false(REDSTONE_ORE)lit = false(RED_CANDLE)lit = false(RED_CANDLE_CAKE)lit =$

$false(SMOKER)lit = false(WHITE_CANDLE)lit = false(WHITE_CANDLE_CAKE)lit =$

$false(YELLOW_CANDLE)lit = false(YELLOW_CANDLE_CAKE)lit = true(CAMPFIRE)lit =$

$true(REDSTONE_TORCH)lit = true(REDSTONE_WALL_TORCH)lit = true(SOUL_CAMPFIRE)locked$

$false(REPEATER)mode = compare(COMPARATOR)mode = load(STRUCTURE_BLOCK)moisture =$

$0(FARMLAND)note = 0(NOTE_BLOCK)open = false(ACACIA_DOOR)open = false(ACACIA_FENCE$

$false(ACACIA_TRAPDOOR)open = false(BARREL)open = false(BIRCH_DOOR)open =$

$false(BIRCH_FENCE_GATE)open = false(BIRCH_TRAPDOOR)open = false(CRIMSON_DOOR)open =$

$false(CRIMSON_FENCE_GATE)open = false(CRIMSON_TRAPDOOR)open = false(DARK_OAK_DOO$

$false(DARK_OAK_FENCE_GATE)open = false(DARK_OAK_TRAPDOOR)open = false(IRON_DOOR)ope$

$false(IRON_TRAPDOOR)open = false(JUNGLE_DOOR)open = false(JUNGLE_FENCE_GATE)open =$

$false(JUNGLE_TRAPDOOR)open = false(MANGROVE_DOOR)open = false(MANGROVE_FENCE_GA$

$false(MANGROVE_TRAPDOOR)open = false(OAK_DOOR)open = false(OAK_FENCE_GATE)open =$

$false(OAK_TRAPDOOR)open = false(SPRUCE_DOOR)open = false(SPRUCE_FENCE_GATE)open =$

$false(SPRUCE_TRAPDOOR)open = false(WARPED_DOOR)open = false(WARPED_FENCE_GATE)op$

$false(WARPED_TRAPDOOR)orientation = north_up(JIGSAW)part = foot(BLACK_BED)part =$

$foot(BLUE_BED)part = foot(BROWN_BED)part = foot(CYAN_BED)part = foot(GRAY_BED)part =$

$foot(GREEN_BED)part = foot(LIGHT_BLUE_BED)part = foot(LIGHT_GRAY_BED)part =$

$foot(LIME_BED)part = foot(MAGENTA_BED)part = foot(ORANGE_BED)part =$

$foot(PINK_BED)part = foot(PURPLE_BED)part = foot(RED_BED)part = foot(WHITE_BED)part =$

$foot(YELLOW_BED)pickles = 1(SEA_PICKLE)rotation = 0(ACACIA_SIGN)rotation =$

$0(BIRCH_SIGN)rotation = 0(BLACK_BANNER)rotation = 0(BLUE_BANNER)rotation =$

$0(BROWN_BANNER)rotation = 0(CREEPER_HEAD)rotation = 0(CRIMSON_SIGN)rotation =$

$0(CYAN_BANNER)rotation = 0(DARK_OAK_SIGN)rotation = 0(DRAGON_HEAD)rotation =$

$0(GRAY_BANNER)rotation = 0(GREEN_BANNER)rotation = 0(JUNGLE_SIGN)rotation =$

$0(LIGHT_BLUE_BANNER)rotation = 0(LIGHT_GRAY_BANNER)rotation = 0(LIME_BANNER)rotation$

$0(MAGENTA_BANNER)rotation = 0(MANGROVE_SIGN)rotation = 0(OAK_SIGN)rotation =$

$0(ORANGE_BANNER)rotation = 0(PINK_BANNER)rotation = 0(PLAYER_HEAD)rotation =$

$0(PURPLE_BANNER)rotation = 0(RED_BANNER)rotation = 0(SKELETON_SKULL)rotation =$

$0(SPRUCE_SIGN)rotation = 0(WARPED_SIGN)rotation = 0(WHITE_BANNER)rotation =$

$0(WITHER_SKELETON_SKULL)rotation = 0(YELLOW_BANNER)rotation = 0(ZOMBIE_HEAD)sculk$

$inactive(SCULK_SENSOR)shape = north_south(ACTIVATOR_RAIL)shape = north_south(DETECTOR_RA$

$north_south(POWERED_RAIL)shape = north_south(RAIL)shape = straight(ACACIA_STAIRS)shape =$

$straight(ANDESITE_STAIRS)shape = straight(BIRCH_STAIRS)shape = straight(BLACKSTONE_STA$

$straight(BRICK_STAIRS)shape = straight(COBBLED_DEEPSLATE_STAIRS)shape =$

$straight(COBBLESTONE_STAIRS)shape = straight(CRIMSON_STAIRS)shape = straight(CUT_COPP$

$straight(DARK_OAK_STAIRS)shape = straight(DARK_PRISMARINE_STAIRS)shape =$

$straight(DEEPSLATE_BRICK_STAIRS)shape = straight(DEEPSLATE_TILE_STAIRS)shape =$

$straight(DIORITE_STAIRS)shape = straight(END_STONE_BRICK_STAIRS)shape =$

$straight(EXPOSED_CUT_COPPER_STAIRS)shape = straight(GRANITE_STAIRS)shape =$

$straight(JUNGLE_STAIRS)shape = straight(MANGROVE_STAIRS)shape = straight(MOSSY_COBBLE$

$straight(MOSSY_STONE_BRICK_STAIRS)shape = straight(MUD_BRICK_STAIRS)shape =$

$straight(NETHER_BRICK_STAIRS)shape = straight(OAK_STAIRS)shape = straight(OXIDIZED_CUT_C$

$straight(POLISHED_ANDESITE_STAIRS)shape = straight(POLISHED_BLACKSTONE_BRICK_STAI$

$straight(POLISHED_BLACKSTONE_STAIRS)shape = straight(POLISHED_DEEPSLATE_STAIRS)sh$

$straight(POLISHED_DIORITE_STAIRS)shape = straight(POLISHED_GRANITE_STAIRS)shape =$

$straight(PRISMARINE_BRICK_STAIRS)shape = straight(PRISMARINE_STAIRS)shape =$

$straight(PURPUR_STAIRS)shape = straight(QUARTZ_STAIRS)shape = straight(RED_NETHER_BRIC$

$straight(RED_SANDSTONE_STAIRS)shape = straight(SANDSTONE_STAIRS)shape =$

$straight(SMOOTH_QUARTZ_STAIRS)shape = straight(SMOOTH_RED_SANDSTONE_STAIRS)shape =$

$straight(SMOOTH_SANDSTONE_STAIRS)shape = straight(SPRUCE_STAIRS)shape =$

$straight(STONE_BRICK_STAIRS)shape = straight(STONE_STAIRS)shape = straight(WARPED_STAI$

$straight(WAXED_CUT_COPPER_STAIRS)shape = straight(WAXED_EXPOSED_CUT_COPPER_STAIRS)$
$straight(WAXED_OXIDIZED_CUT_COPPER_STAIRS)shape = straight(WAXED_WEATHERED_CUT_C$
$straight(WEATHERED_CUT_COPPER_STAIRS)signal_fire = false(CAMPFIRE)signal_fire =$
$false(SOUL_CAMPFIRE)snowy = false(GRASS_BLOCK)snowy = false(MYCELIUM)snowy =$
$false(PODZOL)thickness = tip(POINTED_DRIPSTONE)type = bottom(ACACIA_SLAB)type =$
$bottom(ANDESITE_SLAB)type = bottom(BIRCH_SLAB)type = bottom(BLACKSTONE_SLAB)type =$
$bottom(BRICK_SLAB)type = bottom(COBBLED_DEEPSLATE_SLAB)type = bottom(COBBLESTONE_S$
$bottom(CRIMSON_SLAB)type = bottom(CUT_COPPER_SLAB)type = bottom(CUT_RED_SANDSTONE_SL$
$bottom(CUT_SANDSTONE_SLAB)type = bottom(DARK_OAK_SLAB)type = bottom(DARK_PRISMARINE$
$bottom(DEEPSLATE_BRICK_SLAB)type = bottom(DEEPSLATE_TILE_SLAB)type =$
$bottom(DIORITE_SLAB)type = bottom(END_STONE_BRICK_SLAB)type = bottom(EXPOSED_CUT_COP$
$bottom(GRANITE_SLAB)type = bottom(JUNGLE_SLAB)type = bottom(MANGROVE_SLAB)type =$
$bottom(MOSSY_COBBLESTONE_SLAB)type = bottom(MOSSY_STONE_BRICK_SLAB)type =$
$bottom(MUD_BRICK_SLAB)type = bottom(NETHER_BRICK_SLAB)type = bottom(OAK_SLAB)type =$
$bottom(OXIDIZED_CUT_COPPER_SLAB)type = bottom(PETRIFIED_OAK_SLAB)type =$
$bottom(POLISHED_ANDESITE_SLAB)type = bottom(POLISHED_BLACKSTONE_BRICK_SLAB)type =$
$bottom(POLISHED_BLACKSTONE_SLAB)type = bottom(POLISHED_DEEPSLATE_SLAB)type =$
$bottom(POLISHED_DIORITE_SLAB)type = bottom(POLISHED_GRANITE_SLAB)type =$
$bottom(PRISMARINE_BRICK_SLAB)type = bottom(PRISMARINE_SLAB)type = bottom(PURPUR_SL$
$bottom(QUARTZ_SLAB)type = bottom(RED_NETHER_BRICK_SLAB)type = bottom(RED_SANDSTONE_S$
$bottom(SANDSTONE_SLAB)type = bottom(SMOOTH_QUARTZ_SLAB)type = bottom(SMOOTH_RED_SA$
$bottom(SMOOTH_SANDSTONE_SLAB)type = bottom(SMOOTH_STONE_SLAB)type =$
$bottom(SPRUCE_SLAB)type = bottom(STONE_BRICK_SLAB)type = bottom(STONE_SLAB)type =$
$bottom(WARPED_SLAB)type = bottom(WAXED_CUT_COPPER_SLAB)type = bottom(WAXED_EXPOSE$
$bottom(WAXED_OXIDIZED_CUT_COPPER_SLAB)type = bottom(WAXED_WEATHERED_CUT_COPPE$
$bottom(WEATHERED_CUT_COPPER_SLAB)type = normal(MOVING_PISTON)type =$
$normal(PISTON_HEAD)type = single(CHEST)type = single(TRAPPED_CHEST)vertical_direction =$
$up(POINTED_DRIPSTONE)waterlogged = false(ACACIA_FENCE)$

A.1.15 Waterlogged

waterlogged=false (ACACIA$_{L}EAVES)waterlogged = false(ACACIA_{S}IGN)waterlogged =$
$false(ACACIA_{S}LAB)waterlogged = false(ACACIA_{S}TAIRS)waterlogged = false(ACACIA_{T}RAPDOO$
$false(ACACIA_{W}ALL_{S}IGN)waterlogged = false(ACTIVATOR_{R}AIL)waterlogged =$
$false(AMETHYST_{C}LUSTER)waterlogged = false(ANDESITE_{S}LAB)waterlogged =$
$false(ANDESITE_{S}TAIRS)waterlogged = false(ANDESITE_{W}ALL)waterlogged = false(AZALEA_{L}EA$
$false(BIG_{D}RIPLEAF)waterlogged = false(BIG_{D}RIPLEAF_{S}TEM)waterlogged = false(BIRCH_{F}ENC$
$false(BIRCH_{L}EAVES)waterlogged = false(BIRCH_{S}IGN)waterlogged = false(BIRCH_{S}LAB)waterlog$
$false(BIRCH_{S}TAIRS)waterlogged = false(BIRCH_{T}RAPDOOR)waterlogged = false(BIRCH_{W}ALL_{S}I$
$false(BLACKSTONE_{S}LAB)waterlogged = false(BLACKSTONE_{S}TAIRS)waterlogged =$
$false(BLACKSTONE_{W}ALL)waterlogged = false(BLACK_{C}ANDLE)waterlogged =$
$false(BLACK_{S}TAINED_{G}LASS_{P}ANE)waterlogged = false(BLUE_{C}ANDLE)waterlogged =$
$false(BLUE_{S}TAINED_{G}LASS_{P}ANE)waterlogged = false(BRICK_{S}LAB)waterlogged =$
$false(BRICK_{S}TAIRS)waterlogged = false(BRICK_{W}ALL)waterlogged = false(BROWN_{C}ANDLE)wat$
$false(BROWN_{S}TAINED_{G}LASS_{P}ANE)waterlogged = false(CAMPFIRE)waterlogged =$
$false(CANDLE)waterlogged = false(CHAIN)waterlogged = false(CHEST)waterlogged =$
$false(COBBLED_{D}EEPSLATE_{S}LAB)waterlogged = false(COBBLED_{D}EEPSLATE_{S}TAIRS)waterlog$
$false(COBBLED_{D}EEPSLATE_{W}ALL)waterlogged = false(COBBLESTONE_{S}LAB)waterlogged =$
$false(COBBLESTONE_{S}TAIRS)waterlogged = false(COBBLESTONE_{W}ALL)waterlogged =$
$false(CRIMSON_{F}ENCE)waterlogged = false(CRIMSON_{S}IGN)waterlogged = false(CRIMSON_{S}LA$
$false(CRIMSON_{S}TAIRS)waterlogged = false(CRIMSON_{T}RAPDOOR)waterlogged =$
$false(CRIMSON_{W}ALL_{S}IGN)waterlogged = false(CUT_{C}OPPER_{S}LAB)waterlogged =$
$false(CUT_{C}OPPER_{S}TAIRS)waterlogged = false(CUT_{R}ED_{S}ANDSTONE_{S}LAB)waterlogged =$
$false(CUT_{S}ANDSTONE_{S}LAB)waterlogged = false(CYAN_{C}ANDLE)waterlogged =$
$false(CYAN_{S}TAINED_{G}LASS_{P}ANE)waterlogged = false(DARK_{O}AK_{F}ENCE)waterlogged =$
$false(DARK_{O}AK_{L}EAVES)waterlogged = false(DARK_{O}AK_{S}IGN)waterlogged = false(DARK_{O}AK_{S}LA$
$false(DARK_{O}AK_{S}TAIRS)waterlogged = false(DARK_{O}AK_{T}RAPDOOR)waterlogged =$
$false(DARK_{O}AK_{W}ALL_{S}IGN)waterlogged = false(DARK_{P}RISMARINE_{S}LAB)waterlogged =$
$false(DARK_{P}RISMARINE_{S}TAIRS)waterlogged = false(DEEPSLATE_{B}RICK_{S}LAB)waterlogged =$
$false(DEEPSLATE_{B}RICK_{S}TAIRS)waterlogged = false(DEEPSLATE_{B}RICK_{W}ALL)waterlogged =$
$false(DEEPSLATE_{T}ILE_{S}LAB)waterlogged = false(DEEPSLATE_{T}ILE_{S}TAIRS)waterlogged =$

$false(DEEPSLATE_TILE_WALL)waterlogged = false(DETECTOR_RAIL)waterlogged =$

$false(DIORITE_SLAB)waterlogged = false(DIORITE_STAIRS)waterlogged = false(DIORITE_WALL)w$

$false(ENDER_CHEST)waterlogged = false(END_STONE_BRICK_SLAB)waterlogged =$

$false(END_STONE_BRICK_STAIRS)waterlogged = false(END_STONE_BRICK_WALL)waterlogged =$

$false(EXPOSED_CUT_COPPER_SLAB)waterlogged = false(EXPOSED_CUT_COPPER_STAIRS)waterlog$

$false(FLOWERING_AZALEA_LEAVES)waterlogged = false(GLASS_PANE)waterlogged =$

$false(GLOW_LICHEN)waterlogged = false(GRANITE_SLAB)waterlogged = false(GRANITE_STAIRS)$

$false(GRANITE_WALL)waterlogged = false(GRAY_CANDLE)waterlogged = false(GRAY_STAINED_GL$

$false(GREEN_CANDLE)waterlogged = false(GREEN_STAINED_GLASS_PANE)waterlogged =$

$false(HANGING_ROOTS)waterlogged = false(IRON_BARS)waterlogged = false(IRON_TRAPDOOR)wa$

$false(JUNGLE_FENCE)waterlogged = false(JUNGLE_LEAVES)waterlogged = false(JUNGLE_SIGN)$

$false(JUNGLE_SLAB)waterlogged = false(JUNGLE_STAIRS)waterlogged = false(JUNGLE_TRAPDO$

$false(JUNGLE_WALL_SIGN)waterlogged = false(LADDER)waterlogged = false(LANTERN)waterlogg$

$false(LARGE_AMETHYST_BUD)waterlogged = false(LIGHTNING_ROD)waterlogged =$

$false(LIGHT_BLUE_CANDLE)waterlogged = false(LIGHT_BLUE_STAINED_GLASS_PANE)waterlogged =$

$false(LIGHT_GRAY_CANDLE)waterlogged = false(LIGHT_GRAY_STAINED_GLASS_PANE)waterlogged =$

$false(LIME_CANDLE)waterlogged = false(LIME_STAINED_GLASS_PANE)waterlogged =$

$false(MAGENTA_CANDLE)waterlogged = false(MAGENTA_STAINED_GLASS_PANE)waterlogged =$

$false(MANGROVE_FENCE)waterlogged = false(MANGROVE_LEAVES)waterlogged =$

$false(MANGROVE_PROPAGULE)waterlogged = false(MANGROVE_ROOTS)waterlogged =$

$false(MANGROVE_SIGN)waterlogged = false(MANGROVE_SLAB)waterlogged = false(MANGROVE$

$false(MANGROVE_TRAPDOOR)waterlogged = false(MANGROVE_WALL_SIGN)waterlogged =$

$false(MEDIUM_AMETHYST_BUD)waterlogged = false(MOSSY_COBBLESTONE_SLAB)waterlogged =$

$false(MOSSY_COBBLESTONE_STAIRS)waterlogged = false(MOSSY_COBBLESTONE_WALL)waterlo$

$false(MOSSY_STONE_BRICK_SLAB)waterlogged = false(MOSSY_STONE_BRICK_STAIRS)waterlogged =$

$false(MOSSY_STONE_BRICK_WALL)waterlogged = false(MUD_BRICK_SLAB)waterlogged =$

$false(MUD_BRICK_STAIRS)waterlogged = false(MUD_BRICK_WALL)waterlogged =$

$false(NETHER_BRICK_FENCE)waterlogged = false(NETHER_BRICK_SLAB)waterlogged =$

$false(NETHER_BRICK_STAIRS)waterlogged = false(NETHER_BRICK_WALL)waterlogged =$

$false(OAK_FENCE)waterlogged = false(OAK_LEAVES)waterlogged = false(OAK_SIGN)waterlogged =$

$false(OAK_SLAB)waterlogged = false(OAK_STAIRS)waterlogged = false(OAK_TRAPDOOR)waterlogge$

$false(OAK_{W}ALL_{S}IGN)waterlogged = false(ORANGE_{C}ANDLE)waterlogged = false(ORANGE_{S}TAIN$

$false(OXIDIZED_{C}UT_{C}OPPER_{S}LAB)waterlogged = false(OXIDIZED_{C}UT_{C}OPPER_{S}TAIRS)waterl$

$false(PETRIFIED_{O}AK_{S}LAB)waterlogged = false(PINK_{C}ANDLE)waterlogged =$

$false(PINK_{S}TAINED_{G}LASS_{P}ANE)waterlogged = false(POINTED_{D}RIPSTONE)waterlogged =$

$false(POLISHED_{A}NDESITE_{S}LAB)waterlogged = false(POLISHED_{A}NDESITE_{S}TAIRS)waterlogg$

$false(POLISHED_{B}LACKSTONE_{B}RICK_{S}LAB)waterlogged = false(POLISHED_{B}LACKSTONE_{B}R$

$false(POLISHED_{B}LACKSTONE_{B}RICK_{W}ALL)waterlogged = false(POLISHED_{B}LACKSTONE_{S}L$

$false(POLISHED_{B}LACKSTONE_{S}TAIRS)waterlogged = false(POLISHED_{B}LACKSTONE_{W}ALL)w$

$false(POLISHED_{D}EEPSLATE_{S}LAB)waterlogged = false(POLISHED_{D}EEPSLATE_{S}TAIRS)waterl$

$false(POLISHED_{D}EEPSLATE_{W}ALL)waterlogged = false(POLISHED_{D}IORITE_{S}LAB)waterlogged$

$false(POLISHED_{D}IORITE_{S}TAIRS)waterlogged = false(POLISHED_{G}RANITE_{S}LAB)waterlogged =$

$false(POLISHED_{G}RANITE_{S}TAIRS)waterlogged = false(POWERED_{R}AIL)waterlogged =$

$false(PRISMARINE_{B}RICK_{S}LAB)waterlogged = false(PRISMARINE_{B}RICK_{S}TAIRS)waterlogged =$

$false(PRISMARINE_{S}LAB)waterlogged = false(PRISMARINE_{S}TAIRS)waterlogged =$

$false(PRISMARINE_{W}ALL)waterlogged = false(PURPLE_{C}ANDLE)waterlogged =$

$false(PURPLE_{S}TAINED_{G}LASS_{P}ANE)waterlogged = false(PURPUR_{S}LAB)waterlogged =$

$false(PURPUR_{S}TAIRS)waterlogged = false(QUARTZ_{S}LAB)waterlogged = false(QUARTZ_{S}TAIRS)w$

$false(RAIL)waterlogged = false(RED_{C}ANDLE)waterlogged = false(RED_{N}ETHER_{B}RICK_{S}LAB)wat$

$false(RED_{N}ETHER_{B}RICK_{S}TAIRS)waterlogged = false(RED_{N}ETHER_{B}RICK_{W}ALL)waterlogged =$

$false(RED_{S}ANDSTONE_{S}LAB)waterlogged = false(RED_{S}ANDSTONE_{S}TAIRS)waterlogged =$

$false(RED_{S}ANDSTONE_{W}ALL)waterlogged = false(RED_{S}TAINED_{G}LASS_{P}ANE)waterlogged =$

$false(SANDSTONE_{S}LAB)waterlogged = false(SANDSTONE_{S}TAIRS)waterlogged =$

$false(SANDSTONE_{W}ALL)waterlogged = false(SCAFFOLDING)waterlogged = false(SCULK_{S}ENSC$

$false(SCULK_{S}HRIEKER)waterlogged = false(SCULK_{V}EIN)waterlogged = false(SMALL_{A}METHY$

$false(SMALL_{D}RIPLEAF)waterlogged = false(SMOOTH_{Q}UARTZ_{S}LAB)waterlogged =$

$false(SMOOTH_{Q}UARTZ_{S}TAIRS)waterlogged = false(SMOOTH_{R}ED_{S}ANDSTONE_{S}LAB)waterlogge$

$false(SMOOTH_{R}ED_{S}ANDSTONE_{S}TAIRS)waterlogged = false(SMOOTH_{S}ANDSTONE_{S}LAB)wate$

$false(SMOOTH_{S}ANDSTONE_{S}TAIRS)waterlogged = false(SMOOTH_{S}TONE_{S}LAB)waterlogged =$

$false(SOUL_{C}AMPFIRE)waterlogged = false(SOUL_{L}ANTERN)waterlogged = false(SPRUCE_{F}ENC$

$false(SPRUCE_{L}EAVES)waterlogged = false(SPRUCE_{S}IGN)waterlogged = false(SPRUCE_{S}LAB)wat$

$false(SPRUCE_{S}TAIRS)waterlogged = false(SPRUCE_{T}RAPDOOR)waterlogged =$

$false(SPRUCE_{W}ALL_{S}IGN)waterlogged = false(STONE_{B}RICK_{S}LAB)waterlogged =$
$false(STONE_{B}RICK_{S}TAIRS)waterlogged = false(STONE_{B}RICK_{W}ALL)waterlogged =$
$false(STONE_{S}LAB)waterlogged = false(STONE_{S}TAIRS)waterlogged = false(TRAPPED_{C}HEST)wa$
$false(WARPED_{F}ENCE)waterlogged = false(WARPED_{S}IGN)waterlogged = false(WARPED_{S}LAB)w$
$false(WARPED_{S}TAIRS)waterlogged = false(WARPED_{T}RAPDOOR)waterlogged =$
$false(WARPED_{W}ALL_{S}IGN)waterlogged = false(WAXED_{C}UT_{C}OPPER_{S}LAB)waterlogged =$
$false(WAXED_{C}UT_{C}OPPER_{S}TAIRS)waterlogged = false(WAXED_{E}XPOSED_{C}UT_{C}OPPER_{S}LAB)w$
$false(WAXED_{E}XPOSED_{C}UT_{C}OPPER_{S}TAIRS)waterlogged = false(WAXED_{O}XIDIZED_{C}UT_{C}OP$
$false(WAXED_{O}XIDIZED_{C}UT_{C}OPPER_{S}TAIRS)waterlogged = false(WAXED_{W}EATHERED_{C}UT_{C}$
$false(WAXED_{W}EATHERED_{C}UT_{C}OPPER_{S}TAIRS)waterlogged = false(WEATHERED_{C}UT_{C}OPPE$
$false(WEATHERED_{C}UT_{C}OPPER_{S}TAIRS)waterlogged = false(WHITE_{C}ANDLE)waterlogged =$
$false(WHITE_{S}TAINED_{G}LASS_{P}ANE)waterlogged = false(YELLOW_{C}ANDLE)waterlogged =$
$false(YELLOW_{S}TAINED_{G}LASS_{P}ANE)$

waterlogged=true (BRAIN$_{C}ORAL)waterlogged = true(BRAIN_{C}ORAL_{F}AN)waterlogged =$
$true(BRAIN_{C}ORAL_{W}ALL_{F}AN)waterlogged = true(BUBBLE_{C}ORAL)waterlogged =$
$true(BUBBLE_{C}ORAL_{F}AN)waterlogged = true(BUBBLE_{C}ORAL_{W}ALL_{F}AN)waterlogged =$
$true(CONDUIT)waterlogged = true(DEAD_{B}RAIN_{C}ORAL)waterlogged = true(DEAD_{B}RAIN_{C}ORAL_{F}$
$true(DEAD_{B}RAIN_{C}ORAL_{W}ALL_{F}AN)waterlogged = true(DEAD_{B}UBBLE_{C}ORAL)waterlogged =$
$true(DEAD_{B}UBBLE_{C}ORAL_{F}AN)waterlogged = true(DEAD_{B}UBBLE_{C}ORAL_{W}ALL_{F}AN)waterlogged$
$true(DEAD_{F}IRE_{C}ORAL)waterlogged = true(DEAD_{F}IRE_{C}ORAL_{F}AN)waterlogged =$
$true(DEAD_{F}IRE_{C}ORAL_{W}ALL_{F}AN)waterlogged = true(DEAD_{H}ORN_{C}ORAL)waterlogged =$
$true(DEAD_{H}ORN_{C}ORAL_{F}AN)waterlogged = true(DEAD_{H}ORN_{C}ORAL_{W}ALL_{F}AN)waterlogged =$
$true(DEAD_{T}UBE_{C}ORAL)waterlogged = true(DEAD_{T}UBE_{C}ORAL_{F}AN)waterlogged =$
$true(DEAD_{T}UBE_{C}ORAL_{W}ALL_{F}AN)waterlogged = true(FIRE_{C}ORAL)waterlogged =$
$true(FIRE_{C}ORAL_{F}AN)waterlogged = true(FIRE_{C}ORAL_{W}ALL_{F}AN)waterlogged =$
$true(HORN_{C}ORAL)waterlogged = true(HORN_{C}ORAL_{F}AN)waterlogged = true(HORN_{C}ORAL_{W}ALL_{F}$
$true(SEA_{P}ICKLE)waterlogged = true(TUBE_{C}ORAL)waterlogged = true(TUBE_{C}ORAL_{F}AN)waterlogg$
$true(TUBE_{C}ORAL_{W}ALL_{F}AN)$

A.2 Material modifiers concatenation

... (how to join modifiers)

Figure A.1: Modifier concatenation

Modifier name	Reason for discarding
has_bottle_X	Inventory dependent
has_record	Inventory dependent
enabled	Adjacent redstone dependent
triggered	Adjacent redstone dependent
instrument	Bottom-block dependent
occupied	Entity dependent
persistent	Admin block
unstable	Admin block
distance	Block dependent
stage	Same block
short	Tick dependent
attached	Block dependent
disarmed	Block dependent
power	Block/event dependent
tilt	Entity dependent
can_summon	Admin block
shrieking	Entity dependent
bloom	Admin block
bottom	Bottom-block dependent
powered	Admin block / block dependent

Table A.1: Unused Spigot BlockData's modifiers

Block name	Modifier name
CAVE_VINES	age
CACTUS	age
FIRE	age
KELP	age
SUGAR_CANE	age
MANGROVE_PROPAGULE	age
TWISTING_VINES	age
WEEPING_VINES	age

Table A.2: Unused Spigot BlockData's modifiers on certain blocks

References

Papermc. (n.d.). Retrieved from https://papermc.io/

Spigot. (n.d.). Retrieved from https://www.spigotmc.org/

Spigot - enum material. (n.d.). Retrieved 2022-08-04, from https://hub.spigotmc.org/javadocs/bukkit/org/bukkit/Material.html

Spigot - interface blockdata. (n.d.). Retrieved 2022-08-04, from https://hub.spigotmc.org/javadocs/bukkit/org/bukkit/block/data/BlockData.html

Worldguard. (n.d.). Retrieved from https://dev.bukkit.org/projects/worldguard

World - minecraft wiki. (n.d.). Retrieved from https://minecraft.fandom.com/wiki/World