WatchWolf API Definition

WatchWolf Contributors

Keywords: WatchWolf; Minecraft plugin testing; Integration testing environment.

Contents

Figures

9

Tables

1 Documentation conventions

1.1 Acronyms

MC Minecraft. 15, 107, 109, 120

1.2 Glossary

ASCII American Standard Code for Information Interchange. Is a 8-bit code where each individual bit represents a unique character.*ASCII table* (n.d.). 18

async Asynchronous; returned non-sequentially. 18, 107, 135

IEEE 754 IEEE Standard for Floating-Point Arithmetic (IEEE 754) is a standard for computing floating-points operations. For more details about it check Rajaraman (2016). 19

Least significant bit The Least significant bit is the bit with the lowest index number. 8, 17, 19

Most significant bit The Most significant bit is the bit with the higher index number. 17, 109

Multidimensional array Array of arrays. 20

NOP No OPeration. It represents a valid operation that means 'do nothing'. 18

2   WatchWolf Introduction

WatchWolf is an integration testing environment for Minecraft plugins. It will validate that your plugin works using multiple real MC servers of different types and versions.

In order to achieve that, WatchWolf splits into 4 different programs, each one with one responsibility:

1. WatchWolf Tester

   WatchWolf Tester if the entry point to the WatchWolf environment.[1] It will orchestrate all the setup/stop process and run the user tests.

2. WatchWolf Servers Manager

   WatchWolf Servers Manager provides MC servers on-demand. It will start them and, after they have been closed, free the allocated resources.

3. WatchWolf Server

   WatchWolf Server is the actual MC server. It will contain the plugin to test and run the commands sent by WatchWolf Tester.

4. WatchWolf Clients Manager

   WatchWolf Clients Manager is the same as WatchWolf Servers Manager, but for clients. It will starts clients on-demand and connect them to the servers allocated by WatchWolf.

5. WatchWolf Client

   WatchWolf Client is a MC client, with the ability to connect to one server and interact with it.

You can see with more detail how the different programs relations on the Figure 2.1, Diagram representing WatchWolf's most important actuators.

---

[1]The WatchWolf environment is the combination of all the WatchWolf parts: Tester, Servers Manager, Server, Clients Manager and Client.
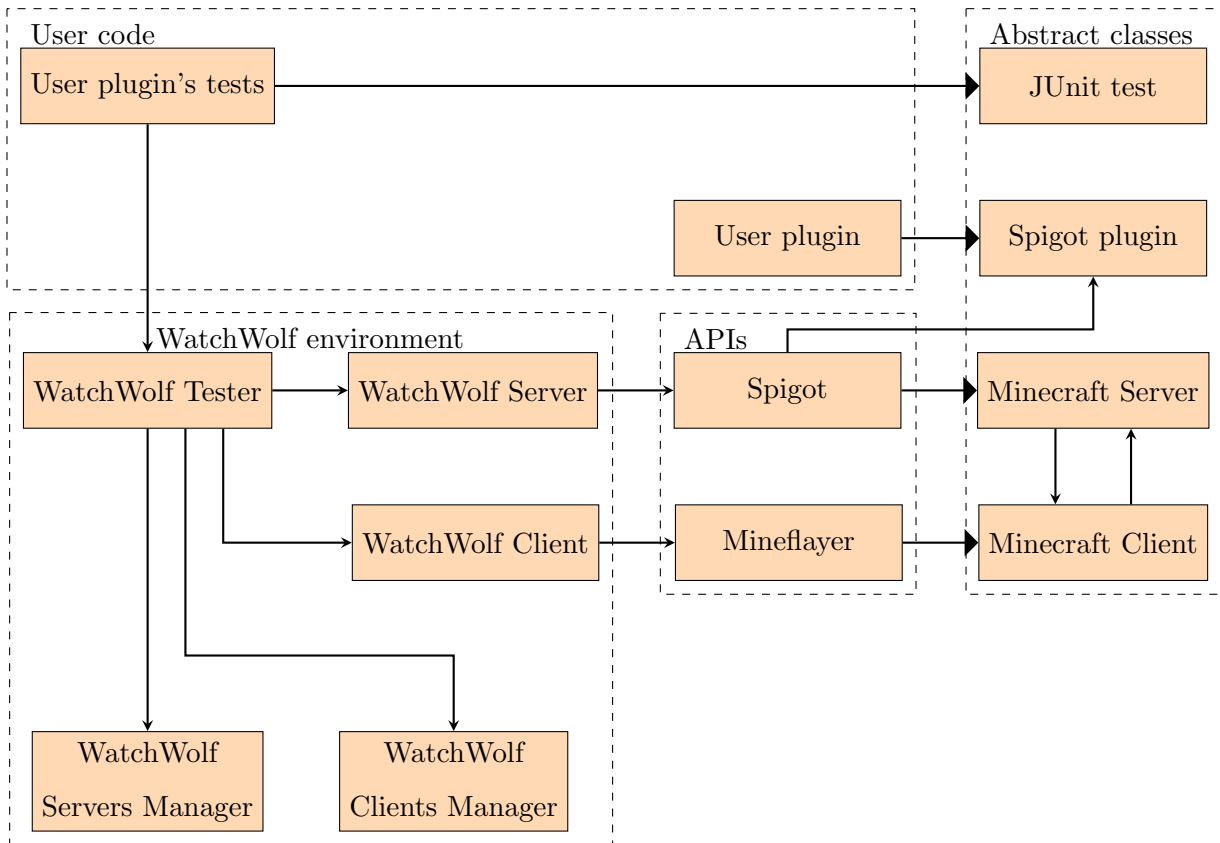
Figure 2.1: Diagram representing WatchWolf's most important actuators

3 API Introduction

In order to interact with the different WatchWolf modules, you'll have to follow the Watch-
Wolf API: a series of supported operations in one program. All the packets sent & received
will follow the structure shown in Figure 3.1, Packet structure.
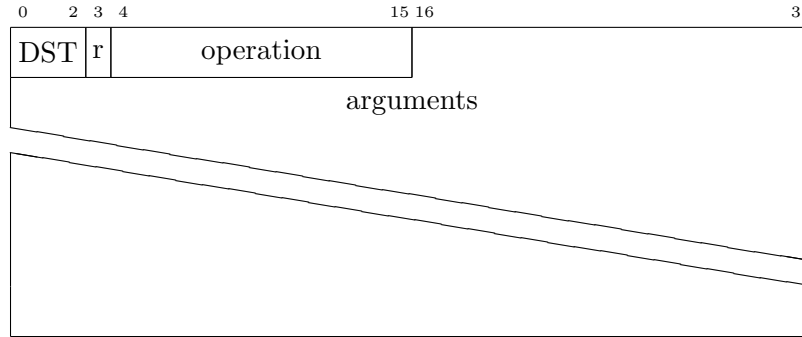
Figure 3.1: Packet structure

As a general rule, both MSB and LSB are preserved. This means that in a 2-bytes
packet, the first part (0..7) will be the MSB, and the last (8..15) MSB.

3.1 Destiny

The first argument (DST) will be the destiny of that packet. This will specify one of
the 4 modules connected to WatchWolf Tester (for more information refer to Section 2,
WatchWolf Introduction). Note that WatchWolf Tester itself is not present, as it will be
indicated with the Response bit at 1. You can see the different DST values for each module
om the Figure 3.1, DST bits meaning.

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

Table 3.1: DST bits meaning

3.2 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" (e.g. Error notification). Those will be sent directly marked as responses (Response bit at '1').

3.3 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 kicked by inactivity[2] if you send this request every few minutes.

3.4 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.

3.4.1 Character

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

3.4.2 Boolean

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

---

[2]This is a safety mechanism to avoid blocking a server to the same user forever. Besides being defined by the API it hasn't been implemented yet, and won't be until WatchWolf offers public servers.

For alignment reasons,[3] booleans will be sent as 1-byte element. To avoid misunderstandings, let's define *false* as 0x00, and *true* as '*not false*'. That way, both figures 3.2 and 3.3 are valid *true* elements.

0   7

0x01
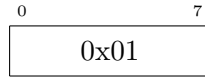
Figure 3.2: True packet with the LSB at 1

0   7

0xFF
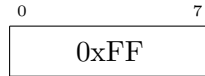
Figure 3.3: True packet with all bits at 1

3.4.3   Double

Doubles are 8-bytes floating-point numbers. They are represented following the IEEE 754[4].

3.4.4   String

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

3.4.5   Array

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

The structure is a 2-byte 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.

---

[3]In order to make the read/write more easy, we want to stick with (at least) 8 bits blocks.

[4]This standard is the one used by C and Java. Cite needed here

0	15 16	23 24	31
size		char[0]	char[1]
...			
char[n-4]	char[n-3]	char[n-2]	char[n-1]
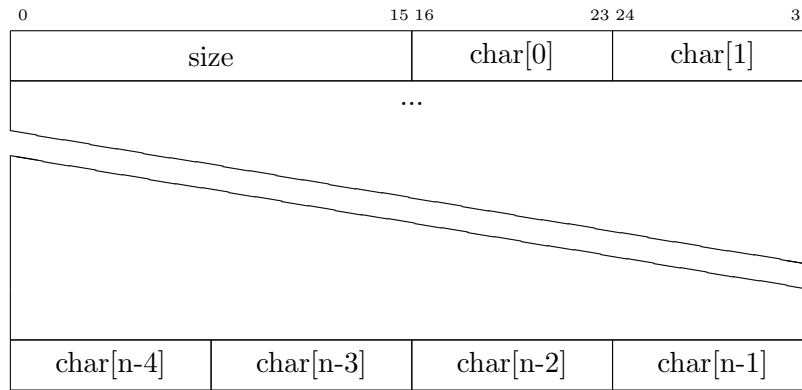
Figure 3.4: Structure of a String

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 3.5, 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 3.5: Example of a string array

3.4.6   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[5] is 4GB.

---

[5]Besides defining here what's allowed, remember that this packet will be inside a TCP payload. This means that the maximum file size will be probably redefined by the machine's TCP firewalls.
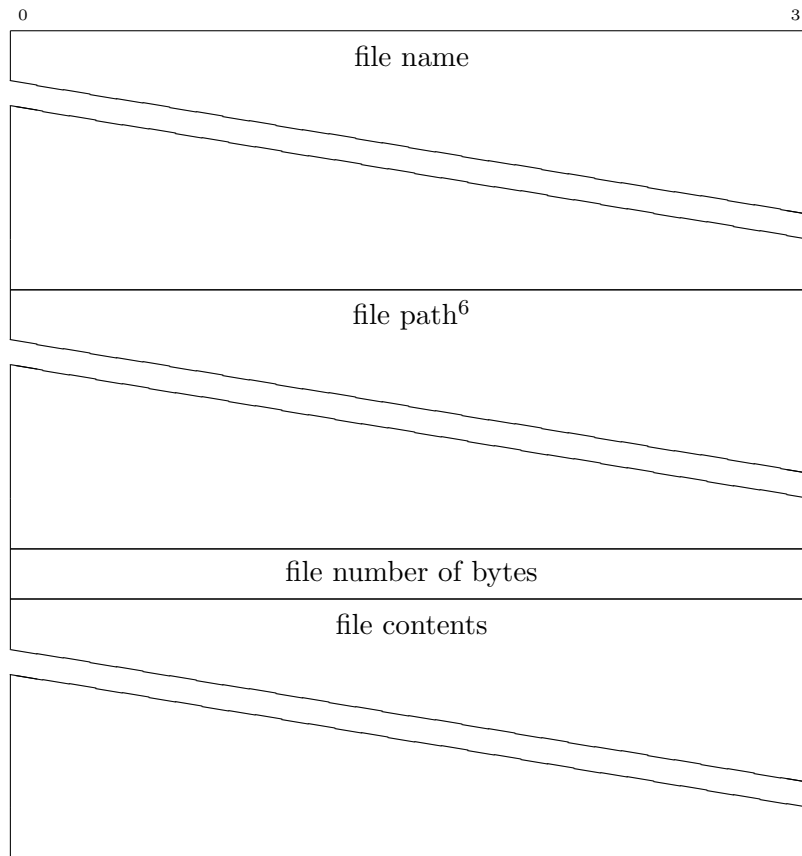
Figure 3.6: File structure

3.4.7 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 scalability reasons this parameter is a String specifying the server type.

3.4.8 Position

One position represents a point in space (world & x-y-z). It can be used to find entities or blocks.

---

[6]The path must be relative, and you can't go outside the Server directory (using '../'). Both " and './' means the root of the Server directory.
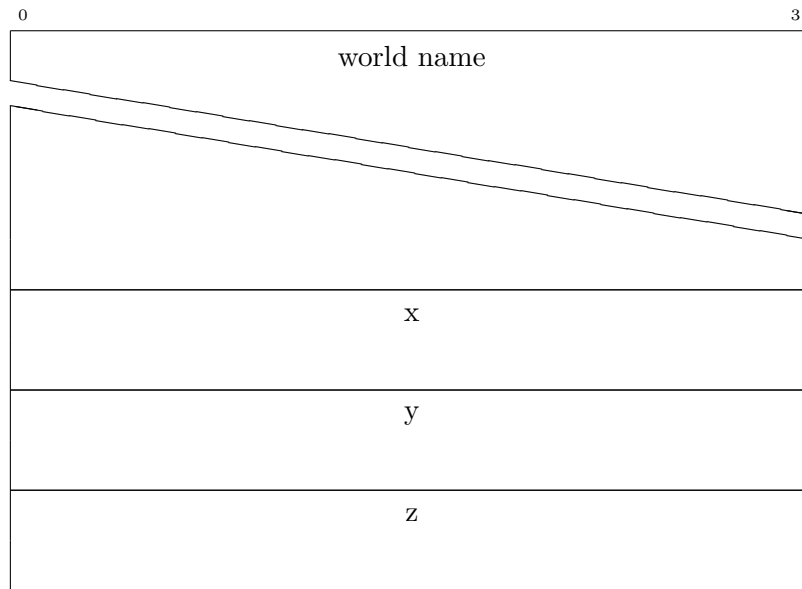
Figure 3.7: Position structure

3.4.9   Block

A block is a 56 bytes argument giving information about its type and (if applicable) properties.

For more information about block properties refer to Appendix A, Blocks.

0							7	8	9	10	11	12	13	14	15
enum value															0
age					direction									axis	
					group_count				delay		eye	hinge	open		
stage					part				rotation						
note					mode				leaves		lit	locked			
					cond	inv	pow								

reserved
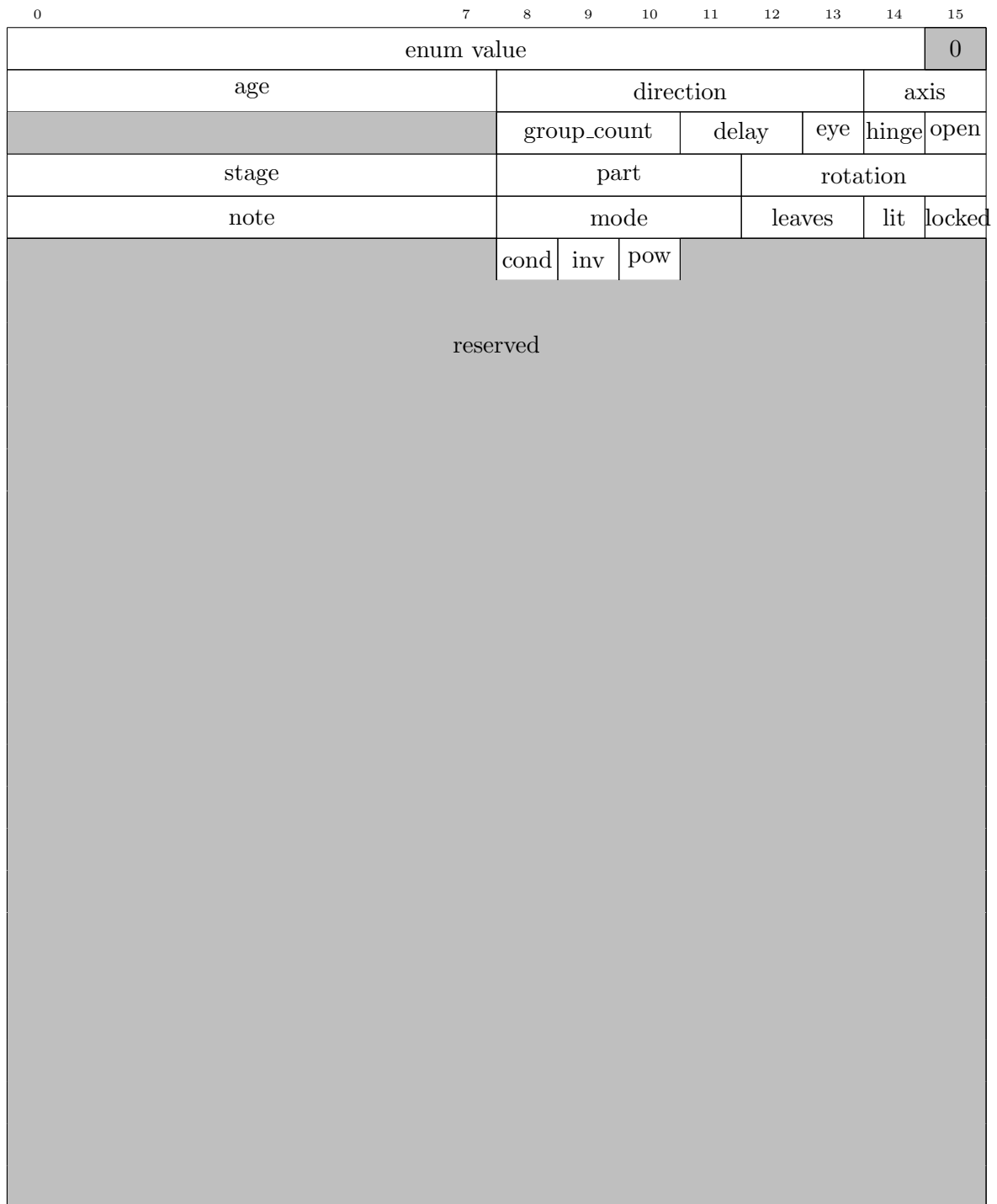
Figure 3.8: Structure of a Block

Enum value	Block name	First Minecraft version
0	AIR	1.8
1	STONE	?
2	GRANITE	?
3	POLISHED_GRANITE	?
4	DIORITE	?
5	POLISHED_DIORITE	?
6	ANDESITE	?
7	POLISHED_ANDESITE	?
8	DEEPSLATE	?
9	COBBLED_DEEPSLATE	?
10	POLISHED_DEEPSLATE	?
11	CALCITE	?
12	TUFF	?
13	DRIPSTONE_BLOCK	?
14	GRASS_BLOCK	?
15	DIRT	?
16	COARSE_DIRT	?
17	PODZOL	?
18	ROOTED_DIRT	?
19	MUD	?
20	CRIMSON_NYLIUM	?
21	WARPED_NYLIUM	?
22	COBBLESTONE	?
23	OAK_PLANKS	?
24	SPRUCE_PLANKS	?
25	BIRCH_PLANKS	?
26	JUNGLE_PLANKS	?
27	ACACIA_PLANKS	?
28	DARK_OAK_PLANKS	?
29	MANGROVE_PLANKS	?

Enum value	Block name	First Minecraft version
30	CRIMSON_PLANKS	?
31	WARPED_PLANKS	?
32	OAK_SAPLING	?
33	SPRUCE_SAPLING	?
34	BIRCH_SAPLING	?
35	JUNGLE_SAPLING	?
36	ACACIA_SAPLING	?
37	DARK_OAK_SAPLING	?
38	MANGROVE_PROPAGULE	?
39	BEDROCK	?
40	SAND	?
41	RED_SAND	?
42	GRAVEL	?
43	COAL_ORE	?
44	DEEPSLATE_COAL_ORE	?
45	IRON_ORE	?
46	DEEPSLATE_IRON_ORE	?
47	COPPER_ORE	?
48	DEEPSLATE_COPPER_ORE	?
49	GOLD_ORE	?
50	DEEPSLATE_GOLD_ORE	?
51	REDSTONE_ORE	?
52	DEEPSLATE_REDSTONE_ORE	?
53	EMERALD_ORE	?
54	DEEPSLATE_EMERALD_ORE	?
55	LAPIS_ORE	?
56	DEEPSLATE_LAPIS_ORE	?
57	DIAMOND_ORE	?
58	DEEPSLATE_DIAMOND_ORE	?
59	NETHER_GOLD_ORE	?

Enum value	Block name	First Minecraft version
60	NETHER_QUARTZ_ORE	?
61	ANCIENT_DEBRIS	?
62	COAL_BLOCK	?
63	RAW_IRON_BLOCK	?
64	RAW_COPPER_BLOCK	?
65	RAW_GOLD_BLOCK	?
66	AMETHYST_BLOCK	?
67	BUDDING_AMETHYST	?
68	IRON_BLOCK	?
69	COPPER_BLOCK	?
70	GOLD_BLOCK	?
71	DIAMOND_BLOCK	?
72	NETHERITE_BLOCK	?
73	EXPOSED_COPPER	?
74	WEATHERED_COPPER	?
75	OXIDIZED_COPPER	?
76	CUT_COPPER	?
77	EXPOSED_CUT_COPPER	?
78	WEATHERED_CUT_COPPER	?
79	OXIDIZED_CUT_COPPER	?
80	CUT_COPPER_STAIRS	?
81	EXPOSED_CUT_COPPER_STAIRS	?
82	WEATHERED_CUT_COPPER_STAIRS	?
83	OXIDIZED_CUT_COPPER_STAIRS	?
84	CUT_COPPER_SLAB	?
85	EXPOSED_CUT_COPPER_SLAB	?
86	WEATHERED_CUT_COPPER_SLAB	?
87	OXIDIZED_CUT_COPPER_SLAB	?
88	WAXED_COPPER_BLOCK	?
89	WAXED_EXPOSED_COPPER	?

Enum value	Block name	First Minecraft version
90	WAXED_WEATHERED_COPPER	?
91	WAXED_OXIDIZED_COPPER	?
92	WAXED_CUT_COPPER	?
93	WAXED_EXPOSED_CUT_COPPER	?
94	WAXED_WEATHERED_CUT_COPPER	?
95	WAXED_OXIDIZED_CUT_COPPER	?
96	WAXED_CUT_COPPER_STAIRS	?
97	WAXED_EXPOSED_CUT_COPPER_STAIRS	?
98	WAXED_WEATHERED_CUT_COPPER_STAIRS	?
99	WAXED_OXIDIZED_CUT_COPPER_STAIRS	?
100	WAXED_CUT_COPPER_SLAB	?
101	WAXED_EXPOSED_CUT_COPPER_SLAB	?
102	WAXED_WEATHERED_CUT_COPPER_SLAB	?
103	WAXED_OXIDIZED_CUT_COPPER_SLAB	?
104	OAK_LOG	?
105	SPRUCE_LOG	?
106	BIRCH_LOG	?
107	JUNGLE_LOG	?
108	ACACIA_LOG	?
109	DARK_OAK_LOG	?
110	MANGROVE_LOG	?
111	MANGROVE_ROOTS	?
112	MUDDY_MANGROVE_ROOTS	?
113	CRIMSON_STEM	?
114	WARPED_STEM	?
115	STRIPPED_OAK_LOG	?
116	STRIPPED_SPRUCE_LOG	?
117	STRIPPED_BIRCH_LOG	?
118	STRIPPED_JUNGLE_LOG	?
119	STRIPPED_ACACIA_LOG	?

Enum value	Block name	First Minecraft version
120	STRIPPED_DARK_OAK_LOG	?
121	STRIPPED_MANGROVE_LOG	?
122	STRIPPED_CRIMSON_STEM	?
123	STRIPPED_WARPED_STEM	?
124	STRIPPED_OAK_WOOD	?
125	STRIPPED_SPRUCE_WOOD	?
126	STRIPPED_BIRCH_WOOD	?
127	STRIPPED_JUNGLE_WOOD	?
128	STRIPPED_ACACIA_WOOD	?
129	STRIPPED_DARK_OAK_WOOD	?
130	STRIPPED_MANGROVE_WOOD	?
131	STRIPPED_CRIMSON_HYPHAE	?
132	STRIPPED_WARPED_HYPHAE	?
133	OAK_WOOD	?
134	SPRUCE_WOOD	?
135	BIRCH_WOOD	?
136	JUNGLE_WOOD	?
137	ACACIA_WOOD	?
138	DARK_OAK_WOOD	?
139	MANGROVE_WOOD	?
140	CRIMSON_HYPHAE	?
141	WARPED_HYPHAE	?
142	OAK_LEAVES	?
143	SPRUCE_LEAVES	?
144	BIRCH_LEAVES	?
145	JUNGLE_LEAVES	?
146	ACACIA_LEAVES	?
147	DARK_OAK_LEAVES	?
148	MANGROVE_LEAVES	?
149	AZALEA_LEAVES	?

Enum value	Block name	First Minecraft version
150	FLOWERING_AZALEA_LEAVES	?
151	SPONGE	?
152	WET_SPONGE	?
153	GLASS	?
154	TINTED_GLASS	?
155	LAPIS_BLOCK	?
156	SANDSTONE	?
157	CHISELED_SANDSTONE	?
158	CUT_SANDSTONE	?
159	COBWEB	?
160	GRASS	?
161	FERN	?
162	AZALEA	?
163	FLOWERING_AZALEA	?
164	DEAD_BUSH	?
165	SEAGRASS	?
166	SEA_PICKLE	?
167	WHITE_WOOL	?
168	ORANGE_WOOL	?
169	MAGENTA_WOOL	?
170	LIGHT_BLUE_WOOL	?
171	YELLOW_WOOL	?
172	LIME_WOOL	?
173	PINK_WOOL	?
174	GRAY_WOOL	?
175	LIGHT_GRAY_WOOL	?
176	CYAN_WOOL	?
177	PURPLE_WOOL	?
178	BLUE_WOOL	?
179	BROWN_WOOL	?

Enum value	Block name	First Minecraft version
180	GREEN_WOOL	?
181	RED_WOOL	?
182	BLACK_WOOL	?
183	DANDELION	?
184	POPPY	?
185	BLUE_ORCHID	?
186	ALLIUM	?
187	AZURE_BLUET	?
188	RED_TULIP	?
189	ORANGE_TULIP	?
190	WHITE_TULIP	?
191	PINK_TULIP	?
192	OXEYE_DAISY	?
193	CORNFLOWER	?
194	LILY_OF_THE_VALLEY	?
195	WITHER_ROSE	?
196	SPORE_BLOSSOM	?
197	BROWN_MUSHROOM	?
198	RED_MUSHROOM	?
199	CRIMSON_FUNGUS	?
200	WARPED_FUNGUS	?
201	CRIMSON_ROOTS	?
202	WARPED_ROOTS	?
203	NETHER_SPROUTS	?
204	WEEPING_VINES	?
205	TWISTING_VINES	?
206	SUGAR_CANE	?
207	KELP	?
208	MOSS_CARPET	?
209	MOSS_BLOCK	?

Enum value	Block name	First Minecraft version
210	HANGING_ROOTS	?
211	BIG_DRIPLEAF	?
212	SMALL_DRIPLEAF	?
213	BAMBOO	?
214	OAK_SLAB	?
215	SPRUCE_SLAB	?
216	BIRCH_SLAB	?
217	JUNGLE_SLAB	?
218	ACACIA_SLAB	?
219	DARK_OAK_SLAB	?
220	MANGROVE_SLAB	?
221	CRIMSON_SLAB	?
222	WARPED_SLAB	?
223	STONE_SLAB	?
224	SMOOTH_STONE_SLAB	?
225	SANDSTONE_SLAB	?
226	CUT_SANDSTONE_SLAB	?
227	PETRIFIED_OAK_SLAB	?
228	COBBLESTONE_SLAB	?
229	BRICK_SLAB	?
230	STONE_BRICK_SLAB	?
231	MUD_BRICK_SLAB	?
232	NETHER_BRICK_SLAB	?
233	QUARTZ_SLAB	?
234	RED_SANDSTONE_SLAB	?
235	CUT_RED_SANDSTONE_SLAB	?
236	PURPUR_SLAB	?
237	PRISMARINE_SLAB	?
238	PRISMARINE_BRICK_SLAB	?
239	DARK_PRISMARINE_SLAB	?

Enum value	Block name	First Minecraft version
240	SMOOTH_QUARTZ	?
241	SMOOTH_RED_SANDSTONE	?
242	SMOOTH_SANDSTONE	?
243	SMOOTH_STONE	?
244	BRICKS	?
245	BOOKSHELF	?
246	MOSSY_COBBLESTONE	?
247	OBSIDIAN	?
248	TORCH	?
249	END_ROD	?
250	CHORUS_PLANT	?
251	CHORUS_FLOWER	?
252	PURPUR_BLOCK	?
253	PURPUR_PILLAR	?
254	PURPUR_STAIRS	?
255	SPAWNER	?
256	CHEST	?
257	CRAFTING_TABLE	?
258	FARMLAND	?
259	FURNACE	?
260	LADDER	?
261	COBBLESTONE_STAIRS	?
262	SNOW	?
263	ICE	?
264	SNOW_BLOCK	?
265	CACTUS	?
266	CLAY	?
267	JUKEBOX	?
268	OAK_FENCE	?
269	SPRUCE_FENCE	?

Enum value	Block name	First Minecraft version
270	BIRCH_FENCE	?
271	JUNGLE_FENCE	?
272	ACACIA_FENCE	?
273	DARK_OAK_FENCE	?
274	MANGROVE_FENCE	?
275	CRIMSON_FENCE	?
276	WARPED_FENCE	?
277	PUMPKIN	?
278	CARVED_PUMPKIN	?
279	JACK_O_LANTERN	?
280	NETHERRACK	?
281	SOUL_SAND	?
282	SOUL_SOIL	?
283	BASALT	?
284	POLISHED_BASALT	?
285	SMOOTH_BASALT	?
286	SOUL_TORCH	?
287	GLOWSTONE	?
288	INFESTED_STONE	?
289	INFESTED_COBBLESTONE	?
290	INFESTED_STONE_BRICKS	?
291	INFESTED_MOSSY_STONE_BRICKS	?
292	INFESTED_CRACKED_STONE_BRICKS	?
293	INFESTED_CHISELED_STONE_BRICKS	?
294	INFESTED_DEEPSLATE	?
295	STONE_BRICKS	?
296	MOSSY_STONE_BRICKS	?
297	CRACKED_STONE_BRICKS	?
298	CHISELED_STONE_BRICKS	?
299	PACKED_MUD	?

Enum value	Block name	First Minecraft version
300	MUD_BRICKS	?
301	DEEPSLATE_BRICKS	?
302	CRACKED_DEEPSLATE_BRICKS	?
303	DEEPSLATE_TILES	?
304	CRACKED_DEEPSLATE_TILES	?
305	CHISELED_DEEPSLATE	?
306	REINFORCED_DEEPSLATE	?
307	BROWN_MUSHROOM_BLOCK	?
308	RED_MUSHROOM_BLOCK	?
309	MUSHROOM_STEM	?
310	IRON_BARS	?
311	CHAIN	?
312	GLASS_PANE	?
313	MELON	?
314	VINE	?
315	GLOW_LICHEN	?
316	BRICK_STAIRS	?
317	STONE_BRICK_STAIRS	?
318	MUD_BRICK_STAIRS	?
319	MYCELIUM	?
320	LILY_PAD	?
321	NETHER_BRICKS	?
322	CRACKED_NETHER_BRICKS	?
323	CHISELED_NETHER_BRICKS	?
324	NETHER_BRICK_FENCE	?
325	NETHER_BRICK_STAIRS	?
326	SCULK	?
327	SCULK_VEIN	?
328	SCULK_CATALYST	?
329	SCULK_SHRIEKER	?

Enum value	Block name	First Minecraft version
330	ENCHANTING_TABLE	?
331	END_PORTAL_FRAME	?
332	END_STONE	?
333	END_STONE_BRICKS	?
334	DRAGON_EGG	?
335	SANDSTONE_STAIRS	?
336	ENDER_CHEST	?
337	EMERALD_BLOCK	?
338	OAK_STAIRS	?
339	SPRUCE_STAIRS	?
340	BIRCH_STAIRS	?
341	JUNGLE_STAIRS	?
342	ACACIA_STAIRS	?
343	DARK_OAK_STAIRS	?
344	MANGROVE_STAIRS	?
345	CRIMSON_STAIRS	?
346	WARPED_STAIRS	?
347	COMMAND_BLOCK	?
348	BEACON	?
349	COBBLESTONE_WALL	?
350	MOSSY_COBBLESTONE_WALL	?
351	BRICK_WALL	?
352	PRISMARINE_WALL	?
353	RED_SANDSTONE_WALL	?
354	MOSSY_STONE_BRICK_WALL	?
355	GRANITE_WALL	?
356	STONE_BRICK_WALL	?
357	MUD_BRICK_WALL	?
358	NETHER_BRICK_WALL	?
359	ANDESITE_WALL	?

Enum value	Block name	First Minecraft version
360	RED_NETHER_BRICK_WALL	?
361	SANDSTONE_WALL	?
362	END_STONE_BRICK_WALL	?
363	DIORITE_WALL	?
364	BLACKSTONE_WALL	?
365	POLISHED_BLACKSTONE_WALL	?
366	POLISHED_BLACKSTONE_BRICK_WALL	?
367	COBBLED_DEEPSLATE_WALL	?
368	POLISHED_DEEPSLATE_WALL	?
369	DEEPSLATE_BRICK_WALL	?
370	DEEPSLATE_TILE_WALL	?
371	ANVIL	?
372	CHIPPED_ANVIL	?
373	DAMAGED_ANVIL	?
374	CHISELED_QUARTZ_BLOCK	?
375	QUARTZ_BLOCK	?
376	QUARTZ_BRICKS	?
377	QUARTZ_PILLAR	?
378	QUARTZ_STAIRS	?
379	WHITE_TERRACOTTA	?
380	ORANGE_TERRACOTTA	?
381	MAGENTA_TERRACOTTA	?
382	LIGHT_BLUE_TERRACOTTA	?
383	YELLOW_TERRACOTTA	?
384	LIME_TERRACOTTA	?
385	PINK_TERRACOTTA	?
386	GRAY_TERRACOTTA	?
387	LIGHT_GRAY_TERRACOTTA	?
388	CYAN_TERRACOTTA	?
389	PURPLE_TERRACOTTA	?

Enum value	Block name	First Minecraft version
390	BLUE_TERRACOTTA	?
391	BROWN_TERRACOTTA	?
392	GREEN_TERRACOTTA	?
393	RED_TERRACOTTA	?
394	BLACK_TERRACOTTA	?
395	BARRIER	?
396	LIGHT	?
397	HAY_BLOCK	?
398	WHITE_CARPET	?
399	ORANGE_CARPET	?
400	MAGENTA_CARPET	?
401	LIGHT_BLUE_CARPET	?
402	YELLOW_CARPET	?
403	LIME_CARPET	?
404	PINK_CARPET	?
405	GRAY_CARPET	?
406	LIGHT_GRAY_CARPET	?
407	CYAN_CARPET	?
408	PURPLE_CARPET	?
409	BLUE_CARPET	?
410	BROWN_CARPET	?
411	GREEN_CARPET	?
412	RED_CARPET	?
413	BLACK_CARPET	?
414	TERRACOTTA	?
415	PACKED_ICE	?
416	DIRT_PATH	?
417	SUNFLOWER	?
418	LILAC	?
419	ROSE_BUSH	?

Enum value	Block name	First Minecraft version
420	PEONY	?
421	TALL_GRASS	?
422	LARGE_FERN	?
423	WHITE_STAINED_GLASS	?
424	ORANGE_STAINED_GLASS	?
425	MAGENTA_STAINED_GLASS	?
426	LIGHT_BLUE_STAINED_GLASS	?
427	YELLOW_STAINED_GLASS	?
428	LIME_STAINED_GLASS	?
429	PINK_STAINED_GLASS	?
430	GRAY_STAINED_GLASS	?
431	LIGHT_GRAY_STAINED_GLASS	?
432	CYAN_STAINED_GLASS	?
433	PURPLE_STAINED_GLASS	?
434	BLUE_STAINED_GLASS	?
435	BROWN_STAINED_GLASS	?
436	GREEN_STAINED_GLASS	?
437	RED_STAINED_GLASS	?
438	BLACK_STAINED_GLASS	?
439	WHITE_STAINED_GLASS_PANE	?
440	ORANGE_STAINED_GLASS_PANE	?
441	MAGENTA_STAINED_GLASS_PANE	?
442	LIGHT_BLUE_STAINED_GLASS_PANE	?
443	YELLOW_STAINED_GLASS_PANE	?
444	LIME_STAINED_GLASS_PANE	?
445	PINK_STAINED_GLASS_PANE	?
446	GRAY_STAINED_GLASS_PANE	?
447	LIGHT_GRAY_STAINED_GLASS_PANE	?
448	CYAN_STAINED_GLASS_PANE	?
449	PURPLE_STAINED_GLASS_PANE	?

Enum value	Block name	First Minecraft version
450	BLUE_STAINED_GLASS_PANE	?
451	BROWN_STAINED_GLASS_PANE	?
452	GREEN_STAINED_GLASS_PANE	?
453	RED_STAINED_GLASS_PANE	?
454	BLACK_STAINED_GLASS_PANE	?
455	PRISMARINE	?
456	PRISMARINE_BRICKS	?
457	DARK_PRISMARINE	?
458	PRISMARINE_STAIRS	?
459	PRISMARINE_BRICK_STAIRS	?
460	DARK_PRISMARINE_STAIRS	?
461	SEA_LANTERN	?
462	RED_SANDSTONE	?
463	CHISELED_RED_SANDSTONE	?
464	CUT_RED_SANDSTONE	?
465	RED_SANDSTONE_STAIRS	?
466	REPEATING_COMMAND_BLOCK	?
467	CHAIN_COMMAND_BLOCK	?
468	MAGMA_BLOCK	?
469	NETHER_WART_BLOCK	?
470	WARPED_WART_BLOCK	?
471	RED_NETHER_BRICKS	?
472	BONE_BLOCK	?
473	STRUCTURE_VOID	?
474	SHULKER_BOX	?
475	WHITE_SHULKER_BOX	?
476	ORANGE_SHULKER_BOX	?
477	MAGENTA_SHULKER_BOX	?
478	LIGHT_BLUE_SHULKER_BOX	?
479	YELLOW_SHULKER_BOX	?

Enum value	Block name	First Minecraft version
480	LIME_SHULKER_BOX	?
481	PINK_SHULKER_BOX	?
482	GRAY_SHULKER_BOX	?
483	LIGHT_GRAY_SHULKER_BOX	?
484	CYAN_SHULKER_BOX	?
485	PURPLE_SHULKER_BOX	?
486	BLUE_SHULKER_BOX	?
487	BROWN_SHULKER_BOX	?
488	GREEN_SHULKER_BOX	?
489	RED_SHULKER_BOX	?
490	BLACK_SHULKER_BOX	?
491	WHITE_GLAZED_TERRACOTTA	?
492	ORANGE_GLAZED_TERRACOTTA	?
493	MAGENTA_GLAZED_TERRACOTTA	?
494	LIGHT_BLUE_GLAZED_TERRACOTTA	?
495	YELLOW_GLAZED_TERRACOTTA	?
496	LIME_GLAZED_TERRACOTTA	?
497	PINK_GLAZED_TERRACOTTA	?
498	GRAY_GLAZED_TERRACOTTA	?
499	LIGHT_GRAY_GLAZED_TERRACOTTA	?
500	CYAN_GLAZED_TERRACOTTA	?
501	PURPLE_GLAZED_TERRACOTTA	?
502	BLUE_GLAZED_TERRACOTTA	?
503	BROWN_GLAZED_TERRACOTTA	?
504	GREEN_GLAZED_TERRACOTTA	?
505	RED_GLAZED_TERRACOTTA	?
506	BLACK_GLAZED_TERRACOTTA	?
507	WHITE_CONCRETE	?
508	ORANGE_CONCRETE	?
509	MAGENTA_CONCRETE	?

Enum value	Block name	First Minecraft version
510	LIGHT_BLUE_CONCRETE	?
511	YELLOW_CONCRETE	?
512	LIME_CONCRETE	?
513	PINK_CONCRETE	?
514	GRAY_CONCRETE	?
515	LIGHT_GRAY_CONCRETE	?
516	CYAN_CONCRETE	?
517	PURPLE_CONCRETE	?
518	BLUE_CONCRETE	?
519	BROWN_CONCRETE	?
520	GREEN_CONCRETE	?
521	RED_CONCRETE	?
522	BLACK_CONCRETE	?
523	WHITE_CONCRETE_POWDER	?
524	ORANGE_CONCRETE_POWDER	?
525	MAGENTA_CONCRETE_POWDER	?
526	LIGHT_BLUE_CONCRETE_POWDER	?
527	YELLOW_CONCRETE_POWDER	?
528	LIME_CONCRETE_POWDER	?
529	PINK_CONCRETE_POWDER	?
530	GRAY_CONCRETE_POWDER	?
531	LIGHT_GRAY_CONCRETE_POWDER	?
532	CYAN_CONCRETE_POWDER	?
533	PURPLE_CONCRETE_POWDER	?
534	BLUE_CONCRETE_POWDER	?
535	BROWN_CONCRETE_POWDER	?
536	GREEN_CONCRETE_POWDER	?
537	RED_CONCRETE_POWDER	?
538	BLACK_CONCRETE_POWDER	?
539	TURTLE_EGG	?

Enum value	Block name	First Minecraft version
540	DEAD_TUBE_CORAL_BLOCK	?
541	DEAD_BRAIN_CORAL_BLOCK	?
542	DEAD_BUBBLE_CORAL_BLOCK	?
543	DEAD_FIRE_CORAL_BLOCK	?
544	DEAD_HORN_CORAL_BLOCK	?
545	TUBE_CORAL_BLOCK	?
546	BRAIN_CORAL_BLOCK	?
547	BUBBLE_CORAL_BLOCK	?
548	FIRE_CORAL_BLOCK	?
549	HORN_CORAL_BLOCK	?
550	TUBE_CORAL	?
551	BRAIN_CORAL	?
552	BUBBLE_CORAL	?
553	FIRE_CORAL	?
554	HORN_CORAL	?
555	DEAD_BRAIN_CORAL	?
556	DEAD_BUBBLE_CORAL	?
557	DEAD_FIRE_CORAL	?
558	DEAD_HORN_CORAL	?
559	DEAD_TUBE_CORAL	?
560	TUBE_CORAL_FAN	?
561	BRAIN_CORAL_FAN	?
562	BUBBLE_CORAL_FAN	?
563	FIRE_CORAL_FAN	?
564	HORN_CORAL_FAN	?
565	DEAD_TUBE_CORAL_FAN	?
566	DEAD_BRAIN_CORAL_FAN	?
567	DEAD_BUBBLE_CORAL_FAN	?
568	DEAD_FIRE_CORAL_FAN	?
569	DEAD_HORN_CORAL_FAN	?

Enum value	Block name	First Minecraft version
570	BLUE_ICE	?
571	CONDUIT	?
572	POLISHED_GRANITE_STAIRS	?
573	SMOOTH_RED_SANDSTONE_STAIRS	?
574	MOSSY_STONE_BRICK_STAIRS	?
575	POLISHED_DIORITE_STAIRS	?
576	MOSSY_COBBLESTONE_STAIRS	?
577	END_STONE_BRICK_STAIRS	?
578	STONE_STAIRS	?
579	SMOOTH_SANDSTONE_STAIRS	?
580	SMOOTH_QUARTZ_STAIRS	?
581	GRANITE_STAIRS	?
582	ANDESITE_STAIRS	?
583	RED_NETHER_BRICK_STAIRS	?
584	POLISHED_ANDESITE_STAIRS	?
585	DIORITE_STAIRS	?
586	COBBLED_DEEPSLATE_STAIRS	?
587	POLISHED_DEEPSLATE_STAIRS	?
588	DEEPSLATE_BRICK_STAIRS	?
589	DEEPSLATE_TILE_STAIRS	?
590	POLISHED_GRANITE_SLAB	?
591	SMOOTH_RED_SANDSTONE_SLAB	?
592	MOSSY_STONE_BRICK_SLAB	?
593	POLISHED_DIORITE_SLAB	?
594	MOSSY_COBBLESTONE_SLAB	?
595	END_STONE_BRICK_SLAB	?
596	SMOOTH_SANDSTONE_SLAB	?
597	SMOOTH_QUARTZ_SLAB	?
598	GRANITE_SLAB	?
599	ANDESITE_SLAB	?

Enum value	Block name	First Minecraft version
600	RED_NETHER_BRICK_SLAB	?
601	POLISHED_ANDESITE_SLAB	?
602	DIORITE_SLAB	?
603	COBBLED_DEEPSLATE_SLAB	?
604	POLISHED_DEEPSLATE_SLAB	?
605	DEEPSLATE_BRICK_SLAB	?
606	DEEPSLATE_TILE_SLAB	?
607	SCAFFOLDING	?
608	REDSTONE_TORCH	?
609	REDSTONE_BLOCK	?
610	REPEATER	?
611	COMPARATOR	?
612	PISTON	?
613	STICKY_PISTON	?
614	SLIME_BLOCK	?
615	HONEY_BLOCK	?
616	OBSERVER	?
617	HOPPER	?
618	DISPENSER	?
619	DROPPER	?
620	LECTERN	?
621	TARGET	?
622	LEVER	?
623	LIGHTNING_ROD	?
624	DAYLIGHT_DETECTOR	?
625	SCULK_SENSOR	?
626	TRIPWIRE_HOOK	?
627	TRAPPED_CHEST	?
628	TNT	?
629	REDSTONE_LAMP	?

Enum value	Block name	First Minecraft version
630	NOTE_BLOCK	?
631	STONE_BUTTON	?
632	POLISHED_BLACKSTONE_BUTTON	?
633	OAK_BUTTON	?
634	SPRUCE_BUTTON	?
635	BIRCH_BUTTON	?
636	JUNGLE_BUTTON	?
637	ACACIA_BUTTON	?
638	DARK_OAK_BUTTON	?
639	MANGROVE_BUTTON	?
640	CRIMSON_BUTTON	?
641	WARPED_BUTTON	?
642	STONE_PRESSURE_PLATE	?
643	POLISHED_BLACKSTONE_PRESSURE_PLATE	?
644	LIGHT_WEIGHTED_PRESSURE_PLATE	?
645	HEAVY_WEIGHTED_PRESSURE_PLATE	?
646	OAK_PRESSURE_PLATE	?
647	SPRUCE_PRESSURE_PLATE	?
648	BIRCH_PRESSURE_PLATE	?
649	JUNGLE_PRESSURE_PLATE	?
650	ACACIA_PRESSURE_PLATE	?
651	DARK_OAK_PRESSURE_PLATE	?
652	MANGROVE_PRESSURE_PLATE	?
653	CRIMSON_PRESSURE_PLATE	?
654	WARPED_PRESSURE_PLATE	?
655	IRON_DOOR	?
656	OAK_DOOR	?
657	SPRUCE_DOOR	?
658	BIRCH_DOOR	?
659	JUNGLE_DOOR	?

Enum value	Block name	First Minecraft version
660	ACACIA_DOOR	?
661	DARK_OAK_DOOR	?
662	MANGROVE_DOOR	?
663	CRIMSON_DOOR	?
664	WARPED_DOOR	?
665	IRON_TRAPDOOR	?
666	OAK_TRAPDOOR	?
667	SPRUCE_TRAPDOOR	?
668	BIRCH_TRAPDOOR	?
669	JUNGLE_TRAPDOOR	?
670	ACACIA_TRAPDOOR	?
671	DARK_OAK_TRAPDOOR	?
672	MANGROVE_TRAPDOOR	?
673	CRIMSON_TRAPDOOR	?
674	WARPED_TRAPDOOR	?
675	OAK_FENCE_GATE	?
676	SPRUCE_FENCE_GATE	?
677	BIRCH_FENCE_GATE	?
678	JUNGLE_FENCE_GATE	?
679	ACACIA_FENCE_GATE	?
680	DARK_OAK_FENCE_GATE	?
681	MANGROVE_FENCE_GATE	?
682	CRIMSON_FENCE_GATE	?
683	WARPED_FENCE_GATE	?
684	POWERED_RAIL	?
685	DETECTOR_RAIL	?
686	RAIL	?
687	ACTIVATOR_RAIL	?
688	STRUCTURE_BLOCK	?
689	JIGSAW	?

Enum value	Block name	First Minecraft version
690	WHEAT	?
691	OAK_SIGN	?
692	SPRUCE_SIGN	?
693	BIRCH_SIGN	?
694	JUNGLE_SIGN	?
695	ACACIA_SIGN	?
696	DARK_OAK_SIGN	?
697	MANGROVE_SIGN	?
698	CRIMSON_SIGN	?
699	WARPED_SIGN	?
700	DRIED_KELP_BLOCK	?
701	CAKE	?
702	WHITE_BED	?
703	ORANGE_BED	?
704	MAGENTA_BED	?
705	LIGHT_BLUE_BED	?
706	YELLOW_BED	?
707	LIME_BED	?
708	PINK_BED	?
709	GRAY_BED	?
710	LIGHT_GRAY_BED	?
711	CYAN_BED	?
712	PURPLE_BED	?
713	BLUE_BED	?
714	BROWN_BED	?
715	GREEN_BED	?
716	RED_BED	?
717	BLACK_BED	?
718	NETHER_WART	?
719	BREWING_STAND	?

Enum value	Block name	First Minecraft version
720	CAULDRON	?
721	FLOWER_POT	?
722	SKELETON_SKULL	?
723	WITHER_SKELETON_SKULL	?
724	PLAYER_HEAD	?
725	ZOMBIE_HEAD	?
726	CREEPER_HEAD	?
727	DRAGON_HEAD	?
728	WHITE_BANNER	?
729	ORANGE_BANNER	?
730	MAGENTA_BANNER	?
731	LIGHT_BLUE_BANNER	?
732	YELLOW_BANNER	?
733	LIME_BANNER	?
734	PINK_BANNER	?
735	GRAY_BANNER	?
736	LIGHT_GRAY_BANNER	?
737	CYAN_BANNER	?
738	PURPLE_BANNER	?
739	BLUE_BANNER	?
740	BROWN_BANNER	?
741	GREEN_BANNER	?
742	RED_BANNER	?
743	BLACK_BANNER	?
744	LOOM	?
745	COMPOSTER	?
746	BARREL	?
747	SMOKER	?
748	BLAST_FURNACE	?
749	CARTOGRAPHY_TABLE	?

Enum value	Block name	First Minecraft version
750	FLETCHING_TABLE	?
751	GRINDSTONE	?
752	SMITHING_TABLE	?
753	STONECUTTER	?
754	BELL	?
755	LANTERN	?
756	SOUL_LANTERN	?
757	CAMPFIRE	?
758	SOUL_CAMPFIRE	?
759	SHROOMLIGHT	?
760	BEE_NEST	?
761	BEEHIVE	?
762	HONEYCOMB_BLOCK	?
763	LODESTONE	?
764	CRYING_OBSIDIAN	?
765	BLACKSTONE	?
766	BLACKSTONE_SLAB	?
767	BLACKSTONE_STAIRS	?
768	GILDED_BLACKSTONE	?
769	POLISHED_BLACKSTONE	?
770	POLISHED_BLACKSTONE_SLAB	?
771	POLISHED_BLACKSTONE_STAIRS	?
772	CHISELED_POLISHED_BLACKSTONE	?
773	POLISHED_BLACKSTONE_BRICKS	?
774	POLISHED_BLACKSTONE_BRICK_SLAB	?
775	POLISHED_BLACKSTONE_BRICK_STAIRS	?
776	CRACKED_POLISHED_BLACKSTONE_BRICKS	?
777	RESPAWN_ANCHOR	?
778	CANDLE	?
779	WHITE_CANDLE	?

Enum value	Block name	First Minecraft version
780	ORANGE_CANDLE	?
781	MAGENTA_CANDLE	?
782	LIGHT_BLUE_CANDLE	?
783	YELLOW_CANDLE	?
784	LIME_CANDLE	?
785	PINK_CANDLE	?
786	GRAY_CANDLE	?
787	LIGHT_GRAY_CANDLE	?
788	CYAN_CANDLE	?
789	PURPLE_CANDLE	?
790	BLUE_CANDLE	?
791	BROWN_CANDLE	?
792	GREEN_CANDLE	?
793	RED_CANDLE	?
794	BLACK_CANDLE	?
795	SMALL_AMETHYST_BUD	?
796	MEDIUM_AMETHYST_BUD	?
797	LARGE_AMETHYST_BUD	?
798	AMETHYST_CLUSTER	?
799	POINTED_DRIPSTONE	?
800	OCHRE_FROGLIGHT	?
801	VERDANT_FROGLIGHT	?
802	PEARLESCENT_FROGLIGHT	?
803	FROGSPAWN	?
804	WATER	?
805	LAVA	?
806	TALL_SEAGRASS	?
807	PISTON_HEAD	?
808	MOVING_PISTON	?
809	WALL_TORCH	?

Enum value	Block name	First Minecraft version
810	FIRE	?
811	SOUL_FIRE	?
812	REDSTONE_WIRE	?
813	OAK_WALL_SIGN	?
814	SPRUCE_WALL_SIGN	?
815	BIRCH_WALL_SIGN	?
816	ACACIA_WALL_SIGN	?
817	JUNGLE_WALL_SIGN	?
818	DARK_OAK_WALL_SIGN	?
819	MANGROVE_WALL_SIGN	?
820	REDSTONE_WALL_TORCH	?
821	SOUL_WALL_TORCH	?
822	NETHER_PORTAL	?
823	ATTACHED_PUMPKIN_STEM	?
824	ATTACHED_MELON_STEM	?
825	PUMPKIN_STEM	?
826	MELON_STEM	?
827	WATER_CAULDRON	?
828	LAVA_CAULDRON	?
829	POWDER_SNOW_CAULDRON	?
830	END_PORTAL	?
831	COCOA	?
832	TRIPWIRE	?
833	POTTED_OAK_SAPLING	?
834	POTTED_SPRUCE_SAPLING	?
835	POTTED_BIRCH_SAPLING	?
836	POTTED_JUNGLE_SAPLING	?
837	POTTED_ACACIA_SAPLING	?
838	POTTED_DARK_OAK_SAPLING	?
839	POTTED_MANGROVE_PROPAGULE	?

Enum value	Block name	First Minecraft version
840	POTTED_FERN	?
841	POTTED_DANDELION	?
842	POTTED_POPPY	?
843	POTTED_BLUE_ORCHID	?
844	POTTED_ALLIUM	?
845	POTTED_AZURE_BLUET	?
846	POTTED_RED_TULIP	?
847	POTTED_ORANGE_TULIP	?
848	POTTED_WHITE_TULIP	?
849	POTTED_PINK_TULIP	?
850	POTTED_OXEYE_DAISY	?
851	POTTED_CORNFLOWER	?
852	POTTED_LILY_OF_THE_VALLEY	?
853	POTTED_WITHER_ROSE	?
854	POTTED_RED_MUSHROOM	?
855	POTTED_BROWN_MUSHROOM	?
856	POTTED_DEAD_BUSH	?
857	POTTED_CACTUS	?
858	CARROTS	?
859	POTATOES	?
860	SKELETON_WALL_SKULL	?
861	WITHER_SKELETON_WALL_SKULL	?
862	ZOMBIE_WALL_HEAD	?
863	PLAYER_WALL_HEAD	?
864	CREEPER_WALL_HEAD	?
865	DRAGON_WALL_HEAD	?
866	WHITE_WALL_BANNER	?
867	ORANGE_WALL_BANNER	?
868	MAGENTA_WALL_BANNER	?
869	LIGHT_BLUE_WALL_BANNER	?

Enum value	Block name	First Minecraft version
870	YELLOW_WALL_BANNER	?
871	LIME_WALL_BANNER	?
872	PINK_WALL_BANNER	?
873	GRAY_WALL_BANNER	?
874	LIGHT_GRAY_WALL_BANNER	?
875	CYAN_WALL_BANNER	?
876	PURPLE_WALL_BANNER	?
877	BLUE_WALL_BANNER	?
878	BROWN_WALL_BANNER	?
879	GREEN_WALL_BANNER	?
880	RED_WALL_BANNER	?
881	BLACK_WALL_BANNER	?
882	BEETROOTS	?
883	END_GATEWAY	?
884	FROSTED_ICE	?
885	KELP_PLANT	?
886	DEAD_TUBE_CORAL_WALL_FAN	?
887	DEAD_BRAIN_CORAL_WALL_FAN	?
888	DEAD_BUBBLE_CORAL_WALL_FAN	?
889	DEAD_FIRE_CORAL_WALL_FAN	?
890	DEAD_HORN_CORAL_WALL_FAN	?
891	TUBE_CORAL_WALL_FAN	?
892	BRAIN_CORAL_WALL_FAN	?
893	BUBBLE_CORAL_WALL_FAN	?
894	FIRE_CORAL_WALL_FAN	?
895	HORN_CORAL_WALL_FAN	?
896	BAMBOO_SAPLING	?
897	POTTED_BAMBOO	?
898	BUBBLE_COLUMN	?
899	SWEET_BERRY_BUSH	?

Enum value	Block name	First Minecraft version
900	WEEPING_VINES_PLANT	?
901	TWISTING_VINES_PLANT	?
902	CRIMSON_WALL_SIGN	?
903	WARPED_WALL_SIGN	?
904	POTTED_CRIMSON_FUNGUS	?
905	POTTED_WARPED_FUNGUS	?
906	POTTED_CRIMSON_ROOTS	?
907	POTTED_WARPED_ROOTS	?
908	CANDLE_CAKE	?
909	WHITE_CANDLE_CAKE	?
910	ORANGE_CANDLE_CAKE	?
911	MAGENTA_CANDLE_CAKE	?
912	LIGHT_BLUE_CANDLE_CAKE	?
913	YELLOW_CANDLE_CAKE	?
914	LIME_CANDLE_CAKE	?
915	PINK_CANDLE_CAKE	?
916	GRAY_CANDLE_CAKE	?
917	LIGHT_GRAY_CANDLE_CAKE	?
918	CYAN_CANDLE_CAKE	?
919	PURPLE_CANDLE_CAKE	?
920	BLUE_CANDLE_CAKE	?
921	BROWN_CANDLE_CAKE	?
922	GREEN_CANDLE_CAKE	?
923	RED_CANDLE_CAKE	?
924	BLACK_CANDLE_CAKE	?
925	POWDER_SNOW	?
926	CAVE_VINES	?
927	CAVE_VINES_PLANT	?
928	BIG_DRIPLEAF_STEM	?
929	POTTED_AZALEA_BUSH	?

Enum value	Block name	First Minecraft version
930	POTTED_FLOWERING_AZALEA_BUSH	?

Table 3.2: Block enum

3.4.10   Item

This section is under construction, as detailed properties cannot be specified.

An item specifies its type and its amount.
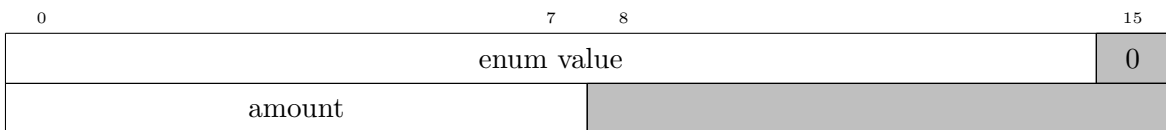
Figure 3.9: Structure of an Item

Enum value	Item name	First Minecraft version
0	AIR	1.8
1	STONE	?
2	GRANITE	?
3	POLISHED_GRANITE	?
4	DIORITE	?
5	POLISHED_DIORITE	?
6	ANDESITE	?
7	POLISHED_ANDESITE	?
8	DEEPSLATE	?
9	COBBLED_DEEPSLATE	?
10	POLISHED_DEEPSLATE	?
11	CALCITE	?
12	TUFF	?
13	DRIPSTONE_BLOCK	?
14	GRASS_BLOCK	?
15	DIRT	?

Enum value	Item name	First Minecraft version
16	COARSE_DIRT	?
17	PODZOL	?
18	ROOTED_DIRT	?
19	MUD	?
20	CRIMSON_NYLIUM	?
21	WARPED_NYLIUM	?
22	COBBLESTONE	?
23	OAK_PLANKS	?
24	SPRUCE_PLANKS	?
25	BIRCH_PLANKS	?
26	JUNGLE_PLANKS	?
27	ACACIA_PLANKS	?
28	DARK_OAK_PLANKS	?
29	MANGROVE_PLANKS	?
30	CRIMSON_PLANKS	?
31	WARPED_PLANKS	?
32	OAK_SAPLING	?
33	SPRUCE_SAPLING	?
34	BIRCH_SAPLING	?
35	JUNGLE_SAPLING	?
36	ACACIA_SAPLING	?
37	DARK_OAK_SAPLING	?
38	MANGROVE_PROPAGULE	?
39	BEDROCK	?
40	SAND	?
41	RED_SAND	?
42	GRAVEL	?
43	COAL_ORE	?
44	DEEPSLATE_COAL_ORE	?
45	IRON_ORE	?

Enum value	Item name	First Minecraft version
46	DEEPSLATE_IRON_ORE	?
47	COPPER_ORE	?
48	DEEPSLATE_COPPER_ORE	?
49	GOLD_ORE	?
50	DEEPSLATE_GOLD_ORE	?
51	REDSTONE_ORE	?
52	DEEPSLATE_REDSTONE_ORE	?
53	EMERALD_ORE	?
54	DEEPSLATE_EMERALD_ORE	?
55	LAPIS_ORE	?
56	DEEPSLATE_LAPIS_ORE	?
57	DIAMOND_ORE	?
58	DEEPSLATE_DIAMOND_ORE	?
59	NETHER_GOLD_ORE	?
60	NETHER_QUARTZ_ORE	?
61	ANCIENT_DEBRIS	?
62	COAL_BLOCK	?
63	RAW_IRON_BLOCK	?
64	RAW_COPPER_BLOCK	?
65	RAW_GOLD_BLOCK	?
66	AMETHYST_BLOCK	?
67	BUDDING_AMETHYST	?
68	IRON_BLOCK	?
69	COPPER_BLOCK	?
70	GOLD_BLOCK	?
71	DIAMOND_BLOCK	?
72	NETHERITE_BLOCK	?
73	EXPOSED_COPPER	?
74	WEATHERED_COPPER	?
75	OXIDIZED_COPPER	?

Enum value	Item name	First Minecraft version
76	CUT_COPPER	?
77	EXPOSED_CUT_COPPER	?
78	WEATHERED_CUT_COPPER	?
79	OXIDIZED_CUT_COPPER	?
80	CUT_COPPER_STAIRS	?
81	EXPOSED_CUT_COPPER_STAIRS	?
82	WEATHERED_CUT_COPPER_STAIRS	?
83	OXIDIZED_CUT_COPPER_STAIRS	?
84	CUT_COPPER_SLAB	?
85	EXPOSED_CUT_COPPER_SLAB	?
86	WEATHERED_CUT_COPPER_SLAB	?
87	OXIDIZED_CUT_COPPER_SLAB	?
88	WAXED_COPPER_BLOCK	?
89	WAXED_EXPOSED_COPPER	?
90	WAXED_WEATHERED_COPPER	?
91	WAXED_OXIDIZED_COPPER	?
92	WAXED_CUT_COPPER	?
93	WAXED_EXPOSED_CUT_COPPER	?
94	WAXED_WEATHERED_CUT_COPPER	?
95	WAXED_OXIDIZED_CUT_COPPER	?
96	WAXED_CUT_COPPER_STAIRS	?
97	WAXED_EXPOSED_CUT_COPPER_STAIRS	?
98	WAXED_WEATHERED_CUT_COPPER_STAIRS	?
99	WAXED_OXIDIZED_CUT_COPPER_STAIRS	?
100	WAXED_CUT_COPPER_SLAB	?
101	WAXED_EXPOSED_CUT_COPPER_SLAB	?
102	WAXED_WEATHERED_CUT_COPPER_SLAB	?
103	WAXED_OXIDIZED_CUT_COPPER_SLAB	?
104	OAK_LOG	?
105	SPRUCE_LOG	?

Enum value	Item name	First Minecraft version
106	BIRCH_LOG	?
107	JUNGLE_LOG	?
108	ACACIA_LOG	?
109	DARK_OAK_LOG	?
110	MANGROVE_LOG	?
111	MANGROVE_ROOTS	?
112	MUDDY_MANGROVE_ROOTS	?
113	CRIMSON_STEM	?
114	WARPED_STEM	?
115	STRIPPED_OAK_LOG	?
116	STRIPPED_SPRUCE_LOG	?
117	STRIPPED_BIRCH_LOG	?
118	STRIPPED_JUNGLE_LOG	?
119	STRIPPED_ACACIA_LOG	?
120	STRIPPED_DARK_OAK_LOG	?
121	STRIPPED_MANGROVE_LOG	?
122	STRIPPED_CRIMSON_STEM	?
123	STRIPPED_WARPED_STEM	?
124	STRIPPED_OAK_WOOD	?
125	STRIPPED_SPRUCE_WOOD	?
126	STRIPPED_BIRCH_WOOD	?
127	STRIPPED_JUNGLE_WOOD	?
128	STRIPPED_ACACIA_WOOD	?
129	STRIPPED_DARK_OAK_WOOD	?
130	STRIPPED_MANGROVE_WOOD	?
131	STRIPPED_CRIMSON_HYPHAE	?
132	STRIPPED_WARPED_HYPHAE	?
133	OAK_WOOD	?
134	SPRUCE_WOOD	?
135	BIRCH_WOOD	?

Enum value	Item name	First Minecraft version
136	JUNGLE_WOOD	?
137	ACACIA_WOOD	?
138	DARK_OAK_WOOD	?
139	MANGROVE_WOOD	?
140	CRIMSON_HYPHAE	?
141	WARPED_HYPHAE	?
142	OAK_LEAVES	?
143	SPRUCE_LEAVES	?
144	BIRCH_LEAVES	?
145	JUNGLE_LEAVES	?
146	ACACIA_LEAVES	?
147	DARK_OAK_LEAVES	?
148	MANGROVE_LEAVES	?
149	AZALEA_LEAVES	?
150	FLOWERING_AZALEA_LEAVES	?
151	SPONGE	?
152	WET_SPONGE	?
153	GLASS	?
154	TINTED_GLASS	?
155	LAPIS_BLOCK	?
156	SANDSTONE	?
157	CHISELED_SANDSTONE	?
158	CUT_SANDSTONE	?
159	COBWEB	?
160	GRASS	?
161	FERN	?
162	AZALEA	?
163	FLOWERING_AZALEA	?
164	DEAD_BUSH	?
165	SEAGRASS	?

Enum value	Item name	First Minecraft version
166	SEA_PICKLE	?
167	WHITE_WOOL	?
168	ORANGE_WOOL	?
169	MAGENTA_WOOL	?
170	LIGHT_BLUE_WOOL	?
171	YELLOW_WOOL	?
172	LIME_WOOL	?
173	PINK_WOOL	?
174	GRAY_WOOL	?
175	LIGHT_GRAY_WOOL	?
176	CYAN_WOOL	?
177	PURPLE_WOOL	?
178	BLUE_WOOL	?
179	BROWN_WOOL	?
180	GREEN_WOOL	?
181	RED_WOOL	?
182	BLACK_WOOL	?
183	DANDELION	?
184	POPPY	?
185	BLUE_ORCHID	?
186	ALLIUM	?
187	AZURE_BLUET	?
188	RED_TULIP	?
189	ORANGE_TULIP	?
190	WHITE_TULIP	?
191	PINK_TULIP	?
192	OXEYE_DAISY	?
193	CORNFLOWER	?
194	LILY_OF_THE_VALLEY	?
195	WITHER_ROSE	?

Enum value	Item name	First Minecraft version
196	SPORE_BLOSSOM	?
197	BROWN_MUSHROOM	?
198	RED_MUSHROOM	?
199	CRIMSON_FUNGUS	?
200	WARPED_FUNGUS	?
201	CRIMSON_ROOTS	?
202	WARPED_ROOTS	?
203	NETHER_SPROUTS	?
204	WEEPING_VINES	?
205	TWISTING_VINES	?
206	SUGAR_CANE	?
207	KELP	?
208	MOSS_CARPET	?
209	MOSS_BLOCK	?
210	HANGING_ROOTS	?
211	BIG_DRIPLEAF	?
212	SMALL_DRIPLEAF	?
213	BAMBOO	?
214	OAK_SLAB	?
215	SPRUCE_SLAB	?
216	BIRCH_SLAB	?
217	JUNGLE_SLAB	?
218	ACACIA_SLAB	?
219	DARK_OAK_SLAB	?
220	MANGROVE_SLAB	?
221	CRIMSON_SLAB	?
222	WARPED_SLAB	?
223	STONE_SLAB	?
224	SMOOTH_STONE_SLAB	?
225	SANDSTONE_SLAB	?

Enum value	Item name	First Minecraft version
226	CUT_SANDSTONE_SLAB	?
227	PETRIFIED_OAK_SLAB	?
228	COBBLESTONE_SLAB	?
229	BRICK_SLAB	?
230	STONE_BRICK_SLAB	?
231	MUD_BRICK_SLAB	?
232	NETHER_BRICK_SLAB	?
233	QUARTZ_SLAB	?
234	RED_SANDSTONE_SLAB	?
235	CUT_RED_SANDSTONE_SLAB	?
236	PURPUR_SLAB	?
237	PRISMARINE_SLAB	?
238	PRISMARINE_BRICK_SLAB	?
239	DARK_PRISMARINE_SLAB	?
240	SMOOTH_QUARTZ	?
241	SMOOTH_RED_SANDSTONE	?
242	SMOOTH_SANDSTONE	?
243	SMOOTH_STONE	?
244	BRICKS	?
245	BOOKSHELF	?
246	MOSSY_COBBLESTONE	?
247	OBSIDIAN	?
248	TORCH	?
249	END_ROD	?
250	CHORUS_PLANT	?
251	CHORUS_FLOWER	?
252	PURPUR_BLOCK	?
253	PURPUR_PILLAR	?
254	PURPUR_STAIRS	?
255	SPAWNER	?

Enum value	Item name	First Minecraft version
256	CHEST	?
257	CRAFTING_TABLE	?
258	FARMLAND	?
259	FURNACE	?
260	LADDER	?
261	COBBLESTONE_STAIRS	?
262	SNOW	?
263	ICE	?
264	SNOW_BLOCK	?
265	CACTUS	?
266	CLAY	?
267	JUKEBOX	?
268	OAK_FENCE	?
269	SPRUCE_FENCE	?
270	BIRCH_FENCE	?
271	JUNGLE_FENCE	?
272	ACACIA_FENCE	?
273	DARK_OAK_FENCE	?
274	MANGROVE_FENCE	?
275	CRIMSON_FENCE	?
276	WARPED_FENCE	?
277	PUMPKIN	?
278	CARVED_PUMPKIN	?
279	JACK_O_LANTERN	?
280	NETHERRACK	?
281	SOUL_SAND	?
282	SOUL_SOIL	?
283	BASALT	?
284	POLISHED_BASALT	?
285	SMOOTH_BASALT	?

Enum value	Item name	First Minecraft version
286	SOUL_TORCH	?
287	GLOWSTONE	?
288	INFESTED_STONE	?
289	INFESTED_COBBLESTONE	?
290	INFESTED_STONE_BRICKS	?
291	INFESTED_MOSSY_STONE_BRICKS	?
292	INFESTED_CRACKED_STONE_BRICKS	?
293	INFESTED_CHISELED_STONE_BRICKS	?
294	INFESTED_DEEPSLATE	?
295	STONE_BRICKS	?
296	MOSSY_STONE_BRICKS	?
297	CRACKED_STONE_BRICKS	?
298	CHISELED_STONE_BRICKS	?
299	PACKED_MUD	?
300	MUD_BRICKS	?
301	DEEPSLATE_BRICKS	?
302	CRACKED_DEEPSLATE_BRICKS	?
303	DEEPSLATE_TILES	?
304	CRACKED_DEEPSLATE_TILES	?
305	CHISELED_DEEPSLATE	?
306	REINFORCED_DEEPSLATE	?
307	BROWN_MUSHROOM_BLOCK	?
308	RED_MUSHROOM_BLOCK	?
309	MUSHROOM_STEM	?
310	IRON_BARS	?
311	CHAIN	?
312	GLASS_PANE	?
313	MELON	?
314	VINE	?
315	GLOW_LICHEN	?

Enum value	Item name	First Minecraft version
316	BRICK_STAIRS	?
317	STONE_BRICK_STAIRS	?
318	MUD_BRICK_STAIRS	?
319	MYCELIUM	?
320	LILY_PAD	?
321	NETHER_BRICKS	?
322	CRACKED_NETHER_BRICKS	?
323	CHISELED_NETHER_BRICKS	?
324	NETHER_BRICK_FENCE	?
325	NETHER_BRICK_STAIRS	?
326	SCULK	?
327	SCULK_VEIN	?
328	SCULK_CATALYST	?
329	SCULK_SHRIEKER	?
330	ENCHANTING_TABLE	?
331	END_PORTAL_FRAME	?
332	END_STONE	?
333	END_STONE_BRICKS	?
334	DRAGON_EGG	?
335	SANDSTONE_STAIRS	?
336	ENDER_CHEST	?
337	EMERALD_BLOCK	?
338	OAK_STAIRS	?
339	SPRUCE_STAIRS	?
340	BIRCH_STAIRS	?
341	JUNGLE_STAIRS	?
342	ACACIA_STAIRS	?
343	DARK_OAK_STAIRS	?
344	MANGROVE_STAIRS	?
345	CRIMSON_STAIRS	?

Enum value	Item name	First Minecraft version
346	WARPED_STAIRS	?
347	COMMAND_BLOCK	?
348	BEACON	?
349	COBBLESTONE_WALL	?
350	MOSSY_COBBLESTONE_WALL	?
351	BRICK_WALL	?
352	PRISMARINE_WALL	?
353	RED_SANDSTONE_WALL	?
354	MOSSY_STONE_BRICK_WALL	?
355	GRANITE_WALL	?
356	STONE_BRICK_WALL	?
357	MUD_BRICK_WALL	?
358	NETHER_BRICK_WALL	?
359	ANDESITE_WALL	?
360	RED_NETHER_BRICK_WALL	?
361	SANDSTONE_WALL	?
362	END_STONE_BRICK_WALL	?
363	DIORITE_WALL	?
364	BLACKSTONE_WALL	?
365	POLISHED_BLACKSTONE_WALL	?
366	POLISHED_BLACKSTONE_BRICK_WALL	?
367	COBBLED_DEEPSLATE_WALL	?
368	POLISHED_DEEPSLATE_WALL	?
369	DEEPSLATE_BRICK_WALL	?
370	DEEPSLATE_TILE_WALL	?
371	ANVIL	?
372	CHIPPED_ANVIL	?
373	DAMAGED_ANVIL	?
374	CHISELED_QUARTZ_BLOCK	?
375	QUARTZ_BLOCK	?

Enum value	Item name	First Minecraft version
376	QUARTZ_BRICKS	?
377	QUARTZ_PILLAR	?
378	QUARTZ_STAIRS	?
379	WHITE_TERRACOTTA	?
380	ORANGE_TERRACOTTA	?
381	MAGENTA_TERRACOTTA	?
382	LIGHT_BLUE_TERRACOTTA	?
383	YELLOW_TERRACOTTA	?
384	LIME_TERRACOTTA	?
385	PINK_TERRACOTTA	?
386	GRAY_TERRACOTTA	?
387	LIGHT_GRAY_TERRACOTTA	?
388	CYAN_TERRACOTTA	?
389	PURPLE_TERRACOTTA	?
390	BLUE_TERRACOTTA	?
391	BROWN_TERRACOTTA	?
392	GREEN_TERRACOTTA	?
393	RED_TERRACOTTA	?
394	BLACK_TERRACOTTA	?
395	BARRIER	?
396	LIGHT	?
397	HAY_BLOCK	?
398	WHITE_CARPET	?
399	ORANGE_CARPET	?
400	MAGENTA_CARPET	?
401	LIGHT_BLUE_CARPET	?
402	YELLOW_CARPET	?
403	LIME_CARPET	?
404	PINK_CARPET	?
405	GRAY_CARPET	?

Enum value	Item name	First Minecraft version
406	LIGHT_GRAY_CARPET	?
407	CYAN_CARPET	?
408	PURPLE_CARPET	?
409	BLUE_CARPET	?
410	BROWN_CARPET	?
411	GREEN_CARPET	?
412	RED_CARPET	?
413	BLACK_CARPET	?
414	TERRACOTTA	?
415	PACKED_ICE	?
416	DIRT_PATH	?
417	SUNFLOWER	?
418	LILAC	?
419	ROSE_BUSH	?
420	PEONY	?
421	TALL_GRASS	?
422	LARGE_FERN	?
423	WHITE_STAINED_GLASS	?
424	ORANGE_STAINED_GLASS	?
425	MAGENTA_STAINED_GLASS	?
426	LIGHT_BLUE_STAINED_GLASS	?
427	YELLOW_STAINED_GLASS	?
428	LIME_STAINED_GLASS	?
429	PINK_STAINED_GLASS	?
430	GRAY_STAINED_GLASS	?
431	LIGHT_GRAY_STAINED_GLASS	?
432	CYAN_STAINED_GLASS	?
433	PURPLE_STAINED_GLASS	?
434	BLUE_STAINED_GLASS	?
435	BROWN_STAINED_GLASS	?

Enum value	Item name	First Minecraft version
436	GREEN_STAINED_GLASS	?
437	RED_STAINED_GLASS	?
438	BLACK_STAINED_GLASS	?
439	WHITE_STAINED_GLASS_PANE	?
440	ORANGE_STAINED_GLASS_PANE	?
441	MAGENTA_STAINED_GLASS_PANE	?
442	LIGHT_BLUE_STAINED_GLASS_PANE	?
443	YELLOW_STAINED_GLASS_PANE	?
444	LIME_STAINED_GLASS_PANE	?
445	PINK_STAINED_GLASS_PANE	?
446	GRAY_STAINED_GLASS_PANE	?
447	LIGHT_GRAY_STAINED_GLASS_PANE	?
448	CYAN_STAINED_GLASS_PANE	?
449	PURPLE_STAINED_GLASS_PANE	?
450	BLUE_STAINED_GLASS_PANE	?
451	BROWN_STAINED_GLASS_PANE	?
452	GREEN_STAINED_GLASS_PANE	?
453	RED_STAINED_GLASS_PANE	?
454	BLACK_STAINED_GLASS_PANE	?
455	PRISMARINE	?
456	PRISMARINE_BRICKS	?
457	DARK_PRISMARINE	?
458	PRISMARINE_STAIRS	?
459	PRISMARINE_BRICK_STAIRS	?
460	DARK_PRISMARINE_STAIRS	?
461	SEA_LANTERN	?
462	RED_SANDSTONE	?
463	CHISELED_RED_SANDSTONE	?
464	CUT_RED_SANDSTONE	?
465	RED_SANDSTONE_STAIRS	?

Enum value	Item name	First Minecraft version
466	REPEATING_COMMAND_BLOCK	?
467	CHAIN_COMMAND_BLOCK	?
468	MAGMA_BLOCK	?
469	NETHER_WART_BLOCK	?
470	WARPED_WART_BLOCK	?
471	RED_NETHER_BRICKS	?
472	BONE_BLOCK	?
473	STRUCTURE_VOID	?
474	SHULKER_BOX	?
475	WHITE_SHULKER_BOX	?
476	ORANGE_SHULKER_BOX	?
477	MAGENTA_SHULKER_BOX	?
478	LIGHT_BLUE_SHULKER_BOX	?
479	YELLOW_SHULKER_BOX	?
480	LIME_SHULKER_BOX	?
481	PINK_SHULKER_BOX	?
482	GRAY_SHULKER_BOX	?
483	LIGHT_GRAY_SHULKER_BOX	?
484	CYAN_SHULKER_BOX	?
485	PURPLE_SHULKER_BOX	?
486	BLUE_SHULKER_BOX	?
487	BROWN_SHULKER_BOX	?
488	GREEN_SHULKER_BOX	?
489	RED_SHULKER_BOX	?
490	BLACK_SHULKER_BOX	?
491	WHITE_GLAZED_TERRACOTTA	?
492	ORANGE_GLAZED_TERRACOTTA	?
493	MAGENTA_GLAZED_TERRACOTTA	?
494	LIGHT_BLUE_GLAZED_TERRACOTTA	?
495	YELLOW_GLAZED_TERRACOTTA	?

Enum value	Item name	First Minecraft version
496	LIME_GLAZED_TERRACOTTA	?
497	PINK_GLAZED_TERRACOTTA	?
498	GRAY_GLAZED_TERRACOTTA	?
499	LIGHT_GRAY_GLAZED_TERRACOTTA	?
500	CYAN_GLAZED_TERRACOTTA	?
501	PURPLE_GLAZED_TERRACOTTA	?
502	BLUE_GLAZED_TERRACOTTA	?
503	BROWN_GLAZED_TERRACOTTA	?
504	GREEN_GLAZED_TERRACOTTA	?
505	RED_GLAZED_TERRACOTTA	?
506	BLACK_GLAZED_TERRACOTTA	?
507	WHITE_CONCRETE	?
508	ORANGE_CONCRETE	?
509	MAGENTA_CONCRETE	?
510	LIGHT_BLUE_CONCRETE	?
511	YELLOW_CONCRETE	?
512	LIME_CONCRETE	?
513	PINK_CONCRETE	?
514	GRAY_CONCRETE	?
515	LIGHT_GRAY_CONCRETE	?
516	CYAN_CONCRETE	?
517	PURPLE_CONCRETE	?
518	BLUE_CONCRETE	?
519	BROWN_CONCRETE	?
520	GREEN_CONCRETE	?
521	RED_CONCRETE	?
522	BLACK_CONCRETE	?
523	WHITE_CONCRETE_POWDER	?
524	ORANGE_CONCRETE_POWDER	?
525	MAGENTA_CONCRETE_POWDER	?

Enum value	Item name	First Minecraft version
526	LIGHT_BLUE_CONCRETE_POWDER	?
527	YELLOW_CONCRETE_POWDER	?
528	LIME_CONCRETE_POWDER	?
529	PINK_CONCRETE_POWDER	?
530	GRAY_CONCRETE_POWDER	?
531	LIGHT_GRAY_CONCRETE_POWDER	?
532	CYAN_CONCRETE_POWDER	?
533	PURPLE_CONCRETE_POWDER	?
534	BLUE_CONCRETE_POWDER	?
535	BROWN_CONCRETE_POWDER	?
536	GREEN_CONCRETE_POWDER	?
537	RED_CONCRETE_POWDER	?
538	BLACK_CONCRETE_POWDER	?
539	TURTLE_EGG	?
540	DEAD_TUBE_CORAL_BLOCK	?
541	DEAD_BRAIN_CORAL_BLOCK	?
542	DEAD_BUBBLE_CORAL_BLOCK	?
543	DEAD_FIRE_CORAL_BLOCK	?
544	DEAD_HORN_CORAL_BLOCK	?
545	TUBE_CORAL_BLOCK	?
546	BRAIN_CORAL_BLOCK	?
547	BUBBLE_CORAL_BLOCK	?
548	FIRE_CORAL_BLOCK	?
549	HORN_CORAL_BLOCK	?
550	TUBE_CORAL	?
551	BRAIN_CORAL	?
552	BUBBLE_CORAL	?
553	FIRE_CORAL	?
554	HORN_CORAL	?
555	DEAD_BRAIN_CORAL	?

Enum value	Item name	First Minecraft version
556	DEAD_BUBBLE_CORAL	?
557	DEAD_FIRE_CORAL	?
558	DEAD_HORN_CORAL	?
559	DEAD_TUBE_CORAL	?
560	TUBE_CORAL_FAN	?
561	BRAIN_CORAL_FAN	?
562	BUBBLE_CORAL_FAN	?
563	FIRE_CORAL_FAN	?
564	HORN_CORAL_FAN	?
565	DEAD_TUBE_CORAL_FAN	?
566	DEAD_BRAIN_CORAL_FAN	?
567	DEAD_BUBBLE_CORAL_FAN	?
568	DEAD_FIRE_CORAL_FAN	?
569	DEAD_HORN_CORAL_FAN	?
570	BLUE_ICE	?
571	CONDUIT	?
572	POLISHED_GRANITE_STAIRS	?
573	SMOOTH_RED_SANDSTONE_STAIRS	?
574	MOSSY_STONE_BRICK_STAIRS	?
575	POLISHED_DIORITE_STAIRS	?
576	MOSSY_COBBLESTONE_STAIRS	?
577	END_STONE_BRICK_STAIRS	?
578	STONE_STAIRS	?
579	SMOOTH_SANDSTONE_STAIRS	?
580	SMOOTH_QUARTZ_STAIRS	?
581	GRANITE_STAIRS	?
582	ANDESITE_STAIRS	?
583	RED_NETHER_BRICK_STAIRS	?
584	POLISHED_ANDESITE_STAIRS	?
585	DIORITE_STAIRS	?

Enum value	Item name	First Minecraft version
586	COBBLED_DEEPSLATE_STAIRS	?
587	POLISHED_DEEPSLATE_STAIRS	?
588	DEEPSLATE_BRICK_STAIRS	?
589	DEEPSLATE_TILE_STAIRS	?
590	POLISHED_GRANITE_SLAB	?
591	SMOOTH_RED_SANDSTONE_SLAB	?
592	MOSSY_STONE_BRICK_SLAB	?
593	POLISHED_DIORITE_SLAB	?
594	MOSSY_COBBLESTONE_SLAB	?
595	END_STONE_BRICK_SLAB	?
596	SMOOTH_SANDSTONE_SLAB	?
597	SMOOTH_QUARTZ_SLAB	?
598	GRANITE_SLAB	?
599	ANDESITE_SLAB	?
600	RED_NETHER_BRICK_SLAB	?
601	POLISHED_ANDESITE_SLAB	?
602	DIORITE_SLAB	?
603	COBBLED_DEEPSLATE_SLAB	?
604	POLISHED_DEEPSLATE_SLAB	?
605	DEEPSLATE_BRICK_SLAB	?
606	DEEPSLATE_TILE_SLAB	?
607	SCAFFOLDING	?
608	REDSTONE	?
609	REDSTONE_TORCH	?
610	REDSTONE_BLOCK	?
611	REPEATER	?
612	COMPARATOR	?
613	PISTON	?
614	STICKY_PISTON	?
615	SLIME_BLOCK	?

Enum value	Item name	First Minecraft version
616	HONEY_BLOCK	?
617	OBSERVER	?
618	HOPPER	?
619	DISPENSER	?
620	DROPPER	?
621	LECTERN	?
622	TARGET	?
623	LEVER	?
624	LIGHTNING_ROD	?
625	DAYLIGHT_DETECTOR	?
626	SCULK_SENSOR	?
627	TRIPWIRE_HOOK	?
628	TRAPPED_CHEST	?
629	TNT	?
630	REDSTONE_LAMP	?
631	NOTE_BLOCK	?
632	STONE_BUTTON	?
633	POLISHED_BLACKSTONE_BUTTON	?
634	OAK_BUTTON	?
635	SPRUCE_BUTTON	?
636	BIRCH_BUTTON	?
637	JUNGLE_BUTTON	?
638	ACACIA_BUTTON	?
639	DARK_OAK_BUTTON	?
640	MANGROVE_BUTTON	?
641	CRIMSON_BUTTON	?
642	WARPED_BUTTON	?
643	STONE_PRESSURE_PLATE	?
644	POLISHED_BLACKSTONE_PRESSURE_PLATE	?
645	LIGHT_WEIGHTED_PRESSURE_PLATE	?

Enum value	Item name	First Minecraft version
646	HEAVY_WEIGHTED_PRESSURE_PLATE	?
647	OAK_PRESSURE_PLATE	?
648	SPRUCE_PRESSURE_PLATE	?
649	BIRCH_PRESSURE_PLATE	?
650	JUNGLE_PRESSURE_PLATE	?
651	ACACIA_PRESSURE_PLATE	?
652	DARK_OAK_PRESSURE_PLATE	?
653	MANGROVE_PRESSURE_PLATE	?
654	CRIMSON_PRESSURE_PLATE	?
655	WARPED_PRESSURE_PLATE	?
656	IRON_DOOR	?
657	OAK_DOOR	?
658	SPRUCE_DOOR	?
659	BIRCH_DOOR	?
660	JUNGLE_DOOR	?
661	ACACIA_DOOR	?
662	DARK_OAK_DOOR	?
663	MANGROVE_DOOR	?
664	CRIMSON_DOOR	?
665	WARPED_DOOR	?
666	IRON_TRAPDOOR	?
667	OAK_TRAPDOOR	?
668	SPRUCE_TRAPDOOR	?
669	BIRCH_TRAPDOOR	?
670	JUNGLE_TRAPDOOR	?
671	ACACIA_TRAPDOOR	?
672	DARK_OAK_TRAPDOOR	?
673	MANGROVE_TRAPDOOR	?
674	CRIMSON_TRAPDOOR	?
675	WARPED_TRAPDOOR	?

Enum value	Item name	First Minecraft version
676	OAK_FENCE_GATE	?
677	SPRUCE_FENCE_GATE	?
678	BIRCH_FENCE_GATE	?
679	JUNGLE_FENCE_GATE	?
680	ACACIA_FENCE_GATE	?
681	DARK_OAK_FENCE_GATE	?
682	MANGROVE_FENCE_GATE	?
683	CRIMSON_FENCE_GATE	?
684	WARPED_FENCE_GATE	?
685	POWERED_RAIL	?
686	DETECTOR_RAIL	?
687	RAIL	?
688	ACTIVATOR_RAIL	?
689	SADDLE	?
690	MINECART	?
691	CHEST_MINECART	?
692	FURNACE_MINECART	?
693	TNT_MINECART	?
694	HOPPER_MINECART	?
695	CARROT_ON_A_STICK	?
696	WARPED_FUNGUS_ON_A_STICK	?
697	ELYTRA	?
698	OAK_BOAT	?
699	OAK_CHEST_BOAT	?
700	SPRUCE_BOAT	?
701	SPRUCE_CHEST_BOAT	?
702	BIRCH_BOAT	?
703	BIRCH_CHEST_BOAT	?
704	JUNGLE_BOAT	?
705	JUNGLE_CHEST_BOAT	?

Enum value	Item name	First Minecraft version
706	ACACIA_BOAT	?
707	ACACIA_CHEST_BOAT	?
708	DARK_OAK_BOAT	?
709	DARK_OAK_CHEST_BOAT	?
710	MANGROVE_BOAT	?
711	MANGROVE_CHEST_BOAT	?
712	STRUCTURE_BLOCK	?
713	JIGSAW	?
714	TURTLE_HELMET	?
715	SCUTE	?
716	FLINT_AND_STEEL	?
717	APPLE	?
718	BOW	?
719	ARROW	?
720	COAL	?
721	CHARCOAL	?
722	DIAMOND	?
723	EMERALD	?
724	LAPIS_LAZULI	?
725	QUARTZ	?
726	AMETHYST_SHARD	?
727	RAW_IRON	?
728	IRON_INGOT	?
729	RAW_COPPER	?
730	COPPER_INGOT	?
731	RAW_GOLD	?
732	GOLD_INGOT	?
733	NETHERITE_INGOT	?
734	NETHERITE_SCRAP	?
735	WOODEN_SWORD	?

Enum value	Item name	First Minecraft version
736	WOODEN_SHOVEL	?
737	WOODEN_PICKAXE	?
738	WOODEN_AXE	?
739	WOODEN_HOE	?
740	STONE_SWORD	?
741	STONE_SHOVEL	?
742	STONE_PICKAXE	?
743	STONE_AXE	?
744	STONE_HOE	?
745	GOLDEN_SWORD	?
746	GOLDEN_SHOVEL	?
747	GOLDEN_PICKAXE	?
748	GOLDEN_AXE	?
749	GOLDEN_HOE	?
750	IRON_SWORD	?
751	IRON_SHOVEL	?
752	IRON_PICKAXE	?
753	IRON_AXE	?
754	IRON_HOE	?
755	DIAMOND_SWORD	?
756	DIAMOND_SHOVEL	?
757	DIAMOND_PICKAXE	?
758	DIAMOND_AXE	?
759	DIAMOND_HOE	?
760	NETHERITE_SWORD	?
761	NETHERITE_SHOVEL	?
762	NETHERITE_PICKAXE	?
763	NETHERITE_AXE	?
764	NETHERITE_HOE	?
765	STICK	?

Enum value	Item name	First Minecraft version
766	BOWL	?
767	MUSHROOM_STEW	?
768	STRING	?
769	FEATHER	?
770	GUNPOWDER	?
771	WHEAT_SEEDS	?
772	WHEAT	?
773	BREAD	?
774	LEATHER_HELMET	?
775	LEATHER_CHESTPLATE	?
776	LEATHER_LEGGINGS	?
777	LEATHER_BOOTS	?
778	CHAINMAIL_HELMET	?
779	CHAINMAIL_CHESTPLATE	?
780	CHAINMAIL_LEGGINGS	?
781	CHAINMAIL_BOOTS	?
782	IRON_HELMET	?
783	IRON_CHESTPLATE	?
784	IRON_LEGGINGS	?
785	IRON_BOOTS	?
786	DIAMOND_HELMET	?
787	DIAMOND_CHESTPLATE	?
788	DIAMOND_LEGGINGS	?
789	DIAMOND_BOOTS	?
790	GOLDEN_HELMET	?
791	GOLDEN_CHESTPLATE	?
792	GOLDEN_LEGGINGS	?
793	GOLDEN_BOOTS	?
794	NETHERITE_HELMET	?
795	NETHERITE_CHESTPLATE	?

Enum value	Item name	First Minecraft version
796	NETHERITE_LEGGINGS	?
797	NETHERITE_BOOTS	?
798	FLINT	?
799	PORKCHOP	?
800	COOKED_PORKCHOP	?
801	PAINTING	?
802	GOLDEN_APPLE	?
803	ENCHANTED_GOLDEN_APPLE	?
804	OAK_SIGN	?
805	SPRUCE_SIGN	?
806	BIRCH_SIGN	?
807	JUNGLE_SIGN	?
808	ACACIA_SIGN	?
809	DARK_OAK_SIGN	?
810	MANGROVE_SIGN	?
811	CRIMSON_SIGN	?
812	WARPED_SIGN	?
813	BUCKET	?
814	WATER_BUCKET	?
815	LAVA_BUCKET	?
816	POWDER_SNOW_BUCKET	?
817	SNOWBALL	?
818	LEATHER	?
819	MILK_BUCKET	?
820	PUFFERFISH_BUCKET	?
821	SALMON_BUCKET	?
822	COD_BUCKET	?
823	TROPICAL_FISH_BUCKET	?
824	AXOLOTL_BUCKET	?
825	TADPOLE_BUCKET	?

Enum value	Item name	First Minecraft version
826	BRICK	?
827	CLAY_BALL	?
828	DRIED_KELP_BLOCK	?
829	PAPER	?
830	BOOK	?
831	SLIME_BALL	?
832	EGG	?
833	COMPASS	?
834	RECOVERY_COMPASS	?
835	BUNDLE	?
836	FISHING_ROD	?
837	CLOCK	?
838	SPYGLASS	?
839	GLOWSTONE_DUST	?
840	COD	?
841	SALMON	?
842	TROPICAL_FISH	?
843	PUFFERFISH	?
844	COOKED_COD	?
845	COOKED_SALMON	?
846	INK_SAC	?
847	GLOW_INK_SAC	?
848	COCOA_BEANS	?
849	WHITE_DYE	?
850	ORANGE_DYE	?
851	MAGENTA_DYE	?
852	LIGHT_BLUE_DYE	?
853	YELLOW_DYE	?
854	LIME_DYE	?
855	PINK_DYE	?

Enum value	Item name	First Minecraft version
856	GRAY_DYE	?
857	LIGHT_GRAY_DYE	?
858	CYAN_DYE	?
859	PURPLE_DYE	?
860	BLUE_DYE	?
861	BROWN_DYE	?
862	GREEN_DYE	?
863	RED_DYE	?
864	BLACK_DYE	?
865	BONE_MEAL	?
866	BONE	?
867	SUGAR	?
868	CAKE	?
869	WHITE_BED	?
870	ORANGE_BED	?
871	MAGENTA_BED	?
872	LIGHT_BLUE_BED	?
873	YELLOW_BED	?
874	LIME_BED	?
875	PINK_BED	?
876	GRAY_BED	?
877	LIGHT_GRAY_BED	?
878	CYAN_BED	?
879	PURPLE_BED	?
880	BLUE_BED	?
881	BROWN_BED	?
882	GREEN_BED	?
883	RED_BED	?
884	BLACK_BED	?
885	COOKIE	?

Enum value	Item name	First Minecraft version
886	FILLED_MAP	?
887	SHEARS	?
888	MELON_SLICE	?
889	DRIED_KELP	?
890	PUMPKIN_SEEDS	?
891	MELON_SEEDS	?
892	BEEF	?
893	COOKED_BEEF	?
894	CHICKEN	?
895	COOKED_CHICKEN	?
896	ROTTEN_FLESH	?
897	ENDER_PEARL	?
898	BLAZE_ROD	?
899	GHAST_TEAR	?
900	GOLD_NUGGET	?
901	NETHER_WART	?
902	POTION	?
903	GLASS_BOTTLE	?
904	SPIDER_EYE	?
905	FERMENTED_SPIDER_EYE	?
906	BLAZE_POWDER	?
907	MAGMA_CREAM	?
908	BREWING_STAND	?
909	CAULDRON	?
910	ENDER_EYE	?
911	GLISTERING_MELON_SLICE	?
912	ALLAY_SPAWN_EGG	?
913	AXOLOTL_SPAWN_EGG	?
914	BAT_SPAWN_EGG	?
915	BEE_SPAWN_EGG	?

Enum value	Item name	First Minecraft version
916	BLAZE_SPAWN_EGG	?
917	CAT_SPAWN_EGG	?
918	CAVE_SPIDER_SPAWN_EGG	?
919	CHICKEN_SPAWN_EGG	?
920	COD_SPAWN_EGG	?
921	COW_SPAWN_EGG	?
922	CREEPER_SPAWN_EGG	?
923	DOLPHIN_SPAWN_EGG	?
924	DONKEY_SPAWN_EGG	?
925	DROWNED_SPAWN_EGG	?
926	ELDER_GUARDIAN_SPAWN_EGG	?
927	ENDERMAN_SPAWN_EGG	?
928	ENDERMITE_SPAWN_EGG	?
929	EVOKER_SPAWN_EGG	?
930	FOX_SPAWN_EGG	?
931	FROG_SPAWN_EGG	?
932	GHAST_SPAWN_EGG	?
933	GLOW_SQUID_SPAWN_EGG	?
934	GOAT_SPAWN_EGG	?
935	GUARDIAN_SPAWN_EGG	?
936	HOGLIN_SPAWN_EGG	?
937	HORSE_SPAWN_EGG	?
938	HUSK_SPAWN_EGG	?
939	LLAMA_SPAWN_EGG	?
940	MAGMA_CUBE_SPAWN_EGG	?
941	MOOSHROOM_SPAWN_EGG	?
942	MULE_SPAWN_EGG	?
943	OCELOT_SPAWN_EGG	?
944	PANDA_SPAWN_EGG	?
945	PARROT_SPAWN_EGG	?

Enum value	Item name	First Minecraft version
946	PHANTOM_SPAWN_EGG	?
947	PIG_SPAWN_EGG	?
948	PIGLIN_SPAWN_EGG	?
949	PIGLIN_BRUTE_SPAWN_EGG	?
950	PILLAGER_SPAWN_EGG	?
951	POLAR_BEAR_SPAWN_EGG	?
952	PUFFERFISH_SPAWN_EGG	?
953	RABBIT_SPAWN_EGG	?
954	RAVAGER_SPAWN_EGG	?
955	SALMON_SPAWN_EGG	?
956	SHEEP_SPAWN_EGG	?
957	SHULKER_SPAWN_EGG	?
958	SILVERFISH_SPAWN_EGG	?
959	SKELETON_SPAWN_EGG	?
960	SKELETON_HORSE_SPAWN_EGG	?
961	SLIME_SPAWN_EGG	?
962	SPIDER_SPAWN_EGG	?
963	SQUID_SPAWN_EGG	?
964	STRAY_SPAWN_EGG	?
965	STRIDER_SPAWN_EGG	?
966	TADPOLE_SPAWN_EGG	?
967	TRADER_LLAMA_SPAWN_EGG	?
968	TROPICAL_FISH_SPAWN_EGG	?
969	TURTLE_SPAWN_EGG	?
970	VEX_SPAWN_EGG	?
971	VILLAGER_SPAWN_EGG	?
972	VINDICATOR_SPAWN_EGG	?
973	WANDERING_TRADER_SPAWN_EGG	?
974	WARDEN_SPAWN_EGG	?
975	WITCH_SPAWN_EGG	?

Enum value	Item name	First Minecraft version
976	WITHER_SKELETON_SPAWN_EGG	?
977	WOLF_SPAWN_EGG	?
978	ZOGLIN_SPAWN_EGG	?
979	ZOMBIE_SPAWN_EGG	?
980	ZOMBIE_HORSE_SPAWN_EGG	?
981	ZOMBIE_VILLAGER_SPAWN_EGG	?
982	ZOMBIFIED_PIGLIN_SPAWN_EGG	?
983	EXPERIENCE_BOTTLE	?
984	FIRE_CHARGE	?
985	WRITABLE_BOOK	?
986	WRITTEN_BOOK	?
987	ITEM_FRAME	?
988	GLOW_ITEM_FRAME	?
989	FLOWER_POT	?
990	CARROT	?
991	POTATO	?
992	BAKED_POTATO	?
993	POISONOUS_POTATO	?
994	MAP	?
995	GOLDEN_CARROT	?
996	SKELETON_SKULL	?
997	WITHER_SKELETON_SKULL	?
998	PLAYER_HEAD	?
999	ZOMBIE_HEAD	?
1000	CREEPER_HEAD	?
1001	DRAGON_HEAD	?
1002	NETHER_STAR	?
1003	PUMPKIN_PIE	?
1004	FIREWORK_ROCKET	?
1005	FIREWORK_STAR	?

Enum value	Item name	First Minecraft version
1006	ENCHANTED_BOOK	?
1007	NETHER_BRICK	?
1008	PRISMARINE_SHARD	?
1009	PRISMARINE_CRYSTALS	?
1010	RABBIT	?
1011	COOKED_RABBIT	?
1012	RABBIT_STEW	?
1013	RABBIT_FOOT	?
1014	RABBIT_HIDE	?
1015	ARMOR_STAND	?
1016	IRON_HORSE_ARMOR	?
1017	GOLDEN_HORSE_ARMOR	?
1018	DIAMOND_HORSE_ARMOR	?
1019	LEATHER_HORSE_ARMOR	?
1020	LEAD	?
1021	NAME_TAG	?
1022	COMMAND_BLOCK_MINECART	?
1023	MUTTON	?
1024	COOKED_MUTTON	?
1025	WHITE_BANNER	?
1026	ORANGE_BANNER	?
1027	MAGENTA_BANNER	?
1028	LIGHT_BLUE_BANNER	?
1029	YELLOW_BANNER	?
1030	LIME_BANNER	?
1031	PINK_BANNER	?
1032	GRAY_BANNER	?
1033	LIGHT_GRAY_BANNER	?
1034	CYAN_BANNER	?
1035	PURPLE_BANNER	?

Enum value	Item name	First Minecraft version
1036	BLUE_BANNER	?
1037	BROWN_BANNER	?
1038	GREEN_BANNER	?
1039	RED_BANNER	?
1040	BLACK_BANNER	?
1041	END_CRYSTAL	?
1042	CHORUS_FRUIT	?
1043	POPPED_CHORUS_FRUIT	?
1044	BEETROOT	?
1045	BEETROOT_SEEDS	?
1046	BEETROOT_SOUP	?
1047	DRAGON_BREATH	?
1048	SPLASH_POTION	?
1049	SPECTRAL_ARROW	?
1050	TIPPED_ARROW	?
1051	LINGERING_POTION	?
1052	SHIELD	?
1053	TOTEM_OF_UNDYING	?
1054	SHULKER_SHELL	?
1055	IRON_NUGGET	?
1056	KNOWLEDGE_BOOK	?
1057	DEBUG_STICK	?
1058	MUSIC_DISC_13	?
, 1059	MUSIC_DISC_CAT	?
1060	MUSIC_DISC_BLOCKS	?
1061	MUSIC_DISC_CHIRP	?
1062	MUSIC_DISC_FAR	?
1063	MUSIC_DISC_MALL	?
1064	MUSIC_DISC_MELLOHI	?
1065	MUSIC_DISC_STAL	?

Enum value	Item name	First Minecraft version
1066	MUSIC_DISC_STRAD	?
1067	MUSIC_DISC_WARD	?
1068	MUSIC_DISC_11	?
, 1069	MUSIC_DISC_WAIT	?
1070	MUSIC_DISC_OTHERSIDE	?
1071	MUSIC_DISC_5	?
, 1072	MUSIC_DISC_PIGSTEP	?
1073	DISC_FRAGMENT_5	?
, 1074	TRIDENT	?
1075	PHANTOM_MEMBRANE	?
1076	NAUTILUS_SHELL	?
1077	HEART_OF_THE_SEA	?
1078	CROSSBOW	?
1079	SUSPICIOUS_STEW	?
1080	LOOM	?
1081	FLOWER_BANNER_PATTERN	?
1082	CREEPER_BANNER_PATTERN	?
1083	SKULL_BANNER_PATTERN	?
1084	MOJANG_BANNER_PATTERN	?
1085	GLOBE_BANNER_PATTERN	?
1086	PIGLIN_BANNER_PATTERN	?
1087	GOAT_HORN	?
1088	COMPOSTER	?
1089	BARREL	?
1090	SMOKER	?
1091	BLAST_FURNACE	?
1092	CARTOGRAPHY_TABLE	?
1093	FLETCHING_TABLE	?
1094	GRINDSTONE	?
1095	SMITHING_TABLE	?

Enum value	Item name	First Minecraft version
1096	STONECUTTER	?
1097	BELL	?
1098	LANTERN	?
1099	SOUL_LANTERN	?
1100	SWEET_BERRIES	?
1101	GLOW_BERRIES	?
1102	CAMPFIRE	?
1103	SOUL_CAMPFIRE	?
1104	SHROOMLIGHT	?
1105	HONEYCOMB	?
1106	BEE_NEST	?
1107	BEEHIVE	?
1108	HONEY_BOTTLE	?
1109	HONEYCOMB_BLOCK	?
1110	LODESTONE	?
1111	CRYING_OBSIDIAN	?
1112	BLACKSTONE	?
1113	BLACKSTONE_SLAB	?
1114	BLACKSTONE_STAIRS	?
1115	GILDED_BLACKSTONE	?
1116	POLISHED_BLACKSTONE	?
1117	POLISHED_BLACKSTONE_SLAB	?
1118	POLISHED_BLACKSTONE_STAIRS	?
1119	CHISELED_POLISHED_BLACKSTONE	?
1120	POLISHED_BLACKSTONE_BRICKS	?
1121	POLISHED_BLACKSTONE_BRICK_SLAB	?
1122	POLISHED_BLACKSTONE_BRICK_STAIRS	?
1123	CRACKED_POLISHED_BLACKSTONE_BRICKS	?
1124	RESPAWN_ANCHOR	?
1125	CANDLE	?

Enum value	Item name	First Minecraft version
1126	WHITE_CANDLE	?
1127	ORANGE_CANDLE	?
1128	MAGENTA_CANDLE	?
1129	LIGHT_BLUE_CANDLE	?
1130	YELLOW_CANDLE	?
1131	LIME_CANDLE	?
1132	PINK_CANDLE	?
1133	GRAY_CANDLE	?
1134	LIGHT_GRAY_CANDLE	?
1135	CYAN_CANDLE	?
1136	PURPLE_CANDLE	?
1137	BLUE_CANDLE	?
1138	BROWN_CANDLE	?
1139	GREEN_CANDLE	?
1140	RED_CANDLE	?
1141	BLACK_CANDLE	?
1142	SMALL_AMETHYST_BUD	?
1143	MEDIUM_AMETHYST_BUD	?
1144	LARGE_AMETHYST_BUD	?
1145	AMETHYST_CLUSTER	?
1146	POINTED_DRIPSTONE	?
1147	OCHRE_FROGLIGHT	?
1148	VERDANT_FROGLIGHT	?
1149	PEARLESCENT_FROGLIGHT	?
1150	FROGSPAWN	?
1151	ECHO_SHARD	?

Table 3.3: Item enum

3.4.11 Entity

This section is under construction, as detailed properties cannot be specified.

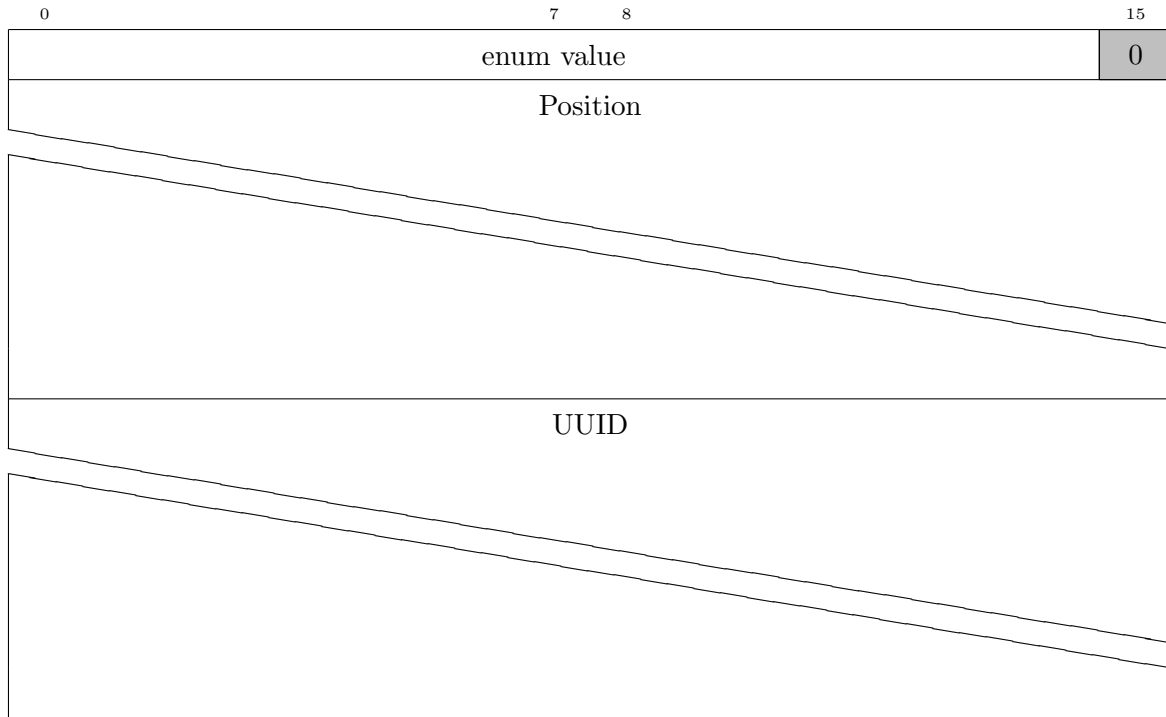One entity is represented by its type. Additionally, it has a unique ID (*UUID*) and a position.

Figure 3.10: Structure of an Entity

Enum value	Entity type	First Minecraft version
0	DROPPED_ITEM	1.8
1	EXPERIENCE_ORB	?
2	AREA_EFFECT_CLOUD	?
3	ELDER_GUARDIAN	?
4	WITHER_SKELETON	?
5	STRAY	?
6	EGG	?
7	LEASH_HITCH	?
8	PAINTING	?

Enum value	Entity type	First Minecraft version
9	ARROW	?
10	SNOWBALL	?
11	FIREBALL	?
12	SMALL_FIREBALL	?
13	ENDER_PEARL	?
14	ENDER_SIGNAL	?
15	SPLASH_POTION	?
16	THROWN_EXP_BOTTLE	?
17	ITEM_FRAME	?
18	WITHER_SKULL	?
19	PRIMED_TNT	?
20	FALLING_BLOCK	?
21	FIREWORK	?
22	HUSK	?
23	SPECTRAL_ARROW	?
24	SHULKER_BULLET	?
25	DRAGON_FIREBALL	?
26	ZOMBIE_VILLAGER	?
27	SKELETON_HORSE	?
28	ZOMBIE_HORSE	?
29	ARMOR_STAND	?
30	DONKEY	?
31	MULE	?
32	EVOKER_FANGS	?
33	EVOKER	?
34	VEX	?
35	VINDICATOR	?
36	ILLUSIONER	?
37	MINECART_COMMAND	?
38	BOAT	?

Enum value	Entity type	First Minecraft version
39	MINECART	?
40	MINECART_CHEST	?
41	MINECART_FURNACE	?
42	MINECART_TNT	?
43	MINECART_HOPPER	?
44	MINECART_MOB_SPAWNER	?
45	CREEPER	?
46	SKELETON	?
47	SPIDER	?
48	GIANT	?
49	ZOMBIE	?
50	SLIME	?
51	GHAST	?
52	ZOMBIFIED_PIGLIN	?
53	ENDERMAN	?
54	CAVE_SPIDER	?
55	SILVERFISH	?
56	BLAZE	?
57	MAGMA_CUBE	?
58	ENDER_DRAGON	?
59	WITHER	?
60	BAT	?
61	WITCH	?
62	ENDERMITE	?
63	GUARDIAN	?
64	SHULKER	?
65	PIG	?
66	SHEEP	?
67	COW	?
68	CHICKEN	?

Enum value	Entity type	First Minecraft version
69	SQUID	?
70	WOLF	?
71	MUSHROOM_COW	?
72	SNOWMAN	?
73	OCELOT	?
74	IRON_GOLEM	?
75	HORSE	?
76	RABBIT	?
77	POLAR_BEAR	?
78	LLAMA	?
79	LLAMA_SPIT	?
80	PARROT	?
81	VILLAGER	?
82	ENDER_CRYSTAL	?
83	TURTLE	?
84	PHANTOM	?
85	TRIDENT	?
86	COD	?
87	SALMON	?
88	PUFFERFISH	?
89	TROPICAL_FISH	?
90	DROWNED	?
91	DOLPHIN	?
92	CAT	?
93	PANDA	?
94	PILLAGER	?
95	RAVAGER	?
96	TRADER_LLAMA	?
97	WANDERING_TRADER	?
98	FOX	?

Enum value	Entity type	First Minecraft version
99	BEE	?
100	HOGLIN	?
101	PIGLIN	?
102	STRIDER	?
103	ZOGLIN	?
104	PIGLIN_BRUTE	?
105	AXOLOTL	?
106	GLOW_ITEM_FRAME	?
107	GLOW_SQUID	?
108	GOAT	?
109	MARKER	?
110	ALLAY	?
111	CHEST_BOAT	?
112	FROG	?
113	TADPOLE	?
114	WARDEN	?
115	FISHING_HOOK	?
116	LIGHTNING	?
117	PLAYER	?

Table 3.4: Entity enum
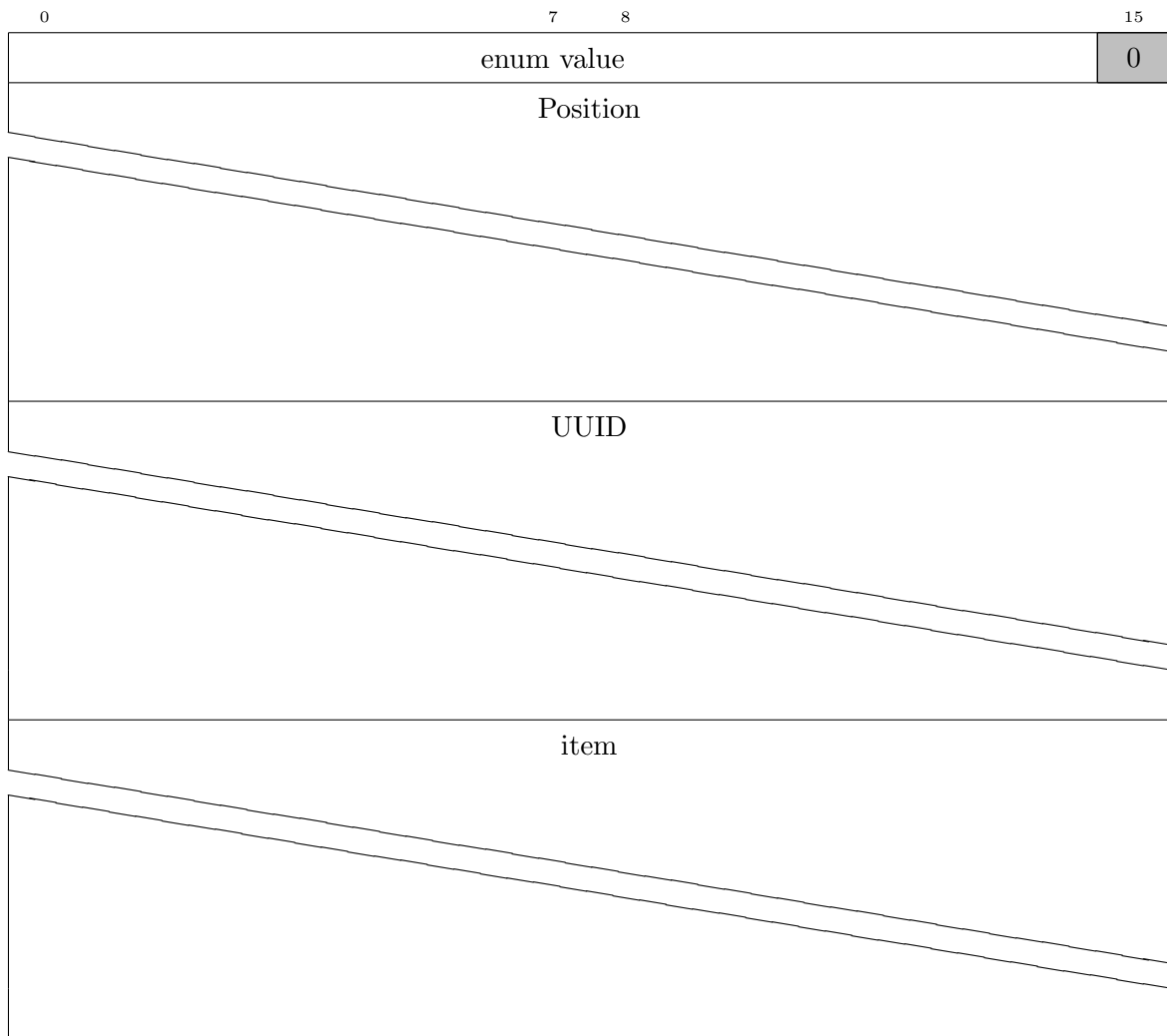
Figure 3.11: Structure of a DroppedItem

3.4.12 Container

This section was only made to get the players' inventories.

0   7   8   15

items
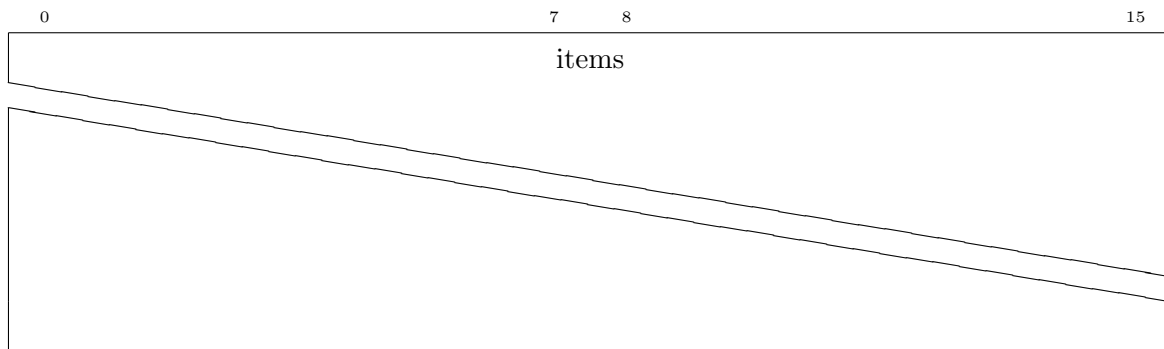
Figure 3.12: Structure of a Container

3.5   Synchronization

As this is a distributed system (multiple programs that needs to be sequential), you'll find one *Synchronization operation* on both the Client and Server. Each program guarantees that the operations will be sequential (operation 1 will be before operation 2, being operation 1 the first sent operation and 2 the second one), but that doesn't applies on two different programs. By calling the *Synchronization operation* and waiting for its reply each time you have to change from one program to another you'll guarantee multi-program sequencing.

4 Servers manager petition

The Servers manager is the responsible for starting the servers with the desired configuration.

All the Servers manager petitions will follow the structure shown in Figure 4.1, Servers manager petition structure.
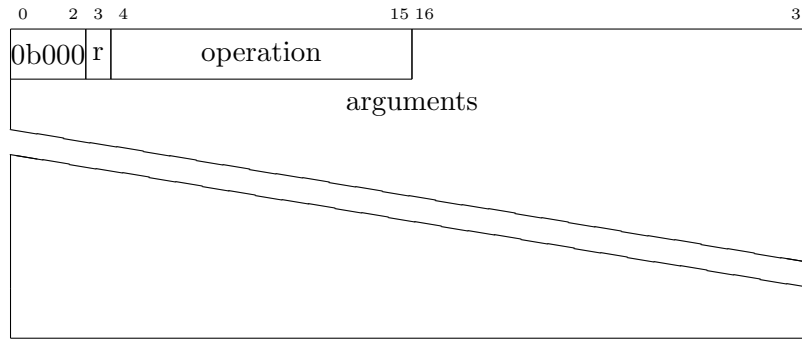
Figure 4.1: Servers manager petition structure

Once all the operations are implemented, here you'll find a table that shows a link to each one of the 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.

4.1 Start server operation

This operation allows WatchWolf Tester to start a server. It provides high customization, as you can specify plugins, worlds and config files (among other parameters).
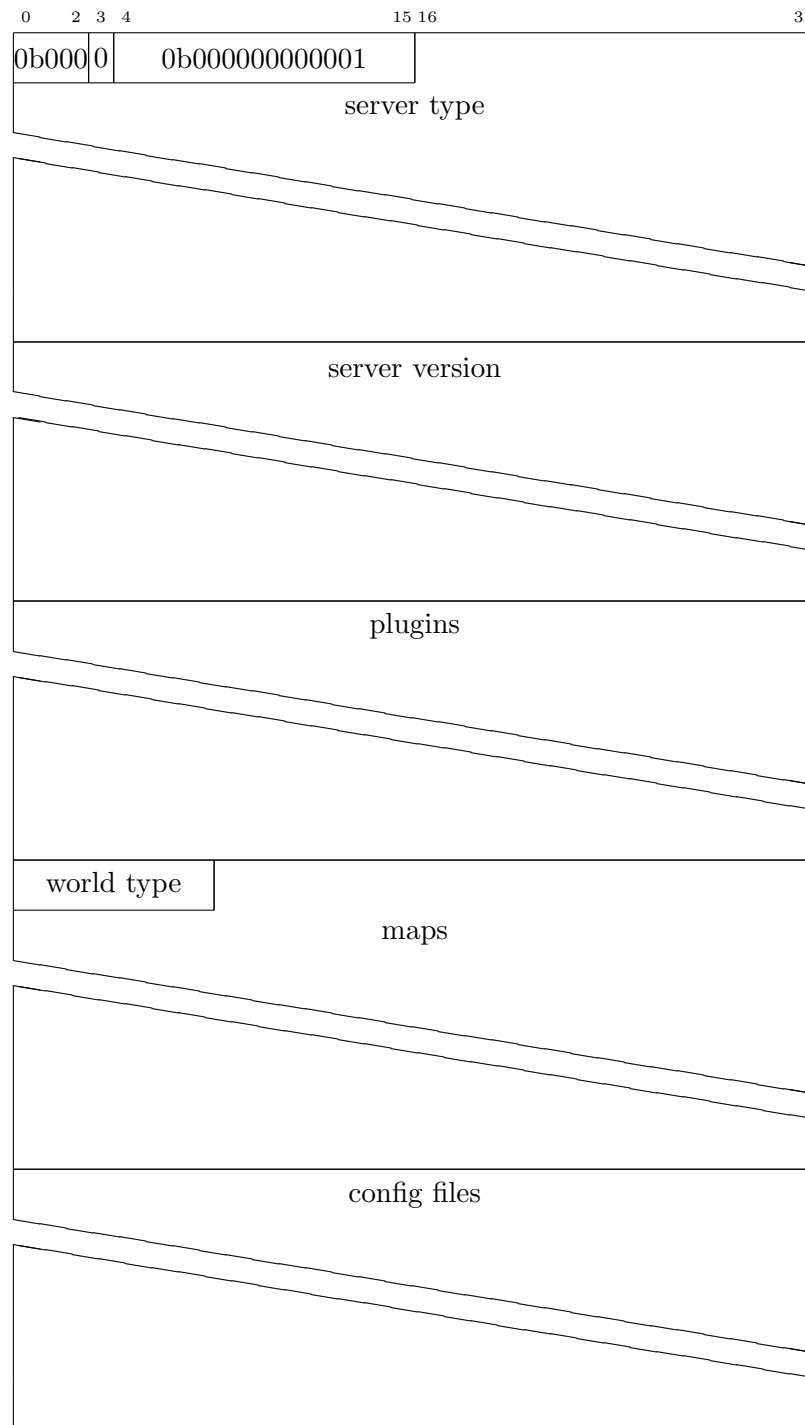
Figure 4.2: Start server petition structure

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 4.3, 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,[7] or there's no free servers of that type), then an empty IP is returned (see Figure 4.4, Start server error response structure).
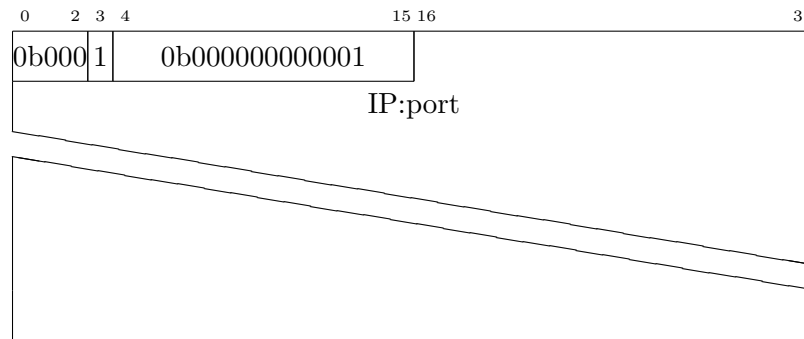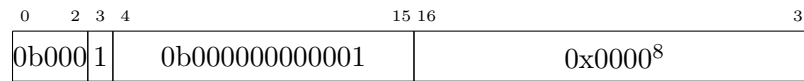

Figure 4.3: Start server response structure

Figure 4.4: Start server error response structure

4.1.1 World type

Minecraft can generate regular worlds, or superflat ones. Use this property to select the desired one.

---

[7]To prevent possible viruses, some servers may have enabled one option that limits which plugins are allowed to be sent. Besides being defined by the API it hasn't been implemented yet, and won't be until WatchWolf offers public servers.

[8]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.

type	Meaning
0	Default
1	Flat

Table 4.1: World types

4.1.2 Maps

Array of maps (worlds; Map[]). To have more information about arrays check the subsection 3.4.5, 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 you can send using this argument your world(s).

Map explanation in more detail is needed.

4.1.3 Plugins

Array of plugins (Plugin[]). To have more information check the subsection 3.4.5, 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[9] dependencies. This allows both security and performance.

   Something to highlight is the fact that, as mentioned in the operation Allows non usual plugins,[10] some ServerManager will only allow plugins that are already in the machine.

---

[9]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.

[10]Reference to the operation is needed here, but not yet implemented. It won't be until WatchWolf offers public servers.

As can be seen in the Figure 4.5, 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.
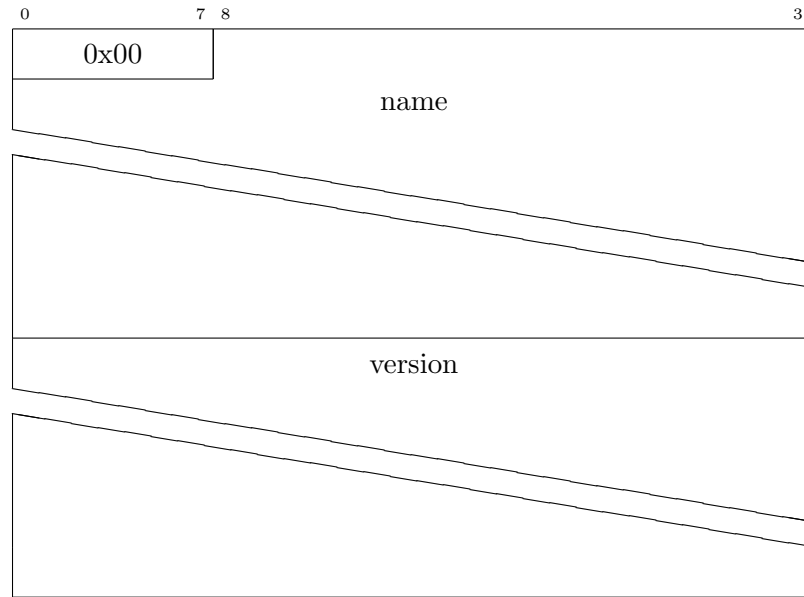

Figure 4.5: Usual plugin structure

2. Uploaded plugins

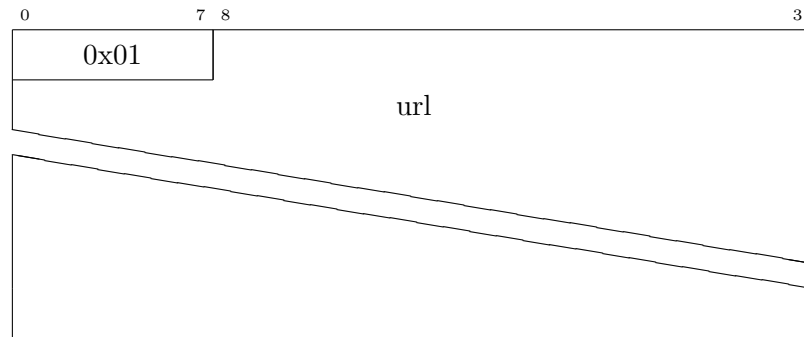   The Uploaded plugins are plugins available in some website, thus can be sent through an URL.

Figure 4.6: Uploaded plugin 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.

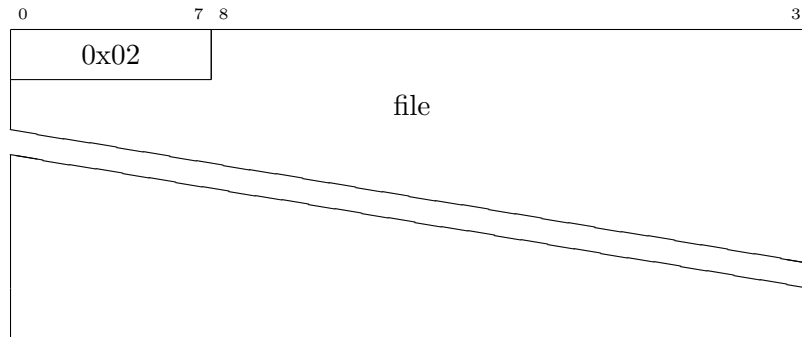As can be seen in the Figure 4.7, File plugin structure, the first argument (that specifies the Plugin type) is 0x02.

Figure 4.7: File plugin structure

A mixed example with different plugin types is needed here.

4.1.4 Server version

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

4.1.5 Config files

Array of files. Usually they have offset 'plugins/', as they provide the default configuration for a given plugin.

4.2 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, while this one goes to the Tester, but both are used for the same purpose: notify the element "1 level up in the hierarchy" that the server has been started. 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.
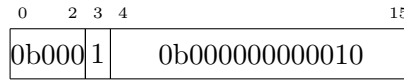
106

Figure 4.8: Server started notification structure

4.3 Error notification

As we're testing a MC plugin it's important to consider all the errors raised during the testing. The *Error notification* is an async petition that it is sent to the WatchWolf Tester orchestrator as a response, without any previous petition.

The *Error notification* will contain the full stack trace of the raised error, captured by the server console.
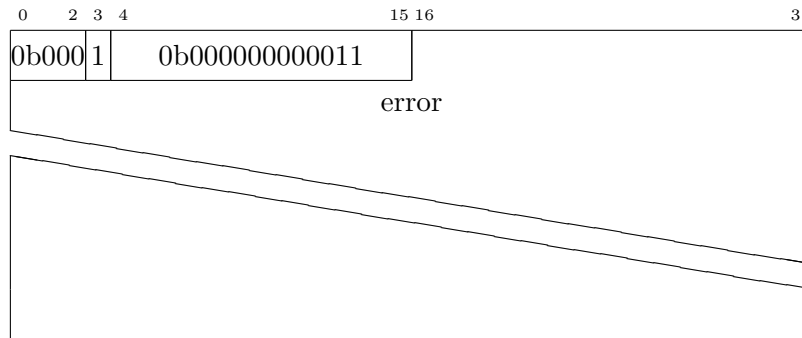

Figure 4.9: Error notification structure

4.4 Get version operation
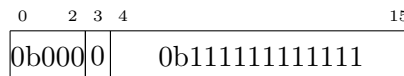
Get the WatchWolf version of the servers manager.

Figure 4.10: Get version operation structure

4.5 Get version response

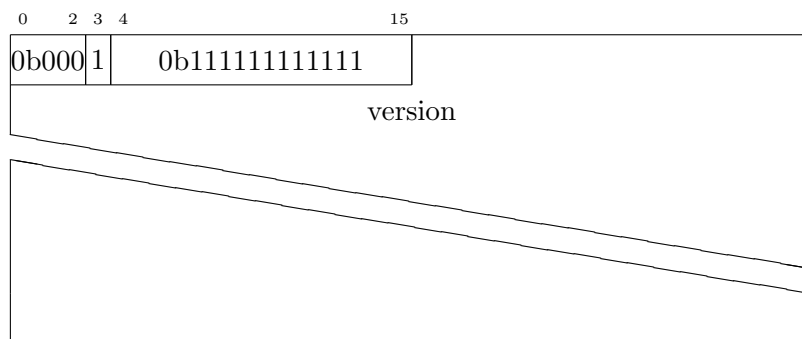The response of *Get version operation* will be the WatchWolf version.

Figure 4.11: Get version response structure

5 Server petition

WathcWolf Server is a Minecraft plugin embedded in a MC server. It will notify to the Servers manager that the server is ready, and perform all the MC server-related requests.

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 3.1, Packet structure) becomes the group, and then the operation is defined on the next 2 bytes, as shown in the Figure 5.1, Server petition structure.
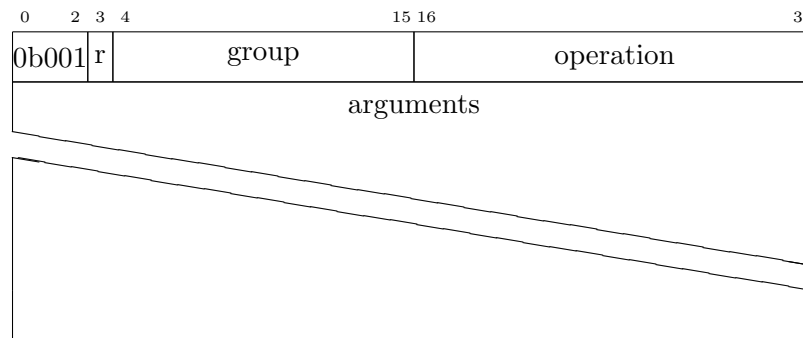
Figure 5.1: Server petition structure

5.1 Server petition group

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

The MSB 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.

type[15]	type[14..4]	Extended type
0	0b00000000000	NOP[11]
0	0b00000000001	Base operations
0	0b00000000010	System operations
0	0b00000000011	Performance operations
0	0b00000010000	WorldGuard operations
0	0b00000010001	Residence operations
1	XXXXXXXXXXX	Reserved for internal use

Table 5.1: Extended types

If you've implemented an extended type and you believe that it makes sense to be part of the API contact contact@watchwolf.dev, or use the Discord to reserve one of the addresses.

5.2 Server petition operation

Like the parameter Operation, it specifies the desired request. For more information, refer to the subsection 3.3, 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 5.2, Implemented group response structure.
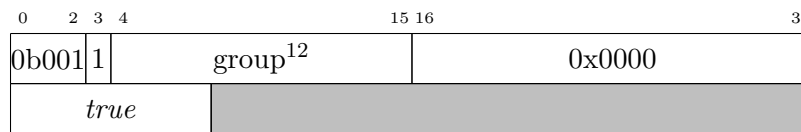

Figure 5.2: Implemented group response structure

5.3 Base operations

The base operations are basic operations needed to interact with the server in order to test the plugins.

---

[11]As stated on the subsection 3.3, Operation, the all-zeroes operation represents a NOP request.

[12]except for groups 0b000000000000 and 0b000000000001

In this case, the 'is implemented' (all zeroes) operation is optional, as it must be present in all the WatchWolf Server implementations.

Once all the operations are implemented, here you'll find a table that shows a link to each one of the operations.

5.3.1 Server stop operation

This operation will stop the server, freeing the resources allocated for it. It must be called once all the testing has been done.

0	2 3 4	15 16	31
0b001	0	0b000000000001	0x0001
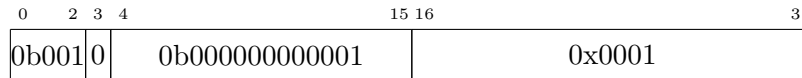
Figure 5.3: Stop server operation structure

5.3.2 Server stopped notification

In response to the *Server stop operation*, you'll get the *Server stopped notification*.

To have more information about the *server id* parameter check the Subsection 5.3.3, Server started notification.

0	2 3 4	15 16	31
0b001	1	0b000000000001	0x0001
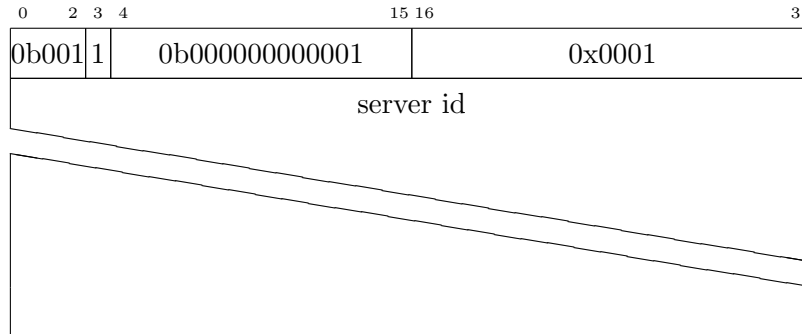
server id

Figure 5.4: Server stopped response structure

5.3.3 Server started notification

This notification is sent to the Server Manager, 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 (because the Tester also knows the server's IP and port), a private hash function is encouraged to be used.

0	2	3	4	15	16	31

0b001	1	0b000000000001	0x0002
server id			
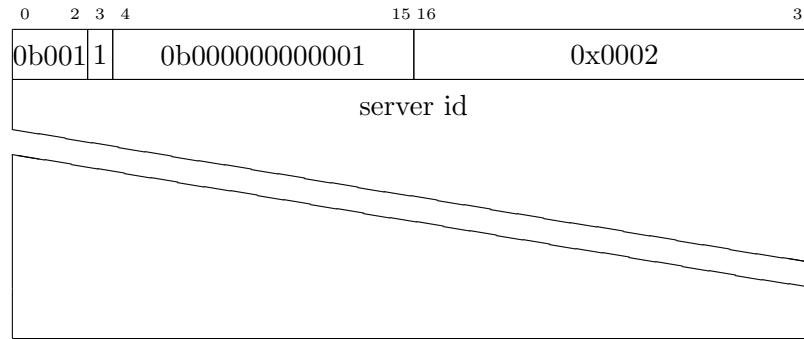
Figure 5.5: Server started response structure

5.3.4 Whitelist player operation

To avoid unwanted players joining the server, all the servers has whitelist on[13] by default. That means that you'll have to add them to the whitelist before they join, and here's where this operation comes in.

Whitelist player operation adds one client to the whitelist.

0	2	3	4	15	16	31

0b001	0	0b000000000001	0x0003
username			
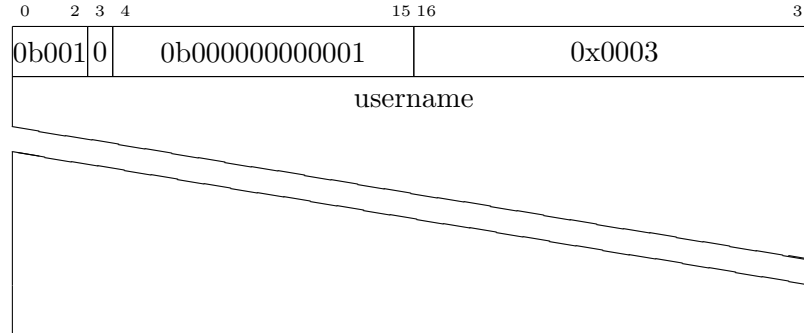
Figure 5.6: Whitelist player operation structure

5.3.5 OP player operation

Some plugin sections may need superuser permissions in order to access them, with the *OP player operation* you'll make one user admin.

---

[13]The whitelist is a set of users that will be allowed to join the world.

0	2 3 4	15 16	31
0b001	0	0b000000000001	0x0004

username
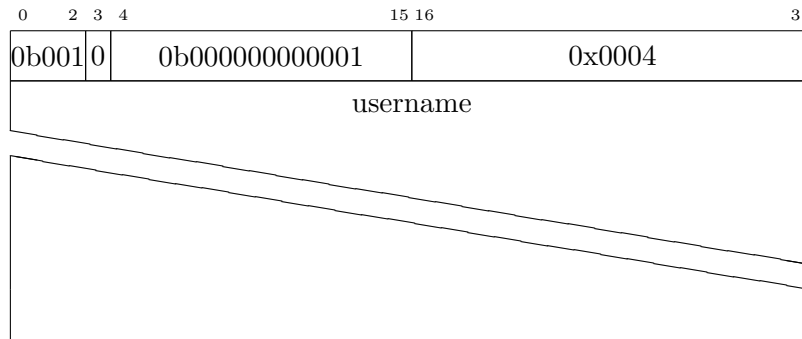
Figure 5.7: OP player operation structure

5.3.6 Set block operation

Place one block somewhere.

0	2 3 4	15 16	31
0b001	0	0b000000000001	0x0005
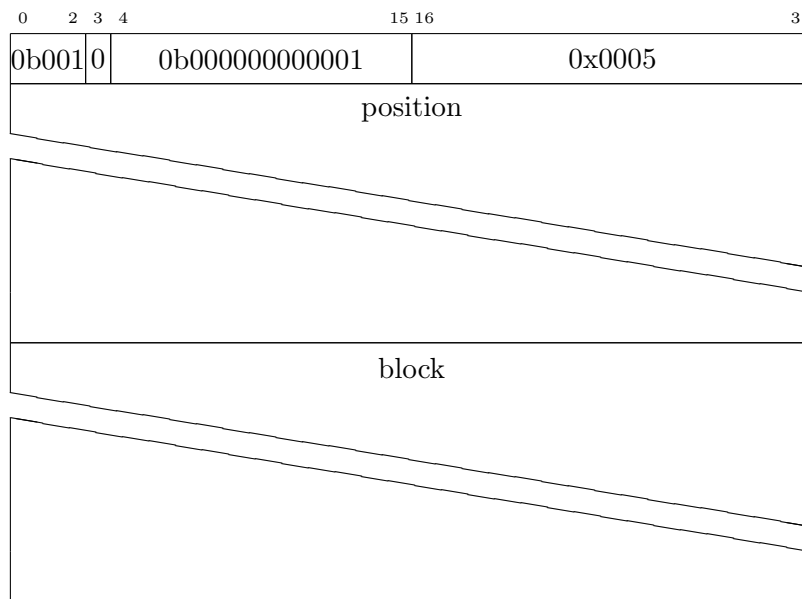
position

block

Figure 5.8: Set block operation structure

5.3.7 Get block operation

Given a position, get the block in that place.

0	2 3	4	15 16	31
0b001	0	0b000000000001	0x0006	
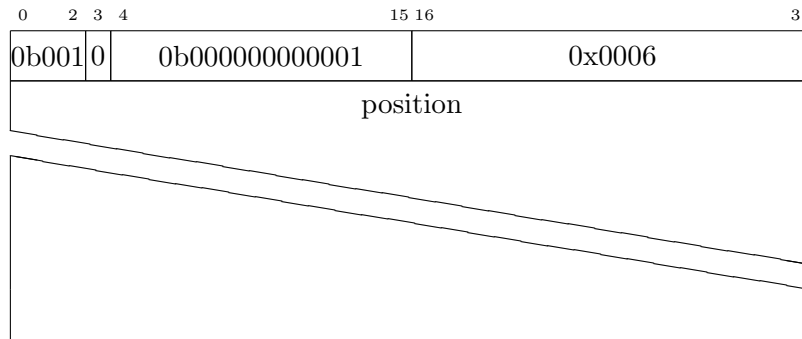
position

Figure 5.9: Get block operation structure

5.3.8 Get block response

As a response of the *Get block operation*, you'll get the obtained block.

0	2 3	4	15 16	31
0b001	1	0b000000000001	0x0006	
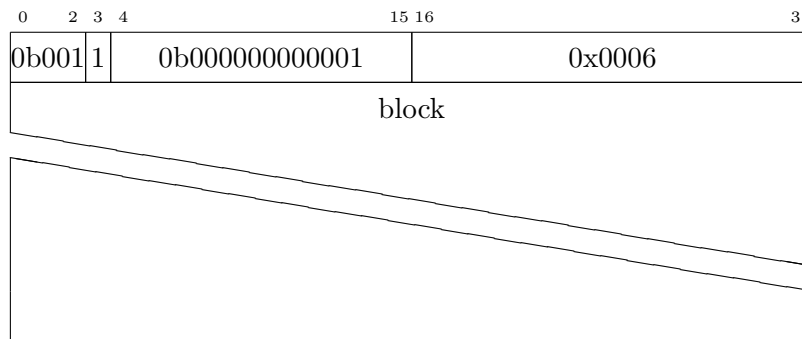
block

Figure 5.10: Get block operation structure

5.3.9 Get player position operation

Get the location of one player.

0	2 3	4	15 16	31
0b001	0	0b000000000001	0x0007	

username

Figure 5.11: Get player position operation structure

5.3.10   Get player position response

As a response of the *Get player position operation*, you'll get the player position.

0	2 3 4	15 16	31
0b001	1	0b000000000001	0x0007
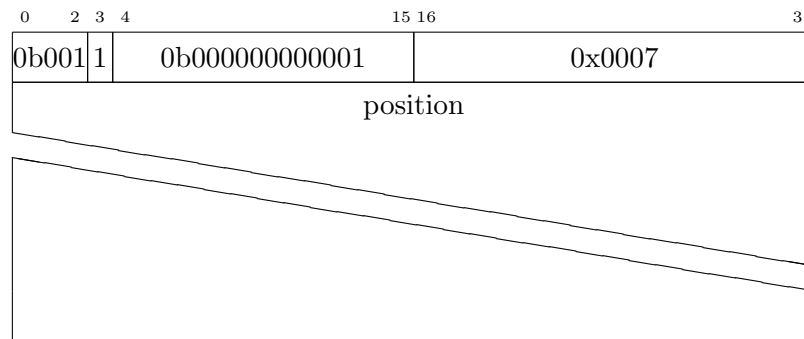
position

Figure 5.12: Get player position response structure

5.3.11   Give item to player operation

Add some item to one user.

0	2 3 4	15 16	31
0b001	0	0b000000000001	0x0008

username
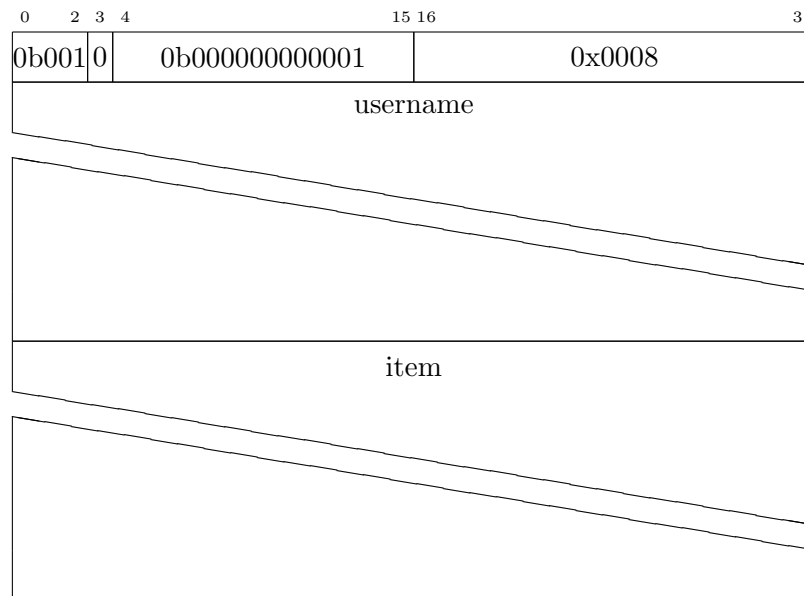
item

Figure 5.13: Give item to player operation structure

5.3.12   Run command operation

Execute one command from a console point of view.

As a side note, this operation may be useful if you need to perform one action not present at this moment by the standard.

0		2	3	4		15	16			31
0b001	0			0b000000000001				0x0009		
cmd										
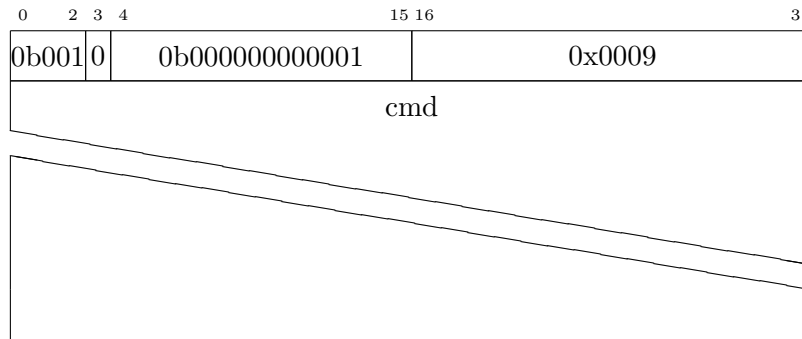
Figure 5.14: Run command operation structure

5.3.13   Run command response

In response of the *Run command operation*, it will return (if any) the response of that command. If there's no response, it will return an empty string ("").

0		2	3	4		15	16			31
0b001	1			0b000000000001				0x0009		
cmd response										
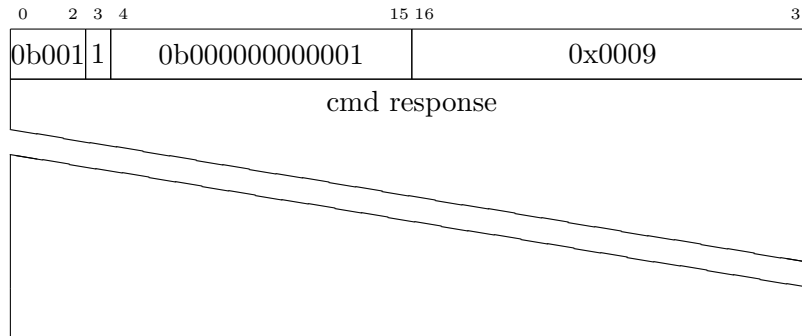
Figure 5.15: Run command operation structure

5.3.14   Get players operation

Get all the players' usernames on-line.

0		2	3	4		15	16			31
0b001	0			0b000000000001				0x000A		
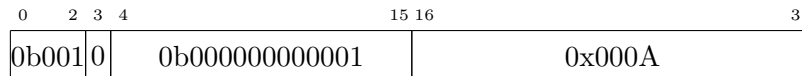
Figure 5.16: Get players operation structure

5.3.15   Get players response

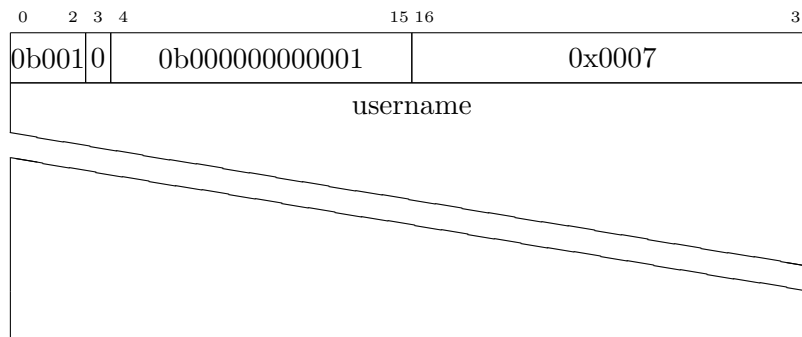As a response of the *Get players operation*, you'll get one String array with all the names.

```
0    2 3 4              15 16                    31
0b001 1   0b000000000001        0x000A
                usernames
```
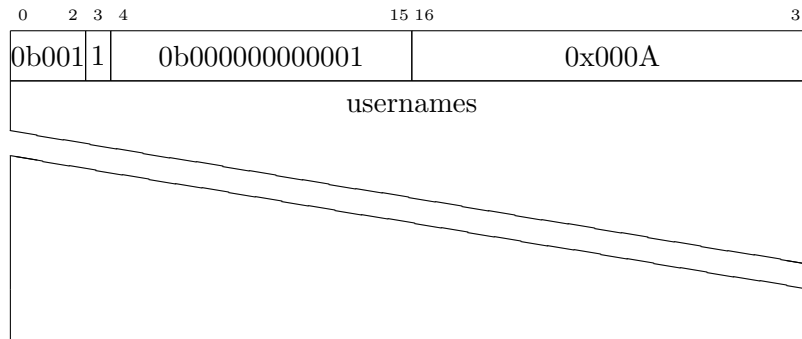
Figure 5.17: Get players response structure

5.3.16 Synchronize operation

As discussed in Subsection 3.5, Synchronization, *Synchronize operation* will guarantee the expected outcome of the tests.

```
0    2 3 4              15 16                    31
0b001 0   0b000000000001        0x000B
```
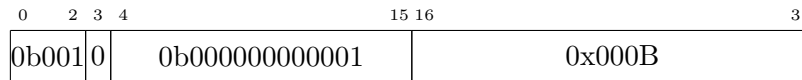
Figure 5.18: Synchronize operation structure

5.3.17 Synchronize response

As a response of the *Synchronize operation*, this indicates that all the operations sent by the Tester has been done, thus being "on the same state" as WatchWolf Tester.

```
0    2 3 4              15 16                    31
0b001 1   0b000000000001        0x000B
```
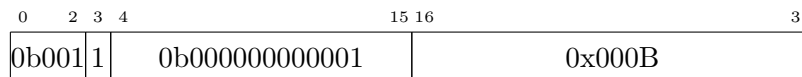
Figure 5.19: Synchronize response structure

5.3.18 Tp player operation

Teleport a player to one location.
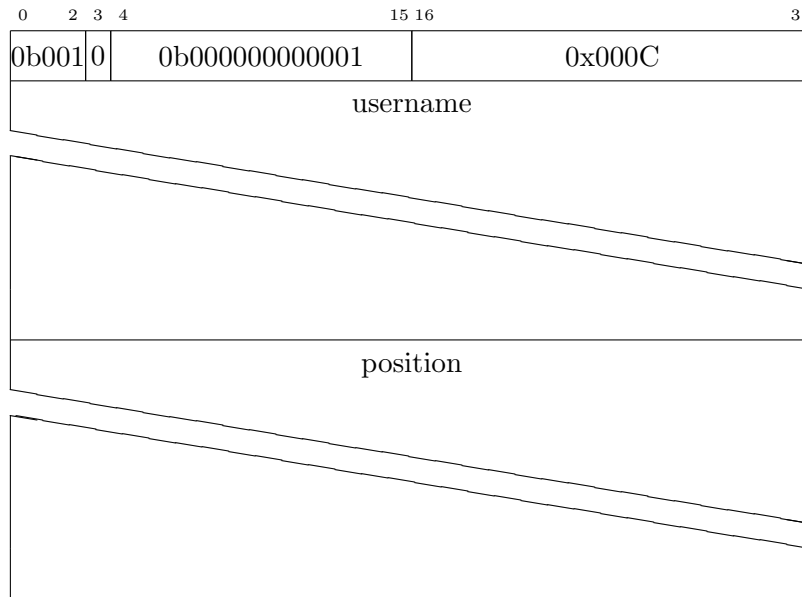
Figure 5.20: Tp player operation structure

5.3.19 Get player pitch operation

With *Get player yaw operation*, it will allow the Tester to know where the player is looking.
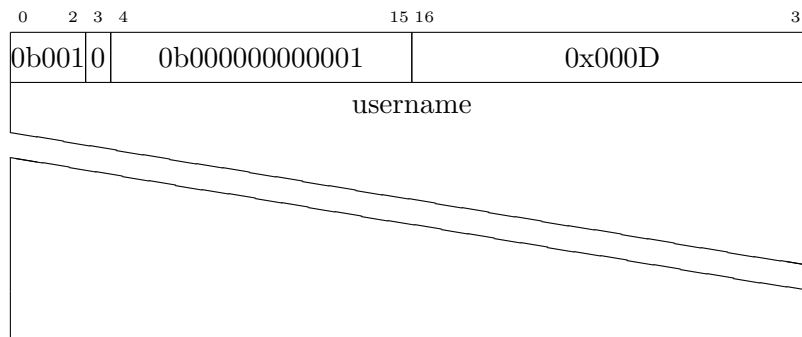
Figure 5.21: Get player pitch operation structure

5.3.20 Get player pitch response

Get player pitch operation response.

0		2	3	4		15	16		31
0b001	1			0b000000000001			0x000D		
pitch									
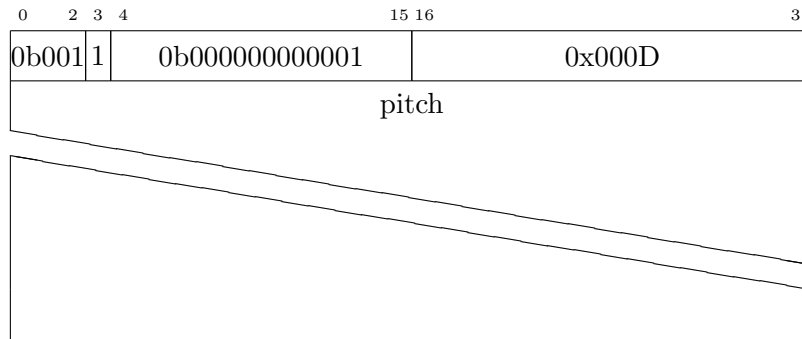
Figure 5.22: Get player pitch response structure

5.3.21   Get player yaw operation

With *Get player pitch operation*, it will allow the Tester to know where the player is looking.

0		2	3	4		15	16		31
0b001	0			0b000000000001			0x000E		
username									
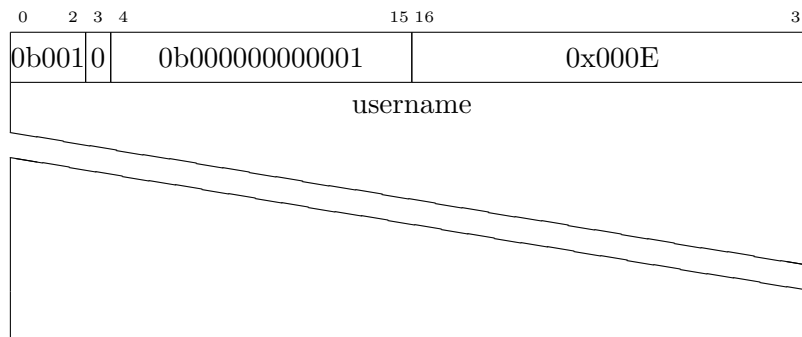
Figure 5.23: Get player yaw operation structure

5.3.22   Get player yaw response

Get player yaw operation response.

0		2	3	4		15	16		31
0b001	1			0b000000000001			0x000E		
yaw									
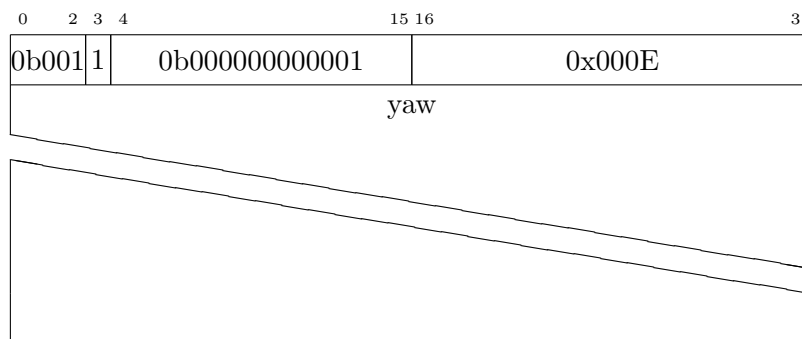
Figure 5.24: Get player yaw response structure

5.3.23 Get player inventory operation

Get the container (inventory) of one player.

0		2	3	4	15	16	31
0b001	0			0b000000000001		0x000F	
username							
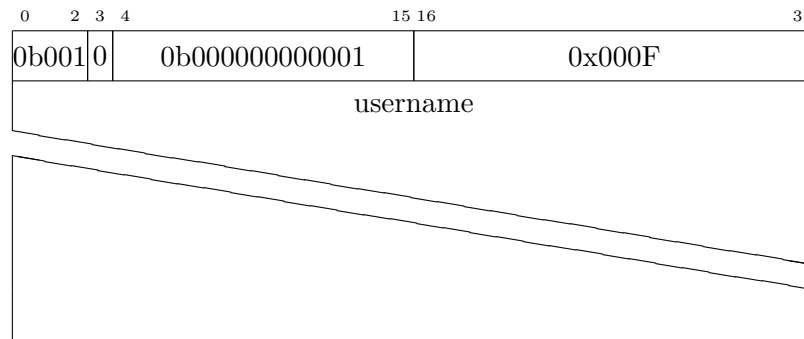
Figure 5.25: Get player inventory operation structure

5.3.24 Get player inventory response

Response to the *Get player inventory operation* with the player's inventory.

0		2	3	4	15	16	31
0b001	1			0b000000000001		0x000F	
inventory							
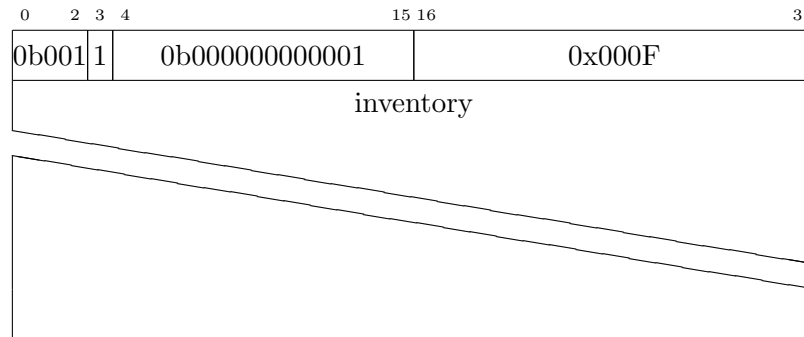
Figure 5.26: Get player inventory response structure

5.3.25 Get entities operation

Given one point,[14] it will return all the entities around this spot.

---

[14]For how Minecraft loading system works, you cannot get all the entities. When a MC chunk is unloaded it's like there's no entities on that place, and they will remain there until it's re-loaded again.
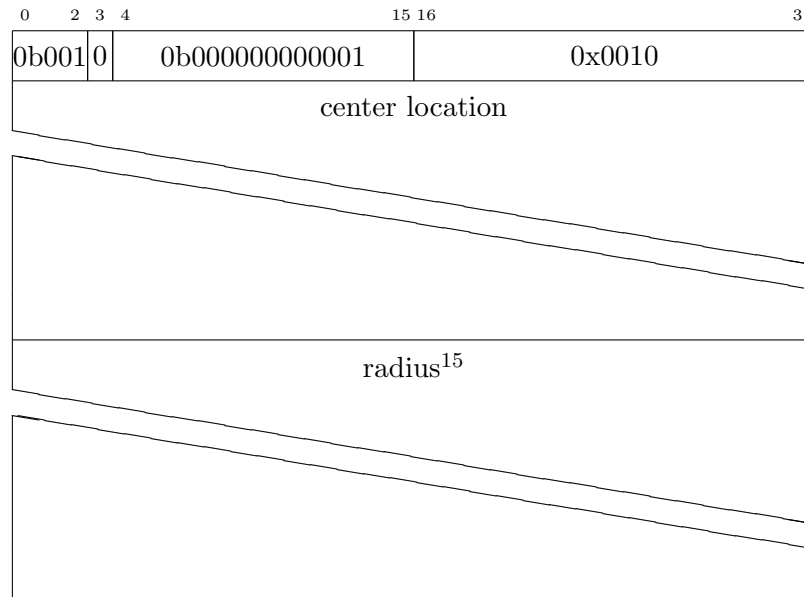
Figure 5.27: Get entities operation structure

5.3.26 Get entities response

Get entities operation response with all the getted entities.
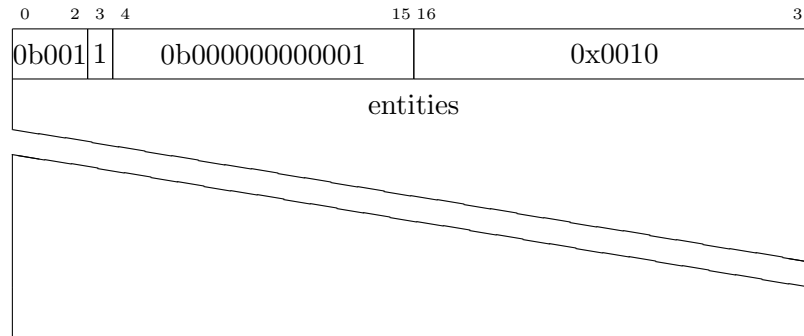
Figure 5.28: Get entities response structure

5.3.27 Get entity operation

Given one Entity UUID, it will return that entity. If the entity doesn't exists, the (fake) returned entity will have a different UUID that the one requested.

---

[15]Large radius or locations in unloaded chunks may return unexpected data.

```
  0     2 3  4                          15 16                                31
┌─────┬─┬──────────────────────┬────────────────────────────────────┐
│0b001│0│   0b000000000001      │             0x0012                 │
├─────┴─┴──────────────────────┴────────────────────────────────────┤
│                           UUID                                     │
│                                                                    │
│                                                                    │
│                                                                    │
└────────────────────────────────────────────────────────────────────┘
```
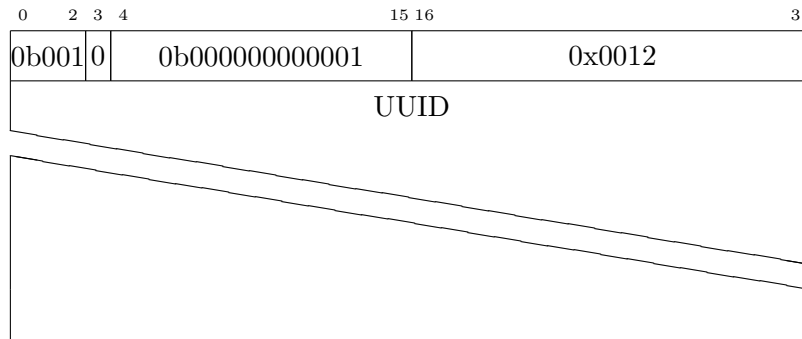
Figure 5.29: Get entities operation structure

5.3.28 Get entity response

Get entity operation response with the getted entity.

```
  0     2 3  4                          15 16                                31
┌─────┬─┬──────────────────────┬────────────────────────────────────┐
│0b001│1│   0b000000000001      │             0x0012                 │
├─────┴─┴──────────────────────┴────────────────────────────────────┤
│                           entity                                   │
│                                                                    │
│                                                                    │
│                                                                    │
└────────────────────────────────────────────────────────────────────┘
```
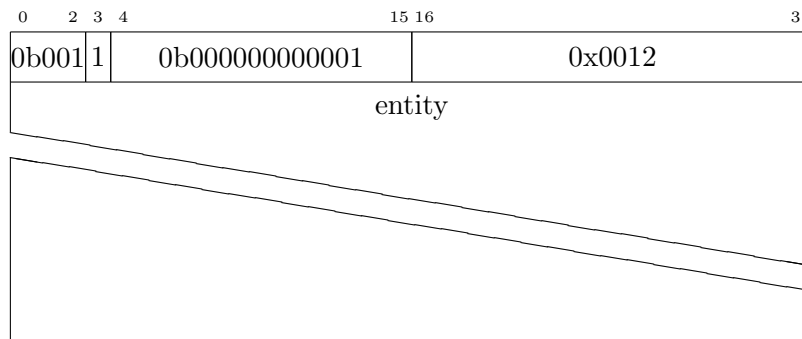
Figure 5.30: Get entities response structure

5.3.29 Spawn entity operation

Summon one entity. Note that Entity's UUID will be ignored, as it's assigned by the server and cannot be set.

```
  0     2 3  4                          15 16                                31
┌─────┬─┬──────────────────────┬────────────────────────────────────┐
│0b001│0│   0b000000000001      │             0x0011                 │
├─────┴─┴──────────────────────┴────────────────────────────────────┤
│                           entity                                   │
│                                                                    │
│                                                                    │
│                                                                    │
└────────────────────────────────────────────────────────────────────┘
```
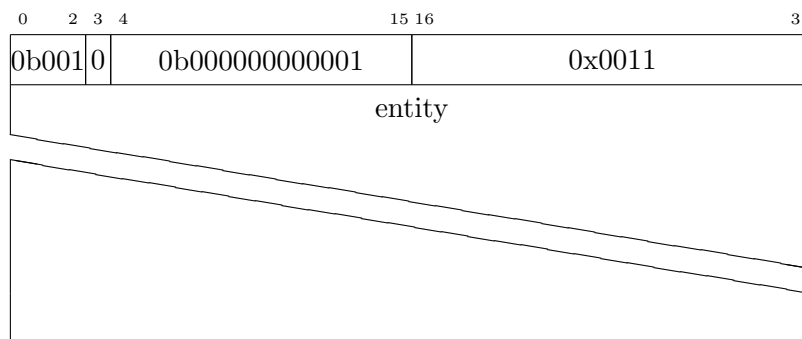
Figure 5.31: Spawn entity operation structure

5.3.30   Spawn entity response

The response of *Spawn entity operation* will be that new entity. It should match the request, except for the UUID (that now it will have a valid one).
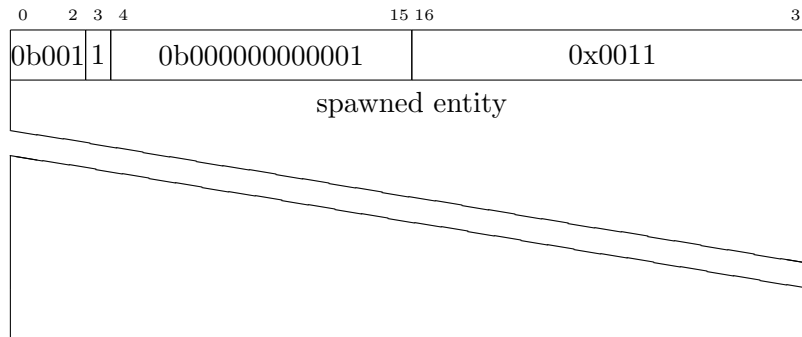
Figure 5.32: Spawn entity response structure

5.3.31   Change difficulty operation

Change the server difficulty. You can set it to peaceful (0), easy (1), normal (2) and hard (3).
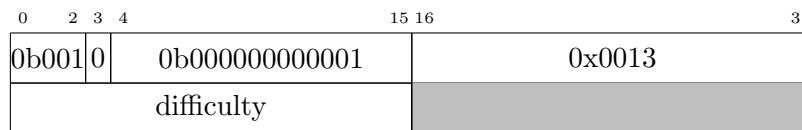
Figure 5.33: Change difficulty operation structure

5.3.32   Get version operation

Get the WatchWolf version of the server.
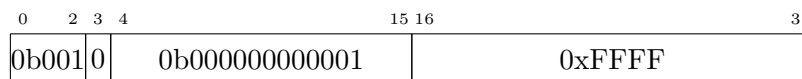
Figure 5.34: Get version operation structure

5.3.33   Get version response

The response of *Get version operation* will be the WatchWolf version.

0	2 3 4		15 16	31
0b001	1	0b000000000001		0xFFFF
version				
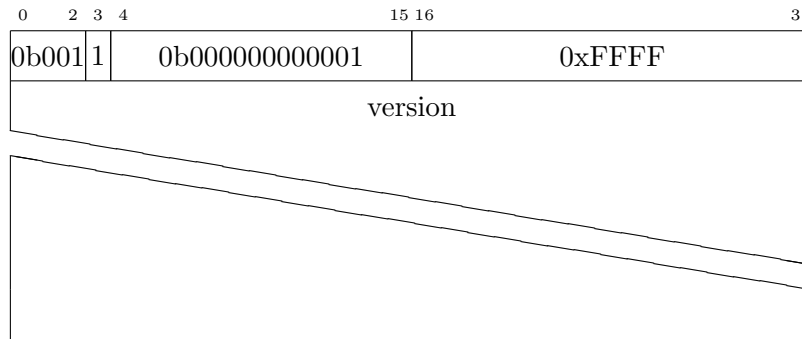
Figure 5.35: Get version response structure

5.4 Enhanced information operations

The *enhanced information* operation provides extra data to be used while reviewing your plugin. Right now there's two type of extra information: a video of what's happening in the world (place camera, move camera, and stop recording), and getting the timings report[16] (incomplete section).

This section is incomplete. Expect some future work here about measuring the performance of your plugins.

5.4.1 Place camera operation

Places a camera in the target location and starts recording.

---

[16]The "timings report" specifies what % of the CPU is being used on each Minecraft command, so you can see performance problems by reviewing that report.

0	2 3 4	15 16	31
0b001	0	0b000000000010	0x0001

position

pitch[17]

yaw[18]
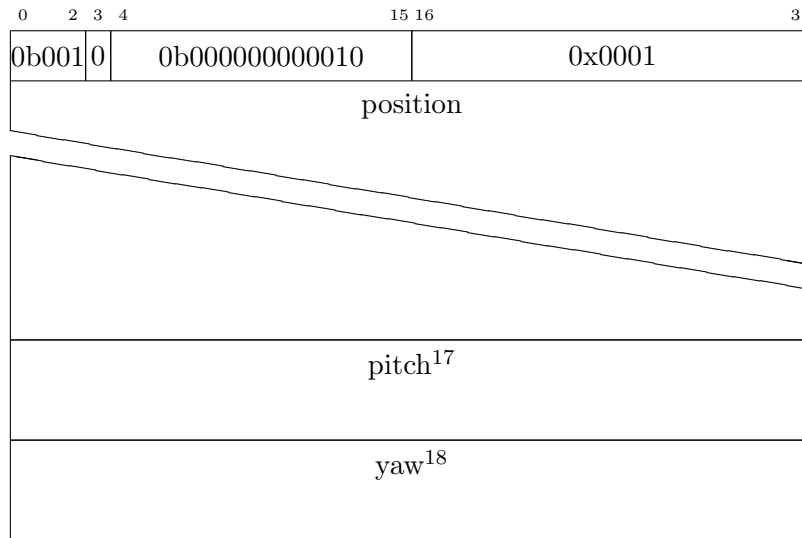
Figure 5.36: Place camera operation structure

5.4.2 Place camera response

The response of *Place camera operation* will be the ID of the placed camera.

0	2 3 4	15 16	31
0b001	1	0b000000000010	0x0001
camera ID			
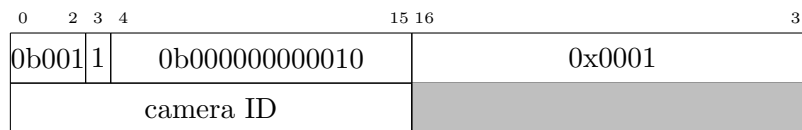
Figure 5.37: Place camera response structure

5.4.3 Move camera operation

Moves an already placed camera into the target location.

---

[17]Minecraft's pitch is the vertical direction the user is looking at. It goes from 90 (down), to 0 (horizontal), to -90 (up).

[18]Minecraft's yaw is the horizontal direction the user is looking at. A value of 0 means south, 90 west, ±180 north, and -90 east.

0		2	3	4		15	16			31
0b001	0			0b000000000010				0x0002		
camera ID							position			

pitch[17]

yaw[18]
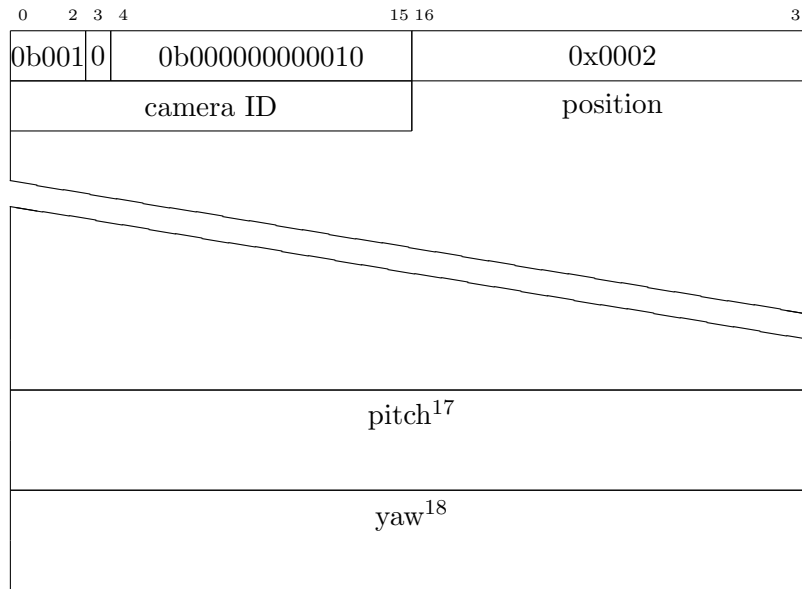
Figure 5.38: Move camera operation structure

5.4.4   Stop camera operation

Stops a camera and retrieves the recording.

0		2	3	4		15	16			31
0b001	0			0b000000000010				0x0003		
camera ID										
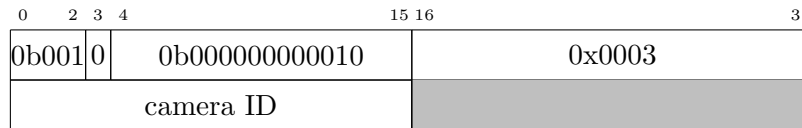
Figure 5.39: Stop camera operation structure

5.4.5   Stop camera response

The response of *Stop camera operation* will be the recording.

0		2	3	4		15	16			31
0b001	1			0b000000000010				0x0003		
recording										
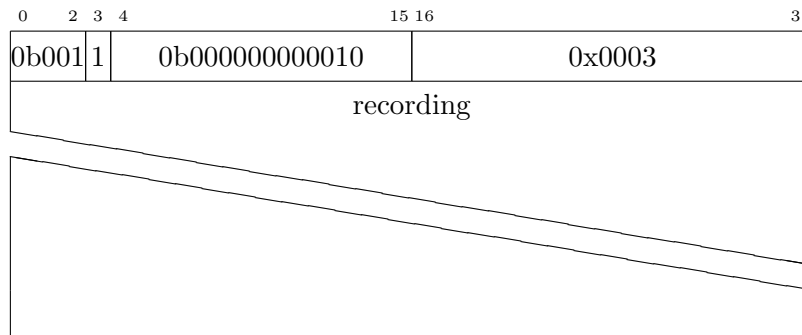
Figure 5.40: Stop camera response structure

126

5.4.6   Start timings report operation

Starts the timings report.

0	2 3 4	15 16	31
0b001	0	0b000000000010	0x0004
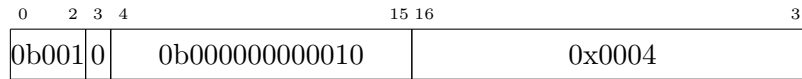
Figure 5.41: Start timings report operation structure

5.4.7   Stop timings report operation

Stops the timings report and retrieves the raw file.

0	2 3 4	15 16	31
0b001	0	0b000000000010	0x0005
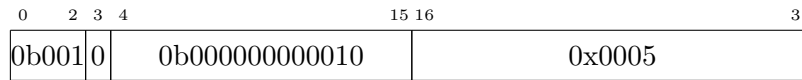
Figure 5.42: Stop timings report operation structure

5.4.8   Stop timings report response

The response of *Stop timings operation* will be that report.

0	2 3 4	15 16	31
0b001	1	0b000000000010	0x0005
report			
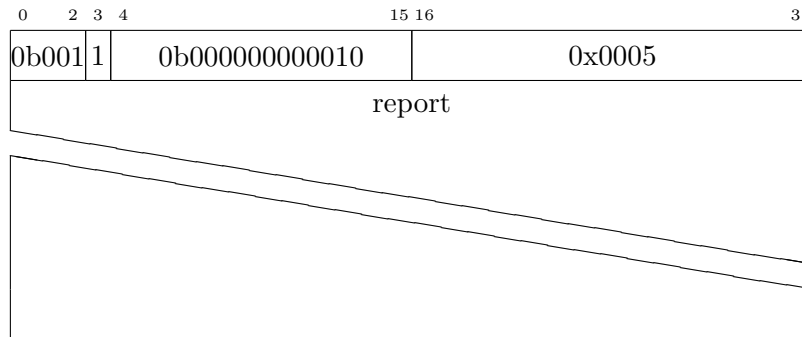
Figure 5.43: Stop timings report response structure

5.5   WorldGuard operations

In this section you'll find some operations to interact with WorldGuard directly with the WatchWolf framework.

Some more methods may be needed to fully support WorldGuard's functionalities. You can request them over the Discord.

5.5.1   Create region operation

Creates a rectangular region.

0	2	3	4		15	16		31

0b001 | 0 | 0b000000000011 | 0x0001

region name

first coordinate

second coordinate
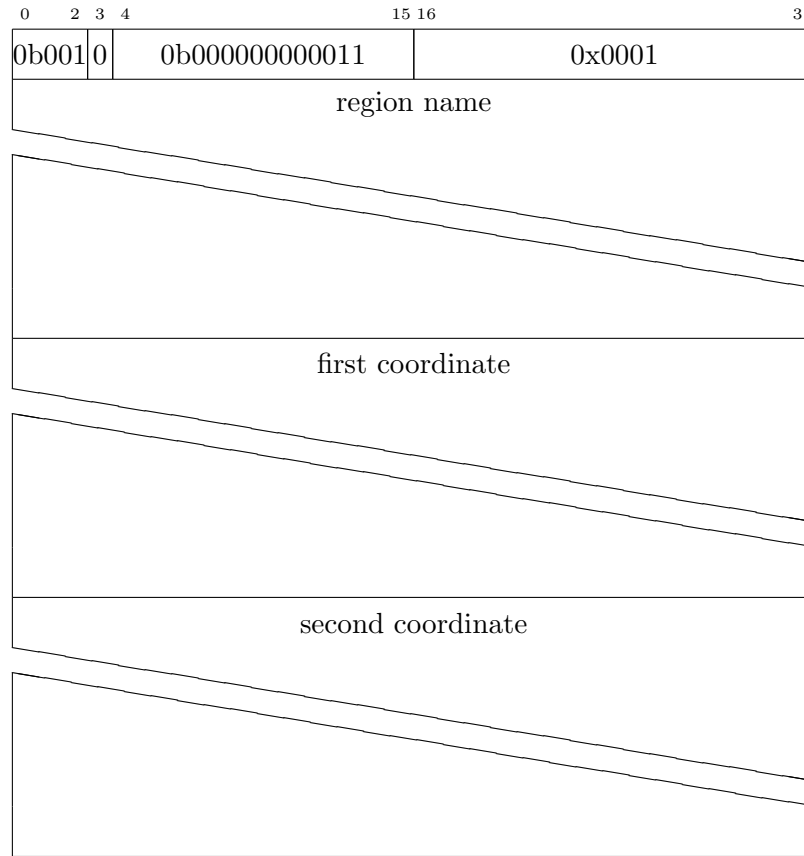
Figure 5.44: Create region operation structure

5.5.2   Get regions operation

Lists all the WorldGuard regions.

0b001 | 0 | 0b000000000011 | 0x0002
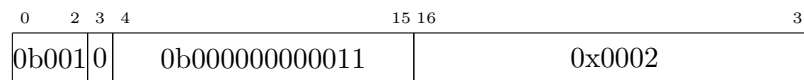
Figure 5.45: Get regions operation structure

5.5.3   Get regions response

The response of *Get regions operation*. You'll get the name of all the regions.

```
0        2  3  4                    15 16                          31
┌──────┬──┬─────────────────────────┬─────────────────────────────┐
│0b001 │1 │  0b000000000011         │           0x0002            │
├──────┴──┴─────────────────────────┴─────────────────────────────┤
│                    array of region names                        │
│                                                                 │
│                                                                 │
└─────────────────────────────────────────────────────────────────┘
```
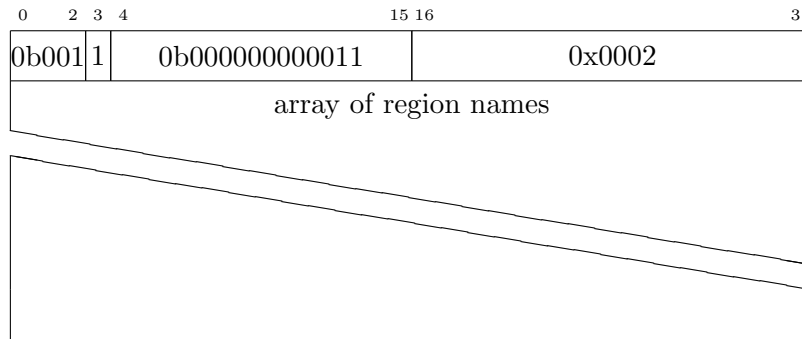
Figure 5.46: Get regions response structure

5.5.4 Get regions at position operation

Lists all the WorldGuard regions in one position.

WorldGuard allows multiple regions/sub-regions to overlap, so it may be possible to get more than one.

```
0        2  3  4                    15 16                          31
┌──────┬──┬─────────────────────────┬─────────────────────────────┐
│0b001 │0 │  0b000000000011         │           0x0003            │
├──────┴──┴─────────────────────────┴─────────────────────────────┤
│                         block                                   │
│                                                                 │
│                                                                 │
└─────────────────────────────────────────────────────────────────┘
```
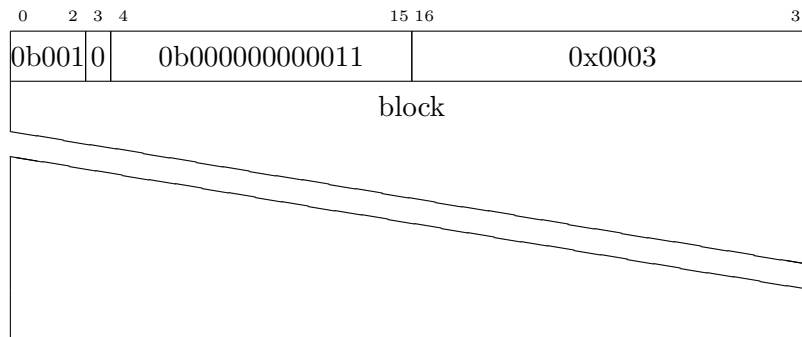
Figure 5.47: Get regions at position operation structure

5.5.5 Get regions at position response

The response of *Get regions at position operation*. You'll get the name of all the regions present in that block.

0	2	3	4	15	16	31

0b001	1	0b000000000011	0x0003

array of region names
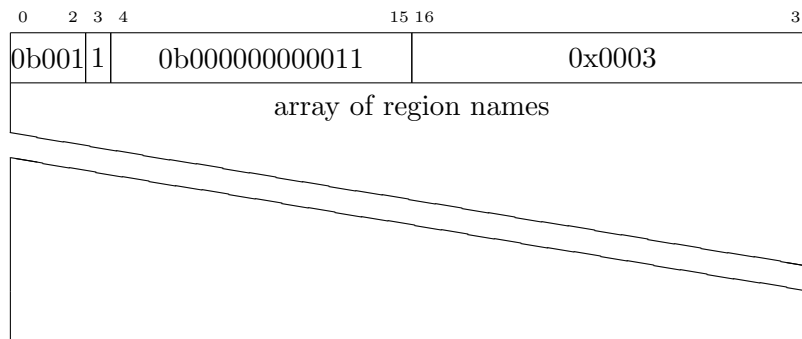
Figure 5.48: Get regions at position response structure

5.6 Residence operations

This section is incomplete. Expect some future work here about creating Residence regions.

6   Clients manager petition

As in Section 4, Servers manager petition, *Clients manager petition* will allow WatchWolf Tester to create users that will connect to the (already) started servers.
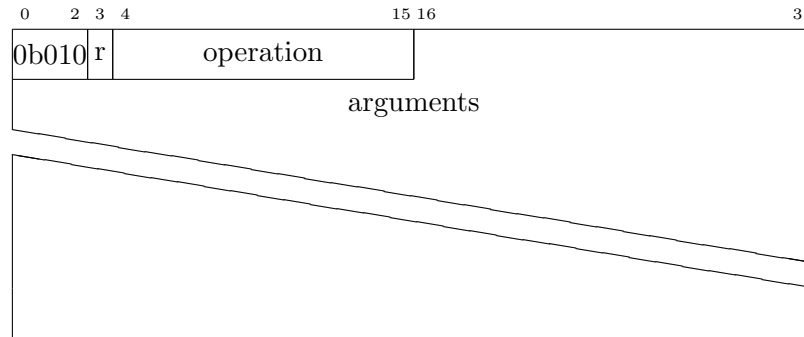
Figure 6.1: Clients manager petition structure

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

Once all the operations are implemented, here you'll find a table that shows a link to each one of the operations.

6.1   Start client operation

This operation will start an offline[19] Minecraft client, and it will be connected to the specified server.

---

[19]Minecraft needs payment in order to create an account. With *offline mode* no account is required, that is, no payment nor password needed to be provided. As a petition by the community, there will be (in the future) one way to embed your Premium Minecraft accounts to the WatchWolf environment.

```
0    2 3 4              15 16                            31
┌──────┬─┬───────────────────┬──────────────────────────┐
│0b010 │0│ 0b000000000001    │                          │
├──────┴─┴───────────────────┴──────────────────────────┤
│                     username                           │
```
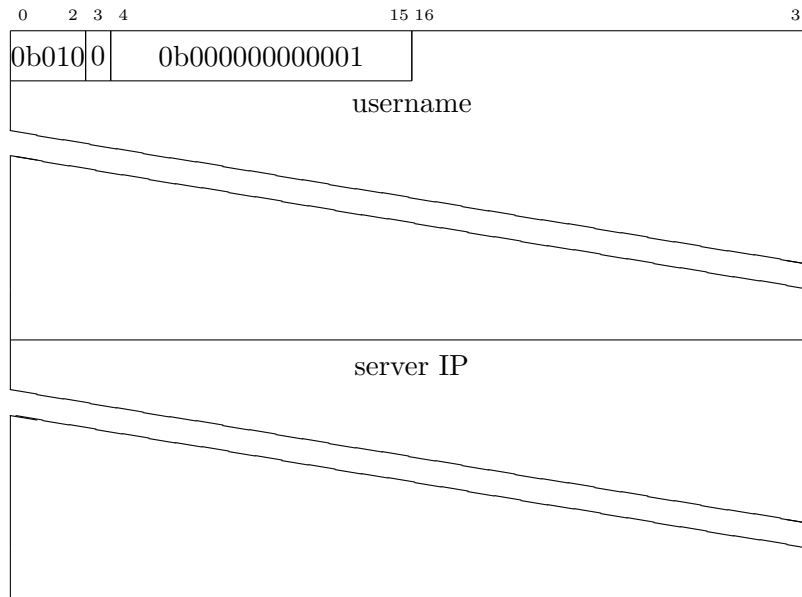
Figure 6.2: Start client petition structure

6.2 Client started response operation

As a response to the *Start client operation*, you'll get the IP (IP & port) to connect to the client.

```
0    2 3 4              15 16                            31
┌──────┬─┬───────────────────┬──────────────────────────┐
│0b010 │1│ 0b000000000001    │                          │
├──────┴─┴───────────────────┴──────────────────────────┤
│                     ip:port                            │
```
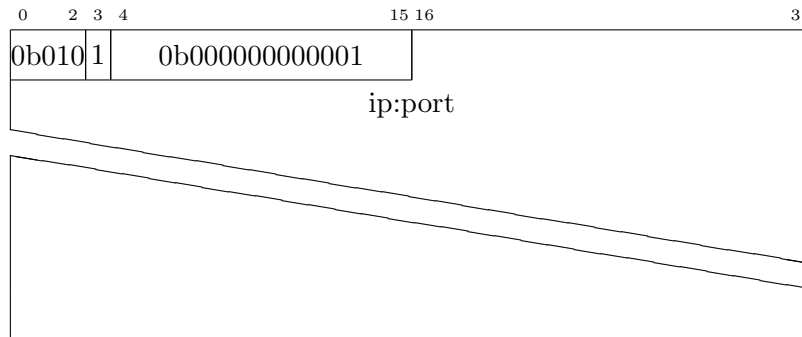
Figure 6.3: Client started petition response structure

6.3 Get version operation

Get the WatchWolf version of the clients manager.

```
0    2 3 4                          15
┌──────┬─┬───────────────────────────┐
│0b010 │0│   0b111111111111          │
└──────┴─┴───────────────────────────┘
```
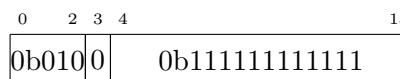
Figure 6.4: Get version operation structure

6.4 Get version response

The response of *Get version operation* will be the WatchWolf version.
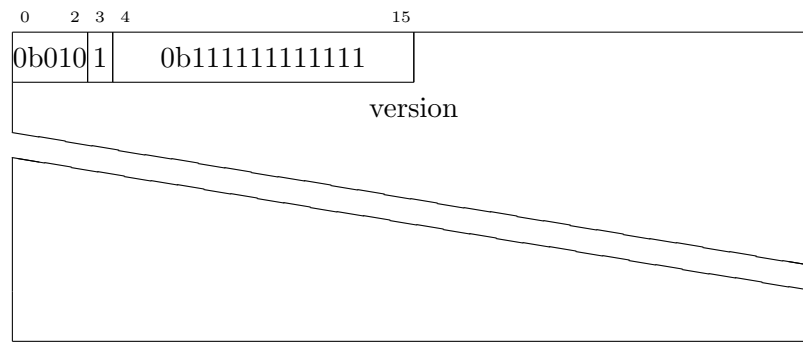
Figure 6.5: Get version response structure

7 Client petition

The client will interact with the server as if it was one normal user.
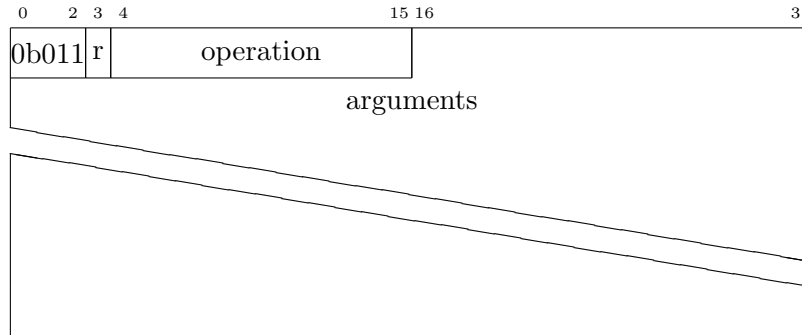
Figure 7.1: Client petition structure

Once all the operations are implemented, here you'll find a table that shows a link to each one of the operations.

7.1 Stop client operation

Leave the server, and destroy the client.
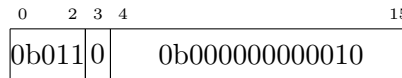
Figure 7.2: Stop client petition structure

7.2 Send message operation

Send a chat message.

Note that this operation is for messages. If you want to run a command, refer to Subsection 7.4, Run command operation.
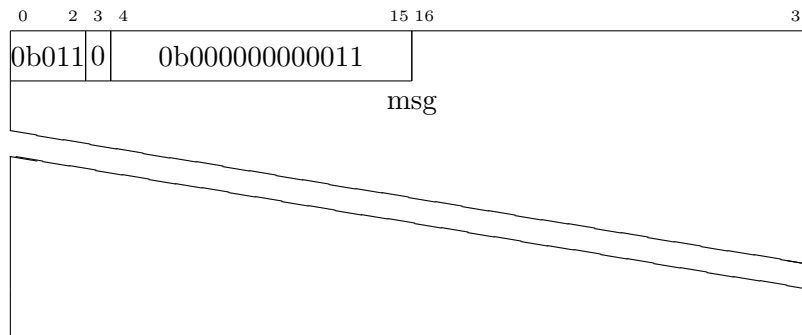
Figure 7.3: Send message petition structure

7.3 Got message notification

This async notification is received when one user reads a chat message from another user.

0	2	3	4	15	16	31

```
0b011  1      0b000000000011
              sender

              msg
```
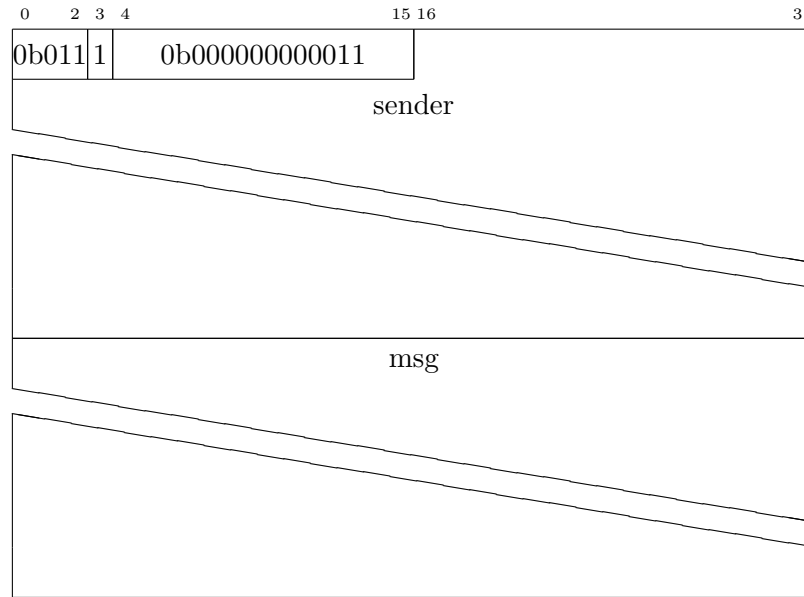
Figure 7.4: Got message notification structure

7.4 Run command operation

Run a command.

As we can't guarantee if the server has already replied, is busy processing data, or has no reply, we need to establish a timeout.

```
0b011  0      0b000000000100
              cmd

       ms to timeout
```
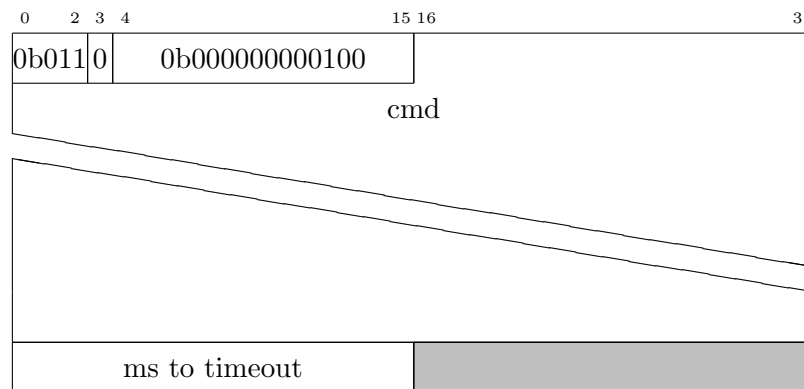
Figure 7.5: Run command petition structure

7.5 Run command response

In response of the *Run command operation*, it will return the message related to the command. If none, it will return an empty string ("").
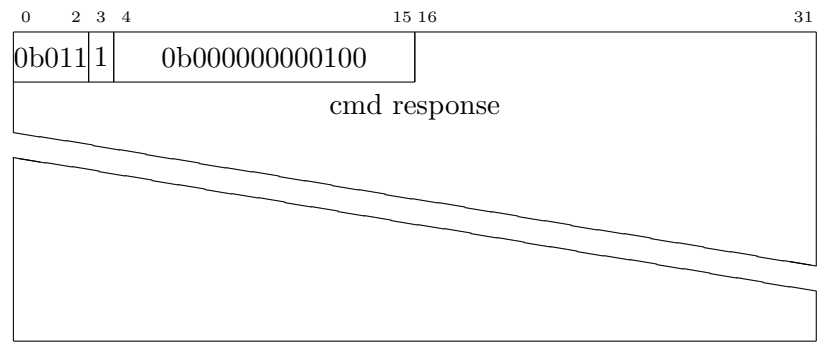
Figure 7.6: Run command petition structure

7.6 Break block operation

Break a block with the held item.
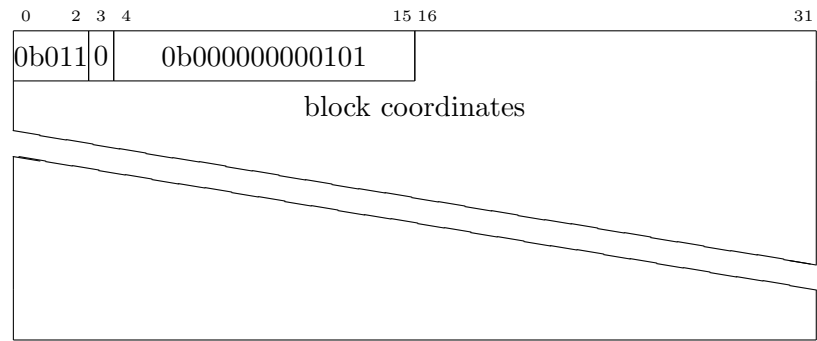
Figure 7.7: Run command petition structure

7.7 Equip item in hand operation

Equip one item (already existing in the inventory) on hand.

0		2	3	4			15	16			31

```
0b011  0    0b000000000110
              item
```
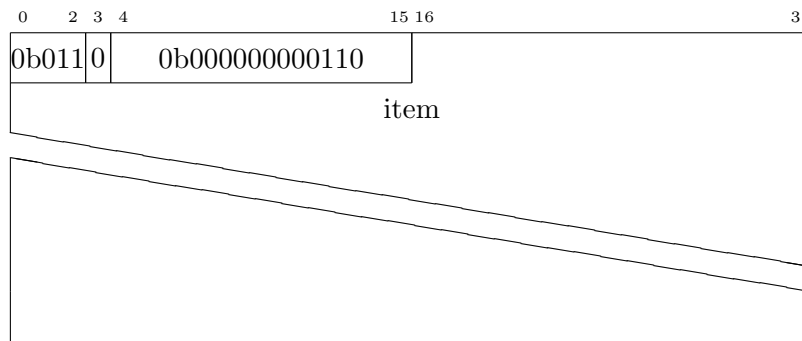
Figure 7.8: Run command petition structure

7.8   Move at operation

Move the player from the current position to the target location.

```
0b011  0    0b000000000111
              pos
```
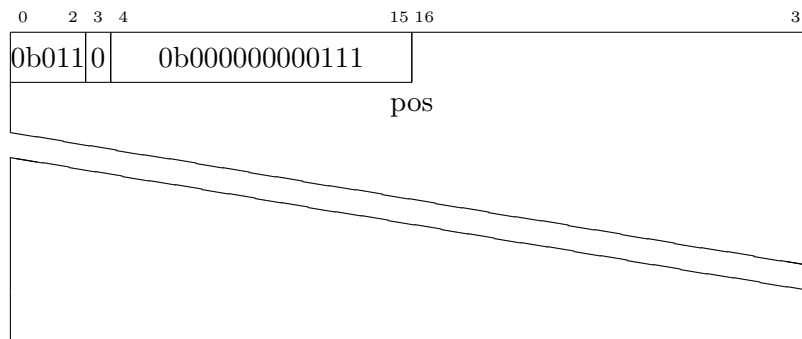
Figure 7.9: Move at petition structure

7.9   Look at operation

Move the player's camera.

```
 0    2 3 4                       15 16                              31
┌─────┬─┬──────────────────────┬─────────────────────────────────────┐
│0b011│0│   0b000000001000     │                                     │
├─────┴─┴──────────────────────┴─────────────────────────────────────┤
│                          pitch[17]                                  │
```
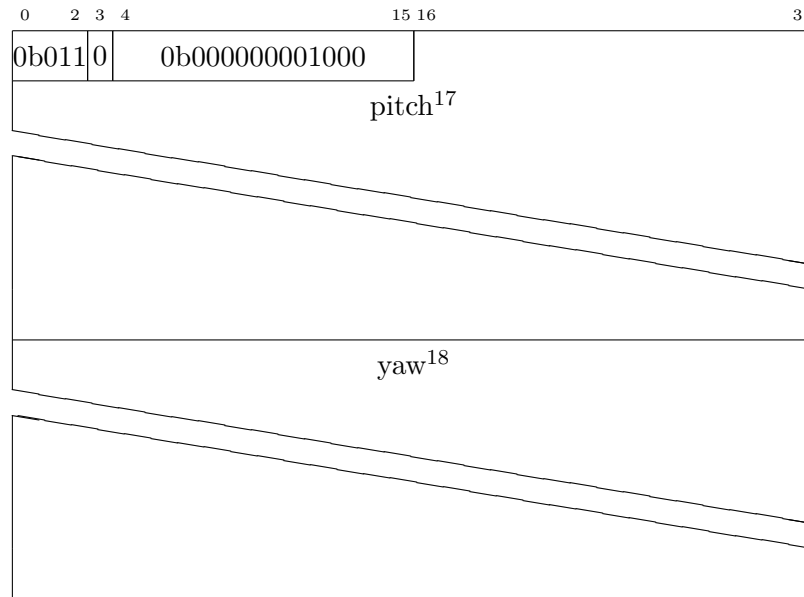
Figure 7.10: Move at petition structure

7.10 Synchronize operation

As discussed in Subsection 3.5, Synchronization, *Synchronize operation* will guarantee the expected outcome of the tests.

```
 0    2 3 4                    15
┌─────┬─┬─────────────────────┐
│0b011│0│    0b000000001001   │
└─────┴─┴─────────────────────┘
```
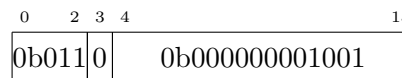
Figure 7.11: Synchronize petition structure

7.11 Synchronize response

As a response of the *Synchronize operation*, this indicates that all the operations sent by the Tester has been done, thus being "on the same state" as WatchWolf Tester.

```
 0    2 3 4                    15
┌─────┬─┬─────────────────────┐
│0b011│1│    0b000000001001   │
└─────┴─┴─────────────────────┘
```
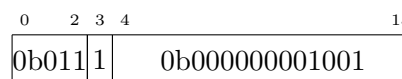
Figure 7.12: Synchronize petition response structure

7.12 Hit operation

Equivalent to the left click with item in hand. Currently (v0.1.15) this operation doesn't work as intended.

```
 0     2  3  4                          15
┌──────┬──┬────────────────────────────┐
│0b011 │0 │    0b000000001010          │
└──────┴──┴────────────────────────────┘
```
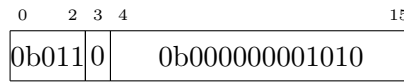
Figure 7.13: Hit petition structure

7.13   Use operation

Equivalent to the right click with item in hand.

```
 0     2  3  4                          15
┌──────┬──┬────────────────────────────┐
│0b011 │0 │    0b000000001011          │
└──────┴──┴────────────────────────────┘
```
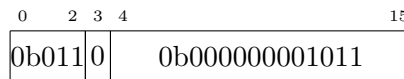
Figure 7.14: Use petition structure

7.14   Place block operation

Place the current item in hand to the specified location.

It's important that the coordinates must be inside the player range; as this is not a *move at* operation.

```
 0     2  3  4              15 16                    31
┌──────┬──┬──────────────────┐
│0b011 │0 │  0b000000001100  │
├──────┴──┴──────────────────┴──────────────────────┤
│                block coordinates                    │
└─────────────────────────────────────────────────────┘
```
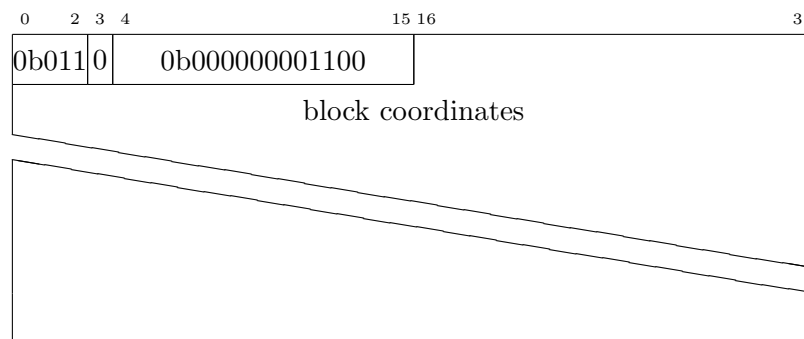
Figure 7.15: Place block petition structure

7.15   Hit entity operation

Attack (1 hit) one entity with the current held item.

```
 0    2 3 4                    15 16                              31
┌──────┬─┬──────────────────────┬──────────────────────────────────┐
│0b011 │0│   0b000000001101      │                                  │
├──────┴─┴──────────────────────┴──────────────────────────────────┤
│                        entity's UUID                              │
│                                                                   │
└───────────────────────────────────────────────────────────────────┘
```
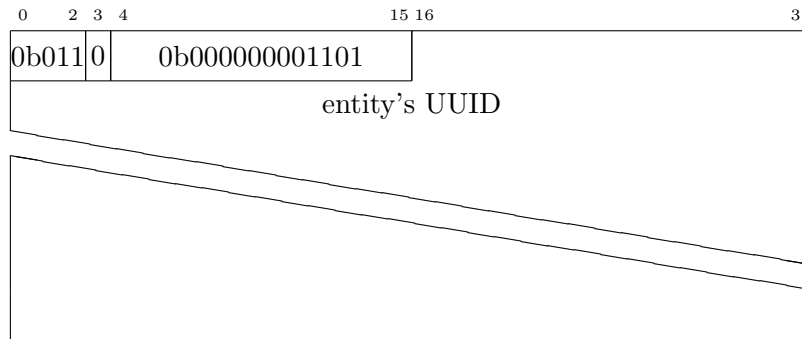
Figure 7.16: Place block petition structure

7.16 Change fly status operation

Only while in creative. Sets the fly status.

```
 0    2 3 4                    15 16          23 24            31
┌──────┬─┬──────────────────────┬──────────────┬───────────────┐
│0b011 │0│   0b000000001110      │   is flying  │░░░░░░░░░░░░░░░│
└──────┴─┴──────────────────────┴──────────────┴───────────────┘
```
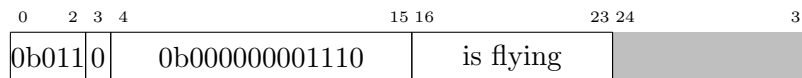
Figure 7.17: Change fly status petition structure

7.17 Start recording operation

Start to record the client's actions.

```
 0    2 3 4                          15
┌──────┬─┬────────────────────────────┐
│0b011 │0│     0b000000001111          │
└──────┴─┴────────────────────────────┘
```
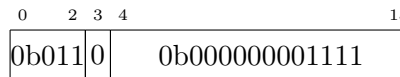
Figure 7.18: Start recording petition structure

7.18 Start recording response

As a response of the *Start recording operation*, you'll get the camera ID.

```
 0    2 3 4                    15 16                              31
┌──────┬─┬──────────────────────┬──────────────────────────────────┐
│0b011 │1│   0b000000001111      │           camera ID              │
└──────┴─┴──────────────────────┴──────────────────────────────────┘
```
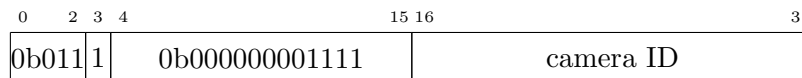
Figure 7.19: Start recording petition response structure

7.19 Stop recording operation

Stop recording the client's actions.

```
 0    2 3 4                    15
┌──────┬─┬──────────────────────┬──────────────────────────────────┐
│0b011 │0│   0b000000010000      │           camera ID              │
└──────┴─┴──────────────────────┴──────────────────────────────────┘
```
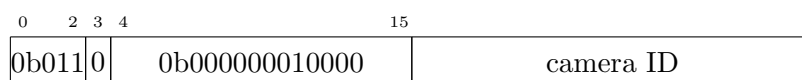
Figure 7.20: Stop recording petition structure

7.20 Stop recording response

As a response of the *Start recording operation*, you'll get the recording.

```
0    2 3  4                15 16                              31
0b011 1      0b000000010000
                        recording
```
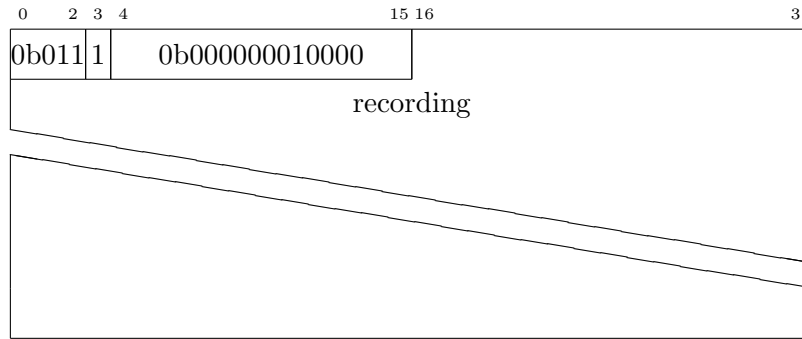
Figure 7.21: Stop recording petition response structure

7.21 Get version operation

Get the WatchWolf version of the client.

```
0    2 3  4                15
0b011 0      0b111111111111
```
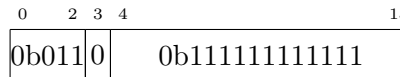
Figure 7.22: Get version operation structure

7.22 Get version response

The response of *Get version operation* will be the WatchWolf version.

```
0    2 3  4                15
0b011 1      0b111111111111
                     version
```
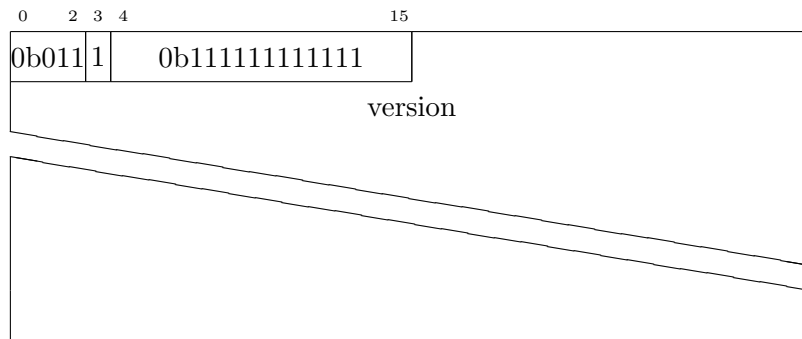
Figure 7.23: Get version response structure

8 Revision history

Date	Revision	WatchWolf version	Changes
18/03/2023	1	v0.1.15	Initial public release.
28/03/2023	2	v0.1.16	Updated API according to the beta feedback.
31/05/2023	3	v0.2.0	Added cameras, player's *change fly status* operation, (some of the) timings operations & WorldGuard operations.
03/06/2023	4	v0.2.0	Check version, specify world type, and set difficulty.

Table 8.1: Revision history

A Blocks

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

A.1 Unused material 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.

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
has_book	Inventory dependent
sculk_sensor_phase	Admin block

Modifier name	Reason for discarding
signal_fire	Bottom-block dependent
north=tall	Top-block dependent
south=tall	Top-block dependent
east=tall	Top-block dependent
west=tall	Top-block dependent
hatch	Unable to concatenate
thickness	Block dependent
snowy	Block dependent
in_wall	Same block
moisture	Block dependent

Table 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.

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
ANDESITE_WALL	up
BLACKSTONE_WALL	up
BRICK_WALL	up
COBBLED_DEEPSLATE_WALL	up
COBBLESTONE_WALL	up

Block name	Modifier name
DEEPSLATE_BRICK_WALL	up
DEEPSLATE_TILE_WALL	up
DIORITE_WALL	up
END_STONE_BRICK_WALL	up
GRANITE_WALL	up
MOSSY_COBBLESTONE_WALL	up
MOSSY_STONE_BRICK_WALL	up
MUD_BRICK_WALL	up
NETHER_BRICK_WALL	up
POLISHED_BLACKSTONE_BRICK_WALL	up
POLISHED_BLACKSTONE_WALL	up
POLISHED_DEEPSLATE_WALL	up
PRISMARINE_WALL	up
RED_NETHER_BRICK_WALL	up
RED_SANDSTONE_WALL	up
SANDSTONE_WALL	up
STONE_BRICK_WALL	up
ACACIA_DOOR	powered
ACACIA_FENCE_GATE	powered
ACACIA_TRAPDOOR	powered
ACTIVATOR_RAIL	powered
BELL	powered
BIRCH_DOOR	powered
BIRCH_FENCE_GATE	powered
BIRCH_TRAPDOOR	powered
CRIMSON_DOOR	powered
CRIMSON_FENCE_GATE	powered
CRIMSON_TRAPDOOR	powered
DARK_OAK_DOOR	powered
DARK_OAK_FENCE_GATE	powered

Block name	Modifier name
DARK_OAK_TRAPDOOR	powered
IRON_DOOR	powered
IRON_TRAPDOOR	powered
JUNGLE_DOOR	powered
JUNGLE_FENCE_GATE	powered
JUNGLE_TRAPDOOR	powered
LECTERN	powered
MANGROVE_DOOR	powered
MANGROVE_FENCE_GATE	powered
MANGROVE_TRAPDOOR	powered
NOTE_BLOCK	powered
OAK_DOOR	powered
OAK_FENCE_GATE	powered
OAK_TRAPDOOR	powered
POWERED_RAIL	powered
SPRUCE_DOOR	powered
SPRUCE_FENCE_GATE	powered
SPRUCE_TRAPDOOR	powered
TRIPWIRE	powered
WARPED_DOOR	powered
WARPED_FENCE_GATE	powered
WARPED_TRAPDOOR	powered
SMOKER	lit
FURNACE	lit
CHEST	type
TRAPPED_CHEST	type

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

A.2 Material modifiers

A.2.1 Age

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

Defaults to 0.

Material	Age range
BEETROOTS	0-3
BAMBOO	0-1
CARROTS	0-7
CHORUS_FLOWER	0-5
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

Table A.3: Ageable materials

A.2.2 Attachment

Denotes how the bell is attached to its block.

Defaults to floor.

Material	Options
BELL	ceiling/double_wall/floor/single_wall

Table A.4: Attachable materials

A.2.3 Axis

Represents the axis along whilst this block is oriented.

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

Material	Age range
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

Material	Age range
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_OAK_LOG	x/y/z
STRIPPED_DARK_OAK_WOOD	x/y/z
STRIPPED_JUNGLE_LOG	x/y/z
STRIPPED_JUNGLE_WOOD	x/y/z
STRIPPED_MANGROVE_LOG	x/y/z
STRIPPED_MANGROVE_WOOD	x/y/z
STRIPPED_OAK_LOG	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

Table A.5: Orientable materials

A.2.4 Berries

Indicates whether the block has berries.

Defaults to false.

Material	Values
CAVE_VINES	true/false
CAVE_VINES_PLANT	true/false

Table A.6: Materials with berries

A.2.5 Bites

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

Defaults to 0.

Material	Values
CAKE	0-6

Table A.7: Cake

A.2.6 Candles

Represents the number of candles which are present.

Defaults to 1.

Material	Values
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

Material	Values
PURPLE_CANDLE	1-4
RED_CANDLE	1-4
WHITE_CANDLE	1-4
YELLOW_CANDLE	1-4

Table A.8: Materials with candles

A.2.7 Charges

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

Defaults to 0.

Material	Values
RESPAWN_ANCHOR	0-4

Table A.9: Charged materials

A.2.8 Conditional

Denotes whether this command block is conditional or not.

Defaults to false.

Material	Values
CHAIN_COMMAND_BLOCK	true/false
COMMAND_BLOCK	true/false
REPEATING_COMMAND_BLOCK	true/false

Table A.10: Conditionable materials

A.2.9 Delay

Propagation delay of a repeater.

Defaults to 1.

Material	Values
REPEATER	1-4

Table A.11: Delayable materials

A.2.10 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.

Material	Values
CHORUS_PLANT	true/false
GLOW_LICHEN	true/false
SCULK_VEIN	true/false
BROWN_MUSHROOM_BLOCK	true/false
MUSHROOM_STEM	true/false
RED_MUSHROOM_BLOCK	true/false

Table A.12: Materials with down option

A.2.11 North, South, East and West

Set which faces of the block textures are displayed on.

As the *tall* option is unused (check Table A.1, Unused Spigot BlockData's modifiers), *none* and *low* will be considered as *false* and *true*, respectively.

Material	Options (default on bold)
ACACIA_FENCE	true/**false**
BIRCH_FENCE	true/**false**
BLACK_STAINED_GLASS_PANE	true/**false**
BLUE_STAINED_GLASS_PANE	true/**false**
BROWN_STAINED_GLASS_PANE	true/**false**
CHORUS_PLANT	true/**false**
CRIMSON_FENCE	true/**false**

Material	Options (default on bold)
CYAN_STAINED_GLASS_PANE	true/**false**
DARK_OAK_FENCE	true/**false**
FIRE	true/**false**
GLASS_PANE	true/**false**
GLOW_LICHEN	true/**false**
GRAY_STAINED_GLASS_PANE	true/**false**
GREEN_STAINED_GLASS_PANE	true/**false**
IRON_BARS	true/**false**
JUNGLE_FENCE	true/**false**
LIGHT_BLUE_STAINED_GLASS_PANE	true/**false**
LIGHT_GRAY_STAINED_GLASS_PANE	true/**false**
LIME_STAINED_GLASS_PANE	true/**false**
MAGENTA_STAINED_GLASS_PANE	true/**false**
MANGROVE_FENCE	true/**false**
NETHER_BRICK_FENCE	true/**false**
OAK_FENCE	true/**false**
ORANGE_STAINED_GLASS_PANE	true/**false**
PINK_STAINED_GLASS_PANE	true/**false**
PURPLE_STAINED_GLASS_PANE	true/**false**
RED_STAINED_GLASS_PANE	true/**false**
SCULK_VEIN	true/**false**
SPRUCE_FENCE	true/**false**
TRIPWIRE	true/**false**
VINE	true/**false**
WARPED_FENCE	true/**false**
WHITE_STAINED_GLASS_PANE	true/**false**
YELLOW_STAINED_GLASS_PANE	true/**false**
BROWN_MUSHROOM_BLOCK	**true**/false
MUSHROOM_STEM	**true**/false
RED_MUSHROOM_BLOCK	**true**/false

Material	Options (default on bold)
REDSTONE_WIRE	**none**/up/slide
ANDESITE_WALL	**none**/low/tall
BLACKSTONE_WALL	**none**/low/tall
BRICK_WALL	**none**/low/tall
COBBLED_DEEPSLATE_WALL	**none**/low/tall
COBBLESTONE_WALL	**none**/low/tall
DEEPSLATE_BRICK_WALL	**none**/low/tall
DEEPSLATE_TILE_WALL	**none**/low/tall
DIORITE_WALL	**none**/low/tall
END_STONE_BRICK_WALL	**none**/low/tall
GRANITE_WALL	**none**/low/tall
MOSSY_COBBLESTONE_WALL	**none**/low/tall
MOSSY_STONE_BRICK_WALL	**none**/low/tall
MUD_BRICK_WALL	**none**/low/tall
NETHER_BRICK_WALL	**none**/low/tall
POLISHED_BLACKSTONE_BRICK_WALL	**none**/low/tall
POLISHED_BLACKSTONE_WALL	**none**/low/tall
POLISHED_DEEPSLATE_WALL	**none**/low/tall
PRISMARINE_WALL	**none**/low/tall
RED_NETHER_BRICK_WALL	**none**/low/tall
RED_SANDSTONE_WALL	**none**/low/tall
SANDSTONE_WALL	**none**/low/tall
STONE_BRICK_WALL	**none**/low/tall

Table A.13: Orientable materials

A.2.12   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.

Material	Options
CHORUS_PLANT	true/false
FIRE	true/false
GLOW_LICHEN	true/false
SCULK_VEIN	true/false
VINE	true/false
BROWN_MUSHROOM_BLOCK	true/false
MUSHROOM_STEM	true/false
RED_MUSHROOM_BLOCK	true/false

Table A.14: Materials with up option

A.2.13 Eggs

Number of eggs which appear in the block.

Defaults to 1.

Material	Values
TURTLE_EGG	1-4

Table A.15: Materials with eggs

A.2.14 Extended

Denotes whether the piston head is currently extended or not.

Defaults to false.

Material	Values
PISTON	true/false
STICKY_PISTON	true/false

Table A.16: Extendable materials

A.2.15 Eye

Defaults to false.

Material	Values
END_PORTAL_FRAME	true/false

Table A.17: Materials with eye

A.2.16 Face

Represents the face to which a lever or button is stuck.

WALL_SIGN

Defaults to wall.

Material	Directions
ACACIA_BUTTON	wall/floor/ceiling
BIRCH_BUTTON	wall/floor/ceiling
CRIMSON_BUTTON	wall/floor/ceiling
DARK_OAK_BUTTON	wall/floor/ceiling
GRINDSTONE	wall/floor/ceiling
JUNGLE_BUTTON	wall/floor/ceiling
LEVER	wall/floor/ceiling
MANGROVE_BUTTON	wall/floor/ceiling
OAK_BUTTON	wall/floor/ceiling
POLISHED_BLACKSTONE_BUTTON	wall/floor/ceiling
SPRUCE_BUTTON	wall/floor/ceiling
STONE_BUTTON	wall/floor/ceiling
WARPED_BUTTON	wall/floor/ceiling

Table A.18: Directional materials

A.2.17 Facing

Represents the face towards which the block is pointing.

Material	Options (default on bold)
HOPPER	**down**/north/south/east/west

Material	Options (default on bold)
OBSERVER	up/down/north/**south**/east/west
BARREL	up/down/**north**/south/east/west
CHAIN_COMMAND_BLOCK	up/down/**north**/south/east/west
COMMAND_BLOCK	up/down/**north**/south/east/west
DISPENSER	up/down/**north**/south/east/west
DROPPER	up/down/**north**/south/east/west
PISTON	up/down/**north**/south/east/west
PISTON_HEAD	up/down/**north**/south/east/west
REPEATING_COMMAND_BLOCK	up/down/**north**/south/east/west
STICKY_PISTON	up/down/**north**/south/east/west
ACACIA_BUTTON	**north**/south/east/west
ACACIA_DOOR	**north**/south/east/west
ACACIA_FENCE_GATE	**north**/south/east/west
ACACIA_STAIRS	**north**/south/east/west
ACACIA_TRAPDOOR	**north**/south/east/west
ACACIA_WALL_SIGN	**north**/south/east/west
ANDESITE_STAIRS	**north**/south/east/west
ANVIL	**north**/south/east/west
ATTACHED_MELON_STEM	**north**/south/east/west
ATTACHED_PUMPKIN_STEM	**north**/south/east/west
BEEHIVE	**north**/south/east/west
BEE_NEST	**north**/south/east/west
BELL	**north**/south/east/west
BIG_DRIPLEAF	**north**/south/east/west
BIG_DRIPLEAF_STEM	**north**/south/east/west
BIRCH_BUTTON	**north**/south/east/west
BIRCH_DOOR	**north**/south/east/west
BIRCH_FENCE_GATE	**north**/south/east/west
BIRCH_STAIRS	**north**/south/east/west
BIRCH_TRAPDOOR	**north**/south/east/west

Material	Options (default on bold)
BIRCH_WALL_SIGN	**north**/south/east/west
BLACKSTONE_STAIRS	**north**/south/east/west
BLACK_BED	**north**/south/east/west
BLACK_GLAZED_TERRACOTTA	**north**/south/east/west
BLACK_WALL_BANNER	**north**/south/east/west
BLAST_FURNACE	**north**/south/east/west
BLUE_BED	**north**/south/east/west
BLUE_GLAZED_TERRACOTTA	**north**/south/east/west
BLUE_WALL_BANNER	**north**/south/east/west
BRAIN_CORAL_WALL_FAN	**north**/south/east/west
BRICK_STAIRS	**north**/south/east/west
BROWN_BED	**north**/south/east/west
BROWN_GLAZED_TERRACOTTA	**north**/south/east/west
BROWN_WALL_BANNER	**north**/south/east/west
BUBBLE_CORAL_WALL_FAN	**north**/south/east/west
CAMPFIRE	**north**/south/east/west
CARVED_PUMPKIN	**north**/south/east/west
CHEST	**north**/south/east/west
CHIPPED_ANVIL	**north**/south/east/west
COBBLED_DEEPSLATE_STAIRS	**north**/south/east/west
COBBLESTONE_STAIRS	**north**/south/east/west
COCOA	**north**/south/east/west
COMPARATOR	**north**/south/east/west
CREEPER_WALL_HEAD	**north**/south/east/west
CRIMSON_BUTTON	**north**/south/east/west
CRIMSON_DOOR	**north**/south/east/west
CRIMSON_FENCE_GATE	**north**/south/east/west
CRIMSON_STAIRS	**north**/south/east/west
CRIMSON_TRAPDOOR	**north**/south/east/west
CRIMSON_WALL_SIGN	**north**/south/east/west

Material	Options (default on bold)
CUT_COPPER_STAIRS	**north**/south/east/west
CYAN_BED	**north**/south/east/west
CYAN_GLAZED_TERRACOTTA	**north**/south/east/west
CYAN_WALL_BANNER	**north**/south/east/west
DAMAGED_ANVIL	**north**/south/east/west
DARK_OAK_BUTTON	**north**/south/east/west
DARK_OAK_DOOR	**north**/south/east/west
DARK_OAK_FENCE_GATE	**north**/south/east/west
DARK_OAK_STAIRS	**north**/south/east/west
DARK_OAK_TRAPDOOR	**north**/south/east/west
DARK_OAK_WALL_SIGN	**north**/south/east/west
DARK_PRISMARINE_STAIRS	**north**/south/east/west
DEAD_BRAIN_CORAL_WALL_FAN	**north**/south/east/west
DEAD_BUBBLE_CORAL_WALL_FAN	**north**/south/east/west
DEAD_FIRE_CORAL_WALL_FAN	**north**/south/east/west
DEAD_HORN_CORAL_WALL_FAN	**north**/south/east/west
DEAD_TUBE_CORAL_WALL_FAN	**north**/south/east/west
DEEPSLATE_BRICK_STAIRS	**north**/south/east/west
DEEPSLATE_TILE_STAIRS	**north**/south/east/west
DIORITE_STAIRS	**north**/south/east/west
DRAGON_WALL_HEAD	**north**/south/east/west
ENDER_CHEST	**north**/south/east/west
END_PORTAL_FRAME	**north**/south/east/west
END_STONE_BRICK_STAIRS	**north**/south/east/west
EXPOSED_CUT_COPPER_STAIRS	**north**/south/east/west
FIRE_CORAL_WALL_FAN	**north**/south/east/west
FURNACE	**north**/south/east/west
GRANITE_STAIRS	**north**/south/east/west
GRAY_BED	**north**/south/east/west
GRAY_GLAZED_TERRACOTTA	**north**/south/east/west

Material	Options (default on bold)
GRAY_WALL_BANNER	**north**/south/east/west
GREEN_BED	**north**/south/east/west
GREEN_GLAZED_TERRACOTTA	**north**/south/east/west
GREEN_WALL_BANNER	**north**/south/east/west
GRINDSTONE	**north**/south/east/west
HORN_CORAL_WALL_FAN	**north**/south/east/west
IRON_DOOR	**north**/south/east/west
IRON_TRAPDOOR	**north**/south/east/west
JACK_O_LANTERN	**north**/south/east/west
JUNGLE_BUTTON	**north**/south/east/west
JUNGLE_DOOR	**north**/south/east/west
JUNGLE_FENCE_GATE	**north**/south/east/west
JUNGLE_STAIRS	**north**/south/east/west
JUNGLE_TRAPDOOR	**north**/south/east/west
JUNGLE_WALL_SIGN	**north**/south/east/west
LADDER	**north**/south/east/west
LECTERN	**north**/south/east/west
LEVER	**north**/south/east/west
LIGHT_BLUE_BED	**north**/south/east/west
LIGHT_BLUE_GLAZED_TERRACOTTA	**north**/south/east/west
LIGHT_BLUE_WALL_BANNER	**north**/south/east/west
LIGHT_GRAY_BED	**north**/south/east/west
LIGHT_GRAY_GLAZED_TERRACOTTA	**north**/south/east/west
LIGHT_GRAY_WALL_BANNER	**north**/south/east/west
LIME_BED	**north**/south/east/west
LIME_GLAZED_TERRACOTTA	**north**/south/east/west
LIME_WALL_BANNER	**north**/south/east/west
LOOM	**north**/south/east/west
MAGENTA_BED	**north**/south/east/west
MAGENTA_GLAZED_TERRACOTTA	**north**/south/east/west

Material	Options (default on bold)
MAGENTA_WALL_BANNER	**north**/south/east/west
MANGROVE_BUTTON	**north**/south/east/west
MANGROVE_DOOR	**north**/south/east/west
MANGROVE_FENCE_GATE	**north**/south/east/west
MANGROVE_STAIRS	**north**/south/east/west
MANGROVE_TRAPDOOR	**north**/south/east/west
MANGROVE_WALL_SIGN	**north**/south/east/west
MOSSY_COBBLESTONE_STAIRS	**north**/south/east/west
MOSSY_STONE_BRICK_STAIRS	**north**/south/east/west
MUD_BRICK_STAIRS	**north**/south/east/west
NETHER_BRICK_STAIRS	**north**/south/east/west
OAK_BUTTON	**north**/south/east/west
OAK_DOOR	**north**/south/east/west
OAK_FENCE_GATE	**north**/south/east/west
OAK_STAIRS	**north**/south/east/west
OAK_TRAPDOOR	**north**/south/east/west
OAK_WALL_SIGN	**north**/south/east/west
ORANGE_BED	**north**/south/east/west
ORANGE_GLAZED_TERRACOTTA	**north**/south/east/west
ORANGE_WALL_BANNER	**north**/south/east/west
OXIDIZED_CUT_COPPER_STAIRS	**north**/south/east/west
PINK_BED	**north**/south/east/west
PINK_GLAZED_TERRACOTTA	**north**/south/east/west
PINK_WALL_BANNER	**north**/south/east/west
PLAYER_WALL_HEAD	**north**/south/east/west
POLISHED_ANDESITE_STAIRS	**north**/south/east/west
POLISHED_BLACKSTONE_BRICK_STAIRS	**north**/south/east/west
POLISHED_BLACKSTONE_BUTTON	**north**/south/east/west
POLISHED_BLACKSTONE_STAIRS	**north**/south/east/west
POLISHED_DEEPSLATE_STAIRS	**north**/south/east/west

Material	Options (default on bold)
POLISHED_DIORITE_STAIRS	**north**/south/east/west
POLISHED_GRANITE_STAIRS	**north**/south/east/west
PRISMARINE_BRICK_STAIRS	**north**/south/east/west
PRISMARINE_STAIRS	**north**/south/east/west
PURPLE_BED	**north**/south/east/west
PURPLE_GLAZED_TERRACOTTA	**north**/south/east/west
PURPLE_WALL_BANNER	**north**/south/east/west
PURPUR_STAIRS	**north**/south/east/west
QUARTZ_STAIRS	**north**/south/east/west
REDSTONE_WALL_TORCH	**north**/south/east/west
RED_BED	**north**/south/east/west
RED_GLAZED_TERRACOTTA	**north**/south/east/west
RED_NETHER_BRICK_STAIRS	**north**/south/east/west
RED_SANDSTONE_STAIRS	**north**/south/east/west
RED_WALL_BANNER	**north**/south/east/west
REPEATER	**north**/south/east/west
SANDSTONE_STAIRS	**north**/south/east/west
SKELETON_WALL_SKULL	**north**/south/east/west
SMALL_DRIPLEAF	**north**/south/east/west
SMOKER	**north**/south/east/west
SMOOTH_QUARTZ_STAIRS	**north**/south/east/west
SMOOTH_RED_SANDSTONE_STAIRS	**north**/south/east/west
SMOOTH_SANDSTONE_STAIRS	**north**/south/east/west
SOUL_CAMPFIRE	**north**/south/east/west
SOUL_WALL_TORCH	**north**/south/east/west
SPRUCE_BUTTON	**north**/south/east/west
SPRUCE_DOOR	**north**/south/east/west
SPRUCE_FENCE_GATE	**north**/south/east/west
SPRUCE_STAIRS	**north**/south/east/west
SPRUCE_TRAPDOOR	**north**/south/east/west

Material	Options (default on bold)
SPRUCE_WALL_SIGN	**north**/south/east/west
STONECUTTER	**north**/south/east/west
STONE_BRICK_STAIRS	**north**/south/east/west
STONE_BUTTON	**north**/south/east/west
STONE_STAIRS	**north**/south/east/west
TRAPPED_CHEST	**north**/south/east/west
TRIPWIRE_HOOK	**north**/south/east/west
TUBE_CORAL_WALL_FAN	**north**/south/east/west
WALL_TORCH	**north**/south/east/west
WARPED_BUTTON	**north**/south/east/west
WARPED_DOOR	**north**/south/east/west
WARPED_FENCE_GATE	**north**/south/east/west
WARPED_STAIRS	**north**/south/east/west
WARPED_TRAPDOOR	**north**/south/east/west
WARPED_WALL_SIGN	**north**/south/east/west
WAXED_CUT_COPPER_STAIRS	**north**/south/east/west
WAXED_EXPOSED_CUT_COPPER_STAIRS	**north**/south/east/west
WAXED_OXIDIZED_CUT_COPPER_STAIRS	**north**/south/east/west
WAXED_WEATHERED_CUT_COPPER_STAIRS	**north**/south/east/west
WEATHERED_CUT_COPPER_STAIRS	**north**/south/east/west
WHITE_BED	**north**/south/east/west
WHITE_GLAZED_TERRACOTTA	**north**/south/east/west
WHITE_WALL_BANNER	**north**/south/east/west
WITHER_SKELETON_WALL_SKULL	**north**/south/east/west
YELLOW_BED	**north**/south/east/west
YELLOW_GLAZED_TERRACOTTA	**north**/south/east/west
YELLOW_WALL_BANNER	**north**/south/east/west
ZOMBIE_WALL_HEAD	**north**/south/east/west
AMETHYST_CLUSTER	**up**/down/north/south/east/west
BLACK_SHULKER_BOX	**up**/down/north/south/east/west

Material	Options (default on bold)
BLUE_SHULKER_BOX	**up**/down/north/south/east/west
BROWN_SHULKER_BOX	**up**/down/north/south/east/west
CYAN_SHULKER_BOX	**up**/down/north/south/east/west
END_ROD	**up**/down/north/south/east/west
GRAY_SHULKER_BOX	**up**/down/north/south/east/west
GREEN_SHULKER_BOX	**up**/down/north/south/east/west
LARGE_AMETHYST_BUD	**up**/down/north/south/east/west
LIGHTNING_ROD	**up**/down/north/south/east/west
LIGHT_BLUE_SHULKER_BOX	**up**/down/north/south/east/west
LIGHT_GRAY_SHULKER_BOX	**up**/down/north/south/east/west
LIME_SHULKER_BOX	**up**/down/north/south/east/west
MAGENTA_SHULKER_BOX	**up**/down/north/south/east/west
MEDIUM_AMETHYST_BUD	**up**/down/north/south/east/west
ORANGE_SHULKER_BOX	**up**/down/north/south/east/west
PINK_SHULKER_BOX	**up**/down/north/south/east/west
PURPLE_SHULKER_BOX	**up**/down/north/south/east/west
RED_SHULKER_BOX	**up**/down/north/south/east/west
SHULKER_BOX	**up**/down/north/south/east/west
SMALL_AMETHYST_BUD	**up**/down/north/south/east/west
WHITE_SHULKER_BOX	**up**/down/north/south/east/west
YELLOW_SHULKER_BOX	**up**/down/north/south/east/west

Table A.19: Directional materials

A.2.18   Half

Denotes which half of a two block tall material this block is.

Material	Options (default on bold)
ACACIA_STAIRS	**bottom**/top
ACACIA_TRAPDOOR	**bottom**/top

Material	Options (default on bold)
ANDESITE_STAIRS	**bottom**/top
BIRCH_STAIRS	**bottom**/top
BIRCH_TRAPDOOR	**bottom**/top
BLACKSTONE_STAIRS	**bottom**/top
BRICK_STAIRS	**bottom**/top
COBBLED_DEEPSLATE_STAIRS	**bottom**/top
COBBLESTONE_STAIRS	**bottom**/top
CRIMSON_STAIRS	**bottom**/top
CRIMSON_TRAPDOOR	**bottom**/top
CUT_COPPER_STAIRS	**bottom**/top
DARK_OAK_STAIRS	**bottom**/top
DARK_OAK_TRAPDOOR	**bottom**/top
DARK_PRISMARINE_STAIRS	**bottom**/top
DEEPSLATE_BRICK_STAIRS	**bottom**/top
DEEPSLATE_TILE_STAIRS	**bottom**/top
DIORITE_STAIRS	**bottom**/top
END_STONE_BRICK_STAIRS	**bottom**/top
EXPOSED_CUT_COPPER_STAIRS	**bottom**/top
GRANITE_STAIRS	**bottom**/top
IRON_TRAPDOOR	**bottom**/top
JUNGLE_STAIRS	**bottom**/top
JUNGLE_TRAPDOOR	**bottom**/top
MANGROVE_STAIRS	**bottom**/top
MANGROVE_TRAPDOOR	**bottom**/top
MOSSY_COBBLESTONE_STAIRS	**bottom**/top
MOSSY_STONE_BRICK_STAIRS	**bottom**/top
MUD_BRICK_STAIRS	**bottom**/top
NETHER_BRICK_STAIRS	**bottom**/top
OAK_STAIRS	**bottom**/top
OAK_TRAPDOOR	**bottom**/top

Material	Options (default on bold)
OXIDIZED_CUT_COPPER_STAIRS	**bottom**/top
POLISHED_ANDESITE_STAIRS	**bottom**/top
POLISHED_BLACKSTONE_BRICK_STAIRS	**bottom**/top
POLISHED_BLACKSTONE_STAIRS	**bottom**/top
POLISHED_DEEPSLATE_STAIRS	**bottom**/top
POLISHED_DIORITE_STAIRS	**bottom**/top
POLISHED_GRANITE_STAIRS	**bottom**/top
PRISMARINE_BRICK_STAIRS	**bottom**/top
PRISMARINE_STAIRS	**bottom**/top
PURPUR_STAIRS	**bottom**/top
QUARTZ_STAIRS	**bottom**/top
RED_NETHER_BRICK_STAIRS	**bottom**/top
RED_SANDSTONE_STAIRS	**bottom**/top
SANDSTONE_STAIRS	**bottom**/top
SMOOTH_QUARTZ_STAIRS	**bottom**/top
SMOOTH_RED_SANDSTONE_STAIRS	**bottom**/top
SMOOTH_SANDSTONE_STAIRS	**bottom**/top
SPRUCE_STAIRS	**bottom**/top
SPRUCE_TRAPDOOR	**bottom**/top
STONE_BRICK_STAIRS	**bottom**/top
STONE_STAIRS	**bottom**/top
WARPED_STAIRS	**bottom**/top
WARPED_TRAPDOOR	**bottom**/top
WAXED_CUT_COPPER_STAIRS	**bottom**/top
WAXED_EXPOSED_CUT_COPPER_STAIRS	**bottom**/top
WAXED_OXIDIZED_CUT_COPPER_STAIRS	**bottom**/top
WAXED_WEATHERED_CUT_COPPER_STAIRS	**bottom**/top
WEATHERED_CUT_COPPER_STAIRS	**bottom**/top
ACACIA_DOOR	**lower**/upper
BIRCH_DOOR	**lower**/upper

Material	Options (default on bold)
CRIMSON_DOOR	**lower**/upper
DARK_OAK_DOOR	**lower**/upper
IRON_DOOR	**lower**/upper
JUNGLE_DOOR	**lower**/upper
LARGE_FERN	**lower**/upper
LILAC	**lower**/upper
MANGROVE_DOOR	**lower**/upper
OAK_DOOR	**lower**/upper
PEONY	**lower**/upper
ROSE_BUSH	**lower**/upper
SMALL_DRIPLEAF	**lower**/upper
SPRUCE_DOOR	**lower**/upper
SUNFLOWER	**lower**/upper
TALL_GRASS	**lower**/upper
TALL_SEAGRASS	**lower**/upper
WARPED_DOOR	**lower**/upper

Table A.20: Two-blocks materials

A.2.19 Hanging

Denotes whether the block is hanging.

Defaults to false.

Material	Values
LANTERN	true/false
MANGROVE_PROPAGULE	true/false
SOUL_LANTERN	true/false

Table A.21: Hangable materials

A.2.20   Hinge

Indicates which hinge this door is attached to and will rotate around when opened.

Defaults to left.

Material	Options
ACACIA_DOOR	left/right
BIRCH_DOOR	left/right
CRIMSON_DOOR	left/right
DARK_OAK_DOOR	left/right
IRON_DOOR	left/right
JUNGLE_DOOR	left/right
MANGROVE_DOOR	left/right
OAK_DOOR	left/right
SPRUCE_DOOR	left/right
WARPED_DOOR	left/right

Table A.22: Doors

A.2.21   Honey level

Represents the amount of honey stored in the hive.

Defaults to 0.

Material	Values
BEEHIVE	$0/5^{20}$
BEE_NEST	$0/5^{20}$

Table A.23: Hives

A.2.22   Inverted

Denotes whether this daylight detector is in the inverted mode.

---

[20]The block is the same from honey-level 0 to 4, and it changes in age 5. That's why age=5 is considered as age=1, and age=0-4 as age=0, as you may notice in Figure ??, Modifier concatenation.

Defaults to false.

Material	Values
DAYLIGHT_DETECTOR	true/false

Table A.24: Invertible blocks

A.2.23 Layers

Represents the amount of layers of snow which are present in this block.

Defaults to 1.

Material	Values
SNOW	1-8

Table A.25: Snow

A.2.24 Leaves

Represents the size of the leaves of the bamboo block.

Defaults to none.

Material	Values
BAMBOO	none/small/large

Table A.26: Bamboo

A.2.25 Level

Represents the amount of fluid contained within this block, either by itself or inside a cauldron.

Defaults on 0 except for POWDER_SNOW_CAULDRON and WATER_CAULDRON (which defaults on 1).

Material	Values
COMPOSTER	0-8
LAVA	0-7
WATER	0-7
POWDER_SNOW_CAULDRON	1-3
WATER_CAULDRON	1-3

Table A.27: Levelled materials

A.2.26 Lit

Denotes whether this block is currently lit.

Except for CAMPFIRE, REDSTONE_TORCH, REDSTONE_WALL_TORCH and SOUL_CAMPFIRE (which defaults to true), it defaults to false.

Material	Options
BLACK_CANDLE	true/false
BLACK_CANDLE_CAKE	true/false
BLAST_FURNACE	true/false
BLUE_CANDLE	true/false
BLUE_CANDLE_CAKE	true/false
BROWN_CANDLE	true/false
BROWN_CANDLE_CAKE	true/false
CANDLE	true/false
CANDLE_CAKE	true/false
CYAN_CANDLE	true/false
CYAN_CANDLE_CAKE	true/false
DEEPSLATE_REDSTONE_ORE	true/false
GRAY_CANDLE	true/false
GRAY_CANDLE_CAKE	true/false
GREEN_CANDLE	true/false
GREEN_CANDLE_CAKE	true/false
LIGHT_BLUE_CANDLE	true/false

Material	Options
LIGHT_BLUE_CANDLE_CAKE	true/false
LIGHT_GRAY_CANDLE	true/false
LIGHT_GRAY_CANDLE_CAKE	true/false
LIME_CANDLE	true/false
LIME_CANDLE_CAKE	true/false
MAGENTA_CANDLE	true/false
MAGENTA_CANDLE_CAKE	true/false
ORANGE_CANDLE	true/false
ORANGE_CANDLE_CAKE	true/false
PINK_CANDLE	true/false
PINK_CANDLE_CAKE	true/false
PURPLE_CANDLE	true/false
PURPLE_CANDLE_CAKE	true/false
REDSTONE_LAMP	true/false
REDSTONE_ORE	true/false
RED_CANDLE	true/false
RED_CANDLE_CAKE	true/false
WHITE_CANDLE	true/false
WHITE_CANDLE_CAKE	true/false
YELLOW_CANDLE	true/false
YELLOW_CANDLE_CAKE	true/false
CAMPFIRE	true/false
REDSTONE_TORCH	true/false
REDSTONE_WALL_TORCH	true/false
SOUL_CAMPFIRE	true/false

Table A.28: Lightable materials

A.2.27 Locked

Denotes whether the repeater is in the locked state or not.

Defaults to false.

Material	Values
REPEATER	true/false

Table A.29: Repeater

A.2.28 Mode

Indicates what mode the block will operate in.

Material	Options (default on bold)
COMPARATOR	**compare**/subtract
STRUCTURE_BLOCK	**load**/corner/save

Table A.30: Materials with mode

A.2.29 Note

Specified tuned pitch that the instrument will be played in.

Defaults to 0.

Material	Values
NOTE_BLOCK	0-24

Table A.31: Note block

A.2.30 Open

Denotes whether this block is currently opened.

Defaults to false.

Material	Values
ACACIA_DOOR	true/false
ACACIA_FENCE_GATE	true/false

Material	Values
ACACIA_TRAPDOOR	true/false
BARREL	true/false
BIRCH_DOOR	true/false
BIRCH_FENCE_GATE	true/false
BIRCH_TRAPDOOR	true/false
CRIMSON_DOOR	true/false
CRIMSON_FENCE_GATE	true/false
CRIMSON_TRAPDOOR	true/false
DARK_OAK_DOOR	true/false
DARK_OAK_FENCE_GATE	true/false
DARK_OAK_TRAPDOOR	true/false
IRON_DOOR	true/false
IRON_TRAPDOOR	true/false
JUNGLE_DOOR	true/false
JUNGLE_FENCE_GATE	true/false
JUNGLE_TRAPDOOR	true/false
MANGROVE_DOOR	true/false
MANGROVE_FENCE_GATE	true/false
MANGROVE_TRAPDOOR	true/false
OAK_DOOR	true/false
OAK_FENCE_GATE	true/false
OAK_TRAPDOOR	true/false
SPRUCE_DOOR	true/false
SPRUCE_FENCE_GATE	true/false
SPRUCE_TRAPDOOR	true/false
WARPED_DOOR	true/false
WARPED_FENCE_GATE	true/false
WARPED_TRAPDOOR	true/false

Table A.32: Openable materials

A.2.31   Orientation

Direction the block is facing.

Defaults to north_up, and can take any of the values shown in Table A.33, Jigsaw orientations.

Orientation
north_up
south_up
east_up
west_up
up_north
up_south
up_east
up_west
down_north
down_south
down_east
down_west

Table A.33: Jigsaw orientations

Material
JIGSAW

Table A.34: Jigsaw

A.2.32   Part

Denotes which half of the bed this block corresponds to.

Defaults to foot.

Material	Values
BLACK_BED	foot/head

Material	Values
BLUE_BED	foot/head
BROWN_BED	foot/head
CYAN_BED	foot/head
GRAY_BED	foot/head
GREEN_BED	foot/head
LIGHT_BLUE_BED	foot/head
LIGHT_GRAY_BED	foot/head
LIME_BED	foot/head
MAGENTA_BED	foot/head
ORANGE_BED	foot/head
PINK_BED	foot/head
PURPLE_BED	foot/head
RED_BED	foot/head
WHITE_BED	foot/head
YELLOW_BED	foot/head

Table A.35: Openable materials

A.2.33   Pickles

Indicates the number of pickles in this block.

Defaults to 1.

Material	Values
SEA_PICKLE	1-4

Table A.36: Materials with pickles

A.2.34   Powered

Indicates whether this block is in the powered state or not (emitting current).

Defaults to false.

175

Material	Powered value
ACACIA_BUTTON	true/false
ACACIA_PRESSURE_PLATE	true/false
BIRCH_BUTTON	true/false
BIRCH_PRESSURE_PLATE	true/false
COMPARATOR	true/false
CRIMSON_BUTTON	true/false
CRIMSON_PRESSURE_PLATE	true/false
DARK_OAK_BUTTON	true/false
DARK_OAK_PRESSURE_PLATE	true/false
DETECTOR_RAIL	true/false
JUNGLE_BUTTON	true/false
JUNGLE_PRESSURE_PLATE	true/false
LEVER	true/false
LIGHTNING_ROD	true/false
MANGROVE_BUTTON	true/false
MANGROVE_PRESSURE_PLATE	true/false
OAK_BUTTON	true/false
OAK_PRESSURE_PLATE	true/false
OBSERVER	true/false
POLISHED_BLACKSTONE_BUTTON	true/false
POLISHED_BLACKSTONE_PRESSURE_PLATE	true/false
REPEATER	true/false
SPRUCE_BUTTON	true/false
SPRUCE_PRESSURE_PLATE	true/false
STONE_BUTTON	true/false
STONE_PRESSURE_PLATE	true/false
TRIPWIRE_HOOK	true/false
WARPED_BUTTON	true/false
WARPED_PRESSURE_PLATE	true/false

Material	Powered value

Table A.37: Powerabled materials

A.2.35 Rotation

Denotes where the block is looking.

Defaults to 0 and goes up to 15.

Rotation value	Direction
0	South
4	West
8	North
12	East

Table A.38: Relation between rotation and where is looking

Material
ACACIA_SIGN
BIRCH_SIGN
BLACK_BANNER
BLUE_BANNER
BROWN_BANNER
CREEPER_HEAD
CRIMSON_SIGN
CYAN_BANNER
DARK_OAK_SIGN
DRAGON_HEAD
GRAY_BANNER
GREEN_BANNER
JUNGLE_SIGN
LIGHT_BLUE_BANNER
LIGHT_GRAY_BANNER

Material
LIME_BANNER
MAGENTA_BANNER
MANGROVE_SIGN
OAK_SIGN
ORANGE_BANNER
PINK_BANNER
PLAYER_HEAD
PURPLE_BANNER
RED_BANNER
SKELETON_SKULL
SPRUCE_SIGN
WARPED_SIGN
WHITE_BANNER
WITHER_SKELETON_SKULL
YELLOW_BANNER
ZOMBIE_HEAD

Table A.39: Directional materials

A.2.36   Shape - Rails

Represents the current layout of a minecart rail.

Defaults to north_south, and it can take any value of those mentioned in Table A.40, Rail directions.

Shape
north_south
east_west
north_east
north_west
south_east

Shape
south_west
ascending_north
ascending_south
ascending_east
ascending_west

Table A.40: Rail directions

Material
ACTIVATOR_RAIL
DETECTOR_RAIL
POWERED_RAIL
RAIL

Table A.41: Rails

A.2.37   Shape - Stairs

Represents the texture and bounding box shape of these stairs.

Defaults to straight.

Material	Shape
ACACIA_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
ANDESITE_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
BIRCH_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
BLACKSTONE_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
BRICK_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right

Material	Shape
COBBLED_DEEPSLATE_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
COBBLESTONE_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
CRIMSON_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
CUT_COPPER_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
DARK_OAK_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
DARK_PRISMARINE_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
DEEPSLATE_BRICK_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
DEEPSLATE_TILE_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
DIORITE_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
END_STONE_BRICK_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
EXPOSED_CUT_COPPER_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
GRANITE_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
JUNGLE_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
MANGROVE_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
MOSSY_COBBLESTONE_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right

Material	Shape
MOSSY_STONE_BRICK_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
MUD_BRICK_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
NETHER_BRICK_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
OAK_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
OXIDIZED_CUT_COPPER_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
POLISHED_ANDESITE_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
POLISHED_BLACKSTONE_BRICK_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
POLISHED_BLACKSTONE_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
POLISHED_DEEPSLATE_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
POLISHED_DIORITE_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
POLISHED_GRANITE_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
PRISMARINE_BRICK_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
PRISMARINE_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
PURPUR_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
QUARTZ_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right

Material	Shape
RED_NETHER_BRICK_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
RED_SANDSTONE_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
SANDSTONE_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
SMOOTH_QUARTZ_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
SMOOTH_RED_SANDSTONE_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
SMOOTH_SANDSTONE_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
SPRUCE_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
STONE_BRICK_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
STONE_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
WARPED_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
WAXED_CUT_COPPER_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
WAXED_EXPOSED_CUT_COPPER_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
WAXED_OXIDIZED_CUT_COPPER_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
WAXED_WEATHERED_CUT_COPPER_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right
WEATHERED_CUT_COPPER_STAIRS	straight/inner_right/inner_left/ outer_right/inner_right

Material	Shape

Table A.42: Stairs

A.2.38 Type

Represents the type of piston which that block corresponds to, or represents what state the slab is in.

Material	Options (default on bold)
ACACIA_SLAB	**bottom**/top/double
ANDESITE_SLAB	**bottom**/top/double
BIRCH_SLAB	**bottom**/top/double
BLACKSTONE_SLAB	**bottom**/top/double
BRICK_SLAB	**bottom**/top/double
COBBLED_DEEPSLATE_SLAB	**bottom**/top/double
COBBLESTONE_SLAB	**bottom**/top/double
CRIMSON_SLAB	**bottom**/top/double
CUT_COPPER_SLAB	**bottom**/top/double
CUT_RED_SANDSTONE_SLAB	**bottom**/top/double
CUT_SANDSTONE_SLAB	**bottom**/top/double
DARK_OAK_SLAB	**bottom**/top/double
DARK_PRISMARINE_SLAB	**bottom**/top/double
DEEPSLATE_BRICK_SLAB	**bottom**/top/double
DEEPSLATE_TILE_SLAB	**bottom**/top/double
DIORITE_SLAB	**bottom**/top/double
END_STONE_BRICK_SLAB	**bottom**/top/double
EXPOSED_CUT_COPPER_SLAB	**bottom**/top/double
GRANITE_SLAB	**bottom**/top/double
JUNGLE_SLAB	**bottom**/top/double
MANGROVE_SLAB	**bottom**/top/double
MOSSY_COBBLESTONE_SLAB	**bottom**/top/double

Material	Options (default on bold)
MOSSY_STONE_BRICK_SLAB	**bottom**/top/double
MUD_BRICK_SLAB	**bottom**/top/double
NETHER_BRICK_SLAB	**bottom**/top/double
OAK_SLAB	**bottom**/top/double
OXIDIZED_CUT_COPPER_SLAB	**bottom**/top/double
PETRIFIED_OAK_SLAB	**bottom**/top/double
POLISHED_ANDESITE_SLAB	**bottom**/top/double
POLISHED_BLACKSTONE_BRICK_SLAB	**bottom**/top/double
POLISHED_BLACKSTONE_SLAB	**bottom**/top/double
POLISHED_DEEPSLATE_SLAB	**bottom**/top/double
POLISHED_DIORITE_SLAB	**bottom**/top/double
POLISHED_GRANITE_SLAB	**bottom**/top/double
PRISMARINE_BRICK_SLAB	**bottom**/top/double
PRISMARINE_SLAB	**bottom**/top/double
PURPUR_SLAB	**bottom**/top/double
QUARTZ_SLAB	**bottom**/top/double
RED_NETHER_BRICK_SLAB	**bottom**/top/double
RED_SANDSTONE_SLAB	**bottom**/top/double
SANDSTONE_SLAB	**bottom**/top/double
SMOOTH_QUARTZ_SLAB	**bottom**/top/double
SMOOTH_RED_SANDSTONE_SLAB	**bottom**/top/double
SMOOTH_SANDSTONE_SLAB	**bottom**/top/double
SMOOTH_STONE_SLAB	**bottom**/top/double
SPRUCE_SLAB	**bottom**/top/double
STONE_BRICK_SLAB	**bottom**/top/double
STONE_SLAB	**bottom**/top/double
WARPED_SLAB	**bottom**/top/double
WAXED_CUT_COPPER_SLAB	**bottom**/top/double
WAXED_EXPOSED_CUT_COPPER_SLAB	**bottom**/top/double
WAXED_OXIDIZED_CUT_COPPER_SLAB	**bottom**/top/double

Material	Options (default on bold)
WAXED_WEATHERED_CUT_COPPER_SLAB	**bottom**/top/double
WEATHERED_CUT_COPPER_SLAB	**bottom**/top/double
PISTON_HEAD	**normal**/sticky

Table A.43: Type materials

A.2.39 Vertical direction

Represents the dripstone orientation.

Defaults to up.

Material	Values
POINTED_DRIPSTONE	up/down

Table A.44: Dripstone

A.2.40 Waterlogged

Denotes whether this block has fluid in it.

Besides underwater blocks[21] (which defaults to true), it defaults to false. All the possible options are true or false.

Material	Aquatic block[22]
ACACIA_FENCE	✗
ACACIA_LEAVES	✗

---

[21]BRAIN_CORAL, BRAIN_CORAL_FAN, BRAIN_CORAL_WALL_FAN, BUBBLE_CORAL, BUBBLE_CORAL_FAN, BUBBLE_CORAL_WALL_FAN, CONDUIT, DEAD_BRAIN_CORAL, DEAD_BRAIN_CORAL_FAN, DEAD_BRAIN_CORAL_WALL_FAN, DEAD_BUBBLE_CORAL, DEAD_BUBBLE_CORAL_FAN, DEAD_BUBBLE_CORAL_WALL_FAN, DEAD_FIRE_CORAL, DEAD_FIRE_CORAL_FAN, DEAD_FIRE_CORAL_WALL_FAN, DEAD_HORN_CORAL, DEAD_HORN_CORAL_FAN, DEAD_HORN_CORAL_WALL_FAN, DEAD_TUBE_CORAL, DEAD_TUBE_CORAL_FAN, DEAD_TUBE_CORAL_WALL_FAN, FIRE_CORAL, FIRE_CORAL_FAN, FIRE_CORAL_WALL_FAN, HORN_CORAL, HORN_CORAL_FAN, HORN_CORAL_WALL_FAN, SEA_PICKLE, TUBE_CORAL, TUBE_CORAL_FAN and TUBE_CORAL_WALL_FAN

Material	Aquatic block[22]
ACACIA_SIGN	✗
ACACIA_SLAB	✗
ACACIA_STAIRS	✗
ACACIA_TRAPDOOR	✗
ACACIA_WALL_SIGN	✗
ACTIVATOR_RAIL	✗
AMETHYST_CLUSTER	✗
ANDESITE_SLAB	✗
ANDESITE_STAIRS	✗
ANDESITE_WALL	✗
AZALEA_LEAVES	✗
BIG_DRIPLEAF	✗
BIG_DRIPLEAF_STEM	✗
BIRCH_FENCE	✗
BIRCH_LEAVES	✗
BIRCH_SIGN	✗
BIRCH_SLAB	✗
BIRCH_STAIRS	✗
BIRCH_TRAPDOOR	✗
BIRCH_WALL_SIGN	✗
BLACKSTONE_SLAB	✗
BLACKSTONE_STAIRS	✗
BLACKSTONE_WALL	✗
BLACK_CANDLE	✗
BLACK_STAINED_GLASS_PANE	✗
BLUE_CANDLE	✗
BLUE_STAINED_GLASS_PANE	✗
BRICK_SLAB	✗
BRICK_STAIRS	✗
BRICK_WALL	✗

Material	Aquatic block[22]
BROWN_CANDLE	✗
BROWN_STAINED_GLASS_PANE	✗
CAMPFIRE	✗
CANDLE	✗
CHAIN	✗
CHEST	✗
COBBLED_DEEPSLATE_SLAB	✗
COBBLED_DEEPSLATE_STAIRS	✗
COBBLED_DEEPSLATE_WALL	✗
COBBLESTONE_SLAB	✗
COBBLESTONE_STAIRS	✗
COBBLESTONE_WALL	✗
CRIMSON_FENCE	✗
CRIMSON_SIGN	✗
CRIMSON_SLAB	✗
CRIMSON_STAIRS	✗
CRIMSON_TRAPDOOR	✗
CRIMSON_WALL_SIGN	✗
CUT_COPPER_SLAB	✗
CUT_COPPER_STAIRS	✗
CUT_RED_SANDSTONE_SLAB	✗
CUT_SANDSTONE_SLAB	✗
CYAN_CANDLE	✗
CYAN_STAINED_GLASS_PANE	✗
DARK_OAK_FENCE	✗
DARK_OAK_LEAVES	✗
DARK_OAK_SIGN	✗
DARK_OAK_SLAB	✗
DARK_OAK_STAIRS	✗
DARK_OAK_TRAPDOOR	✗

Material	Aquatic block[22]
DARK_OAK_WALL_SIGN	✗
DARK_PRISMARINE_SLAB	✗
DARK_PRISMARINE_STAIRS	✗
DEEPSLATE_BRICK_SLAB	✗
DEEPSLATE_BRICK_STAIRS	✗
DEEPSLATE_BRICK_WALL	✗
DEEPSLATE_TILE_SLAB	✗
DEEPSLATE_TILE_STAIRS	✗
DEEPSLATE_TILE_WALL	✗
DETECTOR_RAIL	✗
DIORITE_SLAB	✗
DIORITE_STAIRS	✗
DIORITE_WALL	✗
ENDER_CHEST	✗
END_STONE_BRICK_SLAB	✗
END_STONE_BRICK_STAIRS	✗
END_STONE_BRICK_WALL	✗
EXPOSED_CUT_COPPER_SLAB	✗
EXPOSED_CUT_COPPER_STAIRS	✗
FLOWERING_AZALEA_LEAVES	✗
GLASS_PANE	✗
GLOW_LICHEN	✗
GRANITE_SLAB	✗
GRANITE_STAIRS	✗
GRANITE_WALL	✗
GRAY_CANDLE	✗
GRAY_STAINED_GLASS_PANE	✗
GREEN_CANDLE	✗
GREEN_STAINED_GLASS_PANE	✗
HANGING_ROOTS	✗

Material	Aquatic block[22]
IRON_BARS	✗
IRON_TRAPDOOR	✗
JUNGLE_FENCE	✗
JUNGLE_LEAVES	✗
JUNGLE_SIGN	✗
JUNGLE_SLAB	✗
JUNGLE_STAIRS	✗
JUNGLE_TRAPDOOR	✗
JUNGLE_WALL_SIGN	✗
LADDER	✗
LANTERN	✗
LARGE_AMETHYST_BUD	✗
LIGHTNING_ROD	✗
LIGHT_BLUE_CANDLE	✗
LIGHT_BLUE_STAINED_GLASS_PANE	✗
LIGHT_GRAY_CANDLE	✗
LIGHT_GRAY_STAINED_GLASS_PANE	✗
LIME_CANDLE	✗
LIME_STAINED_GLASS_PANE	✗
MAGENTA_CANDLE	✗
MAGENTA_STAINED_GLASS_PANE	✗
MANGROVE_FENCE	✗
MANGROVE_LEAVES	✗
MANGROVE_PROPAGULE	✗
MANGROVE_ROOTS	✗
MANGROVE_SIGN	✗
MANGROVE_SLAB	✗
MANGROVE_STAIRS	✗
MANGROVE_TRAPDOOR	✗
MANGROVE_WALL_SIGN	✗

Material	Aquatic block[22]
MEDIUM_AMETHYST_BUD	✗
MOSSY_COBBLESTONE_SLAB	✗
MOSSY_COBBLESTONE_STAIRS	✗
MOSSY_COBBLESTONE_WALL	✗
MOSSY_STONE_BRICK_SLAB	✗
MOSSY_STONE_BRICK_STAIRS	✗
MOSSY_STONE_BRICK_WALL	✗
MUD_BRICK_SLAB	✗
MUD_BRICK_STAIRS	✗
MUD_BRICK_WALL	✗
NETHER_BRICK_FENCE	✗
NETHER_BRICK_SLAB	✗
NETHER_BRICK_STAIRS	✗
NETHER_BRICK_WALL	✗
OAK_FENCE	✗
OAK_LEAVES	✗
OAK_SIGN	✗
OAK_SLAB	✗
OAK_STAIRS	✗
OAK_TRAPDOOR	✗
OAK_WALL_SIGN	✗
ORANGE_CANDLE	✗
ORANGE_STAINED_GLASS_PANE	✗
OXIDIZED_CUT_COPPER_SLAB	✗
OXIDIZED_CUT_COPPER_STAIRS	✗
PETRIFIED_OAK_SLAB	✗
PINK_CANDLE	✗
PINK_STAINED_GLASS_PANE	✗
POINTED_DRIPSTONE	✗
POLISHED_ANDESITE_SLAB	✗

Material	Aquatic block[22]
POLISHED_ANDESITE_STAIRS	✗
POLISHED_BLACKSTONE_BRICK_SLAB	✗
POLISHED_BLACKSTONE_BRICK_STAIRS	✗
POLISHED_BLACKSTONE_BRICK_WALL	✗
POLISHED_BLACKSTONE_SLAB	✗
POLISHED_BLACKSTONE_STAIRS	✗
POLISHED_BLACKSTONE_WALL	✗
POLISHED_DEEPSLATE_SLAB	✗
POLISHED_DEEPSLATE_STAIRS	✗
POLISHED_DEEPSLATE_WALL	✗
POLISHED_DIORITE_SLAB	✗
POLISHED_DIORITE_STAIRS	✗
POLISHED_GRANITE_SLAB	✗
POLISHED_GRANITE_STAIRS	✗
POWERED_RAIL	✗
PRISMARINE_BRICK_SLAB	✗
PRISMARINE_BRICK_STAIRS	✗
PRISMARINE_SLAB	✗
PRISMARINE_STAIRS	✗
PRISMARINE_WALL	✗
PURPLE_CANDLE	✗
PURPLE_STAINED_GLASS_PANE	✗
PURPUR_SLAB	✗
PURPUR_STAIRS	✗
QUARTZ_SLAB	✗
QUARTZ_STAIRS	✗
RAIL	✗
RED_CANDLE	✗
RED_NETHER_BRICK_SLAB	✗
RED_NETHER_BRICK_STAIRS	✗

Material	Aquatic block[22]
RED_NETHER_BRICK_WALL	✗
RED_SANDSTONE_SLAB	✗
RED_SANDSTONE_STAIRS	✗
RED_SANDSTONE_WALL	✗
RED_STAINED_GLASS_PANE	✗
SANDSTONE_SLAB	✗
SANDSTONE_STAIRS	✗
SANDSTONE_WALL	✗
SCAFFOLDING	✗
SCULK_SENSOR	✗
SCULK_SHRIEKER	✗
SCULK_VEIN	✗
SMALL_AMETHYST_BUD	✗
SMALL_DRIPLEAF	✗
SMOOTH_QUARTZ_SLAB	✗
SMOOTH_QUARTZ_STAIRS	✗
SMOOTH_RED_SANDSTONE_SLAB	✗
SMOOTH_RED_SANDSTONE_STAIRS	✗
SMOOTH_SANDSTONE_SLAB	✗
SMOOTH_SANDSTONE_STAIRS	✗
SMOOTH_STONE_SLAB	✗
SOUL_CAMPFIRE	✗
SOUL_LANTERN	✗
SPRUCE_FENCE	✗
SPRUCE_LEAVES	✗
SPRUCE_SIGN	✗
SPRUCE_SLAB	✗
SPRUCE_STAIRS	✗
SPRUCE_TRAPDOOR	✗
SPRUCE_WALL_SIGN	✗

Material	Aquatic block[22]
STONE_BRICK_SLAB	✗
STONE_BRICK_STAIRS	✗
STONE_BRICK_WALL	✗
STONE_SLAB	✗
STONE_STAIRS	✗
TRAPPED_CHEST	✗
WARPED_FENCE	✗
WARPED_SIGN	✗
WARPED_SLAB	✗
WARPED_STAIRS	✗
WARPED_TRAPDOOR	✗
WARPED_WALL_SIGN	✗
WAXED_CUT_COPPER_SLAB	✗
WAXED_CUT_COPPER_STAIRS	✗
WAXED_EXPOSED_CUT_COPPER_SLAB	✗
WAXED_EXPOSED_CUT_COPPER_STAIRS	✗
WAXED_OXIDIZED_CUT_COPPER_SLAB	✗
WAXED_OXIDIZED_CUT_COPPER_STAIRS	✗
WAXED_WEATHERED_CUT_COPPER_SLAB	✗
WAXED_WEATHERED_CUT_COPPER_STAIRS	✗
WEATHERED_CUT_COPPER_SLAB	✗
WEATHERED_CUT_COPPER_STAIRS	✗
WHITE_CANDLE	✗
WHITE_STAINED_GLASS_PANE	✗
YELLOW_CANDLE	✗
YELLOW_STAINED_GLASS_PANE	✗
BRAIN_CORAL	✓
BRAIN_CORAL_FAN	✓
BRAIN_CORAL_WALL_FAN	✓
BUBBLE_CORAL	✓

Material	Aquatic block[22]
BUBBLE_CORAL_FAN	✓
BUBBLE_CORAL_WALL_FAN	✓
CONDUIT	✓
DEAD_BRAIN_CORAL	✓
DEAD_BRAIN_CORAL_FAN	✓
DEAD_BRAIN_CORAL_WALL_FAN	✓
DEAD_BUBBLE_CORAL	✓
DEAD_BUBBLE_CORAL_FAN	✓
DEAD_BUBBLE_CORAL_WALL_FAN	✓
DEAD_FIRE_CORAL	✓
DEAD_FIRE_CORAL_FAN	✓
DEAD_FIRE_CORAL_WALL_FAN	✓
DEAD_HORN_CORAL	✓
DEAD_HORN_CORAL_FAN	✓
DEAD_HORN_CORAL_WALL_FAN	✓
DEAD_TUBE_CORAL	✓
DEAD_TUBE_CORAL_FAN	✓
DEAD_TUBE_CORAL_WALL_FAN	✓
FIRE_CORAL	✓
FIRE_CORAL_FAN	✓
FIRE_CORAL_WALL_FAN	✓
HORN_CORAL	✓
HORN_CORAL_FAN	✓
HORN_CORAL_WALL_FAN	✓
SEA_PICKLE	✓
TUBE_CORAL	✓
TUBE_CORAL_FAN	✓
TUBE_CORAL_WALL_FAN	✓

Table A.45: Waterlogged materials

A.3 Material modifiers aggregation

A.3.1 Direction aggregation

modifier, followed by the condition to set the bit to 1

Property	U	D	N	S	E	W
up	true					
down		true				
north			true			
south				true		
east					true	
west						true
face	ceiling	floor				
attachment	ceiling	floor				
half	top or upper	bottom or lower				
facing	up	down	north	south	east	west
vertical-direction	up	down				
type	top or double	bottom or double				

---

[22] If it's an underwater block (defaults to true).

Property	U	D	N	S	E	W
orientation	up_X	down_X	X_north or north_up	X_south or south_up	X_east or east_up	X_west or west_up
shape	ascending_X		ascending_north or north_X	ascending_south or south_X or north_south	ascending_east or X_east or east_west	ascending_west or X_west
hanging	true					

Table A.46: Direction aggregation

A.3.2 Axis aggregation

modifier up to 2b

Property	MSB	LSB
axis	Y or Z	X or Z
attachment	double_wall	single_wall

Table A.47: Axis aggregation

A.3.3   Age aggregation

modifier up to 8b

Property
age
berries
honey_level

Table A.48: Age aggregation

A.3.4   Grouped materials aggregation

modifier up to 3b

Property
candles
eggs
pickles

Table A.49: Grouped materials aggregation

A.3.5   Stages aggregation

modifier up to 8b

Property
charges
level
bites
layers

Table A.50: Stages aggregation

A.3.6   Parts aggregation

modifier; up to 4b

Property
part
piston_head
stair_shape

Table A.51: Parts aggregation

References

Ascii table. (n.d.). Retrieved from `https://www.ascii-code.com/`

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

Rajaraman, V. (2016, January). Ieee standard for floating point numbers. *Resonance*. Retrieved from `https://www.ias.ac.in/article/fulltext/reso/021/01/0011-0030`

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

Spigot - enum material - 1.19. (n.d.). Retrieved 2022-08-05, from `https://hub.spigotmc.org/javadocs/spigot/org/bukkit/material/package-summary.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`