LU Documentation Documentation

Release 1.0

humanoid lcdr

Mar 22, 2024

CONTENTS:

1 Introduction **3**
- 1.1 Quick notes to get started . . . 3
- 1.2 Packet Captures . . . 4
- 1.3 Appendix A: LEGO data format and data type IDs . . . 4
- 1.4 Appendix B: Maps info and checksum . . . 5

2 Contributing **7**
- 2.1 When should I contribute? . . . 7
- 2.2 Filing an Issue . . . 7
- 2.3 Creating a pull request . . . 7
- 2.4 Building the docs locally . . . 8

3 Packets **9**

4 Common Structures **11**
- 4.1 Type `padding` . . . 11
- 4.2 Type `header` . . . 11
- 4.3 Type `lu_str` . . . 12
- 4.4 Type `lu_wstr` . . . 12
- 4.5 Enum `service_id` . . . 12

5 General **13**
- 5.1 Type `packet` . . . 13
- 5.2 Type `handshake` . . . 14
- 5.3 Type `disconnect_notify` . . . 14
- 5.4 Enum `packet_id` . . . 14
- 5.5 Enum `disconnect_reason` . . . 14

6 Introduction **17**
- 6.1 Tools . . . 17
- 6.2 Resources . . . 17
- 6.3 Common Types . . . 17

7 Compression formats **21**
- 7.1 Segmented (.sd0/.si0) . . . 21
- 7.2 Manifest (.txt) . . . 22
- 7.3 Catalog (.pki) . . . 22
- 7.4 Pack (.pk) . . . 23

8 Game data **27**
- 8.1 Database (.fdb) . . . 27

8.2 Triggers (.lutriggers) . . . 33
8.3 Zone (.luz) . . . 35
8.4 Level (.lvl) . . . 43
8.5 Terrain (.raw) . . . 51

9 Other formats **55**
9.1 Environment (.evc) . . . 55
9.2 Assets (.zal/.ast) . . . 55
9.3 Animations (.gfx) . . . 55

10 Game Mechanics **57**
10.1 Skill / Behavior System . . . 57
10.2 Kit Factions . . . 93
10.3 Flag System . . . 94
10.4 Mission System . . . 102
10.5 Scripting (LUA) . . . 105
10.6 Minimaps . . . 108
10.7 Properties . . . 108
10.8 Loading a world . . . 110

11 Game Database **111**
11.1 AICombatRoles . . . 111
11.2 AccessoryDefaultLoc . . . 111
11.3 Activities . . . 113
11.4 ActivityRewards . . . 114
11.5 ActivityText . . . 114
11.6 AnimationIndex . . . 114
11.7 Animations . . . 115
11.8 BaseCombatAIComponent . . . 115
11.9 BehaviorEffect . . . 116
11.10 BehaviorParameter . . . 116
11.11 BehaviorTemplate . . . 117
11.12 BehaviorTemplateName . . . 117
11.13 Blueprints . . . 117
11.14 BrickColors . . . 118
11.15 BrickIDTable . . . 118
11.16 BuffDefinitions . . . 118
11.17 BuffParameters . . . 118
11.18 Camera . . . 119
11.19 CelebrationParameters . . . 120
11.20 ChoiceBuildComponent . . . 121
11.21 CollectibleComponent . . . 121
11.22 ComponentsRegistry . . . 121
11.23 ControlSchemes . . . 122
11.24 CurrencyDenominations . . . 122
11.25 CurrencyTable . . . 123
11.26 DBExclude . . . 123
11.27 DeletionRestrictions . . . 123
11.28 DestructibleComponent . . . 124
11.29 DevModelBehaviors . . . 124
11.30 Emotes . . . 124
11.31 EventGating . . . 125
11.32 ExhibitComponent . . . 125
11.33 Factions . . . 125

11.34 FeatureGating . . . 126
11.35 FlairTable . . . 126
11.36 Icons . . . 126
11.37 InventoryComponent . . . 126
11.38 ItemComponent . . . 127
11.39 ItemEggData . . . 128
11.40 ItemFoodData . . . 128
11.41 ItemSetSkills . . . 128
11.42 ItemSets . . . 129
11.43 JetPackPadComponent . . . 129
11.44 LUPExhibitComponent . . . 129
11.45 LUPExhibitModelData . . . 130
11.46 LUPZoneIDs . . . 130
11.47 LanguageType . . . 130
11.48 LevelProgressionLookup . . . 130
11.49 LootMatrix . . . 131
11.50 LootMatrixIndex . . . 131
11.51 LootTable . . . 131
11.52 LootTableIndex . . . 131
11.53 MinifigComponent . . . 132
11.54 MinifigDecals_Eyebrows . . . 132
11.55 MinifigDecals_Eyes . . . 132
11.56 MinifigDecals_Legs . . . 133
11.57 MinifigDecals_Mouths . . . 133
11.58 MinifigDecals_Torsos . . . 133
11.59 MissionEmail . . . 133
11.60 MissionNPCComponent . . . 134
11.61 MissionTasks . . . 134
11.62 MissionText . . . 134
11.63 Missions . . . 135
11.64 ModelBehavior . . . 136
11.65 ModularBuildComponent . . . 137
11.66 ModuleComponent . . . 137
11.67 MotionFX . . . 137
11.68 MovementAIComponent . . . 138
11.69 MovingPlatforms . . . 138
11.70 NpcIcons . . . 139
11.71 ObjectBehaviorXREF . . . 139
11.72 ObjectBehaviors . . . 139
11.73 ObjectSkills . . . 140
11.74 Objects . . . 140
11.75 PackageComponent . . . 140
11.76 PetAbilities . . . 141
11.77 PetComponent . . . 141
11.78 PetNestComponent . . . 141
11.79 PhysicsComponent . . . 142
11.80 PlayerFlags . . . 142
11.81 PlayerStatistics . . . 142
11.82 PossessableComponent . . . 143
11.83 Preconditions . . . 143
11.84 PropertyEntranceComponent . . . 144
11.85 PropertyTemplate . . . 145
11.86 ProximityMonitorComponent . . . 145
11.87 ProximityTypes . . . 146

11.88 RacingModuleComponent . . . 146
11.89 RailActivatorComponent . . . 146
11.90 RarityTable . . . 147
11.91 RarityTableIndex . . . 147
11.92 RebuildComponent . . . 147
11.93 RebuildSections . . . 148
11.94 Release_Version . . . 148
11.95 RenderComponent . . . 149
11.96 RenderComponentFlash . . . 149
11.97 RenderComponentWrapper . . . 150
11.98 RenderIconAssets . . . 150
11.99 ReputationRewards . . . 150
11.100 RewardCodes . . . 150
11.101 Rewards . . . 151
11.102 RocketLaunchpadControlComponent . . . 151
11.103 SceneTable . . . 151
11.104 ScriptComponent . . . 152
11.105 SkillBehavior . . . 152
11.106 SmashableChain . . . 153
11.107 SmashableChainIndex . . . 153
11.108 SmashableComponent . . . 153
11.109 SmashableElements . . . 153
11.110 SpeedchatMenu . . . 154
11.111 SubscriptionPricing . . . 154
11.112 SurfaceType . . . 154
11.113 TamingBuildPuzzles . . . 155
11.114 TextDescription . . . 155
11.115 TextLanguage . . . 155
11.116 TrailEffects . . . 156
11.117 UGBehaviorSounds . . . 156
11.118 VehiclePhysics . . . 157
11.119 VehicleStatMap . . . 159
11.120 VendorComponent . . . 159
11.121 WhatsCoolItemSpotlight . . . 160
11.122 WhatsCoolNewsAndTips . . . 160
11.123 WorldConfig . . . 160
11.124 ZoneLoadingTips . . . 162
11.125 ZoneSummary . . . 162
11.126 ZoneTable . . . 162
11.127 brickAttributes . . . 164
11.128 dtproperties . . . 164
11.129 mapAnimationPriorities . . . 164
11.130 mapAssetType . . . 164
11.131 mapIcon . . . 165
11.132 mapItemTypes . . . 165
11.133 mapRenderEffects . . . 165
11.134 mapShaders . . . 165
11.135 mapTextureResource . . . 166
11.136 map_BlueprintCategory . . . 166
11.137 sysdiagrams . . . 166

12 Components **167**
12.1 Controllable Physics Component (1) . . . 167
12.2 Render Component (2) . . . 168

12.3 Simple Physics Component (3) . . . 168
12.4 Character Component (4) . . . 168
12.5 Script Component (5) . . . 172
12.6 Bouncer Component (6) . . . 172
12.7 Destroyable Component (7) . . . 172
12.8 Ghost Component (8) . . . 173
12.9 Skill Component (9) . . . 173
12.10 Spawner Component (10) . . . 174
12.11 Item Component (11) . . . 174
12.12 Rebuild Component (12) . . . 175
12.13 Rebuild Start Component (13) . . . 175
12.14 Rebuild Activator Component (14) . . . 175
12.15 Icon Only Component (15) . . . 175
12.16 Vendor Component (16) . . . 175
12.17 Inventory Component (17) . . . 176
12.18 Projectile Physics Component (18) . . . 178
12.19 Shooting Gallery Component (19) . . . 179
12.20 RigidBodyPhantomPhysics Component (20) . . . 179
12.21 Drop Effect Component (21) . . . 179
12.22 Chest Component (22) . . . 179
12.23 Collectible Component (23) . . . 179
12.24 Blueprint Component (24) . . . 179
12.25 Moving Platform Component (25) . . . 180
12.26 Pet Component (26) . . . 180
12.27 Platform Boundary Component (27) . . . 180
12.28 Module Component (28) . . . 181
12.29 Jet Pack Pad (29) . . . 181
12.30 Vehicle Physics Component (30) . . . 181
12.31 MovementAI Component (31) . . . 181
12.32 Exhibit Component (32) . . . 181
12.33 OverheadIcon Component (33) . . . 182
12.34 Pet Control Component (34) . . . 182
12.35 Minifig Component (35) . . . 183
12.36 Property Component (36) . . . 184
12.37 Pet Creator Component (37) . . . 184
12.38 Model Builder Component (38) . . . 184
12.39 Scripted Activity Component (39) . . . 184
12.40 Phantom Physics Component (40) . . . 184
12.41 Springpad Component (41) . . . 184
12.42 B3 Behaviors Component (42) . . . 185
12.43 Property Entrance Component (43) . . . 185
12.44 FX Component (44) . . . 185
12.45 Property Management Component (45) . . . 185
12.46 Vehicle Physics Component (46) . . . 185
12.47 Physics System Component (47) . . . 185
12.48 Quick Build Component (48) . . . 186
12.49 Switch Component (49) . . . 186
12.50 Minigame Component (50) . . . 186
12.51 Changling Component (51) . . . 186
12.52 Choice Build Component (52) . . . 186
12.53 Package Component (53) . . . 187
12.54 Sound Repeater Component (54) . . . 187
12.55 Sound Ambient 2D Component (55) . . . 187
12.56 Sound Ambient 3D Component (56) . . . 187

12.57 Precondition Component (57) . . . 187
12.58 Player Flags Component (58) . . . 187
12.59 Custom Build Assembly Component (59) . . . 189
12.60 Base Combat AI Component (60) . . . 189
12.61 Module Assembly Component (61) . . . 190
12.62 Showcase Model Handler Component (62) . . . 190
12.63 Racing Module Component (63) . . . 190
12.64 Generic Activator Component (64) . . . 190
12.65 Property Vendor Component (65) . . . 191
12.66 HFLightDirectionGadget Component (66) . . . 191
12.67 Rocket Launch Component (67) . . . 191
12.68 Rocket Landing Component (68) . . . 191
12.69 Trigger Component (69) . . . 191
12.70 Dropped Loot Component (70) . . . 191
12.71 Racing Control Component (71) . . . 192
12.72 Faction Trigger Component (72) . . . 192
12.73 Mission Offer Component (73) . . . 192
12.74 Racing Stats Component (74) . . . 192
12.75 LUP Exhibit Component (75) . . . 192
12.76 BBB Component (76) . . . 193
12.77 Sound Trigger Component (77) . . . 193
12.78 Proximity Monitor Component (78) . . . 193
12.79 Racing Sound Trigger Component (79) . . . 193
12.80 Chat Component (80) . . . 193
12.81 Friends List Component (81) . . . 193
12.82 Guild Component (82) . . . 194
12.83 Local System Component (83) . . . 194
12.84 Mission Component (84) . . . 194
12.85 Mutable Model Behaviors Component (85) . . . 195
12.86 Pathfinding Control Component (86) . . . 195
12.87 Pet Taming Control Component (87) . . . 196
12.88 Property Editor Component (88) . . . 196
12.89 Skinned Render Component (89) . . . 196
12.90 Slash Command Component (90) . . . 196
12.91 Status Effect Component (91) . . . 196
12.92 Teams Component (92) . . . 196
12.93 Text Effect Component (93) . . . 196
12.94 Trade Component (94) . . . 196
12.95 User Control Component (95) . . . 197
12.96 Ignore List Component (96) . . . 197
12.97 LUP Launchpad Component (97) . . . 197
12.98 Buff Component (98) . . . 197
12.99 Interaction Manager Component (99) . . . 197
12.100 Donation Vendor Component (100) . . . 197
12.101 Combat Mediator Component (101) . . . 198
12.102 Player Forced Movement Component (106) . . . 198
12.103 Brick-by-Brick Component (107) . . . 198
12.104 Level Progression Component (109) . . . 198
12.105 Possession Control Component (110) . . . 199
12.106 Commendation Vendor Component (102) . . . 199
12.107 Rail Activator Component (104) . . . 199
12.108 Roller Component (105) . . . 200
12.109 Possessable Component (108) . . . 200
12.110 Property Plaque Component (113) . . . 200

12.111 Build Border Component (114) 200
12.112 Culling Plane Component (116) 200
12.113 Mount Control Component (?) 200

13 Contact Info **203**

Note: This is a read-the-docs port of lu_packet_structs, written by humanoid, lcdr and others on Google Docs. It was ported by @Xiphoseer. This is currently a proof of concept and is not guaranteed to reflect the latest changes.

The purpose of this documentation is to list and protocol all the information about the network packets of the game LEGO Universe. For organization purposes the contents of the documentation is extended to the following documents:

CHAPTER

ONE

INTRODUCTION

Note: This is a read-the-docs port of the original google docs lu_packet_structs, written by humanoid, lcdr and others, ported by @Xiphoseer. This is currently a proof of concept and is not guaranteed to reflect the latest changes.

1.1 Quick notes to get started

1.1.1 Client

- This documentation is targeted towards the latest publicly released client (1.10.64)
- To redirect the client to a different server simply change the `AUTHSERVERIP` host info in the `boot.cfg` file to a new host.
- The client stores an additional config and a log file from the last session in the *`SystemDrive`*`:\Users{User}\AppData\Local\LEGO Software\LEGO Universe\` folder.

1.1.2 Server

- The client uses the RakNet network library (v3.25) to communicate with the server, therefore it is recommended to use it in the server if you want to work on one and are new to this project.
- You can download it here (documentation and sample projects are included), note that later versions of the library won't work due to changes in the network protocol. Alternatively lcdr wrote a python version with the minimum features needed to run a server for the game implemented, available here (no documentation yet so not recommended for inexperienced users)
- The listening port for the Authentication Server is hardcoded to 1001 (UDP), the ports for the following instances (char, world) depend on what the Authentication Server sends to the client but the port range used for the original servers was 2001-2200 (UDP)
- In order for the server to establish a working connection with the client it is required to set up a pre-set password in RakNet by calling `SetIncomingPassword("3.25 ND1", 8);` for the *RakPeerInterface* instance before listening for packets
- It seems that the server used the `SYSTEM_PRIORITY` and `RELIABLE_ORDERED` options for all outgoing packets to the client (though that's probably not a requirement)

1.2 Packet Captures

Thanks to pwjones we have access to quite a lot of (partial) traffic sessions of the original server which serve as a basis for this documentation, if you want to dig into them yourself, here is a download link.

Since the original captures (*.*pcap* files) were encrypted by RakNet it was required to decrypt them again (*.*bin* files stored in *.*zip* archives where each archive represents a session) which was only possible using a piece of information that RakNet exchanges at the beginning of a traffic.

However since more than half of the captured traffics only consist of a fraction of the entire session this information is missing for those captures (marked as *_*unresolved.pcap*), making them undecryptable at this point (they're still included in the archives though, so all available captures are in one place).

There are a few exceptions where it was possible to decrypt them using the same information from other traffic sessions but the remaining ones can't be decrypted with this method so they'll remain unreadable for the time being.

The naming format used is: `[packet number]_[source port]-[destination port]_[packet header]_[optional].bin` where `[optional]` is used to display game message ids or network ids and LOTs in round brackets for the according packet types.

Extracting the entire .zip capture(s) is not recommended, since this many files (several ten thousand) will have a huge overhead on the file system (because of things like last modified info, which isn't applicable here anyways), resulting in a much larger file space consumption and slower access times when trying to list the files in the explorer. If you want to search multiple captures for specific files, use this script: https://github.com/lcdr/utils/ find_packets.py

(The script also yields the binary content of the packets, which can be useful for further filtering or logging) If you want to look at the raw data of a packet yourself (not recommended for inexperienced users) you can of course extract single files from the .zip archive using an archive extractor of your choice (I recommend 7-Zip). Then you can open the extracted *.*bin* file(s) using a hex editor of your choice (for some packet types it might be useful to have an editor that can shift the bits in the data, no recommendation here). Alternatively a graphical viewer for capture files is available at https://github.com/lcdr/utils/tree/master/utils/captureviewer.py (takes the entire .zip archive of a traffic as input, no need to extract anything)

1.3 Appendix A: LEGO data format and data type IDs

LDF is used in boot.cfg, client xml settings, .luz and .lvl files, and the binary part of the chardata packet.

This binary data format is used in various packets, for example the chardata packet.

[u32] - number of keys
  [u8] - key length in bytes
  [wchar] - key
  [u8] - data type (see below)
  [according to data type] - data

The text format has the format: `key=type:value`

- 0: String (variable wstring?)
- 1: s32
- 2: ??? (haven't found an occurrence of this type so far)
- 3: Float (32bit, signed)

- 4: ??? (Location&Size, appeared on lwo_override.xml)
- 5: u32
- 6: ??? (haven't found an occurrence of this type so far)
- 7: Boolean (8bit, 0 or 1)
- 8: s64
- 9: s64, Used only for (object?) IDs?
- 10: ??? (haven't found an occurrence of this type so far)
- 11: ??? (haven't found an occurrence of this type so far)
- 12: ??? (haven't found an occurrence of this type so far)
- 13: in chardata this was XML data, in client settings checksum, in lvl files strings/GUIDs (maybe it's for bytes)

1.4 Appendix B: Maps info and checksum

Here are the checksums I found. Probably need to go back through and find the different map instances if I can.

Map Name	Zone ID	Checksum
Venture Explorer	1000	7c 08 b8 20
Return to Venture Explorer	1001	3c 0a 68 26
Avant Gardens	1100	11 55 52 49
Avant Gardens Survival	1101	e2 14 82 53
Spider Queen Battle	1102	da 03 d4 0f
Block Yard	1150	da 03 d4 0f
Avant Grove	1151	03 03 89 0a
Nimbus Station	1200	30 6b 1e da
Pet Cove	1201	30 13 6e 47
Vertigo Loop Racetrack	1203	02 05 fc 10
Battle of Nimbus Station	1204	58 02 d4 07
Nimbus Rock	1250	91 01 8d 05
Nimbus Isle	1251	5d 04 4f 09
Frostburgh	1260	currently disabled in the client
Gnarled Forest	1300	90 c2 ea 12
Canyon Cove	1302	ef 02 77 0b
Keelhaul Canyon	1303	todo
Chantey Shantey	1350	5c 01 b6 04
Forbidden Valley	1400	0d 76 19 85
Forbidden Valley Dragon	1402	87 01 f5 02
Dragonmaw Chasm	1403	4e 0f 85 81
Raven Bluff	1450	26 01 f0 03
Starbase 3001	1600	ee 02 c2 07
Deep Freeze	1601	06 01 32 02
Robot City	1602	7f 03 93 07
Moon Base	1603	ad 01 3b 04
Portabello	1604	dd 07 15 18
LEGO Club	1700	38 01 04 02
Crux Prime	1800	99 a3 17 4b
Nexus Tower	1900	3c f4 4a 9e

continues on next page

Table 1 – continued from previous page

Map Name	Zone ID	Checksum
Ninjago	2000	74 2c 69 4d
Frakjaw Battle	2001	ef 00 eb 09

CHAPTER

TWO

CONTRIBUTING

2.1 When should I contribute?

This documentation was created with the intention that anybody could provide their own research to the community at large. Because of this we want to make it as straightforward as possible for people to edit and update the documentation. We appreciate any kind of contribution, be it as small as fixing typos, or as large as a full reverse engineered structure for a file type. Feel free to ask via Issue or IRC when you are unsure about anything.

2.2 Filing an Issue

When there is something wrong or missing within the documentation, that needs to be addressed but it's not clear what should be done about it, you can open an issue on the repository for theses docs. When the issue is related to a specific part of the docs, open that file on GitHub, select the relevant lines, and click on "Open a new Issue" in the context menu for the selection. When the issue is not related to a specific part, such a part does not exist, open one directly from the issues overview page.

2.2.1 Issue layout

Every issue should contain a meaningful title and description on what is meant to be discussed. When referring to parts of the docs, a link to the docs page or its source code should be provided. Issues are the right place for discussion, before changing some part of the docs.

2.3 Creating a pull request

Usually you will just want to edit a page of the docs. For that, click on "Edit on GitHub" on any page of the actual docs. Then click the edit symbol (pen) above the preview of the file. Be aware that github can not render/preview all elements of the docs correctly. Then edit the file and save your changes.

You usually won't have write access to the repository, so github will create a *fork* of the repository and you will be asked if you want to create a pull request for your changes. Confirm that and we'll review the request and merge it into the docs if we don't have any objections.

2.4 Building the docs locally

If you want to build the docs on your own computer, please install `make` (on linux) and `python`. It is recommended to use *virtualenv* if you can, run

- `virtualenv venv` within the main directory after you cloned the repository and
- `source venv/bin/activate` every time you open a new console to build the docs

With *pip* installed, run

- `pip install sphinx`
- `pip install sphinx-rtd-theme`

To build the docs, run `make html` within the `/docs` folder

CHAPTER

THREE

PACKETS

CHAPTER

FOUR

COMMON STRUCTURES

Note:

id
`net`

endian
`le`

4.1 Type `padding`

4.1.1 params

1. **id**
   `size`

   type
   `u2`

4.1.2 Sequence

1. **Size:** `size`

4.2 Type `header`

4.2.1 Sequence

1. `[83]` *raknet_user_packet_id*
2. **[u2:service_id]** *service_id*
3. **[u4]** *packet_id*
4. **[u1]**

4.3 Type `lu_str`

4.3.1 params

1. **id**
   `size`

   type
   `u2`

4.3.2 Sequence

1. **[strz]**
   Size: `size`
   Encoding: `latin1`

4.4 Type `lu_wstr`

4.4.1 params

1. **id**
   `size`

   type
   `u2`

4.4.2 Sequence

1. **[strz]**
   Size: `size * 2`
   Encoding: `utf16-le`

4.5 Enum `service_id`

0
`general`

1
`auth`

2
`chat`

4
`world`

5
`client`

CHAPTER

FIVE

GENERAL

Note:

id
`general`

endian
`le`

imports

1. `net`

5.1 Type `packet`

5.1.1 params

1. **id**
   `id`

   type
   `u4`

5.1.2 Sequence

1. **[switch-on:id]**

Table 1: Type Cases

0	`handshake`
1	`disconnect_notify`

5.2 Type handshake

5.2.1 Sequence

1. **[u4]** *network_version*
2. **[net::padding(4)]**
3. **[u2:net::service_id]** *service_id*
4. **[net::padding(2)]**
5. **[u4]** *process_id*
6. **[u2]** *port*
7. **[net::padding(35)]**

5.3 Type disconnect_notify

5.3.1 Sequence

1. **[u4:disconnect_reason]** *reason*
2. **if:** `reason == disconnect_reason::wrong_game_version or reason == disconnect_reason::wrong_server_version`
   [u4] *version*

5.4 Enum packet_id

0

`handshake`

1

`disconnect_notify`

5.5 Enum disconnect_reason

0

`unknown_server_error`

Unspecified disconnect reason.

1

`wrong_game_version`

The client's [`handshake::network_version`] did not match the server's [`handshake::network_version`]. The message contains the server network version number.

2

`wrong_server_version`
Unused for client-server.

3

`connection_on_invalid_port`
Connection attempt on invalid port, server emulators probably won't send this as they usually won't have server-server communication using LU's protocol.

4

`duplicate_login`
There was another login with your account and your session has been closed in favor of the new login.

5

`server_shutdown`
The server is shutting down.

6

`unable_to_load_map`
No server hosting this map is available.

7

`invalid_session_key`
The provided [`world::client_validation::session_key`] is incorrect.

8

`account_not_in_pending_list`
Server did not expect a [`world::client_validation`] at this time.

9

`character_not_found`
The provided [`world::character_login_request::char_id`] was not a valid character ID of this account.

10

`character_corruption`
The character seems to be corrupted in the database.

11

`kick`
You were kicked from the server.

12

`save_failure`
Error saving or loading progress.

13

`free_trial_expired`
The account's time-limited free trial expired, unused.

14

`play_schedule_time_up`
The parental controls for this account prevent it from further play.

CHAPTER

SIX

INTRODUCTION

Note: This is a read-the-docs port of the original google docs lu_file_structs, written by humanoid, lcdr and others, ported by @Xiphoseer. This is currently a proof of concept and is not guaranteed to reflect the latest changes.

The purpose of this document is to list and protocol all the information about the client files of the game LEGO Universe (at least the ones that might be helpful for the process of creating a private server). Note that usually most of the client files are packed into `.pk` files which are stored in the `client/res/pack` folder and need to be extracted first to be able to work on them (see Tools section for a link to a simple extractor).

6.1 Tools

- Extract PK files: LUPKExtractor (source code included, linked to from here, original post (most likely))
- Decompress sd0 compressed files: https://bitbucket.org/lcdr/utils/src decompress_sd0.py
- Find keys for fsb files: http://hcs64.com/files/guessfsb03.zip (linked to from here)
- View .nif files http://sourceforge.net/projects/niftools/ (linked to from here)
- Convert FDB to SQLite: https://bitbucket.org/lcdr/utils/src fdb_to_sqlite.py

6.2 Resources

- There are scripts, which require stuff from an "TestAndExample" folder, which is missing, you can get it from here.
- Official LXFML Documentation from the LEGO Group back from 2007

6.3 Common Types

Note:

id
: `common`

endian
: `le`

bit-endian
`le`

6.3.1 Type `vector2`

Sequence

1. **[f4]** *x*
2. **[f4]** *y*

6.3.2 Type `vector3`

Sequence

1. **[f4]** *x*
2. **[f4]** *y*
3. **[f4]** *z*

6.3.3 Type `object_id`

Sequence

1. **[u8]** *object_id*

6.3.4 Type `lot`

Sequence

1. **[s4]** *lot*

6.3.5 Type `quaternion`

Sequence

1. **[f4]** *x*
2. **[f4]** *y*
3. **[f4]** *z*
4. **[f4]** *w*

6.3.6 Type `quaternion_wxyz`

Sequence

1. **[f4]** *w*
2. **[f4]** *x*
3. **[f4]** *y*
4. **[f4]** *z*

6.3.7 Type `u1_str`

Sequence

1. **[u1]** *length*
2. **[str]** *str*
   Size: `length`
   Encoding: `ascii`

6.3.8 Type `u1_wstr`

Sequence

1. **[u1]** *length*
2. **[str]** *str*
   Size: `length * 2`
   Encoding: `utf-16le`

6.3.9 Type `u4_str`

Sequence

1. **[u4]** *length*
2. **[str]** *str*
   Size: `length`
   Encoding: `ascii`

6.3.10 Type `u4_wstr`

Sequence

1. **[u4]** *length*
2. **[str]** *str*
   Size: `length * 2`
   Encoding: `utf-16le`

6.3.11 Type `bool`

Sequence

1. **[u1:boolean]** *bool*

6.3.12 Enum `boolean`

0
: `False`

1
: `True`

CHAPTER

SEVEN

COMPRESSION FORMATS

7.1 Segmented (.sd0/.si0)

To download content as needed from the server while playing the game, LEGO introduced a custom *segmented* file format. The actual files lived in *Segmented Data (.sd0)* files, while the compression process also generated *Segmented Index (.si0)* files to be used when creating the *Catalog (.pki)*.

7.1.1 Segmented Data (.sd0)

Note: There is a decompressor available, see the *Tools* section.

[L:5] - header, 's'-'d'-'0'-01-ff
repeated:
[L:4] - length of compressed chunk
*a chunk usually consists of 1024*256 uncompressed bytes*
[L:V] - compressed (zlib) chunk

7.1.2 Segmented Index (.si0)

The first line is a header of the following form:

`%s%s:%08x:%s:%08x\r`

with the following data:

1) the file extension `si0`
2) the magic bytes 0x01 0xff as `x01xff`
3) the total size of the input
4) the MD5 hash of the input
5) the chunk size

The rest of the file is one line for every compressed block, in the following form:

`%08x:%08x:%s:%s:%08x:%08x:%s\r`

with the following data:

1) start of the block in the raw file
2) size of the block
3) Adler32 of the raw bytes modulo 0xFFFFFFFF, as hex, with the first two and last letters removed
4) MD5 hash of the raw bytes
5) number of bytes already written to compressed file (without magic bytes)
6) number of compressed bytes
7) MD5 hash of compressed bytes

7.2 Manifest (.txt)

(in `/versions` folder)

[version] section:

There is a single line in this section. The elements are:

1) version number
2) md5 hash of version number as text

[files] section:

Every line represents a file entry which consists of six values (in ASCII format), separated by a `,`

1) filename
2) filesize
3) md5 hash of file
4) compressed filesize
5) md5 hash of compressed file
6) md5 hash of values 1) to 5) (includes ',' separators except the one preceding this value)

7.3 Catalog (.pki)

Note:

- The game had the majority of its data files packed in a custom dynamic archive. Within that system, each file was identified by the CRC-32 value of its filename relative to the installation root.
- The crc value uses the standard CRC-32 polynomial *0x04C11DB7*, an init value of *0xFFFFFFFF*, no output XOR and reverses neither input nor output. The filenames are processed in lowercase, with Win32 \ delimiters and a padding of 4 *0x00* bytes at the end.
- Both filetypes include a representation of a binary tree for all their entries. The root node is always at `size / 2`. Each entry has a field for a left and a right child entry. Both filetypes have their main entry list sorted by the crc value, making it possible to use binary search.

Note:

id
`pki`

file-extension
`pki`

endian
`le`

imports

1. `../common/common`

7.3.1 Sequence

1. `[3, 0, 0, 0]` *version*
2. **[u4]** *num_file_names*
3. **[common::u4_str]** *file_names*
   repeat-expr: `num_file_names`
4. **[u4]** *num_pack_files*
5. **[master_pack_index]** *pack_files*
   repeat-expr: `num_pack_files`

7.3.2 Type `master_pack_index`

Sequence

1. **[u4]** *crc*
2. **[s4]** *lower_crc*
3. **[s4]** *upper_crc*
4. **[u4]** *pack_files_index*
5. **[common::bool]** *is_compressed*
   Size: `4`

7.4 Pack (.pk)

Note:

id
`pk`

file-extension
`pk`

endian
`le`

imports

1. `../common/common`

7.4.1 Sequence

1. `['ndpk', 1, 255, 0]` *header*

7.4.2 Instance `ofs_toc`

[u4]
Position: `_io.size - 8`

7.4.3 Instance `file_revision`

[u4]
Position: `_io.size - 4`

7.4.4 Instance `toc`

[toc]
Position: `ofs_toc`

7.4.5 Type `toc`

Sequence

1. **[u4]** *num_files*
2. **[pack_index]** *file_indices*
   repeat-expr: `num_files`

7.4.6 Type `pack_index`

Sequence

1. **[u4]** *crc*
2. **[s4]** *lower_crc*
3. **[s4]** *upper_crc*
4. **[u4]** *uncompressed_size*
5. **[strz]** *uncompressed_checksum*
   Size: `36`

Encoding: `ascii`

6. **[u4]** *compressed_size*
7. **[strz]** *compressed_checksum*
   Size: `36`
   Encoding: `ascii`
8. **[u4]** *ofs_data*
9. **[common::bool]** *is_compressed*
   Size: `4`

Instance `data`

Position: `ofs_data`
Size: `is_compressed.bool == common::boolean::true ? compressed_size :  uncompressed_size`

Instance `data_divider`

`[255, 0, 0, 221, 0]`
Position: `ofs_data + (is_compressed.bool == common::boolean::true ? compressed_size : uncompressed_size)`

7.4.7 Old Format

CHAPTER

EIGHT

GAME DATA

8.1 Database (.fdb)

You can use the tools collection from assembly to work with FDB files.

Note: There is a converter from fdb to sqlite available, see the *Tools* section. This file type has no relation to firebird database files of the same extension.

Note:

It seems like:

- Tables are sorted by their name in ascii representation. Uppercase letters then underscore then lowercase letters.
- Tables themselves are hash maps. Use *id % row_count* to get the appropriate *row_info*, then follow the *linked_row_info* until all entries with that ID are found.
- When the primary key is a string, a dedicated hash function is used to determin the index of the *row_info* slot.
- Strings are stored spearately for each row, even if they have the same content. This makes for a great amount of redundancy in the file, but keeps editing simple.

Note:

id
`fdb`

file-extension
`fdb`

endian
`le`

imports

1. `../common/common`

8.1.1 Sequence

1. **[u4]** *num_tables*
2. **[u4]** *ofs_table*

8.1.2 Instance `tables`

[table]
Position: `ofs_table`
repeat-expr: `num_tables`

8.1.3 Type `table`

Sequence

1. **[u4]** *ofs_table_desc*
2. **[u4]** *ofs_hash_table*

Instance `table_desc`

[table_description]
Position: `ofs_table_desc`

Instance `hash_table`

[hash_table]
Position: `ofs_hash_table`

8.1.4 Type `table_description`

Sequence

1. **[u4]** *num_columns*
2. **[text]** *table_name*
3. **[u4]** *ofs_columns*

Instance `columns`

[column_description]
Position: `ofs_columns`
repeat-expr: `num_columns`

8.1.5 Type `column_description`

Sequence

1. **[u4:variant_type]** *data_type*
2. **[text]** *column_name*

8.1.6 Type `hash_table`

Sequence

1. **[u4]** *table_size*
2. **[u4]** *ofs_buckets*

Instance `buckets`

[hash_bucket]
Position: `ofs_buckets`
repeat-expr: `table_size`

8.1.7 Type `hash_bucket`

Sequence

1. **[u4]** *ofs_data*

Instance `data`

if: `ofs_data != 0xffffffff`
[list_rows]
Position: `ofs_data`

8.1.8 Type `list_rows`

Sequence

1. **[u4]** *ofs_row_data*
2. **[u4]** *ofs_next_data*

Instance `row_data`

[row_data]
Position: `ofs_row_data`

Instance `next_data`

if: `ofs_next_data != 0xffffffff`
[list_rows]
Position: `ofs_next_data`

8.1.9 Type `row_data`

Sequence

1. **[u4]** *num_data*
2. **[u4]** *ofs_data_array*

Instance `data_array`

[variant_data]
Position: `ofs_data_array`
repeat-expr: `num_data`

8.1.10 Type `variant_data`

Sequence

1. **[u4:variant_type]** *data_type*
2. **[switch-on:data_type]** *data*

Table 1: Type Cases

variant_type::null	`null_data`
variant_type::i32	`s4`
variant_type::u32	`u4`
variant_type::real	`f4`
variant_type::nvarchar	`text`
variant_type::bool	`common::bool`
variant_type::i64	`i64`
variant_type::u64	`u64`
variant_type::text	`text`

8.1.11 Type `null_data`

Sequence

1. `[0, 0, 0, 0]` *null_data*

8.1.12 Type `i64`

Sequence

1. **[u4]** *ofs_i64*

Instance `i64`

[s8]
Position: `ofs_i64`

8.1.13 Type `u64`

Sequence

1. **[u4]** *ofs_u64*

Instance `u64`

[u8]
Position: `ofs_u64`

8.1.14 Type `text`

Sequence

1. **[u4]** *ofs_text*

Instance `text`

[strz]
Position: `ofs_text`
Encoding: `ascii`

8.1.15 Enum `variant_type`

0
: `null`

1
: `i32`

2
: `u32`

3
: `real`

4
: `nvarchar`

5
: `bool`

6
: `i64`

7
: `u64`

8
: `text`

Note:

- Address pointers can be -1 which most likely means an invalid address (just skip those)
- Strings types (TEXT and VARCHAR) are always null-terminated (with some over allocated bytes afterwards it seems, apparently string length are filled to be modulo 4 = 0?)
- Strings and int64 (BIGINT) types are always stored with an additional address pointer, like this: [pointer]->[data]

8.1.16 SQLite Conversion

lcdr's tools rely on https://www.sqlite.org/datatype3.html#determination_of_column_affinity to assign the type of columns in SQLite while preserving the original type:

```
SQLITE_TYPE = {}
SQLITE_TYPE[0] = "none"
SQLITE_TYPE[1] = "int32"
SQLITE_TYPE[3] = "real"
SQLITE_TYPE[4] = "text_4"
SQLITE_TYPE[5] = "int_bool"
SQLITE_TYPE[6] = "int64"
SQLITE_TYPE[8] = "text_8"
```

8.2 Triggers (.lutriggers)

plain text, xml structure

trigger - A trigger
- **id** - as referenced in in the .lvl
- **event** - event type on which the trigger should fire
  - **id** - A EventID value
  - **command** - command to be executed on trigger
    - **id** - command type todo: document possible values
    - **target**
      - `self` for the trigger,
      - `target` for the object that triggered it,
      - `zone` probably the ZoneControlObject,
      - `objGroup` which instantiates another attribute called targetName
    - **args** - command-specific arguments todo:

8.2.1 Possible Values (EventIDs)

- OnDestroy
- OnCustomEvent
- OnEnter
- OnExit
- OnCreate
- OnHit
- OnTimerDone
- OnRebuildComplete
- OnActivated
- OnDectivated [sic]
- OnArrived
- OnArrivedAtEndOfPath
- OnZoneSummaryDismissed
- OnArrivedAtDesiredWaypoint
- OnPetOnSwitch
- OnPetOffSwitch
- OnInteract

8.2.2 Possible Values (Commands)

Command	Parameters
zonePlayer	[zone ID],(0 for non-instanced, 1 for instanced), (x, y, z position), (y rotation), (spawn point name)
fireEvent	(String to send to the recipient)
destroyObj	(0 for violent, 1 for silent)
toggleTrigger	[0 to disable, 1 to enable]
resetRebuild	(0 for normal reset, 1 for "failure" reset)
setPath	[new path name],(starting point index),(0 for forward, 1 for reverse)
setPickType	[new pick type, or -1 to disable picking]
moveObject	[x offset],[y offset],[z offset]
rotateObject	[x rotation],[y rotation],[z rotation]
pushObject	[x direction],[y direction],[z direction]
repelObject	(force multiplier)
setTimer	[timer name],[duration in seconds]
cancelTimer	[timer name]
playCinematic	[cinematic name],(lead-in in seconds),("wait" to wait at end),("unlock" to NOT lock the player cor
toggleBBB	("enter" or "exit" to force direction)
updateMission	[taskType],[targetid],[value1],[value2],[wsValue]
setBouncerState	["on" to activate bouncer or "off" to deactivate bouncer]
bounceAllOnBouncer	No Parameters Required
turnAroundOnPath	No Parameters Required
goForwardOnPath	No Parameters Required
goBackwardOnPath	No Parameters Required
stopPathing	No Parameters Required
startPathing	No Parameters Required
LockOrUnlockControls	["lock" to lock controls or "unlock" to unlock controls]
PlayEffect	[nameID],[effectID],[effectType],[priority(optional)]
StopEffect	[nameID]
activateMusicCue	DEPRECATED. Does nothing.
deactivateMusicCue	DEPRECATED. Does nothing.
flashMusicCue	DEPRECATED. Does nothing.
setMusicParameter	DEPRECATED. Does nothing.
play2DAmbientSound	DEPRECATED. Does nothing.
stop2DAmbientSound	DEPRECATED. Does nothing.
play3DAmbientSound	DEPRECATED. Does nothing.
stop3DAmbientSound	DEPRECATED. Does nothing.
activateMixerProgram	DEPRECATED. Does nothing.
deactivateMixerProgram	DEPRECATED. Does nothing.
CastSkill	[skillID]
displayZoneSummary	[1 for zone start, 0 for zone end]
SetPhysicsVolumeEffect	["Push", "Attract", "Repulse", "Gravity", "Friction"],[amount],(direction x, y, z),("True" or "False
SetPhysicsVolumeStatus	["On", "Off"]
setModelToBuild	[template ID]
spawnModelBricks	[amount, from 0 to 1],[x],[y],[z]
ActivateSpawnerNetwork	[Spawner Network Name]
DeactivateSpawnerNetwork	[Spawner Network Name]
ResetSpawnerNetwork	[Spawner Network Name]
DestroySpawnerNetworkObjects	[Spawner Network Name]
Go_To_Waypoint	[Waypoint index],("true" to allow direction change, otherwise "false"),("true" to stop at waypoint,
ActivatePhysics	"true" to activate and add to world, "false" to deactivate and remove from the world

8.3 Zone (.luz)

Note:

id
`luz`

file-extension
`luz`

endian
`le`

imports

1. `../common/common`

8.3.1 Sequence

1. **[u4]** *file_version*
2. **if:** `file_version >= 36`
   [u4] *file_revision*
3. **[u4]** *zone_id*
4. **if:** `file_version >= 38`
   [common::vector3] *player_start_pos*
5. **if:** `file_version >= 38`
   [common::quaternion] *player_start_rot*
6. **[switch-on:file_version >= 37]** *num_scene_files*

Table 3: Type Cases

True	`u4`
False	`u1`

7. **[scene]** *scenes*
   repeat-expr: `num_scene_files`
8. **[u1]** *num_zone_boundary_lines*
9. **[boundary_info]** *boundary_lines*
   repeat-expr: `num_zone_boundary_lines`
10. **[common::u1_str]** *raw_filename*
11. **if:** `file_version >= 31`
    [common::u1_str] *zone_name*
12. **if:** `file_version >= 31`
    [common::u1_str] *zone_description*

13. **if:** `file_version >= 32`
    [transition_data] *zone_transition_data*
14. **if:** `file_version >= 35`
    [u4] *path_chunk_size*
15. **if:** `file_version >= 35`
    [path_chunk] *path_chunk*
    Size: `path_chunk_size`

8.3.2 Type scene

Sequence

1. **[common::u1_str]** *scene_file_name*
2. **if:** `_root.file_version >= 33 or _root.file_version < 30`
   [u4] *scene_id*
3. **if:** `_root.file_version >= 33`
   [u4] *layer_id*
4. **if:** `_root.file_version >= 33`
   [common::u1_str] *scene_display_name*
5. **if:** `_root.file_version == 33`
   [common::vector3] *unknown1*
6. **if:** `_root.file_version == 33`
   [f4] *unknown2*
7. **if:** `_root.file_version >= 33`
   [u1] *scene_color_r*
8. **if:** `_root.file_version >= 33`
   [u1] *scene_color_g*
9. **if:** `_root.file_version >= 33`
   [u1] *scene_color_b*

8.3.3 Type boundary_info

Sequence

1. **[common::vector3]** *normal*
2. **[common::vector3]** *point*
3. **[u4]** *dest_zone_id*
4. **[u4]** *dest_scene_id*
5. **[common::vector3]** *spawn_loc*

8.3.4 Type `transition_data`

Sequence

1. **[u4]** *num_transitions*
2. **[transition_info]** *transitions*
   repeat-expr: `num_transitions`

8.3.5 Type `transition_info`

Sequence

1. **if:** `_root.file_version < 40`
   [common::u1_str] *unknown1*
2. **if:** `_root.file_version < 40`
   [f4] *unknown2*
3. **[transition_point]** *transition_points*
   repeat-expr: `(_root.file_version <= 33 or _root.file_version >= 39) ? 2 :  5`

8.3.6 Type `transition_point`

Sequence

1. **[u4]** *scene_id*
2. **[u4]** *layer_id*
3. **[common::vector3]** *transition_point*

8.3.7 Type `path_chunk`

Sequence

1. **[u4]** *path_chunk_version*
2. **[u4]** *num_paths*
3. **[path]** *paths*
   repeat-expr: `num_paths`

8.3.8 Type `path`

Sequence

1. **[u4]** *version*
2. **[common::u1_wstr]** *name*
3. **if:** `version <= 2`
   [common::u1_wstr] *type_name*
4. **[u4:path_type]** *type*
5. **[u4]** *flags*
6. **[u4:path_behavior]** *behavior*
7. **[switch-on:type]** *data*

Table 4: Type Cases

path_type::platform	`platform_data`
path_type::property	`property_data`
path_type::camera	`camera_data`
path_type::spawner	`spawner_data`

8. **[u4]** *num_waypoints*
9. **[waypoint]** *waypoints*
   repeat-expr: `num_waypoints`

8.3.9 Type `platform_data`

Sequence

1. **if:** `_parent.version >= 13 and _parent.version < 18`
   [common::u1_wstr] *traveling_audio_guid*
2. **if:** `_parent.version >= 18`
   [common::bool] *time_based_movement*

8.3.10 Type `property_data`

Sequence

1. **[u4:property_path_type]** *property_path_type*
2. **[u4]** *price*
3. **[u4]** *time*
4. **[u8]** *associated_zone*
5. **if:** `_parent.version >= 5`
   [common::u1_wstr] *name*
6. **if:** `_parent.version >= 5`
   [common::u4_wstr] *description*

7. **if:** `_parent.version >= 6`
   [u4:property_type] *property_type*
8. **if:** `_parent.version >= 7`
   [u4] *clone_limit*
9. **if:** `_parent.version >= 7`
   [f4] *reputation_multiplier*
10. **if:** `_parent.version >= 7`
    [u4] *period_type*
11. **if:** `_parent.version >= 8`
    [u4] *achievement_required*
12. **if:** `_parent.version >= 8`
    [common::vector3] *zone_position*
13. **if:** `_parent.version >= 8`
    [f4] *max_build_height*

8.3.11 Type `camera_data`

Sequence

1. **[common::u1_wstr]** *next_path*
2. **if:** `_parent.version >= 14`
   [common::bool] *rotate_player*

8.3.12 Type `spawner_data`

Sequence

1. **[common::lot]** *spawned_lot*
2. **[u4]** *respawn_time*
3. **[s4]** *max_to_spawn*
4. **[u4]** *num_to_maintain*
5. **[common::object_id]** *object_id*
6. **if:** `_parent.version >= 9`
   [common::bool] *activate_on_load*

8.3.13 Type `waypoint`

Sequence

1. **[common::vector3]** *position*
2. **[switch-on:_parent.type]** *data*

Table 5: Type Cases

path_type::npc	`npc_waypoint_data`
path_type::platform	`platform_waypoint_data`
path_type::camera	`camera_waypoint_data`
path_type::spawner	`spawner_waypoint_data`
path_type::racing	`racing_waypoint_data`
path_type::rail	`rail_waypoint_data`

8.3.14 Type `npc_waypoint_data`

Sequence

1. **[lnv]** *config*

8.3.15 Type `platform_waypoint_data`

Sequence

1. **[common::quaternion_wxyz]** *rotation*
2. **[common::bool]** *lock_player*
3. **[f4]** *speed*
4. **[f4]** *wait*
5. **if:** `_parent._parent.version >= 13`
   [common::u1_wstr] *depart_audio_guid*
6. **if:** `_parent._parent.version >= 13`
   [common::u1_wstr] *arrive_audio_guid*

8.3.16 Type `camera_waypoint_data`

Sequence

1. **[common::quaternion_wxyz]** *rotation*
2. **[f4]** *time*
3. **[f4]** *fov*
4. **[f4]** *tension*
5. **[f4]** *continuity*
6. **[f4]** *bias*

8.3.17 Type `spawner_waypoint_data`

Sequence

1. **[common::quaternion_wxyz]** *rotation*
2. **[lnv]** *config*

8.3.18 Type `racing_waypoint_data`

Sequence

1. **[common::quaternion_wxyz]** *rotation*
2. **[common::bool]** *is_reset_node*
3. **[common::bool]** *is_non_horizontal_camera*
4. **[f4]** *plane_width*
5. **[f4]** *plane_height*
6. **[f4]** *shortest_distance_to_end*

8.3.19 Type `rail_waypoint_data`

Sequence

1. **[common::quaternion_wxyz]** *rotation*
2. **if:** `_parent._parent.version > 16`
   [f4] *speed*
3. **[lnv]** *config*

8.3.20 Type `lnv`

Sequence

1. **[u4]** *num_entries*
2. **[lnv_entry]** *entries*
   repeat-expr: `num_entries`

8.3.21 Type `lnv_entry`

Sequence

1. **[common::u1_wstr]** *name*
2. **[common::u1_wstr]** *type_value*

8.3.22 Enum path_type

0
: npc

1
: platform

2
: property

3
: camera

4
: spawner

5
: buildarea

6
: racing

7
: rail

8.3.23 Enum path_behavior

0
: loop

1
: bounce

2
: once

8.3.24 Enum property_path_type

0
: bounded

1
: entire_zone

2
: generated_rectangle

8.3.25 Enum property_type

0
premiere

1
prize

2
lup

3
headspace

8.4 Level (.lvl)

Note:

- It seems the structure is split in chunks marked by "CHNK", somewhat similar to the IFF file format
- It seems Chunks can only begin on addresses % 16 == 0, if the chunk wouldn't start on one padding is inserted until it matches
- Padding always seems to be the 0xcd byte, but that's probably just a side effect of not writing data to it

---

Note:

id
lvl

file-extension
lvl

endian
le

imports

1. ../common/common

8.4.1 Sequence

1. **[fib_chunk]** *fib_chunk*

8.4.2 Type chunk

Sequence

1. `'CHNK'` *header*
2. **[u4:chunk_type]** *type*
3. **[u2]** *header_version*
4. **[u2]** *data_version*
5. **[u4]** *size*
6. **[u4]** *data_offset*

Instance data

[switch-on:type]
Position: `data_offset`

Table 6: Type Cases

chunk_type::fib	`fib_data`
chunk_type::environment	`environment_data`
chunk_type::object	`object_data`
chunk_type::particle	`particle_data`

8.4.3 Type fib_chunk

Sequence

1. `'CHNK'` *header*
2. `[232, 3, 0, 0]` *type*
3. **[u2]** *header_version*
4. **[u2]** *data_version*
5. **[u4]** *size*
6. **[u4]** *data_offset*

Instance data

[fib_data]
Position: `data_offset`

8.4.4 Type `fib_data`

Sequence

1. **[u4]** *version*
2. **[u4]** *revision*
3. **[u4]** *ofs_environment_chunk*
4. **[u4]** *ofs_object_chunk*
5. **[u4]** *ofs_particle_chunk*

Instance `environment_chunk`

if: `ofs_environment_chunk != 0`
[chunk]
Position: `ofs_environment_chunk`

Instance `object_chunk`

if: `ofs_object_chunk != 0`
[chunk]
Position: `ofs_object_chunk`

Instance `particle_chunk`

if: `ofs_particle_chunk != 0`
[chunk]
Position: `ofs_particle_chunk`

8.4.5 Type `environment_data`

Sequence

1. **[u4]** *ofs_lighting*
2. **[u4]** *ofs_skydome*
3. **[u4]** *ofs_editor_settings*

Instance `lighting_info`

[lighting_info]
Position: `ofs_lighting`

Instance `skydome_info`

[skydome_info]
Position: `ofs_skydome`

Instance `editor_settings`

if: `_root.fib_chunk.data.version >= 37`
[editor_settings]
Position: `ofs_editor_settings`

8.4.6 Type `lighting_info`

Sequence

1. **if:** `_root.fib_chunk.data.version >= 45`
   [f4] *blend_time*
2. **[f4]** *ambient*
   repeat-expr: `3`
3. **[f4]** *specular*
   repeat-expr: `3`
4. **[f4]** *upper_hemi*
   repeat-expr: `3`
5. **[common::vector3]** *position*
6. **if:** `_root.fib_chunk.data.version >= 39`
   [scene_draw_distances] *min_draw_distances*
7. **if:** `_root.fib_chunk.data.version >= 39`
   [scene_draw_distances] *max_draw_distances*
8. **if:** `_root.fib_chunk.data.version >= 40`
   [cull_data] *cull_data*
9. **if:** `_root.fib_chunk.data.version >= 31 and _root.fib_chunk.data.version < 39`
   [f4] *fog_near*
10. **if:** `_root.fib_chunk.data.version >= 31 and _root.fib_chunk.data.version < 39`
    [f4] *fog_far*
11. **if:** `_root.fib_chunk.data.version >= 31`
    [f4] *fog_color*
    repeat-expr: `3`
12. **if:** `_root.fib_chunk.data.version >= 36`

[f4] *dir_light*
repeat-expr: `3`

13. **if:** `_root.fib_chunk.data.version < 42`
    [common::vector3] *start_position*
14. **if:** `_root.fib_chunk.data.version >= 33 and _root.fib_chunk.data.version < 42`
    [common::quaternion] *start_rotation*

8.4.7 Type `scene_draw_distances`

Sequence

1. **[f4]** *fog_near*
2. **[f4]** *fog_far*
3. **[f4]** *post_fog_solid*
4. **[f4]** *post_fog_fade*
5. **[f4]** *static_object_distance*
6. **[f4]** *dynamic_object_distance*

8.4.8 Type `cull_data`

Sequence

1. **[u4]** *num_cull_vals*
2. **[cull_val]** *cull_vals*
   repeat-expr: `num_cull_vals`

8.4.9 Type `cull_val`

Sequence

1. **[u4]** *group_id*
2. **[f4]** *min*
3. **[f4]** *max*

8.4.10 Type `skydome_info`

Sequence

1. **[common::u4_str]** *filename*
2. **if:** `_root.fib_chunk.data.version >= 34`
   [common::u4_str] *sky_layer_filename*
3. **if:** `_root.fib_chunk.data.version >= 34`
   [common::u4_str] *ring_layer_0_filename*

4. **if:** `_root.fib_chunk.data.version >= 34`
   [common::u4_str] *ring_layer_1_filename*
5. **if:** `_root.fib_chunk.data.version >= 34`
   [common::u4_str] *ring_layer_2_filename*
6. **if:** `_root.fib_chunk.data.version >= 34`
   [common::u4_str] *ring_layer_3_filename*

8.4.11 Type `editor_settings`

Sequence

1. **[u4]** *chunk_size*
2. **[u4]** *num_saved_colors*
3. **[color]** *saved_colors*
   repeat-expr: `num_saved_colors`

8.4.12 Type `color`

Sequence

1. **[f4]** *r*
2. **[f4]** *g*
3. **[f4]** *b*

8.4.13 Type `object_data`

Sequence

1. **[u4]** *num_objects*
2. **[object_info]** *objects*
   repeat-expr: `num_objects`

8.4.14 Type `object_info`

Sequence

1. **[common::object_id]** *object_id*
2. **[common::lot]** *lot*
3. **if:** `_root.fib_chunk.data.version >= 38`
   [u4:node_type] *obj_type*
4. **if:** `_root.fib_chunk.data.version >= 32`
   [u4] *glom_id*
5. **[common::vector3]** *position*

6. **[common::quaternion_wxyz]** *rotation*
7. **[f4]** *scale*
8. **[u4]** *config_size*
9. **Size:** `config_size * 2`
10. **if:** `_root.fib_chunk.data.version >= 7`
    [render_technique] *render_technique*

8.4.15 Type `render_technique`

Sequence

1. **[u4]** *num_render_attrs*
2. **if:** `num_render_attrs > 0`
   [strz] *name*
   Size: `64`
   Encoding: `ascii`
3. **[render_attr]** *render_attrs*
   repeat-expr: `num_render_attrs`

8.4.16 Type `render_attr`

Sequence

1. **[strz]** *name*
   Size: `64`
   Encoding: `ascii`
2. **[u4]** *num_floats*
3. **[common::bool]** *is_color*
4. **[f4]** *floats*
   repeat-expr: `4`

8.4.17 Type `particle_data`

Sequence

1. **[u4]** *num_particles*
2. **[particle]** *particles*
   repeat-expr: `num_particles`

8.4.18 Type particle

Sequence

1. **if:** _root.fib_chunk.data.version >= 43
   [u2] *priority*
2. **[common::vector3]** *position*
3. **[common::quaternion_wxyz]** *rotation*
4. **[common::u4_wstr]** *effect_names*
5. **if:** _root.fib_chunk.data.version < 46
   [0, 0] *null_terminator*
6. **[common::u4_wstr]** *config_data*

8.4.19 Enum chunk_type

1000
: fib

2000
: environment

2001
: object

2002
: particle

8.4.20 Enum node_type

0
: environment_obj

1
: building

2
: enemy

3
: npc

4
: rebuilder

5
: spawned

6
: cannon

7
: bouncer

8
`exhibit`

9
`moving_platform`

10
`springpad`

11
`sound`

12
`particle`

13
`generic_placeholder`

14
`error_marker`

15
`player_start`

8.4.21 Old Format

[L:265] - ???
[std::string] - skybox
[std::string] - "(invalid)"
[std::string] - "(invalid)"
[std::string] - "(invalid)"
[std::string] - "(invalid)"
[std::string] - "(invalid)"
[L:4] - ???
[u32] - count
- **[float]** - ???
- **[float]** - ???
- **[float]** - ???

[Chunk 2001 (Objects) Structure]

8.5 Terrain (.raw)

Used for terrain data. See also: http://legouniverse.wikia.com/wiki/User_blog:Jamesster.LEGO/Terrain_files

---

Note:

id
`raw`

file-extension
`raw`

endian
`le`

imports

1. `../common/common`

8.5.1 Sequence

1. **[u2]** *version*
2. **[u1]** *dev*
3. **if:** `dev == 0`
   [u4] *num_chunks*
4. **if:** `dev == 0`
   [u4] *num_chunks_width*
5. **if:** `dev == 0`
   [u4] *num_chunks_height*
6. **if:** `dev == 0`
   [chunk] *chunks*
   repeat-expr: `num_chunks`

8.5.2 Type chunk

Sequence

1. **[u4]** *id*
2. **[u4]** *width*
3. **[u4]** *height*
4. **[f4]** *offset_world_x*
5. **[f4]** *offset_world_z*
6. **if:** `_root.version < 32`
   [u4] *shader_id*
7. **[u4]** *texture_ids*
   repeat-expr: `4`
8. **[f4]** *density*
9. **[f4]** *height_map*
   repeat-expr: `width * height`
10. **if:** `_root.version >= 32`
    [u4] *diffuse_res*
11. **if:** `_root.version >= 32`
    Size: `diffuse_res * diffuse_res * 4`
12. **if:** `_root.version < 32`
    [u1] *unknown1*
    repeat: `eos`

13. **if:** `_root.version >= 32`
    [u4] *diffuse_map_dds_size*
14. **if:** `_root.version >= 32`
    Size: `diffuse_map_dds_size`
15. **[u4]** *blend_res*
16. **Size:** `blend_res * blend_res * 4`
17. **if:** `_root.version >= 32`
    [u1] *bits*
18. **if:** `_root.version >= 32`
    [u4] *blend_map_dds_size*
19. **if:** `_root.version >= 32`
    Size: `blend_map_dds_size`
20. **[u4]** *num_flairs*
21. **[flair_attributes]** *flairs*
    repeat-expr: `num_flairs`
    Size: `36`
22. **if:** `_root.version >= 32`
    Size: `diffuse_res * diffuse_res`
23. **if:** `_root.version < 32`
    [u1] *unknown2*
    repeat: `eos`
24. **if:** `_root.version >= 32`
    [u4] *vert_size*
25. **[u2]** *mesh_vert_usage*
    repeat-expr: `vert_size`
26. **[u2]** *mesh_vert_size*
    repeat-expr: `16`
27. **[mesh_tri]** *mesh_tri*
    repeat-expr: `16`

8.5.3 Type `flair_attributes`

Sequence

1. **[u4]** *id*
2. **[f4]** *scale_factor*
3. **[common::vector3]** *pos*
4. **[common::vector3]** *rot*
5. **[u1]** *color_r*
6. **[u1]** *color_g*
7. **[u1]** *color_b*

8.5.4 Type `mesh_tri`

Sequence

1. **[u2]** *mesh_tri_list_size*
2. **[u2]** *mesh_tri_list*
   repeat-expr: `mesh_tri_list_size`

CHAPTER

NINE

OTHER FORMATS

9.1 Environment (.evc)

plain text, xml structure, environment-config?

9.2 Assets (.zal/.ast)

Plain text, lists paths to additional files (to load?), one line for each file

- zal = zone asset list?
- ast = asset list?

9.3 Animations (.gfx)

Note: Used for small animations, such as minifig faces. Essentially a .swf flash file, with a different file header. To convert to a .swf file, change the "GFX" in the beginning of the file header to "FWS". See also: http://wwwimages.adobe.com/content/dam/Adobe/en/devnet/swf/pdf/swf-file-format-spec.pdf

CHAPTER

TEN

GAME MECHANICS

Note: This is a read-the-docs port of the original google docs lu_game_mechanics, written by humanoid, lcdr and others, ported by @Xiphoseer. This is currently a proof of concept and is not guaranteed to reflect the latest changes.

10.1 Skill / Behavior System

When an object in the game has a *Skill Component (9)* attached, it has one or more skills attached to it, which it can trigger. A skill is the root of a tree of behaviors that get executed once the skill is triggered.

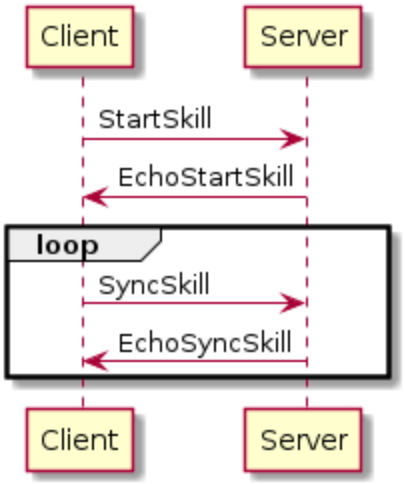
In the case of an attack by a player, these skills are executed on the client and the synchronized to the server. The server may then interpret what the client has done (and possibly sanity-check it) and execute any server side consequences of the attack. This includes updating the health of enemies for all players and smashing an object once it has been killed.

For enemies, the Combat AI triggers the skill and as such the server must execute it. That information is then synchronized to the clients for display.

10.1.1 Handling a behavior

When a skill is triggered on the client, it will send a StartSkill message. This message contains a bitstream that represents all the decisions the client has made whenever a behavior has multiple paths to continue. An example for this would be the *Area of Effect / AoE (7)* which will serialize the amount of objects within the area, and then the object id and remaining bitstream for all of these objects.

When a behavior is not completed immediately, it will serialize a `handleID`, which will then later be used to identify a SyncSkill message that serializes the continued execution. It is possible that this happens multiple times for a single skill execution.



10.1.2 Behavior Templates

Via the *BehaviorTemplate* table, each behavior is assigned a template. This template defines what the behavior does, what it serializes to the bit stream and which subsequent actions will be triggered. These templates are parameterized via the *BehaviorParameter* table.

The following is a list of (networked) behavior templates.

Basic Attack (1)

This behavior is used to deal damage to a target.

Parameters

Name	Description
dir_angle_xz	The direction to modify the attackers angle by.
dir_angle_y	The direction to modify the attackers angle by.
dir_force	The amount of force to apply towards angle specified in dir_angle_xz or dir_angle_y.
dont_apply_immune	Whether or not to apply immunity to the caster of the behavior.
max damage	The maximum amount of damage to be dealt.
min damage	The minimum amount of damage to be dealt.
on_fail_armor	The behavior to use on failure due to the target having armor.
on_fail_blocked	The behavior to use on failure due to the attack being blocked.
on_fail_immune	The behavior to use on failure due to the target being immune.
on_success	The behavior to use on success of the attack.
radius	The radius of the attack.
use_caster_velocity	Whether or not to use the casters velocity.
velocity_multiplier	Value to multiply velocity by when attacking(?).

It is unknown what the following variables are used for: - dont_apply_immune - dir_angle_xz - dir_angle_y - dir_force - radius - use_caster_velocity - velocity_multiplier

BitStream Serialization

Align to the byte boundary

[u16] - Number of bits used for this basic attack serialization and all sub-branches. (referred to as `allocatedSize`)
Save the offset at this position for use later (referred to here as `startOffset`)

Note:

If the target blocked the attack, `allocatedSize` would be a 1. If the target was immune it would be a 2. If the success state branches are reached, then this value also represents the
sum of all the sub behaviors!

[bit] - True if the attack was blocked, false otherwise.
if blocked:
    `on_fail_blocked`
    align the bitStream to `startOffset + allocatedSize` and return
[bit] - True if the the target is immune, false otherwise.
if immune:
    `on_fail_immune`
    align the bitStream to `startOffset + allocatedSize` and return
[bit] - True if the attack dealt any damage at all, false otherwise.
if any damage was done at all:
    [u32] - The amount of armor damage that was dealt.
    [u32] - Amount of life damage that was dealt.
    [bit] - Whether or not the target died from the basic attack.
[u8] - The success state of the attack.
if success state == 1:
    `on_success`
else if success state == 2:
    `on_fail_armor`
else:
    if success state != 3:
        align the bitStream to `startOffset + allocatedSize` and return
    else:
        `on_fail_immune`
align the bitStream to `startOffset + allocatedSize` and return

Note:

For serializing the behavior, the success state is determined as follows:
if *any* health damage was done at all, the success state is 1.

if *zero* health damage was done and armor damage is greater than zero, the success state is 2. Has one caveat mentioned below.

if none of the above are true and any of the following are true, the success state is 3:

- armor damage was dealt but no `on_fail_armor` behavior was present.
- the attack was not successful i.e. zero damage was done, both in armor and in health

TacArc (2)

This behavior executes the action on a group of object nearby which fit the given parameters.

Parameters

Name
action
affects_caster
angle
angle_weight
blocked action
check_env
clear_provided_target
defers_smashables
distance_weight
far_height
far_width
first_within_range
height
ignore_faction
ignore_faction_1
include_faction
include_faction1
include_faction2
include_faction3
include_faction_1
include_faction_2
include_faction_3
lower_bound
max range
max target
max targets
max_range
method
min range
miss action
near_height
near_width
offset_x
offset_y
offset_z

continues on next page

Table 1 – continued from previous page

Name
prefers_enemies
radius
run_speed
target_enemy
target_friend
target_self
target_team
upper_bound
use_attack_priority
use_picked_target
use_target_position

BitStream Serialization

hit_something= **[bit]**
if hit_something:
    if `check_env` parameter:
        [bit] - ???, always 0?
    [u32] - number of targets
        [s64] - target object id
    for each target:
        -> *action*
else:
    if `blocked_action` parameter exists:
        [bit] - is blocked
        if blocked -> *blocked action*, else -> *miss action*
    else:
        -> miss action

And (3)

This behavior executes all of its component behaviors in parallel.

Parameters

The valid parameters all take the form `behavior X` where X is a number starting at 1. Then execute the behaviors in this order. If there are more than 10, the client sorts the list by the behavior number.

Example

The Behavior 1698 will execute the behaviors in the order 375 (behavior 1), 1700 (behavior 2), 1699 (behavior 3)

BitStream Serialization

[foreach behavior in behaviors]
-> behavior

Projectile Attack (4)

This behavior is used to launch a projectile that can hit other objects.

Parameters

Name
LOT_ID
clear_provided_target
max_distance
no_ally_check
notify_target
offset_x
offset_y
offset_z
projectile_speed
projectile_type
rotate_x_degrees
spread_angle
spread_count
spread_z_load_fudge
track_radius
track_target
unauth_impact
use_high_arc
use_mouseposit
use_prediction

BitStream Serialization

[s64] - target id
projectile count = "spread_count" parameter, minimum 1
[projectile count]
    [s64] - local projectile id
        used for projectile impact message (behavior of impact message determined by projectile LOT skill)

Heal (5)

This behavior is used to heal an object.

Parameters

Name	Description
health	The amount of hearts regenerated

BitStream Serialization

This behavior does not serialize anything

Movement Switch (6)

Calls different behaviors depending on what the current movement type of the originator object is.

Parameters

Name
air_action
double_jump_action
falling_action
ground_action
jetpack_action
jump_action
moving_action

BitStream Serialization

Todo: Figure out what type id air and moving are

[u32] - movement type, 1 -> ground, 2 -> jump, 3 -> falling, 4 -> double-jump, 6 -> jetpack

Area of Effect / AoE (7)

This behavior calls the specified action on all / a maximum number of entities in the casters' radius.

Parameters

Name	Description
action	The behavior to be performed.
ignore_faction	A faction to ignore during targetting.
ignore_faction1	A faction to ignore during targetting.
ignore_faction2	A faction to ignore during targetting.
ignore_faction3	A faction to ignore during targetting.
include_faction	A faction to include in targetting
max targets	The maximum number of allowed targets.
radius	The radius to check for targets in.
target_enemy	Whether or not to target an enemy.
target_friend	Whether or not to target a friend.
target_self	Whether or not to target self.
target_team	Whether or not to target a team.
use_target_as_caster	Whether or not to use the target as the caster.
use_target_position	Whether or not to use the targets' position.

BitStream Serialization

[u32] - The number of targets.
    [s64] - The target object id.
[for target in targets]
    -> action(target)

Play Effect (8)

Plays the effect specified for the behavior in *BehaviorTemplate*

Parameters

This behavior has no additional parameters, altough the game database seems to have some wrongly attributed ones present.

BitStream Serialization

This behavior does not serialize anything

Immunity (9)

This behavior grants the player/object immunity against some actions

Parameters

Name
immune_basic_attack
immune_damage_over_time
immune_imagination_gain
immune_imagination_loss
immune_interrupt
immune_knockback
immune_quickbuild_interrupts
immune_speed
immune_stun_attack
immune_stun_equip
immune_stun_interact
immune_stun_move
immune_stun_rotate

BitStream Serialization

This behavior does not serialize anything

Damage Buff (10)

Details unknown

Damage Absorption (11)

Details unknown

Over Time (12)

Details unknown

Parameters

Name
action
delay
num_intervals

BitStream Serialization

Todo: Figure out serialization, possibly just a handleID

Imagination (13)

This behavior is used to restore imagination for an object.

Parameters

Name	Description
imagination	The amount of imagination regenerated

BitStream Serialization

Todo: Figure out serialization

Target Caster (14)

Set the caster as the target for subsequent actions.

Parameters

Name	Description
action	The action to execute with the new target set

BitStream Serialization

Todo: Figure out serialization

-> action

Stun (15)

Temporarily removes the characters ability to do certain things

Parameters

Name	Description
cant_attack	If the player is blocked from attacking
cant_equip	If the player is blocked from equipping gear
cant_interact	If the player is blocked from interacting
cant_jump	If the player is blocked from jumping
cant_move	If the player is stopped from moving
cant_turn	If the player is stopped from turning
cant_use_item	If the player is blocked from using items
dont_terminate_interact	If the stun will exit existing interactions
ignore_immunity	If the behavior will ignore immunity
stun_caster	Whether to stun the caster

Possibly deprecated

Name	Description
action	The next action to execute
duration	How long the stun will take
radius	In which radius targets will be stunned
target_enemy	Whether to target enemies
target_friend	Whether to target friends

Likely typos

Name	Description
can't_equip	Misspelling of *cant_equip*
can't_interact	Misspelling of *cant_interact*
can't_move	Misspelling of *cant_move*
can't_turn	Misspelling of *cant_turn*

BitStream Serialization

Todo: investigate

if target != self:

note that for some reason this does not work for projectiles

[bit] - ???, always False?

Duration (16)

This behavior describes that the subsequent ones are active (only) for some stretch of time.

Parameters

Name	Description
action	The behavior to execute next
duration	How long the subsequent behaviors are active
originator_is_owner	???

Possibly deprecated

Name	Description
behavior 1	Subsequent behavior?
behavior 2	Subsequent behavior?

Likely errors

Name	Description
strength	Belonging to another behavior?
angle	Belonging to another behavior?
delay	Should have been *duration*?

BitStream Serialization

Todo: investigate

-> action

Knockback (17)

This behavior results in a knockback effect on targeted players/enemies

Parameters

Name	Description
angle	The direction of the knockback
caster	whether to affect the caster ????
ignore_self	whether to affect the current object ????
relative	???
strength	how strong the knockback is

Possibly deprecated

Name	Description
time_ms	How long the knockback is in effect?

BitStream Serialization

Todo: investigate

[bit] - ???, always False?

Attack Delay (18)

When this node is reached in a behavior tree, the current traversal through this behavior branch is halted and will SyncSkill after a `delay` `num_intervals` times. The server is expected to keep track of what behavior is waiting for a sync so the sync can be done at a later time.

The server is then expected to EchoStartSkill to all connected clients.

The attack is then continued later with a SyncSkill message from the client.

See the diagram in *Skill / Behavior System* for a diagram on how this interaction looks.

For skills cast by a client, it is not advised to predict whether or not the player was actually able to cast a skill in real time. This is because you will get desync in what a client expects to happen vs what actually happened which will cause invisible enemies, mis-match health bars and further issues.

Parameters

Name	Value	Description
action	int32	The action to take when `delay` has passed
delay	float	The delay between intervals
ignore_interrupts	bool	Whether or not to ignore interrupts
num_intervals	int32	The number of times to sync this behavior

Deprecated parameters

There is 1 singular behavior that has two extra parameters, `Behavior 1` and `Behavior 2`.

This Behavior 25652, whose root is this Behavior 25607, does not appear to be used by and scripts, has no entry in the *SkillBehavior*

BitStream Serialization

[u32] - behavior handle

Car Boost (19)

Boost a player in a car for `time`. If the player does not have any boosts active, you will cast `action` otherwise you will cast `action_failed`

Parameters

Name	Description
action	Action to execute on being able to add the boost.
action_failed	Action to execute when the boost could not be activated.
active	Unused in 1.10.64.
time	The amount of time to add the boost for.

Note: The effect played when you succeed in boosting is hard coded in the client. If the `time` is greater than 3.5, `boostCam1` is played, otherwise if the `time` is greater than 2.0, `boostCam2` will play, otherwise `boostCam3` will play.

BitStream Serialization

[bit] - 1 if able to add active boost, 0 otherwise.

Fall Speed (20)

Parameters

Name	Description
percent_slowed	???

BitStream Serialization

Todo: investigate

Shield (21)

Parameters

This behavior has no parameters

BitStream Serialization

Todo: investigate

Repair Armor (22)

Parameters

Name	Description
armor	Amount of armor restored

BitStream Serialization

Todo: investigate

Speed (23)

Parameters

Name	Description
affects_caster	Whether the speed affects the caster?
attack_speed	New attack speed?
run_speed	New run speed?

BitStream Serialization

Todo: investigate

Dark Inspiration (24)

Parameters

Name	Description
action	The action to trigger next?
faction_list	???

BitStream Serialization

Todo: investigate

Loot Buff (25)

This behavior manages whether or not a player has an increased loot pickup radius. The result of this behavior is serialized in the Controllable Physics Component of the Entity

Parameters

Name	Type	Description
scale	float	The pickup radius of this buff

Relevant Serialization

Controllable Physics Component (1)

Venture Visison (26)

This behavior manages whether or not the player can see minibosses or collectibles on their personal minimap. Pet digs seem to be disabled altogether however. In the 1.10.64 client, this behavior is referenced as `ShowMinimapExtrasBehavior`

Parameters

Name	Type	Description
show_collectibles	Boolean	Whether or not to show collectibles.
show_minibosses	Boolean	Whether or not to show minibosses.
show_pet_digs	Boolean	Whether or not to show pet digs. Non-functional during live.

Relevant Game Messages

UiMessageServerToSingleClient

Amf3 Serialization

In an amf3 Array:
Associative map:
`bShowPetDigs`: boolean
`bShowMinibosses`: boolean
`bShowCollectibles`: boolean

Spawn Object (27)

Details unknown

Parameters

Name	Description
LOT_ID	The template of the object to spawn
distance	???
objectRadius	???
spawn_fail_action	???
updatePositionWithParent	???

BitStream Serialization

Todo: investigate

Lay Brick (28)

Details unknown

Parameters

Name	Description
distance	???
duration	???
forceflat	???
forcelay	???
forcestack	???
maxbricks	???
offset_forward	???
offset_up	???
templateID	???

BitStream Serialization

Todo: investigate

Switch (29)

Details unknown

Parameters

Name	Description
action_false	???
action_true	???
distance	???
faction	???
imagination	???
isEnemyFaction	???
target_has_buff	???

BitStream Serialization

Todo: investigate

state = True
if “imagination” parameter > 0 or not “isEnemyFaction” parameter:
    state= **[bit]** - switch state
if state:

-> action_true
else:
-> action_false

Buff (30)

Increases the player stats while the behavior is active

Parameters

Name	Description
armor	???
attack_speed	???
brain	???
imag	???
life	???
run_speed	???

BitStream Serialization

Todo: investigate

Jetpack (31)

Details unknown

Parameters

Name	Description
airspeed	???
bypass_checks	???
enable_hover	???
max_airspeed	???
vertical_velocity	???
warning_effect_id	???

BitStream Serialization

Todo: investigate

Skill Event (32)

Details unknown

Parameters

Name	Description
target_caster	???

BitStream Serialization

Todo: investigate

Consume Item (33)

Details unknown

Parameters

Name	Description
action_consumed	???
consume_lot	???
effect_id	???
num_to_consume	???
run_speed	???

BitStream Serialization

Todo: investigate

Skill Cast Failed (34)

Details unknown

Likely deprecated parameters

Name	Description
effect_id	???

BitStream Serialization

Todo: investigate

Imitation Skunk Stink (35)

Details unknown

Likely deprecated parameters

Name	Description
effect_id	???

BitStream Serialization

Todo: investigate

Change Idle Flags (36)

Changes the idle flags of the caster by turning the requested flags off/on. No serialization.

Parameters

Name	Description
flags_on	The idle flag to turn on
flags_off	The idle flag to turn off

Note: The values in the flags are *not* literal numbers but rather bit indexes. This means the values in the flags *must* be between the values 0 and 32 inclusive. Anything outside of this range is undefined behavior. To add onto this, the behavior is as follows - If flag is zero, use 0. - Otherwise get a value that is `(1 << (flag - 1))` and set/unset that bit

Apply Buff (37)

Details unknown

Parameters

Name	Description
add_immunity	???
apply_on_teammates	???
buff_id	???
cancel_on_damaged	???
cancel_on_death	???
cancel_on_logout	???
cancel_on_remove_buff	???
cancel_on_ui	???
cancel_on_unequip	???
cancel_on_zone	???
duration_secs	???
ignore_uncast	???
target_caster	???
use_ref_count	???

BitStream Serialization

Todo: investigate

Chain (38)

Details unknown

Parameters

Name	Description	Default value
behavior X	Behavior to execute as a chain step, should be sequential starting from 1 and increasing by 1 for each behavior.	No default
chain_delay	The delay to apply between each attack of the chain.	0.5 if field is not present.

BitStream Serialization

[u32] - The smallest behavior such that behavior n satisfies the following criteria
- `chain_delay * n < delay_since_first_behavior_called`

-> `behavior X`

Note:

- Like with attack_delay, this delay should not be simulated on a server due to latency adding onto the delay.

Change Orientation (39)

Details unknown

Parameters

Name	Description
orient_caster	If true, the `target_to_orient` is the caster and the target is `target_to_orient_to`. Otherwise flip the targets.
to_angle	If true, orient the `target_to_orient` based on angle below
angle	The angle in degrees to orient to
relative	If true, apply the above angle orientation relative to the current orientation
to_target	If true, orient the `target_to_orient` to the `target_to_orient_to`.

Note: If Both `to_target` and `to_angle` are set to true, `to_target` will take priority if there is a `target_to_orient_to`.

Removed in 1.10.64

Name	Description
behavior 1	???
behavior 2	???
behavior 3	???
to_point	???
duration	???

BitStream Serialization

Todo: investigate

Force Movement (40)

Details unknown

Parameters

Name	Description
collide_with_faction	???
duration	???
forward	???
hit_action	???
hit_action_enemy	???
hit_action_faction	???
ignore_projectile_collision	???
left	???
move_target	???
relative	???
timeout_action	???
yaw	???
yaw_abs	???

BitStream Serialization

if any of “hit_action”, “hit_action_enemy”, “hit_action_faction” is not 0:

[u32] - behavior handle

-> SyncSkill, see AirMovement for details

Interrupt (41)

This behavior interrupts a `target` which is either the actual target of this behavior, or the caster. This discrepency is controlled by the `target` parameter in the database.

Parameters

Name	Type	Description
interrupt_attack	int_bool	True to interrupt attacks
interrupt_block	int_bool	True to interrupt blocks
interrupt_charge	int_bool	True to interrupt charge-up
interupt_attack	int_bool	typo, unused parameter in 1.10.64.
interupt_charge	int_bool	type, unused parameter in 1.10.64.
target	int_bool	True if you use the target of this branch, false to target caster.

BitStream Serialization

if target != self:

[bit] - True if target is immune to stuns. Return if this is true.

if “interrupt_block” false:

[bit] - True if target is blocking interrupts. Return if this is true.

[bit] - Unused system from live that likely sent the skillUid that was interrupted? If true, serialization is as follows.

[bit] Has another interrupted skillUid

[u32] Interrupted skillUid

Alter Cooldown (42)

Details unknown

Parameters

Name	Description
add	???
amount	???

BitStream Serialization

Todo: investigate

Charge Up (43)

Details unknown

Parameters

Name	Description
action	???
max_duration	???

BitStream Serialization

Todo: investigate

Switch Multiple (44)

Details unknown, mostly used for charge up action

Parameters

Name	Description
behavior X	???
charge_time	???
distance_to_target	???
value X	???

BitStream Serialization

Todo: investigate

[float] - value
if value <= “value_1” parameter:
    -> behavior_1
else:
    -> behavior_2

Start (45)

Details unknown

Parameters

Name	Description
action	???
use_target	???

BitStream Serialization

Todo: investigate

End (46)

Details unknown

Parameters

Name	Description
start_action	???
use_target	???

BitStream Serialization

Todo: investigate

Alter Chain Delay (47)

Details unknown

Parameters

Name	Description
chain_action	???
new_delay	???

BitStream Serialization

Todo: investigate

Camera (48)

Details unknown

Parameters

Name	Description
lock_camera	???
lookat_relative	???
lookat_x	???
lookat_y	???
lookat_z	???
pos_relative	???
pos_x	???
pos_y	???
pos_z	???

BitStream Serialization

Todo: investigate

Remove Buff (49)

Details unknown

Parameters

Name	Description
buff_id	???
remove_immunity	???

BitStream Serialization

Todo: investigate

Grab (50)

Details unknown

Parameters

Name	Description
dir_angle_xz	???
dir_angle_y	???
dir_force	???

BitStream Serialization

Todo: investigate

Modular Build (51)

Details unknown

Parameters

This component has no parameters

BitStream Serialization

Todo: investigate

NPC Combat Skill (52)

Details unknown

Parameters

Name	Description
behavior	???
max range	???
min range	???
npc skill time	???

Possible errors

Name	Description
behavior 1	???

BitStream Serialization

Todo: investigate

Block (53)

Blocks `num_attacks_can_block` attacks and casts the `break_action` when you have reached the limit of attacks you can block.

Parameters

Name	Description
block_damage	True to block basic attack behaviors
block_knockback	True to block knockback behaviors
block_knockbacks	True to block knockback behaviors
block_stuns	True to block stun behaviors
break_action	The behavior to cast when the damage blocked is greater than the amount of damage this behavior allowed you to block
num_attacks_can_block	The number of attacks that can be blocked

BitStream Serialization

No serialization

Verify (54)

Details unknown

Parameters

Name	Description
action	???
blocked_action	???
check_blocking	???
check_range	???
range	???

BitStream Serialization

Todo: investigate

Taunt (55)

Details unknown

Parameters

Name	Description
threat to add	???

BitStream Serialization

Todo: investigate

Air Movement (56)

Details unknown

Parameters

Name	Description
goto_target	???
gravity_scale	???
ground_action	???
hit_action	???
hit_action_enemy	???
move_target	???
stop_input	???
timeout_action	???
timeout_ms	???
use_collision_delay	???
x_velocity	???
y_velocity	???
z_velocity	???

BitStream Serialization

Todo: investigate

Note: like Attack Delay, this causes SyncSkill messages, which use the behavior handle as ID but have the behavior to execute specified in the SyncSkill bitstream

[u32] - behavior handle
SyncSkill structure:

[u32] - behavior id
[u64] - target object id

Spawn Quickbuild (57)

Details unknown

Parameters

Name	Description
LOT_ID	???
distance	???
objectRadius	???
offsetX	???
offsetY	???
offsetZ	???
repositionPlayer	???
spawn_fail_action	???

BitStream Serialization

Todo: investigate

Pull to Point (58)

Details unknown

Parameters

Name	Description
arc_height	???
distance_offset	???

BitStream Serialization

Todo: investigate

Property Rotate (59)

Details unknown

Parameters

This behavior has no parameters

BitStream Serialization

Todo: investigate

Damage Reduction (60)

Details unknown

Name	Description
reduction_amount	???

BitStream Serialization

Todo: investigate

Property Teleport (61)

Details unknown

Name	Description
cancel_if_interacting	???
mapID	???

BitStream Serialization

Todo: investigate

Clear Target (62)

Details unknown

Name	Description
action	???
clear_if_caster	???

BitStream Serialization

Todo: investigate

Take Picture (63)

Details unknown

Name	Description
overlay_type	???
save_to_disk	???
upload_to_web	???

BitStream Serialization

Todo: investigate

Mount (64)

Details unknown

Name	Description
mount	???

BitStream Serialization

Todo: investigate

SkillSet (65)

Details unknown

Name	Description
set_id	???

BitStream Serialization

Todo: investigate

10.2 Kit Factions

The game has four so-called *Kit Factions*: **Assembly**, **Paradox**, **Sentinels** and **Venture League**. This has nothing to do with the *Factions* table, which specifies which objects can attack each other. The wiki article on Factions describes their purpose from a player perspective.

Factions are implemented using the *Flag System* and the *Mission System*. While the mission system controls which mission path you take, the flag system unlocks the faction vendors (and probably more, not tested more than that yet) to buy special gear and is used in certain *lua scripts* to determine the player's faction.

One side thing to mention is, that you can activate all faction flags without any trouble other than unlocking stuff from all factions.

10.2.1 Missions

The following missions control the faction missions:

venture	Mission 555	Mission 556	Mission 778
assembly	Mission 544	Mission 545	Mission 778
paradox	Mission 577	Mission 578	Mission 778
sentinel	Mission 566	Mission 567	Mission 778

10.2.2 Item Sets

Each faction has a unique `kitType` value in the *ItemSets* table.

Sentinel	1
Assembly	2
Paradox	3
Venture	4
Bat Lord / Mosaic Jester	5

10.2.3 factionKitID

Todo: Where is this used?

The following numbers represent the `factionKitID`:

venture	1
assembly	2
paradox	3
sentinel	4

10.3 Flag System

The flag system is used to track various YES or NO decisions within the game for each player. This includes information on the *Kit Factions*, the minimap, ingame objects that the player interacted with, the last VE-mission and more.

The flags are stored into a 12008 bit number as taskmask. Because there is no numberic datatype, which can hold up to 12008 bits, the taskmask is split up into tiny parts, each 64 bits.

Each flag is configured using the *PlayerFlags* table.

10.3.1 Derived Flag Numbers

The flag ids used for binoculars and story plaques is generally derived from the current zone ID and an additional index:

800+ID

The player has tamed the pet with Pet Component #ID

ZZNN

The player has looked through binocular NN on zones ZZ...
Source: Script `02_client/map/general/l_binoculars_client.lua`

10000+Z+N

The player has read story plaque N in zone Z
Source: Script `02_client/map/general/l_story_box_interact_client.lua`

10.3.2 Known Flags

Flag	Description	Source
0	btarr testing	1.4.49 client
1	player has entered pet ranch	1.1.18 client
2	minimap unlocked	1.1.18 client
3	activity rebuilding fail time	1.1.18 client
4	activity rebuilding fail range	1.1.18 client
5	activity shooting gallery help	1.1.18 client
6	help walking controls	1.1.18 client
7	first smashable	1.1.18 client
8	first imagination pickup	1.1.18 client
9	first damage	1.1.18 client
10	first item	1.1.18 client
11	first brick	1.1.18 client
12	first consumable	1.1.18 client
13	first equippable	1.1.18 client
14	chat help	1.1.18 client
15	first pet taming minigame	1.1.18 client
16	first pet on switch	1.1.18 client
17	first pet jumped on switch	1.1.18 client
18	first pet found treasure	1.1.18 client
19	first pet dug treasure	1.1.18 client
20	first pet owner on pet bouncer	1.1.18 client
21	first pet despawn no imagination	1.1.18 client
22	first pet selected enough bricks	1.1.18 client
23	first emote unlocked	1.1.18 client
24	GF - Pirate Rep	1.1.18 client
25	AG - bpb cinematic event	1.1.18 client
26	help jumping controls	1.1.18 client
27	help double jump controls	1.1.18 client
28	help camera controls	1.1.18 client
29	help rotate controls	1.1.18 client
30	help smash	1.1.18 client
31	Monument Intro Music Played	1.1.18 client
32	Beginning Zone Summary Displayed	1.1.18 client
33	AG - Finish Line Built	1.1.18 client
34	AG - Boss Area Found	1.1.18 client
35	AG - Landed in Battlefield	1.1.18 client
36	GF - Player has been to the ravine	1.1.18 client
37	Modular Build Started	1.1.18 client
38	Modular Build Finished click button	1.1.18 client
39	Thinking Hat received go to modular build area	1.1.18 client
40	Build Area entered mod NOT activated put on Hat	1.1.18 client
41	Hat on inside of mod build equip a module from LEG	1.1.18 client
42	Module equipped place on glowing blue spot	1.1.18 client
43	First module placed correctly now do the rest	1.1.18 client
44	Rocket complete now launch from pad	*Preconditions*
45	joined a faction	1.1.18 client
46	The player has joined the **Venture League** *kit faction*	1.1.18 client
47	The player has joined the **Assembly** *kit faction*	1.1.18 client
48	The player has joined the **Paradox** *kit faction*	1.1.18 client

continues on next page

Table 2 – continued from previous page

Flag	Description	Source
49	The player has joined the **Sentinel** *kit faction*	1.1.18 client
50	LUP World Access	1.1.18 client
51	AG first flag collected	1.1.18 client
52	tooltip talk to skyland to get hat	1.1.18 client
53	modular build player places first model in scratch	1.1.18 client
54	modular build first arrow display for module	1.1.18 client
55	AG beacon QB, so the player can always build them	1.1.18 client
56	GF Pet Dig Flag 1	1.1.18 client
57	GF Pet Dig Flag 2	1.1.18 client
58	GF Pet Dig Flag 3	1.1.18 client
59	Suppress Spaceship Cinematic Flythrough	1.1.18 client
60	GF Player Fall Death	1.1.18 client
61	GF Player can get Flag 1	1.1.18 client
62	GF Player can get Flag 2	1.1.18 client
63	GF Player can get Flag 3	1.1.18 client
64	Enter BBB from Property Edit confirmation dialog	1.1.18 client
65	AG First Combat Complete	1.1.18 client
66	AG - Complete Bob Mission	Script `client/mission_bob.lua`
67	Player can tame the lion pet	1.1.18 client
68	FV On Free the Ninjas Mission	1.1.18 client
69	First manual pet hibernate	1.1.18 client
70	First time in pet taming while having a pet out	1.1.18 client
71	Defeated maelstrom on small AG property	1.1.18 client
72	Player has completed the hammer mission	1.1.18 client
73	Placed first model on AG small property	1.1.18 client
79	Player secured property	1.1.18 client
80	Hat ON inside Property Edit	1.1.18 client
81	*Can do the Panda Race* Player has completed all missions for Brickmaster Clang	*Preconditions*
82	Player has tamed a panda	1.1.18 client
83	First 'Out of Imagination'	1.1.18 client
84	Delete Item from Inventory confirmation dialog	1.1.18 client
85	Completed Nimbus Station Race	1.1.18 client
86	First pickup when bag is full	1.1.18 client
87	First model	1.1.18 client
88	First behavior	1.1.18 client
89	First booster pack	1.1.18 client
90	First *package*	1.1.18 client
92	Delete Model from Inventory confirmation dialog	1.1.18 client
93	Delete Brick from Inventory confirmation dialog	1.1.18 client
94	Delete Behavior from Inventory confirmation dialog	1.1.18 client
95	Delete Property from Inventory confirmation dialog	1.1.18 client
96	Player tutorial mode	1.1.18 client
97	Defeat maelstrom from small NS property	1.1.18 client
98	Defeat maelstrom from small GF property	1.1.18 client
99	Defeat maelstrom from small FV property	1.1.18 client
101	Place 1st model on Property	1.1.18 client
102	place 2nd model on property	1.1.18 client

continues on next page

Table 2 – continued from previous page

Flag	Description	Source
103	place 3rd model on property	1.1.18 client
104	place 4th model on property	1.1.18 client
105	Placed first model on NS small property	1.1.18 client
106	Placed first model on GF small property	1.1.18 client
107	Placed first model on FV small property	1.1.18 client
108	Claimed AG Small Property	1.1.18 client
109	Pick Up a Model	1.1.18 client
110	Rotate a Model	1.1.18 client
111	Put Away a Model	1.1.18 client
112	Have played the LS intro cinematic	1.1.18 client
113	Player has finished AG property tutorials	1.1.18 client
114	Player can now see the news screen	1.1.18 client
115	Player is in a Foot Race	1.1.18 client
116	Player has completed the Winter Foot Race	1.4.49 client
117	The player has powered the (RtVE?) launcher with the console	*Preconditions*
118	Defeat maelstrom from medium AG property	1.4.49 client
119	Placed first model on AG medium property	1.4.49 client
120	No login fade on load	1.4.49 client
121	CP Sheild Generator flag	1.4.49 client
801	Elephant Pet	1.1.18 client
802	Not used	1.1.18 client
803	Triceratops Pet	1.1.18 client
804	Reindeer - not in live 1	1.1.18 client
805	not used	1.1.18 client
806	Skunk Pet -	1.1.18 client
807	Cat Pet	1.1.18 client
808	Not Used	1.1.18 client
809	Not Used	1.1.18 client
810	Reindeer - not in Live 1	1.1.18 client
811	Terrier Pet	1.1.18 client
812	Random unused pet	1.1.18 client
813	bunny - not used	1.1.18 client
814	Doberman Pet	1.1.18 client
815	Buffalo Pet	1.1.18 client
816	Robot Dog Pet	1.1.18 client
817	Not Used	1.1.18 client
818	European Dragon Pet	1.1.18 client
819	Tortoise Pet	1.1.18 client
820	Asian Dragon pet	1.1.18 client
821	Mantis Pet	1.1.18 client
822	Panda Pet	1.1.18 client
823	Warthog Pet	1.1.18 client
824	Crab Pet	1.1.18 client
825	Lion Pet	1.1.18 client
826	Crocodile Pet	1.1.18 client
827	Goat Pet	1.1.18 client
828	Coalessa's lion Cant Tame	1.1.18 client
829	Raindeer Pet	1.4.49 client
834	Bone Dragon Pet	1.4.49 client
1001	AG Space Ship Binoc at launch	1.1.18 client

continues on next page

Table 2 – continued from previous page

Flag	Description	Source
1002	AG Space Ship Binoc at launch platform	1.1.18 client
1003	AG Space Ship Binoc on platform to left of start	1.1.18 client
1004	AG Space Ship Binoc on platform to right of start	1.1.18 client
1005	AG Space Ship Binoc at Bob	1.1.18 client
1010	SpaceShipDaily0	1.4.49 client
1011	SpaceShipDaily1	1.4.49 client
1012	SpaceShipDaily2	1.4.49 client
1013	SpaceShipDaily3	1.4.49 client
1014	SpaceShipDaily4	1.4.49 client
1015	SpaceShipDaily5	1.4.49 client
1016	SpaceShipDaily6	1.4.49 client
1017	SpaceShipDaily7	1.4.49 client
1018	SpaceShipDaily8	1.4.49 client
1019	SpaceShipDaily9	1.4.49 client
1101	AG Battle Binoc for triceretops	1.1.18 client
1102	AG Battle Binoc at Paradox	1.1.18 client
1103	AG Battle Binoc at mission giver	1.1.18 client
1104	AG Battle Binoc at Beck	1.1.18 client
1105	AG Monument Binoc Intro	1.1.18 client
1106	AG Monument Binoc Outro	1.1.18 client
1107	AG Launch Binoc Intro	1.1.18 client
1108	AG Launch Binoc Bison	1.1.18 client
1109	AG Launch Binoc Shark	1.1.18 client
1110	Set when a player has dug once on a property	1.10.64 client
1201	NS Binoc Concert Transition	1.1.18 client
1202	NS Binoc Race Place Transition	1.1.18 client
1203	NS Binoc Brick Annex Transition	1.1.18 client
1204	NS Binoc GF Launch	1.1.18 client
1205	NS Binoc FV Launch	1.1.18 client
1206	NS Binoc Brick Annex Water	1.1.18 client
1207	NS Binoc AG Launch at Race Place	1.1.18 client
1208	NS Binoc AG Launch at Brick Annex	1.1.18 client
1209	NS Binoc AG Launch at Plaza	1.1.18 client
1210	NS Binoc TBA	1.1.18 client
1211	NS Binoc in Brick Annex looking at Pet Rock	1.1.18 client
1212	NS Flag Collectable 2 - under concert bridge	1.1.18 client
1213	NS Flag Collectable 3 - by FV launch	1.1.18 client
1214	NS Flag Collectable 4 - in plaza behind building	1.1.18 client
1215	NS Flag Collectable 5 - by GF launch	1.1.18 client
1216	NS Flag Collectable 6 - by Duck SG	1.1.18 client
1217	NS Flag Collectable 7 - by LUP launch	1.1.18 client
1218	NS Flag Collectable 8 - by NT luanch	1.1.18 client
1219	NS Flag Collectable 9 - by race build	1.1.18 client
1220	NS Flag Collectable 10 - on AG launch path	1.1.18 client
1221	NS Binoc TBA	1.1.18 client
1251	PR Binoc at launch pad	1.1.18 client
1252	PR Binoc at beginning of island B	1.1.18 client
1253	PR Binoc at first pet bouncer	1.1.18 client
1254	PR Binoc on by crows nest	1.1.18 client
1261	PR Pet Dig at beginning of Island B	1.1.18 client

continues on next page

Table 2 – continued from previous page

Flag	Description	Source
1262	PR Pet Dig at the location of old bounce back	1.1.18 client
1263	PR Pet Dig under QB bridge	1.1.18 client
1264	PR Pet Dig back side by partner bounce	1.1.18 client
1265	PR Pet Dig by launch pad	1.1.18 client
1266	PR Pet Dig by first pet bouncer	1.1.18 client
1280	Frostburgh Binoc 1	1.4.49 client
1281	Frostburgh Binoc 2	1.4.49 client
1282	Frostburgh Binoc 3	1.4.49 client
1283	Frostburgh Plaque 1	1.4.49 client
1284	Frostburgh Plaque 2	1.4.49 client
1285	Frostburgh Plaque 3	1.4.49 client
1286	Frostburgh Plaque 4	1.4.49 client
1301	GF Binoc on Landing pad	1.1.18 client
1302	GF Binoc at Ravine Start	1.1.18 client
1303	GF Binoc on top of Ravine Head	1.1.18 client
1304	GF Binoc at Turtle Area	1.1.18 client
1305	GF Binoc in tunnel to Elephants	1.1.18 client
1306	GF Binoc in Elephants area	1.1.18 client
1307	GF Binoc in racing area	1.1.18 client
1308	GF Binoc in croc area	1.1.18 client
1309	GF Binoc in jail area	1.1.18 client
1310	GF Binoc telescope next to captain jack	1.1.18 client
1401	FV Binoc at the gate	1.1.18 client
1402	FV Binoc at the tree	1.1.18 client
1403	FV Binoc in the tree	1.1.18 client
1404	FV Binoc at Panda Paw	1.1.18 client
1405	FV Binoc at the tree (behind)	1.1.18 client
1406	FV Binoc looking at Brick Fury	1.1.18 client
1407	FV Binoc above the facility	1.1.18 client
1408	FV Binoc looking up the cliff	1.1.18 client
1409	FV Binoc at the facility	1.1.18 client
1410	FV Binoc at the dragon crevice	1.1.18 client
1601	LUP Station Binoc 1	1.1.18 client
1602	LUP Station Binoc 2	1.1.18 client
1801	AM Binocular 1	1.4.49 client
1802	AM Binocular 2	1.4.49 client
1803	AM Binocular 3	1.4.49 client
1804	AM Binocular 4	1.4.49 client
1805	AM Binocular 5	1.4.49 client
1806	AM Binocular 6	1.4.49 client
1807	AM Binocular 7	1.4.49 client
1808	AM Binocular 8	1.4.49 client
1809	AM Binocular 9	1.4.49 client
1810	AM Binocular 10	1.4.49 client
1900	NT Build Finish	1.4.49 client
1901	NT Binoc Vent Overlook	1.4.49 client
1902	NT Binoc Vent Telescope	1.4.49 client
1903	NT Binoc Beam Room	1.4.49 client
1904	NT Binoc Para Entrance	1.4.49 client
1905	NT Binoc WarRoom Overlook	1.4.49 client

continues on next page

Table 2 – continued from previous page

Flag	Description	Source
1906	NT Binoc Landing Pad	1.4.49 client
1907	NT Binoc Battlefield Launchpad	1.4.49 client
1908	NT Binoc Armory Overlook	1.4.49 client
1911	NT Paradox Trial	1.4.49 client
1912	Assembly Pipe Rebuild 1	1.4.49 client
1913	Assembly Pipe Rebuild 2	1.4.49 client
1914	Assembly Pipe Rebuild 3	1.4.49 client
1915	NT Sent Shortcut 6	1.4.49 client
1916	NT Sent Shortcut 7	1.4.49 client
1917	NT Sent Shortcut 6	1.4.49 client
1918	Nexus Tower Welcome	1.4.49 client
1930	NT Vent Shortcut 1	1.4.49 client
1931	NT Vent Shortcut 2	1.4.49 client
1932	NT Vent Shortcut 3	1.4.49 client
1933	NT Vent Shortcut 4	1.4.49 client
1934	NT Vent Shortcut 5	1.4.49 client
1935	NT Assem Shortcut 1	1.4.49 client
1936	NT Assem Shortcut 2	1.4.49 client
1937	NT Para Shortcut 1	1.4.49 client
1938	NT Para Shortcut 2	1.4.49 client
1939	NT Para Shortcut 3	1.4.49 client
1940	NT Para Shortcut 4	1.4.49 client
1941	NT Sent Shortcut 1	1.4.49 client
1942	NT Sent Shortcut 2	1.4.49 client
1943	NT Sent Shortcut 3	1.4.49 client
1944	NT Sent Shortcut 4	1.4.49 client
1945	NT Sent Shortcut 5	1.4.49 client
1946	Pipe Rebuild 1	1.4.49 client
1947	Pipe Rebuild 2	1.4.49 client
1948	Pipe Rebuild 3	1.4.49 client
1949	Pipe Rebuild 4	1.4.49 client
1950	Pipe Rebuild 5	1.4.49 client
1951	Pipe Rebuild 6	1.4.49 client
1952	Banking Flag	1.4.49 client
1953	Bank Slot Flag	1.4.49 client
1954	Sentinel Trial	1.4.49 client
1955	NT Dirt Cloud 1	1.4.49 client
1956	NT Dirt Cloud 2	1.4.49 client
1957	NT Dirt Cloud 3	1.4.49 client
1958	NT Dirt Cloud 4	1.4.49 client
1959	NT Dirt Cloud 5	1.4.49 client
1960	NT Dirt Cloud 6	1.4.49 client
1961	NT Dirt Cloud 7	1.4.49 client
1962	NT Dirt Cloud 8	1.4.49 client
1963	NT Dirt Cloud 9	1.4.49 client
1964	NT Dirt Cloud 10	1.4.49 client
1965	NT Dirt Cloud 11	1.4.49 client
1966	NT Dirt Cloud 12	1.4.49 client
1967	Paradox Panel 1	1.4.49 client
1968	Paradox Panel 2	1.4.49 client

continues on next page

Table 2 – continued from previous page

Flag	Description	Source
1965	Paradox Panel 3	1.4.49 client
1970	Paradox Panel 4	1.4.49 client
1971	Paradox Panel 5	1.4.49 client
1972	Paradox Panel 6	1.4.49 client
1973	Paradox Panel 7	1.4.49 client
1974	Paradox Sentinel Spy	1.4.49 client
1975	Paradox Pet	1.4.49 client
1976	Paradox Assembly Spy	1.4.49 client
1977	Paradox Venture Spy	1.4.49 client
2095	Player has the Sensei wu daily dragon mission	1.10.64 client
1100	SS Plaque 1	1.1.18 client
1100	SS Plaque 2	1.1.18 client
1100	SS Plaque 3	1.1.18 client
1110	AG Plaque 1	1.1.18 client
1110	AG Plaque 2	1.1.18 client
1110	AG Plaque 3	1.1.18 client
1110	AG Plaque 4	1.1.18 client
1110	AG Plaque 5	1.1.18 client
1120	NS Plaque 1	1.1.18 client
1120	NS Plaque 2	1.1.18 client
1120	NS Plaque 3	1.1.18 client
1120	NS Plaque 4	1.1.18 client
1120	NS Plaque 5	1.1.18 client
1130	GF Plaque 1	1.1.18 client
1130	GF Plaque 2	1.1.18 client
1130	GF Plaque 3	1.1.18 client
1130	GF Plaque 4	1.1.18 client
1130	GF Plaque 5	1.1.18 client
1140	FV Plaque 1	1.1.18 client
1140	FV Plaque 2	1.1.18 client
1140	FV Plaque 3	1.1.18 client
1140	FV Plaque 4	1.1.18 client
1140	FV Plaque 5	1.1.18 client
1140	FV Plaque 6	1.1.18 client
1140	FV Plaque 7	1.1.18 client
1150	PC Plaque 1	1.1.18 client
1150	PC Plaque 2	1.1.18 client
1180	CP Plaque 1	1.4.49 client
1180	CP Plaque 2	1.4.49 client
1180	CP Plaque 3	1.4.49 client
1180	CP Plaque 4	1.4.49 client
1180	CP Plaque 5	1.4.49 client
1180	CP Plaque 6	1.4.49 client

10.4 Mission System

The main storyline and achievements in the game are all backed by the mission system. A mission is a collection of one or more tasks, which the player must fulfill to progress. A mission may also reward the player with items, emotes, currency, increased health, imagination or backpack storage.

A mission is registered in the *Missions* table along with information on rewards, prerequisites and the template ids of the objects offering the mission. These objects then have a *Mission Offer Component (73)* component attached to them, which describe which missions that NPC offers and accepts.

Each Task from the *MissionTasks* table with the same id as the mission has to be fulfilled for the mission to be considered as completed. There are several types of tasks available, including one general *script* type

10.4.1 Mission Task Types

Smash (0)

The player is required to smash a count of `targetValue` objects of the templates specified in `target` or `targetGroup`.

Script (1)

Complete a condition specified in `target` or `targetGroup` scripts `targetValue` times. The condition will vary drastically between scripts.

QuickBuild (2)

The player is required to quick-build a count of `targetValue` objects of the templates specified in `target` or `targetGroup`.

Collect (3)

The player needs to collect (collide with) a count of `targetValue` objects of the templates specified in `target` or `targetGroup`. The object which needs to be collected will have a *Collectible Component (23)* component attached to it, specifying the mission it belongs to.

GoToNPC (4)

The player need to go to the NPC of the template specified in `target`.

UseEmote (5)

The player needs to play any emote id within `taskParam1` near an object of the template specified in `target`.

UseConsumable (9)

The player needs to consume the template specified in `target` `targetValue` times.

UseSkill (10)

The player needs to trigger `targetValue` skill(s) from the comma-delimited set in `taskParam1`.

Example mission.

If `target` or `targetGroup` are present, then the skills in `taskParam1` must be used on those target LOTs

Example mission.

ObtainItem (11)

The player needs to somehow obtain a count of `targetValue` items of the template specified in `target`. This is usually used to implement quests, asking the player to buy something from a vendor or to pick up an item in the world.

`taskParam1` does not affect the mission progression but rather what happens to the items at mission turn in. Depending on `taskParam1`:

- 0 or no value: No extra parameters apply.
- 1: The `target` item is not taken from the players inventory on mission turn in.
- 2: The `target` item is taken from the players inventory on mission turn in.

Note:

Items are only ever removed from the inventory types ITEMS and HIDDEN, should they need to be removed

---

- 5: Items are not taken from the inventory nor will losing these items before the mission is completed decrement progress.

Discover (12)

The player needs to travel to the area specified by the `targetGroup`. Possibly related to environment triggers.

MinigameAchievement (14)

Achieve at least `targetValue` at the `targetGroup` statistic in a minigame, such as `survival_time_solo`. `target` specifies the relevant Activity ID.

Example: Avant Gardens Survival

Some minigame missions like mission 229 to *1* or *true* instead of setting them to their `targetValue` since you are intended to get this score in one attempt.

Interact (15)

Interact with the `target` template `targetValue` times.

MissionComplete (16)

The player needs to complete a count of `targetValue` of the missions specified by `target` and `targetGroup`.

EarnReputation (17)

The player needs to earn `targetValue` reputation.

CollectPowerup (21)

The player needs to collect `targetValue` powerups of the `target` or `targetGroup` LOTs.

TamePet (22)

The player needs to tame a count of `targetValue` of the pet objects specified by `target` and `targetGroup`. If `taskParam1` is set, taming must take less than that amount of seconds.

Racing (23)

Depending on `taskParam1`:

- 1: Be at or above the `targetValue` place in the race world specified by `target`.
- 2: Achieve a `targetValue` ms lap time or better in the race world specified by `target`.
- 3: Achieve a `targetValue` ms time or better in the race world specified by `target`.
- 4: Complete `targetValue` achievements from the `targetGroup`.
- 5: Achieve `targetValue` achievements of the ones in `targetGroup`.
- 6: Complete a task during while in modular building `targetValue` times.
- 10: Complete a race at the race world specified by `target` without (less than `targetValue` ???) wrecking.
- 11: Smash any smashable in any world contained in `targetGroup` `targetValue` times.
- 12: Collect `targetValue` imagination orbs in the racing worlds specified by `targetGroup`.
- 13: Enter the race world specified by `target`.
- 14: Win `targetValue` races at the world specified by `target`.
- 15: Win `targetValue` races at the worlds specified by `targetGroup`.

Note:

Task type 15 is a bitflag! The number of bits that are on dictates the progress of the mission.

- 16: Finish in last place `targetValue` times in `targetGroup` race worlds.
- 17: Smash `targetValue` of the objects specified by `targetGroup`.

Flag (24)

The player needs to activate a count of `targetValue` of the flags specified by `target` and `targetGroup`.

Place Model (25)

The player needs to place models `targetValue` times.

Add Behavior (27)

The player needs to add behaviors to models `targetValue` times.

VisitProperty (30)

The player needs to visit a count of `targetValue` properties of template `target` or `targetGroup`.

NexusTowerBrickDonation (32)

Depending on `taskParam1`:

- 0: Donate `targetValue` bricks to the NexusJawbox (what is `target=9999` ???)

10.5 Scripting (LUA)

The client and world server used the LUA scripting language to implement much of AI and special features, which are only used on some small amount of objects. These scripts are attached via a *Script Component (5)* or the configuration in the *Level (.lvl)* files.

Some considerable amount of the server side scripts have been removed from the client files and only appear as `__removed` in the database.

As the scripting engine has access to some interface with the game engine, documenting the relevant LUA-exposed functions can be of advantage in understanding the game architeture as well as facilitate implementation of a scripting engine in a server project, or even client modding.

10.5.1 Functions

On every game message to an object, the scripting engine will call the appropriate `on`*`MessageName`* function in the scripts. When a lua notfication is requested, the corresponding `notify`*`Message`* will be called.

10.5.2 Methods

- Localize(key: string)

10.5.3 GameObject

- obj:GetVar(key: string)
- obj:SetVar(key: string, value: any)
- obj:SetNetworkVar(key: string, value: any)
- obj:GetNetworkVar(key: string) -> any
- obj:SendLuaNotificationRequest{requestTarget: GameObject, messageName: string}
- obj:SendLuaNotificationCancel{requestTarget: GameObject, messageName: string}
- obj:GetRotation() -> Rotation
- obj:Exists() -> bool
- obj:GetFlag{iFlagID: int} -> Flag
- obj:GetLocationsVisited() -> Locations
- obj:SetRotation{x: float , y: float, z: float , w: float}
- obj:SetProximityRadius{iconID: int, radius: int, name: string}
- obj:UnsetProximityRadius{name: string}
- obj:CheckListOfPreconditionsFromLua{PreconditionsToCheck: ?, requestingID: GameObject}
- obj:RequestPickTypeUpdate()
- obj:GetID()
- obj:NotifyObject{ name: string, param1: int, ObjIDSender: ObjectID}

Game Messages

- SetStunned
- PlayAnimation
- PlayCinematic
- PlayNDAudioEmitter
- PlayFXEffect
- StopFXEffect
- TerminateInteraction
- PlayAnimation
- FireEventServerSide
- DisplayMessageBox

10.5.4 LEVEL

- LEVEL:CLUTEffect(clut: string, fadeDuration: int, startIntensity: float, endIntensity: float, uiOverlay: bool)
- LEVEL:GetCinematicInfo(cinematicName: string)
- LEVEL:GetCurrentZoneID()
- LEVEL:SetLights(modifyAmbientColor: bool, ambientColor: int, modifyDirectionalColor: bool, directionalColor: int, modifySpecularColor: bool, specularColor: int, modifyUpperHemiColor: bool, upperHemiColor: int, modifyDirectionalDirection: bool, directionalDirection: {x: float, y: float, z: float}, modifyFogColor: bool, fogColor: int, modifyDrawDistance: bool, fogNearMin: float, fogNearMax: float, fogFarMin: float, fogFarMax: float, postFogSolidMin: float, postFogSolidMax: float, postFogFadeMin: float, postFogFadeMax: float, staticObjectCutoffMin: float, staticObjectCutoffMax: float, dynamicObjectCutoffMin: float, dynamicObjectCutoffMax: float, modifySkyDome: bool, skyDome: string, blendTime: float)
- LEVEL:SetSkyDome(skyDome: string)

10.5.5 GAMEOBJ

- GAMEOBJ:GetControlledID() -> GameObject
- GAMEOBJ:GetZoneControlID()
- GAMEOBJ:GetTimer() -> Timer
- GAMEOBJ:GetObjectByID() -> GameObject
- GAMEOBJ:GetLocalCharID() -> ObjectID
- GAMEOBJ:DeleteObject(obj: GameObject)

10.5.6 UI

- UI:SendMessage(msg: string, data: NDGfxValue)
- UI:SendChat(ChatString: string, ChatType: int, Timestamp: int)
- UI:DisplayToolTip(strDialogText: string, strImageName: string, bShow: boolean, iTime: int)
- UI:CallService(serviceName: string, data: NDGFxValue)

10.5.7 Timer

- timer:AddTimerWithCancel(delay: float, message: string, object: GameObject)
- timer:CancelAllTimers(object: GameObject)

10.5.8 Flag

- flag.bFlag -> bool

10.5.9 Locations

- locations.locations -> list<ZoneID>

10.6 Minimaps

The background tiles for the minimaps are stored in the *client/res/maps/minimaps* folder. For each *Zone (.luz)* file there is one folder. Each of these folders contains three zoom levels:

```
$ ls
zoom_0
zoom_1
zoom_2
```

Level 2 has the highest resolution, level 0 the lowest. Each of those folders contains multiple 256x256 pixel `image_XXXX.dds` tiles that make up the map when stiched together.

10.6.1 Manual Stitching

You can use ImageMagick with the following command to stitch a single map. `W` and `H` need to be replaced by the number of tiles per row and column respectively. Usually that's the square root of the total number of tiles as all known maps are square.:

```
$ montage -tile WxH -mode concatenate image_*.dds image.png
```

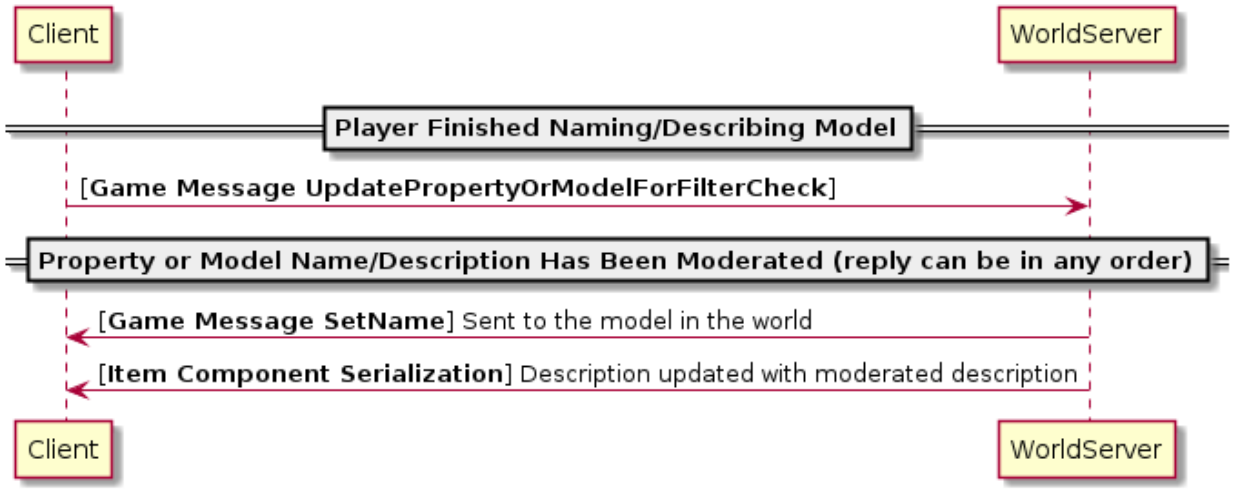
10.7 Properties

Note: This is a read-the-docs port of the original google docs lu_game_mechanics, written by humanoid, lcdr and others, ported by @Xiphoseer. This is currently a proof of concept and is not guaranteed to reflect the latest changes.

10.7.1 Naming and Describing Models

While editing on properties, players have the ability to name and describe their models. Players also have the ability to name and describe their property.

The following diagram shows the expected reply from a server in order to succeed in the naming/describing process



Networked message definitions

- Game Message UpdatePropertyOrModelForFilterCheck
- Game Message SetName

Component serialization

- *Component Serialization*

Until the SetName message **and** Item Component serialization are sent, the client will be unable to name or describe models until a time out occurs, upon which the client side name and description will revert to their previous values.

The following will happen if the client does not receive **both** of these replies:

- The name will not be updated unless the SetName Game Message is sent from the WorldServer to the Client.
- The description will not be updated unless the Item Component Serialized with the moderated description.

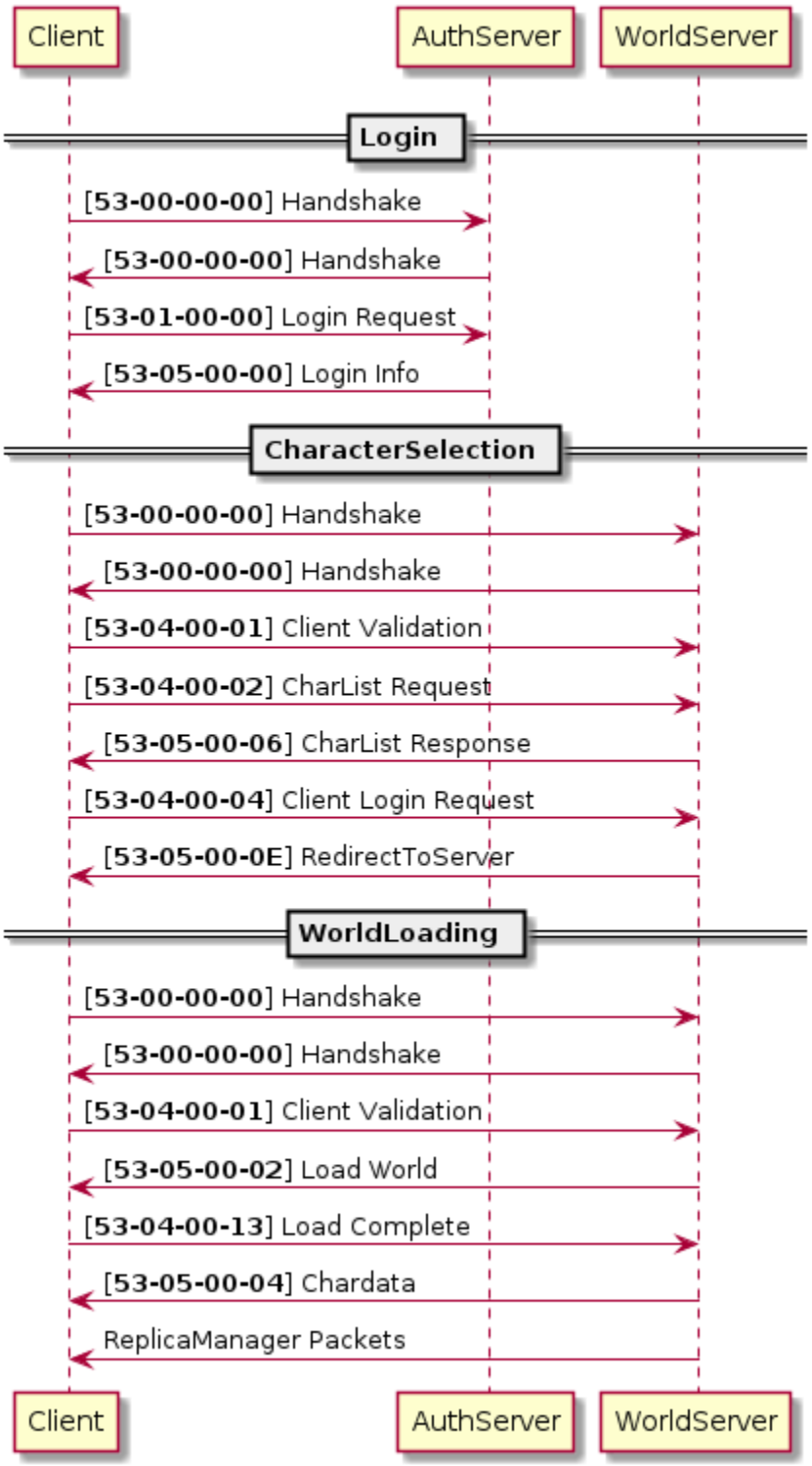
10.7.2 What should happen after receiving the Client message

If a user successfully changes the name and/or description and the values are approved and are different from their defaults, the model should become a user generated model and lose its default model property so that the name can be preserved when the model is picked up. While there is no live packet captures showing the FilterCheck Game Message being sent, other live packet captures imply that even a simple name/description change did change the models LOT to Object 14 and when placed back in the inventory, was changed to Object 6662.

This can be assumed because Entity Construction Serialization from live packet captures would show that models LOT would now be 14. Even when an entity had no description and still had the default name, the entity was still set to LOT 14.

10.8 Loading a world

This diagram describes the packets required to load a world in the client. The systems involved are an Auth Server (NetID 1), a world server (NetID 4) and a Client (NetID 5).



CHAPTER

ELEVEN

GAME DATABASE

The game database is a collection of information tables provided in the latest game version via the *CDClient.fdb Database (.fdb)* file. It used to be named *ivantest.fdb* and previous to that *ivantest.xml*.

It is used to describe the majority of content within the game, or rather their registration and properties, relying on and defining the assets that are present in the resource folder.

The client database is usually compressed within a *Pack (.pk)* file, but may be decompressed and manipulated to change the game's behavior.

As the table columns follow no common naming convention, this is very much a system which must have evolved over time, with many people working on getting all LU systems to be backed by the database.

The database is censored in some places where the client does not have and does not need some information, such as server-side script files. In those cases strings are replaced with a string like `TableName__123__column__removed`.

11.1 AICombatRoles

Column	Type
id	INTEGER
preferredRole	INTEGER
specifiedMinRangeNOUSE	FLOAT
specifiedMaxRangeNOUSE	FLOAT
specificMinRange	FLOAT
specificMaxRange	FLOAT

128 Slots

11.2 AccessoryDefaultLoc

This table provides the default relative positions of equipped objects on the player.

See also: AccessoryDefaultLoc

Column	Type
GroupID	INTEGER
Description	TEXT
Pos_X	FLOAT
Pos_Y	FLOAT
Pos_Z	FLOAT
Rot_X	FLOAT
Rot_Y	FLOAT
Rot_Z	FLOAT

256 Slots

11.3 Activities

Table 1: :widths: 15 15 30 :header-rows: 1

Column	Type	Description
ActivityID	INTEGER	The activity ID (Primary Key)
locStatus	INTEGER	The locale status of the activity
instanceMapID	INTEGER	The instance map ID of the activity
minTeams	INTEGER	The minimum number of teams needed to start the activity
maxTeams	INTEGER	The maximum number of teams needed to start the activity
minTeamSize	INTEGER	The minimum number of players allowed per team (for races this i used to put each player on their own team).
maxTeamSize	INTEGER	The maximum number of players allowed per team (for races this i used to put each player on their own team).
waitTime	INTEGER	The wait time in milliseconds before an activity starts.
startDelay	INTEGER	If all players have readied up early, this is the remaining delay to wait before starting.
requiresUniqueData	BOOLEA	If true, the activity requires unique data. Does not seem to be used in live maps and is only set for map 21.
leaderboardType	INTEGER	The leaderboard type of the activity (TODO: make an enum for this)
localize	BOOLEA	If true, the activity is localized.
optionalCostLOT	INTEGER	An item to take as an optional cost.
optionalCostCount	INTEGER	The count of `optionalCostLOT` to take.
showUIRewards	BOOLEA	Seems to always be true except for 3 activities (Activity 116, Activity 117,:ac:*999*).
CommunityActivityFlagID	INTEGER	Always null.
gate_version	TEXT	The gate version to lock the activity behind.
noTeamLootOnDeath	BOOLEA	If true, players will not drop loot on death.
optionalPercentage	FLOAT	Unknown.

256 Slots

11.4 ActivityRewards

Column	Type
objectTemplate	INTEGER
ActivityRewardIndex	INTEGER
activityRating	INTEGER
LootMatrixIndex	INTEGER
CurrencyIndex	INTEGER
ChallengeRating	INTEGER
description	TEXT

512 Slots

11.5 ActivityText

Column	Type
activityID	INTEGER
type	TEXT
localize	BOOLEAN
locStatus	INTEGER
gate_version	TEXT

4 Slots

11.6 AnimationIndex

Column	Type
animationGroupID	INTEGER
description	TEXT
groupType	TEXT

1024 Slots

11.7 Animations

Column	Type
animationGroupID	INTEGER
animation_type	TEXT
animation_name	TEXT
chance_to_play	FLOAT
min_loops	INTEGER
max_loops	INTEGER
animation_length	FLOAT
hideEquip	BOOLEAN
ignoreUpperBody	BOOLEAN
restartable	BOOLEAN
face_animation_name	TEXT
priority	FLOAT
blendTime	FLOAT

8192 Slots

11.8 BaseCombatAIComponent

Column	Type
id	INTEGER
behaviorType	INTEGER
combatRoundLength	FLOAT
combatRole	INTEGER
minRoundLength	FLOAT
maxRoundLength	FLOAT
tetherSpeed	FLOAT
pursuitSpeed	FLOAT
combatStartDelay	FLOAT
softTetherRadius	FLOAT
hardTetherRadius	FLOAT
spawnTimer	FLOAT
tetherEffectID	INTEGER
ignoreMediator	BOOLEAN
aggroRadius	FLOAT
ignoreStatReset	BOOLEAN
ignoreParent	BOOLEAN

256 Slots

11.9 BehaviorEffect

Column	Type
effectID	INTEGER
effectType	TEXT
effectName	TEXT
trailID	INTEGER
pcreateDuration	FLOAT
animationName	TEXT
attachToObject	BOOLEAN
boneName	TEXT
useSecondary	BOOLEAN
cameraEffectType	INTEGER
cameraDuration	FLOAT
cameraFrequency	FLOAT
cameraXAmp	FLOAT
cameraYAmp	FLOAT
cameraZAmp	FLOAT
cameraRotFrequency	FLOAT
cameraRoll	FLOAT
cameraPitch	FLOAT
cameraYaw	FLOAT
AudioEventGUID	TEXT
renderEffectType	INTEGER
renderEffectTime	FLOAT
renderStartVal	FLOAT
renderEndVal	FLOAT
renderDelayVal	FLOAT
renderValue1	FLOAT
renderValue2	FLOAT
renderValue3	FLOAT
renderRGBA	TEXT
renderShaderVal	INTEGER
motionID	INTEGER
meshID	INTEGER
meshDuration	FLOAT
meshLockedNode	TEXT

16384 Slots

11.10 BehaviorParameter

This table contains the parameters for any behavior in the game. Depending on the Template of the behavior, this may be information on subsequent behaviors, durations, flags or even imagination, armor or health to apply.

Usually, there will be multiple rows for a single `behaviorID`, but all with a different `parameterID`.

Column	Type
behaviorID	INTEGER
parameterID	TEXT
value	FLOAT

65536 Slots

11.11 BehaviorTemplate

Column	Type
behaviorID	INTEGER
templateID	INTEGER
effectID	INTEGER
effectHandle	TEXT

65536 Slots

11.12 BehaviorTemplateName

Column	Type
templateID	INTEGER
name	TEXT

128 Slots

11.13 Blueprints

Column	Type
id	BIGINT
name	TEXT
description	TEXT
accountid	BIGINT
characterid	BIGINT
price	INTEGER
rating	INTEGER
categoryid	INTEGER
lxfpath	TEXT
deleted	BOOLEAN
created	BIGINT
modified	BIGINT

2 Slots

11.14 BrickColors

Column	Type
id	INTEGER
red	FLOAT
green	FLOAT
blue	FLOAT
alpha	FLOAT
legopaletteid	INTEGER
description	TEXT
validTypes	INTEGER
validCharacters	INTEGER
factoryValid	BOOLEAN

64 Slots

11.15 BrickIDTable

Column	Type
NDObjectID	INTEGER
LEGOBrickID	INTEGER

2048 Slots

11.16 BuffDefinitions

Column	Type
ID	INTEGER
Priority	FLOAT
UIIcon	TEXT

128 Slots

11.17 BuffParameters

Column	Type
BuffID	INTEGER
ParameterName	TEXT
NumberValue	FLOAT
StringValue	TEXT
EffectID	INTEGER

128 Slots

11.18 Camera

Column	Type
camera_name	TEXT
pitch_angle_tolerance	FLOAT
starting_zoom	FLOAT
zoom_return_modifier	FLOAT
pitch_return_modifier	FLOAT
tether_out_return_modifier	FLOAT
tether_in_return_multiplier	FLOAT
verticle_movement_dampening_modifier	FLOAT
return_from_incline_modifier	FLOAT
horizontal_return_modifier	FLOAT
yaw_behavior_speed_multiplier	FLOAT
camera_collision_padding	FLOAT
glide_speed	FLOAT
fade_player_min_range	FLOAT
min_movement_delta_tolerance	FLOAT
min_glide_distance_tolerance	FLOAT
look_forward_offset	FLOAT
look_up_offset	FLOAT
minimum_vertical_dampening_distance	FLOAT
maximum_vertical_dampening_distance	FLOAT
minimum_ignore_jump_distance	FLOAT
maximum_ignore_jump_distance	FLOAT
maximum_auto_glide_angle	FLOAT
minimum_tether_glide_distance	FLOAT
yaw_sign_correction	FLOAT
set_1_look_forward_offset	FLOAT
set_1_look_up_offset	FLOAT
set_2_look_forward_offset	FLOAT
set_2_look_up_offset	FLOAT
set_0_speed_influence_on_dir	FLOAT
set_1_speed_influence_on_dir	FLOAT
set_2_speed_influence_on_dir	FLOAT
set_0_angular_relaxation	FLOAT
set_1_angular_relaxation	FLOAT
set_2_angular_relaxation	FLOAT
set_0_position_up_offset	FLOAT
set_1_position_up_offset	FLOAT
set_2_position_up_offset	FLOAT
set_0_position_forward_offset	FLOAT
set_1_position_forward_offset	FLOAT
set_2_position_forward_offset	FLOAT
set_0_FOV	FLOAT
set_1_FOV	FLOAT
set_2_FOV	FLOAT
set_0_max_yaw_angle	FLOAT

continues on next page

Table 3 – continued from previous page

Column	Type
set_1_max_yaw_angle	FLOAT
set_2_max_yaw_angle	FLOAT
set_1_fade_in_camera_set_change	INTEGER
set_1_fade_out_camera_set_change	INTEGER
set_2_fade_in_camera_set_change	INTEGER
set_2_fade_out_camera_set_change	INTEGER
input_movement_scalar	FLOAT
input_rotation_scalar	FLOAT
input_zoom_scalar	FLOAT
minimum_pitch_desired	FLOAT
maximum_pitch_desired	FLOAT
minimum_zoom	FLOAT
maximum_zoom	FLOAT
horizontal_rotate_tolerance	FLOAT
horizontal_rotate_modifier	FLOAT

16 Slots

11.19 CelebrationParameters

Column	Type
id	INTEGER
animation	TEXT
backgroundObject	INTEGER
duration	FLOAT
subText	TEXT
mainText	TEXT
iconID	INTEGER
celeLeadIn	FLOAT
celeLeadOut	FLOAT
cameraPathLOT	INTEGER
pathNodeName	TEXT
ambientR	FLOAT
ambientG	FLOAT
ambientB	FLOAT
directionalR	FLOAT
directionalG	FLOAT
directionalB	FLOAT
specularR	FLOAT
specularG	FLOAT
specularB	FLOAT
lightPositionX	FLOAT
lightPositionY	FLOAT
lightPositionZ	FLOAT
blendTime	FLOAT
fogColorR	FLOAT
fogColorG	FLOAT

continues on next page

Table 4 – continued from previous page

Column	Type
fogColorB	FLOAT
musicCue	TEXT
soundGUID	TEXT
mixerProgram	TEXT

16 Slots

11.20 ChoiceBuildComponent

Column	Type
id	INTEGER
selections	TEXT
imaginationOverride	INTEGER

8 Slots

11.21 CollectibleComponent

Column	Type
id	INTEGER
requirement_mission	INTEGER

128 Slots

11.22 ComponentsRegistry

Column	Type
id	INTEGER
component_type	INTEGER
component_id	INTEGER

32768 Slots

11.23 ControlSchemes

Column	Type
control_scheme	INTEGER
scheme_name	TEXT
rotation_speed	FLOAT
walk_forward_speed	FLOAT
walk_backward_speed	FLOAT
walk_strafe_speed	FLOAT
walk_strafe_forward_speed	FLOAT
walk_strafe_backward_speed	FLOAT
run_backward_speed	FLOAT
run_strafe_speed	FLOAT
run_strafe_forward_speed	FLOAT
run_strafe_backward_speed	FLOAT
keyboard_zoom_sensitivity	FLOAT
keyboard_pitch_sensitivity	FLOAT
keyboard_yaw_sensitivity	FLOAT
mouse_zoom_wheel_sensitivity	FLOAT
x_mouse_move_sensitivity_modifier	FLOAT
y_mouse_move_sensitivity_modifier	FLOAT
freecam_speed_modifier	FLOAT
freecam_slow_speed_multiplier	FLOAT
freecam_fast_speed_multiplier	FLOAT
freecam_mouse_modifier	FLOAT
gamepad_pitch_rot_sensitivity	FLOAT
gamepad_yaw_rot_sensitivity	FLOAT
gamepad_trigger_sensitivity	FLOAT

16 Slots

11.24 CurrencyDenominations

This table contains the games coin currency denominations. Each row maps a currency amount to a LOT which is the item that is spawned when said value is dropped. An example is the LOT for one coin which is what is spawned when 1 coin is dropped.

Column	Type	Description
value	INTEGER	The currency amount in in game value
objectid	INTEGER	The LOT of the coin this denomination represents

Allocated rows: 16 Slots

11.25 CurrencyTable

Column	Type
currencyIndex	INTEGER
npcminlevel	INTEGER
minvalue	INTEGER
maxvalue	INTEGER
id	INTEGER

128 Slots

11.26 DBExclude

This table marks what `columns` in what `table` to exclude from the database. The column can be a `*` to denote to exclude the whole table.

Column	Type	Description
table	TEXT	The table affected by the exclude
column	TEXT	The column to exclude, or the whole table

Allocated rows: 128 Slots

11.27 DeletionRestrictions

Column	Type
id	INTEGER
restricted	BOOLEAN
ids	TEXT
checkType	INTEGER
localize	BOOLEAN
locStatus	INTEGER
gate_version	TEXT

16 Slots

11.28 DestructibleComponent

Column	Type
id	INTEGER
faction	INTEGER
factionList	TEXT
life	INTEGER
imagination	INTEGER
LootMatrixIndex	INTEGER
CurrencyIndex	INTEGER
level	INTEGER
armor	FLOAT
death_behavior	INTEGER
isnpc	BOOLEAN
attack_priority	INTEGER
isSmashable	BOOLEAN
difficultyLevel	INTEGER

2048 Slots

11.29 DevModelBehaviors

Column	Type
ModelID	INTEGER
BehaviorID	INTEGER

0 Slots

11.30 Emotes

Column	Type
id	INTEGER
animationName	TEXT
iconFilename	TEXT
channel	TEXT
command	TEXT
locked	BOOLEAN
localize	BOOLEAN
locStatus	INTEGER
gate_version	TEXT

512 Slots

11.31 EventGating

Column	Type
eventName	TEXT
date_start	BIGINT
date_end	BIGINT

8 Slots

11.32 ExhibitComponent

Column	Type
id	INTEGER
length	FLOAT
width	FLOAT
height	FLOAT
offsetX	FLOAT
offsetY	FLOAT
offsetZ	FLOAT
fReputationSizeMultiplier	FLOAT
fImaginationCost	FLOAT

16 Slots

11.33 Factions

Column	Type
faction	INTEGER
factionList	TEXT
factionListFriendly	BOOLEAN
friendList	TEXT
enemyList	TEXT

128 Slots

11.34 FeatureGating

Column	Type
featureName	TEXT
major	INTEGER
current	INTEGER
minor	INTEGER
description	TEXT

64 Slots

11.35 FlairTable

Column	Type
id	INTEGER
asset	TEXT

64 Slots

11.36 Icons

Column	Type
IconID	INTEGER
IconPath	TEXT
IconName	TEXT

4096 Slots

11.37 InventoryComponent

Column	Type
id	INTEGER
itemid	INTEGER
count	INTEGER
equip	BOOLEAN

1024 Slots

11.38 ItemComponent

Column	Type
id	INTEGER
equipLocation	TEXT
baseValue	INTEGER
isKitPiece	BOOLEAN
rarity	INTEGER
itemType	INTEGER
itemInfo	BIGINT
inLootTable	BOOLEAN
inVendor	BOOLEAN
isUnique	BOOLEAN
isBOP	BOOLEAN
isBOE	BOOLEAN
reqFlagID	INTEGER
reqSpecialtyID	INTEGER
reqSpecRank	INTEGER
reqAchievementID	INTEGER
stackSize	INTEGER
color1	INTEGER
decal	INTEGER
offsetGroupID	INTEGER
buildTypes	INTEGER
reqPrecondition	TEXT
animationFlag	INTEGER
equipEffects	INTEGER
readyForQA	BOOLEAN
itemRating	INTEGER
isTwoHanded	BOOLEAN
minNumRequired	INTEGER
delResIndex	INTEGER
currencyLOT	INTEGER
altCurrencyCost	INTEGER
subItems	TEXT
audioEventUse	TEXT
noEquipAnimation	BOOLEAN
commendationLOT	INTEGER
commendationCost	INTEGER
audioEquipMetaEventSet	TEXT
currencyCosts	TEXT
ingredientInfo	TEXT
locStatus	INTEGER
forgeType	INTEGER
SellMultiplier	FLOAT

8192 Slots

11.39 ItemEggData

Column	Type
id	INTEGER
chassie_type_id	INTEGER

4 Slots

11.40 ItemFoodData

Column	Type
id	INTEGER
element_1	INTEGER
element_1_amount	INTEGER
element_2	INTEGER
element_2_amount	INTEGER
element_3	INTEGER
element_3_amount	INTEGER
element_4	INTEGER
element_4_amount	INTEGER

0 Slots

11.41 ItemSetSkills

Column	Type
SkillSetID	INTEGER
SkillID	INTEGER
SkillCastType	INTEGER

128 Slots

11.42 ItemSets

Column	Type
setID	INTEGER
locStatus	INTEGER
itemIDs	TEXT
kitType	INTEGER
kitRank	INTEGER
kitImage	INTEGER
skillSetWith2	INTEGER
skillSetWith3	INTEGER
skillSetWith4	INTEGER
skillSetWith5	INTEGER
skillSetWith6	INTEGER
localize	BOOLEAN
gate_version	TEXT
kitID	INTEGER
priority	FLOAT

64 Slots

11.43 JetPackPadComponent

Column	Type
id	INTEGER
xDistance	FLOAT
yDistance	FLOAT
warnDistance	FLOAT
lotBlocker	INTEGER
lotWarningVolume	INTEGER

4 Slots

11.44 LUPExhibitComponent

Column	Type
id	INTEGER
minXZ	FLOAT
maxXZ	FLOAT
maxY	FLOAT
offsetX	FLOAT
offsetY	FLOAT
offsetZ	FLOAT

1 Slots

11.45 LUPExhibitModelData

Column	Type
LOT	INTEGER
minXZ	FLOAT
maxXZ	FLOAT
maxY	FLOAT
description	TEXT
owner	TEXT

4 Slots

11.46 LUPZoneIDs

Column	Type
zoneID	INTEGER

8 Slots

11.47 LanguageType

Column	Type
LanguageID	INTEGER
LanguageDescription	TEXT

1 Slots

11.48 LevelProgressionLookup

This table contains a level and the uscore required to be that level. All `BehaviorEffect` columns are empty.

Column	Type	Description
id	INTEGER	The level
RequiredUScore	INTEGER	The required UScore to be this level
BehaviorEffect	TEXT	Unknown. Always empty in all currently known databases (Please update if we get information regarding this column!)

Allocated rows: 128 Slots

11.49 LootMatrix

Column	Type
LootMatrixIndex	INTEGER
LootTableIndex	INTEGER
RarityTableIndex	INTEGER
percent	FLOAT
minToDrop	INTEGER
maxToDrop	INTEGER
id	INTEGER
flagID	INTEGER
gate_version	TEXT

1024 Slots

11.50 LootMatrixIndex

Column	Type
LootMatrixIndex	INTEGER
inNpcEditor	BOOLEAN

1024 Slots

11.51 LootTable

Column	Type
itemid	INTEGER
LootTableIndex	INTEGER
id	INTEGER
MissionDrop	BOOLEAN
sortPriority	INTEGER

8192 Slots

11.52 LootTableIndex

Column	Type
LootTableIndex	INTEGER

1024 Slots

11.53 MinifigComponent

Column	Type
id	INTEGER
head	INTEGER
chest	INTEGER
legs	INTEGER
hairstyle	INTEGER
haircolor	INTEGER
chestdecal	INTEGER
headcolor	INTEGER
lefthand	INTEGER
righthand	INTEGER
eyebrowstyle	INTEGER
eyesstyle	INTEGER
mouthstyle	INTEGER

1024 Slots

11.54 MinifigDecals_Eyebrows

Column	Type
ID	INTEGER
High_path	TEXT
Low_path	TEXT
CharacterCreateValid	BOOLEAN
male	BOOLEAN
female	BOOLEAN

64 Slots

11.55 MinifigDecals_Eyes

Column	Type
ID	INTEGER
High_path	TEXT
Low_path	TEXT
CharacterCreateValid	BOOLEAN
male	BOOLEAN
female	BOOLEAN

64 Slots

11.56 MinifigDecals_Legs

Column	Type
ID	INTEGER
High_path	TEXT

256 Slots

11.57 MinifigDecals_Mouths

Column	Type
ID	INTEGER
High_path	TEXT
Low_path	TEXT
CharacterCreateValid	BOOLEAN
male	BOOLEAN
female	BOOLEAN

64 Slots

11.58 MinifigDecals_Torsos

Column	Type
ID	INTEGER
High_path	TEXT
CharacterCreateValid	BOOLEAN
male	BOOLEAN
female	BOOLEAN

512 Slots

11.59 MissionEmail

Column	Type
ID	INTEGER
messageType	INTEGER
notificationGroup	INTEGER
missionID	INTEGER
attachmentLOT	INTEGER
localize	BOOLEAN
locStatus	INTEGER
gate_version	TEXT

512 Slots

11.60 MissionNPCComponent

Column	Type
id	INTEGER
missionID	INTEGER
offersMission	BOOLEAN
acceptsMission	BOOLEAN
gate_version	TEXT

256 Slots

11.61 MissionTasks

Column	Type
id	INTEGER
locStatus	INTEGER
taskType	INTEGER
target	INTEGER
targetGroup	TEXT
targetValue	INTEGER
taskParam1	TEXT
largeTaskIcon	TEXT
IconID	INTEGER
uid	INTEGER
largeTaskIconID	INTEGER
localize	BOOLEAN
gate_version	TEXT

2048 Slots

11.62 MissionText

Column	Type
id	INTEGER
story_icon	TEXT
missionIcon	TEXT
offerNPCIcon	TEXT
IconID	INTEGER
state_1_anim	TEXT
state_2_anim	TEXT
state_3_anim	TEXT

continues on next page

Table 6 – continued from previous page

Column	Type
state_4_anim	TEXT
state_3_turnin_anim	TEXT
state_4_turnin_anim	TEXT
onclick_anim	TEXT
CinematicAccepted	TEXT
CinematicAcceptedLeadin	FLOAT
CinematicCompleted	TEXT
CinematicCompletedLeadin	FLOAT
CinematicRepeatable	TEXT
CinematicRepeatableLeadin	FLOAT
CinematicRepeatableCompleted	TEXT
CinematicRepeatableCompletedLeadin	FLOAT
AudioEventGUID_Interact	TEXT
AudioEventGUID_OfferAccept	TEXT
AudioEventGUID_OfferDeny	TEXT
AudioEventGUID_Completed	TEXT
AudioEventGUID_TurnIn	TEXT
AudioEventGUID_Failed	TEXT
AudioEventGUID_Progress	TEXT
AudioMusicCue_OfferAccept	TEXT
AudioMusicCue_TurnIn	TEXT
turnInIconID	INTEGER
localize	BOOLEAN
locStatus	INTEGER
gate_version	TEXT

2048 Slots

11.63 Missions

Column	Type
id	INTEGER
defined_type	TEXT
defined_subtype	TEXT
UISortOrder	INTEGER
offer_objectID	INTEGER
target_objectID	INTEGER
reward_currency	BIGINT
LegoScore	INTEGER
reward_reputation	BIGINT
isChoiceReward	BOOLEAN
reward_item1	INTEGER
reward_item1_count	INTEGER
reward_item2	INTEGER
reward_item2_count	INTEGER
reward_item3	INTEGER
reward_item3_count	INTEGER

continues on next page

Table 7 – continued from previous page

Column	Type
reward_item4	INTEGER
reward_item4_count	INTEGER
reward_emote	INTEGER
reward_emote2	INTEGER
reward_emote3	INTEGER
reward_emote4	INTEGER
reward_maximagination	INTEGER
reward_maxhealth	INTEGER
reward_maxinventory	INTEGER
reward_maxmodel	INTEGER
reward_maxwidget	INTEGER
reward_maxwallet	BIGINT
repeatable	BOOLEAN
reward_currency_repeatable	BIGINT
reward_item1_repeatable	INTEGER
reward_item1_repeat_count	INTEGER
reward_item2_repeatable	INTEGER
reward_item2_repeat_count	INTEGER
reward_item3_repeatable	INTEGER
reward_item3_repeat_count	INTEGER
reward_item4_repeatable	INTEGER
reward_item4_repeat_count	INTEGER
time_limit	INTEGER
isMission	BOOLEAN
missionIconID	INTEGER
prereqMissionID	TEXT
localize	BOOLEAN
inMOTD	BOOLEAN
cooldownTime	BIGINT
isRandom	BOOLEAN
randomPool	TEXT
UIPrereqID	INTEGER
gate_version	TEXT
HUDStates	TEXT
locStatus	INTEGER
reward_bankinventory	INTEGER

2048 Slots

11.64 ModelBehavior

Column	Type
id	INTEGER
definitionXMLfilename	TEXT

8 Slots

11.65 ModularBuildComponent

Column	Type
id	INTEGER
buildType	INTEGER
xml	VARCHAR
createdLOT	INTEGER
createdPhysicsID	INTEGER
AudioEventGUID_Snap	TEXT
AudioEventGUID_Complete	TEXT
AudioEventGUID_Present	TEXT

4 Slots

11.66 ModuleComponent

Column	Type
id	INTEGER
partCode	INTEGER
buildType	INTEGER
xml	VARCHAR
primarySoundGUID	TEXT
assembledEffectID	INTEGER

512 Slots

11.67 MotionFX

Column	Type
id	INTEGER
typeID	INTEGER
slamVelocity	FLOAT
addVelocity	FLOAT
duration	FLOAT
destGroupName	TEXT
startScale	FLOAT
endScale	FLOAT
velocity	FLOAT
distance	FLOAT

8 Slots

11.68 MovementAIComponent

Column	Type
id	INTEGER
MovementType	TEXT
WanderChance	FLOAT
WanderDelayMin	FLOAT
WanderDelayMax	FLOAT
WanderSpeed	FLOAT
WanderRadius	FLOAT
attachedPath	TEXT

128 Slots

11.69 MovingPlatforms

Column	Type
id	INTEGER
platformIsSimpleMover	BOOLEAN
platformMoveX	FLOAT
platformMoveY	FLOAT
platformMoveZ	FLOAT
platformMoveTime	FLOAT
platformStartAtEnd	BOOLEAN
description	TEXT

64 Slots

11.70 NpcIcons

Column	Type
id	INTEGER
color	INTEGER
offset	FLOAT
LOT	INTEGER
Texture	TEXT
isClickable	BOOLEAN
scale	FLOAT
rotateToFace	BOOLEAN
compositeHorizOffset	FLOAT
compositeVertOffset	FLOAT
compositeScale	FLOAT
compositeConnectionNode	TEXT
compositeLOTMultiMission	INTEGER
compositeLOTMultiMissionVentor	INTEGER
compositeIconTexture	TEXT

256 Slots

11.71 ObjectBehaviorXREF

Column	Type
LOT	INTEGER
behaviorID1	BIGINT
behaviorID2	BIGINT
behaviorID3	BIGINT
behaviorID4	BIGINT
behaviorID5	BIGINT
type	INTEGER

0 Slots

11.72 ObjectBehaviors

Column	Type
BehaviorID	BIGINT
xmldata	VARCHAR

4 Slots

11.73 ObjectSkills

Column	Type
objectTemplate	INTEGER
skillID	INTEGER
castOnType	INTEGER
AICombatWeight	INTEGER

4096 Slots

11.74 Objects

Column	Type
id	INTEGER
name	TEXT
placeable	BOOLEAN
type	TEXT
description	TEXT
localize	BOOLEAN
npcTemplateID	INTEGER
displayName	TEXT
interactionDistance	FLOAT
nametag	BOOLEAN
_internalNotes	TEXT
locStatus	INTEGER
gate_version	TEXT
HQ_valid	BOOLEAN

16384 Slots

11.75 PackageComponent

Column	Type
id	INTEGER
LootMatrixIndex	INTEGER
packageType	INTEGER

256 Slots

11.76 PetAbilities

Column	Type
id	INTEGER
AbilityName	TEXT
ImaginationCost	INTEGER
locStatus	INTEGER

4 Slots

11.77 PetComponent

Column	Type
id	INTEGER
minTameUpdateTime	FLOAT
maxTameUpdateTime	FLOAT
percentTameChance	FLOAT
tamability	FLOAT
elementType	INTEGER
walkSpeed	FLOAT
runSpeed	FLOAT
sprintSpeed	FLOAT
idleTimeMin	FLOAT
idleTimeMax	FLOAT
petForm	INTEGER
imaginationDrainRate	FLOAT
AudioMetaEventSet	TEXT
buffIDs	TEXT

64 Slots

11.78 PetNestComponent

Column	Type
id	INTEGER
ElementalType	INTEGER

1 Slots

11.79 PhysicsComponent

Column	Type
id	INTEGER
static	FLOAT
physics_asset	TEXT
jump	FLOAT
doublejump	FLOAT
speed	FLOAT
rotSpeed	FLOAT
playerHeight	FLOAT
playerRadius	FLOAT
pcShapeType	INTEGER
collisionGroup	INTEGER
airSpeed	FLOAT
boundaryAsset	TEXT
jumpAirSpeed	FLOAT
friction	FLOAT
gravityVolumeAsset	TEXT

8192 Slots

11.80 PlayerFlags

Configuration for the *Flag System*.

Column	Type	Description
id	INTEGER	A flag id.
SessionOnly	BOOLEAN	Whether or not this flag is set only for a play session.
OnlySetByServer	BOOLEAN	Whether or not this flag is only set by the server.
SessionZoneOnly	BOOLEAN	Whether or not this flag is only set for the session in the current zone(?)

Allocated space: 512 Slots

11.81 PlayerStatistics

Column	Type
statID	INTEGER
sortOrder	INTEGER
locStatus	INTEGER
gate_version	TEXT

32 Slots

11.82 PossessableComponent

Column	Type
id	INTEGER
controlSchemeID	INTEGER
minifigAttachPoint	TEXT
minifigAttachAnimation	TEXT
minifigDetachAnimation	TEXT
mountAttachAnimation	TEXT
mountDetachAnimation	TEXT
attachOffsetFwd	FLOAT
attachOffsetRight	FLOAT
possessionType	INTEGER
wantBillboard	BOOLEAN
billboardOffsetUp	FLOAT
depossessOnHit	BOOLEAN
hitStunTime	FLOAT
skillSet	INTEGER

16 Slots

11.83 Preconditions

This tables defines the preconditions as used by the `reqPrecondition` field of the *ItemComponent* table.

Column	Type
id	INTEGER
type	INTEGER
targetLOT	TEXT
targetGroup	TEXT
targetCount	INTEGER
iconID	INTEGER
localize	BOOLEAN
validContexts	BIGINT
locStatus	INTEGER
gate_version	TEXT

512 Slots

11.83.1 Column `type`

0	Item Equipped
1	Item Not Equipped
2	Player has item
3	Player does not have item
4	Player has achievement `targetLOT`
5	Mission available to player
6	Player on Mission
7	Player completed mission
8	Player has a pet deployed
9	The *flag* `targetLOT` needs to be set to **True**.
10	Player within some shape thing (see Craig)
11	Player is engaged in the right kind of Build
12	Minigame Team Check
13	Player Is In Pet Taming Minigame
14	Has faction
15	Does not have faction
16	Has racing license
17	Does not have license
18	Is a LEGO Club Member
19	NoInteraction (Uchu?)
20	???
21	???
22	Player has Level `targetLOT` or greater

11.83.2 Column `validContexts`

This column is a bitmask that describes the set of circumstances in which this precondition applies.

Todo: What is the meaning of the individual bits?

11.84 PropertyEntranceComponent

Column	Type
id	INTEGER
mapID	INTEGER
propertyName	TEXT
isOnProperty	BOOLEAN
groupType	TEXT

16 Slots

11.85 PropertyTemplate

Column	Type
id	INTEGER
mapID	INTEGER
vendorMapID	INTEGER
spawnName	TEXT
type	INTEGER
sizecode	INTEGER
minimumPrice	INTEGER
rentDuration	INTEGER
path	TEXT
cloneLimit	INTEGER
durationType	INTEGER
achievementRequired	INTEGER
zoneX	FLOAT
zoneY	FLOAT
zoneZ	FLOAT
maxBuildHeight	FLOAT
localize	BOOLEAN
reputationPerMinute	INTEGER
locStatus	INTEGER
gate_version	TEXT

128 Slots

11.86 ProximityMonitorComponent

Column	Type
id	INTEGER
Proximities	TEXT
LoadOnClient	BOOLEAN
LoadOnServer	BOOLEAN

64 Slots

11.87 ProximityTypes

Column	Type
id	INTEGER
Name	TEXT
Radius	INTEGER
CollisionGroup	INTEGER
PassiveChecks	BOOLEAN
IconID	INTEGER
LoadOnClient	BOOLEAN
LoadOnServer	BOOLEAN

8 Slots

11.88 RacingModuleComponent

Column	Type
id	INTEGER
topSpeed	FLOAT
acceleration	FLOAT
handling	FLOAT
stability	FLOAT
imagination	FLOAT

64 Slots

11.89 RailActivatorComponent

Column	Type
id	INTEGER
startAnim	TEXT
loopAnim	TEXT
stopAnim	TEXT
startSound	TEXT
loopSound	TEXT
stopSound	TEXT
effectIDs	TEXT
preconditions	TEXT
playerCollision	BOOLEAN
cameraLocked	BOOLEAN
StartEffectID	TEXT
StopEffectID	TEXT
DamageImmune	BOOLEAN
NoAggro	BOOLEAN
ShowNameBillboard	BOOLEAN

16 Slots

11.90 RarityTable

Column	Type
id	INTEGER
randmax	FLOAT
rarity	INTEGER
RarityTableIndex	INTEGER

64 Slots

11.91 RarityTableIndex

Column	Type
RarityTableIndex	INTEGER

32 Slots

11.92 RebuildComponent

Column	Type
id	INTEGER
reset_time	FLOAT
complete_time	FLOAT
take_imagination	INTEGER
interruptible	BOOLEAN
self_activator	BOOLEAN
custom_modules	TEXT
activityID	INTEGER
post_imagination_cost	INTEGER
time_before_smash	FLOAT

256 Slots

11.93 RebuildSections

Column	Type
id	INTEGER
rebuildID	INTEGER
objectID	INTEGER
offset_x	FLOAT
offset_y	FLOAT
offset_z	FLOAT
fall_angle_x	FLOAT
fall_angle_y	FLOAT
fall_angle_z	FLOAT
fall_height	FLOAT
requires_list	TEXT
size	INTEGER
bPlaced	BOOLEAN

128 Slots

11.94 Release_Version

Column	Type
ReleaseVersion	TEXT
ReleaseDate	BIGINT

128 Slots

11.95 RenderComponent

Column	Type
id	INTEGER
render_asset	TEXT
icon_asset	TEXT
IconID	INTEGER
shader_id	INTEGER
effect1	INTEGER
effect2	INTEGER
effect3	INTEGER
effect4	INTEGER
effect5	INTEGER
effect6	INTEGER
animationGroupIDs	TEXT
fade	BOOLEAN
usedropshadow	BOOLEAN
preloadAnimations	BOOLEAN
fadeInTime	FLOAT
maxShadowDistance	FLOAT
ignoreCameraCollision	BOOLEAN
renderComponentLOD1	INTEGER
renderComponentLOD2	INTEGER
gradualSnap	BOOLEAN
animationFlag	INTEGER
AudioMetaEventSet	TEXT
billboardHeight	FLOAT
chatBubbleOffset	FLOAT
staticBillboard	BOOLEAN
LXFMLFolder	TEXT
attachIndicatorsToNode	BOOLEAN

16384 Slots

11.96 RenderComponentFlash

Column	Type
id	INTEGER
interactive	BOOLEAN
animated	BOOLEAN
nodeName	TEXT
flashPath	TEXT
elementName	TEXT
_uid	INTEGER

128 Slots

11.97 RenderComponentWrapper

Column	Type
id	INTEGER
defaultWrapperAsset	TEXT

2 Slots

11.98 RenderIconAssets

Column	Type
id	INTEGER
icon_asset	TEXT
blank_column	TEXT

1024 Slots

11.99 ReputationRewards

Column	Type
repLevel	INTEGER
sublevel	INTEGER
reputation	FLOAT

0 Slots

11.100 RewardCodes

Column	Type
id	INTEGER
code	TEXT
attachmentLOT	INTEGER
locStatus	INTEGER
gate_version	TEXT

128 Slots

11.101 Rewards

Column	Type
id	INTEGER
LevelID	INTEGER
MissionID	INTEGER
RewardType	INTEGER
value	INTEGER
count	INTEGER

16 Slots

11.102 RocketLaunchpadControlComponent

Column	Type
id	INTEGER
targetZone	INTEGER
defaultZoneID	INTEGER
targetScene	TEXT
gmLevel	INTEGER
playerAnimation	TEXT
rocketAnimation	TEXT
launchMusic	TEXT
useLaunchPrecondition	BOOLEAN
useAltLandingPrecondition	BOOLEAN
launchPrecondition	TEXT
altLandingPrecondition	TEXT
altLandingSpawnPointName	TEXT

64 Slots

11.103 SceneTable

Column	Type
sceneID	INTEGER
sceneName	TEXT

64 Slots

11.104 ScriptComponent

Column	Type
id	INTEGER
script_name	TEXT
client_script_name	TEXT

2048 Slots

11.105 SkillBehavior

Column	Type
skillID	INTEGER
locStatus	INTEGER
behaviorID	INTEGER
imaginationcost	INTEGER
cooldowngroup	INTEGER
cooldown	FLOAT
inNpcEditor	BOOLEAN
skillIcon	INTEGER
oomSkillID	TEXT
oomBehaviorEffectID	INTEGER
castTypeDesc	INTEGER
imBonusUI	INTEGER
lifeBonusUI	INTEGER
armorBonusUI	INTEGER
damageUI	INTEGER
hideIcon	BOOLEAN
localize	BOOLEAN
gate_version	TEXT
cancelType	INTEGER

2048 Slots

11.106 SmashableChain

Column	Type
chainIndex	INTEGER
chainLevel	INTEGER
lootMatrixID	INTEGER
rarityTableIndex	INTEGER
currencyIndex	INTEGER
currencyLevel	INTEGER
smashCount	INTEGER
timeLimit	INTEGER
chainStepID	INTEGER

4 Slots

11.107 SmashableChainIndex

Column	Type
id	INTEGER
targetGroup	TEXT
description	TEXT
continuous	INTEGER

1 Slots

11.108 SmashableComponent

Column	Type
id	INTEGER
LootMatrixIndex	INTEGER

8 Slots

11.109 SmashableElements

Column	Type
elementID	INTEGER
dropWeight	INTEGER

1024 Slots

11.110 SpeedchatMenu

Column	Type
id	INTEGER
parentId	INTEGER
emoteId	INTEGER
imageName	TEXT
localize	BOOLEAN
locStatus	INTEGER
gate_version	TEXT

64 Slots

11.111 SubscriptionPricing

Column	Type
id	INTEGER
countryCode	TEXT
monthlyFeeGold	TEXT
monthlyFeeSilver	TEXT
monthlyFeeBronze	TEXT
monetarySymbol	INTEGER
symbolIsAppended	BOOLEAN

4 Slots

11.112 SurfaceType

Column	Type
SurfaceType	INTEGER
FootstepNDAudioMetaEventSetName	TEXT

16 Slots

11.113 TamingBuildPuzzles

Column	Type
id	INTEGER
PuzzleModelLot	INTEGER
NPCLot	INTEGER
ValidPiecesLXF	TEXT
InvalidPiecesLXF	TEXT
Difficulty	INTEGER
Timelimit	INTEGER
NumValidPieces	INTEGER
TotalNumPieces	INTEGER
ModelName	TEXT
FullModelLXF	TEXT
Duration	FLOAT
imagCostPerBuild	INTEGER

32 Slots

11.114 TextDescription

Column	Type
TextID	INTEGER
TestDescription	TEXT

0 Slots

11.115 TextLanguage

Column	Type
TextID	INTEGER
LanguageID	INTEGER
Text	TEXT

0 Slots

11.116 TrailEffects

Column	Type
trailID	INTEGER
textureName	TEXT
blendmode	INTEGER
cardlifetime	FLOAT
colorlifetime	FLOAT
minTailFade	FLOAT
tailFade	FLOAT
max_particles	INTEGER
birthDelay	FLOAT
deathDelay	FLOAT
bone1	TEXT
bone2	TEXT
texLength	FLOAT
texWidth	FLOAT
startColorR	FLOAT
startColorG	FLOAT
startColorB	FLOAT
startColorA	FLOAT
middleColorR	FLOAT
middleColorG	FLOAT
middleColorB	FLOAT
middleColorA	FLOAT
endColorR	FLOAT
endColorG	FLOAT
endColorB	FLOAT
endColorA	FLOAT

16 Slots

11.117 UGBehaviorSounds

Column	Type
id	INTEGER
guid	TEXT
localize	BOOLEAN
locStatus	INTEGER
gate_version	TEXT

32 Slots

11.118 VehiclePhysics

Column	Type
id	INTEGER
hkxFilename	TEXT
fGravityScale	FLOAT
fMass	FLOAT
fChassisFriction	FLOAT
fMaxSpeed	FLOAT
fEngineTorque	FLOAT
fBrakeFrontTorque	FLOAT
fBrakeRearTorque	FLOAT
fBrakeMinInputToBlock	FLOAT
fBrakeMinTimeToBlock	FLOAT
fSteeringMaxAngle	FLOAT
fSteeringSpeedLimitForMaxAngle	FLOAT
fSteeringMinAngle	FLOAT
fFwdBias	FLOAT
fFrontTireFriction	FLOAT
fRearTireFriction	FLOAT
fFrontTireFrictionSlide	FLOAT
fRearTireFrictionSlide	FLOAT
fFrontTireSlipAngle	FLOAT
fRearTireSlipAngle	FLOAT
fWheelWidth	FLOAT
fWheelRadius	FLOAT
fWheelMass	FLOAT
fReorientPitchStrength	FLOAT
fReorientRollStrength	FLOAT
fSuspensionLength	FLOAT
fSuspensionStrength	FLOAT
fSuspensionDampingCompression	FLOAT
fSuspensionDampingRelaxation	FLOAT
iChassisCollisionGroup	INTEGER
fNormalSpinDamping	FLOAT
fCollisionSpinDamping	FLOAT
fCollisionThreshold	FLOAT
fTorqueRollFactor	FLOAT
fTorquePitchFactor	FLOAT
fTorqueYawFactor	FLOAT
fInertiaRoll	FLOAT
fInertiaPitch	FLOAT
fInertiaYaw	FLOAT
fExtraTorqueFactor	FLOAT
fCenterOfMassFwd	FLOAT
fCenterOfMassUp	FLOAT
fCenterOfMassRight	FLOAT
fWheelHardpointFrontFwd	FLOAT
fWheelHardpointFrontUp	FLOAT
fWheelHardpointFrontRight	FLOAT
fWheelHardpointRearFwd	FLOAT

continues on next page

Table 8 – continued from previous page

Column	Type
fWheelHardpointRearUp	FLOAT
fWheelHardpointRearRight	FLOAT
fInputTurnSpeed	FLOAT
fInputDeadTurnBackSpeed	FLOAT
fInputAccelSpeed	FLOAT
fInputDeadAccelDownSpeed	FLOAT
fInputDecelSpeed	FLOAT
fInputDeadDecelDownSpeed	FLOAT
fInputSlopeChangePointX	FLOAT
fInputInitialSlope	FLOAT
fInputDeadZone	FLOAT
fAeroAirDensity	FLOAT
fAeroFrontalArea	FLOAT
fAeroDragCoefficient	FLOAT
fAeroLiftCoefficient	FLOAT
fAeroExtraGravity	FLOAT
fBoostTopSpeed	FLOAT
fBoostCostPerSecond	FLOAT
fBoostAccelerateChange	FLOAT
fBoostDampingChange	FLOAT
fPowerslideNeutralAngle	FLOAT
fPowerslideTorqueStrength	FLOAT
iPowerslideNumTorqueApplications	INTEGER
fImaginationTankSize	FLOAT
fSkillCost	FLOAT
fWreckSpeedBase	FLOAT
fWreckSpeedPercent	FLOAT
fWreckMinAngle	FLOAT
AudioEventEngine	TEXT
AudioEventSkid	TEXT
AudioEventLightHit	TEXT
AudioSpeedThresholdLightHit	FLOAT
AudioTimeoutLightHit	FLOAT
AudioEventHeavyHit	TEXT
AudioSpeedThresholdHeavyHit	FLOAT
AudioTimeoutHeavyHit	FLOAT
AudioEventStart	TEXT
AudioEventTreadConcrete	TEXT
AudioEventTreadSand	TEXT
AudioEventTreadWood	TEXT
AudioEventTreadDirt	TEXT
AudioEventTreadPlastic	TEXT
AudioEventTreadGrass	TEXT
AudioEventTreadGravel	TEXT
AudioEventTreadMud	TEXT
AudioEventTreadWater	TEXT
AudioEventTreadSnow	TEXT
AudioEventTreadIce	TEXT
AudioEventTreadMetal	TEXT
AudioEventTreadLeaves	TEXT

continues on next page

Table 8 – continued from previous page

Column	Type
AudioEventLightLand	TEXT
AudioAirtimeForLightLand	FLOAT
AudioEventHeavyLand	TEXT
AudioAirtimeForHeavyLand	FLOAT
bWheelsVisible	BOOLEAN

32 Slots

11.119 VehicleStatMap

Column	Type
id	INTEGER
ModuleStat	TEXT
HavokStat	TEXT
HavokChangePerModuleStat	FLOAT

8 Slots

11.120 VendorComponent

This table tells a Vendor Component information about the vendor. The loot matrix index is the items the vendor sells, for example.

Column	Type	Description
id	INTEGER	The component id
buyScalar	FLOAT	The multiplier to apply to items sold by this vendor
sellScalar	FLOAT	The multiplier to apply to items sold to this vendor
refreshTimeSeconds	FLOAT	The amount of time in seconds it takes this vendors' inventory to refresh
LootMatrixIndex	INTEGER	The loot matrix index of the inventory of this vendor

Allocated rows: 128 Slots

11.121 WhatsCoolItemSpotlight

Column	Type
id	INTEGER
itemID	INTEGER
localize	BOOLEAN
gate_version	TEXT
locStatus	INTEGER

128 Slots

11.122 WhatsCoolNewsAndTips

Column	Type
id	INTEGER
iconID	INTEGER
type	INTEGER
localize	BOOLEAN
gate_version	TEXT
locStatus	INTEGER

128 Slots

11.123 WorldConfig

Column	Type
WorldConfigID	INTEGER
pegravityvalue	FLOAT
pebroadphaseworldsize	FLOAT
pegameobjscalefactor	FLOAT
character_rotation_speed	FLOAT
character_walk_forward_speed	FLOAT
character_walk_backward_speed	FLOAT
character_walk_strafe_speed	FLOAT
character_walk_strafe_forward_speed	FLOAT
character_walk_strafe_backward_speed	FLOAT
character_run_backward_speed	FLOAT
character_run_strafe_speed	FLOAT
character_run_strafe_forward_speed	FLOAT
character_run_strafe_backward_speed	FLOAT
global_cooldown	FLOAT
characterGroundedTime	FLOAT
characterGroundedSpeed	FLOAT
globalImmunityTime	FLOAT

continues on next page

Table 9 – continued from previous page

Column	Type
character_max_slope	FLOAT
defaultrespawntime	FLOAT
mission_tooltip_timeout	FLOAT
vendor_buy_multiplier	FLOAT
pet_follow_radius	FLOAT
character_eye_height	FLOAT
flight_vertical_velocity	FLOAT
flight_airspeed	FLOAT
flight_fuel_ratio	FLOAT
flight_max_airspeed	FLOAT
fReputationPerVote	FLOAT
nPropertyCloneLimit	INTEGER
defaultHomespaceTemplate	INTEGER
coins_lost_on_death_percent	FLOAT
coins_lost_on_death_min	INTEGER
coins_lost_on_death_max	INTEGER
character_votes_per_day	INTEGER
property_moderation_request_approval_cost	INTEGER
property_moderation_request_review_cost	INTEGER
propertyModRequestsAllowedSpike	INTEGER
propertyModRequestsAllowedInterval	INTEGER
propertyModRequestsAllowedTotal	INTEGER
propertyModRequestsSpikeDuration	INTEGER
propertyModRequestsIntervalDuration	INTEGER
modelModerateOnCreate	BOOLEAN
defaultPropertyMaxHeight	FLOAT
reputationPerVoteCast	FLOAT
reputationPerVoteReceived	FLOAT
showcaseTopModelConsiderationBattles	INTEGER
reputationPerBattlePromotion	FLOAT
coins_lost_on_death_min_timeout	FLOAT
coins_lost_on_death_max_timeout	FLOAT
mail_base_fee	INTEGER
mail_percent_attachment_fee	FLOAT
propertyReputationDelay	INTEGER
LevelCap	INTEGER
LevelUpBehaviorEffect	TEXT
CharacterVersion	INTEGER
LevelCapCurrencyConversion	INTEGER

1 Slots

11.124 ZoneLoadingTips

Column	Type
id	INTEGER
zoneid	INTEGER
imagelocation	TEXT
localize	BOOLEAN
gate_version	TEXT
locStatus	INTEGER
weight	INTEGER
targetVersion	TEXT

64 Slots

11.125 ZoneSummary

Column	Type
zoneID	INTEGER
type	INTEGER
value	INTEGER
_uniqueID	INTEGER

64 Slots

11.126 ZoneTable

This table contains information about a zone configuration.

Column	Type	Description
zoneID	INTEGER	A unique zone id
locStatus	INTEGER	The localization status
zoneName	TEXT	The zone name
scriptID	INTEGER	The zone script ID
ghostdistance_min	FLOAT	The ghosting distance outer ring range.
ghostdistance	FLOAT	The ghosting distance inner ring range.
population_soft_cap	INTEGER	The maximum number of Civilians allowed in a zone.
population_hard_cap	INTEGER	The absolute maximum number of users allowed in a world. Not even a GameMaster can get in if the world population is at this value.
DisplayDescription	TEXT	The display description of this zone.
mapFolder	TEXT	Unknown use but is always NULL.
smashableMinDistance	FLOAT	The minimum distance for a smashable to render? Only used in Shooting Galleries.
smashableMaxDistance	FLOAT	The maximum distance for a smashable to render? Only used in Shooting Galleries.
mixerProgram	TEXT	Unknown definition.
clientPhysicsFramerate	TEXT	The clients' physics framerate. Has values of either NULL or high.
serverPhysicsFramerate	TEXT	The servers' physics framerate. Has values of either NULL or high.
zoneControlTemplate	INTEGER	The zone control LOT.
widthInChunks	INTEGER	The width of the zone in chunks. A zone is always a square but not necessarily a cube. A chunk is…
heightInChunks	INTEGER	The height of the zone in chunks. A chunk is…
petsAllowed	BOOLEAN	If true, pets are allowed to be deployed on this zone.
localize	BOOLEAN	Unknown definition
fZoneWeight	FLOAT	Unknown definition
thumbnail	TEXT	The thumbnail for this zone during rocket transitions.
PlayerLoseCoinsOnDeath	BOOLEAN	If true, players lose their coins on death. The number of coins is controled by *WorldConfig*.
disableSaveLoc	BOOLEAN	If true, save the players location while in this world.
teamRadius	FLOAT	Unknown, values can be -1, 200 and NULL. Avant Gardens is the only zone with a value of 200.
gate_version	TEXT	The gating version of this zone.
mountsAllowed	BOOLEAN	If true, mounts are allowed to be used in this zone.

Hash bucket count: 2048 slots

11.127 brickAttributes

Column	Type
ID	INTEGER
icon_asset	TEXT
display_order	INTEGER
locStatus	INTEGER

32 Slots

11.128 dtproperties

Column	Type
id	INTEGER
objectid	INTEGER
property	TEXT
value	TEXT
uvalue	TEXT
lvalue	NOTHING
version	INTEGER

0 Slots

11.129 mapAnimationPriorities

Column	Type
id	INTEGER
name	TEXT
priority	FLOAT

16 Slots

11.130 mapAssetType

Column	Type
id	INTEGER
label	TEXT
pathdir	TEXT
typelabel	TEXT

16 Slots

11.131 mapIcon

Column	Type
LOT	INTEGER
iconID	INTEGER
iconState	INTEGER

4096 Slots

11.132 mapItemTypes

Column	Type
id	INTEGER
description	TEXT
equipLocation	TEXT

32 Slots

11.133 mapRenderEffects

Column	Type
id	INTEGER
gameID	INTEGER
description	TEXT

16 Slots

11.134 mapShaders

Column	Type
id	INTEGER
label	TEXT
gameValue	INTEGER
priority	INTEGER

128 Slots

11.135 mapTextureResource

Column	Type
id	INTEGER
texturepath	TEXT
SurfaceType	INTEGER

128 Slots

11.136 map_BlueprintCategory

Column	Type
id	INTEGER
description	TEXT
enabled	BOOLEAN

1 Slots

11.137 sysdiagrams

The table can be ignored because it was generated by Database Diagrams, to store the diagram data. Database Diagrams is a Microsoft SQL Server feature, with which you can create an ERM and the database will be generated automatically in the background.

Column	Type
name	TEXT
principal_id	INTEGER
diagram_id	INTEGER
version	INTEGER
definition	TEXT

1 Slots

CHAPTER

TWELVE

COMPONENTS

Every object within the game is made up of several components. These components are relevant to how the object gets serialized from the server to the client, but also attach specific functionality and behaviors to the object. Components may react to game messages but also express other aspects of the game.

As per this test object's description, the internal name of a component class would have been `LWO`*`Name`*`Component` as in `LWOModelBuilderComponent`.

Note: Some objects have a component id of 0 in the registry. This means the object has that component, but there is no data associated with it. For example, Entity Object 3002 must get a ScriptComponent, but it will just be an empty script.

12.1 Controllable Physics Component (1)

This component expresses that the object has a physics model attached to it, which may be controlled or manipulated to walk or move around. The most common example for this component is the player object, which gets its input from the users keyboard. Other objects such as enemies may have this component too.

12.1.1 Relevant Database Tables

This component uses the *PhysicsComponent* table.

12.1.2 Component Construction

ControllablePhysicsConstruction

12.1.3 Component Serialization

ControllablePhysicsSerialization

12.2 Render Component (2)

This Component is responsable for displaying objects within the game. The component may contain information on the model to render in the game world, an item for display within the inventory or in editors, animation and shader information and more.

12.2.1 Relevant Database Tables

This component uses the *RenderComponent* table.

12.3 Simple Physics Component (3)

This component expresses that the model has simple physics, which allows for collision to happen with that object. It may use a simple 3-dimensional shape to do that.

I assume that objects with this component usually don't move around much.

12.3.1 Relevant Database Tables

This component uses the *PhysicsComponent* table.

12.4 Character Component (4)

This component does not have any client database table associated with it, as it represents and manages the state of the character of some player. It holds information such as the lego score (U-Score), account information and the passport statistics.

There is a very strange struct in the serialization, notably the `TransitionState` in the Character Component. It is a 2 bit enum defined as the following:

[uint2_t] - TransitionState
if TransitionState == 1
    [uint16_t] - lastCustomBuildParts (presumably the rocket they are arriving on)
        [wchar] - wCharacterOfTheAboveString

12.4.1 Component Dependencies

Possession Control Component (110)
Level Progression Component (109)
Player Forced Movement Component (106)

12.4.2 Component Construction

Possession Control
Level Progression
Player Forced Movement
Character

12.4.3 Component Serialization

Possession Control
Level Progression
Player Forced Movement
Character

12.4.4 Relevant Game Messages

Server received

SetTooltipFlag
ModifyPlayerZoneStatistic
UpdatePlayerStatistic
SetEmotesEnabled

12.4.5 Component XML Format

`char` - Character Component data
`attr acct` - account ID
`attr cc` - Currency
`atrr cm` - Maximum Currency
`attr co` - Unknown, related to claim codes?
`attr edit` - Unknown, Maybe related to HF editor?
`attr ft` - FreeToPlay status?
`attr gid` - Guild ID
`attr gm` - GM level
`attr gn` - Guild name
`attr lcbp` - modular info of last used rocket
`attr llog` - Timestamp of last login as this character
`attr lrx` - Last respawn point position x
`attr lry` - Last respawn point position y
`attr lrz` - Last respawn point position z
`attr lrrw` - Last respawn point rotation w
`attr lrrx` - Last respawn point rotation x
`attr lrry` - Last respawn point rotation y
`attr lrrz` - Last respawn point rotation z
`attr ls` - Lego score/Universe score.
`attr lzcs` - Last Zone Check Sum, stored as an int32_t

`attr lzid` - The last zone clone ID, instance ID and zone ID concatenated into 1 64 bit number. See *this footnote* for more info.
`attr lzrw` - Last world rotation w
`attr lzrx` - Last world rotation x
`attr lzry` - Last world rotation y
`attr lzrz` - Last world rotation z
`attr lzx` - Last world position x
`attr lzy` - Last world position y
`attr lzz` - Last world position z
`attr mldt` - "Prop mod last display time"
`attr stt` - Player stats. See *this footnote* for more information about the format.
`attr time` - Total time played, in seconds.
`attr ttip` - "tool tip flags"
`attr v` - Unknown, maybe version? Always 3 in caps
`attr vd` - Unknown, some packet cap values are 15368, 15318, 15367
- `ue` - Unlocked emotes
  - `e` - An unlocked emote
    - `attr id` - Emote ID
- `vl` - Visited worlds
  - `l` - A visited world
    - `attr cid` - Clone ID (used for properties, 0 if not a property)
    - `attr id` - World ID.
- `zs` - World Statistics
  - `s` - Statistics for a world
    - `attr ac` - Achievements collected
    - `attr bc` - Bricks collected
    - `attr cc` - Coins collected
    - `attr es` - Enemies smashed
    - `attr map` - ID of the world the statistics are for
    - `attr qbc` - Quick build count

Note:

`lzid` a binary concatenation of world ID, world instance and world clone, e.g:
lzid = `2341502167811299`
hex representation of lzid = `00 08 51 95 74 f4 04 e3`
hex representation of lzid, byte reversed (= packet byte order, Little Endian) = `e3 04 f4 74 95 51 08 00`
World ID = `e3 04`
World Instance = `f4 74`
World Clone = `95 51 08 00`

12.4.6 Character Statistics Format

The character statistics are formatted as follows with a semicolon delimiting each statistic, including the last one. Fill in empty statistics with a zero.

Example:

`10809;543;106;43;257;3;41;0;532;236;123;32403;1;58;7;55;101;111;0;0;0;0;0;0;0;0;0;`

All stats are `uint64_t` except where noted otherwise:

CurrencyCollected
BricksCollected (`int64_t`)
SmashablesSmashed
QuickBuildsCompleted
EnemiesSmashed
RocketsUsed
MissionsCompleted
PetsTamed
ImaginationPowerUpsCollected
LifePowerUpsCollected
ArmorPowerUpsCollected
MetersTraveled
TimesSmashed
TotalDamageTaken
TotalDamageHealed
TotalArmorRepaired
TotalImaginationRestored
TotalImaginationUsed
DistanceDriven
TimeAirborneInCar
RacingImaginationPowerUpsCollected
RacingImaginationCratesSmashed
RacingCarBoostsActivated
RacingTimesWrecked
RacingSmashablesSmashed
RacesFinished
FirstPlaceRaceFinishes

12.4.7 Tooltip flags

Todo: bitmask?

12.5 Script Component (5)

The script component allows for very fine grained control over the behavior of the object. Each object may have a server and client side script attached which can recieve and send game messages, start and stop timers and manipulate the world and other objects. These scripts are written in LUA and some examples are found in the `res/scripts` directory of an unpacked client.

12.5.1 Relevant Database Tables

This component uses the *ScriptComponent* table.

12.6 Bouncer Component (6)

A bouncer is an object that launches the player to a specific other location nearby, by throwing that player through the air.

12.7 Destroyable Component (7)

An object with this component may be destroyed by attacking it, and will drop some specific loot when destroyed. The loot matrix to be used is configurable in this component.

The component also stores the health, imagination and armor of the object alongside the faction and what happends when the object dies.

Faction in this case does not represent the Nexus Force player faction, but rather groups of objects that can destroy only some other groups of objects. For example, players could not hit each other, as could stromlings. But players could destroy stromlings and the other way around.

12.7.1 Relevant Database Tables

This component uses the *DestructibleComponent* table.

12.7.2 Relevant Game Messages

- SetStatusImmunity

12.7.3 XML Serialization `<dest>`

This component is serialized to XML to store its data. The attributes are:

am
: Maximum Armor

ac
: Current Armor

d
: Dead

hc
: Current Health

hm
: Maximum Health

ic
: Current Imagination

im
: Maximum imagination.

imm
: Immune

rsh
: Respawn Health

rsi
: Respawn Imagination

12.8 Ghost Component (8)

Details unknown

12.9 Skill Component (9)

This component expresses, that the object can trigger a *skill* that manipulates the world around it. A skill is, generally speaking, the root of a behavior tree that may be executed.

Execution may be triggered in different ways, such as equipping an item, attacking another object in the game, or clicking a button in the hotbar.

Skills also contain information on which icon this skill/attack has, as well as what amount of damage the skill will deal or how much health, armor or imaginagtion it will restore.

12.9.1 Relevant Database Tables

This component uses the following tables:

- *ObjectSkills*
- *SkillBehavior*

12.9.2 Relevant Game Messages

- EchoStartSkill
- StartSkill
- SelectSkill
- AddSkill
- RemoveSkill

12.9.3 XML Serialization `<skil>`

Note: What kind of skills, active ones? Why would they be saved? Action bar skills and skill uses are handled using different packets, so what would this be?

This was empty in the packet, if you find a sample that isn't empty please add content.

12.10 Spawner Component (10)

A spawńer is an object that creates a specified object in a specified place under some condition.

12.11 Item Component (11)

This component represents an item which may be held in your inventory. It contains information on maximum stack size, requirement, cost, type and more.

12.11.1 Relevant Database Tables

This component uses the *ItemComponent* table.

12.11.2 Component Construction

Item

12.11.3 Component Serialization

Item

12.12 Rebuild Component (12)

A rebuild object is an object that has fallen to pieces and may be reconstructed to its complete form by the player.

12.12.1 Relevant Database Tables

This component uses the *RebuildComponent* table.

12.13 Rebuild Start Component (13)

Details unknown

12.14 Rebuild Activator Component (14)

Details unknown

12.15 Icon Only Component (15)

Details unknown

12.15.1 Relevant Database Tables

This component uses the *RenderComponent* table.

12.16 Vendor Component (16)

A vendor is some object which can be interacted with to open a trade window, where you can buy items from the collection of that vendor, or sell any of your existing items.

12.16.1 Relevant Database Tables

This component uses the *VendorComponent* table.

12.17 Inventory Component (17)

The inventory component holds and manages which items an object has on itself.

The inventory of the player. This is actually a collection of storage and does not only represent the backpack (e.g Vault items are also in here).

12.17.1 Relevant Database Tables

This component uses the *InventoryComponent* table.

12.17.2 Relevant Game Messages

- PopEquippedItemsState
- MoveItemInInventory
- AddItemToInventoryClientSync
- RemoveItemFromInventory
- EquipInventory
- UnequipInventory
- SetInventorySize
- UseNon-EquipmentItem
- MoveInventoryBatch
- MoveItemBetweenInventoryTypes
- NotifyNotEnoughInvSpace
- MarkInventoryItemAsActive

12.17.3 XML Serialization `<inv>`

csl
: LOT of the item in the consumable slot (“consumable slot lot” ?)

Storage containers `<bag>`

A storage container `<b>`

(e.g Items, Models, Vault Items, Behaviors)

m
: Size of the bag. (Amount of slots)

t
: Type of the bag. See InventoryType enum for values.

User Item groups `<grps>`

This is used to selectively display models or bricks.

A group `<grp>`

id
: ID of the group. In the captures this was usually the literal string "user_group" and a unique number.

l
: LOTs of the items in this group, separated by spaces.

n
: Displayed name of the group.

t
: Type of the group. See bag types for values.

u

The contents of the "bags"/storage containers `<items>`

These don't actually have to be items, e.g models and bricks are listed here too.

nn
: (?)

Items in the storage container `<in>`

t
: Type of the bag. See InventoryType enum for values.

An item <i>

b
: Boolean whether the item is bound. If it isn't, this attribute isn't there at all, if it is, it's set to 1.

c
: Amount of items for stackable items.

eq
: Boolean whether the item is equipped. If it isn't, this attribute isn't there at all, if it is, it's set to 1.

id
: Object ID of the item.

l
: LOT of the item. See cdclient for correct values.

s
: Slot of the item. (0-indexed)

sk
: Some kind of ID for models. Investigate. Referred to by client strings as "subkey"?

Extra info <x>

b
ma
: Module assembly info

ub
ud
ui
um
un
: UGC name (?)

uo
up

12.18 Projectile Physics Component (18)

This component manages the physics for a projectile which flies around until it hits a target or maximum distance.

12.18.1 Relevant Database Tables

This component uses the *PhysicsComponent* table.

12.19 Shooting Gallery Component (19)

Details unknown

12.20 RigidBodyPhantomPhysics Component (20)

This component is yet another modus of physics.

12.20.1 Relevant Database Tables

This component uses the *PhysicsComponent* table.

12.21 Drop Effect Component (21)

Details unknown

12.22 Chest Component (22)

Details unknown

12.23 Collectible Component (23)

This represents objects which have to be collected by the player to complete missions.

12.23.1 Relevant Database Tables

This component uses the *CollectibleComponent* table.

12.24 Blueprint Component (24)

Represents user generated content.

12.24.1 Relevant Database Tables

This component uses the *Blueprints* table.

12.25 Moving Platform Component (25)

This components allow objects to move along a fixed path.

12.25.1 Relevant Database Tables

This component uses the *MovingPlatforms* table.

12.26 Pet Component (26)

This object is a Pet. It is attached to the actual pet model when it is spawned into the game, as opposed to the *Pet Control Component (34)* which is attached to the player owning the pet.

12.26.1 Relevant Database Tables

This component uses the *PetComponent* table.

12.26.2 Relevant Game Messages

- DespawnPet
- CommandPet
- RegisterPetId
- RegisterPetDbid
- ClientNotifyPet
- NotifyPet
- RequestSetPetName
- SetPetName
- PetNameChanged
- SetPetNameModerated

12.27 Platform Boundary Component (27)

Details unknown

12.28 Module Component (28)

Objects with this component may be used as parts is modular building.

12.28.1 Relevant Database Tables

This component uses the *ModuleComponent* table.

12.29 Jet Pack Pad (29)

Objects with this component were used for jet pack launch areas and defined the areas you could fly in.

12.29.1 Relevant Database Tables

This component uses the *JetPackPadComponent* table.

12.30 Vehicle Physics Component (30)

Physics for racing vehiles.

12.30.1 Relevant Database Tables

This component uses the *VehiclePhysics* table.

12.31 MovementAI Component (31)

AI which handles how an object moves around.

12.31.1 Relevant Database Tables

This component uses the *MovementAIComponent* table.

12.32 Exhibit Component (32)

Details unknown

12.32.1 Relevant Database Tables

This component uses the *ExhibitComponent* table.

12.33 OverheadIcon Component (33)

Displays an icon over the head of a person.

12.33.1 Relevant Database Tables

This component uses the follwing tables:

- *mapIcon*
- *NpcIcons*

12.34 Pet Control Component (34)

This component is likely responsible for managing the pets that the local player has tamed. It is attached to the player as opposed to the multiple *Pet Component (26)* attached to the individual pets.

12.34.1 Relevant Game Messages

- PetResponse
- RegisterPetId
- RegisterPetDbid
- AddPetToPlayer
- ShowPetActionButton

12.34.2 XML Serialization `<pet>`

This component is serialized to XML to store its data.

a
: (?)

A pet `<p>`

id
: Pet ObjectID

l
: Pet LOT

m
: (?)

n
: Pet Name

t
: (?)

12.35 Minifig Component (35)

This component stores information on what colors and styles the object minifigure has.

12.35.1 Relevant Database Tables

- *MinifigComponent*
- *MinifigDecals_Eyebrows*
- *MinifigDecals_Eyes*
- *MinifigDecals_Legs*
- *MinifigDecals_Mouths*
- *MinifigDecals_Torsos*
- *BrickColors*

12.35.2 XML Serialization `<mf>`

This component is serialized to XML to store its data. The attributes are:

cd
: Chest Decal

es
: Eyebrow Style

ess
: Eye Style

hc
: Hair Color

hd
: Head Style

hdc
: Head Color

hs
: Hair Style

l
: Legs Color

lh
: Left Hand Color

ms
: Mouth Style

rh
: Right Hand Color

t
: Torso Color

12.36 Property Component (36)

This component stores information on player properties.

12.37 Pet Creator Component (37)

Details unknown.

12.38 Model Builder Component (38)

Details unknown.

12.39 Scripted Activity Component (39)

Details unknown.

12.40 Phantom Physics Component (40)

Details unknown

12.40.1 Relevant Database Tables

This component uses the *PhysicsComponent* table.

12.41 Springpad Component (41)

A new framework that was never really used. Is considered a physics component.

12.42 B3 Behaviors Component (42)

Likely something with the behaviors that could be added to BrickByBrick (BBB/B3) user generated models on properties.

12.43 Property Entrance Component (43)

This likely refers to the launchpads that were present on properties, as well as those leading to these properties.

12.43.1 Relevant Database Tables

This component uses the *PropertyEntranceComponent* table.

12.44 FX Component (44)

This component is likely related to some sort of effects.

12.45 Property Management Component (45)

This component is the logic for the Property Management Console. It is attached only to the Property Plaque Object. It allows the player to edit the settings for the current property.

12.46 Vehicle Physics Component (46)

Physics for racing vehiles. Likely a newer version of *Vehicle Physics Component (30)*.

12.46.1 Relevant Database Tables

This component uses the following tables:

- *VehiclePhysics*
- *PhysicsComponent*

12.47 Physics System Component (47)

This seems to be a physics component that may delegate the physics calculation to different system. It seems to be only used for testing and some automatically converted LUP objects.

12.47.1 Relevant Database Tables

This component uses the following tables:

- *PhysicsComponent*

12.48 Quick Build Component (48)

This refers to an object which may be rebuild.

12.48.1 Relevant Database Tables

This component uses the following tables:

- *RebuildComponent*

12.49 Switch Component (49)

Details unknown

12.50 Minigame Component (50)

Details unknown.

12.51 Changling Component (51)

Details unknown.

12.52 Choice Build Component (52)

Details unknown

12.52.1 Relevant Database Tables

This component uses the following tables:

- *ChoiceBuildComponent*

12.53 Package Component (53)

A package is an item the player may obtain which, when opened will give the player a random set of items according to the specified loot table.

12.53.1 Relevant Database Tables

This component uses the following tables:

- *PackageComponent*

12.54 Sound Repeater Component (54)

Details unknown.

12.55 Sound Ambient 2D Component (55)

Details unknown.

12.56 Sound Ambient 3D Component (56)

Details unknown.

12.57 Precondition Component (57)

Details unknown.

12.57.1 Relevant Database Tables

This component uses the following tables:

- *Preconditions*

12.58 Player Flags Component (58)

This component manages the player flags. See *Flag System* for the list of all known flags.

Note: Tooltip flags are unrelated to this component. See *Character Component (4)* for information regarding these.

12.58.1 Relevant Database Tables

This component uses the following tables:

- *PlayerFlags*

12.58.2 Relevant Game Messages

- SetFlag
- NotifyClientFlagChange

12.58.3 Component XML Format

`flag` - Player Flag Component data
  `f` - Player flag
  `attr id` - This flags index
  `attr v` - This flags value
  `s` - Session flag
  `attr si` - Session flag Id as a literal number (ex. 114). Each session flag gets its own `s` element.

Flags are always saved as blocks of 64 bits. When a flag is set, its index in the flags list is the flag id divided by 64, truncated to an int. The position at the index calculated above is the flag id modulo 64.

Example code to set a player flag:

```
player_flags = {}
def set_player_flag (flag_id, turn_flag_on):
    # First calculate the index, in this case it equals 17
    flag_index = int(flag_id / 64)
    # Then calculate the position, which is also 17, and set the bit at that position.
    flag_value_shifted = 1 << flag_id % 64
    # Then check if we already have flags at this flag index
    flag_to_update = player_flags.get(flag_index)
    if flag_to_update != None:
        if turn_flag_on == True:
            # Turn the bit at flag_value_shifted in flag_to_update to True
            flag_to_update = flag_to_update | flag_value_shifted
        else:
            # Turn the bit at flag_value_shifted in flag_to_update to False by inverting␣
↪the binary
            # value of flag_value_shifted and ANDing with flag_to_update
            flag_to_update = flag_to_update & ~flag_value_shifted
        player_flags[flag_index] = flag_to_update
    else:
        # Create the new flag value and insert it into the dictionary of flags
        new_flag_value = flag_value_shifted
        player_flags[flag_index] = new_flag_value

# Turns player flag 1105 on
set_player_flag(1105, True)
```

(continues on next page)

(continued from previous page)

```
# {17: 131072}
print(player_flags)
# Does nothing since player flag 1105 is already on
set_player_flag(1105, True)
# {17: 131072}
print(player_flags)
# Turns player flag 2 on
set_player_flag(2, True)
# {17: 131072, 0: 4}
print(player_flags)
# Turns player flag 1105 off
set_player_flag(1105, False)
# {17: 0, 0: 4}
print(player_flags)
```

Here is how flag changes are communicated between the client and the WorldServer:

If a flag is a session flag, it should be set for the duration of a session and when the player changes character or logs out, these flags should be cleared. A session flag can be found by querying the *PlayerFlags* table and if the `SessionOnly` boolean is set to true, the flag is set for the session.

There are two flags in live, flags `1110` and `2099` which have `SessionZoneOnly` set to true. The use of `SessionZoneOnly` is guessed to be to detect if a player has done something in a zone this session.

12.59 Custom Build Assembly Component (59)

Details unknown.

12.60 Base Combat AI Component (60)

This component manages the basic attack AI of objects.

12.60.1 Relevant Database Tables

This component uses the following tables:

- *BaseCombatAIComponent*

12.61 Module Assembly Component (61)

This component is attached to any object which can be constructed from multiple part. For the live game, this were only

- Custom Modular Rocket Ship (6416)
- Custom Racing Car (8092)

as well as some testing objects.

12.61.1 Relevant Database Tables

This component uses the following tables:

- *ModularBuildComponent*

12.62 Showcase Model Handler Component (62)

This is a component for which only test objects exist. It was some sort of mission giver and had to do with exhibits.

12.63 Racing Module Component (63)

This component is attached to objects which may be used as custom car components.

It likely holds attributes to calculate the speed, acceleration, handling, stability and imagination for the car.

12.63.1 Relevant Database Tables

This component uses the following tables:

- *RacingModuleComponent*

12.64 Generic Activator Component (64)

Had something to do with choice builds. Attached to only two models (and a test one), on each for rockets and cars.

12.65 Property Vendor Component (65)

This seems to be attached to any NPC vendor that allows you to buy a property, such as the "Property Guard", or "Guy the Land Agent".

It is possible, this component was only used in BETA.

12.66 HFLightDirectionGadget Component (66)

HappyFlower (HF) was the world editor within the game. It seems to only be attached to a single object responsible for setting the environment light direction. That object is HF Light Direction Gadget.

12.67 Rocket Launch Component (67)

This component identifies an object as a place to launch a rocket to another world. There is a different object for every Launchpad in the game.

12.67.1 Relevant Database Tables

This component uses the following tables:

- *RocketLaunchpadControlComponent*

12.68 Rocket Landing Component (68)

This objects seems to only be attached to player objects. It may be used for processing the landing via rocket.

12.69 Trigger Component (69)

This objects seems not to be attached to any object within the game database. It is dynamically added, if the object config (from the luz/lvl files) contains the key trigger_id. Events are then bound by the .lutrigger files.

12.70 Dropped Loot Component (70)

This object does not seem to be attached to any object in the database. It is possible that it only gets attached to an item object once it is dropped on the floor and handles the pick-up logic.

12.71 Racing Control Component (71)

This component seems to be attached to the zone control object of each racing based zone.

12.72 Faction Trigger Component (72)

This component seems to be attached to all the faction gear objects.

12.73 Mission Offer Component (73)

This component manages the object offering missions.

12.73.1 Relevant Database Tables

This component (likely) uses the following tables:

- *MissionNPCComponent*

12.74 Racing Stats Component (74)

This component is attached to the custom car component and likely collects stats about the car while racing.

It may include data like how far the car travelled. It's possible that the data then gets written to the player, but has to be tracked on the actual car object.

12.75 LUP Exhibit Component (75)

Something to do with showcases.

12.75.1 Relevant Database Tables

This component uses the following tables:

- *LUPExhibitComponent*
- *LUPExhibitModelData*

12.76 BBB Component (76)

Something with BrickByBrick. No objects from tha database have it attached, as it is likely only attached to user generated models.

12.77 Sound Trigger Component (77)

Triggers music/sound for objects colliding with / within the object. There are exactly 2 objects with this component:

- SOUND TRIGGER – BOX
- SOUND TRIGGER – SPHERE

12.78 Proximity Monitor Component (78)

This component seems to be attached only to launchpads. It is possible that it is used to capture players walking up to a launchpad while holding a rocket, which would then trigger the launch sequence.

12.78.1 Relevant Database Tables

This component uses the following tables:

- *ProximityMonitorComponent*

12.79 Racing Sound Trigger Component (79)

Triggers music/sound for objects colliding with / within the object. There are exactly 2 objects with this component:

- SOUND TRIGGER (for Racing) – BOX
- SOUND TRIGGER (for Racing) – SPHERE

12.80 Chat Component (80)

This component is not attached to any object in the game database.

12.81 Friends List Component (81)

This component is not attached to any object in the game database.

12.82 Guild Component (82)

This component is not attached to any object in the game database.

12.83 Local System Component (83)

This component is not attached to any object in the game database.

12.84 Mission Component (84)

This component is responsible for missions and achievements.

Note: This component is not attached to any object in the game database.

12.84.1 Relevant Game Messages

- OfferMission
- RespondToMission
- NotifyMission
- NotifyMissionTask
- CancelMission
- ResetMission
- SetMissionType-State
- NotifyRewardMailed
- RequestLinkedMission
- MissionDialogueOk

12.84.2 XML Serialization `<mis>`

Currently Active `<cur>`

Mission (Active) `<m>`

id
: ID of the mission/achievement.

o
: (?)

Progress for a task <sv>

For achievements like collecting flags, there is one of this that has the displayed progress N, and N other <sv> elements that seem to have a bitflag in the id?

v
: Value of the progress.

Completed <done>

Mission (Complete) <m>

cct
: Amount of times completed (this can be more than 1 for repeatable missions)

cts
: Timestamp of last completion in seconds.

id
: ID of the mission/achievement.

Type State <ts>

Type <type>

v
: (?)

Subtype <st>

sub
: (?)

val
: (?)

12.85 Mutable Model Behaviors Component (85)

This component is not attached to any object in the game database.

12.86 Pathfinding Control Component (86)

This component is not attached to any object in the game database.

12.87 Pet Taming Control Component (87)

This component is not attached to any object in the game database.

12.88 Property Editor Component (88)

This component is not attached to any object in the game database.

12.89 Skinned Render Component (89)

This component is not attached to any object in the game database.

12.90 Slash Command Component (90)

This component is not attached to any object in the game database.

12.91 Status Effect Component (91)

This component is not attached to any object in the game database.

12.92 Teams Component (92)

This component is not attached to any object in the game database.

12.93 Text Effect Component (93)

This component is not attached to any object in the game database.

12.94 Trade Component (94)

This component is not attached to any object in the game database.

12.95 User Control Component (95)

This component was supposed to be used for mountable objects such as the TEST Skateboard Mount which never saw the light of day.

12.96 Ignore List Component (96)

Likely something to do with social features.

12.97 LUP Launchpad Component (97)

This seems to be the component responsible for the launchpad at Stabase 3001, possibly handling the selection of the WBL world to go to.

12.98 Buff Component (98)

A **buff** is a game mechanic to increase the player status values such as health, imagination and armor temporarily through items/actions or permanently through equipment.

This component is likely responsible for storing and applying active buffs on the local player.

12.98.1 Relevant Game Messages

- AddBuff
- RemoveBuff

12.98.2 XML Serialization `<buff>`

This component is serialized to XML to store its data. The attributes are currently unknown.

12.99 Interaction Manager Component (99)

Details unknown

12.100 Donation Vendor Component (100)

This component seems to be responsible for the brick donation interaction which was used for the construction of NexusTower.

12.101 Combat Mediator Component (101)

Details unknown

12.102 Player Forced Movement Component (106)

This seems to be something related to the skateboard or mounting feature. Could be related to physics or hovering or something else. Is only attached to a normal movement speed skateboard.

12.102.1 Component Construction

Player Forced Movement

12.102.2 Component Serialization

Player Forced Movement

12.103 Brick-by-Brick Component (107)

12.104 Level Progression Component (109)

This component handles the progression in levels for the local player. See also Player_Leveling in the wiki.

12.104.1 Relevant Data Tables

- *Rewards*

12.104.2 Relevant Game Messages

- NotifyLevelRewards

12.104.3 Component Construction

Level Progression

12.104.4 Component Serialization

Level Progression

12.105 Possession Control Component (110)

Represents an entity that can possess other entities. Generally used by players.

12.105.1 Relevant Game Messages

- SetMountInventoryId
- VehicleUnlockInput
- SetStunned
- SetPlayerControlScheme

12.105.2 Component Construction

Possession Control

12.105.3 Component Serialization

Possession Control

12.106 Commendation Vendor Component (102)

This seems to be the component for a vendor that can sell you lost or deleted faction items for faction tokens. The only object this is attached to is Honor Accolade – Commendation Vendor.

More information on may be found here: Honor_Accolade

12.107 Rail Activator Component (104)

This seems to be part of the ninjago rail system.

12.108 Roller Component (105)

This seems to be some sort of sidescrolling platform, as may be used in a shooting gallery or as "roller cylinder" in a challenge.

12.109 Possessable Component (108)

This component seems to be attached to all the cars, skateboards, hovercraft, moon buggies and other vehicles that were meant to be driven around the place.

12.110 Property Plaque Component (113)

This component is attached only to the property plaque component, so it was likely used for something property related.

12.111 Build Border Component (114)

This component seems to have been used to limit the accessible area when building rockets or cars.

12.112 Culling Plane Component (116)

Components with this object ar supposed to occlude other objects behind them. The only object in the database with this component is Culling Plane

12.113 Mount Control Component (?)

This component would handle the local player mounting some other object. It appears in the chardata packet captures, but is not otherwise known. Consequently the ID is unknown too.

12.113.1 XML Serialization `<mnt>`

a
(?)

Note: If an object has the LDF key `is_smashable` set to 1, or it is configured at the `lvl` level to have `is_smashable` set, then this object should get a `Destroyable Component` if it does not already have one. If `is_smashable` is set at the `lvl` level, then you need to insert and set `is_smashable` to true.

If any of these documents helped you in some way or another for one of your projects, please credit us and/or include a direct link to this document!

Warning: Neither the original documentation nor this read-the-docs port is associated or involved with The LEGO Group. Likewise, the creators of the aforementioned documents are not associated or involved with The LEGO Group.

CHAPTER

THIRTEEN

CONTACT INFO

Most developer activity happens on the LEGO Universe Community Hub (LUCH) Discord server nowadays. If you join there and ask a question, it's quite likely that someone will be able to help you out.

Developers from all major server projects are active in LUCH.