

Network packets

Network packets are a way of contacting to the host or client via sending or receiving data. The amount of defined network packets has been changing ever since the implementation of Http Dll 2.1¹⁾ in [Brick Hill Legacy Beta](#). This article is about the latest [Brick Hill Legacy](#) network packets.

Before you'll start implementing Brick Hill network packets: Client and Host should send the Network Packet Type ID before sending other data, listed on next tables.

What client receives?

Network Packet type	What does the network packet contain?
1 (Authentication packet, defined by Node Hill as Authentication)	Network ID ²⁾ (32-bit integer) Brick Count (32-bit integer) User ID ³⁾ (32-bit integer) Username (string) Are you an admin ? ⁴⁾ (8-bit integer) Membership type (8-bit integer) Game ID (32-bit integer) Game Name (string)
3 (New player packet, client creates the obj_figure Object, defined by Node Hill as SendPlayers)	Amount of players (8-bit integer) Network ID (32-bit integer) Username (string) User ID ⁵⁾ (32-bit integer) Is new player an admin ? ⁶⁾ (8-bit integer) Membership type (8-bit integer)

Network Packet type	What does the network packet contain?
<p>4 (Client Data Packet, defined in GameMaker Client via packet_handler script; defined by Node Hill as Figure)</p>	<p>ID type (string) Network ID (32-bit integer) ID types, defined in GameMaker client: A - Position by X (float int) B - Position by Y (float int) C - Position by Z (float int) D - ? (float int, unused) E - Camera Rotation (float int, unused) F - Player rotation by Z (float int) G - Player scale by X (float int) H - Player scale by Y (float int) I - Player scale by Z (float int) K - Head color (32-bit integer) L - Torso color (32-bit integer) M - Left Arm color (32-bit integer) N - Right arm color (32-bit integer) O - Left leg color (32-bit integer) P - Right Leg color (32-bit integer) Q - Face Item ID (string) R - Shirt Item ID⁷⁾ (string) S - Pants Item ID (string) T - T-Shirt Item ID (string) U - First Equipped Hat Item ID, receives Mesh UUID (string) and Texture UUID (string) V - Second Equipped Hat Item ID, receives Mesh UUID (string) and Texture UUID (string) W - Third Equipped Hat Item ID, receives Mesh UUID (string) and Texture UUID (string) X - Score (32-bit integer, can be negative) Y - Team (32-bit integer)</p> <p>The next ID Types are defined by GameMaker client as Local Changes: 1 - Walking Speed (32-bit integer) 2 - Jump Height (32-bit integer) 3 - Field of View (32-bit integer) 4 - Camera distance (32-bit integer, can be negative) 5 - Camera position by X (float int) 6 - Camera position by Y (float int) 7 - Camera position by Z (float int) 8 - Camera rotation by X (float int) 9 - Camera rotation by Y (float int) a - Camera rotation by Z (float int) b - Camera Type (string) c - Camera Object (32-bit integer) e - Health (float int) f - Speech⁸⁾ (string, limited to 500 characters⁹⁾) g - Equipped tool: Slot ID (32-bit integer), Mesh UUID (string), Texture UUID (string) h - Sets Arm value to -1 i - Is player alive?¹⁰⁾ (8-bit integer)</p> <p>The next ID Types are defined by GameMaker client as “new clothing system”: j - Clothe1 (string) k - Clothe2 (string) l - Clothe3 (string) m - Clothe4 (string) n - Clothe5 (string)</p>

Network Packet type	What does the network packet contain?
5 (Remove Player Packet, defined by Node Hill as RemovePlayer)	User's Network ID (32-bit integer) and client removes its obj_figure object from the game.
6 (Chat Message Packet, defined by Node Hill as Chat)	A formed by Host chat Message (string, 500 limit)
7 (Environment Data, GUI; defined by Node Hill as PlayerModification)	Data Type (string): topPrint - Top Text: Message (string), Time (32-bit integer) centerPrint - Middle Text: Message (string), Time (32-bit integer) bottomPrint - Bottom Text: Message (string), Time (32-bit integer) Ambient - Ambient color (32-bit integer) Sky - Sky color (32-bit integer) BaseCol - Baseplate color (32-bit integer) BaseSize - Baseplate size (32-bit integer) Sun - Sun intensity (32-bit integer) kick - Kicks the client ¹¹⁾ prompt - Prompt message (string) WeatherSnow - Sets the weather to snow WeatherRain - Sets the weather to rain WeatherSun - Sets the weather to sun TweakDisableFigureCulling - Disables the Figure culling option TweakDisableLighting - Disables lighting TweakRaySpacing - ? (float int) MoreClickInfo - enables more information on click (8-bit integer)
8 (Kill Packet, defined by Node Hill as Kill)	User's Network ID (32-bit integer), then client sets the alive variable to 0 to its corresponding obj_figure object.
9 (Brick Modification Packet, defined by Node Hill as Brick)	The obj_brick's Network ID (32-bit integer) Type of Modification (string). Brick Modifications: pos ¹²⁾ - Position by X (float int) Position by Y (float int) Position by Z (float int) rot ¹³⁾ - Rotation by Z (float int) scale ¹⁴⁾ - Scale by X (float int) Scale by Y (float int) Scale by Z (float int) kill ¹⁵⁾ - Time until destruction (32-bit integer) destroy - Removes the brick col ¹⁶⁾ - Color (32-bit integer) alpha ¹⁷⁾ - Alpha (float int) lightcol ¹⁸⁾ - Color (32-bit integer) lightrange ¹⁹⁾ - Range (32-bit integer) model ²⁰⁾ - Mesh UUID (string) Texture UUID (string) collide ²¹⁾ - Collision state (8-bit integer) clickable ²²⁾ - Clickable state (8-bit integer) Clickable distance (32-bit integer) global - ? ignoreLight ²³⁾ - makes ignoreLight variable true
10 (New team packet, defined by Node Hill as Team)	Team ID (32-bit integer) Name (string) Color (32-bit integer)

Network Packet type	What does the network packet contain?
11 (New tool packet, defined by Node Hill as Tool)	Active state (8-bit integer) Slot ID (32-bit integer) Name (string)
12 (Bot Data packet, defined in GameMaker Client via packet_handler_figure script; defined by Node Hill as Bot)	String ID (string) Figure ID (32-bit integer) Regarding the packet_handler_figure Script: It generally contains the same information as packet_handler Script, except it does not need Local Changes.
13 (New Projectile Packet, defined by Node Hill as Projectile)	Active state (8-bit integer) Projectile ID (32-bit integer) Diameter (32-bit integer) Color (32-bit integer) Position by X (float int) Position by Y (float int) Position by Z (float int) Direction (32-bit integer) Direction by Z (32-bit integer) Velocity (32-bit integer)
14 (Clear Map Packet, defined by Node Hill as ClearMap)	Client removes all the bricks after it has received this packet
15 (Remove Bot packet, defined by Node Hill as DestroyBot)	Figure ID (32-bit integer), then client removes the bot by its Network ID
16 (Remove Brick packet, defined by Node Hill as DeleteBrick)	Client receives the Brick ID (32-bit integer) and removes the brick by its ID.
17 (SendBrick Packet, Node Hill definition)	Network ID (32-bit integer) Position by X (float int) Position by Y (float int) Position by Z (float int) Scale by X (float int) Scale by Y (float int) Scale by Z (float int) Color (32-bit integer) Transparency (float int)

What client sends?

Network Packet type	What does the network packet contain?
1 (Authentication packet, defined by Node Hill as Authentication)	Token ²⁴⁾ (string) Client version (string)
2 (Player position Packet, defined by Node Hill as Position) ²⁵⁾	Position by X (float int) Position by Y (float int) Position by Z (float int) Rotation by Z (float int) Camera rotation by Z (float int)
3 (Command Packet)	Command (via /, string)
4 (Projectile Packet) ²⁶⁾	The projectile ID , the client has made a collision with (32-bit integer) It's own Network ID (32-bit integer)
5 (ClickDetection Packet)	Brick Network ID (32-bit integer)
6 (PlayerInput Packet)	Mouse click (8-bit integer) Current Keyboard Key (string)
18 (Heartbeat packet, sent in Brick Hill Legacy Client via update_delta script) ²⁷⁾	Message (sent via socket_write_message ²⁸⁾)

Notes

1)

GameMaker 8.1 extension with better networking features.

2)

Object id of a `obj_client`, assigned server-side.

3)

Retrieved via Brick Hill site API, hence the auth key **must be verified**.

4) , 6) , 10)

1 - true, 0 - false.

5)

Retrieved via Brick Hill site API server-side.

7)

unused, old. Same with Pants, T-Shirt

8)

bubble chat message

9)

If this packet function didn't check the length of the chat message then hosts using `Game.on('chat')` could cause clients to crash by relaying very large chat messages.

11)

This will become a vulnerability if the player socket will not be destroyed by Host itself.

12)

Changes brick position.

13)

Changes brick rotation by Z.

14)

Changes the scale of brick.

15)

Unanchors the brick.

16)

Changes the brick color.

17)

Changes the brick's transparency.

18)

Changes the light color.

19)

Changes the light range.

20)

Sets the custom model to brick.

21)

Changes the brick collision.

22)

Changes the click state and distance for a brick.

23)

Logically, this should ignore the light, Client source code got its option commented out (was supposed to support 8-bit integer).

24)

User's authentication key.

25)

Earlier: unused method of loading bricks. Exists yet commented out in the game code of latest Brick Hill Legacy Client.

26)

Node Hill does not have an implementation for this packet (it's empty).

27)

The client should send this packet constantly (**every 10 seconds**).

28)

Http Dll 2.1 citation: Sends a message. A message is sent by first sending the size of the message (as uintv), followed by the actual message data. This is handled automatically by the DLL, you don't need to do anything special if the receiver also uses Http Dll 2..

From:

<http://brickrot.app.tc/> - **brickrot**

Permanent link:

<http://brickrot.app.tc/doku.php?id=hosting:netpackets>

Last update: **2025/12/05 15:37**

