

# MagicDNS

## Documentation

MagicDNS is a simple word to address resolution service which also provides NAT traversal using middleman hole punching.

Basic event structure:

- 1) Server player makes a request to get a MDNS name.
- 2) MDNS name is retrieved and server player sends hole punching packets.
- 3) MDNS returns successful hole punching status and the name is shown to the server player.
- 4) Players ask MDNS for server player's address and port based on words.
- 5) Players receive server player's address and connect to him.
- 6) Game starts, request to remove address from MDNS is called.

Features:

- Players can share easy to remember words which are simple enough to not be misspelled by most.
- Hole punching removes the need for inexperienced users to forward ports.
- Seamless integration by using the same field as hostname/ip/domain connections – differentiated by space.
- User generated session id allows multiple server players under same public IP.
- Mandatory keep-alive and MEMORY database table mode make sure that expired addresses are not visible to the users and are removed immediately, ensuring privacy.
- Usage of game identity ensures that multiple games can use MagicDNS, and it also allows word reuse. Additionally, matchmaking can be easily implemented by adding a suffix, i.e. \_MM, thus isolating public from private servers.

Drawbacks:

- User generated session ids can have collisions (as unlikely as they are, having  $35^8$  combinations and only being able to happen when multiple servers are under same NAT) and make a mess.
- No IPv6 support.
- Currently no DOS countermeasures.

Players losing connection to the server should not rely on MagicDNS to provide data. This may cause problems if the client crashes and attempts to rejoin the game using words.

All commands and replies are uppercase. Requests containing lowercase or invalid characters may be ignored.

## HTTP portion

GET request to **dns.stabyourself.net/command/data1/data2**

LuaSocket user agent required (set by default).

Valid http response is 200, content type text, contents RESPONSE/DATA1/DATA2 etc.

**Uppercase is mandatory.**

Server player > DNS	Note
MAKE/MARIO/ABCD1234/7331	Request words. Previous data with same IP and session is deleted. On success: MADE - Game identity (love_conf t.identity) - Session. Used to differentiate between users on same external ip. This should be generated on game start and stay consistent as long as game is active. - Local server port
KEEP/MARIO/ABCD1234	Keep-alive to prevent word expiry. This should be called at least once every 30 seconds when idle. <u>Response to this packet also notifies the server player if hole punching is complete.</u> On success: KEPT - Game identity - Session
REMOVE/MARIO/ABCD1234	Request record deletion. This should be called once lobby closes and game starts. On success: REMOVED - Session
DNS > Server player	
MADE/ADJ/NOUN	Words created. To be displayed in the game lobby. - Adjective - Noun
KEPT/ or KEPT/56203	Keep-alive successful. If data is returned, that data is the latest acquired UDP server port. -Server player's external port acquired by the UDP monitor (if detected)
REMOVED	All user's data has been removed.
Player > DNS	
FIND/MARIO/ADJ/NOUN	Request address based on words. - Game identity - Adjective - Noun
DNS > Player	
FOUND/199.199.199.199/7331/ or FOUND/199.199.199.199/7331/17331	Game has been found. Player should re-fetch the data a few seconds later if external port is not known as connection might not be possible using only server player's local port. - Server player address - Server player local port - Server player external port (if already known)
NOTFOUND	Game has not been found at given words (either removed or expired).

## Error reporting

Server timeout should be reported to user as **Server down**.

Non standard HTTP returns should be reported to user as **Server error**.

Normal errors are reported as ERROR/CODE (in same manner as normal HTTP communication). The code will contain explanation of the first occurred error and may contain punctuation. Contrary to other parts, lowercase characters are used. These should be reported to user as **MagicDNS error: <code>**.

## UDP portion

Sent as plaintext in packet to dns.stabyourself.net:5450

Server player > DNS	Note
PUNCH/MARIO/ABCD 1234	Hole punching attempt. The server will not reply to this; instead, HTTP KEEP should be used to verify that hole punching has been successful. While NAT UDP should not expire in less than 2 minutes (RFC4787 4.3), this should be called at least once every 30 seconds while lobby is active. <ul style="list-style-type: none"><li>- Game identity</li><li>- Session</li></ul>

## Input validity

If inputs are outside the given bounds, an error will occur.

Data	Conditions
Adjective Noun	A-Z Combined max length 31
Game identity	A-Z 0-9 - _ Max length 16
Session	A-Z 0-9 Length 8
Port Local port	> 0 < 65536

## Database table structure

MYSQL, table dns\_entries, MEMORY (HEAP)

Name	Datatype	Notes
ip_address	int_unsigned	<u>Primary key</u>
identity	varchar(16)	<u>Primary key</u> , <u>Unique(+magic_word)</u> , game identity
session	char(8)	<u>Primary key</u> , generated by player
port_local	smallint_unsigned	
port	smallint_unsigned	<u>Nullable</u>
magic_word	varchar(32)	<u>Unique(+identity)</u> , actual words seperated by /
timestamp	timestamp	ON UPDATE CURRENT_TIMESTAMP

## SQL DDL

```
CREATE TABLE `dns_entries` (  
  `ip_address` int(10) unsigned NOT NULL,  
  `identity` varchar(16) NOT NULL,  
  `session` char(8) NOT NULL,  
  `port_local` smallint(5) unsigned NOT NULL,  
  `port` smallint(5) unsigned DEFAULT NULL,  
  `magic_word` varchar(32) NOT NULL,  
  `timestamp` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE  
CURRENT_TIMESTAMP,  
  PRIMARY KEY (`ip_address`,`identity`,`session`),  
  UNIQUE KEY `MULTIGAME_SUPPORT` (`identity`,`magic_word`)  
) ENGINE=MEMORY DEFAULT CHARSET=latin1;
```