

Research On GameSpy Protocol

Arves100, Xiaojiuwo



First Edition

June 26, 2019

Contents

I	Research On GameSpySDK	2
1	GameSpy General Construction	3
1.1	GameSpy SDK Module	3
2	GameSpy Presence Servers	4
2.0.1	Server IP and Ports	4
2.0.2	Database Key Field	4
2.0.3	Protocol Descriptions	4
2.0.3.1	The Pattern of String	5
2.0.3.2	Login Phase	5
3	summary	6
4	introduction3	7
II	RetroSpy System Architecture	8
5	introduction	9
6	conclusion	10

Part I

Research On GameSpySDK

Chapter 1

GameSpy General Construction

In GameSpy SDK there are 16 modules, which constructed the GameSpy main functions.

1.1 GameSpy SDK Module

- GameSpy Presence Servers
 - GameSpy Presence Connection Manager
 - GameSpy Presence Search Player
- Nat Negotiation
- Master Server: Query Report 2
- Master Server: Server Browser
- Master Server: Available Check
- Game Patching
- Game Tracking
- Master Server Patching: Downloading files from FilePlanet
- Peer SDK
- Game Statistics
- Chat Server

Chapter 2

GameSpy Presence Servers

GameSpy Presence Servers contain two server, GameSpy Presence Connection Manager (GPCM) and GameSpy Presence Search Player (GPSP). GPCM is a server that handle login request and response with corresponding user information stored on GameSpy. GPSP is a server that handle search request for user.

2.0.1 Server IP and Ports

Table 2.1 are the GPCM and GPSP IP and Ports that client/game connect to.

IP	Port
gpcm.gamespy.com	29900
gpsp.gamespy.com	29901

Table 2.1: IP and Ports for GameSpy Presence Servers

2.0.2 Database Key Field

These keys is that GameSpy Presence SDK using to find a user in their database. Keys are shown in Table 2.2.

Keys	Description
User	An user contains the Email and the password, but contains multiple profiles
ProfileID	The profile contains the name, surname, birth date and all the rest user info, including an unique nickname used to identify the profile and a generic nickname used to show for example in games

Table 2.2: Key Field

2.0.3 Protocol Descriptions

In this part, we show the protocol detail in GameSpy Presence SDK.

2.0.3.1 The Pattern of String

We first introduce the pattern of the string, which is using to make up a request. This kind of string is represent a value in a request sends by the client as Table 2.3.

String	Description
<code>\<content>\</code>	The value is <i><content></i>

Table 2.3: Value string

This kind of string is represent a command in a request sends by the client as Table 2.4.

String	Description
<code>\command\</code>	This is a command
<code>\error\</code>	Error command
<code>\lc\</code>	Login command

Table 2.4: Command string

This kind of string is represent a parameter in a request sends by the client 2.5. GameSpy uses the combination of the parameter to search the string with value, and sends the data back to client use this kind of parameter string.

String	Description
<code>\id\1\</code>	This is a parameter string the value of <i>id</i> is 1
<code>\profileid\007\</code>	This is a parameter string the value of <i>profileid</i> is 007

Table 2.5: Parameter string

Error response string for (GPCM, GPSP):

$$\backslash error \backslash err \backslash \langle errorcode \rangle \backslash fatal \backslash errmsg \backslash \langle errormessage \rangle \backslash id \backslash 1 \backslash final \backslash \quad (2.1)$$

2.0.3.2 Login Phase

There are three ways of login:

- AuthToken: Logging using an alphanumeric string that rapresents an user
- UniqueNick: Logging using a nickname that is unique from all the players
- User: Logging with the nickname and the password

Login response string:

$$\backslash lc \backslash 1 \quad (2.2)$$

This response string 2.2 is send by the server when a connection is accepted, and followed by a challenge2.3, which verifies the server that client connect to.

The challenge string:

$$\backslash challenge \backslash \langle challenge \rangle \quad (2.3)$$

The *<challenge>* in 2.3 is a 10 byte alphanumeric string.

Chapter 3

summary

Chapter 4

introduction3

Part II

RetroSpy System Architecture

Chapter 5

introduction

Chapter 6

conclusion