Gigassoft	**Aeriagames Billing system**		**Payletter**
Doc Name	Billing interface(For 12Sky)	Doc Class	*Confidential*
Writer	Development team of payletter	Last Revised	2007-10-17

Billing Interface

(For 12Sky)

Gigassoft	Aeriagames Billing system		Payletter
Doc Name	Billing interface(For 12Sky)	Doc Class	*Confidential*
Writer	Development team of payletter	Last Revised	2007-10-17

PAYLETTER, Inc.

DanWoo B/D, 852-22, Bangbae-Dong, Seocho-Gu, Seoul, Korea

Fax: +82-02-590-3710

Email: tech-help@payletter.com

Table of contents

1. Abstract 4
2. Communication Flow 4
3. Considerations 4
   1) About TCP/IP Session 4
   2) Byte ordering for numeric variables. 4
   3) Recovering the TCP/IP Session 5
   4) Packing alignment of structure 5
4. Interface details 5
5. Interface Packet details 5
   1) Connection Request 6
   2) Balance check request 7
   3) Charge(Deduct AP) request 8
   Appendix : System error list 9

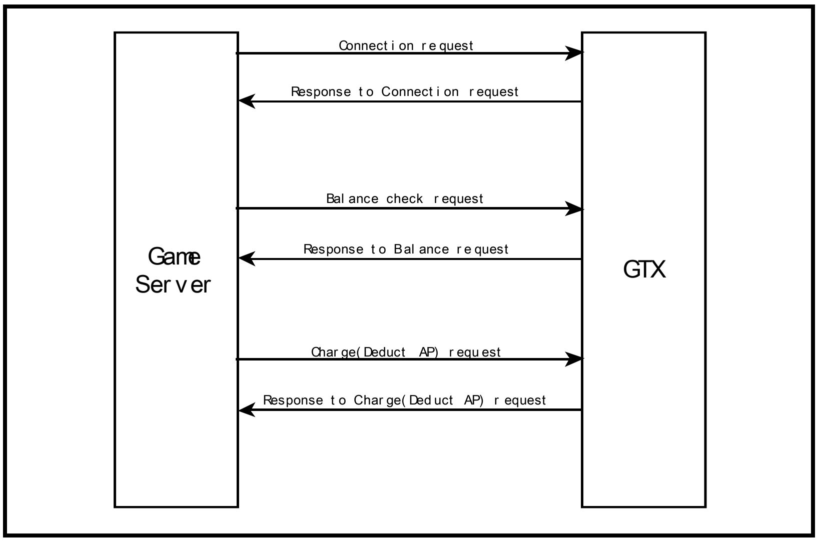
1. Abstract

This document describes the protocol which is needed when the game server and the billing system(GTX) communicate each other.

2. Communication Flow

Describe core communication flow between game server and GTX.

[Figure]Packet flow between game server and GTX



Connection request
Response to Connection request
Balance check request
Response to Balance request
Game Server
GTX
Charge(Deduct AP) request
Response to Charge(Deduct AP) request

3. Considerations

1) About TCP/IP Session

   The game server must try to connect to the billing server(GTX) when it starts. When the first trial fails, game server has to retry 3times with 20 seconds interval. If all the trials fail, the game server has to connect to the backup billing server(GTX).

2) Byte ordering for numeric variables.

   When the game server sends packet to the billing server(GTX), the game server has to change the byte ordering to the “network byte”, and also, the game server has to change the byte ordering to the “host byte” after receiving the packet from the billing server(GTX).

   *If the server OS types are same, this can be ignored.(The billing system uses MS windows)

3) Recovering the TCP/IP Session

If the connection between the game server and the billing server was broken by various reasons, the game server has to reconnect to the billing system.

4) Packing alignment of structure

The game server must pack the structure to 1 byte boundary

Ex)

```
#pragma pack(push, 1)
Declear the structures……
#pragma pack(pop)
```

4. Interface details

The game server has to use following protocol when it communicates with the billing server(GTX).

[Table]Kinds of interface

Kind	Description	Type(response)	Sender
Connection request	The game server has to request connection after physically connected to the billing system.	10(11)	Game server
Balance check request	Check the user's current remain AP.	20(21)	Game server
Charge(Deduct AP) request	Deduct AP for purchasing items of 12Sky.	30(31)	Game server

※ It can be changed, if it is needed.

5. Interface Packet details

Inteface Packet consists of two parts "header" and "body". The "header" is same to all the packets, but "body" is different with each packet.

header(8Bytes)	body(it varies for each packet)

**"header" consists of type(2bytes), size(2bytes) and packet sequence(4bytes).

field	description
WORD wPacketType	Type

WORD wPacketSize	Size(it includes the size of the “body”)
DWORD dwSeqNo	Packet sequence

1) Connection Request

a. Request packet (game server -> GTX)

field	description
WORD wPacketType	Type(10)
WORD wPacketSize	Size
DWORD dwSeqNo	Packet sequence
WORD wLoginServerNo	Identifier of the game server

```
typedef struct _ST_GAMESERVER_CONN_REQ
{
            WORD wPacketType;
            WORD wPacketSize;
            DWORD dwSeqNo;
            WORD wLoginServerNo;
} ST_GAMESERVER_CONN_REQ, *LPST_GAMESERVER_CONN_REQ;
```

b. response packet (GTX -> game server)

field	description
WORD wPacketType	Type(11)
WORD wPacketSize	Size
DWORD dwSeqNo	Packet sequence
BYTE bResult	Response code(0:allowed, 1:denied, >50:system error) *Refer to the error code list for system errors.

```
typedef struct _ST_GAMESERVER_CONN_REP
{
            WORD wPacketType;
            WORD wPacketSize;
            DWORD dwSeqNo;
            BYTE bResult;
} ST_GAMESERVER_CONN_REP, *LPST_GAMESERVER_CONN_REP;
```

2) Balance check request

a. Request packet (game server -> GTX)

field	description
WORD wPacketType	Type(20)
WORD wPacketSize	Size
DWORD dwSeqNo	Packet sequence
DWORD dwIPAddr	IP address of the user.
char szUserID[50+1]	User's login ID *it should include the NULL terminator.

```
typedef struct _ST_USER_BALANCE_REQ
{
        WORD wPacketType;
        WORD wPacketSize;
        DWORD dwSeqNo;
        DWORD dwIPAddr;
        char szUserID[50+1];
} ST_USER_BALANCE_REQ, *LPST_USER_BALANCE_REQ;
```

b. Response packet (GTX -> game server)

field	description
WORD wPacketType	Type(21)
WORD wPacketSize	Size
DWORD dwSeqNo	Packet Sequence
BYTE bResult	Response code(0:succeeded, 1:failed, >50:system error) *Refer to the error code list for system errors.
DWORD dwRemainAP	Remain AP

```
typedef struct _ST_USER_BALANCE_REP
{
        WORD wPacketType;
        WORD wPacketSize;
        DWORD dwSeqNo;
```

```
            BYTE bResult;
            DWORD dwRemainAP;
} ST_USER_BALANCE_REP, *LPST_USER_BALANCE_REP;
```

3) Charge(Deduct AP) request

a. Request packet (game server -> GTX)

field	description
WORD wPacketType	Type(30)
WORD wPacketSize	Size
DWORD dwSeqNo	Packet sequence
DWORD dwIPAddr	IP address of the user.
char szUserID[50+1]	User's login ID *it should include the NULL terminator.
char szItemKey[50+1]	Unique key value of the item in the game DB
char szItemName[50+1]	Name of the item
DWORD dwPrice	Price of the item(it should be bigger than 0)

```
typedef struct _ST_USER_CHARGE_REQ
{
            WORD wPacketType;
            WORD wPacketSize;
            DWORD dwSeqNo;
            DWORD dwIPAddr;
            char szUserID[50+1];
            char szItemKey[50+1];
            char szItemName[50+1];
            DWORD dwPrice;
} ST_USER_CHARGE_REQ, *LPST_USER_CHARGE_REQ;
```

b. Response packet (GTX -> game server)

field	description
WORD wPacketType	Type(31)
WORD wPacketSize	Size
DWORD dwSeqNo	Packet sequence

BYTE bResult	Response code(0:succeeded, 1:failed, 2:bad user, 3:insufficient AP, 4:no such a userid, >50:system error) *Refer to the error code list for system errors.
DWORD dwRemainAP	Remain AP
Char szPurchaseNo[15+1]	Unique transaction key of the billing system

```
typedef struct _ST_USER_CHARGE_REP
{
        WORD wPacketType;
        WORD wPacketSize;
        DWORD dwSeqNo;
        BYTE bResult;
        DWORD dwRemainAP;
        char szPurchaseNo[15+1];
} ST_USER_CHARGE_REP, *LPST_USER_CHARGE_REP;
```

Appendix : System error list

code	description
51	Invalid parameters
61	DB processing failure
69	Exception in DB processing
71	Busy(There are so may queued requests)
79	Exception in GTX processing