服务器消息定义：map server

/\*

ADMIN\_OP\_GM\_ALL = 0,//默认

ADMIN\_OP\_GM\_PUT = 2,//运营消息

ADMIN\_OP\_GM\_USER = 10,//炫耀消息断点

ADMIN\_OP\_GM\_LUCK\_EQUIPMENT = 11,//抽奖获得星或以上装备

ADMIN\_OP\_GM\_LUCK\_PLAYER = 12,//抽奖获得角色

ADMIN\_OP\_GM\_STEP\_EQUIPMENT = 13,//进阶装备

ADMIN\_OP\_GM\_STEP\_PLAYER = 14,//进阶角色

ADMIN\_OP\_GM\_COMPOSE =15,//合成装备

ADMIN\_OP\_GM\_PROPS = 16 //道具

\*/

#define MSG\_GM\_REQUEST MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x0260)//GM请求

typedef struct tagMsgAnnouncementRequest : public MsgBaseRequest

{

ACE\_UINT32 unTypeID;//消息类型

}MsgAnnouncementRequest;

//

#define MSG\_DB\_GM\_REQUEST MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x0261)//GM DB请求

typedef struct tagMsgGMRequest : public MsgBaseRequest

{

ACE\_UINT32 unTypeID;//消息类型

ACE\_UINT32 unToID;//请求人

}MsgGMRequest;

#define MSG\_DB\_GM\_RESPONSE MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x0262)////公告牌请求返回

#define MSG\_GM\_RESPONSE MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x0263)////公告牌请求返回

#define MSG\_GM\_MAX\_LEN 20

typedef struct tagMsgGMRsponse : public MsgBaseResponse

{

ACE\_UINT32 unCount;

GMInfo info[MSG\_GM\_MAX\_LEN];//gm推送数据

}MsgGMRsponse;

#define MSG\_GET\_LA\_LIST\_REQUEST MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x0264)//获取限时活动请求

//MsgBaseRequest

#define MSG\_GET\_LA\_LIST\_RESPONST MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x0265)//返回限时活动请求

//#define MAX\_LIMITED\_ACTIVITY\_COUNT 32//限时活动最大数

typedef struct tagMsgGMLimitedActivityList :MsgBaseResponse

{

ACE\_UINT32 m\_unCount;

LimitedActivityInfo m\_LAList[MAX\_LIMITED\_ACTIVITY\_COUNT];//限时活动列表

}MsgGMLimitedActivityListResponse;

#define MSG\_GET\_LA\_REWARD\_REQUEST MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x0266)//限时活动领取请求

typedef struct tagMsgGMLARewardRequest : public MsgBaseRequest

{

ACE\_UINT32 unLAMID;//限时活动任务列表id

}MsgGMLARewardRequest;

#define MSG\_GET\_LA\_REWARD\_RESPONST MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x0267)//限时活动领取返回结果

typedef struct tagMsgGMLARewardResponse :MsgBaseResponse

{

RewardInfo stReward;//奖励内容

}MsgGMLARewardResponse;

#define MSG\_DO\_BUYENERGY\_REQUEST MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x0268)//买体力

typedef struct tagMsgDoBuyEnergyRequest : public MsgBaseRequest

{

}MsgDoBuyEnergyRequest;

#define MSG\_DO\_BUYENERGY\_RESPONSE MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x0269)//买体力返回

typedef struct tagMsgDoBuyEnergyResponse : public MsgBaseResponse

{

//int nGold;

ACE\_UINT32 unCurrentCount;//已购买次数

ACE\_UINT32 unEnergy;//买了多少体力

ACE\_UINT32 unMoneyCost;//消耗多少钱

//RewardInfo reward;

}MsgDoBuyEnergyResponse;

#define MSG\_ADMIN\_OP\_RECHARGE\_REQUEST MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x026a)//充值

#define MSG\_DO\_ORDER\_REQUEST MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x0270)//充值订单请求

typedef struct tagMsgDoOrderRequest : public MsgBaseRequest

{

ACE\_UINT32 product\_id;//订单类型

ACE\_UINT32 product\_count;//订单类型

}MsgDoOrderRequest;

#define MSG\_DO\_ORDER\_RESPONSE MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x0271)//充值订单请求返回结果

typedef struct tagMsgDoOrderResponse : public MsgBaseResponse

{

//ACE\_UINT32 m\_unID;

char orderID[RECHARGE\_ORDERID\_LEN+1];//充值头

}MsgDoOrderResponse;

#define MSG\_DO\_RECHARGE\_REQUEST MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x0272)//充值成功检测

typedef struct tagMsgDoRechargeRequest : public MsgBaseResponse

{

char orderID[RECHARGE\_ORDERID\_LEN+1];//充值头

}MsgDoRechargeRequest;

#define MSG\_DO\_RECHARGE\_RESPONSE MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x0273)//充值成功返回结果

typedef struct tagMsgDoRechargeResponse : public MsgBaseResponse

{

char orderID[RECHARGE\_ORDERID\_LEN+1];//充值头

ACE\_UINT32 deal\_time;//订单完成时间

ACE\_UINT32 unRenminbi;//充值金额(人民币)

ACE\_UINT32 unMoney;//充值金卡数量

ACE\_INT32 unProductId;//商品id

ACE\_INT32 unProductCount;//商品数量

//ACE\_UINT32 unChannel;//充值渠道id

//OrderInfo info;

ACE\_UINT32 unbitField;

ACE\_UINT32 unMoneyEx;//赠送的钱

}MsgDoRechargeResponse;

#define MSG\_USER\_INFO\_ADD\_MAIL MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x0274)//添加邮件

#define MSG\_DO\_PIECE\_ROLE\_REQUEST MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x0280)//碎片转角色请求

typedef struct tagMsgDoPieceToRoleRequest : public MsgBaseRequest

{

ACE\_UINT32 unItemID;//29开头

}MsgDoPieceToRoleRequest;

#define MSG\_DO\_PIECE\_ROLE\_RESPONSE MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x0281)//碎片转角色请求返回

typedef struct tagMsgDoPieceToRoleResponse : public MsgBaseResponse

{

ACE\_UINT32 unPlayerID;//角色id

ACE\_UINT32 unPieceRoleID;//碎片id

//CostInfoList cost;

}MsgDoPieceToRoleResponse;

#define MSG\_DO\_UPDATE\_LIVEL\_REQUEST MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x0282)//更新等级

#define MSG\_Update\_Bitfield MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x0283)//更新标志

typedef struct tagMsgDoBitFieldRequest : public MsgBaseRequest

{

ACE\_UINT32 unCharactID;

ACE\_UINT32 bitfield;//标记

}MsgDoBitFieldRequest;

#define MSG\_DO\_GET\_RECHARGE\_INFO\_REQUEST MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x0284)//获取充值或vip信息

#define MSG\_DO\_GET\_RECHARGE\_INFO\_RESPONSE MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x0285)//获取充值或vip信息

typedef struct tagMsgDoGetRechargeInfoResponse : public MsgBaseResponse

{

ACE\_UINT32 m\_bitField;

ProductList m\_Productlist;//已充值列表

VipInfo info;//vip信息

}MsgDoGetRechargeInfoResponse;

typedef struct tagBitfield

{

int firstRecharge:1;//首充

int monthCard:1;//月卡

int sharedActivites:1;//分享得奖励

tagBitfield() { memset(this, 0, sizeof(\*this)); }

tagBitfield(ACE\_UINT32 value){SetValue(value);}

void SetValue(ACE\_UINT32 value)

{

memcpy(this,&value,sizeof(Bitfield));

}

ACE\_UINT32 GetValue() const

{

ACE\_UINT32 n = 0;

memcpy(&n,this,sizeof(Bitfield));

return n;

}

}Bitfield;

#define MSG\_DO\_SHARED\_ACTIVITES\_REQUEST MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x0286)//分享活动

#define MSG\_DO\_SHARED\_ACTIVITES\_RESPONSE MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x0287)//分享活动

typedef struct tagMsgSharedActivitesResponse : public MsgBaseResponse

{

RewardInfo reward;//分享获得的奖励

}MsgSharedActivitesResponse;

#define MSG\_DO\_VIP\_GIFT\_REQUEST MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x0288)//VIP等级礼包

typedef struct tagMsgVipGiftRequest : public MsgBaseRequest

{

ACE\_UINT32 ungiftId;//礼包id

ACE\_UINT32 unvipLevel;//等级

}MsgVipGiftRequest;

#define MSG\_DO\_VIP\_GIFT\_RESPONSE MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x0289)//VIP等级礼包

typedef struct tagMsgVipGiftResponse : public MsgBaseResponse

{

ACE\_UINT32 unGiftID;//礼包id

RewardInfo reward;//获得的奖励

}MsgVipGiftResponse;

#define MSG\_DO\_DB\_SERVER\_TIME\_HARD MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x028A)//服务器压力太大，需要休息秒钟时间

#define ITME\_HARD\_SLEEP 1000 //睡眠时间

typedef struct tagMsgDoDBTimeHardRequest : public MsgBaseRequest

{

}MsgDoDBTimeHardRequest;

#define MSG\_DO\_Get\_BadgeGaint\_REQUEST MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x028B)//获得加成目标，更新曾益

#define MSG\_DO\_Get\_BadgeGaint\_RESPONSE MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x028C)//获得加成目标，更新曾益

typedef struct tagMsgGetBadgeGaintResponse : public MsgBaseResponse

{

struct tagGetBadgeGaintInfo

{

ACE\_UINT32 unGaint[4];

};

ACE\_UINT32 unCount;

tagGetBadgeGaintInfo getBadgeGaintinfo[1];//加成详细信息

}MsgGetBadgeGaintResponse;

#define MSG\_DO\_Up\_BadgeGaint\_REQUEST MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x028D)//升级徽章

#define MsgBadgeGaint\_MAX 128

typedef struct tagMsgDoUpBadgeGaintRequest : public MsgBaseRequest

{

ACE\_UINT32 unCount;

ACE\_INT32 unIndex[MsgBadgeGaint\_MAX];//装备索引

ACE\_UINT32 unIds[MsgBadgeGaint\_MAX];//装备id

}MsgDoUpBadgeGaintRequest;

#define MSG\_DO\_Up\_BadgeGaint\_RESPONSE MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x028E)//升级徽章

#define MSG\_DO\_SUAN\_PAN MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x028F)//升级徽章s

#define MSG\_DO\_GET\_Badge\_REQUEST MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP,0x0290)//获取徽章列表

typedef struct tagMsgGETBadgeRequest : public MsgBaseRequest

{

ACE\_UINT32 unPage;//第几页

ACE\_INT32 isClear;//是否清除首次获得

}MsgGETBadgeRequest;

#define MSG\_DO\_GET\_Badge\_RESPONSE MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP,0x0291)//获取徽章列表

#define BADGE\_MAX\_PAGE 64

typedef struct tagMsgGETBadgeResponse : public MsgBaseResponse

{

ACE\_UINT32 unMaxPage;//最大页数

ACE\_UINT32 unPage;//第几页了

ACE\_UINT32 unCount;

BadgeInfo unBadgeInfo[BADGE\_MAX\_PAGE];//当前页的内容

}MsgGETBadgeResponse;

#define MSG\_PROPS\_LIST\_RESPONSE MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x0292)//道具

typedef struct tagMsgPropsListResponse : public MsgBaseResponse

{

ACE\_UINT32 unCount;

PropsInfo list[1];

}MsgPropsListResponse;

#define MSG\_INVITE\_FRIEND\_RESPONSE MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x0293)//添加好友用于返回值判断

//获取用户信息

#define MSG\_GET\_CHARACTER\_INFO\_REQUEST MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x0294)

typedef struct tagMsgGetCharacterInfoRequest : public MsgBaseRequest

{

ACE\_UINT32 unCharacterID;//要请求的id

}MsgGetCharacterInfoRequest;

#define MSG\_GET\_CHARACTER\_INFO\_RESPONSE MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x0295)

typedef struct tagMsgGetCharacterInfoResponse : public MsgBaseResponse

{

CharacterInfoExt info;

}MsgGetCharacterInfoResponse;

//-------------------------------------------------------

//-------------------------------------------------------pvp竞技场相关

//获取pvp信息

#define MSG\_GET\_PVP\_INFO\_REQUEST MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x0296)//获取pvp信息

#define MSG\_GET\_PVP\_INFO\_RESPONSE MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x0297)//获取pvp信息

typedef struct tagMsgGetPVPInfoResponse : public MsgBaseResponse

{

PVPInfoEx info;

}MsgGetPVPInfoResponse;

#define MSG\_PVP\_ENTER\_FIGHT\_REQUEST MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x0298)//开始战斗请求

typedef struct tagMsgPVPEnterFightRequestEx : public MsgBaseResponse

{

int isReFreash;//是否刷新

}MsgPVPEnterFightRequestEx;

typedef struct tagMsgPVPEnterFightRequest : public MsgBaseRequest

{

int isReFreash;

ACE\_UINT32 m\_characterID;

HLInfo quitList[3];//排除列表

}MsgPVPEnterFightRequest;

#define MSG\_PVP\_ENTER\_FIGHT\_RESPONSE MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x0299)//开始战斗返回

typedef struct tagMsgPVPEnterFightResponse : public MsgBaseResponse

{

int isReFreash;

PVPInfo info;//最新信息

HLInfoArray \_array;//上战列表

}MsgPVPEnterFightResponse;

#define MSG\_PVP\_GET\_ORDER\_REQUEST MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x029A)//获取pvp排名信息

#define MSG\_PVP\_GET\_ORDER\_RESPONSE MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x029B)//获取pvp排名信息

typedef struct tagMsgPVPGetOrderResponse : public MsgBaseResponse

{

ACE\_UINT32 unCount;

HLUserInfo userInfoArray[MAX\_HONOUR\_LEVEL\_COUNT];//pvp排名

}MsgPVPGetOrderResponse;

#define MSG\_PVP\_BUY\_TIMES\_REQUEST MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x029C)//买pvp竞技场次数

#define MSG\_PVP\_BUY\_TIMES\_RESPONSE MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x029D)//买pvp竞技场次数

typedef struct tagMsgPVPBuyTimesResponse : public MsgBaseResponse

{

ACE\_UINT32 unMoney;

PVPInfo info;//最新信息

}MsgPVPBuyTimesResponse;

#define MSG\_PVP\_ENTER\_REQUEST MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x029E)//进入

#define MSG\_PVP\_ENTER\_RESPONSE MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x029F)//进入

typedef struct tagMsgPVPEnterResponse : public MsgBaseResponse

{

PVPInfo info;//最新pvp信息

}MsgPVPEnterResponse;

#define MSG\_PVP\_ACCOUNTS\_REQUEST MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x02A0)//pvp结算请求

typedef struct tagMsgPVPAccountRequest : public MsgBaseRequest

{

FightRecordInfo info;

//FightPlayerInfoBase fpInfo;

}MsgPVPAccountRequest;

#define MSG\_PVP\_ACCOUNTS\_RESPONSE MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x02A1)//pvp结算请求返回

typedef struct tagMsgPVPAccountResponse : public MsgBaseResponse

{

ACE\_INT32 score;//得分

PVPInfo info;//最新pvp信息

}MsgPVPAccountResponse;

//FightRecordInfo

#define MSG\_PVP\_GET\_FightRecordInfo\_REQUEST MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x02A2)//pvp获取战绩

typedef struct tagMsgPVPFightRecordInfoRequest : public MsgBaseRequest

{

ACE\_UINT32 m\_characterID;

}MsgPVPFightRecordInfoRequest;

#define MSG\_PVP\_GET\_FightRecordInfo\_RESPONSE MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x02A3)//pvp获取战绩

typedef struct tagMsgPVPFightRecordInfoResponse : public MsgBaseResponse

{

PVPInfo pvpinfo;//最新pvp信息

FightRecordInfoList info;//战绩

}MsgPVPFightRecordInfoResponse;

//--------------------------------------------------------------end

#define MSG\_DO\_Up\_BadgeGaint\_REQUESTEX MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x02A4)//升级徽章图鉴

typedef struct tagMsgDoUpBadgeGaintRequestEx : public MsgBaseRequest

{

ACE\_UINT32 unCount;

ACE\_UINT32 unIds[MsgBadgeGaint\_MAX];//图鉴列表

ACE\_INT32 unIndex[MsgBadgeGaint\_MAX];//多少个

}MsgDoUpBadgeGaintRequestEx;

#define MSG\_DO\_Up\_BadgeGaint\_RESPONSEEX MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x02A5)//升级徽章

//------------------闯关最高伤害排行榜相关

// #define MSG\_FIGHT\_ROOLE\_REQUEST MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x02A6)//更新闯关最高伤害排行榜

//

// #define MSG\_FIGHT\_ROOLE\_LIST\_REQUEST MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x02A8)//获取闯关最高伤害排行榜

// #define MSG\_FIGHT\_ROOLE\_LIST\_RESPONSE MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x02A9)//闯关最高伤害排行榜

//------------------------------------------end

//徽章信息相关

#define MSG\_Update\_badge\_info\_REQUESTE MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x02A6)//更新徽章属性

typedef struct tagMsgUpdateBadgeInfoRequest : public MsgBaseRequest

{

ACE\_UINT32 unCharacterID;

BadgeShuxingS bsInfos;//属性信息

tagMsgUpdateBadgeInfoRequest()

{

ZeroVar(\*this);

}

}MsgUpdateBadgeInfoRequest;

#define MSG\_Update\_badge\_info\_RESPONSE MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x02A76)//更新徽章属性

#define MSG\_Get\_badge\_info\_REQUESTE MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x02A8)//获取徽章属性

typedef struct tagMsgGetBadgeInfoRequest : public MsgBaseRequest

{

ACE\_UINT32 unCharacterID;

}MsgGetBadgeInfoRequest;

#define MSG\_Get\_badge\_info\_RESPONSE MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x02A9)//获取徽章属性

typedef struct tagMsgGetBadgeInfoResponse : public MsgBaseResponse

{

ACE\_UINT32 unCharacterID;

BadgeShuxingS bsInfos;//属性信息

}MsgGetBadgeInfoResponse;

//----------------------------------------------------end

#define MSG\_Get\_active\_inofo\_REQUEST MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x02AA)//任务刷新//

#define MSG\_Get\_active\_inofo\_RESBONSE MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x02AB)//任务刷新

typedef struct tagMsgGetActiveInfoResponse : public MsgBaseResponse

{

ACE\_UINT32 key;//要显示红点的值

}MsgGetActiveInfoResponse;

#define MSG\_Get\_POWER\_REQUEST MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x02AC)//更新战力//

typedef struct tagMsgGetPowerRequest : public MsgBaseRequest

{

ACE\_INT32 power;//当前战斗力

}MsgGetPowerRequest;

#define MSG\_Get\_POWER\_RESBONSE MAKE\_MSG\_TYPE(GAME\_BASE\_PROTOCOL\_GROUP, 0x02AD)

typedef struct tagMsgGetPowerResponse : public MsgBaseResponse

{

ACE\_INT32 power;//当前战斗力

}MsgGetPowerResponse;

服务器消息定义：DB server

#define MSG\_DB\_LOG\_INFO\_REQUEST MAKE\_MSG\_TYPE(DB\_PROTOCOL\_GROUP, 0x0063)//向数据库写日志请求

#define MSG\_DB\_LOG\_INFO\_MAX\_LEN 20

typedef struct tagMsgDBLogInfoRequest

{

ACE\_UINT32 unCount;

LogItemInfo logInfoItem[MSG\_DB\_LOG\_INFO\_MAX\_LEN];//日志信息

}MsgDBLogInfoRequest;

#define MSG\_DB\_ADMINCOMMAND\_LOG\_INFO\_REQUEST MAKE\_MSG\_TYPE(DB\_PROTOCOL\_GROUP, 0x0070)//获取日志信息请求

typedef struct tagMsgDBAdminLogInfoRequest

{

tagMsgDBAdminLogInfoRequest()

{

ZeroVar(\*this);

}

ACE\_UINT32 m\_unType;//log类别0表示忽略此项

ACE\_UINT32 m\_unUserID;//帐号查询0表示忽略此项

ACE\_UINT32 m\_unLevel;//等级查询0表示忽略此项

ACE\_UINT32 m\_goodsID;//装备ID查询0表示忽略此项

ACE\_INT32 m\_op\_action;//动作发生原因查询0表示忽略此项

int m\_pageNum;//显示页数每次以项为一页

ObjAdr m\_srcAdr;

}MsgDBAdminLogInfoRequest;

#define MSG\_DB\_ADMINCOMMAND\_LOG\_INFO\_RESPONSE MAKE\_MSG\_TYPE(DB\_PROTOCOL\_GROUP, 0x0071)//获取日志信息结果

#define MSG\_DB\_ADMINCOMMAND\_LOG\_MAX\_COUNT 100

typedef struct tagMsgDBAdminLogInfoResponse

{

ACE\_UINT32 m\_unCount;

LogItemInfo m\_logInfoItem[MSG\_DB\_ADMINCOMMAND\_LOG\_MAX\_COUNT];//日志内容

}MsgDBAdminLogInfoResponse;

// #define MSG\_DB\_UPDATE\_TASK\_LIST MAKE\_MSG\_TYPE(DB\_PROTOCOL\_GROUP, 0x0048)

// #define MSG\_DB\_INSERT\_TASK\_LIST MAKE\_MSG\_TYPE(DB\_PROTOCOL\_GROUP, 0x0049)

#define MSG\_DB\_UPDATE\_LA\_PROGRESS\_REQUEST MAKE\_MSG\_TYPE(DB\_PROTOCOL\_GROUP, 0x0072)//限时活动任务

#define MSG\_DB\_UPDATE\_LA\_PROGRESS\_RESPONSE MAKE\_MSG\_TYPE(DB\_PROTOCOL\_GROUP, 0x0073)

#define MSG\_DB\_UPDATE\_LA\_PROGRESS\_RESPONSE1 MAKE\_MSG\_TYPE(DB\_PROTOCOL\_GROUP, 0x0074)

typedef struct tagMsgDBLimitedActivityList

{

ACE\_UINT32 m\_unCount;

LimitedActivityInfo m\_LAList[MAX\_LIMITED\_ACTIVITY\_COUNT];//限时活动列表

}MsgDBLimitedActivityList;

#define MSG\_DB\_UPDATE\_GOLDEN\_TOUCH\_REQUEST MAKE\_MSG\_TYPE(DB\_PROTOCOL\_GROUP, 0x0075)

typedef struct tagMsgDBUpdateGoldenTouchRequest

{

ACE\_UINT32 m\_unUsedCount;

}MsgDBUpdateGoldenTouchRequest;

#define MSG\_DB\_ADMINCOMMAND\_RECHARGE\_INFO\_REQUEST MAKE\_MSG\_TYPE(DB\_PROTOCOL\_GROUP, 0x0076)//充值请求

typedef struct tagMsgDBAdminRechargeRequest

{

tagMsgDBAdminRechargeRequest()

{

ZeroVar(\*this);

}

char orderID[RECHARGE\_ORDERID\_LEN+1];//充值头

ACE\_UINT32 unProductCount;//购买商品数量

ACE\_UINT32 unCharacterID;//充值用户id

ACE\_UINT32 unPayStats;//支付状态

ACE\_UINT32 unProductId;//商品id

double unRenminbi;//充值金额(人民币)

char strChannel[RECHARGE\_CHANNEL\_LEN + 1];//充值渠道id

char deal\_time[33];//处理时间

}MsgDBAdminRechargeRequest;

#define MSG\_DB\_ADMINCOMMAND\_RECHARGE\_INFO\_RESPONSE MAKE\_MSG\_TYPE(DB\_PROTOCOL\_GROUP, 0x0077)//充值请求返回结果

typedef struct tagMsgDBAdminRechargeResponse: public MsgDBBaseResponse

{

tagMsgDBAdminRechargeResponse()

{

ZeroVar(\*this);

}

char orderID[RECHARGE\_ORDERID\_LEN];//充值头

int unErrorflag;//附加标记位，主要用于错误类型标记

}MsgDBAdminRechargeResponse;// #define MSG\_DB\_ADMINCOMMAND\_Announcement\_REQUEST MAKE\_MSG\_TYPE(DB\_PROTOCOL\_GROUP, 0x0073)//公告请求

// typedef struct tagMsgDBAdminAnnouncementRequest

// {

// ACE\_UINT32 m\_unOpType;

// ACE\_UINT32 m\_unCount;

// char m\_strBuffer[MAX\_ADMIN\_OP\_BUFFER\_ANNOUNCEMENT\_LENGTH+1];

//}MsgDBAdminAnnouncementRequest;

#define MSG\_DB\_GM\_ADD\_REQUEST MAKE\_MSG\_TYPE(DB\_PROTOCOL\_GROUP, 0x0078)//GM DB 添加请求

typedef struct tagMsgGMAddRequest

{

GMInfo gm;//gm内容

}MsgGMAddRequest;

#define MSG\_DB\_UPDATE\_BUY\_ENERGY\_REQUEST MAKE\_MSG\_TYPE(DB\_PROTOCOL\_GROUP, 0x0079)

typedef struct tagMsgDBUpdateBuyEnergyRequest

{

ACE\_UINT32 m\_unCurrentCount;

}MsgDBUpdateBuyEnergyRequest;

#define MSG\_DB\_ORDER\_REQUEST MAKE\_MSG\_TYPE(DB\_PROTOCOL\_GROUP, 0x0080)

typedef struct tagMsgDBOrderRequest

{

ACE\_UINT32 unSig;

ACE\_UINT32 unSN;

ACE\_INT32 product\_id;//商品id

ACE\_INT32 product\_count;//商品数量

}MsgDBOrderRequest;

#define MSG\_DB\_ORDER\_RESPONSE MAKE\_MSG\_TYPE(DB\_PROTOCOL\_GROUP, 0x0081)

typedef struct tagMsgDBOrderResponse: public MsgDBOrderRequest

{

char orderID[RECHARGE\_ORDERID\_LEN+1];//充值头

ACE\_UINT32 m\_unID;

ACE\_INT32 nResult;

ACE\_UINT32 unSig;

}MsgDBOrderResponse;

#define MSG\_DB\_UPDATE\_VIP\_GIFT\_INFO MAKE\_MSG\_TYPE(DB\_PROTOCOL\_GROUP, 0x0082)//更新已领取vip礼包

typedef struct tagMsgDBVipGiftInfo

{

VipGift m\_vipGift;//vip礼包内容

}MsgDBVipGiftInfo;

#define MSG\_DB\_UPDATE\_DELERE\_FRIEND\_LIST MAKE\_MSG\_TYPE(DB\_PROTOCOL\_GROUP, 0x0083)//更新删除好友列表

//徽章系统

#define MSG\_DB\_UPDATE\_BADGE\_PROGRESS\_RESPONSE MAKE\_MSG\_TYPE(DB\_PROTOCOL\_GROUP, 0x0084)//更新徽章系统

#define MSG\_DB\_UPDATE\_BADGE\_PROGRESS\_RESPONSE1 MAKE\_MSG\_TYPE(DB\_PROTOCOL\_GROUP, 0x0085)//更新徽章系统

typedef struct tagMsgDBBadgeList

{

ACE\_UINT32 m\_unCount;

BadgeInfo m\_BadgeList[1];//徽章内容

}MsgDBBadgeList;

#define MSG\_DB\_PROPS\_LIST MAKE\_MSG\_TYPE(DB\_PROTOCOL\_GROUP, 0x0086)//获取道具

typedef struct tagMsgDBPropsListResponse : public MsgDBBaseResponse

{

ACE\_UINT32 unCount;

PropsInfo list[1];

}MsgDBPropsListResponse;

#define MSG\_DB\_UPDATE\_PROPS\_LIST MAKE\_MSG\_TYPE(DB\_PROTOCOL\_GROUP, 0x0087)//更新道具

typedef struct tagMsgDBUpdatePropsListRequest

{

ACE\_UINT32 unCount;

PropsInfo list[1];

}MsgDBUpdatePropsListRequest;