

# 淘米数据平台接入指南-1.0.0

C01 研发部-统计平台项目组

2013 年 11 月

## 目录

淘米数据平台接入指南-1.0.0 .....	1
C01 研发部-统计平台项目组 .....	1
2013 年 11 月 .....	1
一、综述.....	2
1、适用范围.....	2
2、统计标准.....	2
3、问题联系人.....	2
二、接入流程.....	2
1、为游戏申请 GameID.....	2
2、向工程中添加 logger 库文件.....	3
3、添加调用方法.....	3
4、进行数据测试.....	3
三、添加调用方法(C++) .....	3
0、构造函数.....	3
1、注册米米号.....	3
2、登录游戏验证用户名和密码.....	4
3、创建游戏中角色.....	5
4、登录游戏服务器(online 服务器) .....	6
5、登录设备数(b03 使用) .....	7
6、用户退出游戏服务器(online 服务器) .....	8
7、统计当前在线人数.....	8
8、用户升级.....	9
9、付费.....	9
10、获得游戏内一级币(这里的获得是指通过游戏内部赠送或完成任务奖励，而非用户购买) .....	10
11、使用游戏内一级货币购买道具 .....	11
12、消耗游戏内一级货币.....	11
13、接收任务.....	12
14、完成任务.....	12
15、放弃任务.....	12
16、退订 VIP 服务(取消自动续费).....	13
17、销户(VIP 时间到期).....	13
18、自定义统计 .....	14
18.1、概述.....	14

18.2、示例.....	15
四、添加调用方法(PHP).....	18
4.1、注册角色.....	18
4.2 用户登录游戏(做活跃统计) .....	19
4.3、付费(开通 VIP、米币购买道具、人民币购买游戏内一级货币) .....	19
4.4、自定义统计项接口(只支持统计人数人次) .....	21
五、web 端统计或 AS 统计.....	21

## 一、综述

### 1、适用范围

淘米数据平台帮助各游戏部门解决玩家数据收集到数据标准化分析的全部繁琐过程，以行业标准形式展现于报表中。

SDK 适用于 C++程序开发的后台程序调用。

### 2、统计标准

统计中我们采用游戏自身的账户(米米号)作为一个玩家单元。

付费: B01 是指通过米币购买道具和开通 VIP 包月服务

B03 是指通过人民币购买游戏内货币

### 3、问题联系人

3.1、系统不支持的需求以及系统中指标不明确的地方，请联系 C01 统计平台项目组 berry

3.2、使用方式和显示方面的问题， 请联系 C01 研发部 violet

3.3、接入方式有不明确的地方，请联系 C01 研发部 henry

## 二、接入流程

### 1、为游戏申请 GameID

申请地址：预先创建一款游戏，您将获得一个整型的 gameid,用于唯一标识您的这款游戏

## 2、向工程中添加 logger 库文件

下载统计的 logger 压缩包并解压至本地,将其中的 lib32bitstatlogger.a(32 位)或 lib64bitstatlogger.a(64 位)和 statlogger.h 文件加入您的工程代码中。

下载地址:

<http://10.1.1.60/dev/doku.php?id=10.1.1.104>

## 3、添加调用方法

参考“三、添加调用方法”

## 4、进行数据测试

调用方法添加完毕后,应当进行数据测试,以确保方法调用正确。

测试方法:

## 三、添加调用方法(C++)

先初始化一个 StatLogger 类的实例,只需要每次后台服务启动时初始化一次即可,建议做成全局的。

StatLogger logger(1);

统计数据一律强制写入/opt/taomee/stat/data/目录下,需要提前建立该目录,并将该目录开通 777 的权限,否则程序无法启动

## 0、构造函数

```
StatLogger( int   game_id, //游戏 ID,由数据分析平台统一分配
            int16_t zone_id = -1, //区 ID, 默认-1 表示全区或不分区
            int16_t svr_id = -1, //服 ID, 默认-1 标识全服或不分服
            int16_t site_id = -1, //平台 ID, 默认为-1 代表全平台,如果这款游戏将来或者现在
            可能放到其他外部平台运营, 那么这里填上对应平台的 ID, 淘米平台用 1 表示
            int isgame = 1 //标识日志是来自游戏后台(1)还是公共平台组(0);游戏后台无需传
            该参数, 采用默认值即可
            );
```

## 1、注册米米号

接口定义

```
void reg_account(std::string acct_id,
                uint32_t cli_ip,
                std::string ads_id,
                std::string browser = "",
                std::string device = "",
                std::string os = "",
                std::string resolution = "",
                std::string network = "",
                std::string isp = "" );
```

#### 参数说明

acct_id	用户账户(米米号)
cli_ip	用户的 IP 地址,无法获取时取值 0
ads_id	用户是从哪个广告渠道跳转过来的
browse	用户使用的浏览器
device	用户使用的设备
os	用户使用的操作系统
resolution	用户屏幕的分辨率
network	用户使用的网络
isp	用户网络的服务提供商

#### 用途

- 1、统计项名称(std): `_newuid_` 子统计项名称(sstd): `_newuid_`  
 用于统计每日注册米米号的数量以及这些用户的地区分布  
 每天从各个广告渠道带来的注册米米号数量  
 使用不同的浏览器注册的米米号数量  
 使用不同的设备注册的米米号数量  
 使用不同的操作系统、分辨率、网络以及网络服务提供商注册的米米号数量

#### 用法

```
logger.reg_account("47159775", "102.78.34.112","4399.com", "firefox", "ipad2.0",
"ubuntu10", "1024*768", "3G", "电信");
```

## 2、登录游戏验证用户名和密码

#### 接口定义

```
void verify_passwd (std::string acct_id,
                   uint32_t cli_ip,
                   std::string ads_id,
                   std::string browser = "",
                   std::string device = "",
                   std::string os = "",
                   std::string resolution = "",
                   std::string network = "",
                   std::string isp = "" );
```

#### 参数说明

acct_id	用户账户(米米号)
cli_ip	用户的 IP 地址, 无法获取时取值 0
ads_id	用户是从哪个广告渠道跳转过来的
browse	用户使用的浏览器
device	用户使用的设备
os	用户使用的操作系统
resolution	用户屏幕的分辨率
network	用户使用的网络
isp	用户网络的服务提供商

#### 用途

- 1、统计项名称(std): `_veripass_` 子统计项名称(sstd): `_veripass_`  
统计每天验证用户名和密码的人数人次以及这些用户的地区分布  
从各个广告位过来的用户验证用户名和密码的人数人次  
使用不同的浏览器进行验证用户名和密码的人数人次  
使用不同设备、操作系统、分辨率、网络以及网络服务提供商进行验证用户名和密码的人数人次

#### 用法

`logger.verify_passwd("47159775", "106.235.12.11", "innermedia.taomee.seer.topbar");//不按照浏览器、设备等维度进行统计分析`

### 3、创建游戏中角色

#### 接口定义

```
void reg_role(std::string acct_id,  
             std::string player_id,  
             std::string race,  
             uint32_t cli_ip,  
             std::string ads_id,  
             std::string browser = "",  
             std::string device = "",  
             std::string os = "",  
             std::string resolution = "",  
             std::string network = "",  
             std::string isp = "");
```

#### 参数说明

acct_id	用户账户(米米号)
player_id	用户角色标识(对于单角色游戏以及不需要分角色查看数据的, 该字段可赋空值),通常用用户创建角色的时间标识
race	用户选择的职业(对于不区分职业或不需要按职业进行统计, 该字段可赋空值)

cli_ip	用户的 IP 地址, 无法获取时取值 0
ads_id	用户是从哪个广告渠道跳转过来的
browse	用户使用的浏览器
device	用户使用的设备
os	用户使用的操作系统
resolution	用户屏幕的分辨率
network	用户使用的网络
isp	用户网络的服务提供商

## 用途

1、统计项名称(std): `_newac_` 子统计项名称(sstid): `_newac_`

统计每天新注册的用户账户数以及这些账户的地区分布、使用的浏览器、设备、操作系统、分辨率、网络、网络服务提供商的人数人次

2、统计项名称(std): `_newpl_` 子统计项名称(sstid): `_newpl_`

统计每天新注册的角色数, 即根据 `acct_id` 和 `player_id` 来唯一标识一个用户

3、统计项名称(std): `_newrace_` 子统计项名称(sstid): 具体的职业名称

统计每天各个职业的新增用户数,根据 `acct_id` 来标识

## 用法

`logger.reg_role("47159775", "1383019208", " 魔 法 师 ", " 61.155.182.56", "innermedia.taomee.mole.banner");`//这里不按照浏览器、设备等指标统计; 如果项目部需要, 可自行带上相应参数即可

## 4、登录游戏服务器(online 服务器)

### 接口定义

```
void login_online(std::string acct_id,
                 std::string player_id,
                 std::string race,
                 bool isvip,
                 int lv,
                 uint32_t cli_ip,
                 std::string ads_id,
                 std::string zone = "",
                 std::string browser = "",
                 std::string device = "",
                 std::string os = "",
                 std::string resolution = "",
                 std::string network = "",
                 std::string isp = "");
```

### 参数说明

acct_id	用户账户(米米号)
player_id	用户角色标识(对于单角色游戏以及不需要分角色查看数据的, 该字段可赋空值),通常用

	用户创建角色的时间标识
race	用户选择的职业(对于不区分职业或不需要按职业进行统计, 该字段可赋空值)
isvip	是否 VIP 用户 (true:是 false: 否)
lv	用户当前等级
cli_ip	用户的 IP 地址, 无法获取时取值 0
ads_id	用户是从哪个广告渠道跳转过来的
zone	默认表示总体登录; 主要是提供给页游这边不区分区服的游戏使用, 用于使用电信或网通登录人数用的
browse	用户使用的浏览器
device	用户使用的设备
os	用户使用的操作系统
resolution	用户屏幕的分辨率
network	用户使用的网络
isp	用户网络的服务提供商

#### 用途

1、统计项名称(stdid): `_lgac_` 子统计项名称(sstid): `_lgac_`

统计帐号登录人数人次, 也就是活跃用户以及活跃用户的地区分布

VIP 和非 VIP 用户的登录人数人次

每个等级的用户登录人数人次

使用不同的浏览器登录人数人次, 不同设备登录人数人次 不同操作系统、分辨率、网络以及网路服务提供商登录人数人次

2、统计项名称(stdid): `_lgpl_` 子统计项名称(sstid): `_lgpl_`

根据角色统计登录人数人次, 已 `acct_id` 和 `player_id` 唯一标识用户

3、统计项名称(stdid): `_lgrace_` 子统计项名称(sstid): 各职业名称

统计各个职业每天的登录人数人次

#### 用法

```
logger.login_online("47159775", "1383019208", "魔法师", true, 15, "115.12.116.57", "", "ie8", "iphone4", "debian", "1280*768", "wifi", "网通");
```

```
logger.login_online("47159775", "1383019208", "", true, 15, "115.12.116.57", "网通一区", "ie8", "iphone4", "debian", "1280*768", "wifi", "网通");
```

## 5、登录设备数(b03 使用)

#### 接口定义

```
void start_device(std::string device_id);
```

#### 参数说明

Device_id	设备 ID
-----------	-------

#### 用途

统计项名称(stdid): `_startdev_` 子统计项(sstid): `_startdev_`

可统计每日登录设备数, 经过加工可以统计出每日新增设备数

用法:

```
logger.start_device("asfdg23432q52345234534");
```

## 6、用户退出游戏服务器(online 服务器)

接口定义

```
void logout(std::string acct_id, bool isvip, int lv, int oltime);
```

参数说明

acct_id	用户账户(米米号)
isvip	是否 VIP 用户(true:是 false:否)
lv	用户退出时的等级
oltime	本次用户总共的在线时长

用途

- 1、统计项名称(std): \_logout\_ 子统计项名称(sstid): \_logout\_  
可统计每天登出游戏的人数人次、总的在线时长(单位秒)以及时长区间分布,默认的时长区间划分为:

0-10

11-60

61-300

301-600

601-1200

1201-1800

1801-2400

2401-3000

3001-3600

3601-7200

7201 以上

用法

```
logger.logout("47159775", true, 13,3204);
```

## 7、统计当前在线人数

接口定义

```
void online_count(int cnt, std::string zone="");
```

参数说明

cnt	当前在线人数
zone	默认表示总体在线人数; 该字段主要是提供给页游不区分区服类的游戏使用的, 用于统计电信或网通的在线

用途

统计项名称(std): \_olcnt\_ 子统计项名称(sstid): \_olcnt\_



统计游戏当前在线人数

#### 用法

```
logger.online_count(103476);//每分钟至少调用一次
```

## 8、用户升级

#### 接口定义

```
void level_up(std::string acct_id, std::string race, int lv);
```

#### 参数说明

acct_id	用户帐号(米米号)
race	用户职业(如果不需要区分职业, 赋值为空即可"")
lv	升之后的等级

#### 用途

- 1、统计项名称(std): `_aclvup_` 子统计项名称(sstd): `_aclvup_`  
统计每天的用户等级分布
- 2、统计项名称(std): `_racelvup_` 子统计项名称(sstd): `_racelvup_`  
分职业统计用户的等级分布

#### 用法

每次用户升级时

```
logger.level_up("47159775", "", 20);
```

## 9、付费

#### 接口定义

```
void pay(std::string acct_id,  
        bool isvip,  
        float pay_amount,  
        CurrencyType currency,  
        PayReason pay_reason,  
        std::string outcome,  
        int outcnt,  
        std::string pay_channel = "米币帐户");
```

#### 参数说明

acct_id	用户账户(米米号)
isvip	是否 VIP 用户(true:是 false:否)
pay_amount	付费额度,单位统一为"分"
currency	货币类型(ccy_mibi:米币 ccy_cny:人民币),如是其他货币类型, 请做转换, 一个游戏中不可交替使用米币和人民币
pay_reason	支付原因(pay_vip:开通 VIP 包月 pay_buy:购买道具 pay_charge:充值游戏币 pay_free:赠送)

outcome	产出的物品 (pay_vip: 可以是包月的类型 pay_buy: 可以是道具的名称或 ID pay_charge: 填写游戏币名称即可)
outcnt	购买道具或游戏币的数量 (pay_vip: 填写包月 时长)
pay_channel	支付渠道 (“支付宝”、“财付通”等)

#### 用途

- 1、统计项名称(std): `_acpay_` 子统计项名称(sstid): `_acpay_`  
可统计付费总额、VIP 和非 VIP 用户分别的付费总额、各个支付渠道付费总额、每种货币单位的付费总额
- 2、统计项名称(std): `_acpay_` 子统计项名称(sstid): `_buycoins_vipmonth buyitem`  
可统计三个子统计项的付费总额、VIP 和非 VIP 用户分别的付费总额、各个支付渠道付费总额、每种货币单位的付费总额
- 3、统计项名称(std): `_getgolds_` 子统计项名称(sstid): `_userbuy_`  
用户购买产出的金币总量
- 4、统计项名称(std): `_buyvip_` 子统计项名称(sstid): `_buyvip_`  
各个包月时长的人数人次以及付费总额
- 5、统计项名称(std): `_buyitem_` 子统计项名称(sstid): `_mibiitem_`  
可统计通过米币购买的道具人数人次、销售数量、销售金额

#### 用法

B01:

```
logger.pay("47159775", true, 1000, ccy_mibi, pay_vip, "1 个月 VIP", 1, "短信");//包月
```

logger.pay("47159775", true, 2000, ccy\_mibi, pay\_buy, "圣诞套装", 20);//米币购买道具 这里不需要在调用一次购买道具的接口了

B03:

```
Logger.pay("47159775", true, 300, ccy_cny, pay_charge, "游戏金币", 30);//购买游戏金币;这里不需要在调用一次获得游戏金币的接口了
```

## 10、获得游戏内一级币(这里的获得是指通过游戏内部赠送或完成任务奖励，而非用户购买)

#### 接口定义

```
void obtain_golds(std::string acct_id, int amt);
```

#### 参数说明

acct_id	用户账户(米米号)
amt	获得金币的数量,单位统一为“分”(1~1 千万)

#### 用途

- 1、统计项名称(std): `_getgolds_` 子统计项名称(sstid): `_systemsend_`  
统计通过各种系统赠送途径获得金币的人数人次以及总数量

#### 用法

比如在 IMOLE 游戏里面通过完成某个任务奖励了 **1000** 个贝壳  
logger.obtain\_golds("47159876", **100000**); //接口中的数量需要乘以 100

## 11、使用游戏内一级货币购买道具

### 接口定义

```
void buy_item(std::string acct_id, bool isvip, int lv, float pay_amount, std::string outcome, int outcnt);
```

### 参数说明

acct_id	用户账户(米米号)
is_vip	是否 VIP 用户(true:是 false: 否)
lv	用户购买道具的时的等级
pay_amount	支付数量,单位统一为"分"
outcome	购买的道具
outcnt	购买的道具数量

### 用途

1、统计项名称(std): \_buyitem\_

子统计项名称(sstid): \_coinsbuyitem\_

可统计通过非米币渠道购买的道具人数人次、销售数量、销售金额

### 用法

Logger.buy\_item("34159876", true, 13, **2000**, "元旦礼包", 10); //用户花了 20 游戏币购买 10 个元旦礼包

通常这里只有 B03 手游这边会调用, B01 的米币购买道具在发送 pay 信息的时候, 已经包含该信息了。

## 12、消耗游戏内一级货币

### 接口定义

```
void use_golds(std::string acct_id, bool isvip, std::string reason, int amt, int lv);
```

### 参数说明

acct_id	用户帐号(米米号)
isvip	是否 VIP 用户(true:是 false:否)
reason	原因 (购买道具、开启新功能等)
amt	支付数量,单位统一为"分"
lv	用户等级

### 用途

1、统计项名称(std): \_userglods\_ 子统计项(sstid): reason

可统计各种原因消耗金币的数量, VIP 和非 VIP 用户消耗的金币数量以及各等级用户消耗的数量

### 用法

logger.user\_golds("47169879", true, "\_开启新功能\_", **1800**, 17); //用户花 18 游戏币开启新功能

//用户购买道具时不需要调用该接口哦

### 13、接收任务

接口定义

```
void accept_task(TaskType type, std::string acct_id, std::string task_name, int lv);
```

参数说明

type	任务类型(新手任务: task_newbie 主线任务: task_story 支线任务:task_supplement )
acct_id	用户(米米号)
task_name	任务名称
lv	接收任务时的等级

用途

- 1、统计项名称(std): \_getnbtsk\_ \_getmaintsk\_ \_getauxtsk\_ \_getetctsk\_  
子统计项名称(sstid): 任务名称  
可统计各个任务的接收人数人次

用法

```
Logger.accept_task(task_newbie, "3781654", "打开背包",20);
```

### 14、完成任务

接口定义

```
void finish_task(TaskType type, std::string acct_id, std::string task_name, int lv);
```

参数说明

type	任务类型(新手任务: task_newbie 主线任务: task_story 支线任务:task_supplement )
acct_id	用户(米米号)
task_name	任务名称
lv	完成任务时的用户等级

用途

- 1、统计项名称(std): \_donenbtsk\_ \_donemaintsk\_ \_doneauxtsk\_ \_doneetctsk\_  
子统计项名称(sstid): 任务名称  
可统计各个任务的完成人数人次以及完成任务时的用户等级分布

用法

```
logger.accept_task(task_newbie, "3781654", "打开背包", 18);
```

### 15、放弃任务

接口说明

```
void abort_task(TaskType type, std::string acct_id, std::string task_name, int lv);
```

参数说明

type	任务类型(新手任务: task_newbie 主线任务: task_story 支线任务:task_supplement )
------	--

acct_id	用户(米米号)
task_name	任务名称
lv	放弃任务时的等级

#### 用途

- 统计项名称(stdid): \_abrtnbtsk\_ \_abrtmaintsk\_ \_abrtauxtsk\_ \_abrtetctsk\_  
子统计项名称(sstid): 任务名称  
可统计各个任务的放弃人数人次,以及放弃任务时的用户等级分布

#### 用法

```
logger.accept_task(task_newbie, "3781654", "打开背包", 19);
```

## 16、退订 VIP 服务(取消自动续费)

#### 接口说明

```
void unsubscribe(std::string acct_id, UnsubscribeChannel uc);
```

#### 参数说明

acct_id	用户(米米号)
uc	渠道, 目前只有短信和米币两个渠道 (StatLogger::uc_duanxin 和 StatLogger::uc_mibi)

#### 用途

统计项名称(stdid): \_unsub\_  
子统计项名称(sstid): \_unsub\_  
对 UC 字段做 item 运算

#### 用法

```
Logger.unsubscribe("342352345", StatLogger::uc_duanxin);  
Logger.unsubscribe("342352345", StatLogger::uc_mibi);
```

## 17、销户(VIP 时间到期)

#### 接口说明

```
void cancel_acct(std::string acct_id, string channel);
```

#### 参数说明

acct_id	用户(米米号)
channel	销户渠道, 如支付宝、财付通等

#### 用途

统计项名称(stdid): \_ccacct\_  
子统计项名称(sstid): \_ccacct\_  
对 channel 字段做 item 运算

#### 用法

```
Logger.cancel_acct("342352345", "zhifubao");
```

# 18、自定义统计

## 18.1、概述

### 接口定义

```
void log(std::string stat_name, std::string sub_stat_name, std::string acct_id,
        std::string player_id, const StatInfo& info = StatInfo());
```

### 参数说明

stat_name	统计项名称
sub_stat_name	子统计项名称
acct_id	用户账户(米米号)
palyer_id	用户角色标识(如果不需要分角色查看，只能传空字符串即可)
info	附加信息

### 用途

默认用于统计子统计项的人数人次

### 用法

```
StatLogger logger(1);
StatInfo info;
Info.add_info("赛尔豆", 843);
Info.add_op(StatInfo:op_sum, "赛尔豆");
Logger.log("游戏输出", "保护导航仪", "47189678", "", info);
//可统计出参与保护导航仪这个小游戏的人数人次以及输出的赛尔豆总量
```

其中 add\_info 和 add\_op 接口声明如下

```
void add_info(std::string key, float value);
void add_info(std::string key, std::string value);
void add_op(OpCode op, std::string key1, std::string key2 = "");
```

上述接口中的 key 由各项目部自定义，注意需要保证同一个项目内部不冲突，一次调用最多允许 30 个 key-value 对；

对 key 的取值不能以 “\_” 开头或结束，StatInfo 会自动过滤，比如\_KEY\_，会自动被修改成 KEY；不能够有 “=:,;.|\t” 字符中的任何一个，否则程序运行会跳过该统计项；

对 value 的取值不能有 “=|\t” 字符中的任何一个

Key1 key2 必须是已经通过 add\_info 方法添加好了的

OpCode 的取值以及说明如下：

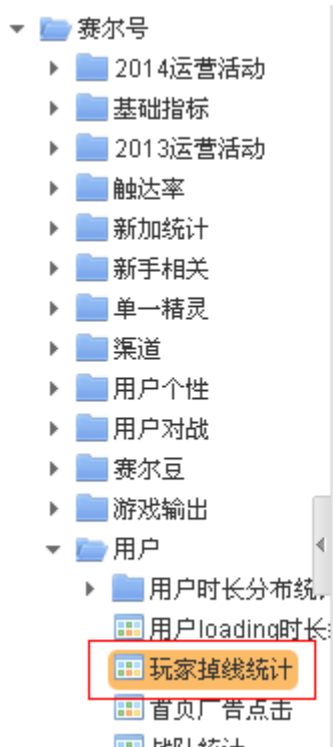
StatInfo::op_sum(key)	对指定 key 的 value 求和，组合键 statname substatname key
StatInfo::op_max(key)	对指定 key 的 value 求最大值；组合键 statname substatname key
StatInfo::op_set(key)	对指定 key 的 value 做 set 做操作；组合键 statname substatname key
StatInfo::op_ucount(key)	对指定 key 的 value 做去重操作；组合键 statname substatname key

StatInfo::op_item(key)	对指定 key 的 value 计算出对应的人数人次; 组合键 statname substatname acctid playerid key
StatInfo::op_item_sum(key1,key2)	按照 key1 分类, 对 key2 的 value 字段求和; 组合键 statname substatname acctid playerid key1 key2
StatInfo::op_item_max(key1, key2)	按照 key1 分类, 对 key2 的 value 字段求最大值; 组合键 statname substatname acctid playerid key1 key2
StatInfo::op_item_set(key1, key2)	按照 key1 分类, 对 key2 的 value 字段做 set 操作, 即取每天最后一个值; 组合键 statname substatname acctid playerid key1 key
StatInfo::op_sum_distr(key)	需要依赖于区间分布, 然后计算出对 key 的 value 字段求和后属于哪个区间的分布; 组合键 statname substatname acctid playerid key
StatInfo::op_max_distr(key)	需要依赖于区间分布, 然后计算出对 key 的 value 字段求最大值后属于哪个区间的分布; ; 组合键 statname substatname acctid playerid key
StatInfo::op_min_distr(key)	需要依赖于区间分布, 然后计算出对 key 的 value 字段求最小值后属于哪个区间的分布; ; 组合键 statname substatname acctid playerid key
StatInfo::op_set_distr(key)	对 key 的 value 字段做 set 操作后做分布; 组合键 statname substatname acctid playerid key1
StatInfo::op_ip_distr(key)	ip 分布, 需要根据 IP 地址查出对应的地区

## 18.2、示例

### 1、默认类型(人数人次)

比如赛尔号->用户->玩家掉线统计 人数人次



增加到我的收藏

### 玩家掉线的人次

全选   反选		2014-03-18	2014-03-17
<input type="checkbox"/>	玩家掉线的人次	1450	2918
<input type="checkbox"/>	玩家掉线的人数	1120	2437

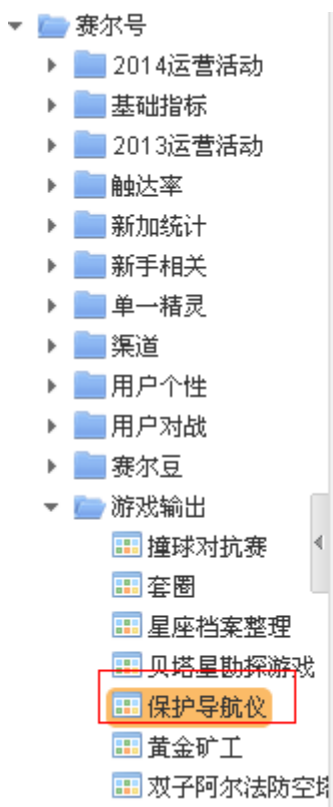
赛尔号游戏 ID 为 2

StatLogger logger(2);

logger.log("用户", "玩家掉线统计", "48197896", "");

2、StatInfo::op\_sum(key) StatInfo::op\_max(key) StatInfo::op\_set(key)

比如赛尔号->游戏输出->撞球对抗赛(人数、人次、产生的赛尔豆)



### 保护导航仪

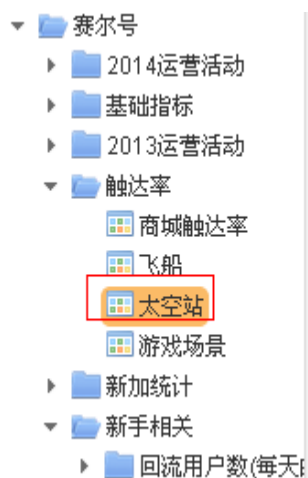
全选   反选		2014-03-18	2014-03-17
<input type="checkbox"/>	保护导航仪赛尔豆输出	416	1773
<input type="checkbox"/>	保护导航仪人次	8	27
<input type="checkbox"/>	保护导航仪人数	4	23



```
StatLogger logger(2);
StatInfo info;
info.add_info("赛尔豆", 100);
info.add_op(StatInfo::op_sum, "赛尔豆");
logger.log("游戏输出", "保护导航仪", "37896574", "", info);
op_max:是对发过来的同一天或同一分钟的数据求最大值
op_set:是对发过来的同一天或同一分钟的数据做 set 操作,即覆盖操作
```

### 3、StatInfo::op\_item(key)

```
StatInfo::op_item_sum(key1,key2)
StatInfo::op_item_max(key1, key2)
StatInfo::op_item_set(key1, key2)
赛尔号->触达率->太空站
```



进入英佩恩堡垒人次				
全选   反选		2014-03-18	2014-03-17	
<input type="checkbox"/>	进入英佩恩堡垒人次	158	4285	
<input type="checkbox"/>	进入英佩恩堡垒人数	75	2524	
<input type="checkbox"/>	进入英佩恩堡垒[场景]人次	158	4285	
<input type="checkbox"/>	进入英佩恩堡垒[场景]人数	75	2524	
<input type="checkbox"/>	进入英佩恩堡垒二层[场景]人次	0	0	
<input type="checkbox"/>	进入英佩恩堡垒二层[场景]人数	0	0	

也可以如下方式:

```
StatLogger logger(2);
StatInfo info;
Info.add_info("item", "进入英佩恩堡垒");
Info.add_info("item", "进入英佩恩堡垒|场景");
Info.add_info("item", "进入英佩恩堡垒二层|场景");
Info.add_op(StatInfo::op_item, "item");
Logger.log(触达率, "太空站", "324234534", "", info);
```

### 4、StatInfo::op\_sum\_distr(key) StatInfo::op\_max\_distr(key) StatInfo::op\_min\_distr(key)

这一项也是用于统计区间分布的,只是这里的区间需要策划或相关开发人员去配置,比如统计今日产出的金豆的区间分布,例如

产出金豆数为 1-100 的人数

产出金豆数为 101-500 的人数

产出金豆数为 501-1000 的人数

产出金豆数大于 1000 的人数

那么就需要在用户产出金豆的时候,按照如下格式发送日志:

```
StatLogger logger(1);
StatInfo info;
```

```
Info.add_info("产出金豆", 56);
```

```
Info.add_op(StatInfo::op_sum_distr, "产出金豆");
```

```
logger.log("小游戏产出", "产出金豆", "34478323", "", info);
```

就会将以用户为单位，计算每个人今日产出的金豆总量，然后看属于哪个区间，做分布

Op\_max\_distr 和 op\_min\_distr 类似,只是分别去每个用户的最大值或最小值而已.

#### 5、StatInfo::op\_set\_distr(key)

这一项用于做等级分布

比如 iseer 用户最高精灵等级分布

```
StatLogger logger(78);
```

```
StatInfo info;
```

```
Info.add("max_level", 19);
```

```
Info.add_op(StatInfo::op_set_distr, "max_level");
```

```
Logger.log("等级分布", "用户最高精灵等级分布", "2341343", "", info);
```

#### 6、StatInfo::op\_ip\_distr(key)

地区分布统计各个地区的人数人次，精确到省份

## 四、添加调用方法(PHP)

Php 这里只提供了注册角色、登录游戏、付费和自定义的接口

### 4.1、注册角色

接口定义:

```
/**
 * @brief stat_reg_role 注册角色
 *
 * @param $gameid      游戏ID
 * @param $platformid  平台ID(不分平台填-1)
 * @param $zoneid      区号(不分区服时填-1)
 * @param $serverid    服号(不分区服时填-1)
 * @param $acct_id     帐号
 * @param $device      设备
 *
 * @return
 */
function stat_reg_role($gameid, $platformid, $zoneid, $serverid, $a
```

## 4.2 用户登录游戏(做活跃统计)

接口定义:

```
/**
 * @brief stat_login_online 活跃
 *
 * @param $gameid      游戏ID
 * @param $platformid  平台ID(不分平台填-1)
 * @param $zoneid      区号(不分区服时填-1)
 * @param $serverid    服号(不分区服时填-1)
 * @param $acct_id     帐号
 * @param $device      设备
 *
 * @return
 */
function stat_login_online($gameid, $platformid, $zoneid, $serverid, $acct_id, $device)
```

## 4.3、付费(开通 VIP、米币购买道具、人民币购买游戏内一级货币)

接口定义: (注意付费金额的单位统一为“分”)

```

/**
 * 用户付费时调用:开通VIP或 购买游戏内道具
 * gameid 游戏ID
 * platformid 平台ID 默认填写-1 表示全平台, 如果一个游戏需要放到不同的平台运营,
 * 这里就需要填写对应的平台ID, 需要与游戏内部保持一致
 * zoneid 游戏区ID 默认填写-1
 * serverid 游戏服ID 默认填写-1
 * acctid 用户帐号ID
 * isvip 是否VIP用户 0: 不是 1: 是
 * pay_amount 本次付费金额
 * currency 本次付费的货币单位 0:米币 1: 人民币
 * pay_reason 可取值 _buyitem_:通过米币购买游戏内道具
 * _vipmonth_:开通游戏内VIP包月
 * _buycoins_:购买游戏内货币
 * output 付费购买的产出物 购买道具时为具体的道具ID或名称
 * 开通VIP时为具体的VIP信息, 比如1个月 3个月
 * 购买游戏币时为游戏币一级货币的名称
 * outcnt 付费购买的产出物数量 购买道具时为道具的数量
 * 开通VIP时为具体的开通时长
 * 购买游戏币时为购买的游戏币数量
 * pay_channel 付费渠道 支付宝 财付通等
 *
 */
function stat_pay($gameid, $platformid, $zoneid, $serverid, $acct_id,
                 $isvip, $pay_amount, $currency, $pay_reason, $output,
                 $outcnt, $paychannel)

```

#### 4.4、自定义统计项接口(只支持统计人数人次)

```
/**
 * 自定义统计项接口 目前只支持人数人次
 * playerid 为角色ID , 默认填写-1即可
 * gameid 游戏ID
 * platformid 平台ID 默认填写-1 表示全平台, 如果一个游戏需要放到不同的平台运营,
 * 这里就需要填写对应的平台ID, 需要与游戏内部保持一致
 * zoneid 游戏区ID 默认填写-1
 * serverid 游戏服ID 默认填写-1
 * acctid 用户帐号ID
 * playerid 用户角色ID 不需要分角色时填写-1即可
 * stname 统计项名称 (如完成任务)
 * sstname 子统计项名称 (如任务1)
 * 下面内容请忽略
 * array 自定义key-value数组,
 * 数组内容格式必须为$arr = array("peter"=>32, "jone"=>34);
 * 其中peter和jone为key 32 34为value
 * key不能以_开头和结尾 不可以包含 = : , | \t ; 中的任意一个字符
 * value 也不可以包含 = : , | \t ; 中的任意一个字符
 */
function stat_logger($gameid, $platformid, $zoneid, $serverid, $acct_id,
                    $playerid, $stname, $sstname)
{
```

## 五、web 端统计或 AS 统计

通过发送 http 请求的形式, 只支持人数人次的统计

http 请 求 地 址 :

[http://misc.taomee.com/weblogger.php?gameid=\\$gameid&stid=\\$stid&ssid=\\$ssid&uid=\\$uid](http://misc.taomee.com/weblogger.php?gameid=$gameid&stid=$stid&ssid=$ssid&uid=$uid)

gameid: 游戏 ID(必须为数字)

stid: 统计项名称(为避免乱码问题, 这里必须填写英文或数字字符串)

ssid: 子统计项名称(为避免乱码问题, 这里必须填写英文或数字字符串)

uid: 用户米米号(获取不到米米号时可不填写, 如果填写必须为英文或数字字符串)