

TDIR 游戏接口使用文档



腾讯科技（深圳）有限公司

版权所有 侵权必究

修订记录

日期	修订版本	描述	作者
2009-2-11	1.0	创建	hardway
2009-11-10	2.0	修改	hardway

目录

1. TDIR 概述.....	4
1.1. 概述.....	4
1.2. 适用范围.....	4
1.3. 词汇表.....	4
2. TDIR 接口.....	4
2.1. 系统拓扑图.....	4
2.2. TDIR 接口定义.....	5
2.2.1. Tbus 上报接口.....	5
2.2.2. tbus 接口使用示例.....	6
2.2.3. udp 上报接口.....	8
2.2.4. udp 接口使用示例.....	9

1. TDIR 概述

1.1. 概述

TDIR 是 tencent directory server 的简称,TDIR 作为统一登陆系统的一部分,为网络游戏登录提供目录服务以及相关服务器状态信息。统一登陆系统介绍详见相关资料。

1.2. 适用范围

本文档为自研类游戏接入目录服务提供接口规范和使用帮助。适用与游戏后台服务器开发人员。

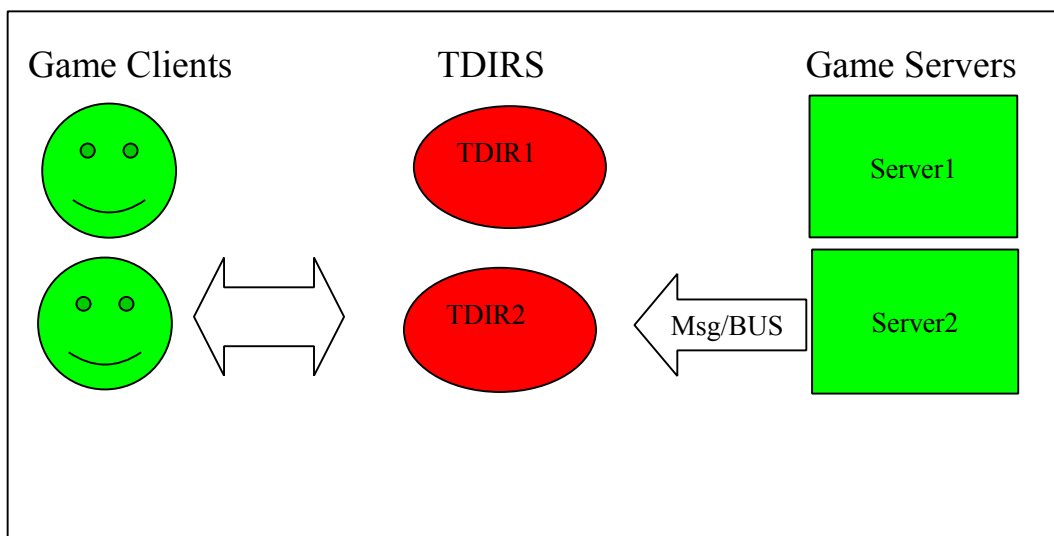
本规范之前的相关标准、规范的内容如与本规范的规定相抵触的,以本规范为准

1.3. 词汇表

词汇	解释
TDIR(Tencent Directory Server)	目录服务器
Game Server	游戏服务器
Game Client	游戏客户端
TBUS	BUS 上报接口

2. TDIR 接口

2.1. 系统拓扑图





对 Game Client 来说不区分 TDIR 的分类,只保留 TDIR 的地址列表。

对 Game Server 来说,只需要知道主目录服务器的地址和备用目录服务器的地址,不需要知道其它的目录服务器地址,Game Server 需要定时向主目录服务器和备用目录服务器上
报服务器状态信息。

在游戏测试阶段,若没有配置备用目录服务器,Game Server 可以认为主目录服务器同时又是备用目录服务器。

2.2. TDIR 接口定义

Game Server 需要定时向主目录服务器和备用目录服务器上
报服务器状态信息。上报时间默认间隔为 5 秒。

2.2.1. Tbus 上报接口

TDIR 接口库文件需要依赖 TDR 库,支持自定义上报数据,适用自研游戏。

TDIR 接口包括三个函数。

接口原型	int tdirapi_init(IN const char * psztdrpath);
参数说明	psztdrpath:TDIR 元数据文件路径
说明	该接口用来加载元数据文件,初始化元数据信息
返回值	< 0 - 失败 0 - 成功
备注	

接口原型	int tdirapi_build_ReportMsg(IN int iGamesvrid, IN GSBASE *pstBaseData, IN GSAPP *pstAppData, INOUT char *pszMsgBuff, INOUT int *iMsgBufflen)
参数说明	iGamesvrid:Game Server ID,Game Server ID/必须跟 tdir 配置文件定义 节点 ID 一致

	<p>pstBaseData:Game Server 上报数据</p> <p>pstAppData:应用自定义数据</p> <p>pszMsgBuff:IN:消息缓存区</p> <p>OUT:生成上报消息的缓存区</p> <p>iMsgBufflen:IN:消息缓存区长度</p> <p>OUT:生成上报消息长度</p>
说明	该接口用来生成服务器状态上报消息
返回值	< 0 - 失败 0 - 成功
备注	<p>引用的数据结构:</p> <pre>typedef struct tagGSBASE GSBASE; struct tagGSBASE { int iMaxOnline; int iCurrentOnline; char szConnectUrl[MAX_CON_URL_LEN]; };</pre> <p>iMaxOnline:Game Server 容纳最大人数</p> <p>iCurrentOnline:Game Server 当前在线人数</p> <p>szConnectUrl:Game Server 连接 URL,格式为 IP 地址:端口。 Eg:"172.19.25.225:9001"</p> <pre>struct tagGSAPP; typedef struct tagGSAPP GSAPP; struct tagGSAPP { int iAttr; };</pre> <p>iAttr:应用自定义数据.目前暂未用.设为零。</p>
接口原型	<pre>void tdirapi_free();</pre>
参数说明	
说明	该接口用来释放元数据文件
返回值	
备注	

2.2.2. tbus 接口使用示例

TDIR 接口头文件参考 tdir_api.h,库文件使用 libtdirapi.a

//伪代码

//initaion

int init(...)

{

 char *psztdrpath="../cfg/tdir_def_level3.tdr"

 iRet =tdirapi_init(psztdrpath);

 if(iRet<0){ ... }

}

int tdir_Send_ReportMsg(...)

{

 int iGameSvrID=(int *)inet_addr("1.2.1.1");

 GSBASE stBase;

 GSAPP stApp;

 char szBuff[REPORT_MSG_BUFFER]={0};

 int nlen=sizeof(szBuff);

 int iRet=0;

 stBase.iMaxOnline=3000;

 stBase.iCurrentOnline=500;

 stBase.szConnectUrl="172.19.25.225:9001" ;

 memset(&stApp,sizeof(stApp),0);

 iRet= tdirapi_build_ReportMsg(iGameSvrID,&stBase,&stApp,szBuff,&nlen);

 if(iRet<0){ ...;return -1; }

 //send to master dir server

 tbus_send(ibus,masterdir,szBuff,nlen,...);

 ...

 //Send to bakup dir server

 tbus_send(ibus,bukupdir,zBuff,nlen);

```
...  
}  
  
//finalize  
  
int init( ... )  
{  
  
    tdirapi_free();  
  
    ...  
  
}
```

2.2.3. udp 上报接口

使用默认上报数据,接口不支持自定义上报数据,适用于代理游戏。TDIR 接口头文件参考 tdir_api_s.h,库文件使用 libtdirapi_s.a

接口原型	<pre>int tdirapi_S_Build_ReportMsg(IN int iGamesvrid, IN GSBASE *pstBaseData, IN GSAPP *pstAppData, INOUT char *pszMsgBuff, INOUT int *iMsgBufflen)</pre>
参数说明	<p>iGamesvrid:Game Server ID/必须跟 tdir 配置文件定义节点 ID 一致</p> <p>pstBaseData:Game Server 上报数据</p> <p>pstAppData:应用自定义数据</p> <p>pszMsgBuff:IN:消息缓存区 OUT:生成上报消息的缓存区</p> <p>iMsgBufflen:IN:消息缓存区长度 OUT:生成上报消息长度</p>
功能说明	该接口用来生成服务器状态上报消息
返回值	<p>< 0 - 失败</p> <p>0 - 成功</p>
备注	<p>引用的数据结构:</p> <pre>typedef struct tagGSBASE GSBASE; struct tagGSBASE { int iMaxOnline; int iCurrentOnline; char szConnectUrl[MAX_CON_URL_LEN]; };</pre>

iMaxOnline:Game Server 容纳最大人数
iCurrentOnline:Game Server 当前在线人数
szConnectUrl:Game Server 连接 URL,格式为 IP 地址:端口。
Eg:"172.19.25.225:9001"

```
struct tagGSAPP;  
typedef struct tagGSAPP    GSAPP;  
struct tagGSAPP  
{  
    int iAttr;  
};  
iAttr:应用自定义数据.目前暂未用.设为零。
```

2.2.4. udp 接口使用示例

TDIR 接口头文件参考 tdir_api_s.h

//伪代码

```
int tdir_Send_ReportMsg( ...)  
{  
    int iGameSvrID=(int *)inet_addr("1.2.1.1");  
    GSBASE stBase;  
    GSAPP    stApp;  
    char szBuff[REPORT_MSG_BUFFER]={0};  
    int nlen=sizeof(szBuff);  
    int iRet=0;  
    stBase.iMaxOnline=3000;  
    stBase.iCurrentOnline=500;  
    stBase.szConnectUrl="172.19.25.225:9001";  
    memset(&stApp,sizeof(stApp),0);  
    iRet=tdirapi_S_Build_ReportMsg(iGameSvrID,&stBase,&stApp,szBuff,&nlen);  
    if(iRet<0){    ...;return -1; }  
    //send to master dir server  
    sendto(s1,szBuff,nlen,...);
```

```
//Send to bakup dir server  
  
sendto(s2,szBuff,nlen);  
  
}
```