**恒生资产管理平台统一接入**

UFX1.1开发指南

**恒生电子股份公司**



修订历史记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本/状态** | **作者** | **参与者** | **日期** | **备注** |
| V1.0 | 吴程宏 |  | 2014.5.19 | 初稿 |

# 版本说明

V1.0 初稿

[版本说明 2](#_Toc388301904)

[文档说明(必读) 3](#_Toc388301905)

[文档阅读方法（初学者必读） 3](#_Toc388301906)

[资料获取方式 4](#_Toc388301907)

[支持及交流QQ群 4](#_Toc388301908)

[**一、** **账户结构** 4](#_Toc388301909)

[**二、** **接口传入格式** 4](#_Toc388301910)

[**三、** **接口传出格式（必读）** 6](#_Toc388301911)

[**四、** **业务基本操作流程** 8](#_Toc388301912)

[**五、** **同步/异步调用** 8](#_Toc388301913)

[**六、** **成交、委托消息订阅/推送** 8](#_Toc388301914)

[**七、** **常见问题** 9](#_Toc388301915)

[**1.** **调用10001登录接口或其他接口，t2sdk函数返回值不为0，GetErrorMsg函数返回“未知错误”** 10](#_Toc388301916)

[**2.** **调用10001登录接口，取不到user\_token（取到的user\_token为空）** 10](#_Toc388301917)

[**3.** **接口支持多个记录传入，前台该如何打包？** 10](#_Toc388301918)

[**4.** **如何同时订阅多种消息类型、多种过滤条件？** 10](#_Toc388301919)

[**5.** **订阅后，收不到系统推送的委托、成交消息** 10](#_Toc388301920)

# 文档说明(必读)

UFX1.1开发分为两层：

1. UFX业务操作层，主要进行业务字段的定义、操作，比如如何下委托、委托业务包体如何按照规定字段打包等。
2. T2SDK数据传输层，主要为底层的数据的传输，包含t2sdk.dll/t2sdk.so导出函数的使用说明等。

本文档主要针对第一点，UFX业务操作进行开发指导；t2sdk.dll/t2sdk.so导出函数的使用说明可参考《T2SDK 外部版开发指南.docx》，文档位置位于t2sdk开发包中的 dev\client\_demo\T2SDK\_External\c++\doc\路径中。

# 文档阅读方法（初学者必读）

1. 打开T2SDK开发包，阅读里面的demo，学习t2sdk的使用，遇到不了解的t2sdk函数，参考《T2SDK 外部版开发指南.docx》
2. 阅读随发布包发布的《恒生资产管理平台统一接入\_接口规范》，了解基本功能点及业务包体结构。
3. 阅读本文档中的**第一章至第四章**，了解UFX包体结构、常见问题及注意事项。
4. 打开UFX1.1的专用Demo，从main函数开始阅读，遇到接口字段问题参考随发布包发布的《恒生资产管理平台统一接入\_接口规范》，遇到t2sdk函数问题，参考头文件或《T2SDK 外部版开发指南.docx》

# 资料获取方式

联系对应的维护专员获取，资料包含三个

1. 随发布包发布的《恒生资产管理平台统一接入\_接口规范》 文档中定义每个接口的功能号、输入输出字段定义
2. T2SDK开发包，包含t2sdk.dll/t2sdk.so链接库及头文件
3. T2连接的License文件，用于连接UFX时，进行的T2连接合法性校验

# 支持及交流QQ群

恒生资管UFX客户群 QQ群：336502532（UFX业务操作交流群）

HS大金融业务开发者群 QQ群：261969915（T2SDK使用、导出函数问题等交流群）

1. **账户结构**

账户

资产单元A

资产单元B

资产单元C

 组合A

 组合B

 组合C

资管账户结构分为三层，如图所示。

账户层：对应接口中account\_code账户编号，在组织结构中对应部门或基金

资产单元层：对应接口中asset\_no资产单元编号，资产单元上维护其使用的资金帐号

组合层：对应接口中combi\_no组合编号，证券持仓所在层次，按实际的业务目的划分

1. **接口传入格式**

请求数据以IF2Packer包方式传入，

传入数据结构：

字段名： 字段1|字段2|…|字段N

数据记录1：数据1|数据2|…|数据N

数据记录2：数据1|数据2|…|数据N

…

数据记录N：数据1|数据2|…|数据N

注意：

1. 字段名和数据根据接口说明确定。
2. 接口默认只支持单记录，部分接口支持多记录，是否支持多条记录传入，见接口业务说明。
3. 多记录打包方式参见下文例子

lpPacker->BeginPack();

//打包字段名

lpPacker->AddField("user\_token");

lpPacker->AddField("account\_code");

lpPacker->AddField("combi\_no");

lpPacker->AddField("batch\_no");

lpPacker->AddField("instance\_no");

lpPacker->AddField("max\_cancel\_ratio");

lpPacker->AddField("market\_no");

lpPacker->AddField("stock\_code");

lpPacker->AddField("entrust\_direction");

lpPacker->AddField("futures\_direction");

lpPacker->AddField("price\_type");

lpPacker->AddField("entrust\_price");

lpPacker->AddField("entrust\_amount");

lpPacker->AddField("stockholder\_id");

lpPacker->AddField("report\_seat");

lpPacker->AddField("invest\_type");

lpPacker->AddField("extsystem\_id");

lpPacker->AddField("third\_reff");

//循环打包字段值

for(vector<CUFXEntrustStock \*>::iterator it = m\_in\_stock\_list.begin(); it != m\_in\_stock\_list.end();it++)

{

lpPacker->AddStr(m\_in\_user\_token.c\_str());

lpPacker->AddStr(m\_in\_account\_code.c\_str());

lpPacker->AddStr(m\_in\_combi\_no.c\_str());

lpPacker->AddInt(m\_in\_batch\_no);

lpPacker->AddStr(m\_instance\_no.c\_str());

lpPacker->AddDouble(m\_in\_max\_cancel\_ratio);

lpPacker->AddStr((\*it)->m\_market\_no.c\_str());

lpPacker->AddStr((\*it)->m\_stock\_code.c\_str());

lpPacker->AddStr((\*it)->m\_entrust\_direction.c\_str());

lpPacker->AddStr((\*it)->m\_futures\_direction.c\_str());

lpPacker->AddStr((\*it)->m\_price\_type.c\_str());

lpPacker->AddDouble((\*it)->m\_entrust\_price);

lpPacker->AddInt((\*it)->m\_entrust\_amount);

lpPacker->AddStr((\*it)->m\_stockholder\_id.c\_str());

lpPacker->AddStr((\*it)->m\_report\_seat.c\_str());

lpPacker->AddStr((\*it)->m\_invest\_type.c\_str());

lpPacker->AddInt((\*it)->m\_extsystem\_id);

lpPacker->AddStr((\*it)->m\_third\_reff.c\_str());

}

lpPacker->EndPack();

1. **接口传出格式（必读）**

返回数据以IF2Packer包方式传出，返回包分两块数据区：包头和数据区（也成为第一个结果集和第二个结果集），包头包含错误信息，数据区包含真正的返回数据。

包头结构：

字段名： ErrorCode|ErrorMsg|MsgDetail|DataCount

字段含义： 错误号|错误信息|详细信息|返回记录数

包体结构：

字段名： 字段1|字段2|…|字段N

数据记录1：数据1|数据2|…|数据N

数据记录2：数据1|数据2|…|数据N

…

数据记录N：数据1|数据2|…|数据N

注意：

1. 返回包第一个包表示调用业务操作是否成功的标识（通过SetCurrentDatasetByIndex(0)取第一个结果集数据）
2. 返回包的第二个结果集才是真正的业务数据（通过SetCurrentDatasetByIndex(1)取第二个结果集数据）

通用解包步骤参考

1. 解返回包时，首先判断T2SDK函数返回值是否正常（一般为0表示正常），否则通过对应类函数的GetErrorMsg获取对应的错误信息，如下

int iRcv = m\_pConnection->RecvBiz(hSend,&m\_pUnpacker,m\_pUFXParam->GetTimeOut());

if (iRcv != 0)

{

printf("RecvBiz错误:%s\n",m\_pConnection->GetErrorMsg(iRcv));

return iRcv;

}

1. 解第一个结果集，判断业务操作是否正常调用

lpUnpacker->SetCurrentDatasetByIndex(0);

m\_out\_ErrorCode = lpUnpacker->GetInt("ErrorCode");

m\_out\_ErrorMsg = charToStr(lpUnpacker->GetStr("ErrorMsg"));

m\_out\_MsgDetail = charToStr(lpUnpacker->GetStr("MsgDetail"));

m\_out\_DataCount = lpUnpacker->GetInt("DataCount");

if(m\_out\_ErrorCode != 0)

{

printf("ErrorCode=:%n\n", m\_out\_ErrorCode);

printf("ErrorMsg=%s\n", m\_out\_ErrorMsg);

printf("MsgDetail=%s\n", m\_out\_MsgDetail);

}

1. 移动到第二个结果集，处理真正的业务返回数据

int CUFXUserLogon::UpackResult(IF2UnPacker \*lpUnpacker)

{

if (NULL == lpUnpacker) return RET\_ERR;

int iErrorCode = UpackHeader(lpUnpacker);

if (iErrorCode != 0) return iErrorCode;

if (lpUnpacker->GetDatasetCount() > 1)

lpUnpacker->SetCurrentDatasetByIndex(1);

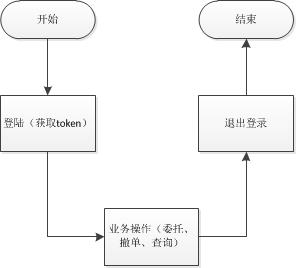
lpUnpacker->first();

m\_out\_user\_token = charToStr(lpUnpacker->GetStr("user\_token"));

return RET\_OK;

}

1. **业务基本操作流程**



1. 根据接口文档打包10001登录接口输入参数的字段描述（打包方式参考【[接口传入格式](#接口传入格式)】）
2. 调用10001登录接口
3. 根据10001登录接口输出参数的字段描述，取返回的业务字段，如user\_token字段（解包方式参考【[接口传出格式](#接口传出格式)】）
4. 根据接口文档打包91001深沪委托接口输入参数的字段描述（打包方式参考【[接口传入格式](#接口传入格式)】）
5. 调用91001委托接口
6. 根据91001委托接口输出参数的字段描述，取返回的业务字段（解包方式参考【[接口传出格式](#接口传出格式)】）
7. 根据接口文档打包10002退出登录接口输入参数的字段描述（打包方式参考【[接口传入格式](#接口传入格式)】）
8. 调用10002退出登录接口
9. 根据10002退出登录接口输出参数的字段描述，判断退出登录是否成功（解包方式参考【[接口传出格式](#接口传出格式)】）

从上可见，每一次调用均分为打包、调用、解包三个步骤；打包、解包均需要根据不同的接口说明，打包/解包不同的字段值，具体实现代码，参考Demo文件。

1. **同步/异步调用**

参考《T2SDK 外部版开发指南.docx》中【开发流程】、【示例代码】

1. **成交、委托消息订阅/推送**

UFX1.1，订阅主题固定为ufx\_topic，订阅关键字/过滤关键字为三个

operator\_no：操作员（必传，UFX目前限定只能订阅本次登录用户消息）

msgtype：消息类型（可选，值域见随发布包发布的《恒生资产管理平台统一接入\_接口规范》）

market\_no：市场代码（可选，值域见随发布包发布的《恒生资产管理平台统一接入\_接口规范》）

注意：

订阅时，需要用户设定登录操作员及密码，用于登录权限的校验。其中登录操作员login\_operator\_no必须与过滤关键字operator\_no一样，限定本次只能当前用户做的委托、成交信息，示例代码如下，详细代码请见对应的Demo。

//创建一个订阅登录校验包

IF2Packer\* pack = NewPacker(2);

pack->AddRef();

pack->BeginPack();

pack->AddField("login\_operator\_no");

pack->AddField("password");

pack->AddStr("10000");

pack->AddStr("12");

pack->EndPack();

//订阅

**int iRet = lpSub->SubscribeTopic(lpSubscribeParam,5000,&lpBack,pack);**

if(iRet>0)

{

subscribeIndex = iRet;

printf("SubscribeTopic info:[%d] 成功\n",iRet);

//保存到map中，用于以后的取消订阅

g\_allSubscribeParam[subscribeIndex] = lpSubscribeParam;

}

else

{

if(lpBack != NULL)

PrintUnPack(lpBack);

printf("SubscribeTopic info:[%d] %s\n",iRet,g\_lpConnection->GetErrorMsg(iRet));

return-1;

}

订阅时，若针对重要数据，需要开启补缺功能的（补缺功能参考T2SDK外部版开发指南SetFromNow函数），要在UFX的配置文件topics.xml中配置可靠级别rely\_level在3级或以上，life\_time不为0. 代价是性能会有所影响。

//订阅参数获取

CSubscribeParamInterface\* lpSubscribeParam = NewSubscribeParam();

lpSubscribeParam->AddRef();

char\* topicName = (char\*)lpConfig->GetString("subcribe","topic\_name","");//主题名字

lpSubscribeParam->SetTopicName(topicName);

char\* isFromNow = (char\*)lpConfig->GetString("subcribe","is\_rebulid","");//是否补缺

if (strcmp(isFromNow,"true")==0)

{

lpSubscribeParam->SetFromNow(true);

}

else

{

lpSubscribeParam->SetFromNow(false);

}

1. **常见问题**
2. **调用10001登录接口或其他接口，t2sdk函数返回值不为0，GetErrorMsg函数返回“未知错误”**

解决：通常是UFX服务端未启动或对应的UFX端口被其他进程占用导致，联系维护专员解决。

1. **调用10001登录接口，取不到user\_token（取到的user\_token为空）**

解决：UFX1.1返回包区分两个结果集，需要使用SetCurrentDatasetByIndex(1)函数移动到第二个结果集中再获取user\_token，具体操作请参考【[接口传出格式](#接口传出格式)】）

1. **接口支持多个记录传入，前台该如何打包？**

解决：打包方式参考【[接口传入格式](#接口传入格式)】

1. **经常提示user\_token错误**

解决：UFX1.1中默认限定服务端user\_token有效期为5分钟，用户必须维护业务层心跳，5分钟内调用一次以保持user\_token的有效性（一般设置为3分钟调用一次心跳）。业务层心跳接口为10001，调用方式与普通接口调用方式一样。

1. **如何同时订阅多种消息类型、多种过滤条件？**
2. 通过设定不同的订阅参数CSubscribeParamInterface\* lpSubscribeParam，调用SubscribeTopic多次
3. 订阅时，对应的过滤条件不设置，前台回调函数接收到消息后，再自行根据接口文档定义的字段解包后判断
4. **订阅后，收不到系统推送的委托、成交消息**
5. 排查是否因为设置了过滤关键字，导致某个类型消息被过滤
6. 与维护专员联系，确认O3资管消息服务器是否开启，推送系统参数60369,60066是否开启