

# 统一金融接入（UFX）业务说明

## 证券普通交易



二〇一四年八月

说明：本文档仅供参考，实际情况可能会因使用系统版本不同而存在差异，具体需联系恒生电子客服总部（0571-28829999）进行咨询！

本文所述内容（包括文字和图片），恒生电子股份有限公司（以下简称“恒生”或“恒生公司”）拥有完全独立的唯一版权。未经恒生公司书面同意或授权，任何单位和个人都不得将其复制、影印或引用。

## 文档修改记录

版本号	日期	说明	编写者	审核者
V8.0.8.0	2014-08-26	创建	魏文杰	
V8.0.8.1	2014-09-03	调整文档目录层次	魏文杰	李玲斌
V8.0.8.2	2014-09-08	1) 参考接入指南编写 2) 调整文档中部分描述问题和措辞	魏文杰	李玲斌

## 目录

文档修改记录 .....	- 2 -
<b>1 前言 .....</b>	<b>- 4 -</b>
1.1 关于手册 .....	- 4 -
1.2 名词定义 .....	- 4 -
<b>2 连接准备 .....</b>	<b>- 5 -</b>
<b>3 业务流程 .....</b>	<b>- 6 -</b>
3.1 普通委托 .....	- 7 -
333002-普通委托 .....	- 7 -
3.2 撤单 .....	- 8 -
333017-委托撤单 .....	- 8 -
3.3 委托查询 .....	- 9 -
333101-证券委托查询 .....	- 9 -
339303-历史证券委托查询 .....	- 9 -
3.4 成交查询 .....	- 9 -
333102-证券成交查询 .....	- 9 -
339304-历史证券成交查询 .....	- 9 -
3.5 持仓查询 .....	- 9 -
333103-证券持仓快速查询 .....	- 10 -
333104-证券持仓查询 .....	- 10 -
3.6 回报 .....	- 11 -
委托回报 .....	- 11 -
成交回报 .....	- 11 -
3.7 其他 .....	- 12 -
银证转账 .....	- 12 -
交割信息 .....	- 13 -
<b>4 开发特别说明 .....</b>	<b>- 13 -</b>
打包 .....	- 13 -
心跳维护 .....	- 14 -
断开重连 .....	- 15 -
分页查询 .....	- 15 -

# 1 前言

## 1.1 关于手册

本手册作为恒生 T2SDK 和恒生周边接口规范的辅助手册，主要有以下三个方面：

- 1) 如何接入统一金融接入平台（UFx）。
- 2) 证券交易基本业务流程、相关功能号、字段含义解释等。
- 3) 常见开发问题说明。

预期读者：1、接入恒生证券交易系统第三方投资者。

技能要求：1、Delphi/C/C++/C#/java 基础编程知识。  
2、了解基本证券交易知识。

## 1.2 名词定义

<b>统一金融接入平台（UFx）</b>	是交易客户端与后台交易系统交互的通道，是一个安全又开放的金融接入平台，保证交易系统的安全性和稳定性；同时可提供 T2、FIX、Web Service 等方式供第三方安全接入。【本文讲述 T2 接入】
<b>T2_SDK</b>	T2_SDK 是开发者接入统一接入的客户端开发包。T2_SDK 以 dll(Windows 下) 或 so (Linux 下) 的形式提供给开发者，通过使用 T2_SDK 开发程序可以轻松实现和 UFx 的对接。【T2SDK 只能 T2 接入】
<b>T2 接入</b>	T2 即恒生第二代通讯协议，消息结构灵活扩展性强方便使用，对外提供 t2sdk.dll 开发包
<b>FIX 接入</b>	采用标准 FIX 协议接入统一金融接入平台
<b>Web Service 接入</b>	提供标准的 Web Service 服务，提供 JSON 轻量级数据交换格式供第三方接入。
<b>消息中心</b>	可以订阅发布消息。
<b>订阅</b>	消息的接收者，不同的消息对应不同的消息类型。

**注：本文档仅描述 T2 方式接入，如想要其它方式接入，请联系恒生客服。**

## 2 连接准备

- 1) 创建配置对象(CConfigInterface::NewConfig)，设置 t2sdk 配置

```
[servers] // 服务器地址  
[license_file] // 许可证文件  
[login_name] // 有名客户端的名字
```

更多 t2sdk 配置说明请详见《T2SDK 外部版开发指南.docx》

创建连接对象(CConnectionInterface::NewConnection)

将配置对象指针传入 NewConnection

- 2) 初始连接对象(CConnectionInterface::Create/CreateEx)

如果不需要任何回调方法，则可向 Create/CreateEx 传递 NULL，而不必自定义回调类和对象

- 3) 开始连接/注册(CConnectionInterface::Connect)

注意：

1. 错误信息 = -11；请先 telnet 对应服务地址，是否可以打开对应端口
2. 错误信息 = -20；连接超时
3. 错误信息 = -26；请联系证券公司，是 t2 的通讯证书不对

更多 t2sdk 错误码请详见《T2SDK 外部版开发指南.docx》

- 4) 发送登录请求

发送 331100 登录请求包

- 5) 发送订阅请求

发送 620001 订阅请求包，对应发布类型如下

12: 成交回报

23: 委托回报

24: 基本行情(部分开发商支持、详细咨询开发商/恒生维护)

25: 实时行情(部分开发商支持、详细咨询开发商/恒生维护)

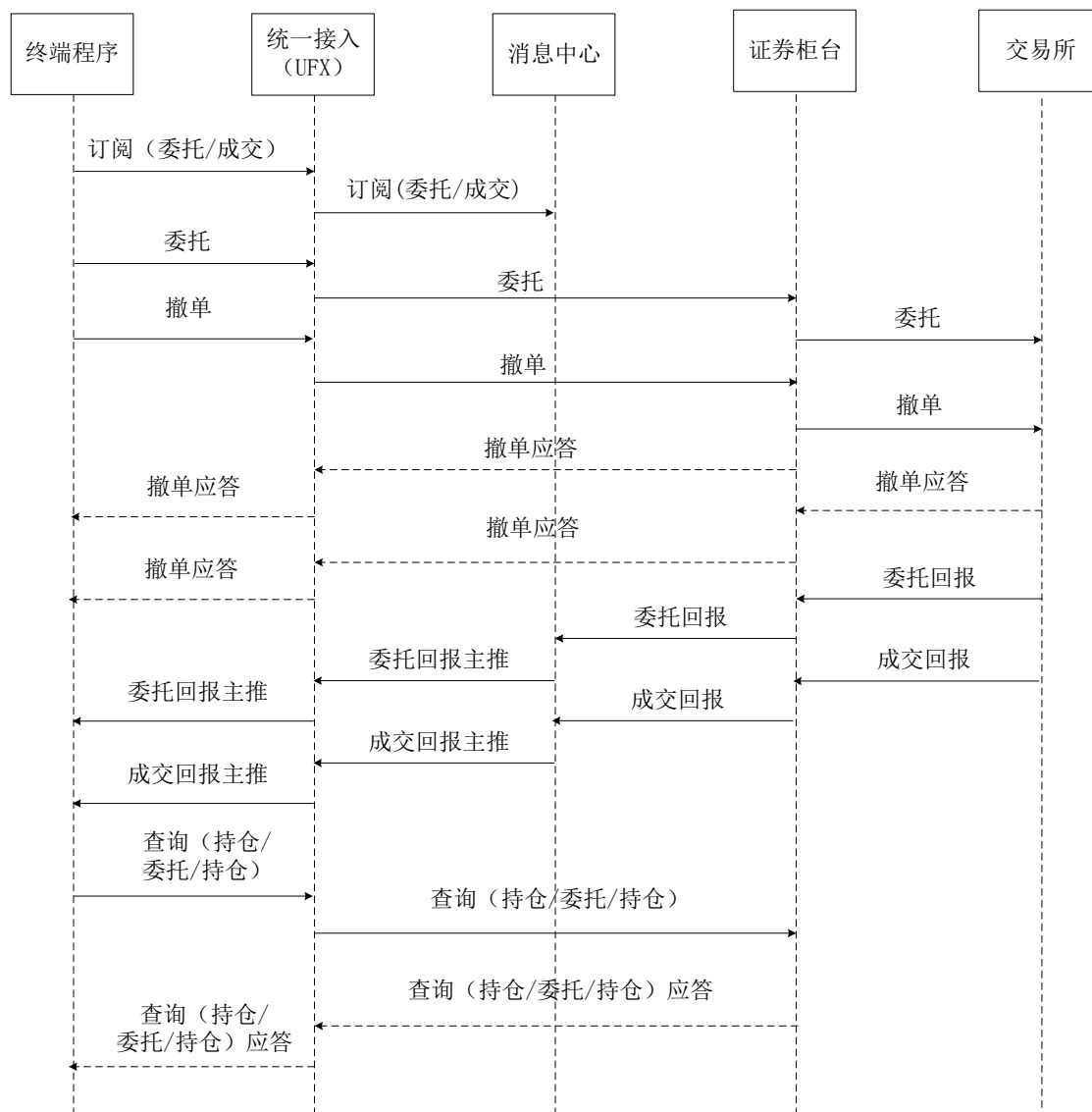
注意：

1. 成功订阅之后，需要维护 620000 心跳（消息中心 1.0）。详见[心跳维护范例]

- 6) 正常做业务操作

### 3 业务流程

证券普通业务包括普通委托、委托撤单、查询（委托、成交、持仓），以及订阅、主推等功能。基本业务流程如下图所示：



注：撤单只有委托状态为“未报” (entrust\_status='0')状态下是在柜台系统中撤单，其它可以撤单状态是柜台系统将请求发给交易所，由交易所撤单。

## 3.1 普通委托

1. 终端程序发送报单请求给柜台，柜台判断合法性之后，柜台将报单转发到交易所。
2. 柜台收到交易所成交，会写给消息中心，消息中心将实时通知已订阅的终端程序。

### 333002-普通委托

普通委托是投资者通过柜台向交易所下单，由柜台负责申报、交易所负责撮合成交的行为，买卖股票均为此接口。

- 必须输入参数

///**委托方式**（恒生有定义委托方式，实际每个券商可能定义的不同）

op\_entrust\_way = '7'

///**营业部号**（此处填所在营业部编号）

branch\_no = 1000

///**密码**（此处填交易密码）

password = "111111"

///**客户号**（此处填写客户号）

client\_id = "10000001"

///**资金账号**（此处填写资金账号）

fund\_account = "100000001"

///**证券账号**（此处填写证券账号）

stock\_account = "A100000001"

///**交易类别**：（

1:上海 A 股

D:上海 B 股

2:深圳 A 股

H:深圳 B 股

9:特转 A(三板)

A:特转 B(三板)

)

exchange\_type = "1"

///**证券代码**

stock\_code = "600570"

///**买卖方向**：1（

1:买入；

2:卖出;)

entrust\_bs = '1'

///**委托数量**：100

entrust\_amount = 100

///**委托价格**：30

entrust\_price = 30

- 非必须输入参数

///操作分支结构: op\_branch\_no 是营业部接入的网点, 这样可以知道客户如果在别的营业部接入的时候, 接入点是哪儿; branch\_no 是客户所在营业部。

op\_branch\_no=1000

///密码类别: (

1: 资金密码

2: 交易密码

) 系统默认为' 2'

password\_type = '2'

///用户口令: 登录时返回, 为提高后台系统效率而设计, password+password\_type 需要去后台校验密码; 如送入 user\_token+password+password\_type 则不需要去后台校验, 在入参 user\_token 中校验 password 是否正确, 减少了与后台一次交互。注: 在客户端接入统一接入 (UFX) 的时候, user\_token 会有时间限制, 过期会让重新登录。具体时间由券商设置, 如不想过期, 可在即将过期时做业务操作、如查询持仓等。(此处填写登录返回的 user\_token 的值)

user\_token=" ADE31CB348C04FE5980FA04A0CAEA0F3D59F256E0BB64029B894CC13499E1B26"

///委托批号: 方便查询以及撤单, 如果下两笔委托, 委托批号一样, 则表示这两笔委托为同一批次。查询/撤单时可以查询/撤销同一批次的委托。

batch\_no=100

## 3.2 撤单

### 333017-委托撤单

委托撤单是投资者通过柜台下一笔委托之后, 不想成交这笔委托, 想要撤销该委托请求的行为。委托撤单如果柜台还未申报给交易所, 则撤单行为在柜台进行; 如果是柜台已经报单给交易所, 撤单行为在交易所进行。

- 委托状态 (entrust\_status) 与是否可撤单说明:

值	说明	是否可撤
0	未报	可以
1	待报	不可以
2	已报	可以
3	已报待撤	不可以
4	部成待撤	不可以
5	部撤	不可以
6	已撤	不可以
7	部成	可以
8	已成	不可以

9	废单	不可以
W	待确认	不可以

### 3.3 委托查询

委托查询功能是查询当日向柜台所下的所有委托，包含所有的委托状态。

#### 333101-证券委托查询

- 与 333100(可撤单委托查询)区别：
  - 333100(可撤单委托查询)仅查询以下几种委托状态的委托：0:未报 1:待报 2:已报 3:已报待撤 4:部成待撤 7:部成 W:待确认
  - 333101(证券委托查询)查询全部委托状态的委托

#### 339303-历史证券委托查询

- 与 333101（证券委托查询）相比多增加了按照日期查询，可以根据选择的起始日期查询，详细说明（略）。

### 3.4 成交查询

成交查询是查询当日已成交委托。

#### 333102-证券成交查询

#### 339304-历史证券成交查询

- 与 333102（证券成交查询）相比多增加了按照日期查询，可以根据选择的起始日期查询。

### 3.5 持仓查询

持仓查询为查询投资者当前账户证券持仓数量。

## 333103-证券持仓快速查询

相关输出参数说明:

- hold\_amount; ///持有数量, 即当前持有证券数量
- current\_amount; ///当前数量, 清算或红冲蓝补后才会改变; 当天买入股票数量, 当前数量不变(后台有开关控制, 也可以实际显示改变, 但实际数据库的值不会改变)
- cost\_price; ///成本价, 不精确、返回费率属性供周边客户端计算
- imcome\_balance; ///盈亏金额, 即收益金额
- frozen\_amount; ///冻结数量, 当前被冻结, 不能用于买卖的股票数量
- unfrozen\_amount; ///解冻数量, 从冻结到解冻的数量
- enable\_amount; ///可用数量, 当前可用数量
- real\_buy\_amount; ///回报买入数量, 委托买入成交后的数量
- real\_sell\_amount; ///回报卖出数量, 委托卖出成交后的数量
- real\_buy\_balance; ///回报买入金额, 委托买入成交后的金额
- real\_sell\_balance; ///回报卖出金额, 委托卖出成交后的金额
- uncome\_buy\_amount; ///未回买入数量, 委托买入申报到交易所, 交易所还未回报的金额
- uncome\_sell\_amount; ///未回卖出数量, 委托卖出申报到交易所, 交易所还未回报的金额
- entrust\_sell\_amount; ///委托卖出数量, 委托卖出股票数量
- asset\_price; ///市值价, 用于计算证券持仓市值, 一般股票市值价=行情最新价
- last\_price; ///最新价, 该股票的行情最新价格
- market\_value; ///市值, 市值=(当前数量+修正数量+回报买入数量-回报卖出数量)×行情市值价
- sum\_buy\_amount; ///累计买入数量, 累计买入一支股票的数量
- sum\_buy\_balance; ///累计买入金额, 累计买入一支股票金额
- sum\_sell\_amount; ///累计卖出数量, 累计卖出一支股票的数量
- sum\_sell\_balance; ///累计卖出金额, 累计卖出一支股票的金额

## 333104-证券持仓查询

333103(证券持仓快速查询)精确计算成本, 而是将费用属性同步返回, 让周边计算成本。333104(证券持仓查询)是后台精确计算成本, 参数由柜台设置, 根据不同的设置成本计算方式不同。

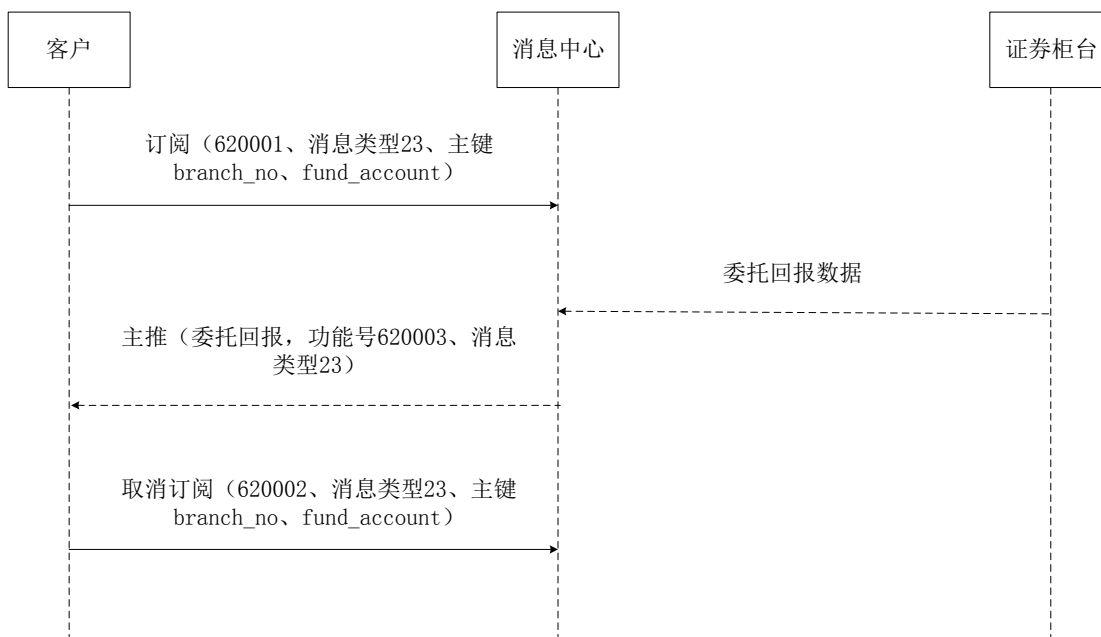
详细算法参照文档[《经纪业务运营平台 V20 维护手册\(算法篇\).pdf》](#)

## 3.6 回报

回报简单的理解是：以前自己买报纸，现在邮递员每天自动送报纸。

### 委托回报

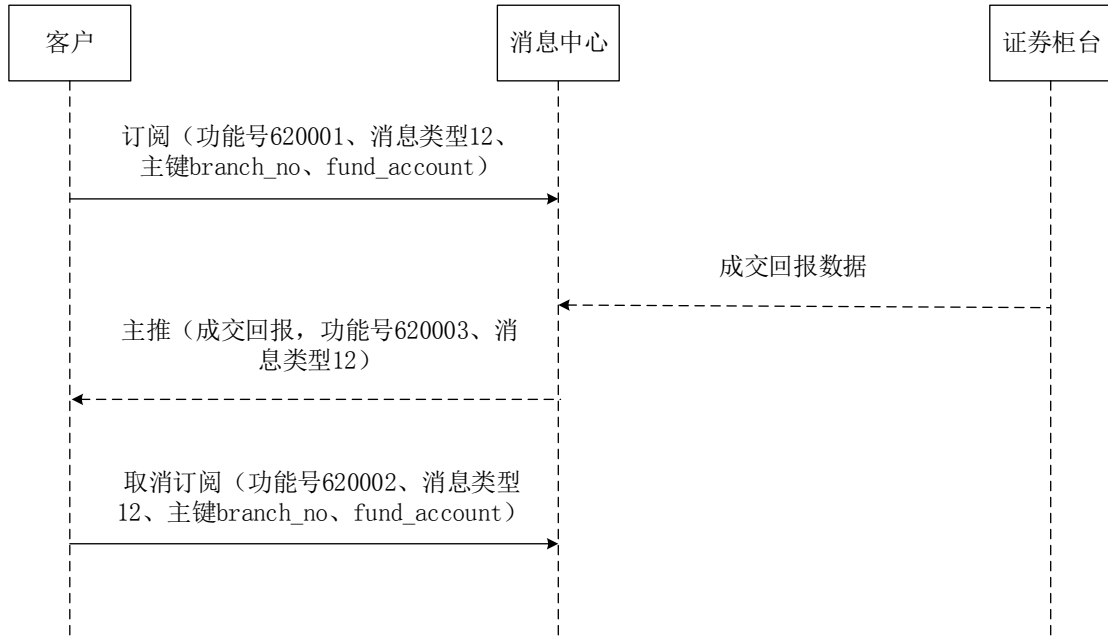
委托回报是客户在委托之后，无需主动调用功能号去查询，只需订阅消息类型为 23 消息功能；当消息中心收到柜台推送的委托信息后，由消息中心推送给订阅的客户；如客户不需要订阅的消息类型，则取消订阅的消息类型即可。



- 订阅  
功能号 620001，消息类型 23，主键 branch\_no, fund\_account (消息)
- 取消订阅  
功能号 620002，消息类型 23，主键 branch\_no, fund\_account (消息)
- 主推  
消息中心自动主推，主推字段详见接口文档。

### 成交回报

成交回报是客户在委托成交之后，无需主动调用功能号去查询，只需订阅消息类型为 12 消息功能；当消息中心收到柜台推送的成交信息后，由消息中心推送给订阅的客户；如客户不需要订阅的消息类型，取消订阅的消息类型即可。



- 订阅  
功能号 620001, 消息类型 12, 主键 branch\_no, fund\_account (消息)
- 取消订阅  
功能号 620002, 消息类型 12, 主键 branch\_no, fund\_account (消息)
- 主推  
部成/已成- entrust\_status=' 7' /' 8'  
当报单在交易所有成交时, 后台收到交易所应答会推送委托状态为部成/已成的委托信息给订阅了客户。

## 3.7 其他

### 银证转账

- 332200-银证转账  
在证券交易活动中, 投资者 (即客户) 的交易结算资金是由证券公司一家统一存管的。后来, 证监会规定, 客户的交易结算资金统一交由第三方存管机构存管 (具备第三方存管资格的商业银行), 即”三方存管”。由存管银行按照法律、法规的要求, 负责客户资金的存取与资金交收, 证券交易操作保持不变。
  - transfer\_direction; ///交易方向 ('1' - 银行转证券 '2' - 证券转银行)
  - occur\_balance; ///发生金额 (“即转账金额”)
  - money\_type; ///币种类别 (常用 0:人民币 1:美元 2:港币)
  - bank\_no; ///银行代码

## 交割信息

### ● 339300-历史交割信息查询

- post\_balance; ///后资金额, 做完该笔交易后的可用金额。
- post\_amount; ///后证券额, 做完该笔交易后的可用数量。
- fare0; ///佣金, (买卖双向收取, 具体比例可跟券商谈, 不足五元按五元收取)
- fare1; ///印花税, (国家征收, 券商代收, 股票卖出时才征收, 目前是千分之一)
- fare2; ///过户费, (证券登记清算机构收入, 券商代收, 按成交股票数量\*万分之六收取。买卖股票都收, 上交所收, 深交所不收)
- fare3; ///费用 3, 委托费用, 券商自行征收。
- farex; ///费用 X, 券商自行征收费用。

## 4 开发特别说明

### 打包

#### 1) 打包顺序→先打头, 再打值

正确 ✓

```
lpPacker->AddField("branch_no");  
lpPacker->AddField("fund_account", 'S');  
lpPacker->AddInt(1000);  
lpPacker->AddStr("100000001");
```

错误 ✗

```
lpPacker->AddField("branch_no");  
lpPacker->AddInt(1000);  
lpPacker->AddField("fund_account", 'S');    ///应该先把Field都打好  
lpPacker->AddStr("100000001");
```

#### 2) 打包对应→Field 数量=Value 数量

正确 ✓

```
lpPacker->AddField("branch_no");  
lpPacker->AddField("fund_account", 'S');  
lpPacker->AddInt(1000);  
lpPacker->AddStr("100000001");
```

错误 ✗

```
lpPacker->AddField("branch_no");
```

```
lpPacker->AddField("fund_account", 'S');
lpPacker->AddInt(1000);
lpPacker->AddStr("100000001");
lpPacker->AddInt(0); //多了一行Value
```

- 订阅消息中心打包，需打到 KeyInfo 中

```
IF2Packer* lpF2Packer = NewPacker(2);
lpF2Packer->AddRef();
lpF2Packer->BeginPack();
lpF2Packer->AddField("fund_account", 'S');//资金账号
lpF2Packer->AddField("branch_no");//营业部号
lpF2Packer->AddStr("100000001");//打资金账号的值
lpF2Packer->AddInt(1000);//打营业部号的值
lpF2Packer->EndPack();

REQ_DATA Request;
Request.fileHeadLen = 0;
Request.issueType = 23;
Request.keyInfoLen = lpF2Packer->GetPackLen();
Request.lpFileHead = NULL;
Request.lpKeyInfo = lpF2Packer->GetPackBuf();
Request.packetType = 0;
Request.sececeNo = 0;

lpConnection->SendBizEx(620001, lpF2Packer, NULL, 1, 0, 1, 0, &Request);

lpF2Packer->FreeMem(lpF2Packer->GetPackBuf());
lpF2Packer->Release();
```

## 心跳维护

```
OnReceivedBizEx(CConnectionInterface *lpConnection, int hSend, LPRET_DATA lpRetData, const void
*lpUnpackerOrStr, int nResult)
{
    if(lpRetData->functionID == 620000)
    {
        REQ_DATA Request;
        memset(&Request, 0, sizeof(REQ_DATA));
        Request.packetType = 1;
        Request.routeInfo = lpRetData->sendInfo;
        lpConnection->SendBizEx(lpRetData->functionID, NULL, NULL, 1, 0, 1, 0, &Request);
    }
}
```

```
return;  
}
```

## 断开重连

由于 T2\_SDK 考虑到多连接应用，所以 SDK 底层实现不支持断开重连。另外，**设法在 OnClose 内部调用 Connect 来实现断开重连也是不可行的**。那调用者该如何实现断开重连功能呢？一种可行的解决方案是在 OnClose 内部触发一个事件，而由 wait 该事件的其他线程去执行断开重连操作，具体请参考 demo。

重新连接成功后，需要重新登录获得 user\_token，并重新订阅。

## 分页查询

恒生分页查询用 position\_str、request\_num，request\_num 是请求数（证券接口中一般默认为 50，最大支持 1000），position\_str 是返回的数据的定位串。

分页查询流程如下：

- 1) position\_str="" 发送第一个查询请求包。
- 2) 从应答包中获得最后一条数据的 position\_str，做为下一个请求包中 position\_str 输入参数。如此循环发送。
- 3) 直止应答包 lpUnpacker->GetRowCount() == 0。