

lev2mdapi (C++) 开发指南

上海泰琰信息科技有限公司

修订历史

日期	版本	修订说明	修订人
20170925	3.4.2_20170925.09:55:55	初稿	Tora
20171202	3.4.11_20171202.11:38:11	<ul style="list-style-type: none"> lev2mdapi 增加通过组播方式订阅 lev2 行情 增加对常见使用问题的说明 	Tora
20180118	3.5.3_20180118.11:59:19	<ul style="list-style-type: none"> 行情快照增加证券状态、涨跌停板价（仅深圳支持）和今日收盘价（仅上海支持）推送 增加对深圳逐笔委托和逐笔成交行情的对应关系说明 	Tora
20180703	3.6.5_20180703.18:09:35	<ul style="list-style-type: none"> OnRtnMarketData 函数增加四个参数用于回传上证 lev2 行情快照一档价上的委托量（最多 50 笔） 	Tora
20190403	3.8.4_20190403.16:35:09	<ul style="list-style-type: none"> 增加接口用于订阅/退订和收取盘后定价行情 	Tora
20190729	3.8.5_20190729.17:28:05	<ul style="list-style-type: none"> 增加接口支持订阅重传的逐笔委托和重传的逐笔成交 	Tora
20191217	3.8.7_20191217.17:00:00	<ul style="list-style-type: none"> 逐笔成交增加字段 TradeBSFlag, 内外盘标志（上海交易所有效） 	Tora
20200325	3.8.9_20200325.17:30:00	<ul style="list-style-type: none"> OnRtnMarketData 增加以下字段推送（只有上海行情有效）： <ul style="list-style-type: none"> TotalBidNumber: 买入总笔数 TotalOfferNumber: 卖出总笔数 BidTradeMaxDuration: 买入委托成交量最大等待时间 OfferTradeMaxDuration: 卖出委托成交量最大等待时间 登录报文增加 IP/MAC 信息的采集 	Tora
20200818	3.9.1_20200818.12:00:00	<ul style="list-style-type: none"> OnRtnMarketData 增加基金实时净值 IOPV 推送 OnRtnMarketData 接口深圳行情也推送以下字段： <ul style="list-style-type: none"> TotalBidVolume: 委托买入总量 AvgBidPrice: 加权平均委托买价格 TotalAskVolume: 委托卖出总量 AvgAskPrice: 加权平均委托卖价格 	Tora
20210222	3.9.3_20210222.18:30:00	<ul style="list-style-type: none"> OnRtnMarketData 增加各档位委托笔数的推送（买卖十档共增加 20 个字段）： <ul style="list-style-type: none"> Ask1NumOrders ~ Ask10NumOrders Bid1NumOrders ~ Bid10NumOrders 	Tora
20210322	4.0.0_20210322.17:00:00	<ul style="list-style-type: none"> 修改接口文件名： 	Tora

	0:00	<p>TORATstpUserApiDataType.h ->TORATstpLev2ApiDataType.h TORATstpUserApiStruct.h -> TORATstpLev2ApiStruct.h</p> <ul style="list-style-type: none"> ● 精简接口文件类型定义、结构体定义，删掉无用类型和结构体 ● 引入命名空间，所有数据类型、结构体和接口类都在命名空间TORALEV2API ● 逐笔委托接口新增字段： OrderNO: 委托序号。用来关联逐笔成交和逐笔委托，详见第 6 节 OrderStatus: 委托状态。用来区分上海证券交易所的普通委托和撤单委托，详见第 6 节说明 ● 修订/补充第 6 节内容 	
20210426	4.0.0_20210426.11:30:00	<ul style="list-style-type: none"> ● 逐笔委托和逐笔成交接口新增字段： ● BizIndex: 业务序号。用来说明上海逐笔委托和逐笔成交的先后关系 	Tora
20210526	4.0.0_20210526.15:00:00	<ul style="list-style-type: none"> ● 修改 TORATstpLev2ApiDataType.h 文件里宏定义，避免与其它 API 同时使用时命名冲突问题 	Tora
20210625	4.0.0_20210625.18:00:00	<ul style="list-style-type: none"> ● 优化组播接收缓存 	Tora
20210707	4.0.0_20210707.10:00:00	<ul style="list-style-type: none"> ● 支持按照代码段订阅各类别行情。例如：订阅证券代码为：600***订阅所有600开头的证券行情，51****订阅所有51开头的行情 ● 修改CTORATstpLev2MdApi::Init接口支持api内部网络接收线程绑核： /// @param cpuCores: API 内部线程绑核参数，默认不绑核运行。"0"表示 API 内部线程绑定到第 0 核上运行。"0,5,18"表示 API 内部线程绑定到第 0, 第 5, 第 18 号核上运行 virtual void Init(const char *cpuCores="") = 0 	Tora
20210713	4.0.1_20210713.10:30:00	<ul style="list-style-type: none"> ● 增加订阅和推送上海新债券行情的接口 CTORATstpLev2MdApi { SubscribeXTSMarketData UnSubscribeXTSMarketData SubscribeXTSTick UnSubscribeXTSTick } CTORATstpLev2MdSpi: { OnRtnXTSMarketData OnRtnXTSTick 	Tora

		OnRspSubXTSMarketData OnRspUnSubXTSMarketData OnRspSubXTSTick OnRspUnSubXTSTick }	
20210719	4.0.1_20210719.17:00:00	<ul style="list-style-type: none"> 上海快照行情OnRtnMarketData和债券XTS快照行情OnRtnXTSMarketData增加以下字段推送(只有上海行情有效) WithdrawBuyNumber: 买入撤单笔数 WithdrawBuyAmount: 买入撤单数量 WithdrawBuyMoney: 买入撤单金额 WithdrawSellNumber: 卖出撤单笔数 WithdrawSellAmount: 卖出撤单数量 WithdrawSellMoney: 卖出撤单金额 	Tora
20210111	4.0.1_20211011.11:00:00	<ul style="list-style-type: none"> 修复部分问题 	Tora
20211223	4.0.1_20211223.17:00:00	<ul style="list-style-type: none"> 修复缓存模式接收 XTS 债券快照部分字段不正确问题 	Tora
20211117	4.0.2_20211117.16:00:00	<ul style="list-style-type: none"> 支持多次调用RegisterMulticast函数注册多个组播地址同时接收组播行情 修改RegisterMulticast函数接口定义, 支持Linux上Solarflare网卡efvi模式接收行情 	Tora
20211126	4.0.2_20211126.17:00:00	<ul style="list-style-type: none"> 修复部分问题 	Tora
20220117	4.0.2_20220117.15:30:00	<ul style="list-style-type: none"> 修复缓存模式接收XTS债券快照部分字段不正确问题 	Tora
20220408	4.0.4_20220408.19:30:00	<ul style="list-style-type: none"> 增加订阅和推送深圳新债券(不含可转债)行情的接口 CTORATstpLev2MdApi { Un/SubscribeBondMarketData Un/SubscribeBondTranction Un/SubscribeBondOrderDetail } CTORATstpLev2MdSpi { OnRtnBondMarketData OnRtnBondTranction OnRtnBondOrderDetail OnRspUn/SubBondMarketData OnRspUn/SubBondTranction OnRspUn/SubBondOrderDetail } 	Tora
20220728	4.0.5_20220728.16:30:00	<ul style="list-style-type: none"> 增加订阅和推送上海逐笔合并行情的接口 	Tora

	0:00	CTORATstpLev2MdApi { Un/SubscribeNGSTick } CTORATstpLev2MdSpi { OnRspUn/SubNGSTick OnRtnNGSTick } <ul style="list-style-type: none"> 重新修订本指南的内容结构。针对各函数接口展开说明相关注意事项 	
20221018	4.0.5_20221018.16:00:00	<ul style="list-style-type: none"> 修复部分问题 	Tora
20221116	4.0.6_20221116.17:00:00	<ul style="list-style-type: none"> 行情订阅模式增加PROXY模式（详见3.3.2节） 增加行情类别订阅申明接口（详见3.3.3节） DeclareMKSubTypes <ul style="list-style-type: none"> 补充对绑核数目的说明（见附录->初始化函数说明） 	Tora
20230213	4.0.6_20230213.17:00:00	<ul style="list-style-type: none"> 修复调用Release可能触发的宕机 	Tora
20231012	4.0.6_20230213.17:00:00	<ul style="list-style-type: none"> 增加在上海非债券类NGTS逐笔合并消息未启用前，上海非债券类逐笔成交和逐笔委托的订阅、退订和接收说明 增加逐笔丢包判断说明 	Tora
20240425	4.0.7_20240425.17:00:00	<ul style="list-style-type: none"> 升级efvi支持X3系列solarflare 网卡 	Tora

目录

修订历史	I
目录	V
1. 编写目的	1
2. 接口概述	1
2.1 接口文件列表	1
2.2 接口运行环境	1
2.4 接口通用参数	1
2.5 接口命名空间	1
3. 接口函数	2
3.1 行情接口初始化	2
3.2 行情登录	3
3.2.1 登录请求	3
3.2.3 登录响应	3
3.3 行情订阅与处理	4
3.3.1 行情类别与订阅	4
3.3.2 行情订阅模式	5
3.3.3 行情类别订阅申明	5
3.3.4 订阅请求	6
3.3.5 订阅响应	6
3.3.6 行情处理	7
3.4 附录一：行情接口清单	7
请求接口类(CTORATstpLev2MdApi)	7
(1) 创建接口对象	7
(2) 释放接口对象	8
(3) 获取版本号	8
(4) 初始化	8
(5) 等待	8
(6) 注册行情服务地址	9
(7) 注册组播行情地址	9
(8) 注册名字服务地址	10
(9) 声明行情订阅类别集合	10
(10) 注册回调接口	10
(11) 用户登录请求	10
(12) 用户登出请求	11
(13) 退订、订阅上海/深圳非债券类、深圳可转债快照行情	11
(14) 退订、订阅指数行情	11
(15) 退订、订阅深圳非债券类、深圳可转债逐笔成交	11
(16) 退订、订阅深圳非债券类、深圳可转债逐笔委托	12
(17) 退订、订阅盘后定价快照行情	12

(18)	退订、订阅盘后定价逐笔成交行情	12
(19)	退订、订阅逐笔重传成交	12
(20)	退订、订阅逐笔重传委托	13
(21)	退订、订阅上海 XTS 债券行情	13
(22)	退订、订阅上海 XTS 债券逐笔行情	13
(23)	退订、订阅上海非债券类逐笔委托 (NGTS 逐笔合并消息启用前)	13
(24)	退订、订阅上海非债券类逐笔成交 (NGTS 逐笔合并消息启用前)	14
(25)	退订、订阅上海 NGTS 非债券类逐笔行情	14
(26)	退订、订阅深圳债券 (不含可转债) 快照行情	14
(27)	退订、订阅深圳债券 (不含可转债) 逐笔成交行情	14
(28)	退订、订阅深圳债券 (不含可转债) 逐笔委托行情	15
回调接口类 (CTORATstpLev2MdSpi)		15
(1)	连接成功	15
(2)	连接断开	15
(3)	错误应答	16
(4)	用户登录响应	16
(5)	用户登出响应	16
(6)	退订、订阅响应	16
(7)	上海/深圳非债券类、深圳可转债行情通知	17
(8)	指数行情通知	17
(9)	深圳非债券类、深圳可转债逐笔成交通知	17
(10)	深圳非债券类、深圳可转债逐笔委托通知	18
(11)	盘后定价行情通知	18
(12)	盘后定价逐笔成交通知	18
(13)	逐笔重传成交通知	18
(14)	逐笔重传委托通知	19
(15)	上海 XTS 债券行情通知	19
(16)	上海 XTS 债券逐笔通知	19
(17)	上海非债券类逐笔成交通知 (NGTS 逐笔合并消息启用前)	20
(18)	上海非债券类逐笔委托通知 (NGTS 逐笔合并消息启用前)	20
(19)	上海 NGTS 非债券类逐笔通知	20
(20)	深圳债券 (不含可转债) 行情通知	21
(21)	深圳债券 (不含可转债) 逐笔成交通知	21
(22)	深圳债券 (不含可转债) 逐笔委托通知	22

1. 编写目的

本文旨在帮助开发者快速查阅上海泰琰信息科技有限公司开发的 lev2 行情服务 API 接口的使用方法、参数说明及相关注意事项。

2. 接口概述

2.1 接口文件列表

文件名	说明
TORATstpLev2ApiDataType.h	包含所有用到的数据的类型定义
TORATstpLev2ApiStruct.h	包含所有用到的数据结构定义
TORATstpLev2MdApi.h	包含交易相关的业务请求和响应接口定义
liblev2mdapi.so	linux 环境交易接口动态库
lev2mdapi.lib	windows 环境交易接口动态库
lev2mdapi.dll	windows 环境交易接口动态库

2.2 接口运行环境

- 开发语言: C++
- windows 环境: VS2013(VC12)
- linux 环境: gcc4.4.7, glibc2.12

2.4 接口通用参数

- **nRequestID**: 开发者调用请求接口时可以为请求指定的一个编号, 该编号会在对应的响应接口中返回, 开发者可利用该编号将响应和请求进行关联。
- **bIsLast**: 当响应需要携带的数据包过大时, 该数据包会被分割成多个小的数据包并按顺序逐次发送, 这种场景下响应函数会被调用多次, 参数 **bIsLast** 用于描述当前收到的响应数据包是否为最后一个报文。目前 lev2api 的响应函数 **bIsLast** 均为 **true**。
- **pRspInfo**: 用来描述请求处理结果, 包含错误代码 **ErrorID** 和错误信息 **ErrorMsg** 两个字段, 开发者可利用该参数中的 **ErrorID** 来判断请求是否成功, **ErrorID** 为 0 表示请求成功, 非 0 表示请求失败, **ErrorMsg** 表示失败原因。

2.5 接口命名空间

lev2mdapi 接口提供的类型和数据结构定义均包含在 TORALEV2API 命名空间内, 开发

者使用 API 时需引入该命名空间。为防止与其它系统接口中的符号定义冲突，建议在使用类型或方法前加命名空间，而不是直接 using namespace TORALEV2API 全局引入。以下示例介绍中仅涉及到 lev2mdapi 接口的使用，故直接全局引入命名空间。

3. 接口函数

3.1 行情接口初始化

第一步，引入行情接口头文件：

```
1. #include "TORATstpLev2MdApi.h"
```

第二步，配置头文件所在路径，动态链接库所在路径及动态链接库；

第三步，引入命名空间：

```
1. using namespace TORALEV2API;
```

第四步，继承回调接口类并根据需要实现相关虚函数：

```
1. class DemoLev2MdSpi : public CTORATstpLev2MdSpi
2. {
3. public:
4.     DemoLev2MdSpi (CTORATstpLev2MdApi *api)
5.     {
6.         m_api = api;
7.         m_request_id = 0;
8.     }
9.
10.    ~ DemoLev2MdSpi ()
11.    {
12.    }
13.
14.    virtual void OnFrontConnected()
15.    {
16.    }
17.
18.    virtual void OnFrontDisconnected(int nReason)
19.    {
20.    }
21. private:
22.     int m_request_id;
23.     CTORATstpLev2MdApi *m_api;
24. };
```

第五步，创建接口对象并完成初始化：

```

1. CTORATstpLev2MdApi *api = CTORATstpLev2MdApi::CreateTstpLev2MdApi();
2. DemoLev2MdSpi spi(api);
3. api->RegisterSpi(&spi);
4. api->RegisterFront("tcp://127.0.0.1:8500");
5. api->Init();

```

自此，一个简单的行情接口 DEMO 就开发完成了，当接口连接成功时会回调 DemoLev2MdSpi 类的 OnFrontConnected 函数，连接断开时会回调 OnFrontDisconnected 函数。

说明：

- 创建接口实例时可指定行情接收方式（tcp、udp 单播、udp 组播、PROXY）和缓存模式，具体参见附录接口说明；
- 注册回调对象、注册行情地址必须在初始化 Init 之前完成，Init 函数调用后接口线程启动开始尝试连接后台服务。

3.2 行情登录

当API接口成功连接至服务端后，需要登录行情服务系统进行身份认证。身份认证方式和信息请咨询券商相关技术支持。

3.2.1 登录请求

```

1. virtual void OnFrontConnected()
2. {
3.     CTORATstpReqUserLoginField req_user_login_field;
4.     memset(&req_user_login_field, 0, sizeof(req_user_login_field));
5.     strcpy(req_user_login_field.LoginAccount, "1111111");
6.     req_user_login_field.LoginAccountType = TORA_TSTP_LACT_UserID;
7.     //req_user_login_field.AuthMode = TORA_TSTP_SP_AM_Password;
8.     strcpy(req_user_login_field.Password, "123456");
9.     strcpy(req_user_login_field.UserProductInfo, "lev2apidemo");
10.
11.     int ret = m_api->ReqUserLogin(&req_user_login_field, ++m_request_id);
12.     if (ret != 0)
13.     {
14.         printf("ReqUserLogin fail, ret[%d]\n", ret);
15.     }
16. }

```

3.2.3 登录响应

登录响应通过 OnRspUserLogin 函数回调，需要在 DemoLev2MdSpi 类中实现该虚函数：

```

1. virtual void OnRspUserLogin(CTORATstpRspUserLoginField *pRspUserLogin,
2. CTORATstpSPRspInfoField *pRspInfo, int nRequestID, bool bIsLast)
3. {
4.     if (pRspInfo->ErrorID == 0)
5.     {
6.         printf("login success\n");
7.     }
8.     else
9.     {
10.        printf("login fail, error_id[%d] error_msg[%s]\n",
11.              pRspInfo ->ErrorID, pRspInfo ->ErrorMsg);
12.    }
13. }

```

登录成功之后，就可以根据需要订阅各类行情。

3.3 行情订阅与处理

3.3.1 行情类别与订阅

交易所	行情类别	退订/订阅函数	行情通知函数	说明
上交所	非债券类逐笔行情	Un/Sub~NGTSTick	OnRtnNGTSTick	交易所启用 NGST 逐笔合 并行情
		Un/Sub~OrderDetail Un/Sub~Transaction	OnRtnOrderDetail OnRtnTransaction	交易所未启用 NGST 逐笔合 并行情
	非债券类快照行情	Un/Sub~MarketData	OnRtnMarketData	
	债券类逐笔行情	Un/Sub~XTSTick	OnRtnXTSTick	
	债券类快照行情	Un/Sub~XTSMarketData	OnRtnXTSMarketData	
	盘后定价快照行情	Un/Sub~PHMarketData	OnRtnPHMarketData	
	盘后定价逐笔成交	Un/Sub~PHTransaction	OnRtnPHTransaction	
	指数行情	Un/Sub~Index	OnRtnIndex	
深交所	非债券类、可转债逐笔行情	Un/Sub~OrderDetail Un/Sub~Transaction	OnRtnOrderDetail OnRtnTransaction	
	非债券类、可转债快照行情	Un/Sub~MarketData	OnRtnMarketData	
	债券类（不含可转债）逐笔行情	Un/Sub~BondOrderDetail Un/Sub~BondTransaction	OnRtnBondOrderDetail OnRtnBondTransaction	
	债券类（不含可转债）快照行情	Un/Sub~BondMarketData	OnRtnBondMarketData	
	盘后定价快照行情	Un/Sub~PHMarketData	OnRtnPHMarketData	
	逐笔重传	Un/Sub~ResendOrderDe	OnRtnResendOrderData	咨询券商技术

	tail Un/Sub~ResendTransac tion	il OnRtnResendTransacti on	支持了解此功 能是否开通
指数行情	Un/Sub~Index	OnRtnIndex	

3.3.2 行情订阅模式

行情订阅模式指行情数据从行情服务端传输到 API 接口的传输方式，在创建 API 接口实例函数（`CreateTstpLev2MdApi`）时通过 `eMDSubMode` 参数传入，可填入的值为：`TORA_TSTP_MST_TCP`、`TORA_TSTP_MST_UDP`、`TORA_TSTP_MST_MCAST`、`TORA_TSTP_MST_PROXY`，分别对应 tcp 传输、UDP 单播传输、UDP 组播传输和 PROXY 模式，以行情服务端实际部署方式为准。

行情订阅模式与后续需要调用的行情地址注册函数有对应关系，说明如下：

行情订阅模式	地址注册函数	实例
<code>TORA_TSTP_MST_TCP</code>	<code>RegistFront</code>	<code>tcp://192.168.0.1:17001</code>
<code>TORA_TSTP_MST_UDP</code>	<code>RegistFront</code>	<code>tcp://192.168.0.1:17001</code>
<code>TORA_TSTP_MST_MCAST</code>	<code>RegisterMulticast</code>	<code>udp://224.3.9.110:34567</code>
<code>TORA_TSTP_MST_PROXY</code>	<code>RegistFront</code>	<code>tcp://127.0.0.1:8899</code>

3.3.3 行情类别订阅申明

为提高在 PROXY 行情订阅模式下行情过滤的效率，可在 `init` 函数调用前调用 `DeclareMKSubTypes` 函数声明后续需要订阅的行情类别集合。行情类别可参考 `TORATstpLev2ApiDataType.h` 文件 `TTORATstpMKSubTypeType` 类型说明。部分行情类别标识列举如下：

行情类别	标识	对应回调接口
快照行情	M	<code>OnRtnMarketData</code>
指数	I	<code>OnRtnIndex</code>
逐笔成交	T	<code>OnRtnTransaction</code>
逐笔委托	O	<code>OnRtnOrderDetail</code>
盘后定价行情	H	<code>OnRtnPHMarketData</code>
盘后定价逐笔成交	P	<code>OnRtnPHTransaction</code>
债券快照行情(上海)	X	<code>OnRtnXTSMarketData</code>
债券逐笔行情(上海)	K	<code>OnRtnXTSTick</code>
债券快照行情	U	<code>OnRtnBondMarketData</code>
债券逐笔成交	W	<code>OnRtnBondTransaction</code>
债券逐笔委托	V	<code>OnRtnBondOrderDetail</code>
逐笔合并行情（上海，待启用）	N	<code>OnRtnNGTSTick</code>

无论是否调用 `DeclareMKSubTypes` 提前声明需要订阅的行情类别集合，登录成功后都需要通过调用 `Subscriber***` 函数订阅代码级别的行情。但一旦调用了

DeclareMKSubTypes 声明了行情类别集合, 后续的 **Subscriber***** 函数也需要在声明的行情类别集合范围内。

实例:

```
TTORATstpMKSubTypesType mksubtypes;
strcpy(mksubtypes, "MTO");
```

DeclareMKSubTypes(mksubtypes): 表示声明订阅行情快照和逐笔行情, 登录成功后需要调用 **SubscribeMarketData**、**SubscribeTransaction**、**SubscribeOrderDetail**。

3.3.4 订阅请求

客户根据自身需要对照 3.3.1 节表格内容在登录成功后订阅各类行情。下面以订阅上海证券交易所非债券类快照行情为例说明订阅函数的使用方式:

```
1. char * security_list[]={ "510050", "600***", "688***" };
2. int ret = m_api->SubscribeMarketData (security_list, 3, TORA_TSTP_EXD_SSE);
3. if (ret != 0)
4. {
5.     printf("SubscribeMarketData fail, ret[%d]\n", ret);
6. }
```

说明:

- 支持按照代码段订阅各类行情。例如: 订阅证券代码为: 600*** 订阅所有 600 开头的证券行情, 51*** 订阅所有 51 开头的行情
- 代码段订阅只能用通配符通配证券代码最后几位, 不能通配以下几类情况: ***110、600**1、51***0、5*0*1*。

3.3.5 订阅响应

订阅请求 **Subscribe~** 通过 **OnRspSub~** 返回处理结果, 需要在 **DemoLev2MdSpi** 中实现该虚函数:

```
1. virtual void
   OnRspSubMarketData(CTORATstpSpecificSecurityField*pSpecificSecurity,
   CTORATstpRspInfoField*pRspInfo, int nRequestID, bool bIsLast)
2. {
3.     if (pRspInfo->ErrorID == 0)
4.     {
5.         printf("sub market data success! \n");
6.     }
7.     else
8.     {
9.         printf("sub market data fail, error_id[%d] error_msg[%s] \n",
10.             pRspInfo->ErrorID, pRspInfo->ErrorMsg);
11.     }
12. }
```

说明:

- 可通过回调接口中 **pRspInfo** 参数中的 **ErrorID** 来判断订阅请求成功或失败, **ErrorID**

为 0 表示订阅请求处理成功， ErrorID 非 0 表示订阅请求处理失败， ErrorMessage 为拒绝原因。在未登录或订阅的证券代码不符合格式要求时，会订阅失败。

- 组播接收模式时，在订阅请求里订阅多只证券，每个证券都会回调一次订阅响应。后续可能会对此进行优化。

3.3.6 行情处理

当订阅行情成功且 lev2 服务有相应的行情数据推送时， api 端通过 OnRtn~函数进行行情回调； api 开发者需要在 DemoLev2MdSpi 中实现 OnRtn~虚函数才能获取推送的行情。

```
1. virtual void OnRtnMarketData(CTORATstpLev2MarketDataField *pDepthMarketData,
2. const int FirstLevelBuyNum, const int FirstLevelBuyOrderVolumes[],
3. const int FirstLevelSellNum, const int FirstLevelSellOrderVolumes[])
4. {
5.     printf("OnRtnMarketData: security_id[%s] last_price[%f] time_stamp[%d]
6.     total_value_trade[%lld]\n",
7.     pDepthMarketData ->SecurityID,
8.     pDepthMarketData ->LastPrice,
9.     pDepthMarketData ->DataTimeStamp,
10.    pDepthMarketData ->TotalValueTrade);
11. }
```

3.4 附录一：行情接口清单

请求接口类(CTORATstpLev2MdApi)

(1) 创建接口对象

函数名称	CreateTstpLev2MdApi
函数原型	static CTORATstpLev2MdApi *CreateTstpLev2MdApi(const TTORATstpMDSubModeType &eMDSubMode= TORA_TSTP_MST_TCP, bool bCachedMode=false);
函数描述	创建接口对象。支持创建多个接口对象实例，每个对象实例单独运行线程。不再使用接口对象时需调用 Release 释放接口对象，否则将引起内存泄漏。
参数说明	eMDSubMode ：行情订阅模式（TORA_TSTP_MST_TCP、TORA_TSTP_MST_UDP、TORA_TSTP_MST_MCAST、TORA_TSTP_MST_PROXY）。目前主要支持组播方式TORA_TSTP_MST_MCAST；lev2 服务程序支持的行情订阅模式请咨询券商技术支持。 bCachedMode ：是否启用缓存模式。缓存模式下，一个接口对象实例会额外创建一个处理线程，底层网络线程先将收到的行情数据缓存在 api 内部的内存流里，再由处理线程读取内存流的行情进行回调到应用层；非缓存模式下，底层网络线程直接将收到的行情回调到应用层。开启缓存模式可以减少非缓存模式下因行情回调函数处理耗时过长引起的组播行情丢包。
返回说明	返回接口对象指针。

(2) 释放接口对象

函数名称	Release
函数原型	<code>virtual void Release() = 0;</code>
函数描述	释放接口对象。需注意不能在回调函数中调用该函数，否则将引起线程死锁。
参数说明	无。
返回说明	无。

(3) 获取版本号

函数名称	GetApiVersion
函数原型	<code>static const char* GetApiVersion();</code>
函数描述	获取接口版本号。
参数说明	无。
返回说明	返回接口版本号。

(4) 初始化

函数名称	Init
函数原型	<code>virtual void Init(const char * cpuCores="") = 0;</code>
函数描述	接口初始化。调用该函数后接口线程开始工作。
参数说明	<p>cpuCores:api 内部线程绑核参数，默认不绑核运行，"0"表示 API 内部线程绑定到第 0 核上运行，"0,5,18"表示 API 内部线程绑定到第 0, 第 5, 第 18 号核上运行。</p> <p>注： 组播模式收取行情时，需要传递的核的数目= 注册的组播地址数目+【缓存模式？ 1: 0】 PROXY 模式收取行情时，具体需要的核的数目与行情服务端配置、行情类别订阅声明函数的传入参数相关，建议配置 4 个以上核，具体咨询券商技术支持。</p>
返回说明	无。

(5) 等待

函数名称	Join
函数原型	<code>virtual int Join() = 0;</code>
函数描述	阻塞当前线程，等待接口线程退出。调用该接口后，当前线程将被阻塞，直到接口线程退出，一般情况下接口线程不会主动退出，除非开发者调用了 Release 接口。开发者可根据需要选择是否调用该接口。
参数说明	无。
返回说明	返回 0。

(6) 注册行情服务地址

函数名称	RegisterFront
函数原型	<code>virtual void RegisterFront(char *pszFrontAddress) = 0;</code>
函数描述	注册行情服务器地址。该函数应在 Init 之前调用。用 tcp 模式收取行情时，需要调用此函数注册 tcp 行情服务地址。
参数说明	pszFrontAddress : 行情服务地址串，格式“tcp://127.0.0.1:6500”，支持注册多个行情服务地址备用，多个地址间用英文逗号隔开，例如“tcp://127.0.0.1:6500,tcp://127.0.0.1:26500”，也可通过多次调用 RegisterFront 来注册多个行情服务地址。注册多个 tcp 行情服务地址是为了实现灾备自动切换行情服务地址，在 api 内部当一个行情服务地址连接不上时，会自动切换地址尝试连接新地址。
返回说明	无。

(7) 注册组播行情地址

函数名称	RegisterMulticast
函数原型	<code>virtual void RegisterMulticast(char *pszMulticastAddress, char *pszInterfaceIP, char *pszSourceIp, const char *pInterfaceName="", int rxqCapacity=0, bool bEFVI=false) = 0;</code>
函数描述	<p>注册组播服务器地址。该函数应在 Init 之前调用。用组播模式收取行情时，需要调用此函数注册组播行情地址。该函数可以调用多次注册多个组播地址，api 内部会开启对应数目的组播接收通道同时从多个组播地址收取行情。注册多个组播地址需要注意：</p> <ul style="list-style-type: none"> ● 请提前咨询相关技术支持有多少组播地址以及每个组播地址发送的行情类别，api 开发者可根据实际需要注册相应的组播地址。 ● 不要重复注册完全相同的组播地址。 ● 每个组播接收通道是独立线程驱动，为了保证性能，init 函数传递的处理器核个数不能比组播接收通道少。 ● 各组播接收通道收到的行情加锁回调，开发者不需要考虑同一时刻有多个响应函数回调的场景。 ● 所有的组播通道都加入成功(都收到了数据)时回调 OnFrontConnected，因此不要注册无效的组播地址。 ● 所有的组播通道在一定时间内都未收到数据时回调 OnFrontDisConnected。 ● 所有组播通道共享订阅信息，如果一个组播通道推送的是逐笔行情，而订阅的是快照行情，则此组播通道无数据回调。
参数说明	<p>pszMulticastAddress: 组播行情服务地址串，格式“udp://224.3.9.110:34567”</p> <p>pszInterfaceIp: 接收网卡地址,如:"127.0.0.1",填 NULL 则依次轮询尝试本机所有网卡加入组播组</p> <p>pszSourceIp: 组播数据包源地址,如:"127.0.0.1",填 NULL 表示不校验数据包源</p> <p>pInterfaceName: 接收网卡名,如 enp101s0f1,efvi 接收时有意义</p> <p>rxqCapacity: 接收队列大小,rxqCapacity<=0 则取默认值, (512, 1024, 2048, 4096),efvi 接收时有意义。为了避免丢包可尝试设置 4096, 如果设置的值超过了网卡的极大值, 程序会直接退出, 此时可减小 rxqCapacity 入参值重启</p>

	bEFVI:EFVI 接收模式开关
返回说明	无。

(8) 注册名字服务地址

函数名称	RegisterNameServer
函数原型	<code>virtual void RegisterNameServer(char *pszNsAddress) = 0;</code>
函数描述	注册名字服务器地址。该函数应在 Init 之前调用。
参数说明	pszNsAddress : 名字服务地址串, 格式“tcp://127.0.0.1:12001”, 支持注册多个名字服务地址备用, 多个地址间用英文逗号隔开, 例如“tcp://127.0.0.1:12001,tcp://127.0.0.1:22001”, 也可通过多次调用 RegisterNameServer 来注册多个服务地址。 此函数功能暂不启用。
返回说明	无。

(9) 声明行情订阅类别集合

函数名称	DeclareMKSubTypes
函数原型	<code>virtual void DeclareMKSubTypes (const TTORATstpMKSubTypesType& mkSubTypes) = 0;</code>
函数描述	声明行情订阅类别集合。该函数应在 Init 之前调用。
参数说明	mkSubTypes : 行情类别集合。
返回说明	无。

(10) 注册回调接口

函数名称	RegisterSpi
函数原型	<code>virtual void RegisterSpi(CTORATstpLev2MdSpi *pSpi) = 0;</code>
函数描述	注册回调接口。该函数应在 Init 之前调用。
参数说明	pSpi : 派生自回调接口类 CTORATstpLev2MdSpi 的实例对象。
返回说明	无。

(11) 用户登录请求

函数名称	ReqUserLogin
函数原型	<code>virtual int ReqUserLogin(CTORATstpReqUserLoginField *pReqUserLoginField, int nRequestID) = 0;</code>
函数描述	用户登录请求。
参数说明	pReqUserLoginField : 请求业务域; nRequestID : 请求编号。
返回说明	0 表示成功, -1 表示未注册行情地址或未连接, -3 或 -5 表示网络写失败。

(12) 用户登出请求

函数名称	ReqUserLogout
函数原型	<code>virtual int ReqUserLogout(CTORATstpSPUserLogoutField *pUserLogoutField, int nRequestID) = 0;</code>
函数描述	用户登出请求。
参数说明	pUserLogoutField : 请求业务域; nRequestID : 请求编号。
返回说明	0 表示成功, -1 表示未连接或未登录, -3 或 -5 表示网络写失败。

(13) 退订、订阅上海/深圳非债券类、深圳可转债快照行情

函数名称	Un/SubscribeMarketData
函数原型	<code>virtual int Un/SubscribeMarketData(char *ppSecurityID[],int nCount,TTORATstpExchangeIDType ExchangeID) = 0;</code>
函数描述	退订/订阅上海非债券类、深圳非债券类、深圳可转债快照行情。
参数说明	ppSecurityID : 退订/订阅证券列表; nCount : 退订/订阅的证券个数。
返回说明	0 表示成功, -1 表示未连接, -3 未登录, -5 表示网络写失败。

(14) 退订、订阅指数行情

函数名称	Un/SubscribeIndex
函数原型	<code>virtual int Un/SubscribeIndex(char *ppSecurityID[],int nCount,TTORATstpExchangeIDType ExchangeID) = 0;</code>
函数描述	退订/订阅沪深指数行情。
参数说明	ppSecurityID : 退订/订阅证券列表; nCount : 退订/订阅的证券个数。
返回说明	0 表示成功, -1 表示未连接, -3 未登录, -5 表示网络写失败。

(15) 退订、订阅深圳非债券类、深圳可转债逐笔成交

函数名称	Un/SubscribeTransaction
函数原型	<code>virtual int Un/SubscribeTransaction(char *ppSecurityID[],int nCount,TTORATstpExchangeIDType ExchangeID) = 0;</code>
函数描述	退订/订阅深圳非债券类、深圳可转债逐笔成交。在上海 NGTS 合并
参数说明	ppSecurityID : 退订/订阅证券列表; nCount : 退订/订阅的证券个数。
返回说明	0 表示成功, -1 表示未连接, -3 未登录, -5 表示网络写失败。

(16) 退订、订阅深圳非债券类、深圳可转债逐笔委托

函数名称	Un/SubscribeOrderDetail
函数原型	<code>virtual int Un/SubscribeOrderDetail (char *ppSecurityID[],int nCount,TTORATstpExchangeIDType ExchangeID) = 0;</code>
函数描述	在上海非债券退订/订阅深圳非债券类、深圳可转债逐笔委托。
参数说明	ppSecurityID: 退订/订阅证券列表; nCount: 退订/订阅的证券个数。
返回说明	0 表示成功, -1 表示未连接, -3 未登录, -5 表示网络写失败。

(17) 退订、订阅盘后定价快照行情

函数名称	Un/SubscribePHMarketData
函数原型	<code>virtual int Un/SubscribePHMarketData (char *ppSecurityID[],int nCount,TTORATstpExchangeIDType ExchangeID) = 0;</code>
函数描述	退订/订阅沪深盘后定价快照行情。
参数说明	ppSecurityID: 退订/订阅证券列表; nCount: 退订/订阅的证券个数。
返回说明	0 表示成功, -1 表示未连接, -3 未登录, -5 表示网络写失败。

(18) 退订、订阅盘后定价逐笔成交行情

函数名称	Un/SubscribePHTransaction
函数原型	<code>virtual int Un/SubscribePHTransaction (char *ppSecurityID[],int nCount,TTORATstpExchangeIDType ExchangeID) = 0;</code>
函数描述	退订/订阅沪深盘后定价逐笔成交行情。目前只有上交所发布此类行情。
参数说明	ppSecurityID: 退订/订阅证券列表; nCount: 退订/订阅的证券个数。
返回说明	0 表示成功, -1 表示未连接, -3 未登录, -5 表示网络写失败。

(19) 退订、订阅逐笔重传成交

函数名称	Un/SubscribeResendTransaction
函数原型	<code>virtual int Un/SubscribeResendTransaction (char *ppSecurityID[],int nCount,TTORATstpExchangeIDType ExchangeID) = 0;</code>
函数描述	退订/订阅逐笔重传成交行情。逐笔重传行情是指 lev2 行情服务程序发现与交易所的行情源有逐笔行情丢包时, 向交易所行情源重新拉取的缺口行情。lev2 行情服务程序是否有转发此类行情请咨询相关技术支持。
参数说明	ppSecurityID: 退订/订阅证券列表; nCount: 退订/订阅的证券个数。
返回说明	0 表示成功, -1 表示未连接, -3 未登录, -5 表示网络写失败。

(20) 退订、订阅逐笔重传委托

函数名称	Un/SubscribeResendOrderDetail
函数原型	<code>virtual int Un/SubscribeResendOrderDetail (char *ppSecurityID[],int nCount,TTORATstpExchangeIDType ExchangeID) = 0;</code>
函数描述	退订/订阅逐笔重传委托行情。lev2 行情服务程序是否有转发此类行情请咨询相关技术支持。
参数说明	ppSecurityID: 退订/订阅证券列表; nCount: 退订/订阅的证券个数。
返回说明	0 表示成功, -1 表示未连接, -3 未登录, -5 表示网络写失败。

(21) 退订、订阅上海 XTS 债券行情

函数名称	Un/SubscribeXTSMarketData
函数原型	<code>virtual int Un/SubscribeXTSMarketData (char *ppSecurityID[],int nCount,TTORATstpExchangeIDType ExchangeID) = 0;</code>
函数描述	退订/订阅上海 XTS 债券行情。
参数说明	ppSecurityID: 退订/订阅证券列表; nCount: 退订/订阅的证券个数。
返回说明	0 表示成功, -1 表示未连接, -3 未登录, -5 表示网络写失败。

(22) 退订、订阅上海 XTS 债券逐笔行情

函数名称	Un/SubscribeXTSTick
函数原型	<code>virtual int Un/SubscribeXTSTick (char *ppSecurityID[],int nCount,TTORATstpExchangeIDType ExchangeID) = 0;</code>
函数描述	退订/订阅上海 XTS 逐笔行情。
参数说明	ppSecurityID: 退订/订阅证券列表; nCount: 退订/订阅的证券个数。
返回说明	0 表示成功, -1 表示未连接, -3 未登录, -5 表示网络写失败。

(23) 退订、订阅上海非债券类逐笔委托 (NGTS 逐笔合并消息启用前)

函数名称	Un/SubscribeOrderDetail
函数原型	<code>virtual int Un/SubscribeOrderDetail (char *ppSecurityID[],int nCount,TTORATstpExchangeIDType ExchangeID) = 0;</code>
函数描述	在上海 NGTS 非债券类逐笔行情启用前, 用此函数退订/订阅上海非债券类逐笔委托。
参数说明	ppSecurityID: 退订/订阅证券列表; nCount: 退订/订阅的证券个数。
返回说明	0 表示成功, -1 表示未连接, -3 未登录, -5 表示网络写失败。

(24) 退订、订阅上海非债券类逐笔成交 (NGTS 逐笔合并消息启用前)

函数名称	Un/SubscribeTransaction
函数原型	<code>virtual int Un/SubscribeTransaction(char *ppSecurityID[],int nCount,TTORATstpExchangeIDType ExchangeID) = 0;</code>
函数描述	在上海 NGTS 非债券类逐笔行情启用前, 用此函数退订/订阅上海非债券类逐笔成交。
参数说明	ppSecurityID : 退订/订阅证券列表; nCount : 退订/订阅的证券个数。
返回说明	0 表示成功, -1 表示未连接, -3 未登录, -5 表示网络写失败。

(25) 退订、订阅上海 NGTS 非债券类逐笔行情

函数名称	Un/SubscribeNGTSTick
函数原型	<code>virtual int Un/SubscribeNGTSTick (char *ppSecurityID[],int nCount,TTORATstpExchangeIDType ExchangeID) = 0;</code>
函数描述	退订/订阅上海 NGTS 非债券类逐笔行情。
参数说明	ppSecurityID : 退订/订阅证券列表; nCount : 退订/订阅的证券个数。
返回说明	0 表示成功, -1 表示未连接, -3 未登录, -5 表示网络写失败。

(26) 退订、订阅深圳债券 (不含可转债) 快照行情

函数名称	Un/SubscribeBondMarketData
函数原型	<code>virtual int Un/SubscribeBondMarketData (char *ppSecurityID[],int nCount,TTORATstpExchangeIDType ExchangeID) = 0;</code>
函数描述	退订/订阅深圳债券类 (不含可转债) 快照行情。
参数说明	ppSecurityID : 退订/订阅证券列表; nCount : 退订/订阅的证券个数。
返回说明	0 表示成功, -1 表示未连接, -3 未登录, -5 表示网络写失败。

(27) 退订、订阅深圳债券 (不含可转债) 逐笔成交行情

函数名称	Un/SubscribeBondTransction
函数原型	<code>virtual int Un/SubscribeBondTransction (char *ppSecurityID[],int nCount,TTORATstpExchangeIDType ExchangeID) = 0;</code>
函数描述	退订/订阅深圳债券类 (不含可转债) 逐笔成交行情。
参数说明	ppSecurityID : 退订/订阅证券列表; nCount : 退订/订阅的证券个数。
返回说明	0 表示成功, -1 表示未连接, -3 未登录, -5 表示网络写失败。

(28) 退订、订阅深圳债券（不含可转债）逐笔委托行情

函数名称	Un/SubscribeBondOrderDetail
函数原型	<code>virtual int Un/SubscribeBondOrderDetail (char *ppSecurityID[],int nCount,TTORATstpExchangeIDType ExchangeID) = 0;</code>
函数描述	退订/订阅深圳债券类（不含可转债）逐笔委托行情。
参数说明	ppSecurityID: 退订/订阅证券列表; nCount: 退订/订阅的证券个数。
返回说明	0 表示成功, -1 表示未连接, -3 未登录, -5 表示网络写失败。

回调接口类(CTORATstpLev2MdSpi)

(1) 连接成功

函数名称	OnFrontConnected
函数原型	<code>virtual void OnFrontConnected(){};</code>
函数描述	连接成功回调。 <ul style="list-style-type: none"> tcp 模式下, 当 api 与服务端 TCP 通信连接成功时, 该方法被调用; 组播模式下, 当 api 收到所有组播通道的数据时, 该方法被调用。
参数说明	无。

(2) 连接断开

函数名称	OnFrontDisconnected
函数原型	<code>virtual void OnFrontDisconnected(int nReason){};</code>
函数描述	连接断开回调。 <ul style="list-style-type: none"> tcp 模式下, 当 api 与服务端 TCP 通信连接断开时, 该方法被调用。连接断开后 api 会定时自动尝试重连服务端, 开发者可不用做任何处理。 组播模式下, 在超时时间内均为收到任何组播通道的数据时, 该方法被调用。组播通道内部会定时重新加入组播尝试收取数据。
参数说明	nReason: 断开原因, 可能取值如下: <ul style="list-style-type: none"> -1: 长时间未收到数据 -3: 连接已断开 -4: 网络读失败 -5: 网络写失败 -6: 订阅流错误 -7: 流序号错误 -8: 错误的心跳报文 -9: 错误的数据报文 -15: 网络读失败 -16: 网络写失败

(3) 错误应答

函数名称	OnRspError
函数原型	<code>virtual void OnRspError(CTORATstpRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};</code>
函数描述	错误应答。 目前未启用。
参数说明	pRspInfo : 处理结果; nRequestID : 请求编号, 同请求接口中的 nRequestID; bIsLast : 是否结束。

(4) 用户登录响应

函数名称	OnRspUserLogin
函数原型	<code>virtual void OnRspUserLogin(CTORATstpRspUserLoginField *pRspUserLogin, CTORATstpRspInfoField *pRspInfo, int nRequestID) {};</code>
函数描述	用户登录响应。
参数说明	pRspUserLogin : 响应业务域; pRspInfo : 处理结果; nRequestID : 请求编号, 同请求接口 ReqUserLogin 中的 nRequestID。

(5) 用户登出响应

函数名称	OnRspUserLogout
函数原型	<code>virtual void OnRspUserLogout(CTORATstpUserLogoutField *pUserLogout, CTORATstpRspInfoField *pRspInfo, int nRequestID) {};</code>
函数描述	用户登出响应。Tcp 模式下请求登出后, 服务端会直接断开与 api 间的连接, 该接口一般不会回调。
参数说明	pUserLogout : 响应业务域; pRspInfo : 处理结果; nRequestID : 请求编号, 同请求接口 ReqUserLogout 中的 nRequestID。

(6) 退订、订阅响应

函数名称	OnRspSub~
函数原型	<code>virtual void OnRspSub~ (CTORATstpSpecificSecurityField *pSpecificSecurity, CTORATstpRspInfoField *pRspInfo, int nRequestID, bool bIsLast) {};</code>
函数描述	订阅退订响应。
参数说明	pSpecificSecurity : 组播模式下 pSpecificSecurityID 揭示本次回调指向的订阅代码。 pRspInfo : 处理结果; nRequestID : 无意义。

(7) 上海/深圳非债券类、深圳可转债行情通知

函数名称	OnRtnMarketData
函数原型	<pre>virtual void OnRtnMarketData (CTORATstpLev2MarketDataField *pMarketData, const int FirstLevelBuyNum, const int FirstLevelBuyOrderVolumes[], const int FirstLevelSellNum, const int FirstLevelSellOrderVolumes[],) {};</pre>
函数描述	上海/深圳非债券类、深圳可转债行情通知。
参数说明	<p>pMarketData: 快照行情字段;</p> <p>FirstLevelBuyNum: 买一档价上揭示的委托数量队列大小 (<=50);</p> <p>FirstLevelBuyOrderVolumes: 买一档价上揭示的委托数量队列</p> <p>FirstLevelSellNum: 卖一档价上揭示的委托数量队列大小 (<=50);</p> <p>FirstLevelSellOrderVolumes: 卖一档价上揭示的委托数量队列;</p> <p>注: 需要根据 FirstLevelBuy/SellNum 的值去取 FirstLevelBuy/SellOrderVolume 里的委托数量, 不可越界访问或修改指针指向的内容。</p> <pre>for (int index=0;index< FirstLevelBuyNum;index++) { printf("%d",FirstLevelBuyOrderVolumes[Index]); }</pre>

(8) 指数行情通知

函数名称	OnRtnIndex
函数原型	<pre>virtual void OnRtnIndex (CTORATstpLev2IndexField *pIndex) {};</pre>
函数描述	沪深指数行情通知。
参数说明	pMarketData : 指数行情字段;

(9) 深圳非债券类、深圳可转债逐笔成交通知

函数名称	OnRtnTransaction
函数原型	<pre>virtual void OnRtnTransaction (CTORATstpLev2TransactionField *pTransaction) {};</pre>
函数描述	深圳非债券类、可转债逐笔成交通知。
参数说明	<p>pTransaction: 逐笔成交字段;</p> <p>注:</p> <ol style="list-style-type: none"> 1、以逐笔成交里的 MainSeq+BuyNo 到逐笔委托里查找 MainSeq+OrderNo 对应相等的那条逐笔委托即为买方委托记录; 以逐笔成交里的 MainSeq+SellNo 到逐笔委托里查找 MainSeq+OrderNo 对应相等的那条逐笔委托即为卖方委托记录。 2、深圳交易所逐笔成交里成交类别字段 ExecType 为 “2 (撤销)” 时表示撤单成交行情, BuyNo 和 SellNo 只有一个是非 0 值, 以(MainSeq, BuyNo/SellNo) 去查找对应的逐笔委托 (MainSeq, OrderNo), 找到的逐笔委托即为被撤单的委托。 3、丢包判断: 在同一主序号下, 深圳交易所对逐笔数据 (逐笔委托和逐笔成交一起编号) 统一从 1 开始编号, 一个交易日内连续单调递增, 通过校验逐笔数据里相同主序号 MainSeq 下的子序号 SubSeq 是否连续单调递增来判断是否丢包。

(10) 深圳非债券类、深圳可转债逐笔委托通知

函数名称	OnRtnOrderDetail
函数原型	<code>virtual void OnRtnOrderDetail (CTORATstpLev2OrdeDetailField *pOrderDetail) {};</code>
函数描述	深圳非债券类、可转债逐笔委托通知。
参数说明	<p>pOrderDetail: 逐笔委托字段;</p> <p>注:</p> <p>丢包判断: 在同一主序号下, 深圳交易所对逐笔数据 (逐笔委托和逐笔成交一起编号) 统一从 1 开始编号, 一个交易日内连续单调递增, 通过校验逐笔数据里相同主序号 MainSeq 下的子序号 SubSeq 是否连续单调递增来判断是否丢包。</p>

(11) 盘后定价行情通知

函数名称	OnRtnPHMarketData
函数原型	<code>virtual void OnRtnPHMarketData (CTORATstpLev2PHMarketDataField *pMarketData, const int FirstLevelBuyNum, const int FirstLevelBuyOrderVolumes[], const int FirstLevelSellNum, const int FirstLevelSellOrderVolumes[],) {};</code>
函数描述	上海/深圳盘后定价行情通知。
参数说明	<p>pMarketData: 快照行情字段;</p> <p>FirstLevelBuyNum: 买一档价上揭示的委托数量队列大小 (≤ 50);</p> <p>FirstLevelBuyOrderVolumes: 买一档价上揭示的委托数量队列</p> <p>FirstLevelSellNum: 卖一档价上揭示的委托数量队列大小 (≤ 50);</p> <p>FirstLevelSellOrderVolumes: 卖一档价上揭示的委托数量队列;</p> <p>注: 需要根据 FirstLevelBuy/SellNum 的值去取 FirstLevelBuy/SellOrderVolume 里的委托数量, 不可越界访问或修改指针指向的内容。</p>

(12) 盘后定价逐笔成交通知

函数名称	OnRtnPHTransaction
函数原型	<code>virtual void OnRtnPHTransaction (CTORATstpLev2PHTransactionField *pTransaction) {};</code>
函数描述	深圳非债券、可转债逐笔成交通知。
参数说明	pTransaction : 逐笔成交字段;

(13) 逐笔重传成交通知

函数名称	OnRtnResendTransaction
函数原型	<code>virtual void OnRtnResendTransaction (CTORATstpLev2ResendTransactionField *pTransaction) {};</code>
函数描述	逐笔重传成交通知。
参数说明	pTransaction : 逐笔成交字段;

(14) 逐笔重传委托通知

函数名称	OnRtnResendOrderDetail
函数原型	<code>virtual void OnRtnResendOrderDetail (CTORATstpLev2ResendOrderDetailField *pOrderDetail) {};</code>
函数描述	逐笔重传委托通知。
参数说明	pOrderDetail: 逐笔委托字段;

(15) 上海 XTS 债券行情通知

函数名称	OnRtnXTSMarketData
函数原型	<code>virtual void OnRtnXTSMarketData (CTORATstpLev2XTSMarketDataField *pMarketData, const int FirstLevelBuyNum, const int FirstLevelBuyOrderVolumes[], const int FirstLevelSellNum, const int FirstLevelSellOrderVolumes[],) {};</code>
函数描述	上海债券 XTS 行情通知。
参数说明	<p>pMarketData: 快照行情字段;</p> <p>FirstLevelBuyNum: 买一档价上揭示的委托数量队列大小 (≤ 50);</p> <p>FirstLevelBuyOrderVolumes: 买一档价上揭示的委托数量队列</p> <p>FirstLevelSellNum: 卖一档价上揭示的委托数量队列大小 (≤ 50);</p> <p>FirstLevelSellOrderVolumes: 卖一档价上揭示的委托数量队列;</p> <p>注: 需要根据 FirstLevelBuy/SellNum 的值去取 FirstLevelBuy/SellOrderVolume 里的委托数量, 不可越界访问或修改指针指向的内容。</p>

(16) 上海 XTS 债券逐笔通知

函数名称	OnRtnXTSTick		
函数原型	virtual void OnRtnXTSTick (CTORATstpLev2XTSTickField *pTick) {};		
函数描述	上海 XTS 债券逐笔通知。		
参数说明	pTick: 逐笔业务字段;		
	注: 上海交易所把 XTS 债券逐笔委托 (新订单、撤单)、逐笔成交、产品状态用一个接口 OnRtnXTSTick 推送, 需要通过 CTORATstpLev2XTSTickField.TickType 字段进行区分三种类别的消息, 具体可参考下表提取各消息对应的有效业务字段		
	TickType	说明	有效业务字段
	LTT_Add	新增委托订单	SecurityID TickTime MainSeq SubSeq Price Volume Side BuyNo/SellNo
	LTT_Delete	撤销委托订单	SecurityID TickTime MainSeq SubSeq Volume Side BuyNo/SellNo
	LTT_Trade	成交	SecurityID TickTime MainSeq SubSeq TradeBSFlag TradeMoney BuyNo SellNo Price Volume
LTT_Status	产品状态	SecurityID TickTime MainSeq SubSeq MDSecurityStat	

(17) 上海非债券类逐笔成交通知 (NGTS 逐笔合并消息启用前)

函数名称	OnRtnTransaction
函数原型	<code>virtual void OnRtnTransaction (CTORATstpLev2TransactionField *pTransaction) {};</code>
函数描述	NGTS 逐笔合并消息启用前, 上海非债券类逐笔成交通过此接口推送。
参数说明	<p>pTransaction: 逐笔成交字段;</p> <p>注:</p> <ol style="list-style-type: none"> 以逐笔成交里的 MainSeq+BuyNo 到逐笔委托里查找 MainSeq+OrderNo 对应相等的那条逐笔委托即为买方委托记录; 以逐笔成交里的 MainSeq+SellNo 到逐笔委托里查找 MainSeq+OrderNo 对应相等的那条逐笔委托即为卖方委托记录。 BizIndex 字段说明参见 (18) 逐笔成交丢包判断: 在同一主序号下, 上海交易所对逐笔成交数据从 1 开始编号, 一个交易日内连续单调递增, 通过校验逐笔成交数据里相同主序号 MainSeq 下的子序号 SubSeq 是否连续单调递增来判断逐笔成交是否丢包

(18) 上海非债券类逐笔委托通知 (NGTS 逐笔合并消息启用前)

函数名称	OnRtnOrderDetail
函数原型	<code>virtual void OnRtnOrderDetail (CTORATstpLev2OrderDetailField *pOrderDetail) {};</code>
函数描述	NGTS 逐笔合并消息启用前, 上海非债券类逐笔成交通过此接口推送。
参数说明	<p>pOrderDetail: 逐笔委托字段;</p> <p>注:</p> <ol style="list-style-type: none"> 上海交易所撤单委托通过逐笔委托接口推送, 当逐笔委托接口里 OrderStatus 为 'D' 时, 表示此逐笔委托是撤单委托, 此时委托序号 OrderNO 指向被撤销的委托。 BizIndex 字段是交易所对逐笔成交与逐笔委托的统一业务编号, 表明逐笔成交和逐笔委托业务发生的先后关系, 按主序号 MainSeq 连续从 1 单调递增。目前上海交易所不保证逐笔委托和逐笔成交两个消息推送的先后顺序, BizIndex 大的有可能先收到 接收到的逐笔委托和逐笔成交按照 BizIndex(同一个主序号 MainSeq)从小到大排序后, 有以下特点: <ul style="list-style-type: none"> 如果报单报入后立即全部成交, 则不推送逐笔委托, 只推送逐笔成交 如果报单报入后立即部分成交, 则先推送逐笔成交, 后推送逐笔委托, 且逐笔委托里的数量是剩余未成交数量。后续成交不会再推送逐笔委托 如果报单报入后不成交, 则推送逐笔委托, 且逐笔委托里的数量是委托数量。后续成交不会再推送逐笔委托 逐笔委托丢包判断: 在同一主序号下, 对逐笔委托数据从 1 开始编号, 一个交易日内连续单调递增, 通过校验逐笔委托数据里相同主序号 MainSeq 下的子序号 SubSeq 是否连续单调递增来判断逐笔委托是否丢包。

(19) 上海 NGTS 非债券类逐笔通知

函数名称	OnRtnNGTSTick
------	---------------

函数原型	virtual void OnRtnNGTSTick (CTORATstpLev2NGTSTickField *pTick) {};																
函数描述	上海 NGTS 非债券逐笔通知。上海交易所启用 NGTS 逐笔合并行情后，OnRtnOrder/OnRtnTransaction 将不再推送上海非债券类的逐笔行情，相应的订阅请求也需要切换成 Un/SubscribeNGTSTick																
参数说明	<p>pTick: 逐笔业务字段；</p> <p>注: 上海交易所把 NGTS 债券逐笔委托（新订单、撤单）、逐笔成交、产品状态用一个接口 OnRtnNGTSTick 推送，需要通过 CTORATstpLev2NGTSTickField.TickType 字段进行区分三种类别的消息，具体可参考下表提取各消息对应的有效业务字段</p> <table border="1"> <thead> <tr> <th>TickType</th><th>说明</th><th>有效业务字段</th></tr> </thead> <tbody> <tr> <td>LTT_Add</td><td>新增委托订单</td><td>SecurityID TickTime MainSeq SubSeq Price Volume Side BuyNo/SellNo TradeMoney (复用 TradeMoney 记录新增委托已成交数量)</td></tr> <tr> <td>LTT_Delete</td><td>撤销委托订单</td><td>SecurityID TickTime MainSeq SubSeq Volume Side BuyNo/SellNo</td></tr> <tr> <td>LTT_Trade</td><td>成交</td><td>SecurityID TickTime MainSeq SubSeq TradeBSFlag TradeMoney BuyNo SellNo Price Volume</td></tr> <tr> <td>LTT_Status</td><td>产品状态</td><td>SecurityID TickTime MainSeq SubSeq MDSecurityStat</td></tr> </tbody> </table>		TickType	说明	有效业务字段	LTT_Add	新增委托订单	SecurityID TickTime MainSeq SubSeq Price Volume Side BuyNo/SellNo TradeMoney (复用 TradeMoney 记录新增委托已成交数量)	LTT_Delete	撤销委托订单	SecurityID TickTime MainSeq SubSeq Volume Side BuyNo/SellNo	LTT_Trade	成交	SecurityID TickTime MainSeq SubSeq TradeBSFlag TradeMoney BuyNo SellNo Price Volume	LTT_Status	产品状态	SecurityID TickTime MainSeq SubSeq MDSecurityStat
TickType	说明	有效业务字段															
LTT_Add	新增委托订单	SecurityID TickTime MainSeq SubSeq Price Volume Side BuyNo/SellNo TradeMoney (复用 TradeMoney 记录新增委托已成交数量)															
LTT_Delete	撤销委托订单	SecurityID TickTime MainSeq SubSeq Volume Side BuyNo/SellNo															
LTT_Trade	成交	SecurityID TickTime MainSeq SubSeq TradeBSFlag TradeMoney BuyNo SellNo Price Volume															
LTT_Status	产品状态	SecurityID TickTime MainSeq SubSeq MDSecurityStat															

(20) 深圳债券（不含可转债）行情通知

函数名称	OnRtnBondMarketData
函数原型	virtual void OnRtnBondMarketData (CTORATstpLev2BondMarketDataField *pMarketData, const int FirstLevelBuyNum, const int FirstLevelBuyOrderVolumes[], const int FirstLevelSellNum, const int FirstLevelSellOrderVolumes[],) {};
函数描述	深圳债券（不含可转债）行情通知。
参数说明	<p>pMarketData: 快照行情字段；</p> <p>FirstLevelBuyNum: 买一档价上揭示的委托数量队列大小（<=50）；</p> <p>FirstLevelBuyOrderVolumes: 买一档价上揭示的委托数量队列</p> <p>FirstLevelSellNum: 卖一档价上揭示的委托数量队列大小（<=50）；</p> <p>FirstLevelSellOrderVolumes: 卖一档价上揭示的委托数量队列；</p> <p>注: 需要根据 FirstLevelBuy/SellNum 的值去取 FirstLevelBuy/SellOrderVolume 里的委托数量，不可越界访问或修改指针指向的内容。</p>

(21) 深圳债券（不含可转债）逐笔成交通知

函数名称	OnRtnBondTransaction
函数原型	virtual void OnRtnBondTransaction (CTORATstpLev2BondTransactionField *pTransaction) {};
函数描述	深圳非债券、可转债逐笔成交通知。

参数说明	<p>pTransaction: 逐笔成交字段;</p> <p>注: 成交类别字段 ExecType 为 “2 (撤销)” 时表示撤单成交行情, BuyNo 和 SellNo 只有一个是非 0 值, 以该非 0 序号去查找逐笔委托的 OrderNo, 找到的逐笔委托即为被撤单的委托</p>
------	---

(22) 深圳债券 (不含可转债) 逐笔委托通知

函数名称	OnRtnBondOrderDetail
函数原型	<code>virtual void OnRtnBondOrderDetail (CTORATstpLev2BondOrdeDetailField *pOrderDetail) {};</code>
函数描述	深圳非债券、可转债逐笔委托通知。
参数说明	pOrderDetail : 逐笔委托字段;