

XTP 支持上海 Level2 逐笔委托说明

一、QuoteAPI 关于 SHL2 更新说明

1. 接口区别

- (1) 弃用：绑核函数 SetUDPRecvThreadAffinity 和 SetUDPParseThreadAffinity
- (2) 新增：设定接收线程绑核 CPU 集接口 SetUDPRecvThreadAffinityArray，最多支持 4 个核，具体使用的核数视实际的 UDP 组播组和订阅情况而定
- (3) 新增：设定解析线程绑核 CPU 集 接口 SetUDPParseThreadAffinityArray，最多支持 4 个核，具体使用的核数视实际的 UDP 组播组和订阅情况而定

2. 结构体区别

- (1) XTPTBT 结构体

```

///逐笔数据信息
typedef struct XTPTickByTickStruct {
    ///交易所代码
    XTP_EXCHANGE_TYPE exchange_id;
    ///合约代码（不包含交易所信息），不带空格，以'\0'结尾
    char ticker[XTP_TICKER_LEN];
    /// SH: 业务序号（委托成交统一编号，同一个channel_no内连续
    /// SZ: 无意义
    int64_t seq;
    ///委托时间 or 成交时间
    int64_t data_time;
    ///委托 or 成交
    XTP_TBT_TYPE type;

    union {
        XTPTickByTickEntrust entrust;
        XTPTickByTickTrade trade;
    };
} XTPTBT;
  
```

新增红框字段 seq

注意：此 seq 区别于联合体内的 seq，channel_no 等同于联合体内的 channel_no

上交所的逐笔委托和逐笔成交，在同一个 channel 下，有不同的 seq：

- (a) XTPTickByTickStruct.seq，该 seq 表示逐笔委托和逐笔成交统一编号，对应原始消息中的 BizIndex 字段。该字段从 1 开始编号，按 channel 连续。由于上交所仅发布剩余逐笔委托，逐笔委托和逐笔成交的先后顺序，直接会影响原始报单数量的计算，因此使用时，请先按照

此 seq 字段对所有逐笔行情数据进行排序，然后再进行处理。（具体说明参见 3）

(b) XTPTickByTickEntrust.seq, 该 seq 表示逐笔委托的编号，从 1 开始，按 channel 连续。

(c) XTPTickByTickTrade.seq, 该 seq 表示逐笔成交的编号，从 1 开始，按 channel 连续。

(2) XTPTickByTickEntrust 结构体

```

///逐笔委托
struct XTPTickByTickEntrust {
    ///频道代码
    int32_t channel_no;
    ///SH: 委托序号(委托单独编号, 同一channel_no内连续)
    ///SZ: 委托序号(委托成交统一编号, 同一channel_no内连续)
    int64_t seq;
    ///委托价格
    double price;
    ///SH: 剩余委托数量(balance)
    ///SZ: 委托数量
    int64_t qty;
    ///SH: 'B':买; 'S':卖
    ///SZ: '1':买; '2':卖; 'G':借入; 'F':出借
    char side;
    ///SH: 'A': 增加; 'D': 删除
    ///SZ: 订单类别: '1': 市价; '2': 限价; 'U': 本方最优
    char ord type;
    ///SH: 原始订单号
    ///SZ: 无意义
    int64_t order_no;
};

```

新增红框字段 order_no，其余字段也相应增加上海市场的含义。

XTPTickByTickEntrust.order_no 表示原始订单号，与 XTPTickByTickTrade.bid_no 或 XTPTickByTickTrade.ask_no 对应。

3. 上交所逐笔委托数据的说明

上交所的逐笔委托数据中的 qty 数量不同于深交所逐笔委托数据中的 qty，它代表的是经过一次成交撮合后剩余数量。当某一个订单经过一次撮合全部成交的话，是不会有对应逐笔委托数据的，只会有逐笔成交数据。

(1) 假如订单原始报单数量是 1000，第一次撮合后，全成。此种情况下，对于这笔报单，只会有一笔 qty=1000 的逐笔成交。

(2) 假如订单原始报单数量是 1000，第一次撮合后，部成 200。此种情况下，对于这笔报单，会先有一笔 qty=200 的逐笔成交，后有一笔 qty=800 的逐笔委托。

(3) 假如订单原始报单数量是 1000，第一次撮合后，没有任何成交，后续撮合的时候部成 200。此种情况下，对于这笔报单，会先有一笔 qty=1000 的逐笔委托，后有一笔 qty=200 的逐笔成交。

(4) 假如订单原始报单数量是 1000，第一次撮合后，部成了 200，然后撤单 800。此种情况下，对于这笔报单，会先有一笔 qty=200 的逐笔成交，后有一笔 qty=800，XTPTickByTickEntrust.ord_type='A' 的逐笔委托，最后有一笔 qty=800，XTPTickByTickEntrust.ord_type='D' 的撤单逐笔委托。

由上面 4 种情况可以看出，不同的委托和成交顺序，代表的意思不同。因此处理数据前请先按照 XTPTickByTickStruct.seq 字段对所有逐笔行情数据进行排序，然后再进行处理。

二、QuoteAPI 使用需注意问题

1. 不更新升级，继续使用 2.2.30 之前的 API

(1) 连接 Level1 服务器

不受影响

(2) 连接 Level2 服务器

在上交所推送逐笔委托数据之后，OnTickByTick()回调函数中会推送上交所的逐笔委托数据，但是收到的数据会不完整，缺少新增的 **order_no** 字段。

若不需要处理 SHL2 逐笔委托消息，请在收到 SHL2 逐笔委托时，过滤掉该消息。过滤条件为：exchange_id=XTP_EXCHANGE_SH, type=XTP_TBT_ENTRUST

建议更新 API 至 2.2.30.3 以上。

2. 更新升级，使用 2.2.30.3 以上版本 API

(1) 连接 Level1 服务器

更新后，重新编译即可，无需修改任何代码。

(2) 连接 Level2 服务器

A. 绑核函数

如果使用了绑核函数，请注释掉以前的调用语句，或者替换为新的绑核函数调用语句。由于新版行情服务器可能分成单个或者多个组播组分发行情，具体使用几个核视订阅情况而定，分别最多可设定 4 个核。

B. 回调线程

回调函数线程的变化如下表所示：

回调函数	旧版 QuoteAPI 线程	新版 QuoteAPI 线程	
		服务器单个组播组	服务器多个组播组（以 4 个为例）
OnDepthMarket Data	第一个 MD 来自 TCP 线程	第一个 MD 来自 TCP 线程	第一个 MD 来自 TCP 线程
	后续 MD 来自	后续 MD 来自 UDP	非期权后续 MD 来自 UDP 线程 1

	UDP 线程	线程	
			期权后续 MD 来自 UDP 线程 2
OnTickByTick	UDP 线程	UDP 线程	UDP 线程 3
OnOrderBook	UDP 线程	UDP 线程	UDP 线程 4

每个组播组将会产生 2 个线程：1 个接收线程，1 个推送线程，最多使用 2 个 CPU 核。用户具体使用的核数情况受服务器设置的组播组个数以及用户自身订阅数据影响。

由于回调函数涉及多个线程，请用户在使用时注意线程间的同步情况。

三、测试环境说明

在上交所华京路机房和深交所东莞机房，分别启动测试服务器，供客户测试上交所逐笔行情和新版本 API。

测试服务器将对含有 SHL2 逐笔委托的历史数据进行回放，并使用组播对外推送行情，具体服务器接入地址和端口号，请以测试环境邮件为准。

测试服务器将轮流采用单个组播组和多个组播组（4 个）方式对外组播，具体组播安排情况，请以测试环境邮件为准。

四、实盘环境说明

实盘服务器目前开启了 2 个组播组，没有 OrderBook 数据。

如果绑核话，需要绑定 4 个核（一个 MD 的接收线程，一个 MD 的 UDP 解析线程，一个 TBT 的接收线程，一个 TBT 的 UDP 解析线程）

目前实盘回调函数线程如下：

回调函数	旧版 QuoteAPI 线程	新版 QuoteAPI 线程	
		服务器单个组播组	服务器多个组播组（以 2 个为例）
OnDepthMarket Data	第一个 MD 来自 TCP 线程	第一个 MD 来自 TCP 线程	第一个 MD 来自 TCP 线程
	后续 MD 来自 UDP 线程	后续 MD 来自 UDP 线程	后续 MD 来自 UDP 线程 1
OnTickByTick	UDP 线程	UDP 线程	UDP 线程 2
OnOrderBook	UDP 线程	UDP 线程	UDP 线程 3（实际无数据，不会回调）