

# XTP 上交所新债券 Level2 行情说明

针对沪深交易所近期推出的新债券行情，XTP API 做了适当调整，具体变化如下：

## 1. 快照行情变化

### 1.1 快照结构体

```
///行情
typedef struct XTPMarketDataStruct
{
    // 代码
    ///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];

    // 价格
    ///最新价
    double last_price;
    ///昨收盘
    double pre_close_price;
    ///今开盘
    double open_price;
    ///最高价
    double high_price;
    ///最低价
    double low_price;
    ///今收盘
    double close_price;

    // 期权数据
    ///昨日持仓量(张) (目前未填写)
    int64_t pre_total_long_position;
    ///持仓量(张)
    int64_t total_long_position;
    ///昨日结算价
    double pre_settl_price;
    ///今日结算价
    double settl_price;

    // 涨跌停
    ///涨停价
    double upper_limit_price;
    ///跌停价
    double lower_limit_price;
    ///预留
    double pre_delta;
    ///预留
    double curr_delta;

    /// 时间类，格式为YYYYMMDDHHMMSSsss
    int64_t data_time;

    // 量额数据
    ///数量，为总成交量（单位股，与交易所一致）
    int64_t qty;
    ///成交金额，为总成交金额（单位元，与交易所一致）
    double turnover;
    ///当日均价=(turnover/qty)
    double avg_price;

    // 买卖盘
    ///十档申买价
    double bid[10];
    ///十档申买价
    double ask[10];
    ///十档申买量
    int64_t bid_qty[10];
    ///十档申买量
    int64_t ask_qty[10];

    // 额外数据
    ///成交笔数
    int64_t trades_count;
    ///当前交易状态说明，参阅《XTP API常见问题.doc》文档
    char ticker_status[8];
    ///数据
    union {
        XTPMarketDataStockExData stk;
        XTPMarketDataOptionExData opt;
        XTPMarketDataBondExData bond;
    };
    ///决定了union是哪种数据类型（2.2.32版本以前所用字段，仅为了保持兼容，不建议使用该字段）
    XTP_MARKETDATA_TYPE data_type;
    ///决定了union是哪种数据类型（2.2.32版本新增字段，更详细区分了行情快照数据类型）
    XTP_MARKETDATA_TYPE_V2 data_type_v2;
} XTPMD;
```

## (1) 新增快照数据类型 XTP\_MARKETDATA\_TYPE\_V2

```
////////////////////////////////////  
///@brief XTP_MARKETDATA_TYPE_V2是行情快照数据类型，2.2.32版本新增字段  
////////////////////////////////////  
enum XTP_MARKETDATA_TYPE_V2 {  
    XTP_MARKETDATA_V2_INDEX = 0, // 指数  
    XTP_MARKETDATA_V2_OPTION = 1, // 期权  
    XTP_MARKETDATA_V2_ACTUAL = 2, // 现货(股票/基金等)  
    XTP_MARKETDATA_V2_BOND = 3, // 债券  
};
```

此字段在原有的快照数据类型基础上做了更详细的划分，以便于客户使用。

XTPMD.data\_type\_v2 = XTP\_MARKETDATA\_V2\_INDEX 时，XTPMD 的联合体无效；

XTPMD.data\_type\_v2 = XTP\_MARKETDATA\_V2\_OPTION 时，XTPMD.opt 有效；

XTPMD.data\_type\_v2 = XTP\_MARKETDATA\_V2\_ACTUAL 时，XTPMD.stk 有效；

XTPMD.data\_type\_v2 = XTP\_MARKETDATA\_V2\_BOND 时，XTPMD.bond 有效；

## (2) 新增债券专用结构体 XTPMarketDataBondExData

此结构体仅保留了债券快照的字段。

仅在 XTPMD.data\_type\_v2 = XTP\_MARKETDATA\_V2\_BOND 时生效。

```
///债券额外数据  
struct XTPMarketDataBondExData {  
    ///委托买入总量(SH,SZ)  
    int64_t total_bid_qty;  
    ///委托卖出总量(SH,SZ)  
    int64_t total_ask_qty;  
    ///加权平均委买价格(SH,SZ)  
    double ma_bid_price;  
    ///加权平均委卖价格(SH,SZ)  
    double ma_ask_price;  
    ///债券加权平均委买价格(SH)  
    double ma_bond_bid_price;  
    ///债券加权平均委卖价格(SH)  
    double ma_bond_ask_price;  
    ///债券到期收益率(SH)  
    double yield_to_maturity;  
    ///预留  
    double r0;  
    ///预留  
    int64_t r1;  
    ///预留  
    double r2;  
    ///预留  
    double r3;  
    ///预留  
    double r4;  
    ///预留  
    double r5;  
    ///预留  
    double r6;  
    ///预留  
    double r7;  
    ///预留  
    double r8;  
    ///买入撤单笔数(SH)  
    int32_t cancel_buy_count;  
    ///卖出撤单笔数(SH)  
    int32_t cancel_sell_count;  
    ///买入撤单数量(SH)  
    double cancel_buy_qty;  
    ///卖出撤单数量(SH)  
    double cancel_sell_qty;  
    ///买入撤单金额(SH)  
    double cancel_buy_money;  
    ///卖出撤单金额(SH)  
    double cancel_sell_money;  
    ///买入总笔数(SH)  
    int64_t total_buy_count;  
    ///卖出总笔数(SH)  
    int64_t total_sell_count;  
    ///买入委托成交最大等待时间(SH)  
    int32_t duration_after_buy;  
    ///卖出委托成交最大等待时间(SH)  
    int32_t duration_after_sell;  
    ///买方委托价位数(SH)  
    int32_t num_bid_orders;  
    ///卖方委托价位数(SH)  
    int32_t num_ask_orders;  
    ///时段(SHL2)，L1快照数据没有此字段，具体字段说明参阅《上海新债券Level2行情说明.doc》文档  
    char instrument_status[3];  
};
```

## 1.2 XTPMD.ticker\_status 含义变化

上交所新债券 Level2 行情系统上线后，对于沪市新债券 Level2 快照行情，XTPMD.ticker\_status 不再有意义，请使用 XTPMD.bond.instrument\_status 替代判断（参阅 1.3 章节说明）。

其他快照行情与以前保持不变，不受影响。

## 1.3 新增字段 instrument\_status 含义

上交所新债券 Level2 行情快照中，有个独有字段 **instrument\_status**，标识当前债券所处的交易状态，此字段对应含义如下：

字段	含义	时间段
ADD	产品未上市	
START	启动	8:45~9:15
OCALL	开市集合竞价	9:15~9:25
TRADE	连续自动撮合	债券现券：9:25~15:00
		债券回购：9:25~15:30
SUSP	停牌	
CLOSE	闭市，自动计算闭市价格	债券现券：15:00 先到
		债券回购：15:30 先到
ENDTR	交易结束	债券现券：15:00 后到
		债券回购：15:30 后到

交易状态说明如下：

（1）对于债券现券产品：

8:45~9:15 发 START 标志；

9:15~9:25 是开盘集合竞价就阶段，发 OCALL 标志；

9:25~15:00 是连续竞价阶段，发 TRADE 标志；

15:00 后，首先发 CLOSE 标志，随后发 ENDTR 标志。

（2）对于债券回购产品：

8:45~9:15 发 START 标志;  
9:15~9:25 是开盘集合竞价就阶段, 发 OCALL 标志;  
9:25~15:30 是连续竞价阶段, 发 TRADE 标志;  
15:30 后, 首先发 CLOSE 标志, 随后发 ENDTR 标志。

(3) Level2 没有全市场收盘标志, 只有每只证券的收盘标志。如果某证券当天停牌, 则在当天先后收到 SUSP、CLOSE 及 ENDTR 标志; 如果某证券连续停牌, 则可以收到该证券快照及 SUSP 标志, 且始终保持 SUSP 标志。

## 2. 逐笔行情变化

上交所的新债券逐笔行情使用 **801** 通道, 同时增加了状态订单, 与其他上交所逐笔行情相比, 债券的逐笔成交、逐笔委托、逐笔状态行情三者进行了**统一编号**。用户可以通过 channel\_no=801 过滤出债券行情, 进行单独处理。

### 2.1 逐笔结构体

```
///逐笔状态订单
struct XTPTickByTickStatus {
    ///频道代码
    int32_t channel_no;
    ///同一channel_no内连续
    int64_t seq;
    ///状态信息
    char flag[8];
};
```

```

///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
    char ticker[XTP_TICKER_LEN];
    /// SH: 业务序号（委托成交统一编号，同
    /// 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;
        XTPTickByTickStatus state;
    };
} XTPTBT;

```

逐笔行情结构体 XTPTBT 中新增了逐笔状态订单 XTPTBT.state

## 2.2 逐笔数据枚举类型

```

////////////////////////////////////
///@brief XTP_TBT_TYPE是一个逐笔回报类型
////////////////////////////////////
typedef enum XTP_TBT_TYPE {
    XTP_TBT_ENTRUST = 1,    ///<逐笔委托
    XTP_TBT_TRADE = 2,      ///<逐笔成交
    XTP_TBT_STATE = 3,      ///<逐笔状态订单，2.2.32版本新增字段，为上海新债券Level2行情中独有
}XTP_TBT_TYPE;

```

**XTP\_TBT\_TYPE** 新增了逐笔状态订单类型

## 2.3 XTPTBT.state.flag 字段含义

同本文 1.2 章节的 instrument\_status 含义

## 2.4 XTPTBT.trade.seq 和 XTPTBT.entrust.seq 字段含义变化

对于上海的新债券行情来说，XTPTBT.trade.seq 和 XTPTBT.entrust.seq 这 2 个字段不再各自排序，而是与 XTPTBT.state.seq 三者一起在同一个 channel\_no(801) 内统一连续排序。

上海的其他逐笔行情不变，仍然保持同一个 channel\_no 内 XTPTBT.trade.seq 和 XTPTBT.entrust.seq 各自独立排序，并在外层 XTPTBT.seq 统一连续排序。

深圳的逐笔行情不变，仅 channel\_no 有变化。

## 2.5 逐笔委托含义变化

之前上海债券的逐笔委托数据是经过一次撮合后的剩余数据，在新债券行情上线后：

- (1) 逐笔委托数据变为原始订单数据。
- (2) 集合竞价及停牌期间不发送逐笔委托数据，到集合竞价或停牌结束时统一发送期间的逐笔委托数据。
- (3) 涉及交易状态改变产生的集中撮合成交数据在产品状态订单之前发布。

## 3. 哪些用户需要修改程序

当交易所上线新债券行情后，用户需要针对不同情况做调整，具体如下：

### 3.1 Level1 用户

无影响，可以不做升级。

### 3.2 使用 2.2.32 以前版本的 Level2 用户

按照客户交易偏好，有如下情形：

#### (1) 不做债券交易（可转债、国债逆回购）或仅做深交所债券交易

可自行判断影响，选择升级与否。

上交所新债券 Level2 行情系统上线后，旧版的 api 将无法识别新增的上交所新债券行情，需用户自行判断当收到的逐笔数据 XTPTBT.type > XTP\_TBT\_TRADE 时，是否会影响已有的程序逻辑，如果有影响，可以按照 channel\_no==801 过滤掉上交所新债券逐笔行情。如果无影响，可不修改。

快照的 `XTPMD.ticker_status` 字段在沪市债券快照中不再有意义，需用户自行判断是否会有影响。如果有影响，且在不升级 API 的情况下，可以按照 `(XTPMD.r4 == 3) &&(XTPMD.exchange_id == XTP_EXCHANGE_SH)` 将沪市新债券快照行情过滤掉。

## (2) 有上交所债券交易

**必须升级**至 2.2.32 以上版本，并修改程序适配新的行情。

上交所此次行情升级不是兼容性升级，旧版的行情数据在新系统上线后不再提供，如果不升级 API，将会导致收不到正确的上交所新债券 Level2 行情数据。

## 3.3 从旧版升级至 2.2.32 以上版本的 Level2 用户

按照行情不同数据类型，有如下改动：

### (1) 深市债券行情

基本上没有变化，仅 `channel_no` 做了改变，对于深圳债券的程序处理逻辑可以不用变化，也可以根据 `XTPMD.data_type_v2` 以及 `XTPMD.bond` 来修改程序处理逻辑。

### (2) 沪市新债券 Level2 行情

变化很大，需要根据新的行情数据来修改程序处理逻辑。

✓ 逐笔行情：按照 `channel_no=801` 过滤出上交所的新债券逐笔行情，单独处理。注意：上交所新债券的逐笔委托、逐笔成交、逐笔状态三者有**统一连续编号**，不再各自编号。

✓ 快照行情：`XTPMD.ticker_status` 字段在沪市新债券 Level2 快照中不再有意义，请使用 `XTPMD.bond.instrument_status` 替代判断（参阅 1.3 章节说明）。