

中信期货 沪、深 Level-2 行情系统 (Citicsf_REMDS) 用户接入方法说明

中信期货信息技术中心

发布日期：202112

发布日期	版本	修改内容
201912	V1.0	使用中信期货接口接收行情
202003	V1.1	新增上证通、南方机房接入 ip 定义
202004	V2.0	新增期权数据结构
202007	V2.1	新增支持指定核运行上证/深证接收线程
202009	V3.1	新增上证，深证 Level1/期权行情说明
202010	V3.2	完善当日合约代码表查询说明
202103	V3.3	增加南方中心 Level1 组播地址定义
202105	V4.0	增加上证逐笔委托消息定义，扩展上证逐笔成交字段
202109	V4.1	增加金桥龙沪路机房上证 Level2 接入地址
202111	V4.2	增加金桥龙沪路机房深证 Level1 接入地址
202112	V4.3	增加上证新债券说明

目录

一、	概述：	4
1、	接入地址定义：	5
2、	行情发布频率及内容：	9
3、	组播设置：	9
二、	协议规范：	11
1、	数据类型	11
(1)	委托队列 OrderQueue	12
(2)	快照行情 SnapShot	12
(3)	逐笔委托 OrderDetail (深证)	15
(4)	逐笔成交 TradeDetail (沪、深)	15
(5)	逐笔委托 OrderDetail (上证, 202105 新增)	16
(6)	个股期权 FutureData (买卖 5 档)	17
2、	API 请求功能。	19
三、	常见问题 (FAQ)：	22

一、概述：

为了提高行情的速度及可靠性，目前证券行情采用 TCP+UDP 组播方式转发。行情系统支持通过 API 查询最新一笔快照。

TCP 主要实现登陆验证，数据丢失重建；UDP 主要实现行情数据实时组播。客户可通过我司提供的行情接收 API 示例程序进行二次开发接收行情。

根据高可用性，我司已部署一主一备两路行情，默认只启用**主行情**，若出现主行情无法工作备行情会启用。或主行情出现不稳定情况，备行情启用。

上证通机房支持：上证 Level-2，上证 Level-1/期权，深证 Level-1/期权行情。

移动二期机房支持：上证 Level-2，上证 Level-1/期权，深证 Level2，深证 Level-1/期权行情。

南方中心机房支持：深证 Level-2，深证 Level-1/期权行情。

金桥龙沪路机房支持：上证 Level-2，上证 Level-1/期权，深证 Level-1/期权。

注：

1. 深证 Level1/期权，上证 Level1/期权为两条单独的线路，与现有 Level2 行情相互独立；
2. 上证 Level2 行情中包含上证期权行情消息，我司已获得上证期权行情授权，客户可在上证 Level2 中获得期权行情，亦可从上证 Level1 线路中获得上证期权行情；
3. 深证 Level2 行情中不包含深证期权行情，客户只能从深证 Level1 行情中获得深证期权消息。

1、接入地址定义：

/*-----移动二期机房-----*/

接入上证客户本地 IP 网段：192.168.109.x

SSE Level-2 REMD 软解方案：

主/备	名称	地址
主	登陆 TCP 服务器	192.168.109.1:16211
	结构数据 UDP 接收地址	225.7.8.11:26211
备	登陆 TCP 服务器	192.168.109.2:16221
	结构数据 UDP 接收地址	225.7.8.21:26221

SSE Level2 FPGA 硬解方案：

名称	地址
SSE-Level2	230.192.30.106:23456

SSE Level-1/期权 Xele 行情

名称	地址
SSE-Level-1/期权	233.0.0.1:28929

SSE Level-1/期权 TIMdPub 行情

名称	地址
SSE-Level-1/期权	239.20.30.90.10908

接入深证客户本地 IP 网段：192.168.115.x

SZSE Level-2 REMD 软解方案：

主/备	名称	地址
主	深证 L2 登陆 TCP 服务器	192.168.115.1:16212
	深证 L2 结构数据 UDP 接收地址	225.7.8.12:26212
备	深证 L2 登陆 TCP 服务器	192.168.115.2:16222
	深证 L2 结构数据 UDP 接收地址	225.7.8.22:26222

SZSE Level-1/期权 REMD 软解方案：

名称	地址
登陆 TCP 服务器	192.168.115.1:37611
结构数据 UDP 接收地址	231.77.13.58:47611

/*-----上证通机房-----*/

接入上证客户本地 IP 网段：192.168.111.x

SSE Level-2 REMD 软解方案：

主/备	名称	地址
主	登陆 TCP 服务器	192.168.111.1:16211
	结构数据 UDP 接收地址	225.7.8.11:26211
备	登陆、数据重建 TCP 服务器	192.168.111.2:16221
	结构数据 UDP 接收地址	225.7.8.21:26221

SSE Level2 FPGA 硬解方案：

名称	地址
SSE-Level2	230.192.30.105:23456

SSE Level-1/期权 Xele 行情

名称	地址
SSE-Level2	233.0.0.1:28929

SSE Level-1/期权 TIMdPub 行情

名称	地址
SSE-Level2	239.20.30.90.10908

独立深证 Level1/期权行情（与 REMD Level2 程序通用）

名称	地址
登陆 TCP 服务器	192.168.111.1:37611
结构数据 UDP 接收地址	231.77.13.58:47611

/*-----南方中心机房-----*/

接入深证本地 IP 网段：192.170.114.x

SZSE Level-2 REMD 软解方案：

主/备	名称	地址
主	深证 L2 登陆 TCP 服务器	192.170.114.1:16212
	深证 L2 结构数据 UDP 接收地址	225.7.8.12:26212
备	深证 L2 登陆 TCP 服务器	192.170.114.2:16222
	深证 L2 结构数据 UDP 接收地址	225.7.8.22:26222

SZSE Level-1/期权 REMD 软解方案：

名称	地址
使用 API（与 Level2 程序通用）接收主地址 注：（仅支持已接入 Level2 线路客户）	Tcp 登录：192.170.114.1:37611 UDP 接收：231.77.13.58:47611
使用 API（与 Level2 程序通用）接收备地址 注：（仅支持已接入 Level2 线路客户）	Tcp 登录：192.170.114.2:37621 UDP 接收：231.77.13.68:47621

/*-----金桥龙沪路机房-----*/

接入上证 Level2 本地 IP 网段: 192.168.57.x

主/备	名称	地址
主	上证 L2 登陆 TCP 服务器	192.168.57.1:16211
	上证 L2/期权结构数据 UDP 接收地址	225.7.8.11:26211
备	上证 L2 登陆、数据重建 TCP 服务器	192.168.57.2:16221
	上证 L2/期权结构数据 UDP 接收地址	225.7.8.21:26221

SSE Level2 FPGA 硬解方案:

名称	地址
SSE-Level2	230.192.30.106:23456

接入沪、深 Level1 本地 IP 网段: 3.3.3.x

SSE Level-1/期权 Xele 行情:

名称	地址
SSE-Level-1/期权	233.0.0.1:28929

SSE Level-1/期权 TIMdPub 行情:

名称	地址
SSE-Level-1/期权	239.20.30.90:10908

SZSE Level-1/期权 REMD 软解方案 (与 REMD Level2 程序通用):

名称	地址
登陆 TCP 服务器	3.3.3.13:37611
结构数据 UDP 接收地址	231.77.13.58:47611

2、行情发布频率及内容：

2.1 上证 Level2:

快照行情（个股）：全量 1 分/次，增量（价位）1 笔/3s，包括买卖一上前 50 委托队列

快照行情（指数）：1 笔/5s，包括开高低收、最新价、成交量、换手率等

逐笔成交：实时

期权快照：价格档位为上下 5 档，2 笔/s。

2.2 深证 Level2:

快照行情：1 笔/3s

逐笔委托（报、撤单）：实时

逐笔成交：实时

全量快照（个股、指数）：2 笔/s

3、组播设置：

3.1 本地机器开启组播接收设置：

如 demo 程序无法接收到数据，请根据以下步骤一一排除。

(1) 网络是否已通：

ping 相应行情的 tcp 重传服务器 IP 地址，如不通请联系我司 IT 排查。如可以通请尝试 tcpdump 命令抓取对应网卡的数据包，查看是否有相应组播地址、端口的数据。

(2) 防火墙：

关闭本机防火墙（不同 Linux 系统命令有所不同）：

```
>>systemctl disable firewalld
```

(3) 查看过滤器过滤规则：

使用命令：>>sysctl -a | grep rp_filter，查看对应的网卡是否为 1，如为 1 需要修改为 0，表示不过滤。

```
>> sysctl -w net.ipv4.conf.网卡名.rp_filter=0
```

3.2 程序配置：

请根据接收 demo 程序中更改 **config.ini** 相应的配置。

配置 config.ini 说明：

```
#上海结构化数据配置
[shse_struct_data]
#1: 启用接收结构化数据, 0: 不启用
enable = 1
#指定绑核, -1为不绑核
cpu_core = -1

multicast_ip = 225.7.8.11
multicast_port = 26211
multicast_interface = 192.168.109.x

retransmit_ip = 192.168.109.1
retransmit_port = 16211

#用户名, 密码, 目前使用test作为用户登陆。
user = test
passwd = test

subscribed_market = SH
#之后不写则全订阅
subscribed_codes = 510050.sh;

#深圳结构化数据配置
[szse_struct_data]
enable = 1
#指定绑核, -1为不绑核
cpu_core = -1

multicast_ip = 225.7.8.12
multicast_port = 26212
multicast_interface = 192.168.115.x
```

```
retransmit_ip = 192.168.115.1
retransmit_port = 16212

user = test
passwd = test

subscribed_market = SZ
subscribed_codes = 000725.sz;000001.sz;399001.sz;
```

二、协议规范：

可参考 demo 程序中的 “REMDApiStruct.h” 文件。

1、数据类型

REMD 行情系统基于交易所行情接口规范文件规范进行开发，具体发布内容及时间段请参考“上证所LDDS系统Level-2行情接口说明书.pdf”，“深证所Binary行情数据接口规范.pdf”。

REMD 行情系统对非整型的数据做了转换：客户收到的所有价格数据（包括指数），Turnover(不包括指数)为原数据**放大 1 万倍**，即*10000 后的整型数据，真实数据小数点后 4 位有效。涉及到的所有 volume 不变。客户根据自身系统需求选择是否还原数据。

沪、深两市对交易状态做了不同的定义，故 REMD 系统做了统一转换，定义如下：

```
// 市场交易状态定义
enum REMD_MARKET_STATUS {
    Y = 'Y', // 'Y' 新产品. (New Product)
    P = 'P', // 'P' 休市，例如，午餐休市。无撮合市场内部信息披露。 (Market Break)
    B = 'B', // 'B' 停牌. (Suspension)
    C = 'C', // 'C' 收市. (Market Closed)
    Z = 'Z', // 'Z' 产品待删除. (Deleted Product)
    V = 'V', // 'V' 结束交易. (End of Trade)
    W = 'W', // 'W' 暂停，除了自有订单和交易的查询之外，任何交易活动都被禁止。 (Pause, Trades/Orders Query Only)
    X = 'X', // 'X' 停牌，( 'X' 和 'W' 的区别在于 'X' 时可以撤单). (Pause, Allow Trades/Orders Query and
```

Cancelling Order)

N = 'N', // 'N' 盘中集合竞价订单簿平衡. (Call auction Order Book Balance)
 L = 'L', // 'L' 盘中集合竞价PreOBB. (Call auction PreOBB)
 I = 'I', // 'I' 开市集合竞价. (Market-Open Call auction)
 J = 'J', // 'J' 收盘集合竞价. (Market-Close Call auction)
 U = 'U', // 'U' 盘后处理. (After-Market Process)
 F = 'F', // 'F' 盘前处理. (Pre-Market Process)
 E = 'E', // 'E' 启动. (Start)
 O = 'O', // 'O' 连续撮合. (Continous Matching/Trading)
 Q = 'Q', // 'Q' 连续交易和集合竞价的波动性中断. (Temporarily Circuit Breaker)
 NONE = 0

};

(1) 委托队列 OrderQueue

REMD 系统中使用委托队列数据结构 (OrderQueue) 表示一条委托队列的全部信息。委托队列数据紧随行情快照数据，即：系统在发送了行情快照数据 (SecuritySnapShot) 之后就会发送该证券的委托队列数据。数据结构如下：

字段名称	字段类型	字段说明
szCode	REMD_Code	证券代码
nMarket	REMD_Market	交易所 ID
nActionDay	int32_t	业务发生日(自然日)
nTradingDay	int32_t	交易日
nTime	int32_t	时间(HHMMSSmmm)
nSide	REMD_OrderSide	买卖方向
nPrice	int64_t	委托价格 (*10000)
nOrders	int32_t	订单数量
nABItems	int32_t	委托队列个数
nABVolume[200]	int32_t	委托明细

数据示例：

```

1588210116,416076112,nSvrTm:93825012,ConnId:7070,Seq:4029565,type:105,[15REMD_OrderQueue
Start]600231 93826000 nSide[S] nPrice[21700] nOrders[28] nABItems[28] nABVolume [1000]
[2600] [10000] [3200] [600] [4400] [1000] [13400] [5000] [15000] [900] [1200] [32200]
[1800] [3000] [11200] [1000] [500] [1000] [400] [2100] [3600] [500] [600] [200] [900] [100]
[300][15REMD_OrderQueue End]600231
  
```

(2) 快照行情 SnapShot

在 REMD 系统分为证券快照 (SecuritySnapShot) 和指数快照 (IndexSnapShot) 结构。其

中快照包括个股和债券。

以下表格对比了两系统的字段，其中 REMD 系统中使用 Security.XXX 表示行情快照 (SecuritySnapShot) 中字段，使用 Index.XXX 表示指数快照行情 (IndexSnapShot) 中的字段。

上海原始快照行情分为全档快照 (1 分钟/笔) 和增量快照 (3 秒/笔)，REMD 已经将增量行情进行了计算，客户获取到的即为当前的全量行情。

证券快照 SecuritySnapShot:

字段名称	字段类型	字段说明
szCode	char [16]	证券代码
nMarket	char	交易所 ID:上海为 1,深圳为 2
nActionDay	int32_t	业务发生日(自然日)
nTradingDay	int32_t	交易日
nTime	int32_t	时间(HHMMSSmmm)
nStatus	int32_t	当前证券状态
nPreClose	int64_t	前收盘价 (*10000)
nOpen	int64_t	开盘价 (*10000)
nHigh	int64_t	最高价 (*10000)
nLow	int64_t	最低价 (*10000)
nMatch	int64_t	最新价 (*10000)
nAskPrice[10]	int64_t	申卖价 (*10000)
nAskVol[10]	int64_t	申卖量
nBidPrice[10]	int64_t	申买价 (*10000)
nBidVol[10]	int64_t	申买量 (*1)
nNumTrades	int32_t	成交笔数
iVolume	int64_t	成交总量
iTurnover	int64_t	成交总金额 (*10000)
nTotalBidVol	int64_t	委托买入总量 (*1)
nTotalAskVol	int64_t	委托卖出总量 (*1)
nWeightedAvgBidPrice	int64_t	加权平均委买价格 (*10000)
nWeightedAvgAskPrice	int64_t	加权平均委卖价格 (*10000)
nIOPV	int32_t	IOPV 净值估值
nYieldToMaturity	int32_t	到期收益率
nHighLimited	int64_t	涨停价 (*10000)
nLowLimited	int64_t	跌停价 (*10000)
nSyl1	int32_t	市盈率 1
nSyl2	int32_t	市盈率 2
nSD1	int32_t	涨跌 1
nSD2	int32_t	涨跌 2

iAfterPrice	int64_t	盘后价格(科创板)
nAfterVolume	int64_t	盘后量(科创板)
iAfterTurnover	int64_t	盘后成交金额(科创板)
nAfterMatchItems	int64_t	盘后成交笔数(科创板)
nReserve	int64_t	预留位

数据示例:

个股

```
1588210116,459490100,nSvrTm:93825289,ConnId:7070,Seq:4030209,type:101,[21REMD_SecuritySnapshot Start]601666 93826000 nStatus[79] nPreClose[41500] nOpen[41600] nHigh[41900] nLow[41500] nMatch[41700] nAskPrice [41700] [41800] [41900] [42000] [42100] [42200] [42300] [42400] [42500] [42600] nAskVol [1100] [84400] [103500] [198700] [71600] [47500] [47300] [90900] [132000] [38300] nBidPrice [41600] [41500] [41400] [41300] [41200] [41100] [41000] [40900] [40800] [40700] nBidVol [102300] [247699] [215500] [175100] [205700] [143300] [58100] [15000] [212000] [5100] nNumTrades[279] iVolume[710401] iTurnover[29606851500] nTotalBidVol[1716799] nTotalAskVol[1645838] nWeightedAvgBidPrice[40910] nWeightedAvgAskPrice[43090] nIOPV[0] nYieldToMaturity[0] nHighLimited[45700] nLowLimited[37400] nSyl1[0] nSyl2[0] nSD2[0][21REMD_SecuritySnapshot End] 601666
```

债券

```
1595829592,860131065,nSvrTm:135952860,ConnId:7070,Seq:11849132,type:101,[21REMD_SecuritySnapshot Start]112457 135954000 nStatus[79] nPreClose[885990] nOpen[885970] nHigh[885970] nLow[885970] nMatch[885970] nAskPrice [883990] [884000] [884500] [884800] [885900] [0] [0] [0] [0] [0] [0] nAskVol [5000] [42500] [12000] [20000] [1400] [0] [0] [0] [0] [0] nBidPrice [880000] [806200] [0] [0] [0] [0] [0] [0] [0] [0] nBidVol [10] [2500] [0] [0] [0] [0] [0] [0] [0] [0] nNumTrades[1] iVolume[92500] iTurnover[81952225000] nTotalBidVol[0] nTotalAskVol[0] nWeightedAvgBidPrice[0] nWeightedAvgAskPrice[0] nIOPV[0] nYieldToMaturity[0] nHighLimited[0] nLowLimited[0] nSyl1[0] nSyl2[0] nSD2[-20][21REMD_SecuritySnapshot End] 112457 28=
```

指数快照: IndexSnapshot

字段名称	字段类型	字段说明
szCode	REMD_Code	证券代码
nMarket	REMD_Market	交易所 ID
nActionDay	int32_t	业务发生日(自然日)
nTradingDay	int32_t	交易日
nTime	int32_t	时间(HHMMSSmmm)
nPreCloseIndex	int64_t	前收盘指数 (*10000)
nOpenIndex	int64_t	今开盘指数 (*10000)
nHighIndex	int64_t	最高指数 (*10000)
nLowIndex	int64_t	最低指数 (*10000)
nLastIndex	int64_t	最新指数 (*10000)
iTotalVolume	int64_t	参与计算相应指数的交易数量 (*1)

iTurnover	int64_t	参与计算相应指数的成交金额 (*1)
-----------	---------	--------------------

数据示例:

1588210545,562624022,nSvrTm:94533127,ConnId:7070,Seq:17672472,type:100,[18REMD_Index
 SnapShot Start]399001 94533000 nOpenIndex[105755242] nHighIndex[106563488]
 nLowIndex[105755242] nLastIndex[106491463] iTotatVolume[2767778218]
 iTurnover[395205209255800] nPreCloseIndex[105141731][18REMD_IndexSnapShot End]399001

(3) 逐笔委托 OrderDetail (深证)

字段名称	字段类型	字段说明
szCode	REMD_Code	证券代码
nMarket	REMD_Market	交易所 ID
nActionDay	int32_t	委托日期(YYMMDD)
nTime	int32_t	委托时间(HHMMSSmmm)
nOrder	uint64_t	委托号
nPrice	int64_t	委托价格 (*10000)
nVolume	int32_t	委托数量 (*1)
nOrderSide	REMD_OrderSide	买卖方向
nOrderType	REMD_OrderType	订单类型

数据示例:

1588210533,21913280,nSvrTm:94521866,ConnId:7070,Seq:17517931,type:104,[16REMD_OrderD
 etail Start]000977 94523220 nOrder[566327653145179777] nPrice[420300] nVolume[28800]
 nOrderSide[B] nOrderType[2][16REMD_OrderDetail End]000977

(4) 逐笔成交 TradeDetail (沪、深)

字段名称	字段类型	字段说明
szCode	REMD_Code	证券代码
nMarket	REMD_Market	交易所 ID
nActionDay	int32_t	自然日
nTime	int32_t	成交时间(HHMMSSmmm)
nIndex	uint64_t	成交编号
nPrice	int64_t	成交价格 (*10000)
nVolume	int32_t	成交数量 (*1)
nTurnover	int64_t	成交金额 (*10000)
nAskOrder	uint64_t	卖方委托序号
nBidOrder	uint64_t	买方委托序号
nExecType	REMD_ExecType	成交类型
nBSFlag	REMD_BSFlag	内外盘类型(上证为 Buy 或 Sell, 深证该字段无意义)

nChannelNo	int32_t	通道号(202105 新增, 上证、深证均为 channel number)
nBizIndex	int64_t	业务号(202105 新增, 上证字段, 深证该字段为同一 channel number 下的递增编号(从 1 开始计数))。

数据示例:

```
nSvrTm:93826487,ConnId:7070,Seq:4036443,type:103,[16REMD_TradeDetail Start]600027
93827510 nIndex[281474977107271] nPrice[36600] nVolume[10000] nTurnover[366000000]
nAskOrder[281474977523312] nBidOrder[281474977477045] nExecType[T]
nBSFlag[S][16REMD_TradeDetail End]600027 20200430 09:28:37,643110160,103,600027
```

备注:

- [1] 逐笔委托和逐笔成交中的编号字段: 该字段的值是根据交易所数据结构中的 ChannelID 和 SequenceID 组合来的, 虽然每个 Channel 的 SequenceID 是连续的, 但是 Channel 之间不能保证 SequenceID 的唯一性, 因此 remd 系统里面组合了这两个值, 确保每个报单的 SequenceID 在系统内都是唯一的。具体来说, 就是把一个 8 字节整数的前 2 个字节赋值为 ChannelID, 后 6 个字节赋值为 SequenceID。

示例代码:

```
uint64_t lIndex = 281474977107271;
//低6个字节
uint64_t seqNo = lIndex & 0xFFFFFFFFFFFF;
//高2个字节
uint16_t chanNo = (lIndex >> (6 * 8)) & 0xFFFF;
```

(5) 逐笔委托 OrderDetail (上证, 202105 新增)

注: 与深证逐笔委托消息内容字段有所不同, 故分开定义。

字段名称	字段类型	字段说明
code	REMD_Code	证券代码
dataStatus	REMD_SHDataStatus	上交所逐笔委托数据状态(直接转发交易所状态值, 参考交易所文档)
channelNo	uint32_t	通道号
orderIndex	uint32_t	委托号
time	int32_t	委托时间(HHMMSSmmm)
orderNO	int64_t	原始订单号
price	int64_t	委托价格(*10000)
balance	int64_t	委托数量
bSFlag	REMD_BSFlag	内外盘类型
orderType	REMD_SHOrderType	逐笔委托类型
bizIndex	int64_t	业务索引

(6) 个股期权 OptionData(买卖 5 档)

字段名称	字段类型	字段说明
szCode	REMD_Code	证券代码
nMarket	REMD_Market	交易所 ID
nActionDay	int32_t	业务发生日(自然日)
nTradingDay	int32_t	交易日
nTime	int32_t	时间(HHMMSSmmm)
nStatus	int32_t	当前证券状态, REMD_TradingStatus
iPreOpenInterest	int64_t	昨持仓 (*1)
nPreClose	int64_t	昨收盘价 (*10000)
nPreSettlePrice	int64_t	昨结算 (*10000)
nOpen	int64_t	开盘价 (*10000)
nHigh	int64_t	最高价 (*10000)
nLow	int64_t	最低价 (*10000)
nMatch	int64_t	最新价 (*10000)
iVolume	int64_t	成交总量 (*1)
iTurnover	int64_t	成交总金额 (*10000)
iOpenInterest	int64_t	持仓总量 (*1)
nClose	int64_t	今收盘 (*10000)
nSettlePrice	int64_t	今结算 (*10000)
nHighLimited	int64_t	涨停价 (*10000)
nLowLimited	int64_t	跌停价 (*10000)
nPreDelta	int32_t	昨虚实度
nCurrDelta	int32_t	今虚实度
nAskPrice[5]	int64_t	申卖价 (*10000)
nAskVol[5]	int64_t	申卖量 (*1)
nBidPrice[5]	int64_t	申买价 (*10000)
nBidVol[5]	int64_t	申买量 (*1)
nAuctionPrice	int64_t	波动性中断参考价 (*10000)
nAuctionQty	int64_t	波动性中断集合竞价虚拟匹配量

数据示例:

```

nSvrTm:93428880,ConnId:7070,Seq:2770180,type:102,[19REMD_FutureSnapshot Start]10002469
93430181 nStatus[79] iPreOpenInterest[5201] nPreClose[3571] nPreSettlePrice[3790]
nOpen[3528] nHigh[3812] nLow[3528] nMatch[3812] iVolume[78] iTurnover[2944630000]
iOpenInterest[5251] nClose[0] nSettlePrice[0] nHighLimited[6619] nLowLimited[961]
nPreDelta[0] nCurrDelta[0] nAskPrice [3821] [3823] [3828] [3833] [3838] nAskVol [20]
[30] [20] [1] [2] nBidPrice [3802] [3794] [3793] [3778] [3768] nBidVol [20] [1] [10] [1]
[1] nAuctionPrice[0] nAuctionQty[0][19REMD_FutureSnapshot End]10002469
  
```

(7) 上证新债券(2021/12/20 上线)

交易时间:

08:45-16:00: 市场总览数据、行情快照数据、逐笔通道序号数据

09:15-15:35: 逐笔委托及成交数据

债券行情快照数据（统一在 REMD 的快照消息中接收，不再定义新消息类型）

行情快照数据 UA3802 提供了债券现券与回购的基本行情信息，十档行情深度信息及买卖一档上的前 50 笔委托数量（由于数据量较大且大多数客户未使用，我司不支持）等。行情快照数据每 3 秒发送一次快照。

逐笔委托及成交数据（统一在 REMD 的逐笔委托和逐笔成交消息中接收）

目前债券逐笔委托及成交数据仅有一个通道（Channel），通道号为 801，该通道内的索引号（TickIndex）都应该连续，并从 1 开始递增。

注：交易所原始债券消息中无 BizIndex 字段，为统一上证 Level2 的消息逻辑，REMD 服务端将索引号（TickIndex）填入 BizIndex。

实时状态消息（使用上证逐笔委托消息发送）

由于新债券的实时状态消息中的通道号(channel)和索引号(index)与逐笔委托和成交统一编号，为方便判断债券同一通道的消息连续性，REMD 将发送债券实时状态消息。并且只有以下字段有意义：通道号，索引号，状态。其他值均为-1。

时间戳说明：

新债券交易所时间新债逐笔和快照都是毫秒(ms)单位，旧版逐笔为 10 毫秒(ms)，旧版快照是秒(s)。我司 REMD 系统内部已统一处理，格式为：

143025006 表示 14:30:25.006

数据示例(测试数据)：

《债券快照》

ID:204001,Tm:154453844,Match:11250,Status:15,PrePx:24800,Open:18500,High:300000,Low:9000,NumTrd:19652,Vol:12465100,Tnov:124651000000000,TtlBVol:141300,TtlAVol:25539200,WtAvgBPx:10460,WtAvgAPx:29530,IOPV:0,YdTM:0,HLmt:0,LLmt:0,Syl1:0,Syl2:0,SD1:0,SD2:0,
 AskP:11200,11250,11300,11500,11800,11900,11950,12000,12100,12150,
 AskV:100,42900,600,100,300,100,100,10209400,1000,4100,
 BidP:11050,10100,10050,10000,9000,1000,0,0,0,0,
 BidV:67500,10000,38000,19600,5900,300,0,0,0,0,
 《债券逐笔委托》
 ID:110083,Tm:93000014,Chnl:801,Idx:238680,Px:1169700,Vol:5,Stat:0,^@,odrNo:150487,BS:B,Type:A,bizIdx:238680,
 ID:113017,Tm:93000014,Chnl:801,Idx:238681,Px:1133000,Vol:120,Stat:0,^@,odrNo:150488,BS:B,Type:A,bizIdx:238681,
 《债券逐笔成交》
 TD,ID:204001,Mkt:1,Tm:93000014,Seq:238684,Chnl:801,AOrd:150490,BOrd:131982,Px:36000,Tnov:1000000000,Vol:100,ExeType:T,BS:S,Chnl:801,BizIdx:238684,
 《债券逐笔委托, 增加委托订单》
 ID:204001,Tm:110426953,Chnl:801,Idx:5132198,Px:36850,Vol:1200,Stat:0,^@,odrNum:3866080,BS:S,Type:A,bizIdx:5132198
 《债券逐笔委托, 删除委托订单》
 ID:110075,Tm:110426968,Chnl:801,Idx:5132200,Px:1334500,Vol:1,Stat:0,^@,odrNum:3816252,BS:B,Type:D,bizIdx:5132200
 《债券逐笔状态消息》
 ID:163316,Tm:93000146,Chnl:801,Idx:244179,Px:-1,Vol:-1,Stat:88,X,odrNo:-1,BS:N,Type:N,bizIdx:244179,

2、API 请求功能。

通过API登陆成功后, 可通过以下函数进行请求服务:

```

// 获取到的代码表, 需要调用RMDS_Free来释放内存
CTCSF_RMDS_API RMDS_ERROR RMDS_GetCodeTable(RMDSHandle, const char* szMarket, RMDS_CODEINFO**
pCodeList, uint32_t* nCount);
  
```

示例代码:

```

    REMD_CODEINFO* pCodeList = nullptr;
    uint32_t nCount = 0;
    auto err = REMD_GetCodeTable(h, szMarket, &pCodeList, &nCount);
    if (err == REMD_ERR_SUCCESS) {
        for (uint32_t i = 0; i < nCount; ++i) {
            auto& code_info_ref = pCodeList[i];
            output(stdout, &code_info_ref);
        }
    }
  
```

```
REMD_Free(pCodeList);
```

运行结果:

```
13REMD_CODEINFO, szCode:157711, 1, szRemdCode:157711.SH, szCNName:19山东22, 48,
13REMD_CODEINFO, szCode:155407, 1, szRemdCode:155407.SH, szCNName:19恒大02, 49,
13REMD_CODEINFO, szCode:157967, 1, szRemdCode:157967.SH, szCNName:19云南22, 48,
13REMD_CODEINFO, szCode:688981, 1, szRemdCode:688981.SH, szCNName:中芯国际, 25,
13REMD_CODEINFO, szCode:157710, 1, szRemdCode:157710.SH, szCNName:19山东21, 48,
13REMD_CODEINFO, szCode:155406, 1, szRemdCode:155406.SH, szCNName:19恒大01, 49,
13REMD_CODEINFO, szCode:157966, 1, szRemdCode:157966.SH, szCNName:19云南21, 48,
13REMD_CODEINFO, szCode:157700, 1, szRemdCode:157700.SH, szCNName:19四川75, 48,
13REMD_CODEINFO, szCode:157956, 1, szRemdCode:157956.SH, szCNName:19吉林20, 48,
13REMD_CODEINFO, szCode:157696, 1, szRemdCode:157696.SH, szCNName:19四川71, 48,
```

// 获取指定代码（格式为RMDS_RMDSCode）的详细期权信息

// pOption指针内容将被修改为对应的期权信息

```
CTCSF_RMDS_API RMDS_ERROR RMDS_GetOptionCodeInfo(RMDSHandle, const char* szRMDSCode,
RMDS_OPTION_CODEINFO* pOption);
```

// 更新订阅信息

// szCodeList为待{设置、增加、删除}的证券列表

// 若订阅更新为RMDS_ALL, 则忽略szCodeList

```
CTCSF_RMDS_API RMDS_ERROR RMDS_UpdateSubscription(RMDSHandle, RMDS_SUBSCRIPTION_UPDATE, const char*
szRMDSCodeList);
```

// 版本号

```
CTCSF_RMDS_API const char* RMDS_Version();
```

////////////////////////////////////

// 以下为主动获取行情接口

// API会主动向服务器请求最新的行情信息, 函数将被阻塞, 直到数据或错误返回

// 因此不可以在任何回调函数内使用, 应当在其他线程中调用

//获取当前最新行情, szMarket (SH, SZ), 需调用RMDS_Free(void*)释放内存

```
CTCSF_RMDS_API RMDS_ERROR RMDS_GetLastMarketData(RMDSHandle handle, const char* szMarket,
RMDS_SecuritySnapShot** pMarketData, int* nCount);
```

//获取当前最新指数, szMarket (SH, SZ), 需调用RMDS_Free(void*)释放内存

```
CTCSF_RMDS_API RMDS_ERROR RMDS_GetLastIndexData(RMDSHandle handle, const char* szMarket,
RMDS_IndexSnapShot** pIndexData, int* nCount);
```

//获取当前最新期货/期权, szMarket (SH, SZ), 需调用RMDS_Free(void*)释放内存

```
CTCSF_RMDS_API RMDS_ERROR RMDS_GetLastFutureData(RMDSHandle handle, const char* szMarket,
RMDS_FutureSnapShot** pFutureData, int* nCount);
```

////////////////////////////////////

// 以下是行情重建接口

```
// 请求重建结构化数据，序列号范围[start, end)
// 收到的数据会在pfDataHandler中得到
CTCSF_RMDS_API RMDS_ERROR RMDS_RebuildStructData(RMDSHandle, SeqType start, SeqType end,
RMDS_DataHandler pfDataHandler);
```

三、常见问题（FAQ）：

1. 上证指数消息中 DataTimeStamp 和 TradeTime 两个时间分别代表什么？

答：按照上证的解释，DataTimeStamp 代表最新订单时间，TradeTime 为成交时间。TradeTime 成交时间较 DataTimeStamp 最新订单时间更为精确。但两者时间与实际时间偏差较大，参考意义不大。

2. 行情里对于涨跌停的情况，是否有特殊的处理或者判断方式？

答：没有特殊处理，交易所发什么我们发什么。涨停时卖盘应该是空的。

3. 关于深圳市场行情，是否如果合约是指数，就没有 latest_price 而只有 current_index？

答：对于合约是 s 指数的情况，行情中只发布当前指数、昨日收盘指数、开盘指数、最高指数、最低指数。

4. 客户目前通过组播接收行情，有没有方法判断是否丢包，如何解决？

答：在 REMD_MSG 数据结构中有一个 nSequence，每一笔行情过来自动加 1，可以判断其是否连续。如果仅订阅部分市场或代码，序列号会不连续。发生丢包可通过 [GetLastMarketData](#)（快照）或 [GetLastIndexData](#)（指数）来获取最新的全量快照。

5. 上交所行情的整个结构是怎么样的？

答：上交所是通过我司本地服务器部署 VDE 后连接到交易所的行情服务器上，VDE 通过 FIX 协议负责向交易所获取数据到本地，行情程序通过 TCP 连接 VDE 获取数据。VDE 的延时大概在 20us 以上，根据数据包大小延时而异。客户无法直接与 VDE 对接。

6. 深交所行情的整个结构是怎么样的？

答：深交所是通过我司本地服务器部署 MDGW 后连接到交易所的行情服务器上，MDGW 通过与交易所通信获取 Binary 流数据到本地，行情程序通过 TCP 连接 MDGW 获取数据。MDGW 的延时大概在 20us 以上，根据数据包大小延时而异。客户无法直接与 MDGW 对接。

7. 深交所行情逐笔委托 OrderDetail 的 nOrder，逐笔成交 TradeDetail 的 nAskOrder 或 nBidOrder 是很大的数值，有何意义？

答：OrderDetail.nOrder 是由 ChannelNo 频道代码+ApplSeqNum 消息记录号(从 1 开始计数)；

TradeDetail.nAskOrder/nBidOrder:ChannelNo 频道代码+买方委托索引(从 1 开始计数，0 表示无对应委托)/卖方委托索引(从 1 开始计数，0 表示无对应委托)。

8. Remd 中消息来之后触发回调函数是在主线程中运行，还是启用了新的线程？REMD.Wait 会把当前线程阻塞,如果两个回调函数和这个调用 wait 的线程在同一个线程中是否会阻塞，wait 函数什么时候会返回？

答：open 调用之后会创建新的后台线程用来接收数据，回调就是在后台线程中调用的，wait 函数只是调用了后台线程的 join，可以在主线程中调用，但是不能在回调函数中调用，wait 的返回是在后台线程结束后返回。

9. 深交所中午休市期间是否会一直推送行情？

答：快照会定时发，逐笔会有心跳。

10. 上证期权行情如何接收？

答：目前上证 Level2 系统中支持接收期权行情，消息类型为 REMD_DATA_FUTURE。

11. 上证通机房内是否可收到中金所行情？

答：我司已将中金的行情引入上证通内，需要单独接入对应的 Level1 行情交换机，组播地址/端口：**227.187.91.32/54332**。数据结构定义如下：

```
#pragma pack(push, 1)
///EESQuoteDateType 是一个日期类型
typedef char EESQuoteDateType[9];
///EESQuoteInstrumentIDType 是一个合约代码类型
typedef char EESQuoteInstrumentIDType[31];
///EESQuoteExchangeIDType 是一个交易所代码类型
/// 取值范围
/// 上交所 SHH
/// 深交所 SHZ
/// 上期所 SHFE
/// 中金所 CFFEX
/// 大商所 DCE
/// 郑商所 CZCE
/// 黄金 SGE
typedef char EESQuoteExchangeIDType[9];
///EESQuotePriceType 是一个价格类型
typedef double EESQuotePriceType;
///EESQuoteLargeVolumeType 是一个大额数量类型
typedef double EESQuoteLargeVolumeType;
///EESQuoteVolumeType 是一个数量类型
typedef int EESQuoteVolumeType;
///EESQuoteMoneyType 是一个资金类型
typedef double EESQuoteMoneyType;
///EESQuoteRatioType 是一个比率类型
typedef double EESQuoteRatioType;
///EESQuoteExchangeInstIDType 是一个合约在交易所的代码类型
typedef char EESQuoteExchangeInstIDType[31];
///EESQuoteMillisecType 是一个时间（毫秒）类型
typedef int EESQuoteMillisecType;
///TFtdcTimeType 是一个时间类型
typedef char EESQuoteTimeType[9];

/// \brief EES 行情结构
struct EESMarketDepthQuoteData
{
    EESQuoteDateType      TradingDay;    ///<交易日
    EESQuoteInstrumentIDType InstrumentID; ///<合约代码
    EESQuoteExchangeIDType ExchangeID;   ///<交易所代码
    EESQuoteExchangeInstIDType ExchangeInstID; ///<合约在交易所的代码
    EESQuotePriceType      LastPrice;     ///<最新价
```


EESQuotePriceType	PreSettlementPrice; /// 上次结算价
EESQuotePriceType	PreClosePrice; /// 昨收盘
EESQuoteLargeVolumeType	PreOpenInterest; /// 昨持仓量
EESQuotePriceType	OpenPrice; /// 今开盘
EESQuotePriceType	HighestPrice; /// 最高价
EESQuotePriceType	LowestPrice; /// 最低价
EESQuoteVolumeType	Volume; /// 数量
EESQuoteMoneyType	Turnover; /// 成交金额
EESQuoteLargeVolumeType	OpenInterest; /// 持仓量
EESQuotePriceType	ClosePrice; /// 今收盘
EESQuotePriceType	SettlementPrice; /// 本次结算价
EESQuotePriceType	UpperLimitPrice; /// 涨停板价
EESQuotePriceType	LowerLimitPrice; /// 跌停板价
EESQuoteRatioType	PreDelta; /// 昨虚实度
EESQuoteRatioType	CurrDelta; /// 今虚实度
EESQuoteTimeType	UpdateTime; /// 最后修改时间
EESQuoteMillisecType	UpdateMillisec; /// 最后修改毫秒
EESQuotePriceType	BidPrice1; /// 申买价一
EESQuoteVolumeType	BidVolume1; /// 申买量一
EESQuotePriceType	AskPrice1; /// 申卖价一
EESQuoteVolumeType	AskVolume1; /// 申卖量一
EESQuotePriceType	BidPrice2; /// 申买价二
EESQuoteVolumeType	BidVolume2; /// 申买量二
EESQuotePriceType	AskPrice2; /// 申卖价二
EESQuoteVolumeType	AskVolume2; /// 申卖量二
EESQuotePriceType	BidPrice3; /// 申买价三
EESQuoteVolumeType	BidVolume3; /// 申买量三
EESQuotePriceType	AskPrice3; /// 申卖价三
EESQuoteVolumeType	AskVolume3; /// 申卖量三
EESQuotePriceType	BidPrice4; /// 申买价四
EESQuoteVolumeType	BidVolume4; /// 申买量四
EESQuotePriceType	AskPrice4; /// 申卖价四
EESQuoteVolumeType	AskVolume4; /// 申卖量四
EESQuotePriceType	BidPrice5; /// 申买价五
EESQuoteVolumeType	BidVolume5; /// 申买量五
EESQuotePriceType	AskPrice5; /// 申卖价五
EESQuoteVolumeType	AskVolume5; /// 申卖量五
EESQuotePriceType	AveragePrice; /// 当日均价
};	
#pragma pack(pop)	

12. 上证逐笔委托 order type 为 D, 撤单的时候, balance 表示的是撤单数量还是订单被撮合后剩余的委托数量?

答：订单被撮合后剩余的委托数量。

13. 上证逐笔委托相关问题 1:

某支股票：

逐笔成交:bid:1896130,ask:1893174

逐笔成交:bid:1896130,ask:1896149 会出现这种情况吗？ 一笔买单与前面的卖单成交，又与后面的卖单成交，中间还不发逐笔委托确认的情况吗？

答：不会出现，因为与前面卖单成交后，会有剩余委托产生，就这时候会发委托。

14. 上证逐笔委托的交易所时间戳和本地时间戳时间差过大（例如本机在 9:30 分收到交易所时间戳为 09:15:30 的逐笔委托），是否正常？

答：上交所自 2021 年 05 月逐笔委托上线以来，对逐笔委托发送采取”限流”方式。开盘集合竞价期间的委托数据等结束后统一发。如开盘后发现延时较大的逐笔委托数据可能是由于交易所限流。

15. 深证某笔逐笔成交后很长时间才收到逐笔委托撤单事件（例如某只股票在 9 点 31 的盘口价格是 16.7 左右，报价为 16.0 的卖价，10 分钟才收到“撤销”的逐笔委托），是否正常？

答：创业板的价格笼子机制（2020 年），如报的买卖价不在价格笼子范围内，交易所不会进行撮合，而是会把这个单子挂起来，用户不撤单需要等价格变动进入价格笼子才会处理。

16. 某个通道行情的 bizIndex 不连续，是否可以判断为丢包？

答：每个通道(Channel) 内逐笔成交与逐笔委托消息合并后生成的连续编号(BizIndex)，并从 1 开始递增。请确认是否统计了所有逐笔成交与委托。

例如：

OrderDetail, ID:204001, Time:15112711.0, BizIndex:15040881
OrderDetail, ID:204001, Time:15112711.0, BizIndex:15040882
OrderDetail, ID:204001, Time:15112717.0, BizIndex:15040884
OrderDetail, ID:204001, Time:15112717.0, BizIndex:15040886

OrderDetail, ID:204001, Time:15112720.0, BizIndex:15040888
TradeDetail, ID:204001, Time:15112708.0, BizIndex:15040880
TradeDetail, ID:204001, Time:15112711.0, BizIndex:15040883
TradeDetail, ID:204001, Time:15112717.0, BizIndex:15040885
TradeDetail, ID:204001, Time:15112720.0, BizIndex:15040887
TradeDetail, ID:204001, Time:15112723.0, BizIndex:15040892
OrderDetail, ID:204001, Time:15112720.0, BizIndex:15040889
OrderDetail, ID:204007, Time:15112720.0, BizIndex:15040890

17. 深证 Level2 中逐笔委托里面市价单的 price, 有 0, 1, 涨停价, 跌停价, 这些是怎么定的, 按市价单类型分的吗?

答: 逐笔委托市价单时价格字段无意义。

18. 行情快照在 09:15~09:25 这个时间段, 收到的买一手数和卖一手数总是相等是否正常?

答: 正常, 集合竞价期间揭示的虚拟成交价和虚拟成交量, 09:25 开始发送集合竞价结果。

19. 沪、深两市中发现某些逐笔成交信息中的 ask/bid order 索引, 找不到对应的逐笔委托, 是否正常?

答: 正常, 通常来说 ask/bid 逐笔委托撮合后生成逐笔成交, 逐笔成交能够对应唯一 ask/bid 逐笔委托。但实际测试发现存在逐笔成交找不到对应的逐笔委托。存在某些协议交易、询价等可能没有委托, 只有成交。

20. 持续更新中

21.