

1 REM 的 API 大版本

因为实际生产环境，目前是 1.0 与 2.0 并存的状态，后续会全部更新到 2.0 版本。

1.0 与 2.0 主要的区别：

第一、ConnServer()函数，最后两个参数const char* qrySvrAddr, int nQrySvrPort在1.0版本是有缺省值的（缺省查询走交易通道），2.0版本不再提供缺省值，并且强制要求查询不能从交易通道走。

第二、下单结构体EES_EnterOrderField中ForceCloseReason移除了。

2 windows/linux 库

REM 目前 windows 下提供 32 位的库，分 release 与 debug 版本；Linux 下为 64 位的库，不区分 release 与 debug，只有发布一个版本。

3 多线程安全

REM 的 API 是非线程安全的，多线程并发操作需要用户自己加锁。

4 用户与账户

（1）概念阐述

1) 账户（account）：即资金账户，是个人或机构客户在期货公司开户产生的；

2) 用户：用以客户端登陆 REM 服务器使用，在 REM 系统中，用户与账户是绑定的关系，通常会为每个账户（account）绑定多个用户，以实现多点登陆的需求。

（2）用户 ID 与登陆 ID 的区别

同一用户有两个属性，一是登陆ID（login id）、二用户ID（user id），login id与user id在REM 柜台是一一对应的关系，前者为字符串、后者为整型数字。在客户端发起登陆请求调用 UserLogon函数的时候，LogonID为其中的一个输入参数，然后OnUserLogon回调会返回UserID。同一Account不同UserID下的单子，其他UserID也会收到，OnOrderAccept会返回下单的UserID，可以通过它来判断是由当前用户还是其他用户下出去的单子。

（3）登陆 ID 不是子账号

下单函数结构体EES_EnterOrderField中m_Account要填资金账户（Account），不能填登陆ID，很多新用户会把LogonID当成子账户，造成下单被REM拒绝的情况。

（4）REM 系统每个用户只允许单点登陆

重复登陆会踢掉原来的 session，这个也是常见的用户反馈下单收不到 REM 回报的原因之一，实际上是因为登陆已经断开造成的。所以策略程序上面最好实现断开后重连的功能。

5 无自动重连机制

REM的API没有断开重连的机制，需要用户自己编程实现。注意下不能在 OnDisConnection() 回调里面直接重新发起ConnectServer()请求，否则程序会crash掉。建议在OnDisConnection() 记录下状态，在主线程中判断状态及发起重连请求。重连成功以后，需要重新登陆。

6 无法连接

REM因为1.0与2.0版本不兼容，很多用户在生产线上都遇到过因为用错版本造成无法登陆的问题，此时无法正常收到OnConnection()就socket连接断开了（如果找期货公司后台人员查Log的话，能看到client version check failed的错误日志）。

7 登陆失败

OnUserLogon()回调函数，m_Result会返回结果，返回0是登陆成功，其它值均为登陆失败。登陆失败的错误码在EesTraderDefine.h均能查到，最常见的情况有：

Result 值	错误原因
1	用户名或密码错误，通常是密码错误。
2	用户未绑定任何资金账户。需要联系期货公司后台人员，查看用户/账户状态。
5	缺少客户端产品标识、mac 地址错误（要求格式正确）

8 订单编号

每一个被交易所接受的订单，都包含 ClientOrderToken、MarketOrderToken、MarketOrderId三个订单编码信息：

订单编号	含义	注意事项
ClientOrderToken	客户端单号	每个用户下的 ClientOrderToken 一个交易日内要求递增（但不要求连续），否则下单会失败
MarketOrderToken	REM 单号	REM 系统的显示单号（非 REM 系统真正内部单号）
MarketOrderId	交易所单号	注意 Market 这个词有时表示交易所、有时表示 REM

9 订单状态按位查看

```
typedef unsigned char EES_OrderStatus;    ///< 按照二进制与存放多个订单状态
EES_OrderStatus      m_OrderStatus;      ///< 请参考EES_OrderStatus的定义

117 | typedef unsigned char EES_OrderStatus;    ///< 按照二进制与存放多个订单状态
118 | #define EES_OrderStatus_shengli_accept 0x80 ///< bit7=1: EES系统已接受
119 | #define EES_OrderStatus_mkt_accept 0x40  ///< bit6=1: 市场已接受或者手工干预订单
120 | #define EES_OrderStatus_executed 0x20   ///< bit5=1: 已成交或部分成交
121 | #define EES_OrderStatus_cancelled 0x10   ///< bit4=1: 已撤销，可以是部分成交后撤销
122 | #define EES_OrderStatus_cxl_requested 0x08 ///< bit3=1: 发过客户撤单请求
123 | #define EES_OrderStatus_reserved1 0x04   ///< bit2: 保留，目前无用
124 | #define EES_OrderStatus_reserved2 0x02   ///< bit1: 保留，目前无用
125 | #define EES_OrderStatus_closed 0x01     ///< bit0=1: 已关闭，(拒绝/全部成交/已撤销)
---
```

m_OrderStatus 值	二进制	业务含义
192	11000000	订单未关闭、订单被 REM 及交易所均接受
217	11011001	订单已关闭、全部撤销
225	11100001	订单已关闭、全部成交
249	11111001	订单已关闭、部成部撤

10 下单失败

下单函数EnterOrder()返回0表示下单成功，否则表示下单失败。下单失败的情况，订单消息是不会送到REM服务端（这种情况无法在REM后台查到订单的错误日志）。发送失败的错误原因，在EesTraderErr.h头文件中有声明。比较典型的错误，如#define ORDER_TOKEN_ERROR 9

11 下单被 REM 拒绝

这种情况，下单函数EnterOrder()返回0，订单消息送到了REM服务端，再触发OnOrderReject回调，m_ReasonCode返回订单被拒绝的错误码，在EesTraderDefine.h文件中可找到错误描述。REM拒绝的原因分两类：1）语法检查错误；2）风控拒绝。

0-22为语法检查错误；50-116为风控拒绝。常见的情况如下：

m_ReasonCode 值	错误描述	备注
4	TIF 不在合法值范围	TIF 不为 0 或 99998
6	委托价格必须>0	价格为 0，没有正常赋值
14	对没有权限的账户进行操作	用户被禁用(REM 后台有该功能)；也可能填错了 account
16	不存在的账户	更多的情况是填错交易所编码
20	资金账号未正确配置交易所编码	下错了投机/套利/套保标识
114	可以资金不足	验资不过
115	可平仓位不足	验仓不过
63	在指定时间1内收到订单次数上限	触发 REM 流控

12 下单被交易所拒绝

下单被交易所拒绝，OnOrderMarketReject回调会被触发，m_ReasonText会返回交易所拒绝的原因（注意下交易所是GB2312的编码格式）。常见的拒绝原因有：

- 1) 价格跌破跌停价，因为 REM 内部的涨跌停价取的是 20%，所以会先触发交易所的限制；
- 2) 当前状态禁止此项操作，因为非交易时段报单；
- 3) 发送报单失败，交易所流控情况；
- 4) 超出客户限仓，客户持仓超出交易所最大限制了。

13 下单立即自动撤单问题

很多用户下单前会做类似memset(&ordField, 0, sizeof(EES_EnterOrderField))的操作，然后再对下单的参数逐个赋值。如果忽略了对m_Tif进行赋值，m_Tif=0表示FAK/FOK订单，则下单后不成交会立即自动撤单回来。正常的日内订单，需要将m_Tif赋值为99998。

14 撤单参数

撤单函数CancelOrder需要填入REM返回的单号m_MarketOrderToken来撤单，这点不同于其他一些平台，可以用客户端的单号来撤单。

15 撤单拒绝

撤单被拒绝的原因也是按二进制位查看的：

```
625 // 撤销指令的拒绝原因对照表,这是用二进制位表示的一个原因组合,一个位为1,表示该位对应的原因因为真
626 // 0 整体校验结果
627 // 1 委托尚未被交易所接受
628 // 2 要撤销的委托找不到
629 // 3 撤销的用户名和委托的用户名不一致
630 // 4 撤销的账户和委托的账户不一致
631 // 5 委托已经关闭,如已经撤销/成交等
632 // 6 重复撤单
633 // 7 被动单不能被撤单
```

撤单被拒绝通常的情况都是该订单已经撤销或成交了；撤单请求发起一次，交易所撤单拒绝的消息会回来两三次，属于正常情况。

16 FAK 时序

FAK的单子，在REM系统里面是OnOrderExecution先触发，然后再触发OnOrderCxled，成交比撤单先收到；而CTP等系统顺序要反一下。这个和REM系统的设计有关系，如果OnOrderCxled先触发，REM里面订单会被关闭，后面再收到的成交就不会被处理，则仓位数量将会出错。FAK/FOK 指令，盛立测试环境暂时还不支持。

17 平仓指令

交易所规则来说，只有上期所提供平今、平昨指令，而中金所、大商所、郑商所都只有平仓指令，其中中金所优先平今仓；而REM里面的仓位严格区分“今仓”与“昨仓”，平仓必须要指定平今、平昨。所以，针对在主席客户端平仓的情况，需要特别注意一些场景。比如，用户持有中金所IF1710多仓10手，其中昨多仓5手、今多仓5手，在快期上平仓时需要分两次下平仓单，先平今5手、再平昨5手。如果直接下平仓10手，会造成REM里面的昨仓出现“负数”！如果先“平昨”，中金所现在是优先平今仓，会造成REM里面剩下今仓5手，而主席上面剩下昨仓5手的不一致情况。

18 下单“收不到”REM 回报

在实盘环境OnOrderAccept返回的速度非常快，往往下单函数EnterOrder还没有返回时，就收到REM回报了。此时客户端程序可能尚未建立起下单的单号对应关系，就可能出现回报然后因为找不到映射关系直接return掉的情况。在互联网上测试因为网速慢，所以不会有这个问题。这种情况，并非是真的没有收到REM的回报。

19 指定席位下单

REM系统支持多席位方式连接交易所，每个席位对应一个Market Session，可以通过调用QueryMarketSession方法查询。OnQueryMarketSession回调会返回Market Session的具体信息。m_SessionCount返回有多少个Session，m_SessionId[255]数组（可能有多）返回SessionId。下单结构体的m_MarketSessionId用来指定Market Session进行下单。如果不指定席位，REM默认采用系统级“轮询”的方式进行报单。指定席位下单可以降低“自成交”概率。

20 交易所合约状态变化通知

OnMarketSessionStatReport回调函数，会推送交易所合约状态变化信息。具体到实际的REM生产环境，是否推送该消息，取决于系统是否向交易所注册公有流信息，目前绝大部分的REM实盘环境都注册了公有流。

21 取 ClinetOrderToken 最大值

交易登陆成功的回调函数OnUserLogon，m_MaxToken会返回用户当前交易日的最大token值；注意下，一个Account会绑定多个用户，每个用户下的client token都独立编制。可以在登陆成功以后，记录下最大的token值，然后每次下单前加N（不一定是加1，只要递增即可）；也可以通过调用GetMaxToken函数，每次下单前查询一遍，然后再加N的方式实现递增。另外，多线程下单时，注意加锁。

22 持仓查询

调用QueryAccountPosition (const char* accountId, int nReqId) 可以查询账户持仓信息，需注意传入的参数是Account而非LoginId。回调函数OnQueryAccountPosition (const char* pAccount, EES_AccountPosition* pAccountnPosition, int nReqId, bool bFinish)会返回具体的持仓信息。同一合约的多仓、空仓为两个不同的EES_AccountPosition对象；查询一次，回调函数会触发多次，仓位数据是逐条返回的。当bFinish=true时为最后一条消息（此时pAccountnPosition传过来的各字段值均为0）。

23 订单与成交查询

查询函数	回调函数	描述
QueryAccountOrder(const char* accountId)	OnQueryTradeOrder(const char* pAccount, EES_QueryAccountOrder* pQueryOrder, bool bFinish)	会返回某账户下所有的订单信息。
QueryAccountOrderExecution (const char* accountId)	OnQueryTradeOrderExec(const char* pAccount, EES_QueryOrderExecution* pQueryOrderExec, bool bFinish)	会返回某账户下所有的成交信息。

对于盘中用户的策略程序有重启的情况，重启后通常需要恢复“挂单”状态，可以通过调用订单查询，对返回的订单逐一判断是否关闭来恢复。

24 外部单

所谓“外部单”是指用户非正常通过REM的API下出去的订单。包括：

第一、使用主席平仓所产生的外部订单；

第二、REM后台人工补单产生的。

外部单会触发OnPostOrder()回调函数，外部单成交则触发OnPostOrderExecution()回调。需要说明一点，不是所有REM系统都会收到来自主席的平仓消息，这取决于席位权限、网络、及REM后台权限配置；如果REM系统不能自动处理主席平仓，而主席平仓发生了，就需要期货公司后台人员，通过REM后台人工补单的方式，来产生外部单，就是第二种情况。

25 时间格式处理

REM多处返回的时间均为G.M.T时间，类型为typedef unsigned long long int EES_Nanosecond，可以通过convertFromTimestamp(EES_Nanosecond timeStamp, tm& tmResult, unsigned int& nanoSsec) 转换成tm格式。

26 同时下多个单子

批量下单可以调用EnterMultiOrders(EES_EnterOrderField* pArrOrders, int nCount)函数，最大支持同时下4个单子，下单速度会比调用4次普通的EnterOrder()函数快。

27 同时交易多账户

有两种方法可以实现：第一、同一个用户绑定多个账户，需要向期货公司申请（因为通常的配置是一个用户只绑定一个账户）；第二、创建多个 API 交易实例。

28 成交回报不含合约代码如何处理

针对一个策略同时交易多个合约的情况，OnOrderExecution(EES_OrderExecutionField* pExec) 回调返回的EES_OrderExecutionField没有合约代码，不能直接判断出是哪个合约发生成交了。但EES_OrderExecutionField有返回ClientOrderToken、MarketOrderToken两个字段，用户只要在调用EnterOrder()函数时根据本地单号创建map或者在OnOrderAccept()根据REM返回的单号创建map就能通过查询找到成交是哪个合约了。