REM柜台极致版和普通版本客户端API的主要区别

# ConnServer接口

连接服务器接口，因为极致版，采用了UDP发送订单和撤单的模式，因此需要新增3个UDP相关参数，为了避免参数过长，将之封装为一个结构体：

struct EES\_TradeSvrInfo

{

char m\_remoteTradeIp[16]; /// 服务器交易IP

unsigned short m\_remoteTradeTCPPort; /// 服务器交易TCP端口

unsigned short m\_remoteTradeUDPPort; /// 服务器交易UDP端口

char m\_remoteQueryIp[16]; /// 服务器查询IP

unsigned short m\_remoteQueryTCPPort; /// 服务器查询TCP端口

char m\_LocalTradeIp[16]; /// 本地交易IP

unsigned short m\_LocalTradeUDPPort; /// 本地交易UDP端口

};

除了新增的3个UDP相关参数，其余4个参数和普通版用法一致。3个新参数分别是：

* 交易服务器UDP端口：默认值19999，UDP的IP实际上就是交易服务器的IP
* 本地交易IP：需要指定为客户端机器上，与交易服务器IP在同网段的IP地址。比如服务器的交易地址是：192.168.100.50，那就应该填入客户端机器上，192.168.100.xxx这个网段的地址。
* 本地UDP端口：随意指定一个客户端机器上未被占用的端口即可，推荐使用10000-65535范围的一个值。

# OnOrderMarketAccept事件

交易所接受事件，参数所用到的EES\_OrderMarketAcceptField这个结构体，新增2个字段：

EES\_UserID m\_UserID; ///< 下单的 user id

EES\_ClientToken m\_ClientOrderToken; ///< 下单的client token

这两个值得含义与EES\_OrderAcceptField结构体的同名字段完全相同，可以方便客户找到原始订单而无需建立二层对照关系。

# OnOrderMarketReject事件

交易所拒绝事件，类似OnOrderMarketAccept时间的改动，EES\_OrderMarketRejectField结构体，也新增了同样2个字段：

EES\_UserID m\_UserID; ///< 下单的 user id

EES\_ClientToken m\_ClientOrderToken; ///< 下单的client token

这两个值得含义与EES\_OrderAcceptField结构体的同名字段完全相同，可以方便客户找到原始订单而无需建立二层对照关系。

# 内置的客户端流控功能

目前的极致版，内置了一个简易的客户端流控，控制客户端单位时间内的下单+撤单次数。参数是每500毫秒总共50次。触发流控时当前下单/撤单会失败，数据不会发送出去，并且返回如下错误码：

#define CLT\_ORDER\_SPEED\_FLOWCTRL 15

在客户端做得流控、直接从网络上省掉了流量，也不会造成服务器流控引起的熔断了。

这个功能日后还会增强，比如参数由服务器控制、客户端可以再调整等等。目前接口上实际上看不出来有流控功能。

# 其它

目前极致版和普通版在使用上没有其它区别了。但是两个版本是不兼容的，不能替换使用，客户需要针对不同版本的头文件和库文件进行链接编译程序后才能使用。