Exchange Simulator简介文档

第一部分: 使用到的FIX tags

. 8 : 该tag用于指示利用的协议版本，本程序为FIX 4.2。

.9 : Message length, in bytes, forward to the CheckSum field. ALWAYS SECOND FIELD IN MESSAGE.（没用到功能，仅占位）

.11 : Unique identifier for Order as assigned by institution 包含订单特定ID，彼此互异，作为辨认订单的依据，本程序中为长度为10的字符串常量

.14 : Total number of shares filled 成交share的数目

.35 : Defines message type. ALWAYS THIRD FIELD IN MESSAGE 包含通信信息的类型

=8 Execution Report server在接收到client发送的信息后，回复client时发送

=D Order Single client欲创建新订单时向server发送

=F Order Cancel Request client欲取消订单时向server发送

.38 : Number of shares ordered. This represents the number of shares for equities or based on normal convention the number of contracts for options, futures, convertible bonds, etc. 预定交易的share数量

.39 : Identifies current status of order 包含订单目前状态

=0 New 新建订单成功

=1 Partially filled 部分share交易成功

=2 Filled 全部share交易成功

=4 Canceled 订单取消成功

=6 Pending Cancel 订单取消请求，订单进入pending状态

=8 Rejected 订单取消驳回，share或已全部交易或有其他原因

.44 : Price per share 每股成交金额

.52 : Time of message transmission 包含时间字符串，表示生成信息（受到信息，交易成功&失败）的时间

. 54 : Side of order 订单类型，本程序中有Sell和Buy两类

=1 Buy 买入

=2 Sell 卖出

.56 : Assigned value used to identify receiving firm 欲买入或卖出股票的种类

.58 : Free format text string 任意字符串

.150 : Describes the specific ExecutionRpt (i.e. Pending Cancel) while OrdStatus will always identify the current order status (i.e. Partially Filled) 与tag 39 具有相似的功能

.151：Amount of shares open for further execution 剩余share的数目

第二部分: 功能简析

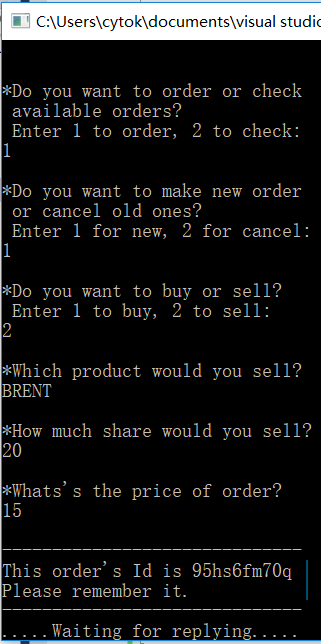
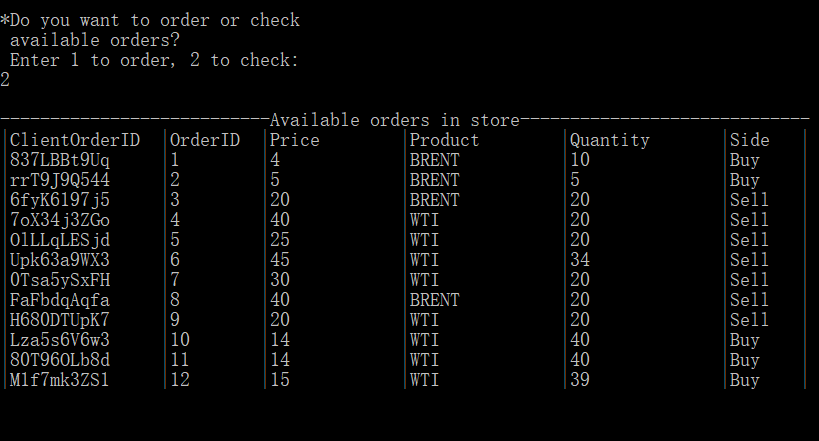
client端口

1通过过程式询问，得到标准化，规范化的FIX协议字符串，通过与server端口建立连接，发送字符串；

2接受server的回应，解析回复字符串， 将FIX码中的内容保存到类下成员变量中，便于随后调整订单状态；

3询问当前可以进行的交易，即得到目前存在的所有未交易的order的信息（由于写代码时只能与自己连接，所以交易的对象全是自己，即自己卖自己买，显示出来的order包括自己，然而实际中应该避免自买自卖的情形，如果需要可以通过修改输出简便地屏蔽掉自己提出的order， 交易的时候同样可以简单判断--当order不是同一个visitor提出的时候才能进行）；

4建立档案，将改动的结果整理到目录文件中，实时更新。



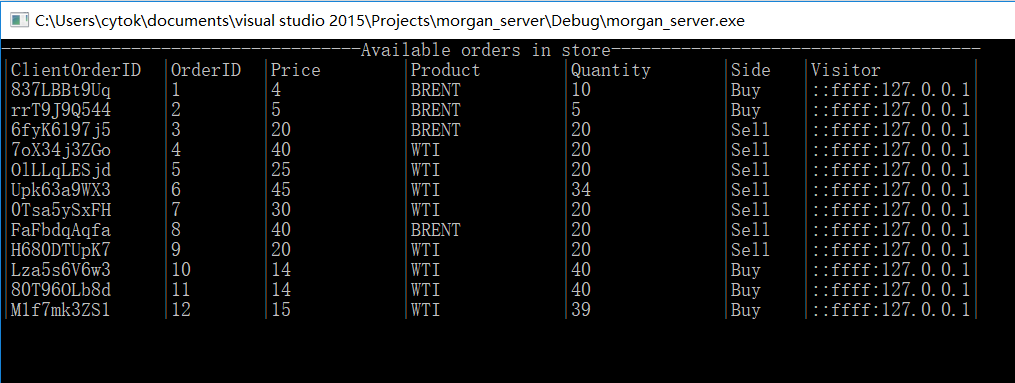
server端口

1从文件中读取目前所有订单内容，储存订单信息，包括类型，Order ID（按加入订单池的时间，从1自动向后排序），Client Order ID（区分不同订单的特定ID， 即tag 11 所包含的10位字符串），价格，可交易share数量，订单方（即提交订单，使用client的visitor）；

2 与client建立通信连接，解析client发送的信息，追加存入订单池，并返回成功收到的消息；

3 自动匹配能够完成的交易，并在毎起订单的发起方连入服务器时向他发送交易信息，包括该订单基本信息，订单share余额，订单成交状况，取消是否成功等；

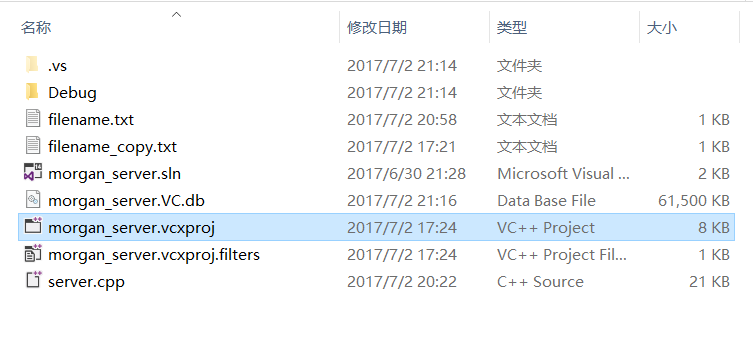
4 每次与该访问者互动后后，格式化输出订单池内现存的所有order。



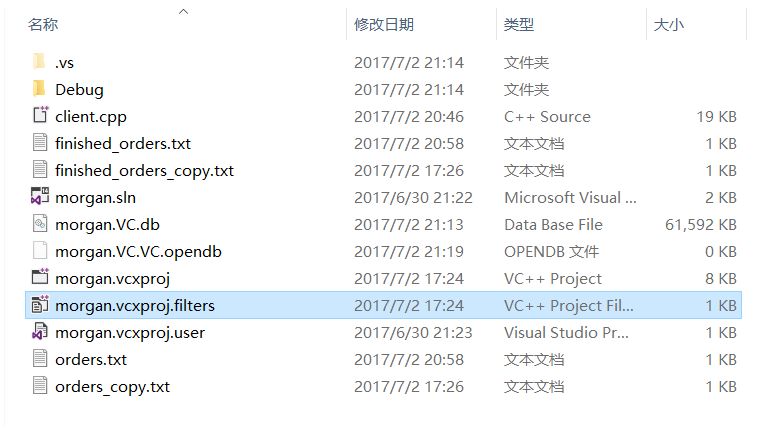
第三部分: 操作

1.首先保证server与client可以相互连接（由于我水平有限，本程序仅在同一台windows64位系统电脑上测试通过，不能保证在不同电脑间或不同配置的主机上连接成功，而且存在第二部分“client端口”3中提出的问题）

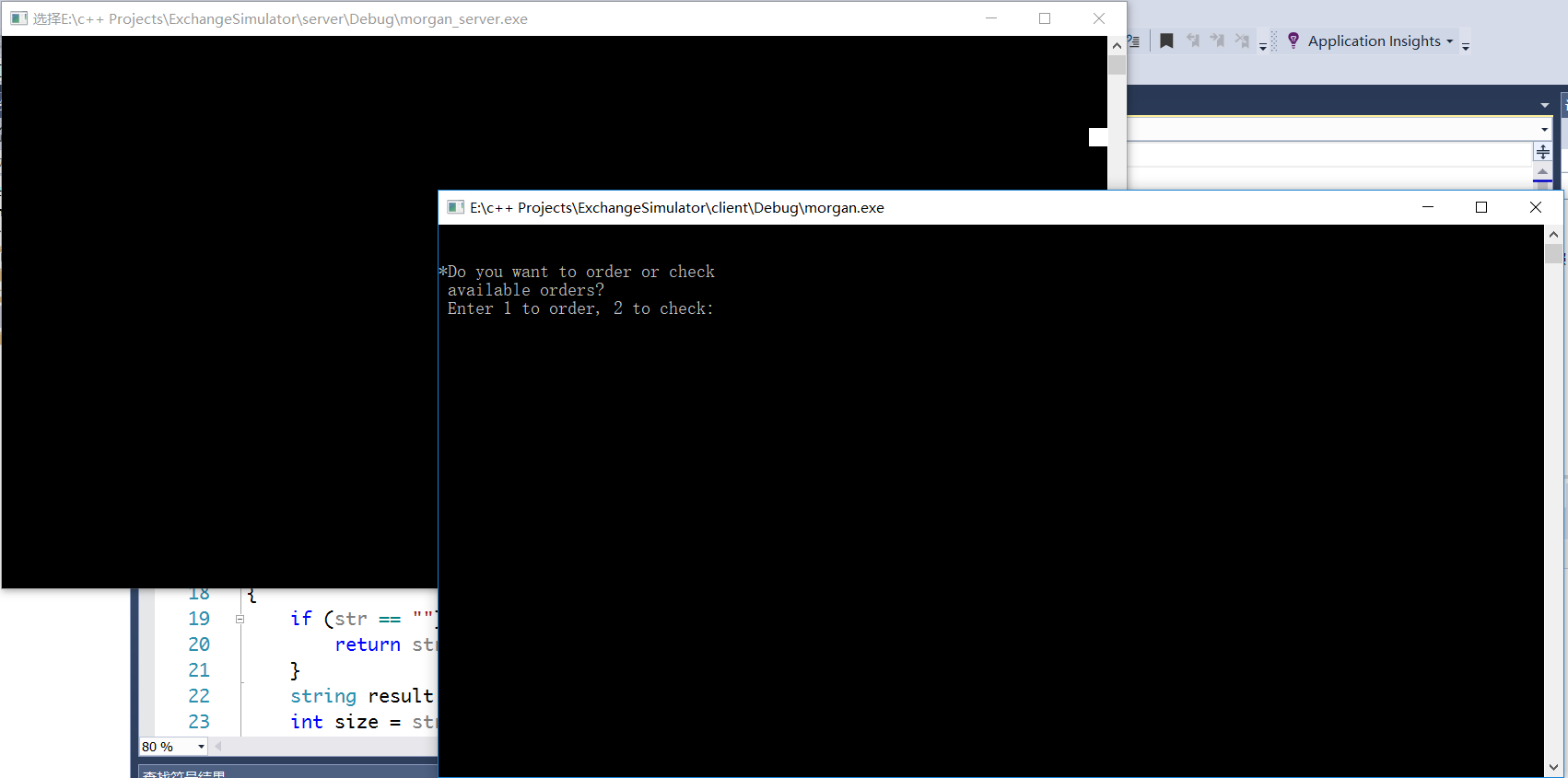
2.先打开server文件夹下的vs project（如下图），运行程序



2.再打开client文件夹下的vs project（如下图），运行程序



3.运行成功后，出现两个窗口



4. 根据提示输入即可。

第四部分: 注意

1.filename\_copy.txt是filename.txt的副本，其他…\_copy.txt文件同样。filename.txt文件内容是初始化的订单池（orders），orders.txt的内容是client还未完成的order，finished\_orders.txt的内容是client已经完成的order。每次被修改的程序是….txt，… \_copy.txt并不会改变。 如果程序异常退出（比如运行到一半被使用者强行关闭），….txt中的内容可能会部分或全部丢失，导致再次运行时报错，这时将…\_copy.txt中的内容覆盖.txt至初始状态即可（成功运行的话，可以更新…\_copy.txt中的内容）。

2. 本程序仅在同一台windows64位系统电脑上测试通过，不能保证在不同电脑间或不同配置的主机上连接成功，而且存在第二部分“client端口”3中提出的问题。所以如果在不同机上测试，使用者必须修改程序，包括IP地址、判断订单是否能够交易时判断是否非同一个visitor所发。

3.由于时间有限，而且最近身体不适，无法及时在程序中写//提示，请您谅解！

感谢您的阅读！