CTP期货期权

交易程序

使用说明

**Rev. 1.0**

**10/11/2022**

*This page has been intentionally left blank.*

简单介绍

特点

ctp-future-option是基于上期技术综合交易平台（CTP）开发的开源程序，适用于中国国内期货期权交易，其特点如下：

* 使用Linux操作系统，也为向Windows操作系统移植核心代码做了适应。
* 行情更新和交易策略模块化设计，易于扩展，一种交策略可以同时在多个期权品种上交易，一个期权品种上也可以有多个策略在交易。
* T型报价根据行情推送实时更新，同时根据期权定价模型完成期权的希腊值和隐含波动率的更新。
* 使用sqlite数据库保存策略参数和交易组合数据。
* 完全使用C/C++语言进行开发，没有混合编程，运行效率高。

…

使用环境

ctp-future-option是在Linux操作系统上开发和使用的，作者使用的Linux发行版本为CentOS 8。如果有兴趣，读者可以自行尝试其它Linux发行版本，不过应该差别不大。

在使用ctp-future-option之前，读者首先需要确保以下工具已经安装在自己的Linux操作系统上：

* gcc/g++/gdb
* cmake
* Qt

另外像git，Visual Studo Code之类的工具也最好装上。如果没有安装，请使用相应的软件包管理器安装（yum/dnf，apt-get）。

目录结构

项目各个目录的内容和作用如下所述：

ctp-future-option/ 项目根目录

├── code 源代码目录

│   ├── core 交易核心源代码目录

│   └── gui 图形用户界面设计文件和代码所在目录

├── config.txt 配置文件，用在CMakeLists.txt中

├── ctp-api CTP API头文件和其它相关文件所在目录

├── doc 文档所在目录

├── solution 项目工程文件所在目录

│   ├── build 在此目录编译链接生成可执行文件

│   │   ├── CMakeLists.txt cmake所用主文件

│   │   ├── …

│   └── TradeDatabase 交易数据库所在目录

└── sqlite-api sqlite数据库API所在目录

免责声明

本设计可以自由使用，作者不索取任何费用。  
作者对使用结果不做任何承诺也不承担其产生的任何法律责任。  
使用者须知晓并同意上述申明，如不同意则不要使用。

2.

使用教程

在使用本教程之前，读者需要到国内支持期货期权交易的期货公司开通期货期权模拟交易账户并申请程序化交易相关的穿透式监管验证代码。一般而言，国内期货公司的期货期权交易都是基于CTP柜台系统的。

读者使用git clone或者其它方式将ctp-future-option复制到自己Linux本机的工作目录下，在命令行终端，按以下步骤生成可执行文件：

1. cd ctp-future-option/solution/build
2. cmake .
3. make

一切正常的情况下，build目录下将会产生一个名为ctp-future-option的可执行文件。

基本用法

得到可执行文件之后，在命令行终端build目录下输入命令

./ctp-future-option

启动程序运行。在第一次运行程序时，需要输入账户以及交易服务器相关信息，这些信息将会保存在交易数据库中。以后再次运行程序时，除了用户密码外，其它信息将直接从数据库中读出，不必重复手动输入。

以下示例第一次运行过程中从Terminal得到的几张截图，图中使用红色矩形框标识出来的部分都是需要用户手工输入的内容，包括

* 交易架构名称（例子用nhqh-sim ，也可以用其它名称）
* 经纪商代码BrokerId，用户账号UserId ，密码Password
* 穿透式监管认证用的UserProductInfo，AppId和AuthCode
* 交易前置服务器TradeFront和行情前置服务器MdFront的地址信息（表示为tcp://IP地址:端口号）

除了手动输入的信息，第一次运行中其它配置信息均从build/config/nhqh-sim下的各个配置文件自动获取。

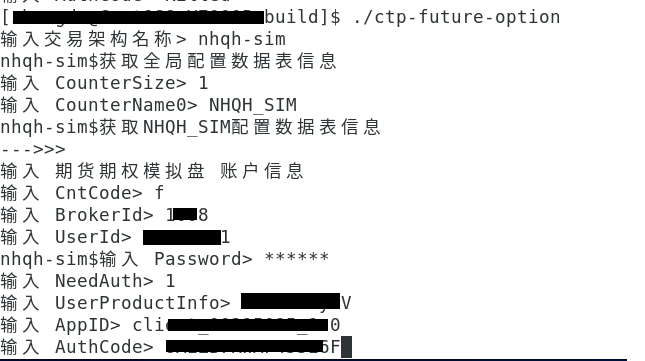


图 1



图 2

在命令行界面参数输入结束后，等待程序继续运行，最后会弹出以下图形用户界面。



图 3

这是一个简洁的用户界面，最上面的表格显示选中期权的T型报价，下面的表格依次显示当前持仓和可撤委托。用户可以在上面用鼠标尝试点取各个图形元素，熟悉用法。

点击“操作”菜单，可以选取下单，撤单操作。双击图中的各期权项或者点击右键选择，将会弹出对应的操作对话框进行下单，平仓或者撤单。

进阶用法

1. 增删交易品种

在第一次运行之后，如果用户需要增加或者删除交易的期权种类，可以修改sqlite数据库solution/TradeDatabase/TradeDb-nhqh-sim.db中的VoidTest\_Config数据表，具体的处理机制可以参考C++源文件StrategyVoidTest.cpp中的Initialize()函数。

VoidTest\_Config数据表中的KeyId为OptionKindListSize的ValueExp值表明需要交易的期权种类数目，KeyId为OptionIdx的ValueExp（比如yz）值表示第x个期权的代码，KeyId值为UnderlyingIdx的ValueExp值表示第x个期权标的代码，KeyId值为ExchangeIdx的表示第x个期权的交易所代码。对于新增加的期权品种yz，还需生成名称为VoidTest\_yz\_Config的数据表，其内容可以从类似的表拷贝过来。以上内容在数据库中均有可供参考的示例，请自行研读。

2. 自动策略交易

ctp-future-option的行情和交易接口是完全开源的，读者可以在此基础上开发属于自己的全新自动策略交易系统。

另外，作者随项目提供了一个策略交易系统的参考接口，读者也可以以此为参考开发交易策略，具体请参见以下文件：

code/core/RealProj/RealTop/StrategysManage.h

code/core/RealProj/Strategy/Top/StrategyBase.h

code/core/RealProj/Strategy/Top/PortfolioBase.h

code/core/RealProj/Strategy/Top/PositionTrade.h

作者开发的策略交易系统采用层次结构，最上层的策略管理器StrategysManage管理多个交易策略，次一层的交易策略可以有多个交易组合，而交易组合则可以拥有多个腿的Instrument交易头寸。

/code/core/RealProj/Strategy/VoidTest目录下示例了一个不做任何自动交易的空策略，前面的图形用户界面也利用了此策略提供数据，可供参考。

按照项目提供的策略交易接口设计完成新策略（假设新策略名为Xyz）之后，还需按照以下步骤将其集成到项目中。

1. 在code/core/RealProj/RealTop/StrategysManage.h文件中包含策略头文件。

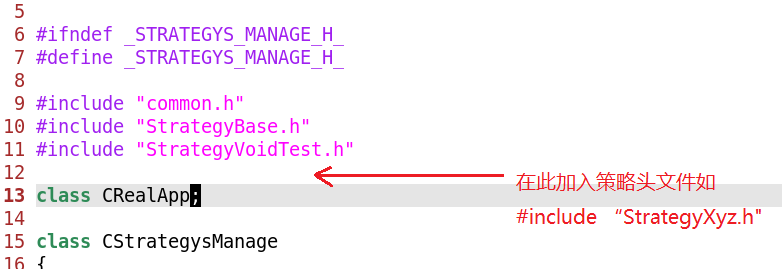


图 4

1. 在code/core/RealProj/RealTop/RealApp.cpp的GetGlobalConfigFromDb()函数中实例化策略对象。



图 5

1. 改写程序的main()函数，在其中调用code/core/RealProj/RealTop/RealApp.cpp的AutoTradeMain()函数。

AutoTradeMain()函数的处理机制读者可以自行去研读。有一点需要提醒读者注意，在策略自动交易模式下，为提高交易效率，尽量不要使用图形用户界面，使用Terminal命令行界面即可。在这种情况下，可以修改CMakeLists.txt文件将Qt相关的功能去除。