上期所CTP的python封装与跨平台编译

1 目的：

本项目在于将上期技术提供的CTP的C++代码库，封装成python可调用的代码库。将所有的C++函数进行一一对应。使得使用python的系统，可以直接调用底层接口，实现正常的交易查询，发单，撤单等交易动作。

2 使用工具及原理：

本项目采用了boost.python框架进行C++和python库的交互，使得python可调用C++封装好的库函数。将api中的结构体封装成python的类，行情API封装成CtpMd和交易API封装成CtpTd.

采用cmake支持跨平台编译，可生成跨平台的编译文件，实现Linux，Windows等平台实现。

使用perl脚本语言，根据数据结构和数据类型，自动生成C++对应结构的函数及其声明。

3 安装及编译过程：

1. 安装配置cmake：

下载安装cmake最新版本后，打开GUI客户端，将工程导入，配置python的编译路径和修改CMAKE的属性配置（如有需要）。在工程项目中，对CMakeList.txt文件，根据实际需求，修改Boost.python的安装路径和项目的路径信息。

2）安装配置boost.python：

下载boost.python并解压；打开VS命令行接口。若要针对面向 x64 平台的 64 位命令行生成配置此命令提示符窗口，请在命令提示符处输入：`vcvarsall amd64`，同理，若要针对面向 x86 平台的 64 位命令行生成配置此命令提示符窗口，请在命令提示符处输入：`vcvarsall amd64\_x86`。若要针对面向 ARM 平台的 64 位命令行生成配置此命令提示符窗口，请在命令提示符处输入：`vcvarsall amd64\_arm`。

cd boost\_x\_xx\_x && ./bootstrap.bat --prefix=c:/boost, 也可以增加`--with-python=/usr/bin/python3 --with-python-root=/usr

需要在bootstrap.bat的输出中指明的配置文件project-config.jam中添加如下一行配置：using python : 2.7 C:\\Python27 : C:\\Python27\\include : C:\\Python27\\libs。（路径根据本机的python版本和位置有所不同）

执行如下命令进行编译,其中如果是64位python则需要address-model=64，32位python则不需要;在linux编译需要添加cxxflags=-fPIC, windows需要添加--build-type=complete --with-date\_time。

`bjam install --toolset=msvc-14.0 --with-python --with-thread link=static threading=multi variant=release,debug runtime-link=static address-model=64`

3）项目的添加和编译：

如上所述，在打开GUI，导入项目后，点击config，检查配置过程和依赖。通过后点击generate来生成项目。没有错误后，点击Open Project打开VS，进行项目工程的编译。在编译过程中注意项目python库依赖的导入，要把boost.python的库文件添加到编译目录中。

4）利用perl脚本对代码的自动生成：

- apiDataType.pl < ThostFtdcUserApiDataType.h

这个脚本用于抽取ThostFtdcUserApiDataType.h文件中typedef的新类型与原类型的对应关系。我们需要对char[]需要专门处理，此脚本可以看到需要专门处理的类型名。

- structExporter.pl [ThostFtdcUserApiDataType.h] [ThostFtdcUserApiStruct.h] > exporter.txt

这个脚本用于对ThostFtdcUserApiStruct.h中定义的struct进行去掉"ThostFtdc"的命名和对char[]类型字段提供get／set方法。

- structWrapper.pl [ThostFtdcUserApiDataType.h] [ThostFtdcUserApiStruct.h] > wrapper.txt

这个脚本用于对ThostFtdcUserApiStruct.h中定义的struct对boost.python的提供封装的支持。

- extractType.sh (ThostFtdcMdApi.h | ThostFtdcTraderApi.h) > field.txt

这个脚本用于从ThostFtdcMdApi.h或ThostFtdcTraderApi.h中抽取出使用的struct名称，并将struct名的前缀ThostFtdc去掉。

- extractStructExporter.pl exporter.txt field.txt

从前面生成的exporter.txt中抽取出在field.txt中出现的结构的exporter。

- extractStructWrapper.pl wapper.txt field.txt

从前面生成的wrapper.txt中抽取出在field.txt中出现的结构的wapper。

- generate.pl (ThostFtdcMdApi.h | ThostFtdcTraderApi.h) 子类名 生成的文件们的前缀

需要三个参数，第一个是头文件，第二个是Api封装后的类名，第三个是产生的文件的后缀名，因为会产生多个文件。

在执行上述脚本后，已经将c++库中数据结构和类型都做了对应，将产生的文件和原本项目中的ctpmd.h,ctpmd.cpp,ctptd.h,ctptd.cpp四个文件进行对应，把新接口添加的函数补充上，然后再VS中编译并解决问题。直到没有错误出现。

在VS中点击生成，即可生成相应平台下的md和td文件。添加到项目中进行测试即可。