**行情接口说明文档**

行情模块使用动态库方式动态加载，主要函数接口如下

1. bool QUATION\_API CreateQuationWnd(HWND hParent);

功能：创建行情窗口

参数：hParent

父窗口句柄

1. void QUATION\_API SetTradingDay(std::string strTradingDay);

功能：设置交易日

参数：strTradingDay

交易日，例：20141009

1. void QUATION\_API ShowInstrument(std::string strInstrument);

功能：设置显示的合约

参数：strInstrument

合约名，例：IF1410

1. void QUATION\_API SetTradingScales(std::string strInstrument, const std::vector<std::pair<std::string,std::string>>& TradingTime);

功能：设置合约交易时段

参数：strInstrument

合约名，例：IF1410

TradingTime

交易时段，例如：中金所交易时段09:15 11:30 13:00 15:15

std::pair<std::string,std::string> 为一个交易时段，例如

std::make\_pair("09:15", "11:30")

1. void QUATION\_API InitFinish(std::string strInstrument);

功能：完成初始化合约

参数：strInstrument

合约名，例：IF1410

示例代码：

//.h文件

HINSTANCE m\_HInstance;

//.cpp文件

CString strPath;

char localPath[256];

memset(localPath, 0, 256);

GetModuleFileName( NULL, localPath, 256);

std::string filename=localPath;

size\_t splitpos=filename.find\_last\_of('\\');

strPath.Format(\_T("%s"), filename.substr(0, splitpos+1).c\_str()) ;

CString strDir;

#ifdef \_DEBUG

strDir.Format(\_T("%sOptionQuationD.dll"), strPath);

#elif

strDir.Format(\_T("%sOptionQuation.dll"), strPath);

#endif

m\_HInstance = LoadLibrary(strDir.GetBuffer(strDir.GetLength()));

Func\_CreateQuationWnd funcCreate=(Func\_CreateQuationWnd)GetProcAddress(m\_HInstance,"CreateQuationWnd");

funcCreate(m\_hWnd);

Func\_SetTradingDay funcSetTrayDay = (Func\_SetTradingDay)GetProcAddress(m\_HInstance,"SetTradingDay");

std::string strTradingday = "20141009";

funcSetTrayDay(strTradingday.c\_str());

Func\_ShowInstrument funcShowInstrument = (Func\_ShowInstrument)GetProcAddress(m\_HInstance,"ShowInstrument");

std::string strInstrument = m\_strInstrument.GetBuffer(m\_strInstrument.GetLength());

funcShowInstrument(strInstrument);

//交易时段

std::string strInstrument2 = m\_strInstrument.GetBuffer(m\_strInstrument.GetLength());

std::vector<std::pair<std::string,std::string>> TradingTime;

TradingTime.push\_back(std::make\_pair("09:15", "11:30"));//第一交易时段的起止时间

TradingTime.push\_back(std::make\_pair("13:00", "15:15"));//第二交易时段的起止时间

Func\_SetTradingScales funcSetTradingScales = (Func\_SetTradingScales)GetProcAddress(m\_HInstance,"SetTradingScales");

funcSetTradingScales(strInstrument2, TradingTime);

Func\_InitFinish funcInitFinish = (Func\_InitFinish)GetProcAddress(m\_HInstance,"InitFinish");

funcInitFinish(strInstrument);