

Wind C++数据及交易接口

修订时间：2015-03-01

上海万得信息技术股份有限公司

Shanghai Wind Information Co., Ltd.

地 址 上海浦东新区福山路 33 号建工大厦 9 楼

邮编 Zip 200120

电话 Tel (8621)6888 2280

传真 Fax (8621)6888 2281

主页 <http://www.wind.com.cn>

目录

1	概述.....	6
2	WIND C++接口配置.....	6
2.1	WIND C++接口文件存放位置.....	6
2.2	WIND C++接口配置的系统环境要求与方法.....	6
2.2.1	系统环境要求.....	6
2.2.2	配置方法.....	6
2.3	详细使用文档的获取方式.....	7
3	C++接口应用示例.....	8
3.1	示例代码存放位置及环境要求.....	8
3.1.1	示例代码存放位置.....	8
3.1.2	系统环境要求:	8
3.2	数据接口示例代码执行和主要代码实现过程展示.....	8
3.2.1	运行程序, 登录执行.....	8
3.2.2	部分关键代码展示.....	9
3.2.3	WindData 数据结构.....	10
3.3	交易接口示例主要代码实现过程展示.....	13
4	C++代码生成器使用说明.....	15
4.1	WINDNAVIGATOR 导航(命令生成器).....	15
4.2	部分应用截图.....	16
4.2.1	WSD 代码生成截图:	16

4.2.2	Trade 代码生成截图.....	18
4.2.3	BKT 代码生成截图.....	19
5	WIND C++接口详细说明.....	19
5.1	接口概要.....	19
5.1.1	接口功能函数列表.....	19
5.2	WIND 接口控制函数明细.....	20
5.2.1	Wind 接口启动函数: start.....	20
5.2.2	Wind 接口登出函数: stop.....	21
5.2.3	Wind 接口连接状态判断函数: isconnected.....	21
5.2.4	获取错误码相应的错误信息: getErrorMsg.....	21
	获取错误码相应的错误信息.....	21
5.3	WIND 数据接口函数明细.....	22
5.3.1	多值函数-日期序列: WSD.....	22
5.3.2	多值函数-截面数据 WSS.....	23
5.3.3	多值函数-分钟序列 WSI.....	23
5.3.4	多值函数-日内跳价 WST.....	24
5.3.5	日期宏.....	25
5.3.5.1	通用日期宏.....	25
5.3.5.2	特殊日期宏.....	25
5.3.6	多值函数-实时行情 WSQ.....	26
5.3.7	多值函数-数据集 WSET.....	27
5.3.8	组合报表函数: wpf.....	28
5.3.9	组合上传函数: wupf.....	28

5.3.10	日期序列函数 TDays.....	29
5.3.11	根据偏移计算日期的函数 TDaysOffset.....	30
5.3.12	计算日期天数函数 TDaysCount 函数.....	30
5.3.13	取消数据请求/订阅 CancelRequest.....	30
5.3.14	取消所有数据请求/订阅 CancelAllRequest.....	31
5.3.15	数据接口错误类型.....	31
5.4	WIND 交易接口函数明细.....	32
5.4.1	Wind 交易登录函数 tlogin.....	32
5.4.2	经纪商登出函数 tlogout.....	34
5.4.3	委托下单函数 torder.....	34
5.4.4	撤销委托函数 tcancel.....	35
	撤销委托函数.....	35
	LONG tcancel(LPCWSTR orderNumber, LPCWSTR options = NULL);.....	35
5.4.5	查询函数 tquery.....	36
5.4.6	交易接口错误类型.....	37
5.5	WIND 回测接口函数明细.....	38
5.5.1	回测开始 bktstart.....	38
5.5.2	回测查询 bktquery.....	39
5.5.3	回测下单 bktorder.....	39
5.5.4	回测结束 bktend.....	40
5.5.5	回测结束 bktstatus.....	40
5.5.6	查看回测状态 bktsummary.....	41
5.5.7	回测删除 bktdelete.....	41
5.5.8	返回策略列表 bktstrategy.....	42
6	WIND C++接口获取帮助途径.....	42
6.1	本用户手册.....	42

6.2	命令生成器.....	42
6.3	大奖章网站.....	42
6.4	量化交易群.....	43
6.5	终端量化交易平台.....	43

1 概述

Wind C++用户手册介绍 Wind C++量化接口使用方法，包括系统配置、函数说明、例子说明等，以便于用户入门和查阅。阅读本手册之前，建议先阅读《Wind 量化平台-用户手册》，以便了解 Wind 量化平台总体情况。

更多专门细节，可前往 <http://www.dajiangzhang.com/download> 参考其他文档：

更多交易参数和细节说明，请参考《Wind 量化平台-交易专题》

使用过程中碰到的疑难杂症，请参考《Wind 量化平台-常见问题 FAQ》

Wind 量化平台总体介绍（不分语言），请参考《Wind 量化平台-用户手册》

2 Wind C++接口配置

Wind C++量化接口提供三类接口函数，数据接口、交易接口和回测接口。用户可以通过接口进行数据提取、挖掘分析、下单交易等量化行为。

2.1 Wind C++接口文件存放位置

从 <http://www.dajiangzhang.com/download> 下载 “Wind 量化平台-接口库文件-C++”，下载完成后解压。其中：

WAPIWrapperCPP\API 含量化接口头文件 WAPIWrapperCpp.h；

WAPIWrapperCPP\lib 含量化接口的 lib 和 dll；

WAPIWrapperCPP\Samples 含两个例子。WindQuantCppSample 演示了如何使用数据接口，WindTradeCppSample 演示了如何使用交易接口。

WAPIWrapperCPP\src 含接口本身的源代码，一般用户无需关注和使用此文件夹。有兴趣的用户可以参考此源码，调整接口甚至写出其他语言的接口；

2.2 Wind C++接口配置的系统环境要求与方法

2.2.1 系统环境要求

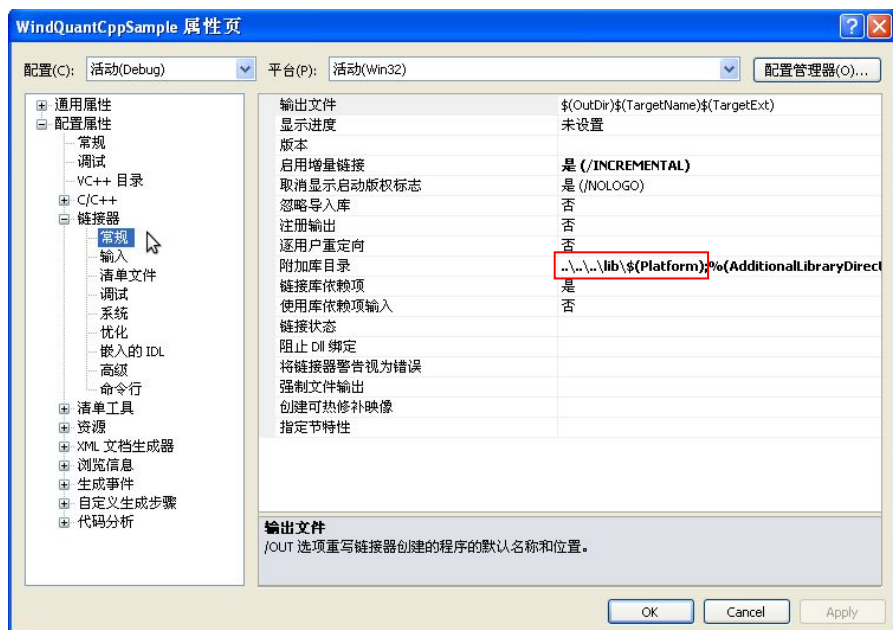
- Windows 系统，支持 32 位和 64 位系统；
- C++常用开发工具，建议使用 VC++6.0 以上的版本，包括 VS2008 和 VS2010 等，支持 32 位和 64 位；
- Wind 终端或个人版量化接口最新版；
- 需要系统管理员权限，根据 C++开发人员的个人开发习惯，可能用到相应插件，如 VisualAssistX 和 H3Viewer 等。
- 首次使用 Wind 量化接口 C++前，需要安装 C++接口
 - 机构版用户：Wind 金融终端->量化->修复插件->修复 C++接口
 - 个人版用户：Wind 资讯量化接口->安装插件->安装 C++接口

2.2.2 配置方法

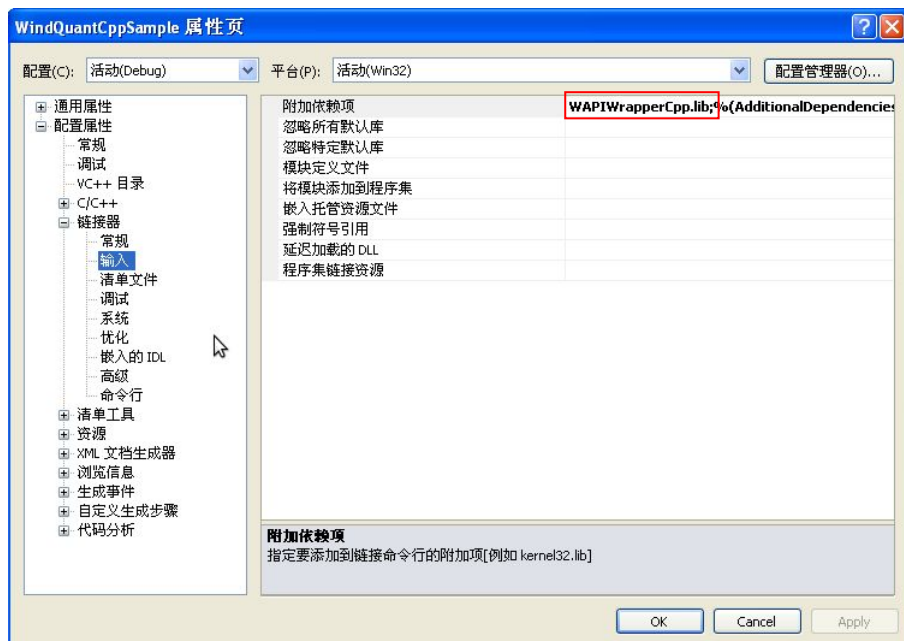
本手册以 32 位 Windows 系统和 VS2010 为运行和开发环境来配置一个 MFC 工程：

- 考虑到便于接口库文件的管理，我们可以把依赖的 API 复制到专门的文件夹中，并把文件夹拷贝到新建的工程文件夹中（此条可根据个人情况自行决定）。
- 项目--属性（注意要先点击已经生成的工程）
 - 1> 配置属性--常规--字符集 “使用 Unicode 字符集”，点击“应用”
 - 配置属性--常规—MFC 使用 “在共享 DLL 中使用 MFC”，点击“应用”

2>、配置属性--C++--常规--附加包含目录--下拉框<编辑...>--空白处单击—根据API存放路径选择，点击“应用”



5>、链接器--输入--附加依赖项--下拉框<编辑...>--复制 lib 路径到此，点击“应用”



需要注意的是：程序调用（包括隐式和显示调用）的 dll 是绑定在 wind 终端的安装文件里的。每次升级过终端，最好都将程序再次编译一下，以免运行出错。

2.3 详细使用文档的获取方式

您还可以通过如下方式获得 Wind 量化接口帮助文档：

Wind 金融终端用户：打开 Wind 资讯金融终端，点击“量化”菜单下的“开放接口”，点击“开放接口”下的“C++接口”，点击进入相应页面，找到“下载”位置，即可获取。



量化接口个人版用户请前往 <http://www.daijiangzhang.com/download> 下载帮助文档

3 C++接口应用示例

利用 VC++ 开发工具，执行该例子程序（数据接口的例子程序可以执行，交易接口的例子程序不可执行），您可以获得 wsi、wsd、wsq、wss 等数据信息，同时，通过**熟悉示例代码**，您可以更快地**掌握各种 API 接口函数**，以备在 Wind 提供的量化平台上，利用 C++ 接口函数，写出自己的查询程序和量化策略，甚至高频交易的策略。

3.1 示例代码存放位置及环境要求

3.1.1 示例代码存放位置

按 1.1 节说明，示例代码位于 WAPIWrapperCPP\Samples

数据接口示例文件：Samples\WindQuantCppSample

交易接口实例文件：Samples\WindTradeCppSample

3.1.2 系统环境要求：

- Windows 系统，VC++ 开发工具，建议使用较新版；
- Wind 终端最新版本；
- 本机的管理员权限；

3.2 数据接口示例代码执行和主要代码实现过程展示

3.2.1 运行程序，登录执行

可以直接运行

WAPIWrapperCPP\Samples\WindQuantCppSample\Release\WindQuantCppSample.exe

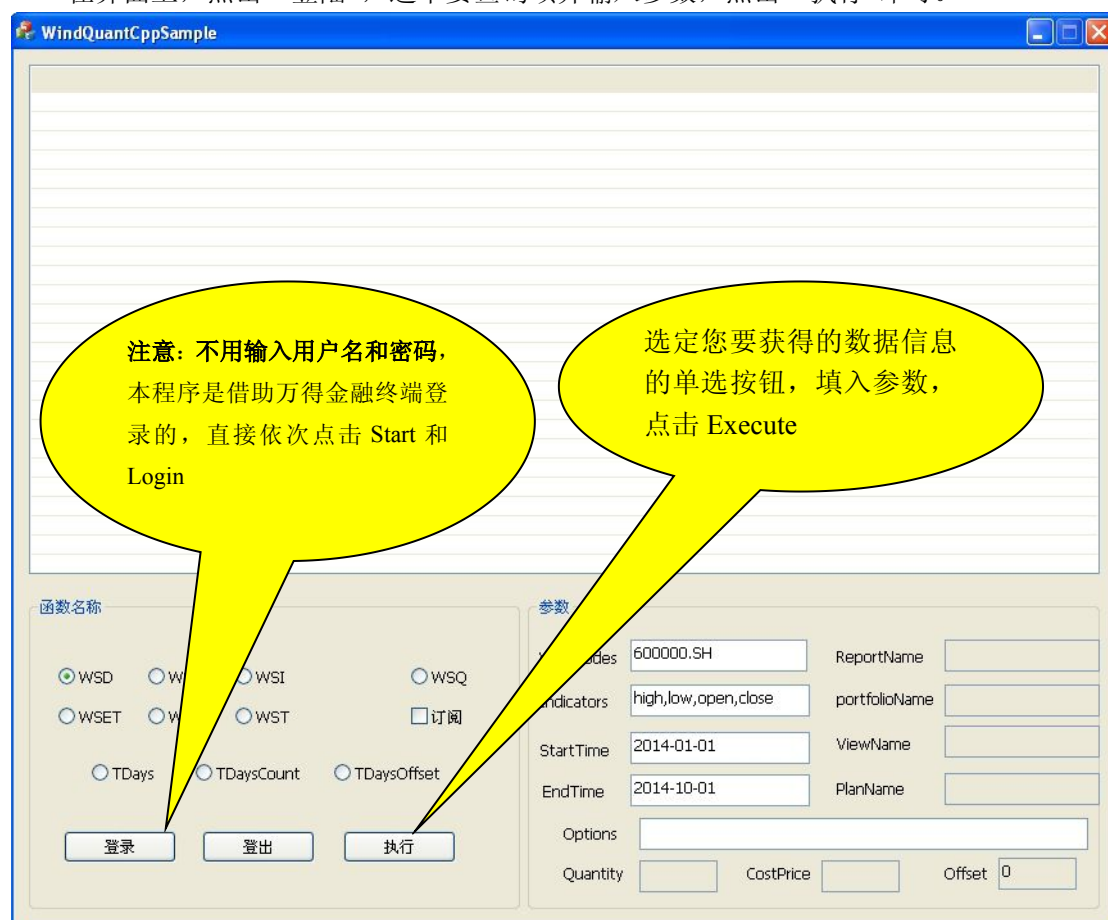
或者自行用 VC++编译后运行

在 vc++中（尽量选用高版本的工具）打开

WAPIWrapperCPP\Samples\WindQuantCppSample\WindQuantCppSample.sln 编译成功后，运行程序。

运行步骤:

在界面上，点击“登陆”，选中要查询项并输入参数，点击“执行”即可。



部分数据函数作用展示:

WSD	多值函数: 日期序列
WSI	多值函数: 分钟序列
WSS	多值函数: 历史快照
WST	多值函数: 日内跳价
WSQ	多值函数: 实时行情
WSET	多值函数: 数据集

3.2.2 部分关键代码展示

1) 点击登陆。

如代码段 1 所示，将执行 CWindQuantCppSampleDlg::OnBnClickedButton1() 运行 CWAPIWrapperCpp::start()。CWAPIWrapperCpp::start() 是 Wind 登陆函数。

2) 点击执行。

如代码段 2 所示，如果选定的是 WSD 这个单选按钮，将进入 CWindQuantCppSampleDlg::OnBnClickedButton3() 执行 CWindWrapperCpp::wsd(wdata, m_windCodes, m_indicators, m_startTime, m_endTime);

CWindWrapperCpp::wsd(wdata, m_windCodes, m_indicators, m_startTime, m_endTime) 用于获取日行情。输入参数包括万得代码、指标、日期；取到的日行情数据放在 wdata 中。

3.2.3 WindData 数据结构

WindData 是 Wind 数据接口、交易接口、回测接口返回的数据。数据结构如代码段 3 所示。

WindData 的成员变量 data 存放了返回的数据，是三维数组。三个维度分别为万得代码个数；指标个数；记录条数（天数，分钟数或 Tick 数）。可以通过成员函数 GetDataItem 获取数组中每个元素，并用 CWindQuantCppSampleDlg::toString 转化为字符串输出（代码段 4）。

代码段 1:

```
void CWindQuantCppSampleDlg::OnBnClickedButton1()
{
    // TODO: 在此添加控件通知处理程序代码
    long errCode = CWAPIWrapperCpp::start(); Wind 登陆认证
    if (errCode == 0)
    {
        MessageBox(L"登录数据接口成功!", L"提示", MB_OK);
    }
    else
    {
        WCHAR buffer[128];
        int bufferSize = 128;
        CWAPIWrapperCpp::getErrMsg(errCode, eCHN, buffer, bufferSize);
        MessageBox(buffer, L"登录失败", MB_OK);
    }
}
```

代码段 2:

```
void CWindQuantCppSampleDlg::OnBnClickedButton3()
{
    // TODO: 在此添加控件通知处理程序代码

    UpdateData(TRUE);

    long errCode = 0;
    WindData wdata;
    switch(m_func)
    {
        case 0://wsd      取日行情
        {
            errCode = CWAPIWrapperCpp::wsd(wdata, m_windCodes, m_indicators, m_startTime, m_endTime);
            if (errCode == 0)
            {
                if (wdata.GetFieldsLength() > 1)
                {
                    updateList(wdata);      返回的行情数据
                }
                else
                {
                    updateList(wdata);
                }
            }
        }
        else
        {
            //          WCHAR buffer[128] = {0};
            //          int bufferSize = 128;
            //          CWAPIWrapperCpp::getErrMsg(errCode, eCHN, buffer, bufferSize);
            //          MessageBox(buffer, L"提示", MB_OK);
            MessageBox(wdata.GetErrMsg(), L"提示", MB_OK);
        }
    }
}
```

输入参数：万得代码，指标名，始末时间

代码段3

```

class WAPIWRAPPERCPP_EXP WindData
{
public:
    VARIANT data;    ///< 数据
    VARIANT codes;    ///< Code列表
    VARIANT fields;   ///< 指标列表
    VARIANT times;    ///< 时间列表
    LONG errorCode;   ///< 错误码
public:
    |
    WindData();
    ~WindData();

    VOID InitWindData();
    VOID ClearWindData();

    //获取错误信息
    LPCWSTR GetErrorMsg() const;

    //日期转化函数(需要delete)
    static WCHAR* DateToString(DATE date, LPCWSTR strFormat = L"%Y-%m-%d");
    static VOID FreeString(WCHAR*& pStr);

    //////////////////////////////////////
    //数据使用

    INT GetCodesLength() const;
    INT GetFieldsLength() const;
    INT GetTimesLength() const;

    LPCWSTR GetCodeByIndex(int index) const;
    LPCWSTR GetFieldsByIndex(int index) const;
    BOOL GetTimeByIndex(int index, WCHAR timeBuffer[], int& length) const;
    DATE GetTimeByIndex(int index) const;

    //获取数据查询接口OneData
    BOOL GetDataItem(int timeIndex, int codesIndex, int fieldsIndex, VARIANT& outItem) const;

    //获取日期列表
    const DATE* GetTDaysInfo(LONG& lDateCount) const;

    //////////////////////////////////////
    //交易使用

    INT GetRecordCount() const;

    //获取交易接口OneData
    BOOL GetTradeItem(int recordIndex, int fieldsIndex, VARIANT& outItem) const;

    //获取登录ID
    LONG GetLogonID() const;

    //获取Order请求Id
    LONG GetOrderRequestID() const;

    //获取OrderNumber
    LPCWSTR GetOrderNumber(int recordIndex = 0) const;
};

```

取出数据接口返回的数据

取出交易接口返回的数据

代码段4

```
CString CWindQuantCppSampleDlg::toString(const LPVARIANT data)
{
    CString msg;
    switch(data->vt & VT_BSTR_BLOB)
    {
        case VT_I4:
            msg.Format(_T("%d"), data->intVal);
            break;
        case VT_I8:
            msg.Format(_T("%I64d"), data->llVal);
            break;
        case VT_R8:
            msg.Format(_T("%f"), data->dblVal);
            break;
        case VT_EMPTY:
            msg = _T("NaN");
            break;
        case VT_BSTR:
            msg.Format(_T("%s"), data->bstrVal);
            break;
        case VT_DATE:
            msg = COleDateTime(data->date).Format();
            break;
        default:
            break;
    }
    return msg;
}
```

3.3 交易接口示例主要代码实现过程展示

示例代码提供了一个 WindTradeCppSample.cpp 文件，_tmain 函数里的函数（代码段 1），展示了代码的实现过程，依次完成：登陆 Wind，交易账户登录，下单，撤单，持仓查询，退出证券账户，退出 Wind。客户可以通过此例子了解交易过程，掌握相关接口的使用，以备在 Wind 交易平台上实现自己程序化交易。

代码段 1:

```

int _tmain(int argc, _TCHAR* argv[])
{
    wcout.imbue(locale("chs"));

    if (!testLogonWFT())
    {
        return 0;
    }
    if (!testLogonAccount())
    {
        return 0;
    }

    testOrder();
    testCancelOrder();

    testQueryPosition();

    testLogoutAccount();
    testLogoutWFT();

    system("pause");
    return 0;
}

```

代码段 2:

```

void testOrder()
{
    cout << "测试下单: " << endl;
    WCHAR* errMsg = NULL;
    WindData wd;
    long errorCode = CWAPIWrapperCpp::torder(wd, L"600000.SH", L"Buy", L"10.3", L"200", L""); //单账户登录可以不指定LogonId
    if (errorCode == 0) //如果没有错误, errMsg仍然为NULL
    {
        reqID1 = wd.GetOrderRequestID();
        cout << "下单 600000.SH 成功, 请求号为: " << reqID1 << endl;
    }
    else
    {
        wcout << wd.GetErrorMsg() << endl;
    }

    errorCode = CWAPIWrapperCpp::torder(reqID2, L"600004.SH", L"Buy", 10.0, 300, L"", &errMsg);
    if (errorCode == 0) //如果没有错误, errMsg仍然为NULL
    {
        cout << "下单 600004.SH 成功, 请求号为: " << reqID2 << endl;
    }
    else
    {
        wcout << errMsg << endl;
        //需要手动释放
        delete[] errMsg;
        errMsg = NULL;
    }

    cout << endl;
}

```

返回的下单相关信息

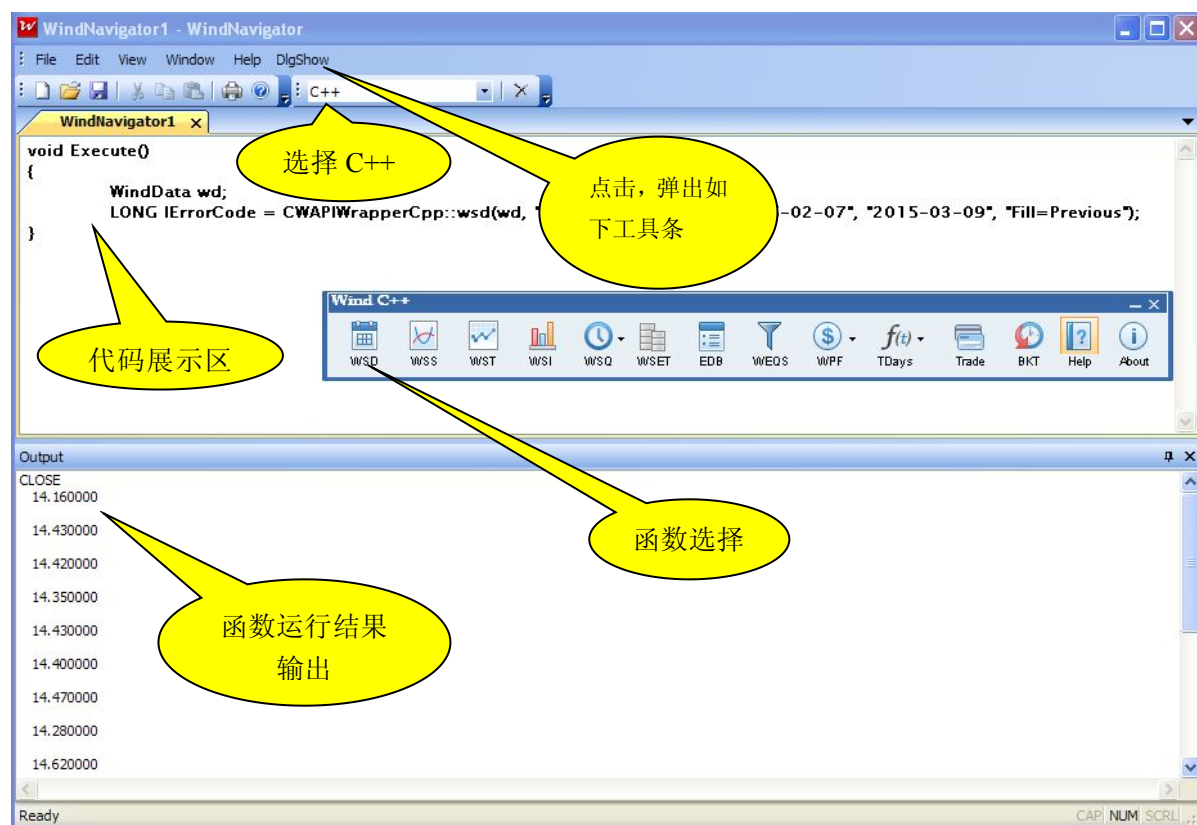
输入参数: 万得代码, 交易方向, 委托价格, 委托数量

4 C++代码生成器使用说明

WIND 提供 WindNavigator 导航界面，为 C++产生相应代码，是 Wind 数据及交易接口的帮助，用于熟悉各种接口及命令。命令生成器还可直接输出函数运行结果。

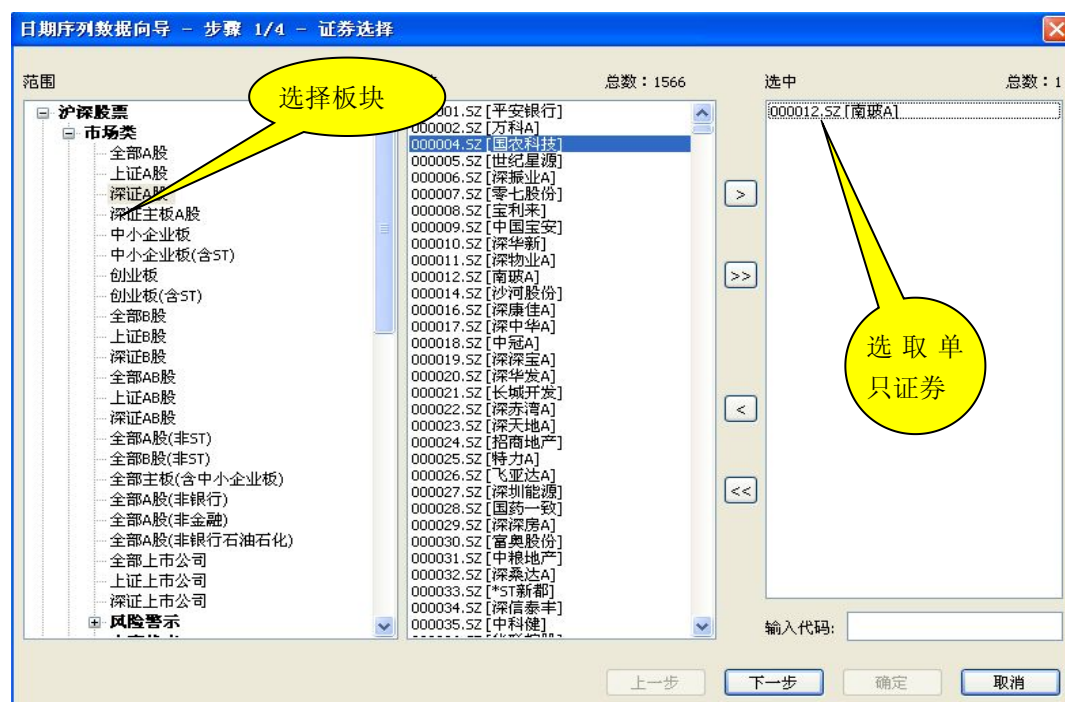
4.1 WindNavigator 导航(命令生成器)

WindNavigator 在 C:\Wind\Wind.NET.Client\WindNET\bin 路径下(可以发送快捷方式到桌面，方便使用)。用户选择接口，然后选取函数并设置参数完毕，界面会显示相应自动生成代码。注意启动该软件时，您的 Wind 终端要处于登录状态。



4.2 部分应用截图

4.2.1 WSD 代码生成截图：



点击下一步，选择指标。



点击下一步，设置时间。

日期序列数据向导 - 步骤 3/4 - 时间设置

周期

☒ 日 ☐ 周 ☐ 月 ☐ 季 (报告期) ☐ 年

时间设置

☒ 固定日期 ☐ 相对日期

起始日期: 2014- 5-13 截止日期: 2014- 6-12 ☒ 当前日期

日期类型

☒ 交易日 ☐ 日历日 ☐ 工作日 证券交易所: 上海证券交易所

上一步 下一步 确定 取消

点击下一步，设置参数。

日期序列数据向导 - 步骤 4/4 - 参数设置

无交易数据处理

☒ 沿用之前数据 ☐ 空值

货币

☒ 原始货币 (ORIG) ☐ 人民币 (CNY) ☐ 美元 (USD) ☐ 港元 (HKD)

价格修正

☒ 复权方式 (适用股票基金) ☐ 价格类型 (适用债券)

☒ 不复权 ☐ 前复权

运行设置

☒ 不运行 ☐ 直接运行

如选中直接运行，则同时输出函数运行结果。

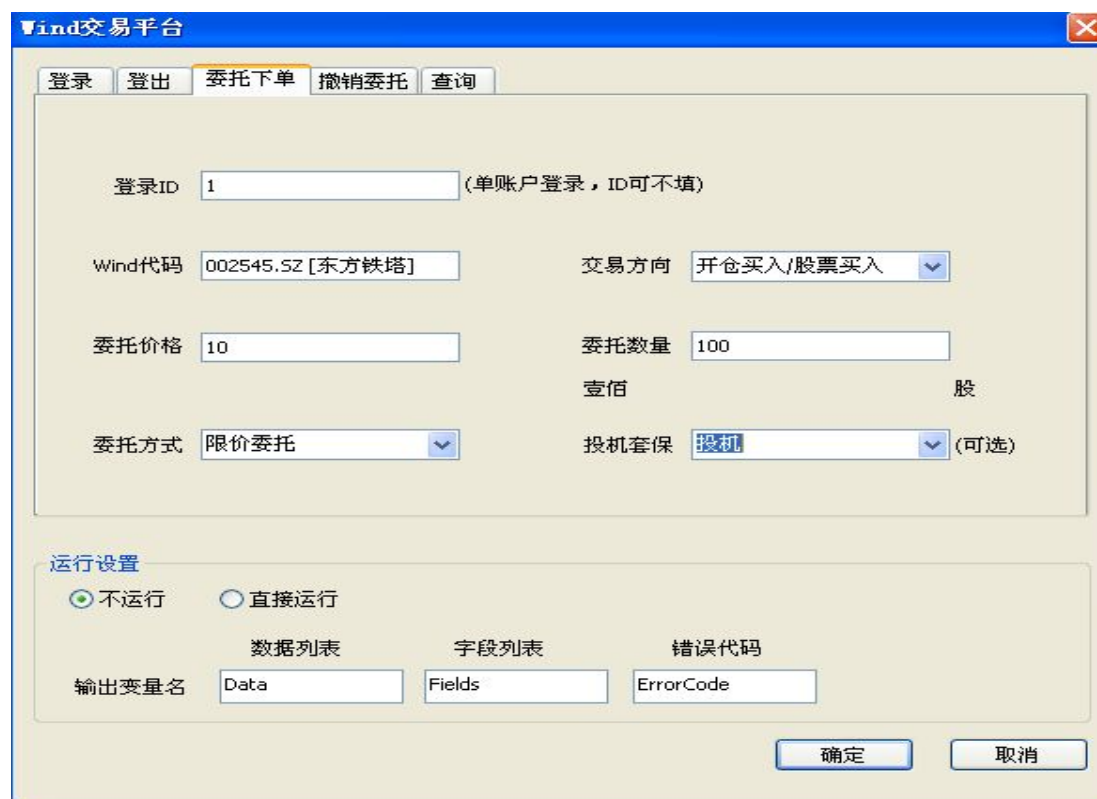
数据列表	证券代码列表	指标列表	时间列表	错误ID	请求ID
输出变量名 w_wsd_data	w_wsd_codes	w_wsd_fields	w_wsd_times	w_wsd_errorid	w_wsd_reqid

上一步 下一步 确定 取消

点击确定，生成 WSD 的回调函数代码。

4.2.2 Trade 代码生成截图

在 WindNavigator 中选取 Trade，选委托下单，在响应选项下填写。



Wind交易平台

登录 登出 委托下单 撤销委托 查询

登录ID: 1 (单账户登录, ID可不填)

Wind代码: 002545.SZ [东方铁塔] 交易方向: 开仓买入/股票买入

委托价格: 10 委托数量: 100

委托方式: 限价委托 投机套保: 投机 (可选)

运行设置

☒ 不运行 ☐ 直接运行

输出变量名: Data 数据列表: Fields 字段列表: ErrorCode 错误代码

确定 取消

在 WindNavigator 中选取 Trade，选查询。



Wind交易平台

登录 登出 委托下单 撤销委托 查询

查询类型: 当日委托查询

登录ID:

请求ID: (可选)

委托号: (可选)

委托类型: 全部

运行设置

☒ 不运行 ☐ 直接运行

输出变量名: Data 数据列表: Fields 字段列表: ErrorCode 错误代码

确定 取消

选择和填写参数

点确定，生成命令。

4.2.3 BKT 代码生成截图

输入相应参数，点击确定，生成代码。**注意：**生成前，一定要对上次生成的代码进行clear一下。

5 Wind C++接口详细说明

Wind C++ 接口封装在类CWAPIWrapperCpp中，以静态成员函数形式提供。

5.1 接口概要

5.1.1 接口功能函数列表

接口控制函数	start	Wind 接口启动
	stop	Wind 接口终止
	isconnected	判断连接状态
	getErrorMsg	获取错误码相应的错误消息
数据接口函数	wsd	多值函数：日期序列

	wsi	多值函数：分钟序列
	wss	多值函数：历史快照
	wst	多值函数：日内跳价
	wsq	多值函数：实时行情
	wset	多值函数：数据集
	wpf	资产管理系统 AMS/PMS 统计报表数据
	wupf	上传组合持仓数据到组合管理系统 PMS
	weqs	证券筛选功能
	tdays	日历日、工作日、交易日的日期序列函数
	tdaysoffset	日历日、工作日、交易日的日期偏移计算
	tdayscount	日历日、工作日、交易日的日期天数计算
	CancelRequest	取消数据请求/订阅
	CancelAllRequest	取消所有数据请求
交易接口函数	tlogin	交易帐号登陆
	tlogout	交易帐号登出
	torder	下单
	tquery	交易情况查询
	tcancel	撤单
回测接口函数	bktstart	回测开始
	bktquery	回测查询
	bktorder	回测下单
	bktend	回测结束
	bktstatus	查看回测状态
	bktsummary	回测概要
	bktdelete	回测删除
	bktstrategy	返回策略列表

注：各版本量化接口的函数使用权限，请查看：

<http://www.dajiangzhang.com/q/529d26bf-7b34-46b1-bc92-ff9a2494b0a6>

5.2 Wind 接口控制函数明细

5.2.1 Wind 接口启动函数：start

说明	接口启动函数 LONG start(LPCWSTR options = NULL, LPCWSTR options2 = NULL, LONG timeout = 5000)		
参数	定义	In/Out	描述
options	LPCWSTR	In	保留参数，默认填NULL, 暂无意义
options2	LPCWSTR	In	保留参数，默认填NULL, 暂无意义
timeout	LONG	In	timeout 毫秒没有响应，则报登陆超时
返回值	0表示执行成功，负数表示失败；可以通过getErrorMsg函数取得错误信息；		

范例	<pre> long errCode = CWAPIWrapperCpp::start(); if (errCode == 0) { MessageBox(L"登录数据接口成功!", L"提示", MB_OK); } else { WCHAR buffer[128]; int bufferSize = 128; CWAPIWrapperCpp::getErrMsg(errCode, eCHN, buffer, bufferSize); MessageBox(buffer, L"登录失败", MB_OK); } </pre>
----	--

5.2.2 Wind 接口登出函数: stop

说明	Wind 接口登出函数 LONG stop();		
参数	定义	In/Out	描述
-	-	-	-
返回值	0 表示执行成功, 负数表示失败; 可以通过 getErrMsg 函数取得错误信息。		
范例	long errCode = CWAPIWrapperCpp::stop();		

5.2.3 Wind 接口连接状态判断函数: isconnected

说明	判断 Wind 接口是否处于连接状态 LONG isconnected ();		
参数	定义	In/Out	描述
-	-	-	-
返回值	0 表示执行成功, 负数表示失败; 可以通过 getErrMsg 函数取得错误信息。		
范例	long is_connected = CWAPIWrapperCpp::isconnected ();		

5.2.4 获取错误码相应的错误信息: getErrMsg

说明	获取错误码相应的错误信息 BOOL getErrMsg(LONG errCode, ErrMsgLang lang, WCHAR msg[], int& /*inout*/msgLength);		
参数	定义	In/Out	描述
errCode	LONG	In	错误码
lang	ErrMsgLang	In	语言类型
msg	WCHAR	In/Out	获取的错误消息
msgLength	int	In/Out	错误消息长度
返回值	0 表示执行成功, 负数表示失败		

范例	<pre> LONG lErrorCode = CWAPIWrapperCpp::tlogon(wd, "11000101", "0", "000001", "123456", "SHSZ", ""); //输入错误的帐号 if (errCode != 0) { WCHAR buffer[128]; int bufferSize = 128; CWAPIWrapperCpp::getErrMsg(errCode, eCHN, buffer, bufferSize); MessageBox(buffer, L"登录失败", MB_OK); } </pre>
----	--

5.3 Wind 数据接口函数明细

5.3.1 多值函数-日期序列：WSD

说明	<p>日期序列函数，返回选定证券品种的历史数据,包括日间的行情数据，基本面数据以及技术指标数据。</p> <p>LONG wsd(WindData& outWindData, LPCWSTR windCodes, LPCWSTR fields, LPCWSTR startTime = NULL, LPCWSTR endTime = NULL, LPCWSTR options = NULL);</p>		
参数	定义	In/Out	描述
outWindData	WindData	Out	返回的日期序列数据
windcode	LPCWSTR	In	<p>只接受 WindCode。例如 _T("000001.SZ")。</p> <p>备注：多个 Windcode 之间用','分割，如_T("000001.SZ,600000.SH")</p> <p>如果用了多个 windcode，则 indicators 不能是多个，只能是单个</p>
indicators	LPCWSTR	In	<p>函数能够接受的指标名称。如果有多个，用半角逗号隔开。例如 _T("high,open,low,close")。</p>
startTime	LPCWSTR	In	<p>起始时间和日期字符串。可接受的字符串必须形如"yyyymmdd", "yyyy-mm-dd"。</p>
endTime	LPCWSTR	In	<p>截止时间和日期字符串。可接受的字符串必须形如"yyyymmdd", "yyyy-mm-dd"。</p>
options	LPCWSTR	In	<p>扩展参数，多个参数用半角分号隔开。比如"Fill=Previous;PriceAdj=F" 表示碰到空值则用前一天的数据填充；并使用前复权。可以使用 WindNavigator 来生成 WSD 命令</p>
返回值	0 表示执行成功，负数表示失败；可以通过 getErrMsg 函数取得错误信息。		
范例	<p>获取某证券在2013-04-15至2013-04-25区间内的每日开盘价、收盘价。</p> <pre> m_windcodes = _T("000001.SZ"); m_indicators = _T("open,high"); </pre>		

	<pre> m_startTime = _T("2013-04-15"); m_endTime = _T("2013-04-26"); WindData wdata; LONG errCode = CWAPIWrapperCpp::wsd(wdata, m_windCodes, m_indicators, m_startTime, m_endTime); </pre>
--	--

5.3.2 多值函数-截面数据 WSS

说明	截面函数，返回选定品种的历史截面数据，支持获取某截面日期的多品种多指标。 LONG wss(WindData& outWindData, LPCWSTR windCodes, LPCWSTR fields, LPCWSTR options = NULL);		
参数	定义	In/Out	描述
outWindData	WindData	In	返回的截面数据
windcode	LPCWSTR	In	只接受 WindCode。如果有多个，用半角逗号隔开。例如"000001.SZ,000002.SZ"。 备注：WSS 支持多品种。
indicators	LPCWSTR	In	函数能够接受的指标名称。如果有多个，用半角逗号隔开。例如 "high,open,low,close"。
options	LPCWSTR	In	扩展参数，多个参数用半角分号隔开。例如"param1=value1;param2=value2",
返回值	0 表示执行成功，负数表示失败；可以通过 getErrorMsg 函数取得错误信息。		
范例	获取2013-08-09某两只证券的开盘价、最高价、最低价、收盘价、成交量。 <pre> m_windCodes = _T("000001.SZ, 000002.SZ"); m_indicators = _T("open, high, low, close, volume"); m_options = _T("tradeDate=20130809");//交易日 WindData wdata; LONG errCode = CWAPIWrapperCpp::wss(wdata, m_windCodes, m_indicators, m_options); </pre>		

5.3.3 多值函数-分钟序列 WSI

说明	分钟序列函数，返回单品种的日内分钟 K 线数据(含当日数据)。 LONG wsi(WindData& outWindData, LPCWSTR windCodes, LPCWSTR fields, LPCWSTR startTime, LPCWSTR endTime, LPCWSTR options = NULL);		
参数	定义	In/Out	描述
outWindData	WindData	Out	返回的分钟序列数据
windcode	LPCWSTR	In	只接受 WindCode。WSI 目前仅支持单品种，用半角逗号隔开。例如"000001.SZ"。
indicators	LPCWSTR	In	函数能够接受的指标名称。如果有多个，

			用半角逗号隔开。例如 "high,open,low,close"。
startTime	LPCWSTR	In	起始时间和日期字符串。可接受的字符串 必须形如"yyyymmdd HHMMSS"或者 "yyyy-mm-dd HH:MM:SS"
endTime	LPCWSTR	In	截止时间和日期字符串。可接受的字符串 必须形如"yyyymmdd HHMMSS"或者 "yyyy-mm-dd HH:MM:SS"
options	LPCWSTR	In	扩展参数，多个参数用半角分号隔开，在 WSI 函数中主要设置指标参数。例如 'BOLL_N=26;BOLL_Width=2;BOLL_IO=1; BarSize=3' 。
返回值	0 表示执行成功，负数表示失败；可以通过 getErrorMsg 函数取得错误信息。		
范例	<p>获取2013-08-09某证券的3分钟时间序列的开盘价、最高价、最低价、收盘价、成交量、布林值数据。</p> <pre> m_windcodes = _T("000001.SZ,000002.SZ"); m_indicators = _T("open,high,low,close,volume,BOLL"); m_startTime = _T("2013-08-09 09:30:00"); m_endTime = _T("2013-08-09 15:00:00"); //设置技术指标参数，BarSize=3表示3分钟级别数据。 M_options = _T("BOLL_N=26;BOLL_Width=2;BOLL_IO=1;BarSize=3"); WindData wdata; LONG errCode = CWAPIWrapperCpp::wsi(wdata, m_windCodes, m_indicators, m_startTime, m_endTime, m_options); </pre>		

5.3.4 多值函数-日内跳价 WST

说明	日内跳价函数，返回当日日内盘口快照数据和成交数据。 LONG wst(WindData& outWindData, LPCWSTR windCodes, LPCWSTR fields, LPCWSTR startTime, LPCWSTR endTime, LPCWSTR options = NULL);		
参数	定义	In/Out	描述
outWindData	WindData	Out	返回的日内 tick 数据
windcode	LPCWSTR	In	只接受 WindCode。WSI 目前仅支持单品种，用半角逗号隔开。例如"000001.SZ"。
indicators	LPCWSTR	In	函数能够接受的指标名称。如果有多个，用半角逗号隔开。例如 "high,open,low,close"。
beginTime	LPCWSTR	In	起始时间和日期字符串。可接受的字符串 必须形如"yyyymmdd HHMMSS"或者 "yyyy-mm-dd HH:MM:SS"
endTime	LPCWSTR	In	截止时间和日期字符串。可接受的字符串 必须形如"yyyymmdd HHMMSS"或者

			"yyyy-mm-dd HH:MM:SS"
options	LPCWSTR	In	扩展参数，多个参数用半角分号隔开，在 WST 函数中待扩展。
返回值	0 表示执行成功，负数表示失败；可以通过 GetLastError 函数取得错误信息。		
范例	<p>获取当日某证券的开盘价、最高价、最低价、收盘价、成交量等日内数据。</p> <pre> m_windCodes = _T("000001.SZ"); m_indicators = _T("open,high,low,close,volume"); m_startTime = _T("2013-08-09 09:30:00"); m_endTime = _T("2013-08-09 15:00:00"); m_options = _T(""); WindData wdata; LONG errCode = CWAPIWrapperCpp::wst(wdata, m_windCodes, m_indicators, m_startTime, m_endTime, m_options); </pre>		

5.3.5 日期宏

5.3.5.1 通用日期宏

支持相对日期表达方式，相对日期周期包括:TD/D/W/M/Q/S/Y,即交易日/日历日/日历周/日历月/日历季/日历半年/日历年

➤ 以“-”代表前推，数字代表 N 个周期，只支持整数；后推没有负号；

比如“-5D”表示从当前最新日期前推 5 个日历日；

➤ 截止日期若为“”空值，取系统当前日期；

➤ 可对日期宏进行加减运算，比如“ED-10d”；

举例：

1. 起始日期为 1 个月前，截至日期为最新

StartDate="-1W",EndDate=""

2. 起始日期为前推 10 个交易日，截至日期为前推 5 个交易日

StartDate="-10TD",EndDate="-5TD"

5.3.5.2 特殊日期宏

目前条件选股，数据浏览器中有许多日期宏，数据接口支持这些日期宏,整理如下：

宏名称	宏助记符
截止日期	ED
开始日期	SD
去年一季	LQ1

去年二季	LQ2
去年三季	LQ3
去年年报	LYR
今年一季	RQ1
今年二季	RQ2
今年三季	RQ3
最新一期	MRQ
本年初	RYF
下半年初	RHYF
本月初	RMF
本周一	RWF
上周末	LWE
上月末	LME
上半年末	LHYE
上年末	LYE
上市首日	IPO

5.3.6 多值函数-实时行情 WSQ

说明	返回选定品种的实时数据，支持一次请求和订阅两种方式。 // 非订阅模式，一次取快照数据 LONG wsq(WindData& outWindData, LPCWSTR windCodes, LPCWSTR fields, LPCWSTR options = NULL); // 订阅模式，订阅实时数据，数据通过回调函数返回 LONG wsq(ULONGLONG &reqId, LPCWSTR windCodes, LPCWSTR fields, WsqCallBack callBack, LPCWSTR options = NULL, BOOL updateAllFields = FALSE);		
	参数	定义	In/Out 描述
outWindData (非订阅模式)	WindData	Out	返回的快照数据
reqId (订阅模式)	ULONGLONG	Out	请求编号
callBack (订阅模式)	WsqCallBack	In	WsqCallBack 为回调函数，类型为 typedef LONG (WINAPI *WsqCallBack) (const WindData &windData);

			每当行情有变化时，量化接口会调用 <code>callBack</code> 函数，通过参数 <code>windData</code> 把行情推送过来。用户需要自行编制 <code>callBack</code> 函数，接收和处理行情。
<code>updateAllFields</code> (订阅模式)	BOOL	In	如果 <code>updateAllFields</code> 为 True，则每次行情改变时，把所有 Field 都推送过来 如果 <code>updateAllFields</code> 为 False，则每次行情改变时，只推送发生过变化的 Field
<code>windcode</code>	LPCWSTR	In	只接受 <code>WindCode</code> 。如果有多个，用半角逗号隔开。例如 "000001.SZ,000002.SZ"。 备注：WSQ 支持多品种。
<code>fields</code>	LPCWSTR	In	函数能够接受的指标名称。如果有多个，用半角逗号隔开。例如 "high,open,low,close"。
<code>options</code>	LPCWSTR	In	扩展参数，多个参数用半角分号隔开。例如 "param1=value1;param2=value2",
返回值	0 表示执行成功，负数表示失败；可以通过 <code>getErrorMsg</code> 函数取得错误信息。		
范例	获取2013-08-09某两只证券的开盘价、最高价、最低价、收盘价、成交量。 <pre> ULONGLONG m_reqId; m_windCodes = _T("000001.SZ,000002.SZ"); m_indicators = _T("rt_last,rt_open,rt_high,rt_low"); options = _T(""); // CallBack 函数详见 3.2 节 数据接口示例代码 LONG errCode = CWAPIWrapperCpp::wsq(m_reqId, m_windCodes, m_indicators, CallBack, m_options, TRUE); </pre>		

5.3.7 多值函数-数据集 WSET

说明	返回股票，基金，债券，商品等专题统计报表的数据。 <code>LONG wset(WindData& outWindData, LPCWSTR reportName, LPCWSTR options = NULL);</code>		
参数	定义	In/Out	描述
<code>outWindData</code>	WindData	Out	返回的报表数据
<code>reportName</code>	LPCWSTR	In	数据集的表名。例如 "SectorConstituent" 表示板块成份。
<code>options</code>	LPCWSTR	In	扩展参数，多个参数用半角分号隔开。例如 "param1=value1;param2=value2",
返回值	0 表示执行成功，负数表示失败；可以通过 <code>getErrorMsg</code> 函数取得错误信息。		
范例	<pre> //取 2013 年 8 月 12 日全部 A 股的股票代码和简称 m_reportName = _T("SectorConstituent"); m_options = _T("date=20130812;sector=全部 A 股"); </pre>		

	<pre>WindData wdata; LONG errCode = CWAPIWrapperCpp::wset(wdata, m_reportName, m_options);</pre>
--	--

5.3.8 组合报表函数: wpf

说明	返回资产管理系统 AMS/PMS 统计报表数据。 LONG wpf(WindData& outWindData, LPCWSTR portfolioName, LPCWSTR viewName, LPCWSTR options = NULL);		
参数	定义	In/Out	描述
outWindData	WindData	Out	返回的报表数据
portfolioName	LPCWSTR	In	提取数据集的组合 ID 或组合名称（在 AMS 系统中是产品名称）。例如："武当一期"
viewName	LPCWSTR	In	提取数据集的报表名称，例如："HoldingDaily"
options	LPCWSTR	In	扩展参数，多个参数用半角分号隔开。例如"param1=value1;param2=value2"，
返回值	0 表示执行成功，负数表示失败；可以通过 getErrorMsg 函数取得错误信息。		
范例	<pre>// 获取“组合管理演示”产品中组合日结算数据。 m_portfolioName = _T("组合管理演示"); m_viewName = _T("PMS.PortfolioDaily"); m_options = _T("startdate=20130509;owner=1178621"); WindData wdata; LONG errCode = CWAPIWrapperCpp::wpf(wdata, m_portfolioName, m_viewName, m_options);</pre>		

5.3.9 组合上传函数: wupf

说明	上传组合持仓数据到组合管理系统PMS。 LONG wupf(WindData& outWindData, LPCWSTR portfolioName, LPCWSTR tradeDate, LPCWSTR windCodes, LPCWSTR quantity, LPCWSTR costPrice, LPCWSTR options = NULL);		
参数	定义	In/Out	描述
outWindData	WindData	Out	返回的数据
portfolioName	LPCWSTR	In	要上传持仓的组合名称。例如："武当一期"。此组合必须已经存在。
tradeDate	LPCWSTR	In	持仓日期，例如："20140210"。

windCodes	LPCWSTR	In	持仓品种的windcode，例如： "000001.SZ"。
quantity	LPCWSTR	In	该品种的持仓数量，股票为股，期货为手，现金为元，必须为整数。例如： "1100"。
costPrice	LPCWSTR	In	成本价，可以为负，可以留空（取当日收盘价）。例如："13.81"。
options	LPCWSTR	In	扩展参数，多个参数用半角分号隔开。 例如 "option1=value1;option2=value2"
options取值	Direction	In	买卖方向，Long 为做多，Short 为做空，默认为做多。
	HedgeType	In	投机套保，Spec 为投机，Hedge 为套保，默认为投机。
	Owner	In	如果此组合为他人共享的可写组合，使用该参数提供组合所有者的windID。
范例	<pre>//上传20140212的两笔持仓数据。 WindData wdata; LONG errCode = CWAPIWrapperCpp::wupf(_T("大波段"), _T("20140716, 20140716 "), _T(" 000001. SZ, 000002. SZ"), _T("100, 300"), _T("15, 8. 88"), _T("Direction=Long, Long;HedgeType=Spec, Spec"))</pre>		

5.3.10 日期序列函数 TDays

说明	返回日历日、工作日、交易日的日期序列。 LONG tdays(WindData& outWindData, LPCWSTR startTime, LPCWSTR endTime, LPCWSTR options = NULL);		
参数	定义	In/Out	描述
outWindData	WindData	Out	返回的日期序列
startTime	LPCWSTR	In	起始时间和日期字符串。可接受的字符串必须形如"yyyymmdd", "yyyy-mm-dd"。
endTime	LPCWSTR	In	截止时间和日期字符串。可接受的字符串必须形如"yyyymmdd", "yyyy-mm-dd"。
options	LPCWSTR	In	扩展参数，多个参数用半角分号隔开。例如"param1=value1;param2=value2",
返回值	0 表示执行成功，负数表示失败；可以通过 getErrorMsg 函数取得错误信息。		
范例	<pre>//获取2013年所有交易日。 WindData wdata; LONG rid=TDays(wdata, _T("20130101"), _T("20131231"), NULL)</pre>		

5.3.11 根据偏移计算日期的函数 TDaysOffset

说明	给定一个日期和偏移量，求另一个日期。 LONG tdaysoffset (DATE& outDate, LPCWSTR startTime, LONG offset, LPCWSTR options = NULL);		
参数	定义	In/Out	描述
outWindData	WindData	Out	返回的日期序列
beginTime	LPCWSTR	In	起始时间和日期字符串。可接受的字符串必须形如"yyyymmdd", "yyyy-mm-dd"。
options	LPCWSTR	In	扩展参数，多个参数用半角分号隔开。例如"param1=value1;param2=value2",
返回值	0 表示执行成功，负数表示失败；可以通过 GetLastError 函数取得错误信息。		
范例	<pre>// 以2013年1月1日为基点，返回前推5个交易日的日期。 DATE date = 0; LONG errCode = CWAPIWrapperCpp::tdaysoffset (date, _T("20130101"), _T("offset=-5"), NULL);</pre>		

5.3.12 计算日期天数函数 TDaysCount 函数

说明	返回日历日、工作日、交易日的日期序列。 LONG tdayscount (LONG& outCount, LPCWSTR startTime, LPCWSTR endTime, LPCWSTR options = NULL);		
参数	定义	In/Out	描述
outCount	LONG	Out	返回的日期天数
startTime	LPCTSTR	In	起始时间和日期字符串。可接受的字符串必须形如"yyyymmdd", "yyyy-mm-dd"。
endTime	LPCTSTR	In	截止时间和日期字符串。可接受的字符串必须形如"yyyymmdd", "yyyy-mm-dd"。
options	LPCTSTR	In	扩展参数，多个参数用半角分号隔开。例如"param1=value1;param2=value2",
返回值	0 表示执行成功，负数表示失败；可以通过 GetLastError 函数取得错误信息。		
范例	获取2013年1月交易日天数 <pre>LONG count = 0; LONG errCode = CWAPIWrapperCpp::tdayscount (count, _T("20130101"), _T("20130131"), _T(""));</pre>		

5.3.13 取消数据请求/订阅 CancelRequest

说明	通过 id 取消某个请求或订阅。 VOID cancelRequest (ULONG requestId);		
参数	定义	In/Out	描述

requestId	ULONGLONG	In	某请求的 ID
范例	<pre>// 先订阅wsq, 然后再取消 Windcodes = _T("000001.SZ, 000002.SZ"); Indicators = _T("rt_last, rt_open, rt_high, rt_low"); // wsqcallback 为客户自定义回调函数 ULONGLONG reqId; LONG errCode = CWAPIWrapperCpp::wsq(reqId, Windcodes, Indicators, wsqcallback, NULL); //.... CancelRequest(reqId)</pre>		

5.3.14 取消所有数据请求/订阅 CancelAllRequest

说明	取消所有请求或订阅。 VOID cancelAllRequest();		
参数	定义	In/Out	描述
无			
范例	<pre>//用户先订阅某些内容 //... CancelAllRequest()</pre>		

5.3.15 数据接口错误类型

Wind 数据接口错误信息定义

错误标识	错误描述	错误代码
WQERR_OK		0
WQERR_BASE		-40520000
WQERR_BASE	一般性错误	-40520000
WQERR_UNKNOWN	未知错误	-40520001
WQERR_INTERNAL_ERROR	内部错误	-40520002
WQERR_SYSTEM_REASON	操作系统原因	-40520003
WQERR_LOGON_FAILED	登陆失败	-40520004
WQERR_LOGON_NOAUTH	无登陆权限	-40520005
WQERR_USER_CANCEL	用户取消	-40520006
WQERR_NO_DATA_AVAILABLE	没有可用数据	-40520007
WQERR_TIMEOUT	请求超时	-40520008
WQERR_LOST_WBOX	Wbox 错误	-40520009
WQERR_ITEM_NOT_FOUND	未找到相关内容	-40520010
WQERR_SERVICE_NOT_FOUND	未找到相关服务	-40520011
WQERR_ID_NOT_FOUND	未找到相关 ID	-40520012
WQERR_LOGON_CONFLICT	已在本机使用其他账号登录万得其	-40520013

	他产，故无法使用指定账号登录	
WQERR_LOGON_NO_WIM	未登录使用 WIM 工具，故无法登录	-40520014
WQERR_TOO_MANY_LOGON_FAILURE	连续登录失败次数过多	-40520015
WQERR_IOERROR_CLASS	网络数据存取错误	-40521000
WQERR_IO_ERROR	IO 操作错误	-40521001
WQERR_SERVICE_NOT_AVAL	后台服务器不可用	-40521002
WQERR_CONNECT_FAILED	网络连接失败	-40521003
WQERR_SEND_FAILED	请求发送失败	-40521004
WQERR_RECEIVE_FAILED	数据接收失败	-40521005
WQERR_NETWORK_ERROR	网络错误	-40521006
WQERR_SERVER_REFUSED	服务器拒绝请求	-40521007
WQERR_SVR_BAD_RESPONSE	错误的应答	-40521008
WQERR_DECODE_FAILED	数据解码失败	-40521009
WQERR_INTERNET_TIMEOUT	网络超时	-40521010
WQERR_ACCESS_FREQUENTLY	频繁访问	-40521011
WQERR_SERVER_INTERNAL_ERROR	服务器内部错误	-40521012
WQERR_INVALID_CLASS	请求输入错误	-40522000
WQERR_ILLEGAL_SESSION	无合法会话	-40522001
WQERR_ILLEGAL_SERVICE	非法数据服务	-40522002
WQERR_ILLEGAL_REQUEST	非法请求	-40522003
WQERR_WINDCODE_SYNTAX_ERR	万得代码语法错误	-40522004
WQERR_ILLEGAL_WINDCODE	不支持的万得代码	-40522005
WQERR_INDICATOR_SYNTAX_ERR	指标语法错误	-40522006
WQERR_ILLEGAL_INDICATOR	不支持的指标	-40522007
WQERR_OPTION_SYNTAX_ERR	指标参数语法错误	-40522008
WQERR_ILLEGAL_OPTION	不支持的指标参数	-40522009
WQERR_DATE_TIME_SYNTAX_ERR	日期与时间语法错误	-40522010
WQERR_INVALID_DATE_TIME	不支持的日期与时间	-40522011
WQERR_ILLEGAL_ARG	不支持的请求参数	-40522012
WQERR_INDEX_OUT_OF_RANGE	数组下标越界	-40522013
WQERR_DUPLICATE_WQID	重复的 WQID	-40522014
WQERR_UNSUPPORTED_NOAUTH	请求无相应权限	-40522015
WQERR_UNSUPPORTED_DATA_TYPE	不支持的数据类型	-40522016
WQERR_DATA_QUOTA_EXCEED	数据提取量超限	-40522017
WQERR_ILLEGAL_ARG_COMBINATION	不支持的请求参数	-40522018

5.4 Wind 交易接口函数明细

5.4.1 Wind 交易登录函数 tlogon

说明	Wind交易登录函数tlogon LONG tlogon(WindData& outWindData, LPCWSTR brokerID, LPCWSTR
----	--

	departmentID, LPCWSTR accountID, LPCWSTR password, LPCWSTR accountType, LPCWSTR options = NULL);		
参数	定义	In/Out	描述
outWindData	WindData	Out	返回的登陆相关信息，包括登陆ID等。
brokerID	LPCWSTR	In	券商代码
departmentID	LPCWSTR	In	营业部代码 (绝大部分证券和期货柜台无需指定营业部代码，即填0；请使用命令生成器确认详情)
accountID	LPCWSTR	In	资金账号
password	LPCWSTR	In	账号密码
accountType	LPCWSTR	In	账号类型，照下表填写
			SHSZ 沪深A股
			SHB 沪市B股
			SZB 深市B股
			CFE 中金所期货
			SHF 上期所期货
			DCE 大商所期货
			CZC 郑商所期货
			SHO 上交所期权
options	LPCWSTR	In	扩展参数，多个参数用半角分号隔开。例如 "param1=value1;param2=value2",
注： 1、可以通过命令生成器生成登陆函数，自动包含券商代码、营业部代码、帐号类型等参数 2、支持同时登陆多家交易所，通过登陆ID区分。CFE,SHF,DCE,CZC只需登陆一家即可。 3、如果使用模拟交易，密码可以随意。对于机构版，模拟交易资金帐号可以从WTTTS中查看，如果Wind金融终端账号是W0813652，那么W081365201是证券模拟账号，W081316502是期货模拟账号，W081316503是衍生品模拟账号；对于个人版，可以登陆大奖章，从个人中心->模拟交易中查看模拟交易资金帐号。			
返回值	0表示执行成功，其他表示失败；可以通过getErrorMsg函数取得错误信息。		
范例	<pre> cout << "输入模拟交易账户:"; WCHAR strUserName[32] = {0}; wcin>> strUserName; WindData wd; long errorCode = CWAPIWrapperCpp::tlogon(wd, L"0000", L"0", strUserName, L"000000", L"SHSZ"); if (errorCode == 0) { logonID = wd.GetLogonID(); cout << "登录交易账户成功，登录 ID: " << logonID << endl; return true; } </pre>		

	<pre> else { wcout << wd.GetErrorMsg() << endl; return false; } </pre>
--	--

5.4.2 经纪商登出函数 tlogout

说明	经纪商登出函数 tlogout LONG tlogout(INT longId=0);		
参数	定义	In/Out	描述
LogonID	INT	In	登录 ID, tlogin 返回的登录 ID
返回值	0表示执行成功, 其他表示失败; 可以通过getErrorMsg函数取得错误信息。		
范例	<pre> long errorCode = CWAPIWrapperCpp::tlogout(logonID); if (errorCode == 0) { cout << "登出交易账户成功" << endl; } else { WCHAR buffer[64]; int length = 64; CWAPIWrapperCpp::getErrorMsg(errorCode, eCHN, buffer, length); wcout << buffer << endl; } cout << endl; </pre>		

5.4.3 委托下单函数 torder

说明	<pre> // 支持同时下多个单 LONG torder(WindData& outWindData, LPCWSTR windCodes, LPCWSTR tradeSide, LPCWSTR orderPrice, LPCWSTR orderVolume, LPCWSTR options = NULL); // 只能下单个单 LONG torder(LONG& reqId, LPCWSTR windCodes, LPCWSTR tradeSide, DOUBLE orderPrice, LONG orderVolume, LPCWSTR options = NULL, WCHAR** pErrosMsg = NULL); </pre>		
参数	定义	In/Out	描述
reqId	LONG	Out	返回的该Order的请求Id
pErrosMsg	WCHAR	Out	返回的错误消息
outWindData	WindData	Out	返回的该Order信息
windCodes	LPCWSTR	In	交易代码 (证券代码/期货合约代码)
TradeSide	LPCWSTR	In	交易方向
OrderPrice	LPCWSTR	In	委托价格

OrderVolume	LPCWSTR	In	委托数量
options	LPCWSTR	In	扩展参数 可使用命令生成器查阅相关参数
返回值	WD_ERR: 0表示执行成功, 其他表示失败; 可以通过getErrorMsg函数取得错误信息。		
范例	<pre> LONG reqId = 0;//Order请求Id, 后面Query时用 WCHAR* pErrMsg = NULL;//下单出错信息 LONG lErrorCode = CWAPIWrapperCpp::torder(reqId, "600000.SH", "Buy", "14.5", "100", "OrderType=LMT;LogonID=1", &pErrMsg);//不需要返回错误信息, pErrMsg位置可以不填 if (NULL != pErrMsg) { CString strOrderErrMsg = pErrMsg; WindData::FreeString(pErrMsg);//不要忘记Free! } // 可以用 tquery 函数, 查询该笔委托的成交情况 </pre>		

5.4.4 撤销委托函数 tcancel

说明	撤销委托函数 LONG tcancel(LPCWSTR orderNumber, LPCWSTR options = NULL);		
参数	定义	In/Out	描述
OrderNumber	LPCWSTR	In	委托序号 委托下单后可查询委托序号
options	LPCWSTR	In	扩展参数 可使用命令生成器查阅相关参数
返回值	0表示执行成功, 其他表示失败; 可以通过getErrorMsg函数取得错误信息。		
范例	<pre> cout << "测试撤单: " << endl; WindData wd; long errorCode = CWAPIWrapperCpp::tquery(wd, L"Order", L"OrderType=Withdrawable"); if (errorCode == 0) { long lRecordCount = wd.GetRecordCount(); if (lRecordCount > 0) { CString strOrderNumber = wd.GetOrderNumber(0); errorCode = CWAPIWrapperCpp::tcancel(strOrderNumber); } } </pre>		

	<pre> if (errorCode == 0) { cout << "撤单成功" << endl; } } else { wcout << wd.GetErrorMsg() << endl; } cout << endl; </pre>
--	---

5.4.5 查询函数 tquery

说明	LONG tquery(WindData& outWindData, LPCWSTR qryCode, LPCWSTR options = NULL);		
参数	定义	In/Out	描述
outWindData	WindData	Out	返回的数据
qryCode	LPCWSTR	In	查询代码
options	LPCWSTR	In	扩展参数，多个参数之间用’，’ 分隔
Qrycode 参数 (为必选参数)	查询内容	可选扩展参数	
Capital	资金查询	LogonID	
Position	持仓查询	LogonID, WindCode	
Order	当日委托查询	LogonID, RequestID, OrderNumber, WindCode	
Trade	当日成交查询	LogonID, OrderNumber, WindCode	
Department	营业部查询	BrokerID	
Account	账号查询	LogonID	
Broker	经纪商查询	无	
LogonID	登录号查询	无	
返回值	0表示执行成功，其他表示失败；可以通过getErrorMsg函数取得错误信息。		
范例	<pre>WindData wd; LONG errorCode = CWAPIWrapperCpp::tquery(wd, "Position", "LogonId=1"); //获取Query数据, 多个Record时循环去取 if (errorCode == 0) { long lRecordCount = wd.GetRecordCount(); for (int i = 0; i < lRecordCount; i++) {</pre>		

```

int fl = wd.GetFieldsLength();
for (int j = 0; j < fl; j++)
{
    VARIANT var;
    wd.GetTradeItem(i, j, var);
    CString str = toString(&var);
    wcout << str.GetBuffer(0) << endl;
}
}
else
{
    wcout << wd.GetErrorMsg() << endl;
}

```

5.4.6 交易接口错误类型

Wind 交易接口错误信息定义

错误标识	错误描述	错误代码
WD_ERR_Success	正确	0
WD_ERR_Base		-40530000
WD_ERR_DataErr	数据错误	-40530001
WD_ERR_Uninit	未初始化	-40530002
WD_ERR_FuncID	功能号错	-40530003
WD_ERR_LogonID	LogonID 错（或未输入此参	-40530004
WD_ERR_BufferOver	回报队列已满	-40530005
WD_ERR_SendRequestFailed	发送请求失败	-40530006
WD_ERR_UnAuthorize	未通过认证	-40530101
WD_ERR_LOGON	登录错误	-40530102
WD_ERR_NETWORK	网络错误	-40530103
WD_ERR_ORDER	委托错误	-40530104
WD_ERR_CANCEL	撤单错误	-40530105
WD_ERR_QUERY	查询错误	-40530106
WD_ERR_COVEREDCHG	备兑证券划转错误	-40530107
WD_ERR_Config	获取配置错误	-40530201
WD_ERR_BrokerID	券商（期货商）代码错	-40530202
WD_ERR_LogonAccount	登录代码错	-40530203
WD_ERR_Password	账号密码错	-40530204
WD_ERR_AccountType	账号类型错	-40530205
WD_ERR_LogonCountOver	登录失败次数过多	-40530206

WD_ERR_ConnectFailed	建立连接失败	-40530301
WD_ERR_TimeOut	处理超时	-40530302
WD_ERR_SecurityCode	交易代码错	-40530401
WD_ERR_OrderType	价格委托方式错	-40530402
WD_ERR_OrderVolume	委托数量错	-40530403
WD_ERR_TradeSide	交易方向错	-40530404
WD_ERR_MarketType	市场代码错	-40530405
WD_ERR_HedgeType	投机套保字段错	-40530406
WD_ERR_OrderListOver	委托队列满	-40530407
WD_ERR_OptionType	期权类别字段错	-40530408
WD_ERR_OptionUnderlyingCode	期权标的券字段错	-40530409
WD_ERR_CancelOrderNumber	撤单委托序号错	-40530501
WD_ERR_OrderStatusCannotCancel	不可撤单	-40530502
WD_ERR_QueryTooQuick	查询过于频繁	-40530601
WD_ERR_ResponseBufferUpdated	回报数据已被新请求覆盖	-40530602
WD_ERR_ResponseCountOver	获取回报数据超出记录数	-40530603
WD_ERR_MutliQueryCondition	多于一个的查询条件	-40530604
WD_ERR_NoRequestID	无对应请求流水号记录	-40530605
WD_ERR_NoOrderReport	无对应委托回报	-40530606
WD_ERR_OrderFailed	委托失败	-40530607
WD_ERR_OrderSent	委托已发送	-40530608
WD_ERR_CoveredQryCode	备兑查询标的券字段错	-40530609
WD_ERR_CoveredQryChg	备兑查询方向字段错	-40530610

5.5 Wind 回测接口函数明细

5.5.1 回测开始 bktstart

说明	回测开始 LONG bktstart(WindData& outWindData, LPCWSTR strategyName, LPCWSTR startDate, LPCWSTR endDate, LPCWSTR options = NULL);		
参数	定义	In/Out	描述
outWindData	WindData	Out	返回的数据
strategyName	LPCWSTR	In	支持中英文；长度不超过 256 字符。如：“全天候策略”。
startDate	LPCWSTR	In	起始时间和日期字符串。支持日期类工具函数输出的标准日期结果。

endDate	LPCWSTR	In	截止时间和日期字符串。可接受的字符串必须形如“yyyymmdd”，“yyyy-mm-dd”。
options	Period	In	设定交易周期，支持 1 分钟、5 分钟、15 分钟、1 天，缺省为 1 天。例：“1m”，“5m”，“15m”，“1d”
	InitialFund		设定账户初始资金，默认为初始资金1亿元。例：“InitialFund =100000000.00”
	FeeLevel		设定股票、基金佣金水平，例：“FeeLevel=0.001”
	FeeMulti		设定期货佣金倍数，例：“FeeMulti=3”
	BKTNotes		记录每次回测的特征信息，例： ” BKTNotes=螺纹钢期指套利，5min，2011.1-2013.9”
	StrategyNotes		记录每次回测的特征信息，例： “StrategyNotes=对冲套利 “
返回值	0 表示执行成功，负数表示失败；可以通过 <code>getErrorMsg</code> 函数取得错误信息。		
范例	<pre>//开启一个名字为 dropmost 的回测,时间从 2014 年 1 月 1 日到 6 月 27 日, 初始资金 1 亿元 WindData wd; LONG errCode = CWAPIWrapperCpp::bktstart (wd, _T("dropmost"), _T("20140101"), _T("20140627"), _T("InitialFund =100000000.00"));</pre>		

5.5.2 回测查询 bktquery

说明	回测查询 LONG bktquery(WindData& outWindData, LPCWSTR qrycode, LPCWSTR qrytime, LPCWSTR options = NULL);		
参数	定义	In/Out	描述
outWindData	WindData	Out	返回的数据
qrycode	LPCWSTR	In	持仓查询返回持仓代码、持仓数量、持仓成本价。例：“Capital”，“Position”。
qrytime	LPCWSTR	In	查询时间。例如“2011-04-01 9:30:00”。
options	LPCWSTR	In	扩展参数，多个参数用半角分号隔开。例如“option1=value1;option2=value2”可缺省，默认为空。
返回值	0 表示执行成功，负数表示失败；可以通过 <code>getErrorMsg</code> 函数取得错误信息。		
范例	查询回测中2014年4月1号的资金量 <pre>WindData wd; LONG errCode = CWAPIWrapperCpp::bktquery(wd, "Capital", "2014-04-01 09:30:00");</pre>		

5.5.3 回测下单 bktorder

说明	回测下单 LONG bktorder(WindData& outWindData, LPCWSTR tradeTime, LPCWSTR
----	---

	securityCode, LPCWSTR tradeSide, LPCWSTR tradeVol, LPCWSTR options = NULL);		
参数	定义	In/Out	描述
outWindData	WindData	Out	返回的数据
tradeTime	LPCWSTR	In	交易时间。例：“2011-04-01 9:30:00”，“20110401 09:30:00”
securityCode	LPCWSTR	In	交易品种列表。证券列表支持 Wind 代码及证券转换类工具函数输出的 Wind 代码结果，支持多品种向量输入。例：“600000.SH, IF.CFE, 510300.SH”
tradeSide	LPCWSTR	In	交易方向。 “Buy” 证券买入/期货买入开仓 “Short” 期货卖出开仓 “Cover” 期货买入平仓 “Sell” 证券卖出/期货卖出平仓 例：“Buy”，“Buy, Sell”
tradeVol	LPCWSTR	In	买卖数量。例：“100”，“100, 200”
options	LPCWSTR	In	Close 所选周期收盘价 Open 所选周期开盘价 High 所选周期最高价 Low 所选周期最低价 Avg 所选周期均价，对于日线 VWAP，为 (High+Low) / 2 例：“Price=Low”
返回值	0 表示执行成功，负数表示失败；可以通过 <code>getErrorMsg</code> 函数取得错误信息。		
范例	<pre>//回测中2014年6月25号10点对000001.sz和600000.sh以最低价下单100股和200股。 WindData wd; LONG errCode = CWAPIWrapperCpp:: bktorder(wd, "20140625 10:00:00", "000001.sz, 600000.sh", "buy", "100, 200", "Price=Low");</pre>		

5.5.4 回测结束 bktend

说明	<pre>//回测结束 LONG bktend(WindData& outWindData, LPCWSTR options);</pre>		
参数	定义	In/Out	描述
outWindData	WindData	Out	返回的数据
options	LPCWSTR	In	扩展参数，多个参数用半角分号隔开。例如“option1=value1;option2=value2”可缺省，默认为空。
返回值	0 表示执行成功，负数表示失败；可以通过 <code>getErrorMsg</code> 函数取得错误信息。		
范例	<pre>回测结束 WindData wd; LONG errCode = CWAPIWrapperCpp:: bktend(wd);</pre>		

5.5.5 回测结束 bktstatus

说明	查看回测状态 LONG bktstatus(WindData& outWindData, LPCWSTR options);		
参数	定义	In/Out	描述
outWindData	WindData	Out	返回的数据
options	StrategyID	In	策略 ID。例：“StrategyID =1365201”
	BKTID	In	回测 ID。例：“BKTID=201”
返回值	0 表示执行成功，负数表示失败；可以通过 <code>getErrorMsg</code> 函数取得错误信息。		
范例	<pre>//回测ID为2496的回测状态。 WindData wd; LONG errCode = CWAPIWrapperCpp::bktstatus("BKTID=2496");</pre>		

5.5.6 查看回测状态 bktsummary

说明	查看回测状态 LONG bktsummary(WindData& outWindData, LPCWSTR bktID, LPCWSTR view, LPCWSTR options);		
参数	定义	In/Out	描述
outWindData	WindData	Out	返回的数据
bktID	LPCWSTR	In	回测 ID。例：“1365201”
View	LPCWSTR	In	KPI 关键指标 NAV 每日净值 Trade 交易明细 Position 每日持仓 PositionRate 每日仓位 PL_Monthly 每月盈亏 例：“NAV”
options	LPCWSTR	In	扩展参数，多个参数用半角分号隔开。例如“option1=value1;option2=value2”可缺省，默认为空。
返回值	0 表示执行成功，负数表示失败；可以通过 <code>getErrorMsg</code> 函数取得错误信息。		
范例	<pre>回测ID为2499的每日净值概要。 WindData wd; LONG errCode = CWAPIWrapperCpp::bktsummary(wd,"2499","NAV");</pre>		

5.5.7 回测删除 bktdelete

说明	回测删除 LONG bktdelete(WindData& outWindData, LPCWSTR bktID, LPCWSTR options);		
参数	定义	In/Out	描述
outWindData	WindData	Out	返回的数据
bktID	LPCWSTR	In	回测 ID。例：“1365201”

options	LPCWSTR	In	扩展参数，多个参数用半角分号隔开。例如“option1=value1;option2=value2”可缺省，默认为空。
	StrategyID	In	策略ID。只有当bktID=0的时候才有效，可删除此策略ID下所有回测。例：“StrategyID =1365201”
返回值	0 表示执行成功，负数表示失败；可以通过 <code>getErrorMsg</code> 函数取得错误信息。		
范例	<pre>//删除ID为1365201的回测。 WindData wd; LONG errCode = CWAPIWrapperCpp::bktdelete(wd, “1365201”);</pre>		

5.5.8 返回策略列表 bktstrategy

说明	返回策略列表 <code>LONG bktstrategy(WindData& outWindData, LPCWSTR options);</code>		
参数	定义	In/Out	描述
outWindData	WindData	Out	返回的数据
options	LPCWSTR	In	扩展参数，多个参数用半角分号隔开。例如“option1=value1;option2=value2”可缺省，默认为空。
	Type	In	0 查询自己创建的策略 1 查询已关注的策略 2 查询其他用户共享的策略 例：“Type=1”
返回值	0 表示执行成功，负数表示失败；可以通过 <code>getErrorMsg</code> 函数取得错误信息。		
范例	返回自己创建的策略列表。 <pre>WindData wd; LONG errCode = CWAPIWrapperCpp::bktstrategy(wd,“Type=1”);</pre>		

6 Wind C++接口获取帮助途径

用户可以通过如下方式获取 Wind C++帮助。

6.1 本用户手册

本用户手册会介绍产品功能和用户经常反馈的信息。对于第一次使用者来说，请从本手册开始浏览。

6.2 命令生成器

用户可使用命令生成器熟悉 C++ 接口及命令。命令生成器的文件位置为 C:\Wind\Wind.NET.Client\WindNET\bin\WindNavigator。

6.3 大奖章网站

大奖章网站 <http://www.dajiangzhang.com> 是 Wind 推出的量化社区，里面提供了和 Wind 量化接口相关的各种资料。

6.4 量化交易群

用户可以通过 iWind 申请加入中国量化交易群(群号 59289)，学习 Wind C++接口使用和量化交易知识。

6.5 终端量化交易平台

可以在终端中输入 QNT 进入“量化交易平台”功能模块。有大量的示例及其他相关内容在此发布，可参考。