

INSIGHT-UDP-CPP 客户端用户手册

copyright©Huatai Securities

V1.10.0-2022/11/02

-
- [用户许可协议](#)
 - [版权与所有权声明](#)
 - [用户许可协议声明](#)
 - [许可协议条款](#)
 - [许可协议终止](#)
 - [适用法律](#)
 - [免责条款](#)
 - [联系方式](#)
- [版本变动](#)
- [介绍](#)
 - [API 结构](#)
 - [通讯模式](#)
- [开发接口](#)
 - [ClientFactory 接口](#)
 - [Instance 方法](#)
 - [Uninstance 方法](#)

- [CreateUdpClient 方法](#)
- [UdpClientInterface 接口](#)
 - [RegistHandle 方法](#)
 - [SetWorkPoolThreadCount 方法](#)
 - [LoginById 方法](#)
 - [SubscribeAll 方法](#)
 - [SubscribeBySourceType 方法](#)
- [MessageHandle 接口](#)
 - [OnMarketData 方法](#)
 - [OnGeneralError 方法](#)
 - [OnLoginSuccess 方法](#)
 - [OnLoginFailed 方法](#)
 - [OnNoConnections 方法](#)

用户许可协议

版权与所有权声明

INSIGHT 由华泰证券股份有限公司设计、开发。系统及其文档的所有权归属于华泰证券股份有限公司（以下简称“华泰证券”或者“本公司”），并受中华人民共和国国家《著作权法》、《商标法》和国际协约条款的保护。由华泰证券股份有限公司负责系统的更新、维护和销售等活动。用户不得从本系统中删去版权声明，要保证为本系统的拷贝（全部或部分）复制版权声明，并同意制止以任何形式非法拷贝本系统及文档。未经授权擅自复制或散布本数据库的部分或全部内容，将会面对民事起诉。

“INSIGHT”的名称已受到注册商标和其它形式的所有权的保护。

用户许可协议声明

本协议一方为本系统的个人或机构使用者，另一方为华泰证券股份有限公司。用户使用本系统之前，须首先认可本许可协议，如持有异议，请不要使用，并于 30 日内删除安装软件并到本公司办理有关事宜。

本说明书中使用的“本产品”是指计算机软件、数据库、记录媒体、印刷品及在线文档或电子文档，包括收录于本软件产品的所有数据、资料文件、执行文件、附加功能、使用说明书、客户方案书、帮助文件及其它文件，且可能并不只限于以上内容。以下出现的“使用”是指在计算机硬盘或在 RAM、CD-ROM 等其它储存设备上存储、安装、执行或显示本软件产品于屏幕的行为。

许可协议条款

- 本系统仅给您提供唯一使用许可权。用户必须承诺不把本系统提供的全部或部分资料和数据以任何形式转移、出售和公开给任何第三者。
- 用户必须同意并保证，采取必要和合适的措施保护本系统提供的资料和数据版权和所有权。
- 用户必须通知其所有相关使用者有关本系统的版权声明和本许可协议，并要求所有相关使用者都必须遵循本许可协议的一切条款。
- 用户必须同意在本许可协议终止前，一直承担本协议所要求的一切责任和义务。

许可协议终止

用户若违反本协议的任一条款或条件，华泰证券可以即时终止其使用许可。一旦许可权利被终止，您必须立即销毁本系统及文档的所有拷贝，或将其归还本公司。

适用法律

中华人民共和国《知识产权保护条例》、《著作权法》、《商标法》、《专利法》等。

免责条款

华泰证券尽力为用户提供可信的、准确的资料和数据，但无法完全保证其百分之百的准确和完整。因此，无论在什么情况下，由使用本系统所产生的任何形式的间接或直接的、特别或意外的、必然或偶然的损失和破坏，本公司概不负责。在上述情况发生时，即使本公司事先被告知此类事情有可能发生，本公司亦不对由此导致的任何后果承担责任。

本公司将尽快更新资料数据，但不承担由于使用数据资料延误造成的损失或责任。如果用户发现数据文件中的错误，请立即通知本公司，本公司将尽最大的努力在下一个版本中更正。

联系方式

如果用户对本协议条款有任何疑问，请按照如下方式与本公司联系：

华泰证券股份有限公司

地址：江苏省南京市建邺区江东中路 228 号华泰证券广场

邮编：210019

电话：95597

传真：025-83387416

电子邮件：service@htsc.com

业务联系方式：

电子邮件：insight@htsc.com

版本变动

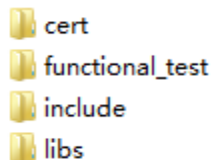
版本号	发布时间	变动项
UDP-1.1.0	2020-8-10	<div>1. 新增动态链接库，boost_system, boost_thread, boost_chrono, Disruptor;</div> <div>2. 登录接口 LoginById, 新增参数备选地址列表 backup_list;</div> <div>3. 订阅接口 SubscribeAll、SubscribeBySourceType 返回值类型由 bool 变为 int 类型;</div> <div>4. 新增登录成功回调 OnLoginSuccess, 新增单个服务器登录失败回调 OnLoginFailed, 新增所有服务器登录失败回调 OnNoConnections;</div>

UDP-1.2.0	2020-9-14	<ol style="list-style-type: none"> 1. 移除动态链接库，boost_system, boost_thread, boost_chrono, Disruptor; 2. 修复入参为 std::string 时拷贝构造函数 bug;
UDP-1.5.0	2021-10-13	<ol style="list-style-type: none"> 1. 适配上交所新债券模型; 2. 增加异常日志及增加 Subscribe 时的异常捕捉; 3. 提升 Aeron 版本;
UDP-1.6.0	2021-11-29	<ol style="list-style-type: none"> 1. 支持上游组包发送功能，增加拆包逻辑 2. 新增多路优选功能，支持重复消息过滤。新增 open_double_select() close_double_select() is_double_select() 对本功能进行配置 3. 新增组播数据拉取线程的 idle sleep 策略
UDP-1.7.0	2022-2-25	<ol style="list-style-type: none"> 1. 适配基金通模型
UDP-1.8.0	2022-4-13	<ol style="list-style-type: none"> 1. 新增对深交所新债券交易系统支持
UDP-1.9.0	2022-6-7	<ol style="list-style-type: none"> 1. 适配 Aeron 保障最慢策略的逻辑调整 2. 调整上报的数据情况，新增解压缩后、过滤的数据条数 3. 准确上报 SDK 版本信息
UDP-1.10.0	2022-11-2	<ol style="list-style-type: none"> 1. 新增高精度的 IOPV 数据字段：HighAccuracyIOPV 、HighAccuracyPreIOPV

介绍

API 结构

INSIGHT-UDP 行情客户端系统 API 是一个基于 C++ 的类库，通过使用和扩展类库提供的接口来实现行情数据的接收。该类库目录结构如下：



include 文件夹为 API 所需的一系列数据结构及接口头文件，libs 文件夹为 API 所需的导入库文件，functional_test 文件夹为 API 使用的示例程序，cert 文件夹为 Gateway 客户端进行认证时的密钥文件，UDP 用户可以忽略。

INSIGHT-UDP API 对系统环境有较高要求。用户需要配置系统的网络发送和接收缓冲不小于 10MB，示例配置命令如下：

```
sudo sysctl net.core.rmem_max=10485760
sudo sysctl net.core.wmem_max=10485760
```

本 API 支持 Linux 平台中 GCC4.8 及以上版本编译器，其中 GCC4.8 为标准版本，其他版本需要联系 INSIGHT 获取。

通讯模式

INSIGHT 的所有通讯都基于某个通讯模式，通讯模式实际上就是通讯双方协同工作的方式。

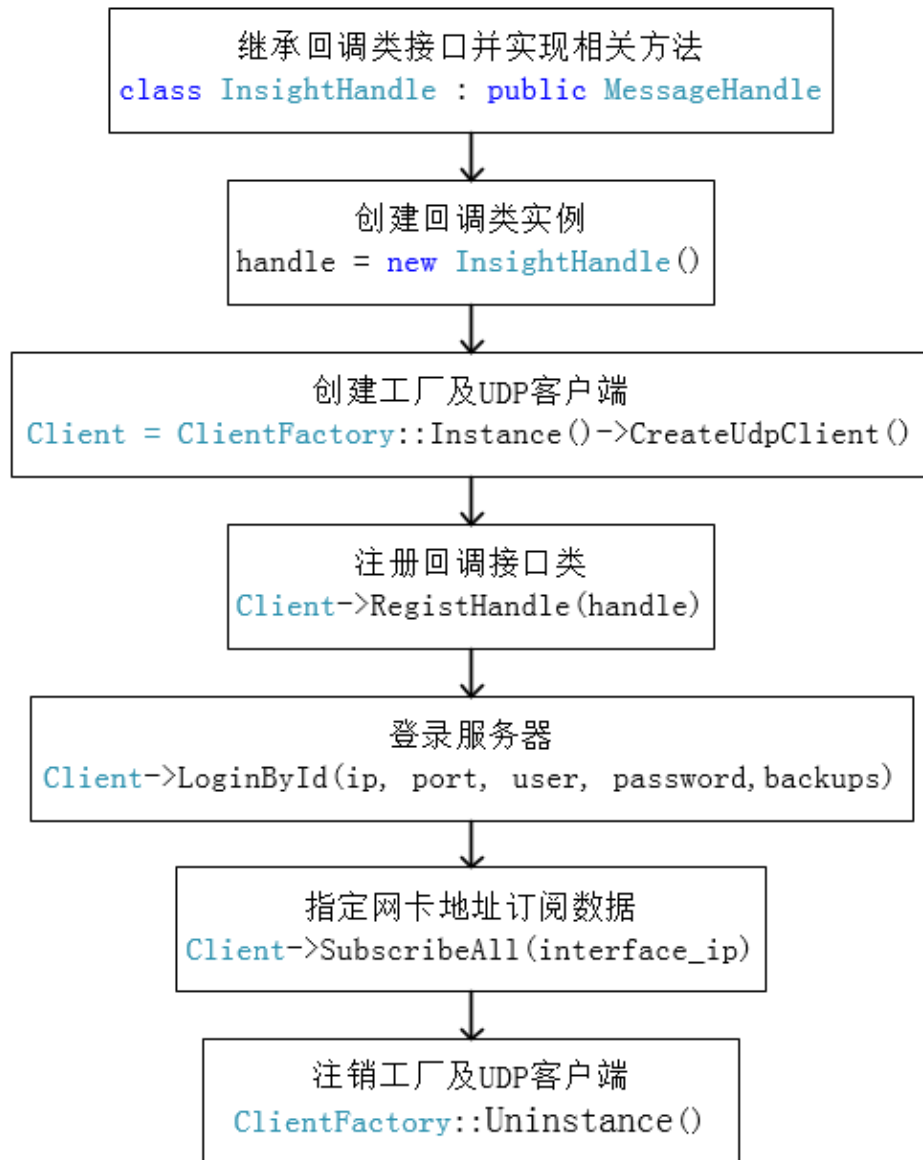
行情发布涉及的通讯模式共有两种：

- 对话通讯模式
- 广播通讯模式

INSIGHT-UDP API 使用广播通讯模式，该模式指服务端向位于同一网络的客户端以 UDP 形式广播相同的行情信息。

开发接口

INSIGHT-UDP 行情系统 API 采用工厂模式，ClientFactory 类用于创建 UDP 客户端，返回的 UdpClientInterface 为 UDP 客户端的接口类，可通过 UdpClientInterface 类型指针进行登录、订阅操作。行情数据通过回调函数的方式返回，用户需要继承 MessageHandle 类用于处理相关数据，MessageHandle 的子类实例需要在 UDP 客户端登录前进行注册。具体使用流程如下：



ClientFactory 接口

ClientFactory 为工厂类，通过工厂类创建 UDP 客户端实例，同时在使用结束后通过工厂类销毁 UDP 客户端实例。

Instance 方法

用于创建 ClientFactory 类型单例。

函数原型：

```
static ClientFactory* Instance()
```

@param[out] ClientFactory* 工厂类指针

Uninstance 方法

销毁 ClientFactory 类型单例及其创建的客户端实例。

函数原型:

```
static void Uninstance();
```

CreateUdpClient 方法

创建 UDP 客户端单例，返回 UDP 客户端接口类指针。

函数原型:

```
UdpClientInterface* CreateUdpClient()
```

@param[out] UdpClientInterface* UDP 客户端接口类指针

UdpClientInterface 接口

UdpClientInterface 为 UDP 客户端接口类，用于向服务端发送登录和订阅请求，并注册处理句柄。

RegistHandle 方法

用于注册回调类 MessageHandle 指针，需要在创建 UDP 客户端实例之后，UDP 客户端登录之前调用。

函数原型:

```
virtual void RegistHandle(MessageHandle* handle) = 0;
```

@param[in] handle 回调类指针

SetWorkPoolThreadCount 方法

设置用于分发数据的线程个数，此类线程会按照数据类型调用 MessageHandle 类的相关接口。

函数原型:


```
virtual void SetWorkPoolThreadCount(short count) = 0;
```

@param[in] count 分发线程数量

LoginById 方法

根据账户密码登录配置服务器，获取组播地址。

函数原型：

```
virtual int LoginById(std::string ip, int port, std::string user_name,  
std::string password, std::map<std::string, int>backup_list) = 0;
```

@param[in] ip	配置服务器地址
@param[in] port	配置服务器端口
@param[in] user_name	用户名
@param[in] password	密码
@param[in] backup_list	备用服务器地址和端口
@param[out] int	登录结果，=0 表示成功，<0 表示失败

SubscribeAll 方法

使用指定网卡地址来订阅所有的组播数据。

函数原型：

```
virtual int SubscribeAll(std::string interface_ip) = 0;
```

@param[in] interface_ip	用于订阅组播的网卡 ip
@param[out] int	订阅结果，0 表示成功，<0 表示失败

SubscribeBySourceType 方法

使用指定网卡地址来订阅指定类型数据。

函数原型：

```
virtual int SubscribeBySourceType(std::string  
interface_ip, ::com::htsc::mdc::insight::model::SubscribeBySourceType*  
source_type) = 0;
```

@param[in] interface_ip 用于订阅组播的网卡 ip
@param[in] source_type 订阅数据类型，具体装填方式可参考 functional_test
@param[out] int 订阅结果，0 表示成功，<0 表示失败

MessageHandle 接口

回调接口类，SDK 将调用改类接口方法进行数据分发，需要在 UDP 客户端登录前注册改类型指针。

OnMarketData 方法

该方法用于分发 UDP 组播数据，根据 SetWorkPoolThreadCount 方法中的数量设置启动的分发线程数量，不设置默认为 5 个分发线程。对于单个标的而言，是线程安全的。

函数原型：

```
virtual void OnMarketData(const com::htsc::mdc::insight::model::MarketData& data);
```

@param[in] data 为 protobuf 类型，MarketData 具体字段可参考数据手册

OnGeneralError 方法

当订阅的组播通道出现心跳超时，会调用该方法，为避免影响后续处理逻辑，请不要在此处进行复杂的业务逻辑

函数原型：

```
virtual void OnGeneralError(const com::htsc::mdc::insight::model::InsightErrorContext& context);
```

@param[in] context 错误内容，InsightErrorContext 的访问方式可参考 functional_test 以及 InsightErrorContext.pb.h

OnLoginSuccess 方法

当登录成功时，会调用该方法，为避免影响后续处理逻辑，请不要在此处进行复杂的业务逻辑

函数原型：

```
virtual void OnLoginSuccess();
```

OnLoginFailed 方法

当单个服务器登录失败时，会调用该方法，为避免影响后续处理逻辑，请不要在此处进行复杂的业务逻辑

函数原型：

```
virtual void OnLoginFailed(int error_no, const std::string& message);
```

@param[in] error_no 错误号

@param[in] message 错误原因

OnNoConnections 方法

当主服务地址加备服务地址均登录失败时，会调用该方法，为避免影响后续处理逻辑，请不要在此处进行复杂的业务逻辑

函数原型：

```
virtual void OnNoConnections();
```