

作者：景色(<https://blog.csdn.net/pjjing>)

公众号：程序化交易入门(QuantRoad2019)

文章结尾扫描二维码关注公众号，一起学习程序化交易！

一、背景

详细背景可以参考证监会 [《关于进一步加强期货经营机构客户交易终端信息 采集有关事项的公告》](#) 及期货市场监控中心 [《期货公司客户交易终端信息采集及接入认证技术规范》](#) 这两篇公告。一句话总结下就是：监控中心为了方便监管，需采集所有通过期货公司入场交易的客户的本地终端信息。

说明如下：

- 这个采集是所有柜台（是指期货公司交易平台）发布的API都会采集，并不是说CTP采集，其他主席如金仕达不采集，次席系统如闪策，易盛等不采集。只要在监控中心报备过的平台厂商，都必须提供这个功能。对应的用了这些平台提供的API的终端，例如快期等，自然会采集客户使用时的电脑的相关信息。
- 采集什么信息在上述第二篇公告里面都写的十分清楚，例如IP，MAC，操作系统版本，硬盘序列号，系统分区等，并不是如某些危言耸听，会扫描硬盘文件采集文件信息等。另外采集到的信息在API底层已经用监控中心下发的公钥加密，只有监控中心才有私钥解密，柜台厂商、期货公司、交易所都无法看到明文。

二、术语

- 直连模式
是指客户直接连到期货公司的柜台进行交易。投资者直接用CTP的API开发交易程序，连到期货公司的CTP系统交易，这就是典型的直连模式。也就是说，市场上大多数程序化交易投资者，直接使用柜台提供的API交易的，都是属于这种直连模式。另外像快期这种终端，也是直接连到CTP交易，所以也属于直连模式。
- 中继模式
是指客户通过第三方软件，向期货公司CTP系统报单交易，这时客户在第三方软件上下单，首先订单是到了第三方软件厂商的服务器，然后第三方软件厂商再调用CTP系统的API，将这笔订单发往CTP系统。这第三方软件厂商便称之为中继。
- 多对多中继模式
第三方软件厂商服务器为每个投资者都调用CTP API创建与CTP之间的单独的链接就是多对多的中继模式。例如文华财经的云条件单的功能，其实是将投资者的订单保存到了文华的云服务器上，投资者虽然退出文华终端，但文华服务器与CTP的链接仍然在，所以条件达到就可以把订单发送到CTP。
- 多对一中继模式
第三方软件厂商只创建一个与CTP之间的链接，用这个链接为所有投资者报单，这就是多对一模式。这种情况下需要第三方软件商以交易员用户登录CTP系统，而且只能为该交易员底下的投资者报单。这种模式多是由于期货公司内部软件和境外中介。

三、需要投资者做什么

3.1 使用第三方厂商客户端的投资者

例如直接使用快期，博易，无限易等软件进行交易的投资者，这些人只需要直接升级软件版本，其他什么都不需要做的，因为第三方厂商已经向期货公司报备APPID，在软件内部逻辑已经写好。

3.2 使用CTP API自开发交易程序的直连投资者

其实很简单，一句话就是，在ReqUserLogin之前需要先调用ReqAuthenticate。下面详细讲一下步骤。

3.2.1 向期货公司申请AuthCode（认证码）

自己查看开户期货公司官网，或者联系对应的经纪经理，申请认证码。通常来说会让你填一张表，示范如下：

header 1	header 2
客户号	000001
软件名称	**量化软件
APPID	abcdef_ghijkl_1.6.0
连接方式	直连类型终端

注意：

- 客户号就是你的交易账号
- APPID是你自己命名的，命名监控中心给出了规范，如下图：

AppID/RelayAppID格式规范

字段名称	终端/中继厂商名称	终端/中继软件名	版本号
最大长度	10	10	8
格式	终端/中继厂商名称_终端/中继软件名称_版本号		
示例	abcdef_ghijkl_1.6.0		
特殊说明	个人开发的终端软件，厂商名称为client		

AppID
RelayAppID

1. AppID/RelayAppID在期货公司内部不能重复。
2. 同一终端软件/中继软件在不同期货公司，AppID/RelayAppID应该保持一致。
3. 个人开发的终端软件，厂商名称为client。

给出几个范例：

个人开发终端：client_abcdf_1.0.0

第三方终端：pobo_iee_1.4.0.0

中继厂商：poboAPP_issle_2.0.0.0

申请通过后，期货公司就会发送Authcode给投资者。

3.2.2 修改程序登录逻辑

- step1:在API连接后发起认证

```
void CUser::OnFrontConnected()
{
    cout << "OnFrontConnected." << endl;
    static const char *version = m_pUserApi->GetApiVersion();
    cout << "-----当前版本号 : " << version << " -----" << endl;
```

```

    ReqAuthenticate();
}
int CUser::ReqAuthenticate()
{
    CThostFtdcReqAuthenticateField field;
    memset(&field, 0, sizeof(field));
    strcpy(field.BrokerID, "8000");
    strcpy(field.UserID, "001888");
    strcpy(field.AppID, "XY_Q7_V1.0.0");
    strcpy(field.AuthCode, "5A5P4V7AZ5LCFEAK");
    return m_pUserApi->ReqAuthenticate(&field, 5);
}

```

注意：

AppID就是客户申请时自己命名的，AuthCode是期货公司发给客户，两者是一一对应的。其实ReqAuthenticate函数早就有了，但以前是通过UserProductInfo和AuthCode这两个字段绑定验证，现在UserProductInfo废弃不用，改为AppID了，填写时注意不要出错。

- step2:认证成功后发起登录

```

void CUser::OnRspAuthenticate(CThostFtdcRspAuthenticateField
*pRspAuthenticateField, CThostFtdcRspInfoField *pRspInfo, int nRequestID, bool
bIsLast)
{
    printf("OnRspAuthenticate\n");
    if (pRspInfo != NULL && pRspInfo->ErrorID == 0)
    {
        printf("认证成功,ErrorID=0x%04x, ErrMsg=%s\n\n", pRspInfo->ErrorID,
pRspInfo->ErrorMsg);
        ReqUserLogin();
    }
    else
    {
        cout << "认证失败, " << "ErrorID=" << pRspInfo->ErrorID << " ,ErrMsg=" <<
pRspInfo->ErrorMsg << endl;
    }
}
int CUser::ReqUserLogin()
{
    printf("====ReqUserLogin====,用户登录中...\n\n");
    CThostFtdcReqUserLoginField reqUserLogin;
    memset(&reqUserLogin, 0, sizeof(reqUserLogin));
    strcpy(reqUserLogin.BrokerID, "8000");
    strcpy(reqUserLogin.UserID, "001888");
    strcpy(reqUserLogin.Password, "1");
    return m_pUserApi->ReqUserLogin(&reqUserLogin, ++RequestID);
}

```

然后就可以在OnRspUserLogin看是否登录成功了。

3.3 使用中继模式，自开发程序下单的投资者

这种主要是第三方软件厂商的工作。而且如果是程序化客户，投资者使用的是不同的第三方厂商提供API，用法也不一致，这里就不细说了。总之，第三方厂商提供的API会用到CTP API包中的DataCollect.h中的函数来采集客户端信息，再在服务器上注册提交给CTP柜台。

四、常见问题

1. API包中多了的DataCollect.h和LinuxDataCollect.so有什么用？

参见上文3.3，是转为中继模式准备的。同样API头文件中新增的两个函数RegisterUserSystemInfo和SubmitUserSystemInfo也是为中继准备的，一般程序化客户都用不到。

2. 是否必须要更新为穿透式版本？

答案是必须。不同于之前的CTP API版本都是向后兼容，后期CTP柜台升级为穿透式版本后，以前所有的API将都不能使用，只能使用最新的6.3.15版本API才能登陆上。根据目前消息，2019年6月14日（含）之后，所有的客户都必须使用6.3.15版本才可以登陆上，否则只能去期货公司页面下载最新软件手动下单了。

3. OnFrontConnected没有回调怎么办？

为了强制客户使用最新的穿透式API，如果客户使用API版本与CTP柜台不一致，将不会回调OnFrontConnected。目前一共有三类CTP API：旧生产版（6.3.11_20180109及以前）；穿透式评测版（6.3.13_20181119和6.3.16_T1_20190508，这两个版本都可以用来评测）；穿透式生产版（6.3.15_20190220）。这三个版本需要配合不同的CTP柜台才能登录上，也就是说一个前置IP地址，你能用某个登录上，就不可能用另外两个登录上。客户无法从API得知连接到的CTP后台前置是什么版本，所以只能用这三个版本的API自己测试得知。现在为止simnow的模拟地址：

BrokerID统一为：9999

标准CTP：

第一组：Trade Front:180.168.146.187:10100,Market Front:180.168.146.187:10110;电信
（看穿式前置，API同生产v6.3.15_20190220）

第二组：Trade Front:180.168.146.187:10101,Market Front:180.168.146.187:10111;电信
（看穿式前置，API同生产v6.3.15_20190220）

第三组：Trade Front:218.202.237.33 :10102,Market Front:218.202.237.33 :10112;移动
（看穿式前置，API同生产v6.3.15_20190220）

也就是说需要用v6.3.15_20190220版本可以连上，用户注册后，默认的APPID为simnow_client_test，认证码为0000000000000000（16个0）。

同时simnow也提供了穿透式评测版后台供测试，参见[《公告：SIMNOW平台将开启终端厂商穿透测试环境》](#)，这个提供的地址

BrokerID: 9999

Trade Front:180.168.146.187:13030

Market Front: 180.168.146.187:13040

需要用6.3.13_20181119连上。

至于穿透式生产版，请各位联系期货公司获取相应地址测试。

关注公众号，一起学习程序化交易！



<https://blog.csdn.net/yishuihan1212>